

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

TRACE MODE 6

Интегрированная SCADA/HMI-SOFTLOGIC-MES-
EAM-HRM-система для разработки АСУ ТП,
АСКУЭ и систем управления производством

Том 1

18-е издание (к релизу **6.10**)

ТУ 5043-001-18957709-2015

Москва, 2015

AdAstra Research Group, Ltd.

TRACE MODE, T-FACTORY.exe, АдАстра, AdAstrA, autobuilding и автопостроение являются зарегистрированными торговыми марками AdAstra Research Group, Ltd.

Торговые марки, торговые имена, сервисные марки и сервисные имена, зарегистрированные другими компаниями и использованные в данном руководстве, принадлежат соответствующим компаниям.

© 1998-2015 AdAstra Research Group, Ltd.

Все права защищены.

Содержание

Глава 1	11
Принципиальные отличия компонентов TRACE MODE 6	11
Модификации программных компонентов TRACE MODE 6	12
Модификации ИС TRACE MODE 6	12
Модификации мониторов	13
Ограничения мониторов для Windows CE	13
Особенности LinMPB	14
Глава 2	17
Проект АСУ в TRACE MODE 6	17
Программный комплекс TRACE MODE 6	18
Основные характеристики TRACE MODE 6	19
Принцип работы монитора. Канал TRACE MODE 6	19
Обеспечение работы распределенных АСУ	20
Резервирование	22
Автопостроение	22
Математическая обработка данных	23
Архивирование каналов узла	23
Архивирование каналов проекта	24
Обработка аварийных ситуаций	24
Отчет тревог и генерация сообщений	25
Файл восстановления	25
Графический интерфейс оператора	25
Генерация документов (отчетов)	26
Защита проекта	26
Конвертирование проектов	26
Документирование проекта TRACE MODE	27
Системы встроенного контроля	27
Принципы разработки проекта в ИС	28
Технология разработки проекта в ИС	29
Классификация объектов структуры проекта	31
Классификация компонентов	31
Классификация слоев	35
Классификация узлов	36
Назначение групп источников (приемников)	38
Группа 'PC-based контроллеры'	39
Группа 'Распределенные УСО'	40
Группа 'Платы ввода-вывода'	41
Группа 'Терминалы'	43
Группа 'PLC'	43
Группа 'ОПС-группа'	43
Группа 'DDE-группа'	44
Группа 'MODBUS-группа'	44
Группа 'Пользовательские драйверы'	44
Группа 'Диагностика и сервис'	44
Группа 'Генераторы'	45
Группа 'Модели'	45

Назначение других групп	50
Группа 'СОМ-порты'	50
Группа 'Словари сообщений'	50
Группы слоя 'Библиотеки компонентов'	50
Группы клемм	51
Прочие группы	51
Операции в ИС	53
Меню и панель инструментов ИС	53
Меню 'Файл' и главная панель инструментов ИС	53
Меню 'Вид'	55
Меню 'Окна'	55
Панель переходов между окнами ИС	56
Задание общих настроек ИС	56
Вкладка 'Интегрированная среда разработки'	57
Вкладка 'Базовый редактор'	61
Сохранение проекта	62
Сохранение проекта для редактирования	62
Сохранение проекта для запуска	63
Файлы узла, создаваемые при экспорте	63
Редактирование структуры проекта	65
Меню и главная панель инструментов навигатора проекта	65
Управление внешним видом навигатора проекта	67
Создание объектов структуры	68
Имена и идентификаторы объектов структуры	68
Изменение класса канала после его создания	69
Выделение объекта структуры	69
Удаление объекта структуры	70
Открытие окна свойств и редактора объекта структуры	70
Групповое редактирование компонентов	70
Операции, общие для объектов структуры	70
Перемещение объектов структуры	71
Копирование и вставка объекта структуры	71
Перепривязка каналов и аргументов при копировании/вставке и перемещении объекта базы каналов	73
Взаимодействие с технологической БД	74
Отображение свойств объектов структуры	76
Окно свойств объекта структуры проекта	78
Вкладка 'Информация'	78
Вкладка 'Флаги'	79
Вкладка 'Аргументы'	81
Вкладка 'Атрибуты'	81
Редакторы объектов структуры проекта	82
Редакторы компонентов	83
Редакторы каналов	83
Редактор шаблонов программ	84
Редактор шаблонов экранов	84
Редактор шаблонов документов	84
Редактор связей с базами данных	84
Редактор библиотек текстов	84
Редактор библиотек изображений	84
Редактор библиотек видеоклипов	84
Редактор параметров СОМ-порта	84

Редактор словарей сообщений.....	93
Редактор клемм.....	95
Редакторы источников (приемников).....	96
Редактор системных переменных TRACE MODE.....	96
Редактор переменной OPC.....	96
Редактор переменной OPC HDA	98
Редактор переменной DDE	98
Редакторы групп компонентов	100
Редактор группы шаблонов экранов.....	100
Базовый редактор группы компонентов	100
Редактор группы 'OPC-сервер'.....	105
Редактор группы 'OPC HDA сервер'.....	106
Редактор группы каналов слоев 'Технология' и 'Топология'	106
Редактор параметров узла.....	108
Вкладки редактора узла	108
Задание параметров узла	108
Табличный редактор аргументов	119
Поля редактора аргументов	119
Специальная обработка аргументов HANDLE	121
Разновидности редактора аргументов	121
Редактирование параметров группы аргументов	123
Привязка аргументов вручную	123
Автопостроение и автопривязка каналов из редактора аргументов.....	125
Автопостроение каналов по команде редактора аргументов	125
Флаг SL.....	126
Замена виртуальной привязки аргументов в канале CALL.....	127
Замена привязок аргументов канала CALL на привязки аргументов другого канала CALL	127
Флаг HW	127
Флаги NP и PO	127
Автопостроение и автопривязка аргументов.....	128
Настройка параметров редактора аргументов.....	129
Корректировка проекта в реальном времени.....	130
Перезагрузка узла из ИС.....	130
Регистратор аварийных событий	132
Глава 3	135
Каналы и системные переменные TRACE MODE 6	135
Пересчет базы каналов.....	136
Потоки монитора	136
Время цикла монитора.....	138
Период пересчета канала.....	139
Особенности пересчета и обработки каналов.....	143
Общие атрибуты каналов.....	145
Атрибуты каналов, отображаемые профайлером.....	149
Числовые каналы	160
Канал класса FLOAT	160
Границы и интервалы канала FLOAT.....	163
Обработка в канале FLOAT	164
Канал класса DOUBLE FLOAT	168
Канал класса HEX16.....	169

Канал класса HEX32.....	171
Канал класса TIME.....	173
Канал класса CALL.....	175
Атрибуты канала класса CALL.....	175
Канал CALL.Program.....	178
Канал CALL.SQLQuery.....	179
Канал CALL.Document(Report).....	179
Канал CALL.Screen.....	179
Канал CALL.Panel.....	179
Канал CALL.ChGroupReq.....	179
Канал CALL.TableFunction.....	182
Канал CALL.LongFromBits.....	182
Канал CALL.BitsFromLong.....	184
Канал CALL с типами вызова 12-14.....	185
Канал CALL.Sum.....	186
Канал CALL.Set.....	191
Канал CALL.RT_Statistics.....	191
Канал CALL.MOVE.....	193
Канал CALL.MResource_1.....	194
Каналы CALL.LArc0 и CALL.LArc1.....	194
Канал CALL.DifSnap.....	194
Канал CALL.LocalStatistics.....	194
Канал CALL.LocalQuick.....	194
Канал CALL.LocalList.....	195
Канал CALL.LocalQuick-T.....	195
Канал CALL.MUX.....	195
Каналы CALL.RemArc0 и CALL.RemArc1.....	196
Канал CALL.AsyncCollection.....	196
Канал CALL.Writer.....	197
Канал CALL.DI_DO.....	198
Канал CALL.MLink.....	199
Канал CALL.EvReg.....	200
Канал CALL.Exec.....	200
Канал CALL.TVC.....	200
Канал CALL.OtherProj.....	202
Канал CALL.STRING.....	203
Канал CALL.Vector.....	203
Канал CALL.ROOT.....	210
Каналы CALL.AS_DOCUMENT и CALL.AS_DATA.....	214
Канал CALL.EXCEPT_DAY.....	214
Канал CALL.EMAIL.....	215
Запись вектора в CALL.ChGroupReq и CALL.TVC.....	216
Запись длинных строк в канал CALL.....	216
Перепривязка аргументов канала CALL.....	218
Канал класса СОБЫТИЕ.....	220
Системные переменные TRACE MODE 6.....	226
Группа СИСТЕМНЫЕ.....	227
Группа ДИАГНОСТИКА.....	256
Канал класса ПОЛЬЗОВАТЕЛЬ.....	276
Глава 4.....	281
Распределенные АСУ.....	281

Конфигурирование межкомпонентного взаимодействия	282
Связь канал-канал	282
Связь через аргументы	285
Особенности взаимодействия	287
Допустимые цепочки связей	288
Создание контура управления	288
Специальные виды связи	288
Обмен по TCP/IP	289
Механизм предотвращения потерь при обмене по сети	289
Обмен по телефонным линиям	290
Резервирование в TRACE MODE 6	292
Синхронизация резервов	293
Структурное резервирование и надежность	295
Резервирование УСО	296
Обмен с базами данных	297
Синтаксис SQL	297
Настройка параметров редактора связей с БД	297
Подключение к базе данных	298
Окно 'Схема'	299
Окно 'Аргументы'	299
Создание и выполнение SQL-запросов	300
Создание SQL-запросов	300
Подстановки в SQL-запросе	300
Выполнение SQL-запросов из ИС	302
Выполнение SQL-запросов в реальном времени	302
Создание SQL-запросов с помощью мастера	306
Построение запроса SELECT с помощью мастера	306
Построение запроса INSERT с помощью мастера	312
Построение запроса UPDATE с помощью мастера	315
Построение запроса DELETE с помощью мастера	319
Сочетания клавиш в редакторе связей с БД	321
Обмен по DDE и NetDDE	322
Обмен по DDE	322
Обмен приложение - MPB как DDE-сервер	322
Обмен MPB как DDE-клиент - приложение	326
Обмен по NetDDE	328
Настройка ОС для обмена по NetDDE	328
Обмен приложение - MPB как NetDDE-сервер	328
Обмен между MPB по NetDDE	330
Обмен по OPC	334
OPC-сервер TRACE MODE 6	334
MPB как клиент сервера OPC HDA	336
TRACE MODE Data Center	337
Хранилище Data Center	337
Консоль Data Center	337
Меню и главная панель инструментов консоли	338
Диалог конфигурирования Data Center	338
Вкладка 'Сервер'	339
Вкладка 'Публикатор'	339
Вкладка 'Веб-консоль'	341
Вкладка 'Принтер'	342

Особенности выполнения узла в браузере	342
Запуск узла в мобильном устройстве	343
Сервер печати TRACE MODE 6	344
Глава 5.....	345
Обмен с контроллерами и УСО.....	345
Средства поддержки оборудования	346
Шаблоны каналов обмена	346
Обмен по встроенным протоколам.....	348
Обмен по DCS	348
Пользовательские SMS.....	348
Обмен по MODBUS	350
Прибор 'Взлет МР'	351
Контроллеры ROC 827 и FloBoss 107.....	352
Обмен по встроенным протоколам по сети.....	353
Обмен по Modbus TCP/IP	359
Обмен по SNMP	359
Обмен с устройствами Mitsubishi.....	359
Обмен с устройствами OMRON.....	359
Обмен по IEC 60870-101	359
Обмен по IEC 60870-104	364
Чтение штрих-кодов и магнитных носителей.....	368
Обмен по произвольному протоколу по RS	370
Обмен по протоколу HydraLink	371
Универсальный механизм обмена с электросчетчиками	376
МРВ как сервер протоколов поверх TCP/IP.....	399
МРВ как сервер MODBUS TCP/IP	399
МРВ как сервер протокола МЭК 60870-104.....	400
Поддержка протокола NMEA.....	404
Управление и диагностика обмена.....	406
Обмен через драйверы.....	409
Обмен с контроллером ЛОМИКОНТ.....	410
Обмен с контроллером РЕМИКОНТ-130.....	411
Обмен с контроллерами Ш711	413
Обмен с контроллером Ш9327	414
Обмен по протоколу HART.....	416
Обмен по протоколу COMM2H.....	419
Обмен по протоколам SNP, SNP-X	420
Обмен с контроллерами Mitsubishi и Delta	424
Обмен по протоколу NAIS-FP.....	431
Обмен по протоколу RKC	432
Обмен по протоколу UNI-Telway (Master).....	433
Обмен с контроллером RE508.....	438
Обмен по протоколу ModbusPlus	439
Обмен с весовыми терминалами ЗАО 'Тензо-М'.....	441
Связь с сетями LONWORKS	442
Обмен по протоколу S-BUS	446
Обмен с контроллерами Honeywell DPR3000, DPR250	450
Обмен с контроллером STEC2000.....	451
Обмен с приборами фирмы 'Дана-Терм'	453
Обмен с дозатором ДОЗА-4ТМ.....	463
Обмен с преобразователями Tosvert фирмы TOSHIBA	469

Обмен с контроллерами TREI	470
Обмен с контроллерами TCS	471
Обмен с контроллерами ADAM по интерфейсу CAN	472
Обмен с электросчетчиками СЭТ-4ТМ	473
Обмен с контроллерами Vantage	476
Обмен по протоколу ВАСnet/IP	479
Обмен с электросчетчиками Меркурий 230	482
Обмен с TwinCAT (Beckhoff)	485
Обмен по DeviceNet	487
Описание драйверов DeviceNet	487
DeviceNet. Обмен explicit-сообщениями	488
DeviceNet. Обмен I/O пакетами	491
Ограничения драйвера DeviceNet	493
Пример. Настройка обмена с контроллером Mitsubishi FX2N-16MR по протоколу DeviceNet	493
Обмен с приборами фирмы МЗТА	497
Обмен с контроллером КМ800 [КОНТАР]	497
Обмен с приборами серии МИНИТЕРМ	499
Обмен с приборами ДАСУ	501
БОО-М2Г	501
ТВА-1	504
Обмен с контроллерами фирмы ОВЕН	507
Контроллеры ПКП1, СИ8, ТРМ101, ТРМ138	507
Контроллеры УКТ38, ТРМ-32, 33, 34, 38	508
Контроллер МПР-51	510
Обмен с контроллерами фирмы ВИБРАТОР	511
Обмен с контроллерами Ф1765, Ф1766	511
Обмен с устройствами фирмы Danfoss	512
Обмен с частотными преобразователями	512
Обмен с тепловычислителями Эксперт-Z	512
Обмен с теплосчетчиками FP-93	518
Обмен с контроллерами Siemens	519
Протокол Profibus DP	519
Протокол 3964R / RK512	522
Протокол Siemens S7	525
Протокол Siemens PPI	528
Обмен с контроллерами фирмы ЭМИКОН	534
Обмен через плату С05, шина ISA	534
Обмен через плату С06, шина PCI	534
Обмен через плату С06PnP, шина PCI	535
Файлы mode.a и mode.b	535
Обмен с контроллерами Allen Bradley	538
Обмен по протоколу Ethernet/IP Rockwell Automation	538
Обмен по протоколу DF1	540
Обмен с контроллерами OMRON	548
Обмен по протоколу HOST LINK	548
Обмен с контроллерами OMRON по сети Ethernet по протоколу UDP	548
Обмен с контроллерами OMRON по сети Controller Link	551
Обмен с контроллерами фирмы ЭНЕРГОМЕРА	554
Обмен со счетчиком ЦЭ6823	554
Обмен со счетчиком ЦЭ6823М	556

Обмен со счетчиком ЦЭ6822.....	557
Обмен со счетчиком ЦЭ6850М.....	559
Обмен с контроллерами фирмы ВОЛМАГ.....	568
Обмен с контроллерами серии КОНТРАСТ.....	568
Обмен с КР-500 по IP.....	570
Особенности устройств.....	573
Устройства ICP-DAS.....	573
Контроллеры XPAC, WinPac, LinPac и WinCon.....	573
Модули i-70xx / i-87xxx.....	575
Устаревшие механизмы поддержки устройств ICP DAS.....	575
Устройства фирмы ТЕКОН.....	577
Модуль D40Im.....	577
Модуль F24.....	577
Модуль L16i.....	577
Устройства ADVANTECH.....	578
Платы аналогового ввода/вывода контроллера ADAM5510.....	578
Плата ADAM-5080 контроллера ADAM5510.....	580
Плата ADAM-5050.....	581
Модуль ADAM-5013.....	581
Плата PCL-839.....	581
Устройства FASTWEL.....	585
Плата FASTWEL АП16-5А.....	585
Платы FASTWEL UNIOxx-5.....	585
Контроллеры КРУИЗ.....	587
Глава 6.....	589
Отладка проекта.....	589
Профайлеры.....	590
Профайлер с поддержкой графических экранов.....	592
Панель MPB.....	598
Вкладка 'Компоненты'.....	598
Ручное конфигурирование канала.....	599
Вкладка 'Диагностика'.....	600
Вкладка 'Протокол 1'.....	603
Вкладка 'Протокол 2'.....	603
Вкладка 'Пользователи'.....	604
Профайлер без поддержки графических экранов.....	605
Файл tm6_log.txt.....	611
Коды диагностируемых ошибок.....	614
Ошибки выборки данных по запросу экрана/документа.....	615
Отладочные сообщения.....	617
Сообщения при DBG_INFO_NET=ON.....	617
Сообщения при DBG_INFO_NODE=ON.....	619
Сообщения при DBG_INFO_MARKER=ON.....	623
Сообщения при SYNC_MARKER=ON.....	625
Сообщения при DBG_PEVE_TCP=ON.....	626
Сообщения при DBG_INFO_SLAVE=ON.....	628
Сообщения при DBG_INFO_TCP=ON.....	629
Сообщения при DBG_INFO_IEC104=ON.....	630
Сообщения при DBG_INFO_LOGGER=ON.....	632

Глава 1

**Принципиальные
отличия
компонентов
TRACE MODE 6**

Модификации программных компонентов TRACE MODE 6

В данном разделе приведены принципиальные отличия программных компонентов TRACE MODE 6 (см. также **Программный комплекс TRACE MODE 6**).

Спецификации отдельных продуктов размещены на сайте компании.

Модификации ИС TRACE MODE 6

ИС обеспечивает разработку проекта на отдельном компьютере.

ИС выпускается в базовом и профессиональном форматах.

Модификации ИС различаются по общему количеству источников/приемников, которые можно создать с их помощью в проекте.

ИС базового формата имеет следующие ограничения:

- заблокировано создание резервов узлов.

Отладочные мониторы (см. также **Профайлеры**), входящие в состав ИС, имеют ограниченное время непрерывной работы – 2 часа для профессионального формата и 1 час – для базового. Через указанное время блокируется весь ввод/вывод данных (в том числе блокируется выполнение SQL-запросов).

Отладочные мониторы базового формата имеют следующие ограничения:

- заблокирована генерация документов;
- заблокирована посылка в регистратор;
- максимальное число перезагрузок (см. описание переменной **@Load** в разделе **Группа СИСТЕМНЫЕ**) – 3;
- тревожное SMS-сообщение отправляется только на один (первый) телефон группы пользователей (см. **Редактор словарей сообщений**);
- максимальное число SMS-сообщений на телефон пользователя с помощью атрибута 28, **Message** канала класса **Пользователь** (см. **Канал класса ПОЛЬЗОВАТЕЛЬ**) – 3;
- максимальное количество обрабатываемых SMS-сообщений – 10;
- не поддерживается ключ автозапуска узла (**RUN** – см. **Профайлеры** и **Задание параметров работы мониторов**);
- максимальная глубина выборка из архива – 1 сутки.

Модификации мониторов

В пределах семейства мониторы отличаются по максимально возможному числу каналов.

Ограничения мониторов для Windows CE

Эти мониторы аналогичны профайлеру без поддержки графических экранов (см. **Профайлер без поддержки графических экранов**).

Отличия (см. также **Графические панели** и **Описание элементов графических панелей**):

- архивы SIAD не поддерживаются;
- набор времен цикла отображения ограничен;
- не поддерживается обмен с терминалом по последовательному порту;
- не поддерживаются модификации канала CALL, предназначенные для работы с SIAD и графическими экранами;
- не поддерживается подключение внешних библиотек.

Мониторы работают корректно в случае выполнения следующих требований к сборке Windows CE:

- версия – не ниже 5;
- ядро – полнофункциональное;
- поддерживаются графика и сеть.

Регистрация монитора

Отличия от типовой регистрации Микро МРВ, описанной ниже, указаны в пособии по инсталляции продукта.

Для регистрации и активации монитора нужно выполнить следующие действия:

- запустите монитор в контроллере (файл **rtm_ce.exe**), на экране появится менеджер лицензий TRACE MODE 6:



- зайдите на сайт компании AdAstra Research Group, Ltd. в раздел регистрации продуктов (<http://www.adastra.ru/support/register/>), выберите пункт «Регистрация коммерческого продукта TRACE MODE 6» и заполните регистрационную форму, указав, в том числе,

Hardware ID контроллера, отображенный менеджером лицензий:

Регистрация коммерческого продукта TRACE MODE 6

Заполните данную форму для регистрации Вашей программы. Только зарегистрированные пользователи TRACE MODE имеют право на льготное обновление (Upgrade) системы и техническую поддержку.

Выделенные жирным шрифтом вопросы являются обязательными:

Регистрационные данные

Код товара MCTM-WCG-6-512-L1-P-RU-CE

Внешний код (например: 12345R) 123456

Регистрационный номер (например: 14010200398765) 123456789

Hardware ID (если требуется активация) 09f3-5d68-0300-00f6-5d68-0300

В течение 10 минут после регистрации на указанный Вами адрес электронной почты будет автоматически отправлено письмо с подтверждением заявки на регистрацию коммерческого продукта. Далее заявка обрабатывается оператором вручную. Если указанные Вами в форме регистрации данные верны, и все обязательные поля заполнены корректно, то не позднее 2 рабочих дней с момента регистрации Вам будет отправлено письмо с **кодом активации**, который потребуется для «привязки» исполнительного модуля к контроллеру;

- введите полученный код активации в поле **Activation Key** менеджера лицензий:

TRACE MODE 6 License Manager

Hardware ID 09f3-5d68-0300-00f6-5d68-0300

To activate the program, enter the activation key in the textbox below and click OK.

Activation Key **abcd-ef11-2233-4455-ef11**

To get the activation key (if it is not supplied), register your software at <http://www.tracemode.com/support/register/>

Если активация прошла успешно, при последующих запусках монитора менеджер лицензий не отображается.

Особенности LinMPB

Монитор реального времени **LinMPB** для операционной системы Linux предназначен для создания АРМ операторов.

Основной операционной системой для LinMPB является **Linux openSUSE 12.1+** (<http://www.opensuse.org/ru/>).

LinMPB программируется в обычной инструментальной системе SCADA TRACE MODE под Windows релиза 6.09 и старше. LinMPB может быть

частью распределенной АСУ на базе TRACE MODE 6, либо полностью обеспечивать работу простейшей АСУТП, состоящей из одного автоматизированного рабочего места (АРМ).

LinMPB выпускается только для профессиональной версии TRACE MODE.

По своей функциональности LinMPB несколько уступает обычному MPB для Windows,

Особенности

Если в проекте используется изменение системного времени, то для исполняемого файла RTC должны быть заданы права CAP_SYS_TIME.

Если в проекте используется обмен по встроенным протоколам сети и задано применение *ping*, должны быть заданы права CAP_NET_RAW.

Если в проекте используется обмен по COM-порту, то владелец RTC должен иметь права на чтение и запись.

Глава 2

Проект АСУ в TRACE MODE 6

Программный комплекс TRACE MODE 6

TRACE MODE 6 – это программный комплекс, предназначенный для разработки и запуска в реальном времени распределенных автоматизированных систем управления технологическими процессами (АСУТП) и решения ряда задач управления предприятием (АСУП).

Для решения задач АСУП в TRACE MODE 6 интегрирован пакет **T-FACTORY**.

Комплекс программ TRACE MODE 6 можно разделить на 4 части.

- **Интегрированная среда разработки проекта (ИС)** – единая программная оболочка, содержащая все необходимые средства для разработки проекта.

Под **проектом** в TRACE MODE 6 понимается вся совокупность данных и алгоритмов функционирования распределенной АСУ (АСУТП и/или T-FACTORY), заданных средствами TRACE MODE.

Итогом разработки проекта в ИС является создание файлов, содержащих необходимую информацию об алгоритмах работы АСУ.

- **Исполнительные модули (мониторы, МРВ)** – программные модули различного назначения, под управлением которых в реальном времени выполняются составные части проекта, размещаемые на отдельных компьютерах или в контроллерах.

Составная часть проекта (набор файлов), размещаемая на отдельном компьютере или в контроллере и выполняемая под управлением одного или нескольких исполнительных модулей TRACE MODE, называется **узлом проекта**.

В общем случае размещение узла на том же аппаратном средстве, на котором он должен исполняться под управлением монитора, не является обязательным – мониторы могут загружать узлы с удаленных аппаратных средств.

- **Драйверы обмена** – драйверы, используемые мониторами TRACE MODE для взаимодействия с устройствами, протоколы обмена с которыми не встроены в мониторы.
- **Дополнительные средства** – программные модули, предназначенные для решения различных задач, за исключением выполнения узлов (например, TRACE MODE Data Center и сервер печати).

Функциональность модулей TRACE MODE 6 определяется лицензионной политикой, описанной на сайте (<http://www.adastra.ru/products/overview/licence/>), в прейскуранте и в иных коммерческих материалах компании Адастра.

Основные характеристики TRACE MODE 6

Принцип работы монитора. Канал TRACE MODE 6

При старте монитор считывает параметры узла, заданные в ходе разработки проекта в ИС, а также параметры других узлов для корректного взаимодействия с ними.

Алгоритм работы любого монитора TRACE MODE заключается в анализе **каналов** – структур переменных, создаваемых как при разработке проекта в ИС, так и в реальном времени. В зависимости от класса и конфигурации канала, по результатам его анализа монитор выполняет ту или иную операцию – запись значений переменных канала в архив, запрос значения источника данных по указанному интерфейсу и запись этого значения в канал, вызов графического экрана оператора на дисплей и т.п.

Под записью значения в канал в общем случае понимается присвоение значения переменной (атрибуту) **Входное значение** этого канала.

Для канала могут быть сконфигурированы два важнейших свойства – **СВЯЗЬ** и **ВЫЗОВ**.

Первое свойство означает способность канала принимать данные от источников и передавать данные приемникам – другими словами, с помощью этого свойства можно конфигурировать информационные потоки АСУ.

Второе свойство означает способность канала вызывать (реализовать) шаблон с передачей ему необходимых параметров (для канала класса **CALL** свойство **ВЫЗОВ** имеет расширенные функции). На основе свойства **ВЫЗОВ** реализуется, например, графический интерфейс оператора, обмен с базой данных и т.д.

Совокупность каналов узла называется **базой каналов** этого узла.

Класс канала определяет его общее назначение. Например, канал класса **FLOAT** предназначен для операций с 4-байтовыми вещественными числами, канал класса **Единица оборудования** – для учета единицы оборудования, планирования и мониторинга ее техобслуживания. При разработке проекта могут быть созданы каналы только predetermined классов.

Переменные, входящие в канал, называются его **атрибутами**. Атрибуты канала имеют различное назначение и различный тип данных. Булевы атрибуты и атрибуты, которые могут принимать только два определенных значения, называются **флагами**. Примером флага может служить **тип** канала, который принимает два значения – INPUT (числовые каналы типа INPUT предназначены для приема данных от источников) и OUTPUT (числовые каналы типа OUTPUT предназначены для передачи своего значения приемникам). Атрибуты, которые используются для передачи значений при вызове шаблона, называются **аргументами** канала. Атрибуты снабжены числовыми **индексами** (индексация атрибутов начинается с 0, индексация аргументов – с 1000). Атрибуты имеют **полное имя** и **короткое имя** (мнемоническое обозначение). Идентификаторами атрибута являются его индекс и, в ряде случаев, короткое имя.

Каналы содержат внутри себя предопределенные алгоритмы (часть из них может настраиваться пользователем), в соответствии с которыми некоторые атрибуты канала устанавливаются или вычисляются монитором в зависимости от состояния или значения других атрибутов. Например, для большинства каналов в атрибут **Время изменения** монитор записывает время изменения атрибута **Реальное значение** канала (по показаниям часов устройства, на котором запущен монитор).

Исполнение внутренних алгоритмов канала и анализ его атрибутов монитором называется **пересчетом канала**.

По результатам анализа атрибутов монитор выполняет действия, заданные с помощью канала (например, вызов шаблона), – эта процедура называется **отработкой канала**. Отработка канала после его пересчета выполняется при определенных условиях. При пересчете базы каналов пересчет конкретного канала также выполняется при определенных условиях.

Алгоритм пересчета базы каналов описан в разделе **Пересчет базы каналов**.

Каналы одного класса обладают идентичным набором атрибутов и предопределенных алгоритмов их обработки. Существуют также атрибуты, которыми обладают все каналы вне зависимости от их класса (такие атрибуты имеют одинаковые индексы во всех каналах). В состав TRACE MODE входит файл **attribute.txt**, в котором указано соответствие коротких имен атрибутов их индексам в каналах всех классов.

Обеспечение работы распределенных АСУ

Для обеспечения функционирования распределенных АСУ мониторы TRACE MODE поддерживают различные виды взаимодействия с аппаратными средствами и программными приложениями.

- Мониторы поддерживают обмен между собой по протоколу **M-**

LINK (открытый протокол фирмы ADASTRА) при связи компьютеров и/или контроллеров, на которых запущены, по последовательным интерфейсам RS-232/422/485, в том числе при подключении к СОМ-портам дополнительного оборудования, которое конфигурируется при разработке проекта в ИС:

- модемов – при обмене по телефонным линиям;
- радиомодемов – при обмене по радиоканалу;
- GSM-модемов – при обмене по GSM или GPRS;
- конвертеров интерфейсов RS-232/422/485.

Под СОМ-портом здесь понимается как штатный последовательный порт устройства, так и, например, порты, доступные при установке расширителей портов в шины ISA/PCI компьютера.

Мониторы поддерживают обмен одновременно по 32 последовательным портам.

Сеть M-LINK – это сеть типа MASTER-SLAVE с одним ведущим узлом в одной сети в один момент времени.

Узел может иметь статус SLAVE только в сети M-LINK, при обмене по последовательному интерфейсу по любому другому протоколу узел имеет статус MASTER.

- Мониторы поддерживают обмен между собой по протоколу TCP/IP при связи компьютеров и/или контроллеров, на которых запущены, по сети (физическая архитектура сети не имеет значения) – в этом случае на прикладном уровне используется протокол **INET** фирмы ADASTRА.

Мониторы поддерживают обмен одновременно по 4 сетевым адаптерам.

- Мониторы поддерживают обмен с платами ввода/вывода, установленными в системные шины (ISA/PCI/PC-104) компьютеров/контроллеров, на которых запущены.
- Мониторы поддерживают обмен с внешними устройствами (распределенными модулями):
 - по некоторым модификациям протокола MODBUS по последовательным интерфейсам;
 - по протоколу MODBUS TCP/IP по сети;
 - по последовательному интерфейсу по протоколам контроллеров/серий модулей LAGOON, ROBO, NuDAM-6000, I-7000, ADAM-4000, ADAM-5000/485, RIO-2000 и т.п. (в TRACE MODE эти протоколы называются протоколами **DCS**).

- Мониторы поддерживают обмен с произвольными устройствами через драйверы как по стандартным интерфейсам (в том числе полевым шинам), так и при использовании дополнительных устройств, реализующих необходимые интерфейсы, – коммуникационных плат, преобразователей интерфейсов и т.п. Номенклатура драйверов TRACE MODE для обмена с различным оборудованием постоянно расширяется. Кроме того, программный интерфейс взаимодействия монитора с такими драйверами – интерфейс **ТСОМ** – является открытым, что позволяет пользователю разработать драйверы обмена с любым оборудованием.
- Мониторы поддерживают обмен с клиентами/серверами OPC.
- Мониторы поддерживают обмен между собой и с приложениями Windows по DDE/NetDDE.
- Мониторы поддерживают обмен с локальными/удаленными базами данных по ODBC.
- Мониторы поддерживают протокол NMEA (для взаимодействия, например, с GPS-устройствами).

Таким образом, TRACE MODE не накладывает практически никаких ограничений на топологию систем управления и используемые в них аппаратные средства.

Резервирование

Резервирование – это метод экстенсивного повышения надежности АСУ посредством использования дополнительных (резервных) аппаратных средств (например, дополнительного сетевого адаптера, дополнительного СОМ-порта с подключенным к нему дополнительным контроллером и т.п.). Для контроля работоспособности оборудования мониторов используют ряд механизмов, в том числе анализируют каналы, связанные со специальными системными переменными TRACE MODE. При обнаружении отказа основного оборудования мониторы переключаются на резервное. Для узла может быть создано до 2 резервов, предусмотрена синхронизация данных архивов дублированных/троированных узлов и т.п. (см. **Резервирование в TRACE MODE 6**).

Автопостроение

ИС содержит информацию о конструктивном исполнении ряда контроллеров, о платах расширения, которые могут быть вставлены в кейс того или иного контроллера, о внешних модулях, которые могут быть подключены к тому или иному контроллеру и т.д., а также об источниках/приемниках, имеющихся на платах/модулях. На базе этой информации в ИС реализованы различные механизмы **автопостроения** – например, источники/приемники платы, выбранной в списке, создаются автоматически, автоматически создаются каналы, связанные с источника-

ми/приемниками и т.п.

Особой разновидностью автопостроения является автоматическое создание каналов мониторами – например, каналов, связанных с источниками/приемниками (такие каналы создаются мониторами в том случае, если в ИС задана связь источников/приемников с другими компонентами проекта через аргументы этих компонентов).

Математическая обработка данных

Любая АСУ требует математической обработки данных – как в измерительных информационных потоках (датчик => УСО => контроллер => операторская станция), так и в управляющих (операторская станция => контроллер => исполнительное устройство).

Для математической обработки данных в TRACE MODE 6 предусмотрены следующие средства:

- внутренние алгоритмы числовых каналов (см. **Классификация компонентов и Числовые каналы**);
- программы. Для разработки программ в ИС встроены языки **Техно ST**, **Техно SFC**, **Техно FBD**, **Техно LD** и **Техно IL**, являющиеся модификациями языков **ST** (Structured Text), **SFC** (Sequential Function Chart), **FBD** (Function Block Diagram), **LD** (Ladder Diagram) и **IL** (Instruction List) стандарта IEC61131-3. Программы, разрабатываемые в ИС, позволяют использовать функции из внешних библиотек (DLL).

Эти средства обеспечивают возможность математической обработки данных в любом звене информационного потока.

Архивирование каналов узла

Для обеспечения архивирования параметров технологического процесса мониторы TRACE MODE поддерживают функцию записи значений атрибутов каналов в базы данных реального времени – архивы СУБД РВ SIAD/SQL 6 (в дальнейшем – архивы СПАД или архивы SIAD).

Мониторы, работающие в контроллерах, поддерживают индивидуальные архивы (см. **Индивидуальный архив**).

Для каждого узла в ИС могут быть определены 3 пользовательских архива SIAD (локальных или удаленных). Существует также системный архив, используемый мониторами для внутренних целей. При конфигурировании канала указывается, в какой из заданных пользовательских файлов он должен архивироваться.

Архивы SIAD имеют следующие основные характеристики:

- точность значения времени – 1 мс;
- скорость записи в архив для рабочей станции с процессором Pentium-4 с тактовой частотой 2 ГГц – свыше 600 тыс. параметров в секунду.

Архивные данные могут использоваться мониторами, экспортироваться в файлы, а также отображаться графически.

Архивирование каналов проекта

Специализированный монитор **Logger** (Регистратор) может записывать в определенный для него архив SIAD значения атрибутов каналов всех узлов проекта. Данные в этот монитор могут быть переданы по протоколам I-NET и M-LINK (см. **Обеспечение работы распределенных АСУ**).

В проекте может присутствовать до 3 регистраторов, в том числе имеющих резервы.

Обработка аварийных ситуаций

При работе АСУ могут возникать различные аварии – неисправность датчика, обрыв линии связи, выход из строя контроллера/компьютера, выход параметров техпроцесса за установленные границы и т.п. Чтобы обеспечить непрерывность процесса, система управления должна быть снабжена средствами обнаружения и обработки аварийных ситуаций. В TRACE MODE 6 с этой целью предусмотрены следующие средства:

- автоматическая установка признака аппаратной недостоверности каналу с настроенным свойством **связь** или **вызов** (см. **Принцип работы монитора. Канал TRACE MODE 6 и Атрибуты каналов, отображаемые профайлером**), – например, каналу, связанному с аппаратурой ввода/вывода или каналом другого узла по любому из доступных интерфейсов взаимодействия в случае сбоя при обмене данными;
- автоматическая установка признака программной недостоверности каналу при выходе его значения за установленные пределы;
- мониторинг значения каналов FLOAT – для этих каналов задаются 6 границ, с помощью которых возможно своевременное обнаружение нештатного состояния процесса (см. **Канал класса FLOAT**);
- мониторинг событий (в том числе аварий) с помощью канала класса **Событие** (см. **Канал класса СОБЫТИЕ**);
- регистратор аварий – см. **Регистратор аварийных событий**;
- резервирование (см. **Резервирование в TRACE MODE 6**).

TRACE MODE 6 располагает средствами, позволяющими выполнить при работе АСУ действия, направленные на предотвращение возникновения или развития аварии. К таким средствам относятся, например, возмож-

ность включения сигнализации, выдачи рекомендации оператору, блокировки и т.п. Информация о состоянии процесса может быть сохранена в архивы и в отчет тревог (см. **Отчет тревог и генерация сообщений**).

Отчет тревог и генерация сообщений

Мониторы генерируют сообщения в различных ситуациях при работе АСУ – например, при выходе значения канала класса FLOAT за установленную границу, при изменении статуса работника (т.е. при изменении соответствующего атрибута канала класса **Персонал**) и т.п. Эти сообщения могут заноситься в специальный текстовый файл – **отчет тревог (ОТ)**, который конфигурируется для узла (см. **Задание параметров узла и Отчет тревог узла**).

Тексты сообщений могут быть заданы в словарях. Если канал связан со словарем, генерируются сообщения из словаря, в противном случае монитор генерирует сообщения по умолчанию.

В словаре могут быть заданы дополнительные направления передачи сообщений – например, в виде SMS-сообщений на указанный номер сотового телефона (см. **Редактор словарей сообщений**).

В TRACE MODE предусмотрены также средства, которые позволяют оператору записывать в отчет тревог произвольные сообщения, а также просматривать сообщения ОТ и квитировать их (информация о квитировании также заносится в отчет тревог).

Файл восстановления

Для узла может быть определен файл восстановления (дамп). В этот файл монитор записывает последние значения атрибутов каналов узла. Данные заносятся в дампы по каналам, для которых установлен соответствующий флаг.

Дамп используется для восстановления значений каналов узла после рестарта монитора.

Графический интерфейс оператора

TRACE MODE 6 обеспечивает графическое представление хода выполнения техпроцесса, а также управление техпроцессом с помощью графических средств.

Графический интерфейс оператора реализуется в нескольких видах:

- в виде набора **графических экранов**, шаблоны которых разрабатываются в **редакторе представления данных (РПД)**, – для узлов, которые исполняются мониторами на аппаратных средствах, имеющих достаточную производительность и другие необходимые

характеристики (например, при использовании объемной графики от видеосистемы требуется поддержка OpenGL 1.1);

- в виде набора **графических панелей**, шаблоны которых разрабатываются в **еРПД** (модификация **РПД**), – для узлов, которые исполняются мониторами на аппаратных средствах, имеющих ограниченную производительность (например, в контроллерах с ОС Windows CE).

РПД и **еРПД** содержат большое количество встроенных графических элементов (соответственно **ГЭ** и **еГЭ**), позволяющих изобразить практически любой техпроцесс, вывести на дисплей всю необходимую информацию о ходе его выполнения, а также управлять техпроцессом. Кроме того, в состав TRACE MODE 6 входит большое количество ресурсов – текстов, изображений, видеоклипов, различных графических объектов, – которые могут использоваться при разработке графического интерфейса оператора. Ресурсы могут создаваться пользователем.

Генерация документов (отчетов)


Мониторы поддерживают функцию генерации файлов документов (отчетов) формата HTML. Эти документы (отчеты), в том числе, могут содержать информацию о текущих параметрах техпроцесса.

При генерации документов могут использоваться как разработанные шаблоны (см. **Использование разработанных шаблонов**), так и шаблоны, встроенные в МРВ (см. **Использование встроенных шаблонов**).

Защита проекта

Для защиты от несанкционированного доступа к редактированию проекта и/или управлению АСУ для каждого узла должны быть определены пользователи и заданы их права. Для этих целей существуют специальные каналы (см. **Канал класса ПОЛЬЗОВАТЕЛЬ**). Мониторы контролируют права пользователей и записывают результаты контроля в архив и ОТ.

Конвертирование проектов

В TRACE MODE 6 предусмотрен импорт проектов, разработанных в TRACE MODE 5 (по команде  **Импорт из предыдущей версии** – см. **Меню 'Файл' и главная панель инструментов ИС**). При импорте проект конвертируется в формат TRACE MODE 6 и может быть далее модифицирован, сохранен и экспортирован для запуска.

Конвертирование зависит от флага **Отключить конвертирование графической базы** (см. **Вкладка 'Интегрированная среда разработки'**).

Если конвертирование графических баз разрешено, перед импортом в

TRACE MODE 6 все необходимые библиотеки и ресурсы должны быть вручную скопированы в папку проекта. Выполнение данной процедуры не требуется, если проект размещен в директории **%TraceMode5%**.

Конвертирование проектов TRACE MODE 5 в формат TRACE MODE 6 выполняется с некоторыми ограничениями:

- FBD-блок, созданный из программы FBD, конвертируется в отдельную программу.

Следующие компоненты проектов TRACE MODE 5 не конвертируются или конвертируются некорректно:

- метапрограммы;
- каналы вызова драйверов t11 и t12.

Документирование проекта TRACE MODE

Интегрированная среда разработки TRACE MODE 6 снабжена функцией сохранения словесного описания проекта в HTML-файл (см. **Меню 'Файл' и главная панель инструментов ИС**).

Системы встроенного контроля

Система встроенного контроля МРВ включает в себя:

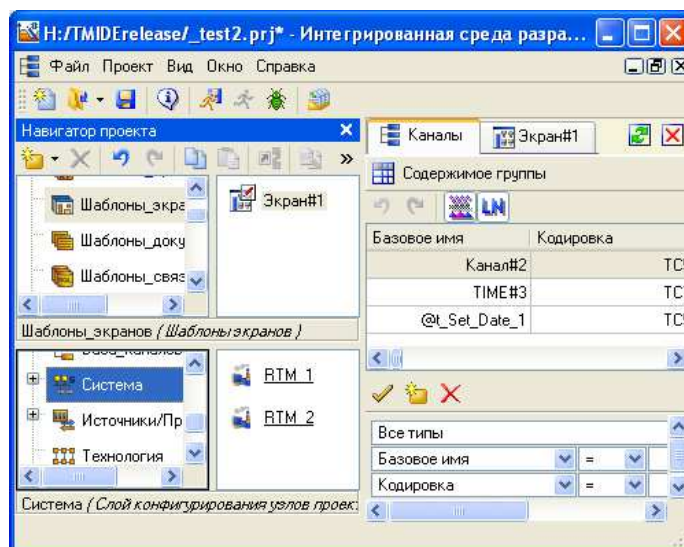
- системные переменные (см. **Группа СИСТЕМНЫЕ** и **Группа ДИАГНОСТИКА**);
- канал **CALL.ChGroupReq** (некоторые конфигурации);
- протокол работы профайлера (файл **<имя файла prj>_<ordinal>.txt** – см. **Профайлеры**);
- протокол системы встроенного контроля монитора (файл **tm6_log.txt** – см. **Файл tm6_log.txt**).

Система встроенного контроля ИС включает в себя:

- протокол конвертера (файлы **<имя файла prj>_<ordinal>.cnv** – см. **Файлы узла, создаваемые при экспорте**).

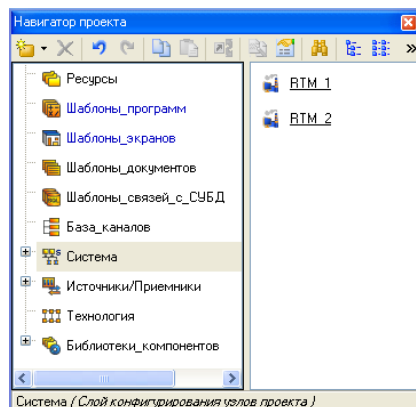
Принципы разработки проекта в ИС

ИС объединяет в единой оболочке **навигатор** и набор редакторов для создания всех составляющих проекта. ИС имеет многооконный интерфейс:



В ИС поддерживаются стандартные операции изменения размеров и перемещения окон (см. **Типовые операции редактирования**).

В навигаторе структура проекта представлена в виде дерева:



Корневые группы этого дерева (**слои**) predetermined and are created automatically when creating a new project (layers are displayed in the left window).

навигатора). Элементарные структурные составляющие (листья структурного дерева) называются **компонентами** проекта. Например, компонентами проекта являются: канал; канал, вызывающий шаблон; шаблон; источник данных и т.д.

Группы компонентов, которые могут быть созданы в структуре проекта, предназначены для структурирования проекта.

Структурирование в значительной степени облегчает редактирование проекта.

В ряде случаев группы имеют и другой, вполне определенный, смысл – например, узлы проекта создаются как корневые группы слоя **Система**.

Слои и группы компонентов могут содержать только определенные группы компонентов и компоненты.

В правом окне навигатора отображается содержимое слоя (группы), выделенной в левом окне, – таким образом, компоненты проекта могут быть отображены только в правом окне.

Все разновидности объектов структурного дерева предопределены и имеют предопределенную **кодировку** TRACE MODE, которая может быть изменена в окне свойств или редакторе объекта (см. **Вкладка ‘Информация’**).

Каждому объекту структурного дерева при его создании присваивается по определенному алгоритму уникальный **идентификатор (ID)**, отображаемый во всплывающей подсказке (см. **Отображение свойств объектов структуры**). По этим идентификаторам объекты структуры анализируются мониторами TRACE MODE.

Структура проекта редактируется в навигаторе с помощью команд меню **Проект**, контекстного меню и панелей инструментов, а также с помощью метода drag-and-drop.

Технология разработки проекта в ИС

Разработка проекта в ИС включает следующие процедуры:

- создание структуры проекта в навигаторе;
- конфигурирование или разработка структурных составляющих – например, разработка шаблонов графических экранов оператора, разработка шаблонов программ, описание источников/приемников и т.д.;
- конфигурирование информационных потоков;
- выбор аппаратных средств АСУ (компьютеров, контроллеров и

т.п.);

- создание узлов в слое **Система** и их конфигурирование;
- распределение каналов, созданных в различных слоях структуры, по узлам и конфигурирование интерфейсов взаимодействия компонентов в информационных потоках;
- сохранение проекта в единый файл для последующего редактирования (см. **Сохранение проекта для редактирования**);
- экспорт узлов в наборы файлов для последующего запуска под управлением мониторов TRACE MODE (см. **Сохранение проекта для запуска**).

Перечисленные процедуры (за исключением двух заключительных) и входящие в их состав операции могут выполняться в произвольном порядке. Например, можно начинать разработку проекта с разработки шаблонов графических экранов оператора, с создания узлов и их каналов в слое **Система** (если аппаратные средства АСУ известны заранее), можно конфигурировать каналы и информационные потоки после распределения каналов по узлам и т.п.

TRACE MODE располагает также средствами для объектного проектирования.

Классификация объектов структуры проекта

Классификация компонентов

По функциональному назначению компоненты проекта относятся к одному из следующих видов:

- **каналы** – компоненты, определяющие алгоритм работы проекта (см. **Алгоритм работы МРВ. Канал TRACE MODE 6**). Каналы могут создаваться в различных слоях, однако их окончательное распределение по узлам в слое **Система** обязательно – в противном случае они не будут экспортированы для МРВ (см. **Технология разработки проекта в ИС**);
- **шаблоны** – компоненты, которые при работе в реальном времени могут вызываться каналами с передачей параметров. Передача параметров настраивается при разработке проекта в ИС посредством привязки **аргументов** шаблона к каналам или источникам/приемникам;
- **источники/приемники** – шаблоны каналов обмена с различными устройствами и приложениями. Под устройствами здесь понимаются контроллеры, а также внешние и внутренние модули/платы различного назначения, обмен с которыми поддерживается мониторами TRACE MODE (в том числе через драйверы). Системные переменные TRACE MODE и встроенные генераторы также создаются в ИС как источники/приемники;
- **наборы ресурсов** – наборы текстов, изображений и видеоклипов, которые могут быть использованы при разработке шаблонов графических экранов;
- **графические объекты** – компоненты, представляющие собой в общем случае несколько графических элементов (из имеющихся в редакторе представления данных), сгруппированных в один. Графические объекты могут быть использованы при разработке шаблонов графических экранов;
- **последовательные порты** – параметры COM-портов;
- **словари сообщений** – наборы сообщений, генерируемых при возникновении различных событий;
- **клеммы** – эти компоненты, описывающие электрические контакты (например, монтажных шкафов), являются элементами схемы электрических соединений АСУ.

Каналы

В TRACE MODE 6 определены каналы нескольких классов.

По функциональному назначению классы каналов можно сгруппировать следующим образом:

- каналы для работы с данными (числовые каналы):
 - **HEX16** – для работы с 2-байтовыми целыми числами;
 - **HEX32** – для работы с 4-байтовыми целыми числами;
 - **FLOAT** – для работы с 4-байтовыми вещественными числами (существуют две разновидности канала этого класса – с обработкой и без обработки в канале);
 - **DOUBLE FLOAT** – для работы с 8-байтовыми вещественными числами;
 - **TIME** – для работы со значениями времени (дата и время);
- каналы для мониторинга:
 - **Событие** – для мониторинга объекта с целью фиксирования возникновения/исчезновения на этом объекте некоторого события или ситуации (например, аварии). Канал хранит историю события и допускает его квитирование;
- каналы для задания прав пользователей:
 - **Пользователь** – для задания прав пользователя на разработку и/или запуск проекта;
- каналы T-FACTORY:
 - **Единица оборудования** – для учета единицы оборудования, планирования и мониторинга ее техобслуживания;
 - **Персонал** – для учета работника, а также планирования и мониторинга его участия в техобслуживании оборудования;
 - **М-ресурс** – для учета складских ресурсов;
 - **D-ресурс** – для мониторинга техобслуживания оборудования и ряда других задач;
- каналы многофункционального назначения:
 - **CALL** – свойство **вызов** канала этого класса конфигурируется для выполнения различных функций. В ИС можно создать следующие каналы этого класса с предустановленным свойством **вызов** (при создании такого канала в соответствующем слое шаблонов создается шаблон, вызываемый каналом):
 - Экран** – канал с вызовом шаблона экрана;
 - Программа** – канал с вызовом шаблона программы;
 - Документ** – канал с вызовом шаблона документа;
 - Связь с БД** – канал с вызовом связи с базой данных.

Для всех классов канала в ИС имеется соответствующий редактор (см. **Редакторы каналов**).

Шаблоны

Шаблон можно рассматривать как функцию, которую вызывает основная программа (монитор) с передачей определенных значений. Шаблоны вызываются каналами класса **CALL** (шаблоны программ могут быть вызваны каналами других классов с настроенным свойством **ВЫЗОВ**) при их обработке монитором (см. **Потоки монитора**).

Значения в шаблон передаются через его аргументы. Эта передача настраивается в ИС с помощью привязки аргументов шаблона к каналам или источникам/приемникам в редакторе аргументов (см. **Табличный редактор аргументов**). Передача аргументов при вызове шаблона обязательна – другими словами, шаблон должен иметь хотя бы один аргумент.

В соответствующих слоях структуры проекта могут быть созданы следующие шаблоны (компоненты проекта):

- **шаблон программы** – разработка программ подробно рассматривается в разделе **Программирование алгоритмов**;
- **шаблон экрана, графической панели, мнемосхемы** – разработка графического интерфейса оператора подробно рассматривается в разделе **Разработка графического интерфейса**;
- **шаблон документа** – разработка документов (отчетов) подробно рассматривается в разделе **Генерация документов**;
- **шаблон связи с базой данных** – компонент проекта, в котором хранятся SQL-запросы к определенной базе данных (см. **Обмен с базами данных**).

Источники/Приемники

Источники/приемники разбиты на predetermined группы в слое **Источники/Приемники** (см. **Назначение групп источников (приемников)**).

Параметры источников/приемников задаются в соответствующих редакторах (за исключением моделей и встроенных генераторов TRACE MODE, которые не имеют редакторов).

Наборы ресурсов и графические объекты

Ресурсы и графические объекты используются при разработке графических экранов оператора (см. **Операции с ресурсными библиотеками** и **Операции с графическими объектами**).

В слое **Ресурсы** могут быть созданы следующие компоненты проекта:

- **Набор текстов;**
- **Набор изображений;**
- **Набор видеоклипов;**
- **Графический объект.**

Наборы ресурсов снабжены соответствующими редакторами (см. **Редактор библиотек текстов**, **Редактор библиотек изображений** и **Редактор библиотек видеоклипов**); графический объект разрабатывается так же, как графический экран, – в редакторе представления данных (РПД).

Последовательные порты

Для описания параметров последовательного порта предназначен компонент **СОМ-порт**. Этот компонент может быть создан в узле в группе **СОМ-порты**.

Параметры последовательных портов задаются в ИС для узлов; при запуске узлов под управлением мониторов эти параметры устанавливаются для портов аппаратных средств, на которых размещены узлы.

Данный компонент имеет соответствующий редактор (см. **Редактор параметров СОМ-порта**).

Словари сообщений

Сообщения, которые будут генерироваться по каналам в различных ситуациях при работе АСУ, могут быть определены в словарях (см. **Редактор словарей сообщений**). Эти сообщения могут быть отправлены в виде SMS-сообщений на указанный номер сотового телефона, переданы по сети консолям и т.д. Словари сообщений задаются в ИС для узла. Направление передачи сообщений задается в редакторах словарей, при этом во всех случаях сообщение заносится в отчет тревог (если отчет тревог для узла не задан, монитор не генерирует сообщений).

В дочерней группе **Словари сообщений** узла могут быть созданы следующие словари:

- **Словарь для HEX16** – сообщения по каналам класса **HEX16**;
- **Словарь для HEX32** – сообщения по каналам класса **HEX32**;
- **Словарь для FLOAT** – сообщения по каналам класса **FLOAT**;
- **Словарь системный** – сообщения по системным событиям;
- **Словарь для персонала** – сообщения по каналам класса **Персонал**;
- **Словарь для оборудования** – сообщения по каналам класса **Единица оборудования**;

- **Словарь для каналов D-ресурс** – сообщения по каналам класса **D-ресурс**;
- **Словарь для каналов Событие** – сообщения по каналам класса **Событие**;
- **Словарь для каналов M-ресурс** – сообщения по каналам класса **M-ресурс**;
- **Пользовательский словарь** – сообщения, генерируемые с помощью системной переменной **@Message** (группа **СИСТЕМНЫЕ**).

В отсутствие словарей в узле, а также в отсутствие связи канала со словарем, монитор генерирует собственные сообщения, которые, как правило, совпадают с сообщениями, заданными в словарях по умолчанию.

Клеммы

Эти компоненты описывают электрические контакты (например, монтажных шкафов) и являются элементами схемы электрических соединений АСУ. В редакторе для клемм задаются параметры подсоединенных проводов с указанием принадлежности проводов к жгутам (см. **Редактор клемм**).

Классификация слоев

Предопределенные слои структуры проекта имеют следующее назначение:

- **Ресурсы** – для создания пользовательских наборов текстов, изображений и видеоклипов, а также графических объектов (см. **Операции с ресурсными библиотеками** и **Операции с графическими объектами** в разделе **Разработка графического интерфейса**);
- **Шаблоны программ** – для создания шаблонов программ;
- **Шаблоны экранов** – для создания шаблонов графических экранов, графических панелей и мнемосхем;
- **Шаблоны связей с БД** – для создания шаблонов связей с базами данных;
- **Шаблоны документов** – для создания шаблонов документов (отчетов);
- **База каналов** – этот слой является хранилищем всех каналов проекта. Выполнять операции с каналами (в том числе создавать их) можно в различных слоях, однако во всех случаях эти операции на самом деле реализуются в слое **База каналов**. В любом другом слое, где выполняется команда для совершения операции с каналом, ее результат только отображается – поэтому существуют команды удаления и уничтожения каналов;

- **Система** – для конфигурирования узлов и их составляющих (узел создается как корневая группа этого слоя);
- **Источники/приемники** – для создания встроенных генераторов, шаблонов каналов обмена с различными устройствами и программными приложениями, а также для конфигурирования системных переменных TRACE MODE 6,
- **Технология** – для разработки проекта от технологии (т.е. с группировкой компонентов по признаку их принадлежности к технологическому объекту). В этом слое кодировка канала строится автоматически с наследованием кодировки всех объектов вышестоящего уровня, в которые канал входит (если установлен флаг **Строить кодировку по технологии** – см. Вкладка ‘Интегрированная среда разработки’). При отладке проекта слой **Технология** может играть роль узла – для него определена команда **Сохранить узел для МРВ**. Кроме того, для этого слоя определены команды взаимодействия с технологической базой данных (см. **Меню и главная панель инструментов навигатора проекта**);
- **Топология** – для разработки проекта от топологии (т.е. с группировкой компонентов по месту расположения);
- **КИПиА** – для описания электрических соединений АСУ;
- **Библиотеки компонентов** – для создания библиотек **объектов** – проектных решений отдельных задач. Этот слой содержит предопределенные группы **Системная** и **Пользовательская**.

Классификация узлов

Узлы проекта создаются как корневые группы слоя **Система**. Предопределенное название узла указывает на семейство мониторов, для которых данный узел предназначен (см. **Модификации мониторов**). Узел может содержать только те компоненты, которые поддерживаются мониторами соответствующего семейства.

В общем случае, узлы могут выполняться под управлением различных мониторов.

Как правило, узел выполняется на отдельном аппаратном средстве. В случае запуска двух и более узлов на одном аппаратном средстве оно должно быть оборудовано соответствующим количеством сетевых карт.

Параметры узлов задаются в соответствующем редакторе (см. **Редактор параметров узла**).

RTM

Узел **RTM** предназначен для запуска на компьютере под управлением исполнительных модулей семейства **RTM** (МРВ) – мониторов с поддержкой отображения графических экранов оператора, поддержкой обмена по по-

следовательному интерфейсу и сети с различным оборудованием и выполняющего пересчет каналов всех классов, кроме каналов T-FACTORY.

T-FACTORY

Узел **T-FACTORY** предназначен для запуска на компьютере под управлением исполнительных модулей семейства **T-FACTORY** – мониторов для решения задач АСУП.

MicroRTM

Узел **MicroRTM** предназначен для запуска на компьютере или в контроллере под управлением исполнительных модулей семейства **Micro RTM**. Основное отличие этих мониторов от МРВ – отсутствие поддержки отображения графических экранов.

Logger

Узел **Logger** предназначен для запуска на компьютере под управлением исполнительного модуля **Logger** (регистратор) – монитора, способного вести архивы по каналам всех узлов проекта.

EmbeddedRTM

Узел **EmbeddedRTM** предназначен для запуска на компьютере или в контроллере под управлением исполнительных модулей семейства **Embedded RTM** – мониторов с поддержкой графических панелей, поддержкой обмена с оборудованием по различным протоколам и выполняющего пересчет каналов.

NanoRTM

Узел **NanoRTM** предназначен для запуска в контроллере под управлением исполнительного модуля **Nano RTM** – монитора, аналогичного **Micro RTM**, но предназначенного для работы с малым числом каналов.

Console

Узел **Console** предназначен для запуска на компьютере под управлением исполнительных модулей, которые, в отличие от МРВ, не выполняют пересчет каналов, предназначенных для работы с данными. Консоли позволяют получать данные от других узлов проекта по сети, отображать их на графических экранах и управлять технологическим процессом из графики. Консоли не могут взаимодействовать с узлами T-FACTORY.

TFactory_Console

Узел **TFactory_Console** предназначен для запуска на компьютере под управлением исполнительных модулей, аналогичных консолям, но, кроме того, способных взаимодействовать с узлами T-FACTORY.

EmbeddedConsole

Этот узел выполняется под управлением мониторов, поддерживающих только графические панели.

Назначение групп источников (приемников)

В слое **Источники/Приемники** могут быть созданы следующие предопределенные корневые группы:

- **PC-based контроллеры;**
- **Распределенные УСО (DCS);**
- **Платы ввода/вывода;**
- **Терминалы;**
- **PLC;**
- **OPC-группа;**
- **DDE-группа;**
- **MODBUS-группа;**
- **Пользовательские драйверы;**
- **Диагностика и сервис;**
- **Генераторы;**
- **Модели.**

Первые 9 групп предназначены для создания шаблонов каналов обмена с различными устройствами и программными приложениями, обмен с которыми поддерживается мониторами TRACE MODE 6. Источники/приемники устройств называются **аппаратными каналами** или **тегами**

При создании описаний конкретных плат/модулей шаблоны каналов обмена с ними создаются в навигаторе автоматически как компоненты проекта (см. **Шаблоны каналов обмена**). Набор автоматически созданных источников/приемников может быть отредактирован вручную (например, с помощью команд контекстного меню).

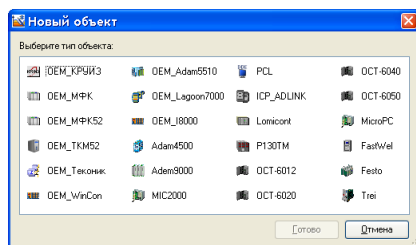
При автоматическом создании для источников/приемников по умолчанию задается ряд параметров, которые могут быть отредактированы в соответствующих редакторах. Группы, непосредственно включающие источники/приемники, также имеют редакторы, с помощью которых возможно групповое редактирование параметров источников/приемников.

В навигаторе проекта поддерживается автопостроение каналов, связанных с источниками/приемниками, при копировании/вставке этих источников/приемников в группу каналов. Каналы, созданные таким способом, в ряде случаев требуют дополнительной конфигурации.

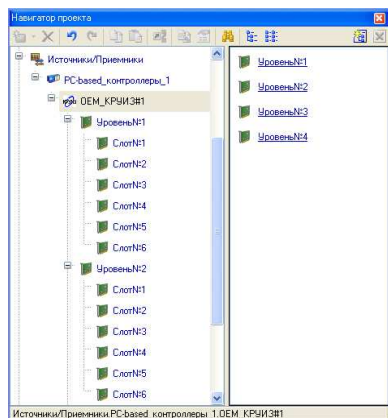
Если источники/приемники связаны с другими компонентами проекта через аргументы, то каналы, настроенные на эти источники/приемники, создаются мониторами в реальном времени.

Группа 'PC-based контроллеры'

В этой группе описываются контроллеры, предназначенные для функционирования в АСУ ТП под управлением исполнительных модулей TRACE MODE. На рисунке ниже показан диалог выбора контроллера, который открывается по команде **Создать группу** из контекстного меню группы **PC-based контроллеры**:

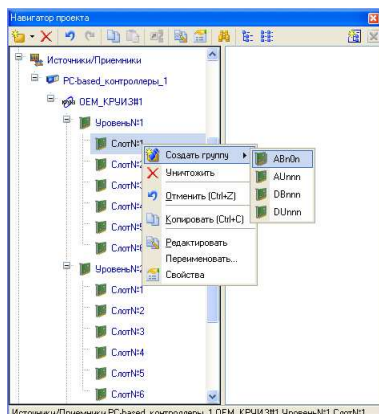


В ИС имеется информация о конструктивном исполнении контроллеров, поэтому при создании контроллера как группы в навигаторе проекта его топология (этажи, слоты и т.п.) создается автоматически в виде соответствующих дочерних групп. На рисунке ниже показан результат создания контроллера **ОЕМ Круиз** в группе **PC-based контроллеры** (все дочерние группы группы **ОЕМ КРУИЗ** созданы автоматически):

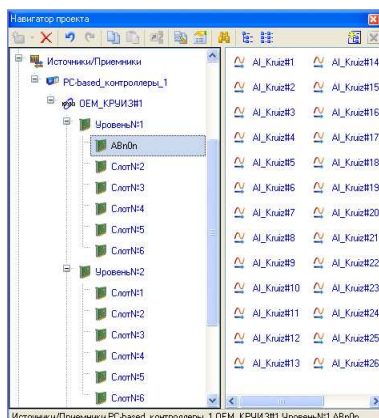


В ИС имеется информация о платах расширения, которые могут быть вставлены в кейс того или иного контроллера, а также о внешних моду-

лях, которые могут быть подключены к тому или иному контроллеру. На рисунке ниже представлен процесс «вставки» платы в слот 1 уровня 1 контроллера КРУИЗ:



При выборе платы (например, **ABn0n**) ее источники/приемники создаются автоматически как компоненты проекта:



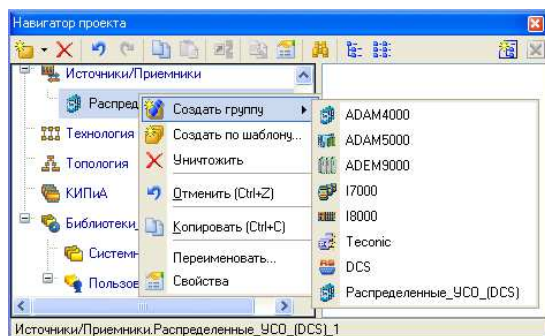
Для PC-based контроллеров необходимо создать соответствующие узлы в слое **Система**.

Группа 'Распределенные УСО'

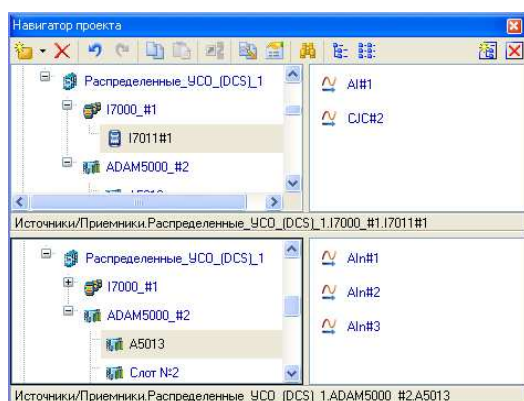
В этой группе описываются внешние устройства, для обмена с которыми мониторы TRACE MODE используют протоколы контроллеров/серий устройств LAGOON, ROBO, NuDAM-6000, I-7000, ADAM-4000, ADAM-5000/485, RIO-2000 и т.п. Эти протоколы называются в TRACE MODE протоколами **DCS**. Кроме того, в этой группе описываются устройства с известным набором переменных, поддерживающие обмен по некоторым модификациям протокола **MODBUS** (см. **Группа 'MODBUS-группа'**). Подобные устройства (контроллеры или модули ввода/вывода) называются

ся в TRACE MODE **распределенными модулями**. Поддержка указанных протоколов встроена в мониторы TRACE MODE.

При создании группы в группе **Распределенные УСО (DCS)** на экране появляется диалог выбора контроллера/серии модулей:



Модуль или плата расширения создается как дочерняя группа (соответственно в группе серии модулей или в группе контроллера). При выборе устройства его источники/приемники создаются автоматически как компоненты проекта:



Группа **DCS Group** в группе **Распределенные УСО (DCS)** предназначена для описания переменных произвольного DCS-устройства. Переменные такого устройства описываются вручную путем выбора в диалоге (см. **Обмен по DCS** и **Подтип 8**).

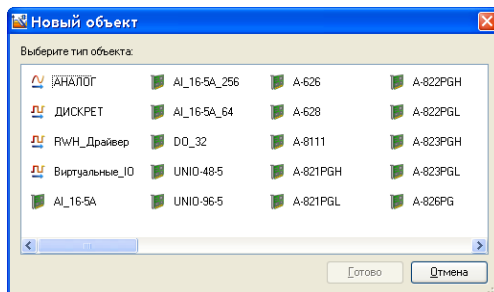
Группа 'Платы ввода-вывода'

В MPB, работающих в среде Windows 7, группа **Платы ввода-вывода** не поддерживается.

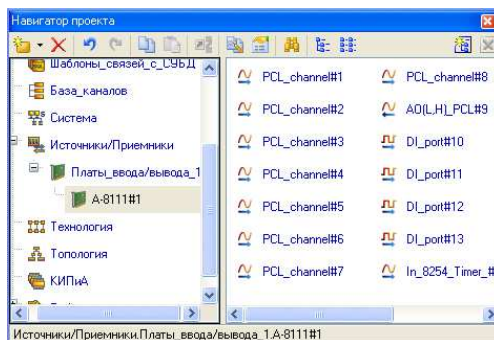
В этой группе описываются источники/приемники плат ввода/вывода, устанавливаемых непосредственно на системные шины компьютеров и

контроллеров, работающих под управлением мониторов TRACE MODE.

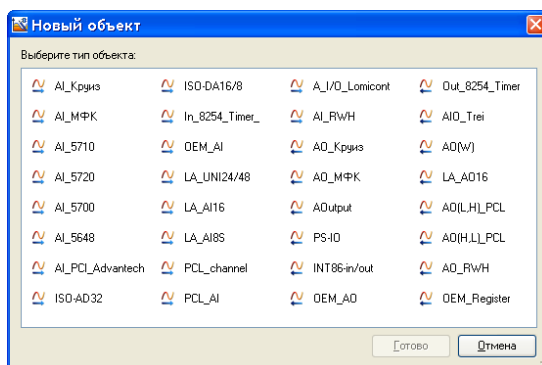
Описание платы создается как группа в группе **Платы ввода/вывода** (при создании группы плата выбирается в диалоге):

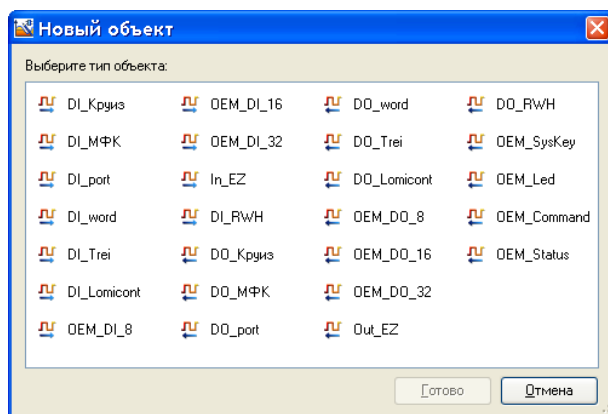


При выборе платы ее источники/приемники создаются автоматически как компоненты проекта:



Кроме плат с известными наборами аппаратных тегов, в группе **Платы ввода/вывода** можно создать описание произвольных аналоговых или дискретных плат ввода/вывода – соответственно как группы **АНАЛОГ** и **ДИСКРЕТ**. Для таких плат источники/приемники создаются вручную путем выбора из диалогов (см. **Подтип 1** и **Подтип 2**):





С помощью некоторых источников/приемников данных групп конфигурируется обмен с УСО через драйвер (см. **Разработка драйверов. Интерфейс ТСОМ**).

С помощью источников/приемников группы **RWH Драйвер** конфигурируется обмен с контроллерами через драйвер (см. **Разработка драйверов. Интерфейс ТСОМ**).

Группа **Виртуальные IO** зарезервирована.

Группа 'Терминалы'

В подгруппе **Terminal** данной группы создаются шаблоны каналов обмена с терминалами и клавиатурами (см. **Подтип 66**).

Группа 'PLC'

В данной группе описываются контроллеры, с которыми мониторы TRACE MODE могут обмениваться данными через соответствующие драйверы. Вначале в группе **PLC** создается группа контроллеров (выбирается в диалоге).

В созданной группе создаются шаблоны каналов обмена с контроллером (см. **Обмен через драйверы**).

Группа 'ОПС-группа'

В дочерних группах **ОПС Сервер** и **ОПС HDA Сервер** данной группы описываются переменные серверов **ОПС** (могут быть созданы вручную и автоматически – см. **Редактор группы 'ОПС-сервер'** и **Редактор группы 'ОПС HDA сервер'**).

Параметры переменных ОПС задаются в соответствующих редакторах (см. **Редактор переменной ОПС**, **Редактор переменной ОПС HDA** и

Обмен по OPC).

Группа 'DDE-группа'

В этой группе описываются источники/приемники **DDE/NetDDE**.

Параметры переменных DDE задаются в редакторе (см. **Редактор переменных DDE**).

Группа 'MODBUS-группа'

В этой группе описываются произвольные устройства, поддерживающие обмен по некоторым модификациям протокола **MODBUS**.

В группе могут быть созданы шаблоны каналов обмена (см. **Подтипы 9 и 100**), которые конфигурируются в соответствующем редакторе (см. **Обмен по MODBUS**).

Группа 'Пользовательские драйверы'

В дочерней группе **TYPE11** данной группы конфигурируется обмен с оборудованием через пользовательские драйверы типа **t11**. В группе могут быть описаны следующие источники/приемники:

- универсальные ПЛК типа 1;
- универсальные ПЛК типа 2;
- универсальные ПЛК типа 3;
- универсальные ПЛК типа 4.

В дочерней группе **TYPE12** данной группы конфигурируется обмен с оборудованием через пользовательские драйверы типа **t12**. В группе могут быть описаны следующие источники/приемники:

- универсальные ПЛК типа 1;
- универсальные ПЛК типа 2;
- универсальные ПЛК типа 3;
- универсальные ПЛК типа 4.

Параметры таких источников/приемников задаются в соответствующих редакторах (см. **Удаленный адрес и разновидности драйверов**).

Группа 'Диагностика и сервис'

В дочерних группах **Группа Системные** и **Группа Диагностика** группы **Диагностика и сервис** описываются системные переменные TRACE MODE (см. **Системные переменные TRACE MODE 6**). Каналы, связанные с этими переменными, могут в реальном времени управлять различными системными параметрами или отображать их состояние.

Например, канал, связанный с переменной **@Net_DDE** (группа СИСТЕМНЫЕ), управляет обменом по сети и DDE; канал, связанный с переменной **@e_DDE** (группа ДИАГНОСТИКА), отображает состояние обмена по DDE и т.п.

Все системные переменные создаются монитором вне зависимости от их наличия в проекте.

Группа ‘Генераторы’

В этой группе как компоненты проекта описываются встроенные генераторы TRACE MODE 6.

Каналы, связанные с генераторами, на каждом такте пересчета изменяют свое входное значение по соответствующему закону:

- **Пила** – по пилообразному закону (от 0 до 100);
- **Треугольник** – входное значение канала последовательно возрастает от 0 до 100/убывает от 100 до 0;
- **Синусоида** – входное значение канала изменяется по закону $50 \sin(0.02\pi N) + 50$, где **N** – номер такта пересчета;
- **Бегущая единица** – входное значение канала последовательно принимает значения $0, 2^0, 2^1, \dots, 2^{16}, 0, 2^0$ и т.д.;
- **Битовый меандр** – входное значение канала последовательно принимает значения 0 и 1;
- **Случайное число** – входное значение канала представляет собой случайную величину с равномерным законом распределения в диапазоне [0, 100];
- **Источник** – пустой источник, закон изменения значения для него может быть задан в программе (собственный отсутствует).

Группа ‘Модели’

В этой группе как компоненты проекта описываются встроенные модели TRACE MODE 6: При перетаскивании моделей в узел (см. **Копирование и вставка объекта структуры**) в нем создаются каналы FLOAT со специфическими внутренними алгоритмами (см. также **Подтип 64**).

Модель объекта первого порядка

Атрибут 0, **R** индицирует отклик объекта на сигнал, поданный в атрибут 2, **In**:

$$R = \frac{HW * HL + A * LL}{HL + 1} + LW$$

где

30, **HW** – реальное значение канала на предыдущем такте пересчета;

26, **HL** – постоянная времени;

27, **LL** – коэффициент усиления.

В алгоритме участвуют также атрибуты (28, **HA**), (29, **LA**) и (32, **HYST**). Если **HYST**=0 при **LA**<>0, то **HYST**=0.5, **R**= $R+\sin(HA*LA)$ и **HA**=**HA**+**HYST**.

Модели клапана и задвижки

В алгоритме, аналогичном алгоритмам FBD-блоков **ZDV** и **KLP**, участвуют следующие атрибуты:

- 26, **HL** – время хода клапана/задвижки. Задание **HL**≤100 в ИС означает задание времени хода в секундах, задание **HL**>100 в ИС означает задание времени хода в миллисекундах. В МРВ время хода всегда в миллисекундах;
- 2, **In** – команда управления.

В режиме потенциального управления (бит 8 (0x100) атрибута 30, **HW** равен 0):

- 0 – остановить;
- 1 – открыть;
- 2 или -1 – закрыть;

В режиме импульсного управления (бит 8 (0x100) атрибута 30, **HW** равен 1; только для модели задвижки; обработка команд производится по переднему фронту импульса):

- импульс в бите 0 **In** – открыть,
- импульс в бите 1 **In** – закрыть,
- импульс в бите 2 **In** – остановить;
- биты реального значения канала (0, **R**) имитируют сигналы конечных выключателей:
 - бит 0 – сигнал концевого выключателя открытия (1 – открыто полностью, 0 – закрыто или открыто не полностью);
 - бит 1 – сигнал концевого выключателя закрытия (1 – закрыто полностью, 0 – открыто или закрыто не полностью);
 - бит 2 – сигнал концевого выключателя муфты (1 – закрыто полностью, 0 – открыто или закрыто не полностью);
- 31, **LW** – в этот атрибут записывается текущее положение клапана/задвижки (вычисляется по **HL** и времени, прошедшему с момента подачи команды). Текущее положение клапана/задвижки индицируется как процент закрытия (0 – полностью открыто, 100 – пол-

ностью закрыто);

- 27, **LL** – процент закрытия (при нулевом значении нулевого бита атрибута **HW** в **LL** записывается значение **LW**). Если установлен бит 20 (0x100000) **HW**, то **LL** индицирует процент открытия (**LL** = 100 – **LW**);
- 30, **HW** – установленные в 1 биты этого атрибута имитируют ошибки или задают режим работы:

Режим вычисляется как результат логического сложения (OR) **HW** и значения ключа **MDLSTSM**=<число> в файле *.CNF.

- бит 0 (0x1) – запрет копирования **LW** в **LL**;
- бит 1 (0x2) – при закрытии/открытии из крайнего положения не отключился концевой выключатель открытия/закрытия по истечении времени **HA** (**0.1*HL**, если **HA** не задано);
- бит 2 (0x4) – при **LW**=0 нет сигнала от концевого выключателя открытия;
- бит 3 (0x8) – при **LW**=100 нет сигналов от концевых выключателей закрытия и муфты (**R**<6);
- биты 4 и 5 (0x10 и 0x20):
0x10 – **R**=3;
0x20 – **R**=0;
0x30 – инвертировать биты 0-2 **R**;
- биты 6 и 7 (0x40 и 0x80):
0x40 – инверсия сигнала электрической части (привода);
0x80 – отключить модель (игнорирование команд);
- бит 8 (0x100) – включить режим импульсного управления (только для модели задвижки);
- бит 9 (0x200) – бит 1 **C5** равен 1 (напряжение есть), в противном случае 0 (авария);
- бит 10 (0x400) – задвижка/клапан не движется (ошибка);
- бит 11 (0x800) – время хода при открытии равно 26, **HL**, при закрытии – примерно в 9 раз меньше;
- бит 14 (0x4000) – инвертировать сигналы концевых выключателей открытия/закрытия и муфты (только для модели задвижки);
- бит 17 (0x20000) – генерировать сигнал электрической части в 98, **C5** с запаздыванием;
- бит 18 (0x40000) – считать задвижку открытой (в отсутствие потенциальной команды открытия/закрытия);
- бит 19 (0x80000) – считать задвижку закрытой (в отсутствие

- потенциальной команды открытия/закрытия);
- бит 20 (0x100000) – индцировать процент открытия в **LL**;
- бит 21 (0x200000) – если 98, **C5** = 1, то установить бит 8 (0x100) **R**;
- 28, **HA** – процент от **HL**; задает время, по истечении которого отключать/включать концевые выключатели;
- 93, **C0** – бит 0 – аналог бита 0 **R**;
- 94, **C1** – бит 0 – аналог бита 1 **R**, бит 1 – аналог бита 2 **R**;
- 95, **C2** – команда открытия в режиме потенциального управления;
- 96, **C3** – команда закрытия в режиме потенциального управления;
- 98, **C5** – сигнал электрической части (привода), устанавливается автоматически при подаче команды открытия/закрытия. Если задвижка (клапан) находится в одном из крайних положений или остановлена, то бит 0 **C5** равен 0; в противном случае бит 0 **C5** равен 1;
- 32, **HYST** – в этот атрибут записывается текущее положение клапана/задвижки (вычисляется по нелинейному закону по **HL** и времени, прошедшему с момента подачи команды). Положение клапана/задвижки индцируется как доля закрытия (0 – полностью открыто, 1 – полностью закрыто) или открытия (0 – полностью закрыто, 1 – полностью открыто) в зависимости от бита 20 **HW**.

Модель мотора

В алгоритме, аналогичном алгоритму FBD-блока **MOTOR**, участвуют следующие атрибуты:

- 96, **C3** и 97, **C4**:
 - **C3**=1 – включить на скорости 1; 2 или 4 – включить на скорости 2;
 - **C4**=1 – выключить;
 - **C3**=**C4**=0 – управление с помощью 2, **In**;
- 26, **HL** – задержка перехода из ON (OFF) в OFF (ON);
- 2, **In** – команда:
 - 2 – выключить;
 - 1 – включить на скорости 1;
 - 4 – включить на скорости 2;
- 31, **LW** – текущее значение задержки включения/выключения (в процентах от **HL**);
- установленные в 1 биты реального значения канала (0, **R**) имитируют наличие сигналов датчиков:
 - бит 0 – сигнал датчика включения на скорости 1;

- бит 1 – сигнал датчика выключения;
- бит 2 – сигнал датчика нагрузки;
- бит 3 – сигнал датчика включения на скорости 2;
- 30, **HW** – установленные в 1 биты этого атрибута интерпретируются следующим образом:
 - бит 0 (0x1) – запрет установки в 1 бита 0 **R**;
 - бит 1 (0x2) – запрет установки в 1 бита 1 **R**;
 - бит 2 (0x4) – запрет установки в 1 бита 2 **R**;
 - бит 3 (0x8) – установить бит 2 **R**;
 - бит 4 (0x10) – **R**=3;
 - бит 5 (0x20) – **R**=0;
 - бит 6 (0x40) – **R**=1.
 - бит 7 (0x80) – отключить модель;
 - бит 8 (0x100) – установить бит 0 **R**;
 - бит 9 (0x200) – установить бит 1 **R**;
 - бит 10 (0x400) – запрет изменения состояния;
 - бит 11 (0x800) – установить бит 3 **R**;
 - бит 12 (0x1000) – сбросить бит 0 **R**;
 - бит 13 (0x2000) – сбросить бит 1 **R**;
 - бит 14 (0x4000) – сбросить бит 2 **R**;
 - бит 15 (0x8000) – сбросить бит 3 **R**;
- 98, **C5** – в бит 0 **C5** копируется бит 2 **R**;
- 32, **HYST** – кривая оборотов (давления) (нарастание, рабочий режим (константа), уменьшение до 0).

Модель резервуара

Алгоритм:

$$R_i = R_{i-1} + A_i + (LL_i - HL_i)\Delta t$$

где

i – текущий такт пересчета;

27, **LL** – скорость поступления ресурса в резервуар (имитация непрерывного процесса);

26, **HL** – скорость расхода ресурса из резервуара (имитация непрерывного процесса);

Δt – период пересчета канала, с.

В атрибут **In** подается объем разового поступления ресурса в резервуар.

Модель печи

Модель печи – это алгоритм вычисления температуры вещества; в алгоритме участвуют следующие атрибуты:

- 0, **R** – температура вещества;
- 30, **HW** – масса вещества;
- 2, **In** – подводимое тепло;
- 31, **LW** – прибыль (**LW**>0) или убыль (**LW**<0) вещества (если убыль задана большей, чем масса вещества, то **LW**=–**HW**);
- 26, **HL** – температура добавляемого вещества;
- 27, **LL** – отводимое тепло (характеристика остывания вещества);

Удельная теплоемкость вещества принимается равной 1.

Если 30, **HW**=0, то **R**=0.

Назначение других групп

Группа ‘СОМ-порты’

Эта группа может быть создана в узле или в слое **База каналов** и предназначена для создания в ней компонентов **СОМ-порт**.

Группа **СОМ-порты** имеет редактор (см. **Базовый редактор группы компонентов**).

Группа ‘Словари сообщений’

Эта группа может быть создана в узле или в слое **База каналов** и предназначена для создания в ней словарей сообщений (компонентов проекта).

Группы слоя ‘Библиотеки компонентов’

Пользовательская библиотека проектных решений создается как группа **Библиотека** в подгруппе **Пользовательская** слоя **Библиотеки компонентов**. Сами проектные решения создаются как группы **Объект** в группе **Библиотека**. Группы **Объект** содержат автоматически создаваемые подгруппы **Ресурсы**, **Шаблоны программ**, **Шаблоны экранов**, **Шаблоны документов**, **Шаблоны связей с БД**, **База каналов** и **Источник/Приемники**. Для объекта библиотеки эти группы выполняют те же функции, что и слои – для проекта (см. **Классификация слоев**).

Группы клемм

Компоненты **Клемма** могут быть созданы в группах **Рейка** и **Клеммный блок** в слое **КИПиА**. Эти группы имеют редактор (см. **Базовый редактор группы компонентов**).

Прочие группы

Группы, описываемые в данном разделе, предназначены для структурирования проекта. Некоторые группы имеют редакторы, с помощью которых возможно групповое редактирование параметров компонентов, входящих в эти группы.

Группы ресурсов

К этим группам относятся группы **Тексты**, **Картинки**, **Анимация** и **Графические объекты**, создаваемые в слое **Ресурсы**. Эти группы предназначены для создания в них соответствующих компонентов – наборов текстов, наборов изображений, наборов видеоклипов и графических объектов.

Группы шаблонов

Группы **Шаблоны программ**, **Шаблоны экранов**, **Шаблоны документов** и **Шаблоны связей с БД** могут быть созданы в соответствующих шаблонных слоях. Эти группы предназначены для создания в них соответствующих шаблонов.

Группа шаблонов экранов имеет редактор (см. **Редактор группы шаблонов экранов**).

Группы каналов

Группы каналов имеют по умолчанию наименования, определяющие их основное назначение:

- **Пользователи** – эти группы могут быть созданы в узле или слое **База каналов** и предназначены для создания в них каналов класса **Пользователь**. Группа **Пользователи**, в свою очередь, может содержать компоненты **Пользователь** и группы **Пользователи**;
- **Каналы** – эти группы могут быть созданы в слое **База каналов** и в узлах. Группа каналов, в свою очередь, может содержать каналы и группу каналов;
- **(HRM) Персонал** – эти группы могут быть созданы в узле **T-FACTORY** и предназначены для создания в них каналов и групп каналов (**HR-группа**) класса **Персонал**;

- **(EAM) Основные фонды** – эти группы могут быть созданы в узле **T-FACTORY** и предназначены для создания в них каналов и групп каналов (**EAM-группа**) класса **Единица оборудования**;
- **(MES) Управление производством** – эти группы могут быть созданы в узле **T-FACTORY** и предназначены для создания в них каналов и групп каналов (**Группа MES**) классов **M-ресурс** и **D-ресурс**. В группе **(MES) Управление производством** может быть также создана группа **Компоненты T-FACTORY**, предназначенная для создания в ней всех каналов T-FACTORY;
- **Объект, Функция, Агрегат, Элемент** – в этих иерархических группах слоя **Технология** могут быть созданы группы, нижерасположенные по иерархии, и каналы всех классов;
- **Объект, Здание, Этаж, Помещение** – в этих иерархических группах слоя **Топология** могут быть созданы группы, нижерасположенные по иерархии, и каналы всех классов.

Возможность создания в узле тех или иных групп каналов, а также перечень каналов, доступных для создания в группе, зависит от типа узла.

В общем случае любая группа каналов может содержать каналы любых классов.

Группа каналов имеет редактор (см. **Базовый редактор группы компонентов**).

Группы слоев **Топология** и **Технология** имеют комбинированный редактор (см. **Редактор группы каналов слоев 'Технология' и 'Топология'**).

Дополнительные группы структурирования

К этим группам относятся иерархические группы, создаваемые в слое **КИПиА**, – **Помещение**, **Шкаф** и **Уровень**.

Данные группы могут содержать следующие по иерархии группы, а также группы клемм **Рейка** и **Клеммный блок**.

Операции в ИС

Меню и панель инструментов ИС

Оболочка ИС имеет главное меню, включающее меню **Файл**, **Вид**, **Окна** и **Справка**, и панель инструментов.

Редакторы, встроенные в ИС, имеют свои меню и панели инструментов, которые при открытии этих редакторов частично или полностью добавляются к имеющимся в ИС. При открытии редактора возможно также модифицирование списка команд меню ИС.

В случае открытия нескольких редакторов, панели инструментов и меню ИС соответствуют редактору, окно которого в текущий момент является активным.

Меню и панель инструментов оболочки ИС доступны во всех случаях.


Инструменты всех редакторов и окон ИС снабжены всплывающими подсказками.


Инструменты и операции редактирования, которые являются типовыми для различных редакторов, описаны в разделе **Типовые средства редактирования**. Специфические инструменты отдельных редакторов рассматриваются в разделах, посвященных работе в этих редакторах.

Меню 'Файл' и главная панель инструментов ИС




Меню **Файл** и главная панель инструментов ИС включают следующие команды:

 **Новый (Ctrl-N)** – создать новый проект;

 – открыть проект (файл с расширением **prj** – см. **Сохранение проекта для редактирования**):

 **Открыть (Ctrl-O)** – выбрать файл в стандартном диалоге операционной системы;


 – выбрать файл из списка последних открытых;

В ИС можно загрузить один проект.


При создании/открытии проекта, ИС загружает пользова-


тельные библиотеки компонентов (файл **tmdevenv.tmul** из директории ИС).

Импорт – по этой команде открывается меню, содержащее следующие команды:


 **Импорт из предыдущей версии** – открыть проект, разработанный в TRACE MODE 5. По этой команде открывается окно выбора файла *.ctm, и выбранный проект конвертируется в TRACE MODE 6 (см. **Конвертирование проектов**);

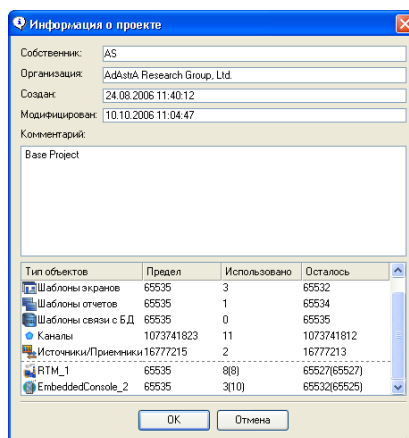
Заккрыть – закрыть проект;

 **Сохранить (Ctrl-S)** – сохранить проект в файл **prj** с тем же именем (пользовательские библиотеки компонентов сохраняются по этой команде в файл **tmdevenv.tmul**);

 **Сохранить как (Ctrl-Shift-S)** – сохранить проект в файл **prj** с заданием его имени (пользовательские библиотеки компонентов сохраняются по этой команде в файл **tmdevenv.tmul**);


Объединить библиотеки – объединить пользовательские библиотеки компонентов (см. **Объединение пользовательских библиотек компонентов**);


 **Информация о проекте** – открыть одноименный диалог (эта команда доступна также из меню **Проект** и контекстного меню навигатора проекта):





В этом диалоге можно указать автора проекта, организацию и комментарий к проекту. В диалоге индицируется время создания и время последнего изменения проекта. В нижней части диалога выводится информация об основных видах компонентов проекта (для узлов индицируется максимально

возможное число каналов, число каналов, созданное в ИС, и (в скобках) число каналов, которое будет создано монитором, а также число каналов, оставшееся в распоряжении разработчика);

 **Сохранить для MPB** – экспортировать узлы для последующего запуска под управлением исполнительных модулей TRACE MODE (см. **Сохранение проекта для запуска**). Экспорт одного узла возможен из меню **Проект** и контекстного меню навигатора проекта (см. **Меню и главная панель инструментов навигатора проекта**);

 **Отладка** – загрузить выделенный узел (слой **Технология**) в профайлер, заданный в настройках ИС (см. **Вкладка ‘Интегрированная среда разработки’**). Эта команда доступна только после экспорта узла (слоя **Технология**). Кроме того, для узла эта команда доступна только в том случае, если узел экспортирован в папку, которая создается по команде **Сохранить для MPB** (см. **Сохранение проекта для запуска**);

 **Шпион** – получить в ИС реальные данные с работающих узлов (см. **Отладка проекта**);

 **Документировать проект (CTRL+P)** – сохранить словесное описание проекта в файл **<имя проекта>.html** в указанной директории (выбирается в стандартном диалоге ОС, который открывается при выполнении команды). Кроме указанного файла, в директории создается поддиректория рисунков **images** и стиловой файл **tmde_print_style.css**;

Настройки ИС – открыть диалог задания общих настроек ИС и редакторов шаблонов (см. **Задание общих настроек ИС**);

Последние файлы – показать список последних файлов, открытых в ИС. Выбранный в списке файл загружается в ИС;


Выход (ALT+F4) – выйти из интегрированной среды.

Меню ‘Вид’

Команды этого меню управляют видимостью панели инструментов ИС, навигатора проекта и строки статуса, отображаемой в нижней части окна ИС. В строку статуса выводится информация о командах меню ИС.

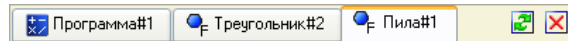
Меню ‘Окна’

Данное меню включают команду **Заккрыть все** – по этой команде закрываются все окна ИС, кроме навигатора проекта.

В данное меню, кроме того, выводится список открытых окон ИС с указанием окна, активного в текущий момент. Это окно помечается в списке значком .


Панель переходов между окнами ИС


Для переходов между открытыми окнами редакторов используется панель, показанная на рисунке ниже. Оранжевая полоска обозначает активное окно.



Для перехода в нужное окно нужно нажать ЛК на соответствующей вкладке этой панели, при этом выбранное окно становится активным.

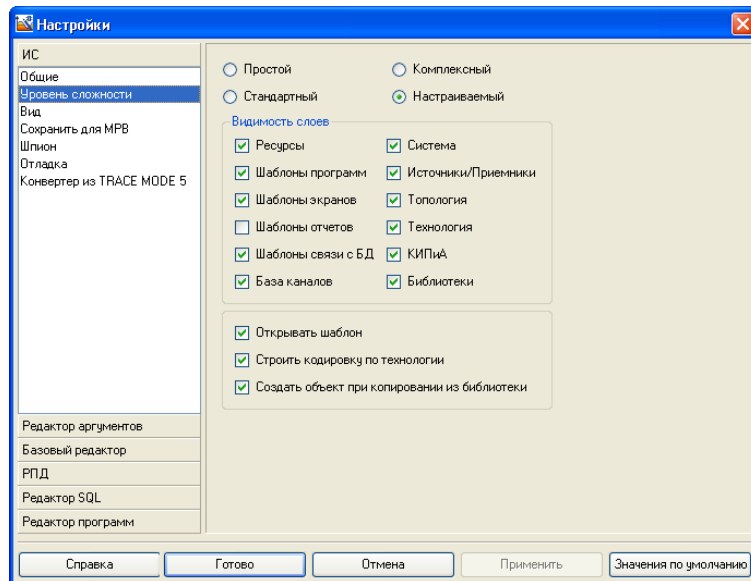
Панель содержит также следующие инструменты:

 – синхронизировать с деревом проекта (по этой команде в навигаторе проекта будет выделен компонент, окно редактора которого активно, а также слой (группа), содержащая этот компонент);

 – закрыть активное окно.

Задание общих настроек ИС

Для задания общих настроек ИС и редакторов шаблонов предназначен диалог, который открывается по команде **Настройки ИС** меню **Файл**:



Настройки ИС задаются на одноименной вкладке этого диалога, набор




инструментов которой изменяется в зависимости от выбранного в левом списке раздела, и вкладке **Базовый редактор**.

Общие настройки редакторов, которые могут быть заданы в диалоге **Параметры**, рассматриваются в разделах, посвященных работе в этих редакторах.

Вкладка ‘Интегрированная среда разработки’

Раздел ‘Общие’

При выборе этого раздела вкладка содержит следующие инструменты:

- переключатели режима отображения редакторов компонентов – **Открывать редакторы как вкладки** и **MDI**. В первом режиме каждый открытый редактор занимает всю рабочую область ИС и изменение размеров окна редактора недоступно. В режиме **MDI** размеры окна каждого редактора могут быть изменены (см. **Типовые операции редактирования**), при этом доступны стандартные инструменты окон (иконка в левом верхнем углу окна – открыть меню, содержащее стандартные команды для работы с окном,  – минимизировать окно,  – восстановить окно,  (**CTRL+F4**) – закрыть окно).
- список **На старте**, определяющий параметры среды при ее запуске. Этот список включает следующие опции:
 - пустая;
 - загрузить проект (файл **prj**), редактировавшийся в ИС последним;
 - создать новый проект;
 - открыть диалог выбора проекта (файла **prj**).
- список, с помощью которого задается глубина списка последних открытых проектов (файлов **prj**);
- флаг **Открывать группы при drag-n-drop** – если этот флаг установлен, группа, над которой в навигаторе проекта выполняется операция методом drag-and-drop, автоматически раскрывается;
- флаг **Показывать содержимое группы в правом окне** – если этот флаг установлен, содержимое выделенной группы отображается в правом окне навигатора;
- флаг **Печатать после документирования проекта** – если этот флаг установлен, файл, в который документируется проект, будет автоматически выведен на печать.

Раздел ‘Уровень сложности’

При выборе этого раздела вкладка содержит переключатели уровня сложности проекта (задают отображаемый набор слоев):

- **простой** – отображаются слои **Ресурсы, Система, Источники/Приемники и Библиотеки компонентов**;
- **стандартный** (значение по умолчанию) – отображаются те же слои, что и для простого уровня, и все слои шаблонов (экранов, программ, связей с БД и документов);
- **комплексный** – отображаются все слои, кроме слоя **База каналов**;
- **настраиваемый** – при выборе этого уровня в диалоге доступны переключатели отображения всех слоев, включая слой **База каналов**.

Если слой **База каналов** скрыт, то:

- при выделении канала в навигаторе проекта недоступна команда **Удалить** (канал можно только уничтожить – см. **Меню и главная панель инструментов навигатора проекта**).

Кроме того, вкладка содержит следующие инструменты:

- **Открывать шаблон по умолчанию** – если этот флаг не установлен (значение по умолчанию), двойное нажатие ЛК на канале класса CALL в навигаторе эквивалентно команде **Редактировать проект** (см. **Меню и главная панель инструментов навигатора проекта**). При установке флага двойное нажатие ЛК на канале класса CALL эквивалентно команде **Редактировать шаблон**;
- **Строить кодировку по технологии** – если этот флаг установлен, кодировка канала, создаваемого в слое **Технология**, строится автоматически с наследованием кодировки всех объектов вышестоящего уровня, в которые канал входит; в противном случае для канала задается кодировка по умолчанию;
- **Создавать объект при копировании из библиотеки** – от этого флага зависит результат копирования и вставки объекта библиотеки компонентов в узел (см. **Копирование и вставка объекта структуры**).

Раздел ‘Вид’

При выборе этого раздела вкладка содержит два раздела – **Вид имен и Стили ИС**.

В разделе **Вид имен** содержатся переключатели вида отображения имен объектов в навигаторе проекта:

- **Имя** (значение по умолчанию);

- **Кодировка.**

Если выбран вид отображения **Кодировка**, при сохранении проекта атрибуту **Имя** объекта присваивается значение кодировки, атрибуту **Кодировка** – имя. Выбранный вид отображения влияет также на отображение пути к выделенному объекту в строке статуса навигатора.

В разделе **Стили ИС** содержатся переключатели, определяющие следующие стили отображения ИС: **TRACE MODE 6, CDE, Windows, WindowsXP, Motif, MotifPlus, Platinum, SGI.**

Раздел ‘Сохранить для MPB’

При выборе этого раздела на вкладке задаются параметры оптимизации экспорта узлов:

- флаг **Подробный отчет** – если этот флаг установлен, протокол конвертера (файл *.cnv – см. **Файлы узла, создаваемые при экспорте**) содержит информацию о каналах, шаблонах и объектах базы каналов;
- флаг **Не создавать каналы CHCS** – если этот флаг установлен, MPB не создает каналы по аргументам, к которым привязаны источники/приемники и удаленные каналы. Консоли игнорируют данный флаг;
- флаг **Файлы *.sav** – если этот флаг не установлен, конфигурация каналов CALL проверяется при каждом экспорте узла. При первом экспорте с установленным флагом такая проверка также производится, а для каждого канала CALL в папке узла создается конфигурационный файл **<ID канала>_<число аргументов>.sav**. При последующих процедурах экспорта с установленным флагом:
 - проверяются новые каналы CALL и для них создаются файлы *.sav;
 - проверяются старые каналы CALL, в которых изменено число аргументов (создаются новые файлы *.sav);
 - не проверяются старые каналы CALL, в которых число аргументов не изменилось, и соответствующие файлы *.sav не обновляются – это приводит к уменьшению времени экспорта проекта.

Если проект экспортирован при установленном флаге, MPB считывает конфигурацию каналов CALL из файлов *.sav;

- флаг **Распределить экспорт** – этот флаг нужно установить для распределения экспорта по процессорам (для многопроцессорных компьютеров);
- флаг **Каналы по кодировке** –
- флаг **Стандартные объекты** – если этот флаг установлен, при

экспорте в файл узла добавляется представление каналов, распределенных по **стандартным объектам TRACE MODE**:

- **1, Hardware_Src** – каналы обмена с УСО;
- **2, RS_Src** – каналы обмена по RS;
- **3, Other_Src** – другие каналы обмена (DCS, OPC и т.п.), а также генераторы и модели;
- **4, RemoteNode_Src** – каналы приема автопосылок;
- **5, Calculation** –
- **6, Calls** – каналы **CALL**;
- **7, Receive by Autosend** – каналы приема автопосылок;
- **8, Users** – каналы **Пользователь**;
- **9, Fast** – каналы с периодом пересчета **быстрый**;
- **10, Events** – каналы **Событие**;
- **11, M-Resource** – каналы **M-Resource**;
- **12, S-Resource** –
- **13, D-Resource** – каналы **D-Resource**;
- **14, Personals** – каналы **Персонал**;
- **15, Passports** – каналы **Единица оборудования**;
- **16, TF-Mathematics** –
- **17, Service** – системные переменные TRACE MODE;
- **18, N_Restored** – каналы **CHCS**;
- **19, Break_Link** –

Вне зависимости от флага **Создать стандартные объекты**, в файле узла создаются стандартные объекты **Receive by Autosend**, **Users** и **Fast**, а также объект <имя dbb-файла>, который содержит все каналы узла (этот объект имеет имя узла, если для узла установлен флаг **Загрузить** – см. **Вкладка 'Флаги'**). Кроме того, в файле узла создаются все объекты, для которых флаг **Загрузить** установлен пользователем.

Стандартные объекты TRACE MODE отображаются только в мониторе с отладкой (профайлере);


- список **Глубина отслеживания источников**, с помощью которого задается порядковый номер источника/приемника в цепочках связей **канал-источник/приемник1-источник/приемник2**, который будет использован при экспорте узла (см. **Допустимые цепочки связей**):
 - **Игнорировать привязку** – игнорировать привязки каналов к источникам/приемникам;
 - **1** – первый (**источник/применик1**) (значение по умолчанию);
 - **2** – второй (**источник/применик2**).

Раздел ‘Шпион’

При выборе этого раздела вкладка содержит инструменты конфигурирования режима **Шпион** интегрированной среды разработки (см. **Отладка проекта**):

- номер сетевого адаптера (1, 2 или 3), через который ИС должна получать данные с работающих узлов. Для получения данных с узла, который выполняется под управлением монитора на том же компьютере, на котором запущена ИС, этот узел должен быть настроен на адаптер, отличный от выбранного для режима **Шпион**, – т.е. в этом случае компьютер должен иметь, по меньшей мере, два сетевых адаптера;
- список, с помощью которого задается периодичность запроса данных с работающих узлов (1–60 с);
- флаг **Использовать ID проекта** – при установке этого флага ИС будет запрашивать данные с работающих узлов с одним и тем же **кодом проекта** (первым обнаруженным). Параметр **код проекта** задается в редакторе узла (см. **Редактор параметров узла**).

Раздел ‘Отладка’

При выборе этого раздела вкладка содержит переключатели для выбора профайлера, который будет запущен по команде  **Отладка** (только для узлов **RTM, Console, T-Factory** и **T-Factory_Console**):

- Профайлер с поддержкой графических экранов;
- Профайлер без поддержки графических экранов.

Для остальных узлов (см. **Классификация узлов**) всегда запускается профайлер без поддержки графических экранов.

Раздел ‘Конвертер из предыдущей версии’

При выборе этого раздела вкладка содержит флаг **Отключить конвертирование графической базы**. Установка этого флага отключает конвертирование графической базы проекта при его импорте в TRACE MODE 6 (см. **Меню ‘Файл’** и **главная панель инструментов ИС**).

Вкладка ‘Базовый редактор’

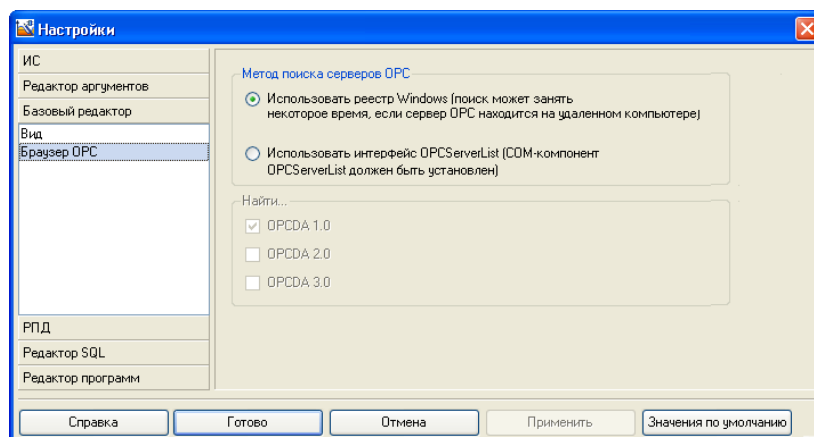
Раздел ‘Вид’

При выборе этого раздела вкладка содержит переключатель, от которого зависит результат выполнения команды на переход к редактированию компонента, имеющего аргументы.

Если выбрана опция **Редактор компонента** (значение по умолчанию), при выполнении команды открывается редактор компонента. Если выбрана опция **Редактор аргументов компонента**, при выполнении команды открывается редактор аргументов компонента.

Раздел 'Браузер OPC'

При выборе этого раздела вкладка принимает следующий вид:




В верхней части вкладки расположены переключатели выбора метода поиска OPC-серверов:

- **Использовать реестр** – через реестр WINDOWS;
- **Использовать OPC интерфейс** – при выборе этого метода в нижней части доступно указание спецификаций OPC – **OPCDA 2.0** и/или **OPCDA 3.0** (см. **Обмен по OPC**).

Сохранение проекта

Сохранение проекта для редактирования


По команде  **Сохранить (Ctrl-S)** или **Сохранить как (Ctrl-Shift-S)** из меню **Файл** проект сохраняется в бинарный файл с расширением **prj** для последующего редактирования в ИС.

При выполнении этих команд пользовательские библиотеки компонентов сохраняются в файл **tmdevenv.tmul** (в директории ИС).

В ИС предусмотрено резервирование предыдущей версии файлов **prj** и **tmul** – при повторном выполнении команды **Сохранить** расширение

файлов, сохраненных ранее, изменяются соответственно на `~prj` и `~tmul`.

Сохранение проекта для запуска


По команде  **Сохранить для MPB** меню **Файл** или панели инструментов ИС все узлы экспортируются в наборы файлов для их последующего копирования на аппаратные средства, на которых они должны исполняться под управлением мониторов TRACE MODE.

Параметры экспорта узлов задаются в диалоге конфигурирования ИС (см. **Задание общих настроек ИС** и **Раздел 'Сохранить для MPB'** в разделе **Вкладка 'Интегрированная среда разработки'**).

Перед экспортом узлов проект должен быть сохранен в файл `prj`.

При выполнении команды **Сохранить для MPB** в директории, содержащей файл `prj`, создается поддиректория **<имя файла prj без расширения>**, в которой для каждого узла создается папка с набором файлов. Папка узла имеет имя, заданное для узла при его конфигурировании в ИС (с заменой пробелов символами «_»). Файлы узлов, имеющих одинаковые имена в ИС, экспортируются в одну папку.

Необходимым условием экспорта узла является наличие в нем хотя бы одного канала.

По команде  **Сохранить узел для MPB** из меню **Проект** или контекстного меню навигатора выделенный узел экспортируется в произвольную папку, при этом при повторном экспорте резервные копии узла не создаются.

В случае применения к слою **Технология**, команда **Сохранить узел для MPB** идентична команде **Сохранить для MPB** для узла.

Файлы узла, создаваемые при экспорте

При экспорте в папке узла создаются следующие файлы:

- **<имя файла prj>_<ordinal>.cnv** – текстовый протокол конвертера `FileCnv32.dll`;

Ordinal – это порядковый номер узла (см. **Имена и идентификаторы объектов структуры**). Порядковый номер узла не следует путать с его индивидуальным номером, который используется при обмене по сети.

- **<имя файла prj>_<ordinal>.dbb** – бинарный файл узла (загружается монитором);
- **<имя файла prj>_<ordinal>.dbx** – бинарный файл узла;
- **<имя файла prj>_<ordinal>.rtp** – вспомогательный файл узла (текстовый);

Файл **rtp** используется при загрузке в **Микро MPB** – по имени этого файла монитор находит и загружает файлы **dbb**.

Несмотря на одно и то же расширение, в ряде случаев файлы **dbb** имеют различный внутренний формат (в зависимости от типа узла, для которого создаются).

- **addr.ind** – описание параметров узлов;
- **<имя программы>.tmsd** – программа, вызываемая каналами узла (код для отладки в ИС);
- ***.res** – шаблоны и ресурсы (библиотеки текстов, библиотеки картинок, библиотеки видеоклипов и графические объекты из слоя **Ресурсы**), используемые в узле (исполняемый код);
- **t_factory.txt** – информация о каналах класса **Пользователь** и каналах **T-FACTORY**;
- **<имя файла prj>_<ordinal>_opc<Num>.cnf** – конфигурационный файл обмена с сервером OPC, **Num** – порядковый номер файла (нумерация начинается с 0);
- **<имя файла prj>_<ordinal>.inf** – в этот файл записывается количество каналов в узле.

В зависимости от параметров экспорта, в папке узла могут создаваться дополнительные файлы.

Редактирование структуры проекта

Навигатор имеет следующие средства для редактирования структуры проекта:

- меню **Проект**;
- панели инструментов;
- контекстное меню.

Кроме того, в навигаторе поддерживается метод перетаскивания объектов мышью (метод drag-and-drop – см. **Типовые операции редактирования**), а также его модификации (drag-and-drop с удержанием служебных клавиш).

Для конфигурации/разработки объектов структуры в навигаторе предусмотрены команды **Свойства** и **Редактировать**, с помощью которых для каждого объекта структурного дерева могут быть открыты соответствующие **окно свойств** и **редактор**.

Меню и главная панель инструментов навигатора проекта



Меню **Проект**, главная панель инструментов и контекстное меню навигатора проекта содержат набор команд, который соответствует выделенному объекту структурного дерева. Для выделения объекта нужно нажать на нем ЛК. Групповое выделение объектов в навигаторе не поддерживается.

Меню **Проект**, главная панель инструментов и контекстное меню навигатора содержат как типовые команды для создания компонентов (групп компонентов), работы с буфером обмена и поиска (см. **Типовые средства редактирования**), так и специфические:



– такой вид приобретает типовой инструмент удаления **X** при выделении канала. При нажатии стрелки открывается меню, содержащее команды **X Удалить** (удалить без удаления из слоя **База каналов**) и **X Уничтожить** (удалить с удалением из слоя **База каналов**);

Если слой **База каналов** скрыт (см. **Задание общих настроек ИС**), команда удаления канала недоступна – в этом случае канал можно только уничтожить.

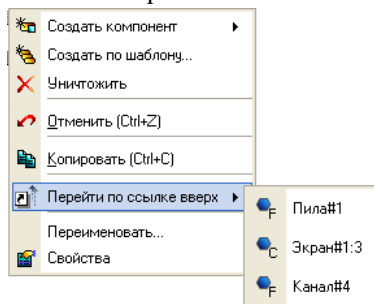


– такой вид приобретает в некоторых случаях типовой инструмент вставки . При нажатии стрелки открывается дополнительное меню, содержащее команды **Вставить** и **Вставить с привязкой** (см. **Копирование и вставка объекта структуры**);



Перейти по ссылке вниз/вверх – по этой команде открывается дополнительное окно навигатора, в котором:

- выделяется компонент, с которым связан данный компонент или который вызывается данным компонентом (в случае перехода по ссылке вниз);
- выделяется компонент, связанный с данным компонентом или вызывающий данный компонент (в случае перехода по ссылке вверх). Если подобных компонентов несколько, один из них выбирается в списке:



Если у компонента имеются связи/вызовы как вниз, так и вверх, данный инструмент принимает вид , и при нажатии стрелки открывается дополнительное меню, содержащее команды **Перейти по ссылке вниз** и **Перейти по ссылке вверх**;



Резервировать – создать резервные узлы для выделенного узла:

- **Нет** – не создавать (значение по умолчанию);
- **Один резерв** – создать один резервный узел для выделенного узла;
- **Два резерва** – создать два резервных узла для выделенного узла;




Редактировать – открыть выделенный объект структурного дерева в соответствующем **редакторе** (см. **Редакторы объектов структуры проекта**);





Редактировать шаблон – открыть шаблон, вызываемый данным компонентом, в соответствующем редакторе (см. **Редакторы**


объектов структуры проекта);


Переименовать (CTRL+ENTER) – перейти к редактированию имени выделенного объекта структуры. Для перехода к редактированию имени выделенного объекта можно также нажать на нем ЛК;


 **Свойства** – открыть **окно свойств** объекта структурного дерева (см. **Окно свойств объекта структуры проекта**);

 **Сохранить узел для МРВ** – экспортировать выделенный узел в произвольную папку. В случае применения к слою **Технология**, эта команда идентична команде **Сохранить для МРВ** для узла (см. **Сохранение проекта для запуска**);

 **Загрузить дамп узла** – восстановить содержимое и конфигурацию компонентов узла из его файла восстановления (дампа – см. **Задание параметров узла**);

 **Загрузить в контроллер** – по этой команде открывается диалог управления копированием/загрузкой узла в реальном времени (см. **Перезагрузка узла из ИС**);


 **Информация о проекте** – открыть одноименный диалог. Эта команда может быть выполнена также из контекстного меню навигатора проекта и с помощью панели инструментов ИС (см. **Меню 'Файл' и главная панель инструментов ИС**);


 **Синхронизировать с базой данных** – запустить мастер конфигурирования взаимодействия с технологической базой данных, эта команда доступна при выделении слоя **Технология** (см. **Взаимодействие с технологической БД**).

Управление внешним видом навигатора проекта










Для управления внешним видом навигатора проекта используются панели, содержащие следующие инструменты:

 – выбор размера иконок объектов структурного дерева в левой части навигатора:

 – маленькие (значение по умолчанию);

 – большие;

 – выбор вида представления объектов структурного дерева в правой части навигатора:

-  – большие иконки;
-  – маленькие иконки;
-  – список (значение по умолчанию);
-  – подробно. При использовании этой опции в правой части навигатора выводится та же информация об объекте, что и на вкладке **Информация** окна свойств (см. **Вкладка 'Информация'**);
-  – открыть дополнительное окно навигатора проекта. С помощью этой команды можно одновременно открыть несколько окон навигатора, для каждого из которых можно задать свой внешний вид;
-  – закрыть окно навигатора проекта.

Создание объектов структуры

Для создания объектов структуры (компонентов и групп компонентов) используются типовые команды меню **Проект**, контекстного меню и панели инструментов навигатора (см. **Меню и главная панель инструментов навигатора проекта** и **Типовые инструменты редактирования**).

При создании канала класса CALL с предустановленным свойством **ВЫЗОВ** в соответствующем слое шаблонов создается шаблон, вызываемый каналом. В навигаторе могут быть созданы следующие каналы класса CALL с предустановленным свойством **ВЫЗОВ**:

- **Экран** – канал с вызовом шаблона экрана;
- **Программа** – канал с вызовом шаблона программы;
- **Документ** – канал с вызовом шаблона документа;
- **Связь с БД** – канал с вызовом связи с базой данных.

Меню **Проект**, контекстное меню и панель инструментов навигатора содержат команды создания только тех объектов, которые может содержать выделенный слой/группа.

При редактировании сохраненного проекта (в том числе после выполнения команды **Сохранить/Сохранить как**) вновь созданные структурные объекты и объекты, их содержащие, выделяются в навигаторе синим шрифтом.

Имена и идентификаторы объектов структуры

В качестве имени объекта структуры можно использовать произвольное строковое выражение. Первый символ имени не может быть цифрой.

При экспорте узла ИС создает директорию с его именем (пробелы заменяются символами «_»).

MPB считывает первые 31 символ имени объекта.

Каждому объекту структурного дерева при его создании присваивается по определенному алгоритму уникальный **идентификатор (ID)**, отображаемый во всплывающей подсказке (см. **Отображение свойств объектов структуры**).

Объекты, входящие в группу, имеют в этой группе свои порядковые номера (по списку). Для изменения порядка расположения объектов в группе (и соответствующего изменения их порядковых номеров) используется операция перемещения в пределах группы (см. **Перемещение объектов структуры**).

Имя объекта структуры, за исключением шаблона, не является его идентификатором (т.е. объекты могут иметь одинаковые имена, а имена шаблонов должны быть разные).

Узлы нумеруются (начиная с 0) по порядку их создания в проекте и сохраняют свои порядковые номера (**ordinal**) при любых операциях с узлами в ИС.

Изменение класса канала после его создания

Операция изменения класса канала после его создания определена для каналов следующих классов:

- **FLOAT** – при установке флага **HEX** на вкладке **Флаги** окна свойств (см. **Окно свойств объекта структуры проекта**) такой канал меняет свой класс на **HEX16**;
- **HEX16** – при снятии флага **HEX** такой канал меняет свой класс на **FLOAT**;
- **HEX32** – при снятии флага **HEX** такой канал меняет свой класс на **FLOAT**.

При изменении класса уничтожаются все пользовательские настройки канала.

Выделение объекта структуры

Некоторые типовые операции с объектами структуры – создание по образцу, создание группы по образцу, копирование, удаление, а также открытие редактора и окна свойств – доступны после выделения объекта в навигаторе. Для выделения объекта структуры нужно нажать на нем ЛК.

Содержимое слоя (группы), выделенного в левом окне навигатора, отоб-

ражается в правом окне. Чтобы раскрыть группу компонентов, можно также дважды нажать на ней ЛК в правом окне.

Удаление объекта структуры

Для удаления выделенного объекта структуры (компонента или группы компонентов) используется типовая команда **Удалить** (см. **Меню и главная панель инструментов навигатора проекта и Типовые инструменты редактирования**).

При удалении канала (группы каналов) из любого слоя, кроме слоя **База каналов**, доступны две команды – **Удалить** (удалить без удаления из слоя **База каналов**) и **Уничтожить** (удалить с удалением из слоя **База каналов**). Если слой **База каналов** скрыт (см. **Задание общих настроек ИС**), команда удаления канала недоступна – в этом случае канал можно только уничтожить.

ИС автоматически удаляет привязки аргументов к компоненту, ссылки на компонент и вызовы компонента при удалении этого компонента (в случае канала – при его уничтожении).

Открытие окна свойств и редактора объекта структуры

Для открытия окна свойств или редактора объекта структуры используются команды меню **Проект**, главной панели инструментов и контекстного меню навигатора проекта (см. **Меню и главная панель инструментов навигатора проекта**).

Чтобы открыть компонент в редакторе, можно также дважды нажать на нем ЛК.

Групповое редактирование компонентов

Групповые операции редактирования возможны в окне свойств группы компонентов (см. **Вкладка 'Флаги'**) и в редакторе группы компонентов, если такой редактор существует (см., например, **Базовый редактор группы компонентов** в разделе **Редакторы групп компонентов**).

Операции, общие для объектов структуры

К операциям, общим для всех объектов структуры, относятся следующие:



- переименование;
- изменение кодировки;
- задание комментария;
- смена иконки.

Эти операции выполняются в окне свойств на вкладке **Информация** (см. **Вкладка 'Информация'**) или в редакторе, если редактор для данного объекта существует (см. **Редакторы объектов структуры проекта**).

Перемещение объектов структуры

Операция перемещения включает два действия – удаление объекта из места его начального расположения и вставку в указанную группу (слой).


Чтобы переместить объект, нужно перетащить его мышью в нужную группу (слой), удерживая клавишу **SHIFT**.

Если группа (слой), на которую указывает курсор в процессе перетаскивания, может содержать перемещаемый объект, курсор принимает вид , в противном случае – .

С помощью перемещения объектов в пределах группы можно изменять их порядковые номера (см. **Имена и идентификаторы объектов структуры**).

Копирование и вставка объекта структуры


Копирование объекта структуры в буфер обмена

Чтобы поместить копию выделенного объекта структуры в буфер обмена, нужно выполнить команду  **Копировать (Ctrl+C)** (см. **Меню и главная панель инструментов навигатора проекта и Типовые инструменты редактирования**).

Если копируется слой, в буфер обмена помещается копия его содержимого (всех дочерних групп и компонентов).

Обычная вставка объекта структуры из буфера обмена

При обычной вставке создается новый объект, который является копией объекта, помещенного в буфер обмена. Операция вставки не очищает буфер обмена, поэтому после однократного копирования объекта возможна его множественная вставка.



Для обычной вставки нужно выделить группу (слой), в которой может быть создан объект того же вида, что и объект, помещенный в буфер обмена, и выполнить команду  **Вставить (Ctrl+V)**.

Указанное условие (условие возможности создания в выде-

ленной группе или слое объекта того вида, который имеет объект, помещенный в буфер обмена) является существенным – в случае его нарушения выполняется специальная вставка (см. ниже).




Копирование и вставка объекта структуры с помощью мыши

Для копирования и вставки объекта нужно перетащить его мышью в нужную группу (слой) с удержанием клавиши **CTRL**. Если группа (слой), на которую указывает курсор в процессе перетаскивания, может содержать

копируемый объект, курсор принимает вид , в противном случае – . При выполнении данной операции объект не помещается в буфер обмена.

Специальная вставка объекта структуры

В навигаторе проекта поддерживается перетаскивание объектов мышью с одновременным удержанием клавиш **CTRL** и **SHIFT**. Если объект, на который указывает курсор в процессе перетаскивания, допускает заверше-

ние операции, курсор принимает вид , в противном случае – . Эквивалентом этой операции является копирование и специальная вставка объекта по команде  **Вставить с привязкой** – см. **Меню и главная панель инструментов навигатора проекта**.

С помощью данной операции выполняются следующие действия:

- при перетаскивании канала (группы каналов) в слой (группу) – автопостроение канала (группы каналов) с настроенным свойством **СВЯЗЬ** (каждый канал, созданный таким образом, связан с соответствующим исходным каналом и имеет тот же класс);
- при перетаскивании источника/приемника (группы источников/приемников) в слой (группу) – автопостроение канала (группы каналов) соответствующего класса с настроенным свойством **СВЯЗЬ** (каждый канал, созданный таким образом, связан с соответствующим источником/приемником);
- при перетаскивании шаблона (группы шаблонов) в слой (группу) – автопостроение канала (группы каналов) класса CALL (каждый канал, созданный таким образом, настроен на вызов соответствующего шаблона и имеет такое же имя, что и шаблон);
- при перетаскивании источника/приемника или шаблона на компонент – настройка компонента на связь с источником/приемником или на вызов шаблона. В последнем случае канал класса CALL

принимает имя шаблона.

Автоматический выбор вида операции вставки

При перемещении объекта структуры обычным методом drag-and-drop (без удерживания служебных клавиш) вид операции вставки (обычная или специальная) выбирается автоматически (идентифицируется по форме курсора).

Копирование и вставка узла в библиотеку компонентов

При копировании и вставке узла в библиотеку компонентов в ней создается новый объект с именем узла (см. **Группы слоя 'Библиотеки компонентов'**). Компоненты проекта, используемые в узле, копируются в соответствующие подгруппы объекта с сохранением структурирования и связей.

Копирование и вставка объекта библиотеки в узел

Результат копирования и вставки объекта библиотеки компонентов (см. **Группы слоя 'Библиотеки компонентов'**) в узел или произвольную группу узла зависит от флага **Создавать объект при копировании из библиотеки** (см. **Задание общих настроек ИС**).

Если флаг установлен, в узел или его группу копируется объект.

Если флаг не установлен, в узел или его группу копируется содержимое объекта.

В обоих случаях компоненты подгрупп объекта копируются в соответствующие слои проекта с сохранением структурирования и связей.

Перепривязка каналов и аргументов при копировании/вставке и перемещении объекта базы каналов

Объект базы каналов в общем случае включает каналы и иерархические группы каналов. Каналы объекта могут иметь настроенные свойства **СВЯЗЬ** и **ВЫЗОВ**, обладая связями двух типов:

- тип 1 – связи с каналами объекта;
- тип 2 – связи с каналами, не входящими в состав объекта.


В результате копирования/обычной вставки или перемещения объекта (пусть его имя – А) создается новый объект В, при этом связи каналов объекта В будут следующими:

- типа 1 – изменятся на привязки к каналам объекта В, соответству-

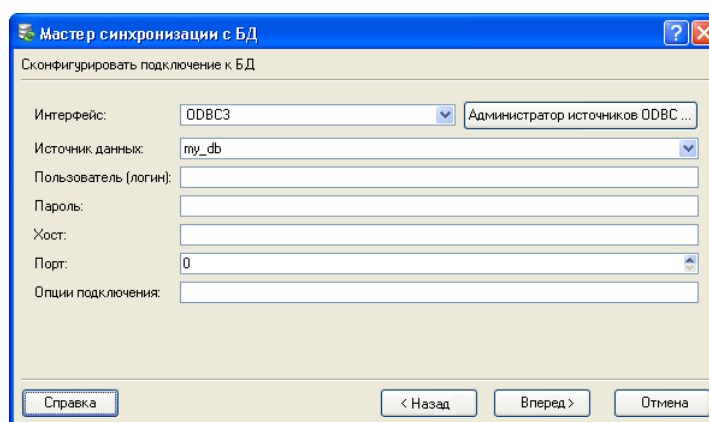
- ющим каналам объекта А;
- типа 2 – не изменятся.

Каналы и аргументы, не имеющие привязки в объекте А, будут таковыми и в объекте В.

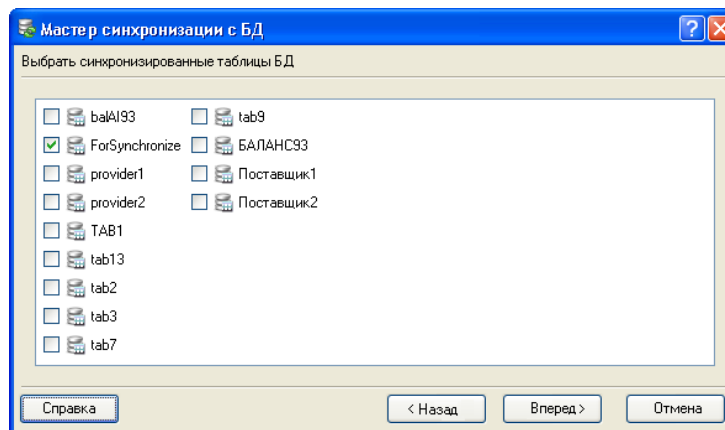
Взаимодействие с технологической БД

Для слоя **Технология** предусмотрена команда  **Синхронизировать с базой данных** (см. **Меню и главная панель инструментов навигатора проекта**), по которой запускается мастер конфигурирования взаимодействия с технологической БД.

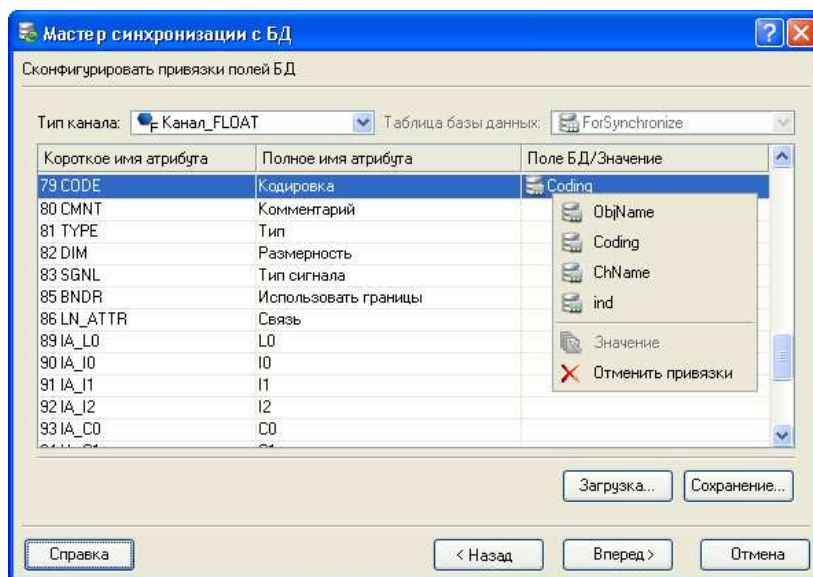
В первом диалоге мастера конфигурируется подключение к БД (см. **Подключение к базе данных**):



В следующем диалоге мастера выбирается таблица БД:



После нажатия кнопки **Вперед** на экране появляется следующий диалог мастера:



Этот диалог содержит следующие инструменты:

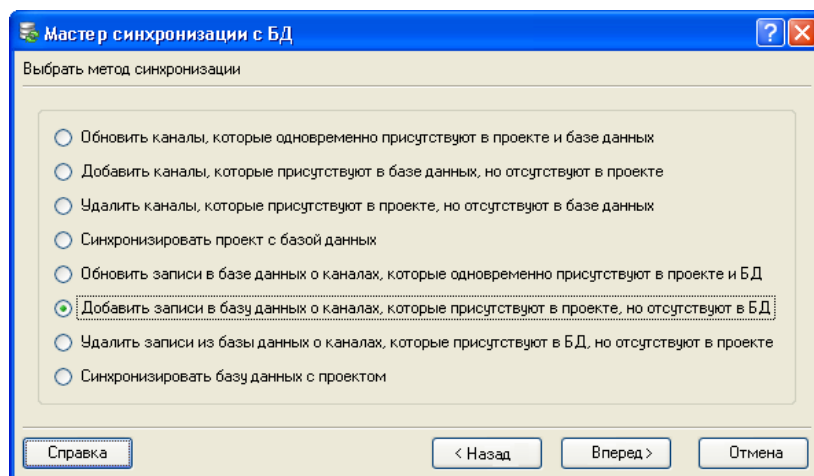
- список **Тип канала** – при любом из видов синхронизации проекта с БД (см. ниже) в проекте будут созданы каналы указанного в этом списке класса;
- таблица – предназначена для задания соответствия атрибутов полям БД. Для синхронизации достаточно задать привязку атрибута **Кодировка** или **Имя**. Поле БД для привязки выбирается из меню, которое появляется на экране при нажатии ПК в поле **Поле БД/Значение** таблицы.

Если поле БД, которое ставится в соответствие атрибуту **Кодировка**, содержит кодировку **AKS** или **KKS**, при синхронизации проекта с БД ИС создает структуру, соответствующую кодировке.

Меню содержит также следующие команды:

- **Отменить привязки** – удалить выделенную привязку;
- **Значение** – по этой команде, доступной при выделении числового атрибута, значение может быть введено с клавиатуры;
- **Сохранение** – по этой команде на экране появляется диалог задания имени, под которым заданные привязки будут сохранены. Именованные наборы привязок, а также последняя редакция привязок, сохраняются в файл **tmdevenv.dbmp**.
- **Загрузка** – по этой команде на экране появляется диалог выбора ранее сохраненных привязок.

В последнем диалоге мастера задается вид синхронизации:



Отображение свойств объектов структуры


Для отображения свойств объекта структуры проекта используются:


- окно свойств (см. **Окно свойств объекта структуры проекта**);
- редактор (см. **Редакторы объектов структуры проекта**);
- правая часть навигатора (см. **Управление внешним видом навигатора проекта**);
- иконка;
- всплывающая подсказка.

Модификация иконки объекта структуры

В навигаторе каждый объект структурного дерева снабжен основной иконкой, соответствующей его типу. ИС накладывает на основную иконку дополнительные микро-иконки (одновременно не более 4-х) для отображения важнейших характеристик объекта. Ниже микро-иконки сгруппированы по занимаемой позиции:

- Слева вверху:
 - канал типа INPUT;
 - канал типа OUTPUT;
 - число ссылок на компонент или число его вызовов отлично от 0;
- Слева внизу:
 - (черная стрелка) – для компонента задано свойство **связь**;

 (зеленая стрелка) – для компонента задано свойство **ВЫЗОВ**;

 (оранжевая стрелка) – для компонента одновременно заданы свойства **СВЯЗЬ** и **ВЫЗОВ**;

- Справа вверху:

 – объект открыт в редакторе.

В качестве примера ниже приведен вид модифицированной иконки канала:



Всплывающая подсказка к объекту структуры

Все объекты структурного дерева снабжены всплывающими подсказками, в которых отображаются их важнейшие свойства:

ID:	12
Кодировка:	ТС5
Узел:	Система.RTM_2
Link:	Saw#1:Значение(Источники/Приемники.Generators_1)
Вызов:	Шаблоны_программ.Program_#1

В первой строке подсказки отображаются идентификатор (**ID**) объекта.

Во второй строке подсказки отображается кодировка типа объекта (**Coding**).

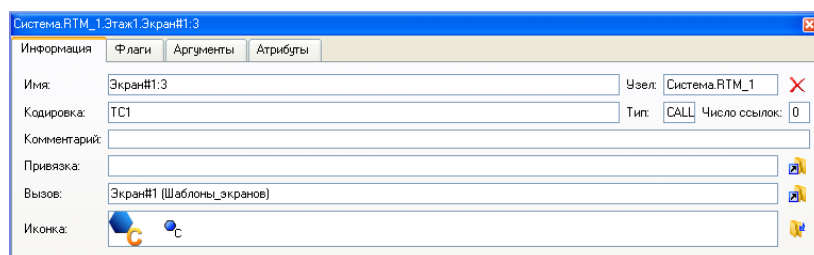
Для объекта слоя **Система** в подсказке отображается имя узла (**Node**), к которому принадлежит объект.

В подсказке отображается также конфигурация свойств **СВЯЗЬ** и **ВЫЗОВ**, если эти свойства заданы для объекта.




Окно свойств объекта структуры проекта

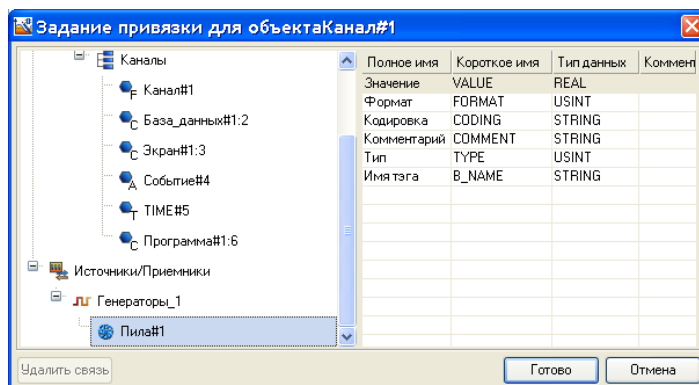
По команде **Свойства** (см. **Меню и главная панель инструментов навигатора проекта**) в нижней части ИС открывается **окно свойств** выделенного объекта структуры проекта.

Вкладка 'Информация'





Эта вкладка присутствует в окне свойств любого объекта структуры. На ней отображаются/редактируются следующие свойства:


- **Имя** – имя объекта (редактируется);
- **Кодировка** – кодировка типа объекта, задаваемая по умолчанию в TRACE MODE. Кодировку объекта можно изменить с помощью этого поля;
- **Комментарий** – комментарий (редактируется);
- **Иконка** – иконка, заданная для объекта по умолчанию. Для смены иконки надо нажать кнопку  и выбрать иконку в появившемся меню;
- **Тип** – predetermined in TRACE MODE type of object;
- **Узел** – узел, в который входит объект. При нажатии кнопки  объект удаляется из узла (для канала – без удаления из слоя **База каналов**, если этот слой открыт – см. **Задание общих настроек ИС**);
- **Счетчик ссылок** – число компонентов, связанных с данным компонентом или вызывающих данный компонент;
- **Привязка** – конфигурация свойства **связь**. Чтобы задать связь, нужно нажать кнопку  справа от данного поля и выбрать в диалоге, показанном на рисунке ниже, компонент (канал или источник/приемник) и его атрибут:

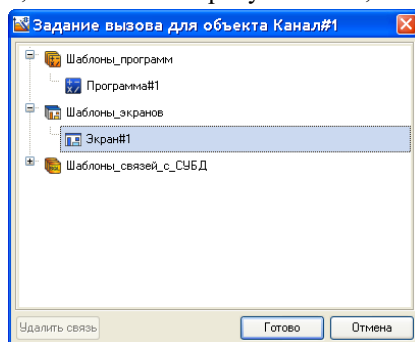



Чтобы удалить связь, надо нажать кнопку  этого диалога.

При задании связи с каналом другого узла справа от поля **Связь** появляется список , в котором выбирается интерфейс взаимодействия между узлами, который будет задействован при реализации связи (см. **Связь канал-канал**);

В слоях, отличных от слоя **Система**, инструмент  недоступен (т.е. интерфейс взаимодействия между каналами не может быть задан). Данный инструмент также отсутствует, если связываемые каналы принадлежат одному узлу – в этом случае задание интерфейса не требуется.

- **Вызов** – конфигурация свойства **вызов**. Чтобы задать это свойство, нужно нажать кнопку  справа от данного поля и выбрать в диалоге, показанном на рисунке ниже, вызываемый шаблон:

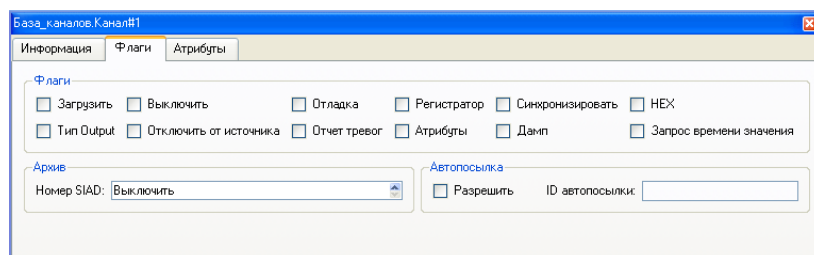


Чтобы удалить вызов, надо нажать кнопку  этого диалога

Вкладка 'Флаги'

Эта вкладка присутствует в окне свойств группы компонентов слоя **Система** и в окне свойств канала вне зависимости от слоя.

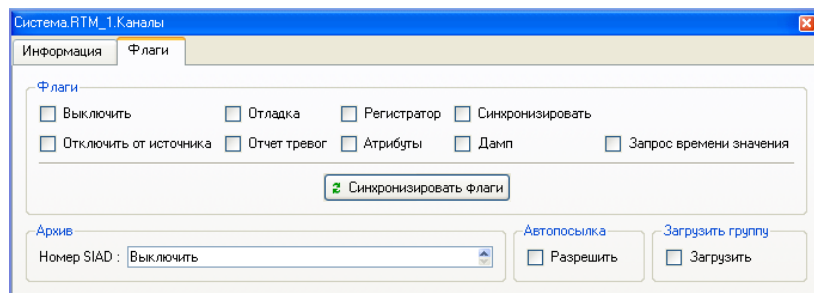
Для канала вкладка имеет следующий вид



На этой вкладке может быть задана часть атрибутов канала, доступных в разделе **Системные** редактора (см. **Общие атрибуты каналов**). Кроме того, на этой вкладке присутствуют следующие флаги:

- **HEX** – этот флаг используется для изменения класса канала после его создания (см. **Изменение класса канала после его создания**);
- **Загрузить** – этот флаг предназначен для отладки TRACE MODE и не используется при разработке проекта;
- **Запрос времени значения** (атрибут 50) – этот флаг используется в случае записи в канал значения некоторого канала другого узла при связи по любому из собственных протоколов TRACE MODE (в т.ч. по OPC):
 - если флаг для принимающего канала не установлен, в его атрибут **Время изменения** записывается время приема;
 - если флаг для принимающего канала установлен, в его атрибут **Время изменения** записывается значение атрибута **Время изменения** передающего канала.

Для группы компонентов вкладка имеет следующий вид:



Эта вкладка может использоваться для установки некоторых атрибутов компонентов группы – для этого нужно установить значения атрибутов и нажать кнопку **Синхронизировать флаги**. Синхронизация распространяется только на компоненты, входящие в группу непосредственно (если группа содержит подгруппу компонентов, то на них синхронизация не распространяется). Сбрасывание флагов в окне свойств группы не при-

водит к сбрасыванию соответствующих флагов компонентов, входящих в группу.

Если флаг **Загрузить** установлен, группе присваивается ID, и монитор загружает ее как объект.

Вкладка 'Аргументы'

Эта вкладка отображается в окне свойств шаблонов и каналов класса **CALL** (компонентов, создание аргументов для которых разрешено), а также в окне свойств других компонентов с настроенным свойством **Вызов** (см. **Табличный редактор аргументов**).

Вкладка 'Атрибуты'

Эта вкладка носит только информационный характер и отображается в окне свойств любых компонентов слоев **Система** и **Источники/Приемники**, а также для канала вне зависимости от слоя.

На эту вкладку выводятся атрибуты компонента, которые могут участвовать в межкомпонентном взаимодействии, их мнемоническое обозначение (**короткое имя**) и тип данных.

Редакторы объектов структуры проекта

По команде **Редактировать** (см. **Меню и главная панель инструментов навигатора проекта**) выделенный объект структуры открывается в соответствующем редакторе.

Открыть компонент проекта в редакторе можно также с помощью нажатия клавиши **ENTER** или двойного нажатия ЛК на компоненте.

С помощью команды **Редактировать шаблон** можно открыть в соответствующем редакторе шаблон, вызываемый выделенным компонентом.

Редакторы компонентов


Редакторы каналов



Начальное конфигурирование каналов выполняется при разработке проекта в ИС в соответствующем редакторе и/или в окне свойств (см. **Окно свойств объекта структуры проекта**). Установке флага в редакторе соответствует, как правило, посылка 1 в соответствующий атрибут.

Для каждого класса канала в ИС встроен свой редактор (ниже показан редактор канала класса FLOAT без аргументов):

Два раздела являются общими для всех редакторов каналов – верхний, содержащий поля задания имени, комментария и кодировки и кнопку вызова справки (справка вызывается также по нажатию **F1**), и раздел **Системные**, содержащий вкладки **Основные**, **Архивация** и **Дополнительно** (атрибуты, общие для каналов всех классов, описаны в разделе **Общие атрибуты каналов**).

Другие разделы редакторов содержат инструменты задания атрибутов, специфичных для каналов соответствующих классов.

Редакторы каналов содержат ту же панель инструментов  для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**). Если канал имеет аргументы, редактор имеет дополнитель-

ную вкладку-редактор аргументов, а панель содержит дополнительные переключатели ( – атрибуты,  – аргументы).

Редактор шаблонов программ

Редактирование программ рассматривается в разделе **Программирование алгоритмов**.

Редактор шаблонов экранов

Редактирование экранов (панелей) оператора рассматривается в разделе **Разработка графического интерфейса**.

Редактор шаблонов документов

Редактирование документов рассматривается в разделе **Генерация документов**.

Редактор связей с базами данных

Редактирование связей с БД рассматривается в разделе **Обмен с базами данных**.

Редактор библиотек текстов

Редактирование библиотек текстов рассматривается в разделе **Операции с ресурсными библиотеками**.

Редактор библиотек изображений

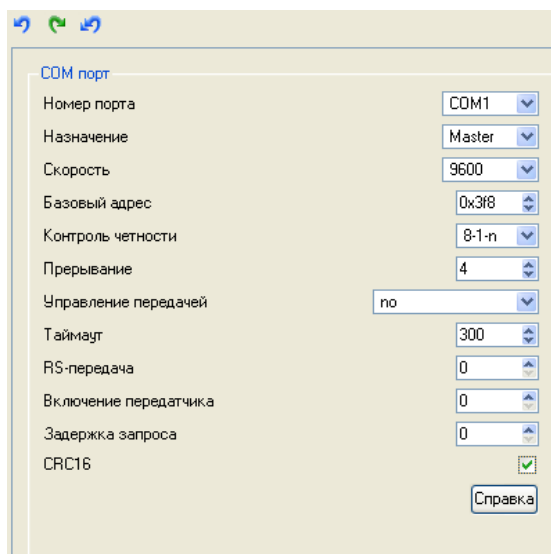
Редактирование библиотек изображений рассматривается в разделе **Операции с ресурсными библиотеками**.

Редактор библиотек видеоклипов

Редактирование библиотек видеоклипов рассматривается в разделе **Операции с ресурсными библиотеками**.

Редактор параметров СОМ-порта

Вид редактора последовательного порта показан на рисунке:



Редактор содержит ту же панель инструментов для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**).

В этом редакторе задаются следующие параметры.

Номер порта

Номер порта, выбирается из списка COM1...COM32 (`rtms_COM_NUM.tmc`).

Назначение

Назначение порта (указание для монитора используемого оборудования, протокола обмена и начального статуса узла при обмене по данному порту). Значение этого параметра выбирается из списка `rtms_COM_DEST.tmc` (номер в скобках соответствует номеру в файле):

- **(0) MASTER** – задание для узла статуса MASTER. Это значение устанавливается при обмене по всем протоколам, кроме M-LINK – см. **Обеспечение работы распределенных АСУ**;
- **(1) SLAVE** – задание для узла статуса SLAVE при обмене по M-LINK;
- **(2) MODEM** – это значение устанавливается в случае подключения модема к порту (модем конфигурируется в редакторе узла – см. **Редактор параметров узла**). Для обмена используется протокол M-LINK, узел может работать одновременно как MASTER и как SLAVE;
- **(3) GSM-SMS** – к порту подключен GSM-модем для обмена SMS-сообщениями (см. **Пользовательские SMS**). Для обмена ис-

пользуется протокол M-LINK, узел может работать одновременно как MASTER и как SLAVE (каналы обмена по M-Link будут работать при таком назначении порта). Параметр **Таймаут** задает минимальный интервал отсылки SMS.

Для диагностики обмена используется канал CALL.ChGroupReq, к которому привязана системная переменная 15.11 **@Modem** (см. **Канал CALL.ChGroupReq**).

- **(4) MASTER M-LINK** – задание для узла статуса MASTER при обмене по M-LINK;
- **(5) RESERVE M-LINK** – это значение устанавливается для резервированных узлов в сети M-LINK;
- **(6) GPS** – к порту подключено устройство, поддерживающее протокол NMEA (например, GPS-приемник, см. **Поддержка протокола NMEA**);
- **(7) Card/Bar Reader** – к порту подключено устройство чтения магнитного носителя (см. **Чтение штрих-кодов и магнитных носителей**);
- **(8) GPRS** – это назначение должно быть задано при обмене по GPRS;
- **(9) DNode_Master** – для резервирования узлов с обменом по RS. **DNode_Master** – это аналог **(0) MASTER**, но перед включением прослушает линию.

Ключ **RS_DNODE_PERIOD_CHECK**=<число секунд> в файле *.spf: задает время прослушивания линии (2с по умолчанию).

RS_DNODE_COUNT_CHAR_CHECK=<число DEC> – 14 по умолчанию. Если число байт, принятых за время **RS_DNODE_PERIOD_CHECK**, больше, чем **RS_DNODE_COUNT_CHAR_CHECK**, RS переключается в активное состояние (переключение из пассивного состояния в активное).

RS_DNODE_EXCHANGE_COUNT=<число DEC> – 10 по умолчанию. Если счетчик обменов больше, чем **RS_DNODE_EXCHANGE_COUNT**, и число ошибок больше, чем число успешных обменов, RS из активного состояния переходит в режим прослушивания линии (переключение из активного состояния в пассивное).

RS_DNODE_PERIOD_RESET=<число секунд> – по умолчанию 30с. Если RS активен, и время, прошедшее с последней ошибки, больше, чем **RS_DNODE_PERIOD_RESET**, счетчики обменов и ошибок обнуляются.

SYNC_SWITCH_MODE_BY_COM=OFF/ON – **OFF** по умолчанию. Если **ON** и узел дублирован, по активности RS узел переходит в состояние TRACE или WORK.

Дополнительный режим

Узел дублирован. Для COM-порта задается назначение **DNode_Master**, но нет ни одного канала обмена по этому RS. COM-порты резервов должны быть соединены кабелем RS-232. Синхронизация резервов производится по сети. В такой конфигурации узел TRACE переключится в состояние WORK в том случае, если нет обмена по сети и у второго узла нет обмена по RS.

- **(10) DRs_Master** – для резервирования RS узла. Для задания номера резервного COM-порта используется параметр **RS передача**. Если резерва нет, **DRs_Master** работает так же, как **(0) MASTER**. При старте узла активен RS с младшим номером.

Алгоритм, описанный далее, выполняется 1 раз в 10с.

Если с последней ошибки прошло больше 30с, счетчики обменов и ошибок обнуляются.

Если число обменов больше, чем задано ключом , и число ошибок больше, чем число успешных обменов, RS считается неактивным, а другой должен стать активным (пока это только флаги).

RS_CHANGE_UNIT_BY_COM=OFF/ON – **OFF** по умолчанию. Если **OFF**, одновременно работает один RS (переключение работает). Если **ON**, одновременно шлют оба RS, и в случае 2 неудачных посылок по какому-нибудь RS устройство переключается на другой RS (у каналов меняется **C0**).

@RS_on_off – может выключать резервный COM-порт. При выключении текущего COM-порта предпринимается попытка перехода на резервный COM-порт;

- **(13) G_SMS** – это назначение предназначено для досылки данных в регистратор с помощью SMS (см. **Пользовательские SMS**). В отличие от назначения **(3) GSM-SMS**, каналы обмена по M-Link не будут работать при таком назначении порта;
- **(14) G_GPRS** – это назначение предназначено для досылки данных в регистратор по GPRS;
- **(15) E_Meter** – см. **Универсальный механизм обмена с электросчетчиками**;
- **(16) IEC 61107** – к порту подключено устройство, поддерживающее протокол IEC 61107 (например, счетчик ЦЭ6850М);
- **(17) Non-M-Link Slave** – для обмена по протоколу Slave, отличному от Slave M-Link (если МРВ поддерживает этот протокол; тип устройства задается в файле *.cnf);
- **(18) IEC 60870-101** – к порту подключено устройство, поддерживающее протокол IEC 60870-101 (телемеханика);
- **(19) Host Variant** – протокол задается в строке **Дополнительно источника/приемника**:

VL=<номер протокола в TRACE MODE>;

- (20) **Host OEM RS** – см. ниже;
- (30) **S_SMS** – для резервирования COM-порта. В узле создается основной COM-порт с назначением **(3) GSM-SMS** или **(13) G_SMS**, параметр **RS-передача** которого задает номер резервного COM-порта. Если резервный порт в узле не создан, он создается автоматически и имеет следующие настройки:
 - назначение – (30) **S_SMS**;
 - **RS-передача** – номер основного порта.

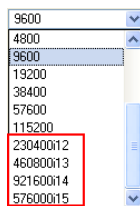
Резервный порт можно создать вручную, задав вышеуказанные параметры;

- (31) **UnUsed** –

Назначение порта сохраняется в файле **addr.ind** (см. **Файлы узла, создаваемые при экспорте**).

Скорость

Скорость обмена, выбирается из списка (**rtms_COM_SPD.tmc**). Список, в том числе, содержит значения с суффиксом **i<nn>**:



Если для порта задано одно из таких значений, то скорость обмена может быть переопределена в файле *.cnf с помощью ключа с соответствующим суффиксом:

- **COM_SPEED_I00**=<baud rate>
- **COM_SPEED_I12**=<baud rate>
- **COM_SPEED_I13**=<baud rate>
- **COM_SPEED_I14**=<baud rate>
- **COM_SPEED_I15**=<baud rate>

Скорость обмена по протоколу M-LINK не должна быть ниже 600 бит/с.

Базовый адрес

Базовый адрес порта.

Контроль четности

Данный параметр выбирается из списка (**rtms_COM_EVEN.tmc**), содер-

жащего значения в формате **k-m-x**:

- **k** – количество информационных бит в посылке;
- **m** – количество стоповых бит;
- **x** – наличие проверки на четность:
 - **n** – отсутствие проверки,
 - **e** – проверка на четность,
 - **o** – проверка на нечетность.

Прерывание

Номер используемого прерывания.

Управление передачей

Режим управления передатчиком. Значение этого параметра выбирается из списка **rtms_TRNFR_CTRL.tmc** (режимы с использованием специальных портов – для RS-485):

Значение	Посылка	Прием
no	нет управления	нет управления
DTR	DTR = on	DTR = off
RTS	RTS = on	RTS = off
DTR-RTS	DTR, RTS = on	DTR, RTS = off
360H	outp(0x360,1)	outp(0x360,0)
DTR, RTS=on	DTR,RTS = on	DTR, RTS = on
DTR=on	DTR = on	DTR = on
20CH	outp(0x20c,1)	outp(0x20c,0)
RTS=on	RTS=on	RTS=on
RTS(auto)	Внешнее управление RTS	Внешнее управление RTS
C3H	outp(0xc3,1)	outp(0xc3,0)
DTR=on, RTS(auto)	DTR=on, Внешнее управление RTS	DTR=on, Внешнее управление RTS
DTR/RTS	RTS=on, DTR=off	RTS=off, DTR=on

Помимо указанных в таблице, список содержит следующие режимы:

- **Modem_0**
- **Modem_1**
- **Modem_2**
- **Modem_3**
- **HART_ER**

Таймаут

Время ожидания ответа от устройства, которому был послан запрос (в миллисекундах, значение по умолчанию – 100 мс). Если в течение этого времени ответ на запрос от устройства или МРВ не пришел, запрашивающему каналу устанавливается флаг аппаратной недостоверности.

Существует механизм задержки выключения передатчика после окончания передачи, реализуемый с помощью передачи после кадра 1-2 лишних байтов. 2 байта формируются в том случае, если таймаут ожидания ответа больше или равен 200 мс. Следует учитывать, что для очень медленных линий передачи этой задержки выключения передатчика может оказаться недостаточно.

RS-передача

Задержка (в миллисекундах) между транзакциями (процедурами ЗАПРОС/КОМАНДА+ОТВЕТ).

При связи по модему этот таймаут задает время ожидания звонка. В режим ожидания звонка узел MASTER переходит в следующих случаях:

- узел не блокирован, для обмена нет ни одного канала;
- после каждого прохода по базе.

Для назначений СОМ-порта 2, 3, 10, 13 и 15 **RS-передача** задает номер резервного СОМ-порта (без смещения, т.е. 8 означает СОМ8).

Включение передатчика

Задержка (в миллисекундах) начала передачи данных после включения передатчика (реализуется только для узла со статусом MASTER).

Задержка запроса

Задержка передачи данных (в миллисекундах) после установления соединения при связи по модему. Величина этого таймаута должна быть меньше значения времени ожидания ответа;

Таймауты используются в случае подключения к порту дополнительного оборудования.

CRC16

Если этот флаг установлен, команда записи по M-LINK дополнительно защищается CRC16, а при приеме ответов по M-LINK вычисляется дополнительная контрольная сумма. Чтобы данный механизм работал, флаг **CRC16** должен быть установлен как для порта узла M-LINK MASTER, так и для порта узла M-LINK SLAVE. При обмене по DCS через порт с

установленным флагом **CRC16** выполняется дополнительная проверка ответов.

При связи по модему, после приема звонка и соединения узел SLAVE переходит в режим ожидания приема данных. Узел выходит из этого режима в следующих случаях:

- ситуация, когда узел вошел в режим приема, а у него ничего не запросили, повторилась трижды;
- узел MASTER «повесил трубку».

Параметры последовательных портов задаются в ИС для узлов. При запуске узлов под управлением мониторов эти параметры устанавливаются для портов аппаратных средств, на которых размещены узлы – при этом данная процедура имеет следующие особенности:

- параметры **Базовый адрес** и **Прерывание** для COM-порта в WINDOWS задаются средствами ОС;
- для работы в WINDOWS параметры **Базовый адрес** и **Прерывание** нельзя задавать нулевыми (рекомендуется указать их реальные значения, установленные в ОС);
- для контроллера ЛАГУНА наличие ненулевых настроек базового адреса и прерывания обязательно, хотя их конкретные значения несущественны.

Host OEM RS

Данное назначение порта используется для обеспечения специфического чтения собственных VCO, а также поддержки протоколов DCS, Modbus RTU и t11.

Регламент обмена:

- обмен производится в режиме «запрос-ответ» (без расширения транзакции в реальном времени);
- со стороны MPB нет управления UART (внешнее управление **RTS(auto)** поддерживается).

Настройки COM-порта имеют специфическое назначение:

- **Прерывание** – при $IRQ > 16$ и обнаружении ошибки в транзакции запрос повторяется до $(IRQ - 16)$ раз. Если ошибка воспроизводится, в канале устанавливается флаг аппаратной недостоверности;
- **Задержка запроса** – этот параметр определяет режим перехода к очередному циклу поиска каналов, требующих обслуживания, после завершения предыдущего прохода базы:
 - 0 (по умолчанию) – пауза длительностью 0.75 цикла потока CALC;

- 1 – если в последнем проходе не было обслуживаемых каналов, то пауза длительностью 0.75 цикла потока CALC, иначе – без паузы;
- 2 – если в последнем проходе не было обслуживаемых каналов, то пауза длительностью 0.5 цикла потока CALC, иначе – без паузы;
- 3 – если в последнем проходе не было обслуживаемых каналов, то пауза до начала следующего цикла CALC, иначе – без паузы;
- 4 – если в последнем проходе не было обслуживаемых каналов, то пауза до начала следующего цикла CALC;
- 5 – если в последнем проходе не было обслуживаемых каналов, то пауза до начала следующего цикла CALC, иначе – ожидание pop-CALC (приостановка потока MAIN) или пауза 1с.

Увеличение указанных чисел на 16 отключает транзакции по данному COM-порту в режиме TRACE.

Использование системных переменных (см. **Группа СИСТЕМНЫЕ** и **Группа ДИАГНОСТИКА**):

- **@Modem_Step** типа INPUT (**Параметр** <> 0 указывает номер COM-порта):
 - 0, **R** – среднее время взаимодействия по указанному COM-порту (включая паузы) за весь цикл базы;
 - 95, **C2** – среднее время на одну успешную транзакцию (с учетом повторов запросов);
 - 92, **I2** – счетчик циклов каналов (по атрибуту 39, **EXEC**) с незавершенным обменом (суммарно для всех каналов);
- **@e_PLC2Type** типа INPUT (**Параметр** <> 0 указывает номер COM-порта):
 - 0, **R** – код ошибки:
 - 32** – фатальная ошибка (соответствующее сообщение записывается в файл tm6_log.txt);
 - 3** – ошибка записи;
 - 9** – в ответе 0 символов (таймаут);
 - 8** – в ответе недостаточно символов;
 - 7** – ответ полный, но с ошибкой;
 - 91, **I1** – счетчик повторов запросов (всех ошибок по всем каналам);
 - 92, **I2** – счетчик неудачных серий повторов запросов;
 - 45, **T** – время последней ошибки (если установлен флаг **Запрос времени значения**).

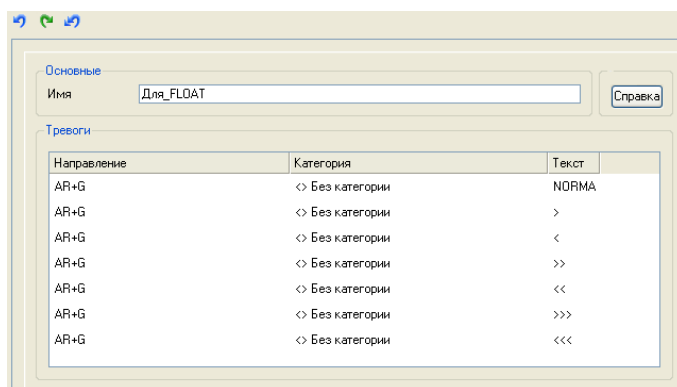
Запись любого значения в **@e_PLC2Type** типа OUTPUT обнуляет атрибуты переменной, а также атрибуты **@Modem_Step**.

Счетчики в **@Modem_Step** и **@e_PLC2Type** обнуляются при переходе на обмен по заданному COM-порту.

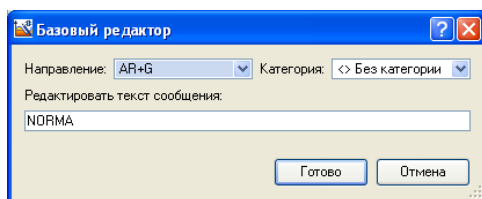
Редактор словарей сообщений

Редактор содержит ту же панель инструментов для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**).

Редактор любого словаря (см. **Классификация компонентов**) содержит поле **Имя**, в котором имя словаря может быть отредактировано, а также таблицу, каждая строка которой отображает собственно сообщение (поле **Текст**) и его параметры – категорию (поле **Категория**) и направление передачи (поле **Назначение**):



Параметры сообщения редактируются в диалоге, который появляется на экране при двойном нажатии ЛК на строке описания сообщения в таблице:



В текст генерируемого сообщения можно вставить значение канала – для этого в поле **Текст**, наряду с обычным текстом, нужно поместить описание формата вывода числа в нотации языка **Си**: (см. **Формат Си вывода чисел**).

Сообщение, которое начинается со знака **@**, не выводится в отчет тревог.

Сообщение может быть отнесено к одной из следующих категорий (см. **category.tmc**), задающих степень его ответственности (список **Категория**):

- < > Без категории;
- <M> Сообщение;
- <W> Предупреждение;
- <E> Ошибка;
- <I> Информация;
- <A> Тревога;
- <R> Изменение атрибутов;
- <S> Пользовательское;
- <_> Невидимое (не передается в графику);
- <-> Неквитируемое;
- <!> Командное;
- <?> Резерв.

Диалог содержит следующие опции (см. **destination.tmc**), задающие направление передачи сообщения (список **Назначение**):

- 0, **AR+G**
- 1, **AR+G+Prn**
- 2, **AR+Prn**
- 3, **AR**
- 4, **AR+G+GSM**
- 5, **AR+GSM**
- 6, **AR+GSM+PRN**
- 7, **AR+G+GSM+PRN**
- 9, **G**
- 10, **AR+Net**
- 11, **AR+G+Net**
- 12, **AR+GSM+Net**
- 13, **Net**
- 14, **Net+G**
- 15, **AR+Play**
- 16, **AR+G+Play**
- 17, **AR+Net+Play**
- 18, **Play**
- 19, **AR+G+PlayStop**
- 20, **AR+G+PlayLoopStop**

Эти опции содержат обозначения, соответствующие следующим направ-

лениям передачи:

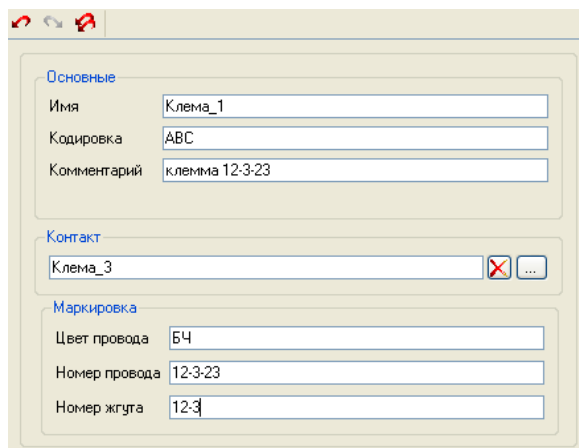
- **AR** – в файл отчета тревог;
- **G** – в исполнительные модули, способные отображать ОТ;
- **PRN** – на принтер, используемый по умолчанию;
- **GSM** – в виде SMS-сообщений на сотовые телефоны пользователей, которые определены в этом узле и для которых заданы телефоны и установлен флаг **Рассылка** (см. **Канал класса ПОЛЬЗОВАТЕЛЬ**). Заданное время хранения SMS-сообщений – 3 часа. Для задания параметров обмена по GSM может быть использована переменная **@RTM_Parameter** (см. **Группа СИСТЕМНЫЕ**);
- **Net** – в сеть;
- **Play, PlayStop, PlayLoopStop** – воспроизведение файла **<текст сообщения>.wav** (файл должен располагаться в папке узла). **Play, PlayStop** и **PlayLoopStop** задают такой же режим воспроизведения, как и системная переменная **@Sound_File**, для которой значение атрибута **Параметр** равно соответственно 0, 2 и 3 (см. **Группа СИСТЕМНЫЕ**).

Словари сообщений подробно рассматриваются в разделах **Системные сообщения**, **Сообщения по каналам**, **Генерация сообщений с помощью переменной MESSAGE** (см. также **Формат строки ОТ**).

Суммарное количество сообщений во всех словарях узла не должно превышать 32000.

Редактор клемм

Вид этого редактора показан на рисунке:



The screenshot shows a window titled "Редактор клемм" (Terminal Editor) with a toolbar at the top. The main area is divided into sections:

- Основные** (Basic):
 - Имя (Name): Клема_1
 - Кодировка (Code): ABC
 - Комментарий (Comment): клемма 12-3-23
- Контакт** (Contact):
 - Клема_3 (Terminal): Клема_3
- Маркировка** (Marking):
 - Цвет провода (Wire color): БЧ
 - Номер провода (Wire number): 12-3-23
 - Номер жгута (Bundle number): 12-3

Редактор содержит ту же панель инструментов для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**).

Редакторы источников (приемников)

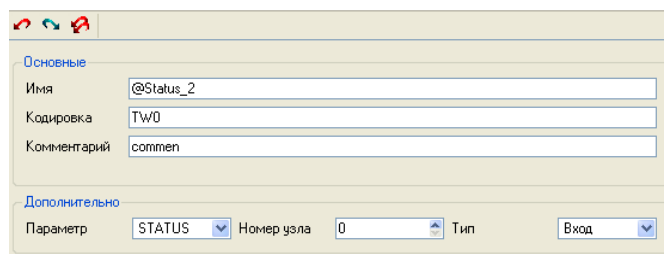
Редакторы источников/приемников отличаются по набору задаваемых в них параметров. Вместе с тем существуют следующие параметры, которые, как правило, задаются для любого источника/приемника в его редакторе:

- **Имя;**
- **Кодировка;**
- **Комментарий;**
- **Тип** или **Направление** – при автопостроении канала из источника/приемника тип созданного канала (INPUT/OUTPUT) будет соответствовать типу, заданному для источника/приемника;
- **Формат** – вид сигнала (аналоговый/цифровой);
- **Тип сигнала** – указание унифицированного сигнала (по списку **signal.tmc**, см. также **Атрибуты каналов, отображаемые профайлером**).

Редакторы источников/приемников содержат ту же панель инструментов для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**).

Редактор системных переменных TRACE MODE

Примерный вид редактора системной переменной TRACE MODE (группы СИСТЕМНЫЕ и ДИАГНОСТИКА) показан на рисунке (см. также **Редакторы источников (приемников)**):

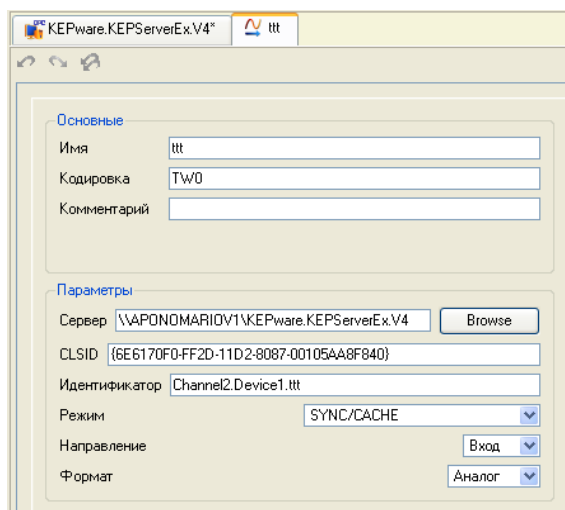


Характеристики системных переменных описаны в разделе **Системные переменные TRACE MODE 6**.

Редактор переменной OPC

Данная переменная используется в том случае, если монитор выступает в роли OPC-клиента – см. **Обмен по OPC**.

Вид редактора переменной OPC показан на рисунке (см. также **Редакто-**

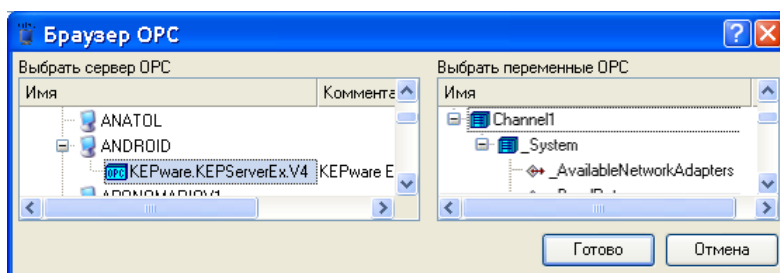
ры источников (приемников):

Список **Режим** включает стандартные обозначения режимов обмена с OPC-сервером:

- SYNC/CACHE – синхронное чтение из кэша;
- SYNC/DEVICE – синхронное чтение из прибора;
- ASYNC/DEVICE – асинхронное чтение из прибора;
- ADVISE – получение данных от сервера «по подписке».

При обмене с OPC-сервером TRACE MODE 6 SYNC/CACHE идентично SYNC/DEVICE.

При нажатии ЛК на кнопке **Обзор** на экране появляется диалог выбора OPC-сервера и его каналов для привязки:



Метод поиска OPC-серверов задается в настройках ИС (см. **Вкладка 'Базовый редактор'**).

При экспорте проекта в папке узла создаются конфигурационные файлы обмена с серверами OPC (см. **Файлы узла, создаваемые при экспорте**).

Редактор переменной OPC HDA

Данная переменная используется в том случае, если монитор выступает в роли OPC-клиента, запрашивающего исторические данные у сервера OPC HDA – см. **MPB как клиент сервера OPC HDA**.

Вид редактора переменной OPC HDA показан на рисунке (см. также **Редакторы источников (приемников)**):

The screenshot shows a dialog box titled "Редактор переменной OPC HDA". It is divided into two sections: "Основные" (Basic) and "Параметры" (Parameters). In the "Основные" section, there are three text input fields: "Имя" (Name) containing "HDA_OPC#1", "Кодировка" (Encoding) containing "TW0", and "Комментарий" (Comment). A "Справка" (Help) button is located to the right of these fields. In the "Параметры" section, there are four fields: "Сервер" (Server) with a "Browse" button, "CLSID", "Идентификатор" (Identifier), "Направление" (Direction) set to "Input", and "Формат" (Format) set to "Аналог" (Analog).

При нажатии ЛК на кнопке **Обзор** на экране появляется диалог выбора сервера OPC HDA и его переменных для привязки (вид браузера OPC показан в разделе **Редактор переменной OPC**).

Редактор переменной DDE

Данная переменная используется в том случае, если монитор выступает в роли DDE-клиента – см. **Обмен по DDE и NetDDE**.

Вид редактора переменной DDE показан на рисунке (см. также **Редакторы источников (приемников)**):

The screenshot shows a dialog box titled "Редактор переменной DDE". It is divided into two sections: "Основные" (Basic) and "Параметры" (Parameters). In the "Основные" section, there are three text input fields: "Имя" (Name) containing "DDE#1", "Кодировка" (Encoding) containing "TW0", and "Комментарий" (Comment). A "Справка" (Help) button is located to the right of these fields. In the "Параметры" section, there are four fields: "Режим" (Mode) set to "ADVISE", "А" (A) set to "0", "В" (B) set to "0", "Направление" (Direction) set to "Input", and "Формат" (Format) set to "Аналог" (Analog). At the bottom of the dialog, there is a "Дополнительно" (Additional) text input field.

Поле **Имя** используется для конфигурирования запроса.

Параметр **Режим** используется для задания режима обмена:

- **REQ/POKE** – если переменная имеет тип INPUT – режим REQUEST, если OUTPUT – режим POKE. Определены следующие модификации этого параметра:
 - **REQ/POKE[data]** – запрос не завершается специальными символами;
 - **REQ/POKE[data/r]** – запрос завершается символом Chr(10) (перенос строки);
 - **REQ/POKE[data/n]** – запрос завершается сочетанием символов Chr(10) и Chr(13) (перевод каретки);
- **ADVISE** – режим ADVISE.

Параметр **Тип** задает тип переменной (INPUT/OUTPUT).

Параметры **A** (атрибут 91, **I1**) и **B** (атрибут 92, **I2**) могут играть роль подстановок в запросе – для этого запрос должен содержать форматы их вставки в нотации Си (см. **Формат Си вывода чисел**). Например, два следующих запроса идентичны:

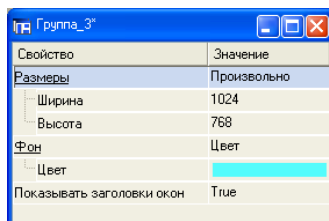
The image shows two side-by-side screenshots of a configuration window. Each window has a 'Основные' (Basic) section and a 'Параметры' (Parameters) section. The left window has 'Имя' (Name) 'Excel Sheet1.R6C7', 'Кодировка' (Encoding) 'TWO', and 'Комментарий' (Comment) empty. Its parameters are: 'Режим' (Mode) 'REQ/POKE[data]', 'A' '0', 'B' '0', 'Направление' (Direction) 'Output', and 'Формат' (Format) 'Аналог'. The right window has 'Имя' 'Excel Sheet1.R%dC%d', 'Кодировка' 'TWO', and 'Комментарий' empty. Its parameters are: 'Режим' 'REQ/POKE[data]', 'A' '6', 'B' '7', 'Направление' 'Output', and 'Формат' 'Аналог'.

Если в запросе используется одна подстановка, в нее выводится значение параметра **A**.

Редакторы групп компонентов

Редактор группы шаблонов экранов

Вид этого редактора показан на рисунке:



В этом редакторе для группы шаблонов экранов может быть задана часть параметров, которые определены для шаблона экрана (см. **Разработка графического интерфейса/Задание параметров графического экрана**).

Базовый редактор группы компонентов

Базовый редактор представляет собой унифицированный редактор для группы COM-портов, каналов, источников/приемников или клемм.

В этом табличном редакторе для компонента (выделенной группы компонентов) могут быть заданы те же параметры/атрибуты, что и в редакторе компонента (см. **Редакторы компонентов**). Кроме того, в редакторе могут быть заданы имена компонентов (имя компонента может быть отредактировано также в окне его свойств или непосредственно в навигаторе – см. **Окно свойств объекта структуры проекта** и **Меню и главная панель инструментов навигатора проекта**).

Ниже в качестве примера показан редактор группы компонентов **COM-порт**:

Базовое имя	Номер порта	Назначение п.	Скорость	Базовый адрес	Контроль четности	Прерывание	Управление	Тайм
COM-порт#1	COM1	Master	9600	1016	8-1-n	4	нет	
COM-порт#2	COM1	Master	110	0	8-1-n	0	нет	

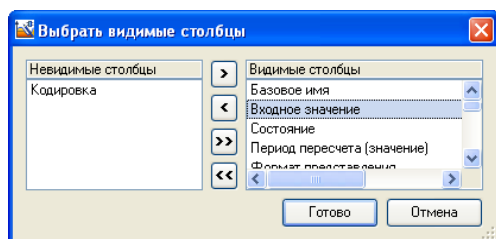
В редакторе поддерживаются стандартные операции изменения ширины столбцов, выделения нескольких компонентов, а при редактировании поля – стандартные операции работы с буфером обмена (см. **Типовые операции редактирования**).

Если компонент не имеет некоторого атрибута, задание этого атрибута для данного компонента в редакторе недоступно.

Управление столбцами

По умолчанию, все столбцы таблицы видимы и располагаются слева направо по возрастанию индексов атрибутов. Для изменения порядка расположения столбцов используется метод drag-and-drop с удержанием клавиши CTRL.

Для управления видимостью столбцов используется диалог, который открывается при нажатии ПК в любом поле заголовка таблицы:



Этот диалог содержит следующие инструменты:



– сделать видимым столбец, выделенный в левой части диалога;



– скрыть столбец, выделенный в правой части диалога;





– сделать видимыми все столбцы;



– скрыть все столбцы.

Фильтрация списка

Редактор снабжен панелью инструментов, которая, помимо стандартных инструментов отмены и возврата последнего действия (соответственно  и ) , содержит следующие:



– если этот переключатель находится в нажатом состоянии, в нижней части редактора открывается окно, в котором могут быть заданы условия отображения компонентов в редакторе;



– переключатель имен полей (в нажатом состоянии отображаются длинные имена, в отжатом – короткие).

Для изменения положения переключателя нужно нажать на нем ЛК.

Окно задания условий, в соответствии с которыми компоненты отобра-

жаются в редакторе, показано на рисунке:

Все типы			
Базовое имя	WC	*порт?	AND
Верхний предел	>=	100	AND

Самое верхнее поле этого окна предназначено для задания условия фильтрации списка по виду компонента (в случае канала – по классу). Вид/класс компонента выбирается в списке, который открывается при нажатии ЛК на кнопке поля. Этот список содержит в том числе опцию **Все типы** (задается по умолчанию), что соответствует отсутствию фильтрации (т.е. в редакторе отображаются компоненты всех видов). Условие фильтрации по виду/классу компонента удалить нельзя.

Для конфигурирования дополнительных условий фильтрации, а также для применения всех заданных условий используются следующие инструменты окна:

- применить все заданные условия к списку компонентов редактора;
- добавить дополнительное условие;
- удалить выделенное дополнительное условие.

Для задания строки дополнительного условия используются 4 поля (слева направо):

- атрибут (выбирается из списка);
- знак сравнения, выбирается из списка. Список, помимо обычных знаков сравнения (=, !=, <, >, <=, >=), содержит следующие:
 - **RE** (regular expression) – если выбрано это значение, в следующее поле можно ввести регулярное выражение Perl;
 - **WC** (wildcard) – если выбрано это значение, в следующем поле можно использовать стандартные знаки замены символов;
- значение (константа сравнения);
- логический оператор связи условий (**AND** или **OR**).

Столбцы атрибутов скрываются, если в результате фильтрации отображаются компоненты, не имеющие их.

Если заданные условия были применены к списку, для их отмены нужно закрыть окно задания условий (перевести переключатель в отжатое состояние).

Сортировка списка

В редакторе поддерживаются следующие виды сортировки списка компо-

нентов:

- по возрастанию порядковых номеров компонентов в группе (сортировка по умолчанию) – см. **Имена и идентификаторы объектов структуры**;
- по любому атрибуту по возрастанию;
- по любому атрибуту по убыванию.

Для переключения вида сортировки нужно нажать ЛК в заголовке соответствующего столбца (для сортировки по возрастанию порядковых номеров – в любом заголовке).

При переключении вида сортировки в используемом заголовке появляется значок, индицирующий направление сортировки:

- ▼ – сортировка по убыванию;
- ▲ – сортировка по возрастанию.

При сортировке по возрастанию порядковых номеров компонентов в группе значок не отображается.

Если компоненты имеют одно и тоже значение атрибута, по которому сортируется список, они будут отсортированы по атрибуту с младшим индексом, значение которого различно для данных компонентов.

Редактирование атрибутов компонента

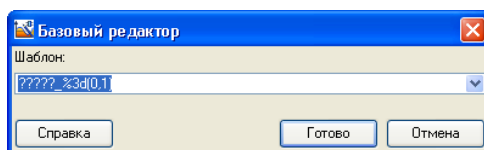
Для перехода к редактированию строкового атрибута компонента нужно дважды нажать ЛК в соответствующем поле. Другие атрибуты задаются выбором из списков. Список открывается при двойном нажатии ЛК в соответствующем поле редактора.

Групповое редактирование атрибутов

Чтобы задать какой-либо атрибут одновременно для всех выделенных компонентов, нужно, удерживая клавишу CTRL, дважды нажать ЛК в соответствующем поле и выбрать значение в открывшемся списке или ввести значение с помощью клавиатуры.

Групповое редактирование строковых атрибутов с инкрементированием

При выделении нескольких компонентов, их строковые атрибуты могут быть модифицированы (в том числе заданы содержащими инкрементированный числовой номер) в соответствии с шаблоном. Шаблон задается в диалоге, который появляется на экране при выполнении команды **Задать строку** из контекстного меню соответствующего поля:



Шаблон имеет следующий формат:

[string1] [%nF(start_num,increment)] [string2]

где:

- **%nF(start_num,increment)** – необязательное выражение, задающие инкрементирование и вид представления номера в редактируемом строковом атрибуте выделенных компонентов. В случае использования в шаблоне этого выражения все его составляющие являются обязательными:
 - **%** – стандартный знак формата в Си;
 - **n** – фиксированное количество знакомест, выделяемое для номера в строковом атрибуте. Пустые знакоместа заполняются нулями;
 - **F** – вид представления номера в строковом атрибуте:
 - d** или **D** – десятичное представление;
 - h** – шестнадцатеричное представление (с префиксом **0x**) строчными символами;
 - H** – шестнадцатеричное представление (с префиксом **0X**) прописными символами;
 - **start_num** – неотрицательный номер для строкового атрибута компонента с младшим порядковым номером;
 - **increment** – неотрицательный инкремент номеров в строковом атрибуте. Номера в строковом атрибуте инкрементируются в соответствии с возрастанием порядковых номеров выделенных компонентов;
- **string1** и **string2** – необязательные строковые выражения, в которых можно использовать знаки «?», «+» и «-».

При посимвольном сравнении исходной строки с шаблоном символ в исходной строке заменяется на символ из шаблона, если символ в шаблоне отличен от знаков «?», «+» и «-».

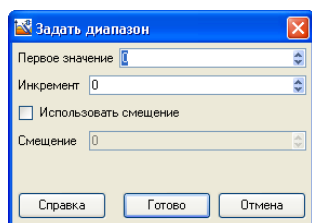
При использовании в шаблоне знаков «?», «+» и «-» символ в исходной строке модифицируется по следующим правилам:

- знак «?» запрещает модификацию соответствующего символа (при наличии этого символа);
- если знак «+» или «-» приходится на строчный символ, он не изменяется;
- если знак «+» или «-» приходится на цифру, она увеличивается

или уменьшается на 1 (при увеличении 9 заменяется на 0, при уменьшении 0 заменяется на 9).

Групповое редактирование числовых атрибутов с инкрементированием

При выделении нескольких компонентов, их числовые атрибуты могут быть модифицированы (в том числе заданы инкрементированными). Правила модификации задается в диалоге, который появляется на экране при выполнении команды **Задать диапазон** контекстного меню соответствующего поля:

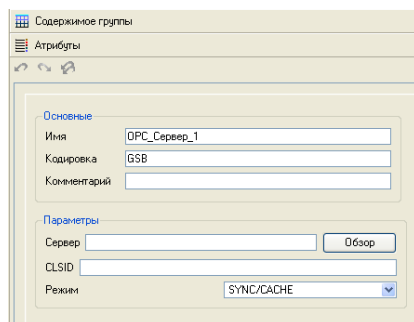


В диалоге задаются **первое значение**, **инкремент** и **смещение**. Результат действия зависит от флага **Использовать смещение**:

- если флаг не установлен, то атрибут компонента с младшим порядковым номером принимает первое значение, а остальные атрибуты инкрементируются относительно него с заданным шагом в соответствии с возрастанием порядковых номеров компонентов;
- если флаг установлен, то смещение прибавляется к значениям всех атрибутов (при этом первое значение и инкремент игнорируются).

Редактор группы 'ОПС-сервер'

Группа **ОПС Сервер** имеет следующий редактор (вкладка **Содержимое группы** аналогична редактору группы каналов (см. **Базовый редактор группы компонентов**):



При нажатии кнопки **Обзор** на экране появляется диалог выбора ОПС-

сервера (вид браузера OPC показан в разделе **Редактор переменной OPC**).

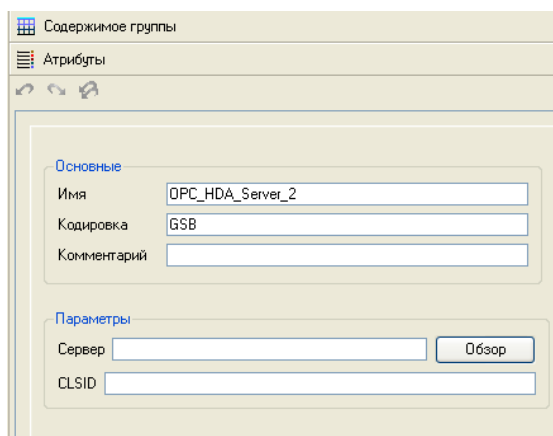
При выборе в этом диалоге переменных OPC-сервера в проекте автоматически создаются переменные OPC.

Метод поиска OPC-серверов задается в настройках ИС (см. **Вкладка 'Базовый редактор' и Обмен по OPC**).

Параметры переменных OPC задаются в редакторе (см. **Редактор переменной OPC**).

Редактор группы 'OPC HDA сервер'

Группа **OPC HDA Сервер** имеет следующий редактор (вкладка **Содержимое группы** аналогична редактору группы каналов (см. **Базовый редактор группы компонентов**):



Для поиска серверов OPC HDA и выбора их переменных нужно нажать кнопку **Обзор** (вид браузера OPC показан в разделе **Редактор переменной OPC**).

Параметры переменных OPC HDA задаются в редакторе (см. **Редактор переменной OPC HDA**).

Редактор группы каналов слоев 'Технология' и 'Топология'

Группы каналов слоев **Топология** и **Технология** имеют комбинированный редактор, включающий две вкладки – **Атрибуты** и **Содержимое группы**. Вкладка **Атрибуты** содержит ту же панель инструментов для работы с буфером обмена, что и редактор узла (см. **Задание параметров узла**):

Содержимое группы

Атрибуты

Основные

Имя:

Кодировка:

Комментарий:

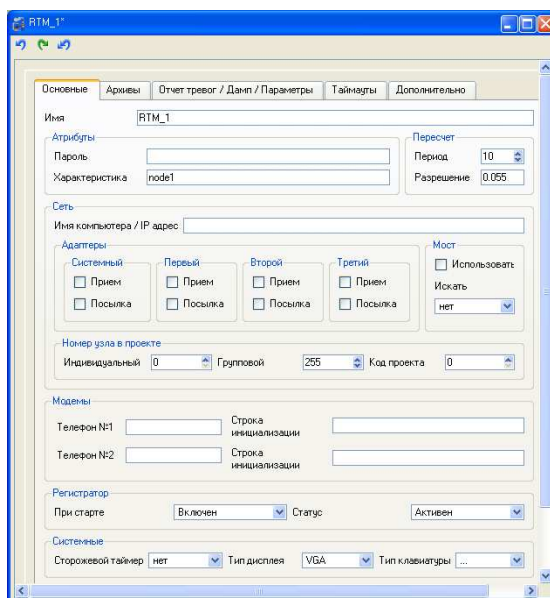
Кодировка:

Вкладка **Содержимое группы** аналогична редактору группы каналов (см. **Базовый редактор группы компонентов**).

Редактор параметров узла

Вкладки редактора узла




Редактор узла содержит вкладки и панель инструментов:





Задание параметров узла

Панель инструментов редактора узла



Эта панель, помимо стандартных инструментов отмены и возврата последнего действия (соответственно  и ) , содержит инструмент  возврата к значениям параметров, сохраненных в файле **prj**.

Действие инструментов  и  распространяется на весь редактор вне зависимости от того, какая вкладка является текущей в данный момент.

Вкладка 'Основные' редактора узла

На этой вкладке задаются следующие параметры узла:

- **Имя** – имя узла (см. **Имена и идентификаторы объектов**)

структуры);

- раздел «Атрибуты»:
 - **Пароль** – зарезервировано;
 - **Характеристика** – комментарий к узлу (строка);
- раздел «Пересчет»:
 - **Период** и **Разрешение** – произведение этих параметров определяет время цикла монитора (см. **Время цикла монитора**);
- раздел «Сеть»:
 - **Имя компьютера/IP-адрес** – сетевое имя или IP-адрес аппаратного средства, на котором будет запущен узел (см. **Формат IP-адреса**). В общем случае, указание этого параметра не является обязательным – монитор способен получить IP-адрес от ОС, поставить его в соответствие своему **индивидуальному номеру** в проекте и передать эту информацию по сети всем узлам. В случае задания, этот параметр записывается в файл **addr.ind** (см. **Файлы узла, создаваемые при экспорте**);
 - подраздел «Адаптеры» – конфигурирование сетевых адаптеров. Установка флага **Прием** разрешает прием через адаптер, установка флага **Посылка** – отправку автопосылок (с помощью автопосылок узел, в том числе, информирует другие узлы о своем присутствии в сети). Информация о возможности обмена по сети записывается в файл **addr.ind**.
Для реализации автоматического выбора адаптера при работе в WINDOWS XP нужно установить флаги адаптера **Системный**. В Windows CE адаптер следует указать явно.
Если сконфигурированы два адаптера, в реальном времени МРВ переключается на тот, через который производится прием. Не рекомендуется конфигурировать два адаптера, если они находятся в одной и той же подсети.
 - подраздел «Мост» – чтобы указать, что для широковещательных рассылок данный узел должен использовать мост, нужно установить флаг **Использовать** и выбрать в списке **Искать** одну из опций (см. **rtm_bridge.tmc**):
 - нет** – не искать мост;
 - первый** – использовать мост, заданный переменной **@IP_parameter** с атрибутом **Параметр=9** (см. **Группа СИСТЕМНЫЕ**);
 - второй** – использовать мост, заданный переменной **@IP_parameter** с атрибутом **Параметр=10**;
 - оба** – использовать оба моста, заданные переменными **@IP_parameter**.

Эти опции равнозначны заданию соответствующих битов переменной **@IP_parameter** с атрибутом **Параметр=3**.

- раздел «Номер узла в проекте»:
 - **Индивидуальный** – индивидуальный номер узла в проекте, при создании узлов инкрементируется автоматически. Этот параметр анализируется мониторами, что позволяет перенести узел на другое аппаратное средство (например, на компьютер с другим IP-адресом) без модификации проекта. Индивидуальный номер узла сохраняется в файле **addr.ind**. Индивидуальный номер узла можно задать с помощью файла *.cnf (см. **Задание параметров работы мониторов**), что позволяет создать в АСУ несколько однотипных рабочих мест, спроектировав только один узел в ИС;
 - **Групповой** – групповой номер узла в проекте (по умолчанию – 255). В TRACE MODE существует механизм запросов ко всем узлам, имеющим одинаковый групповой номер. Этот параметр сохраняется в файле **addr.ind**;
 - **Код проекта** – мониторы могут взаимодействовать по I-NET только в том случае, если узлы имеют один и тот же код проекта. При связи по M-LINK код проекта не анализируется. Данный параметр позволяет в одном проекте создавать подсистемы, не взаимодействующие между собой;
- раздел «Модемы» – задание для 1-2 модемов, подключенных к данному узлу, номеров телефонов (поля **Телефон1** и **Телефон2**, номер задается в международном формате) и строк инициализации (поля **Строка инициализации**). Номера телефонов сохраняются в файле **addr.ind**. Для задания номеров телефонов и строк инициализации могут использоваться следующие ключи в файле *.cnf (**xxx** – номер узла):

```

TM_H_PHONE_NODE_<xxx>=<телефон 1>
TM_S_PHONE_NODE_<xxx>=<телефон 2>
TM_3_PHONE_NODE_<xxx>=<телефон 3>
TM_4_PHONE_NODE_<xxx>=<телефон 4>

```

```

TM_PHONE_INITH=<строка инициализации модема 1>
TM_PHONE_INITS=<строка инициализации модема 2>
TM_PHONE_INIT3=<строка инициализации модема 3>
TM_PHONE_INIT4=<строка инициализации модема 4>

```

Для эмуляции работы модема в строку инициализации нужно добавить слово **EMULATE**, а в качестве номера телефона указать 8 произвольных символов. При этом для узла-партнера не обязательно конфигурировать модем в режиме эмуляции – достаточно сконфигурировать M-Link.

Для обмена SMS-сообщениями строка инициализации GSM-модема имеет следующий вид:

```
PIN:<PIN-код>;SCA:<телефон сервис-центра>[;NO_DSR]
```

Номер телефона задается в международном формате. Необязательный параметр **NO_DSR** отключает проверку DSR при обмене по GSM.

Для обмена между узлами по GPRS строка инициализации модема должна быть дополнена параметрами в соответствии с инструкциями провайдера. Например:

```
...;gprs_pdp:1,"ip","Internet.beeline.ru";gprs_atd:*99***1#;gprs_login:beeline;gprs_pass:beeline
```

При обмене по GPRS взаимодействие каналов должно быть **GSM SMS 1** или **GSM SMS 2** (см. **Связь канал-канал**).

Для эмуляции работы GPRS в строку инициализации модема надо добавить слово **EMULATE_CLI**, **EMULATE_SRV** или **EMULATE_BOTH** (для эмуляции соответственно клиента, сервера или клиента и сервера одновременно) При этом не должна быть задана сеть, но физически она должна функционировать. Кроме того, в качестве номера телефона нужно задать 8 произвольных символов.

При обмене по GPRS используются методы аутентификации **MS CHAP** и **PAP**.

Для обычного модемного обмена через GSM-модем (назначение COM-порта – МОДЕМ), строка инициализации задается в формате **gsm(<параметры>)** (набор параметров – в соответствии с документацией модема). Например, для модема Муха Technologies:

```
NO_DSR;gsm(+CR=0+CRC=0+ILRR=0+CBST=7,0,1+CPIN=<PIN-код>...)
```

- раздел «Регистратор»:
 - **При старте – Включен/Выключен.**

Для архивирования данных в регистратор используется только значение **При старте=Включен** (для узлов, передающих данные в регистратор; см. **Архивирование в регистратор**).

Значение **При старте=Выключен** используется для уменьшения потерь при обычном обмене по сети между узлами (см. **Механизм предотвращения потерь при обмене по сети**).
 - **Статус – Активен/Пассивен** (см. **Архивирование в регистратор**).
- раздел «Системные» (настройки этого раздела не влияют на работу

мониторов OEM-продуктов):

- **Сторожевой таймер** – тип аппаратного сторожевого таймера, выбирается из следующего списка (**rtms_watchdog.tmc**):

- 0 – no (запрет работы с таймером, в том числе для OEM);
- 1 – type1 (201);
- 2 – type2 (20C);
- 3 – type3 (443);
- 4 – type4;
- 5 – Wafer-4823;
- 6 – type6;
- 7 – type7;
- 8 – type8;
- 9 – type9;

- **Тип дисплея** – тип терминала, выбирается из следующего списка (**rtms_display.tmc**):

- 0 – VGA;
- 1 – Smart_Link (эмулятор терминала OCTAGON);
- 2 – VT100;
- 3 – DK8070 (4-строчный дисплей ИКОС);
- 4 – V04 (8-строчный, по командам совместим с DK8070);
- 8 – stdout (вывод в стандартный поток);
- 9 – Smart Link;
- 10 – VT100;
- 11 – DK8070;
- 12 – V04.

Если монитор должен инициализировать COM-порт обмена с терминалом Smart Link / VT100 / DK8070 / V04, следует выбрать позицию с номером меньше 8 (соответственно 1...4); если инициализация COM-порта не требуется, нужно выбрать позицию с номером больше 8 (соответственно 9...12);

- **Тип клавиатуры** – тип клавиатуры, выбирается из следующего списка (**rtms_keyboard.tmc**):

- 1 – Standard;
- 2 – int17h.

Вкладка ‘Архивы’ редактора узла

Эта вкладка имеет 4 раздела (по числу архивов – системный, СПАД1, СПАД2 и СПАД3), содержащих идентичные инструменты конфигурирования соответствующего архива:

- **Имя файла** – в общем случае – полный путь к файлу архива. Файл может иметь произвольное расширение или не иметь расширения. Если указано только имя файла, он создается в текущей директории ОС. Если имени предшествует знак «*», файл создается в папке узла. Имя файла должно состоять не менее чем из 2 символов. Для уменьшения времени операций с архивом файл кэшируется;
- **Состояние архива** – разрешение/запрещение использования архива (TRUE/FALSE). В реальном времени архивированием управляет переменная **@Logging** (группа СИСТЕМНЫЕ);
- **Размер файла, Мб** – предельно допустимый размер файла архива в мегабайтах. Минимальный размер файла (значение по умолчанию) – 128 МБ, для Windows CE – 32 МБ. При переполнении новые сообщения записываются с начала архива;
- **Максимум очереди записей** – максимально возможное число записей в очереди;
- подраздел «Кэш»:
 - **Период сохранения** – с этим периодом архивные сообщения сбрасываются из кэша в файл (в отсутствие переполнения кэша). Период сохранения задается в десятых долях секунды, в МРВ делится на 600 и округляется до целых минут (задаваемое значение не может быть меньше 1 мин., значение по умолчанию – 30с);
 - **Размер, Мб** – размер кэша в мегабайтах, по умолчанию – 48 МБ (шаг изменения размера кэша – 8 МБ, этот параметр не редактируется).

Вкладка ‘Отчет тревог/Дамп/Параметры’ редактора узла

На этой вкладке для узла конфигурируются отчет тревог (ОТ), дампы, параметры архивирования (SIAD), а также некоторые другие параметры.

Раздел конфигурирования ОТ содержит следующие параметры:

- **Имя файла** – в общем случае – полный путь к файлу ОТ. Файл может иметь произвольное расширение или не иметь расширения. Если указано только имя файла, он создается в текущей директории ОС. Если имени предшествует знак «*», файл создается в папке узла. Имя файла ОТ должно состоять не менее чем из 2 символов;
- **Максимум записей** – предельно допустимое число записей в ОТ. Значение по умолчанию (0) соответствует 1 млн. сообщений. При переполнении новые сообщения записываются с начала ОТ (со

второй строки). Размер файла ОТ можно изменять с помощью системной переменной **@AR_Length** (группа СИСТЕМНЫЕ);

- **Состояние** – разрешение/запрещение использования ОТ (TRUE/FALSE). Для управления разрешением использования ОТ в реальном времени используется переменная **@Logging** (группа СИСТЕМНЫЕ);
- **Формат даты** – форматы даты и времени в ОТ (см. **Формат Си вывода даты и времени**). Значение по умолчанию – **%x %H:%M:%S**. К значению даты и времени монитор через точку добавляет значение десятых долей секунды.

В Windows CE эта опция не работает. Формат даты и времени при работе в этой ОС устанавливается в соответствии с языком (русский – формат соответствует русскому по умолчанию, английский и китайский – формат соответствует английскому (США) по умолчанию).

- **Считывать при старте** – если TRUE, монитор при старте считывает ОТ (см. **Отчет тревог узла**, а также **Группа ГЭ 'Отчет тревог'**). Число считываемых строк ОТ задается на вкладке **Дополнительно**.

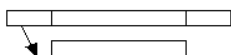
Раздел конфигурирования дампа узла содержит следующие параметры:

- **Имя файла** – в общем случае – полный путь к файлу дампа. Файл может иметь произвольное расширение или не иметь расширения. Если указано только имя файла, он создается в текущей директории ОС. Если имени предшествует знак «*», файл создается в папке узла. Имя файла должно состоять не менее чем из 2 символов;
- флаг **Считывать при старте** – если этот флаг установлен, чтение дампа при старте монитора разрешено;
- флаг **Сохранять** – если этот флаг установлен, сохранение данных в дампы разрешено;
- **Период сохранения состояния системы** – период перезаписи дампа, задается в секундах (1-255);
- **Период сохранения доп. информации** – период записи в файлы дополнительной информации, необходимой для корректного восстановления работоспособности узла, натуральное число (1 – однократная запись при штатной остановке узла; 2, 3 и т.д. – период сохранения в минутах, 2 соответствует периоду 1 мин, 3 – периоду 2 мин и т.д.). Например, с помощью данного параметра задается период записи в файл **<ID>.DRG** буфера тренда (см. **ГЭ 'Тренд'**).

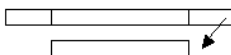
Раздел «Функции синхронизации» содержит флаги, определяющие маску синхронизации **Sync_mask**. В реальном времени значение этой маски индицирует/задает переменная **@RTM_parameter.Sync_mask** (см. **Группа СИСТЕМНЫЕ**). Функции синхронизации резервов выполняются

для каналов **FLOAT**, **DOUBLE FLOAT**, **HEX16**, **HEX32**, **Событие** и **CALL.57**, если в этих каналах установлен бит 0 атрибута 58, **DumpSync** (см. **Общие атрибуты каналов**, а также **Резервирование в АСУ**).

- **Синхронизация на старте** – если этот флаг установлен, на старте монитора выполняются следующие функции:
 - считываются стеки удаленных каналов **Событие**;
 - каналы **CALL.57** считывают удаленные аргументы;
 - числовые каналы типа **OUTPUT** считывают значения привязанных каналов-приемников и записывают в себя;
- **Архивы по началу** – если этот флаг установлен, резервный узел при старте однократно синхронизирует свои архивы с архивами узла **WORK** по начальным данным:



- **Архивы по концу** – если этот флаг установлен, резервный узел при старте однократно синхронизирует свои архивы с архивами узла **WORK** по конечным данным:



- **Разрешение tm6_log** – разрешение вывода в панель **MPB** сообщений файла **tm6_log.txt** (см. **Файл tm6_log.txt**). В файл **tm6_log.txt** записывают информацию все **MPB** (не только профайлеры).
- **Запрет непрерывной синхронизации глобальных переменных** – установка этого флага запрещает непрерывную синхронизацию глобальных переменных программ, но не запрещает такую синхронизацию при старте резерва.
- **Глобальные переменные** – зарезервировано.
- **Дополнительные атрибуты** – зарезервировано.
- **Принудительная отработка каналов OUTPUT** – если этот флаг установлен, при появлении в сети узла **A** выполняется принудительная отработка каналов **OUTPUT** с (58, **DumpSync**)=3, взаимодействующих по **CHCOPY** с каналами узла **A** (см. также описание переменной **@Status.NodeState** в разделе **Группа СИСТЕМНЫЕ**). Данная функция не работает, если **A** является резервом.
- **Время** – разрешение коррекции времени при старте.
- **Пересчет каналов INPUT с недоверностью** – этот флаг разрешает пересчет каналов **INPUT** с недоверностью.
- **Запись в архивы среза по всем каналам** – если этот флаг установлен, при старте узла в архивы записывается соответствующая информация по всем архивируемым каналам.

При первоначальном создании файла архива MPB всегда записывает в него информацию по всем архивируемым каналам. Поэтому, если флаг **Запись в архивы среза по всем каналам** установлен, при создании архива срез по каналам записывается в него дважды.

Раздел «Начальные значения режимов» содержит поля для задания начальных значений системных переменных **@Net_DDE**, **@Logging** и **@Input_Output**.

Раздел «Маски переключения режимов» содержит поля задания масок, используемых для вычисления значений соответствующих системных переменных в состояниях WORK и TRACE (см. **Группа СИСТЕМНЫЕ**):

- **Маска для @Net_DDE;**
- **Маска для @Logging;**
- **Маска для @Input_Output.**

В состоянии WORK значение переменной вычисляется по следующей формуле:

$$\mathbf{SYS} = (\mathbf{SYS} \wedge \mathbf{mask}) \ \& \ \mathbf{SYS}$$

В состоянии TRACE значение переменной вычисляется по следующей формуле:

$$\mathbf{SYS} = (\mathbf{SYS} \mid \mathbf{mask}) \ \& \ \mathbf{mask} = \mathbf{mask}$$

где **SYS** – значение переменной, **mask** – значение маски, \wedge – операция побитового XOR, $\&$ – операция побитового AND, \mid – операция побитового OR.

С помощью данных масок можно управлять автоматическим включением/отключением функций, соответствующих битам указанных системных переменных, при изменении состояния узла.

Вкладка ‘Таймауты’ редактора узла

На этой вкладке настраиваются параметры обмена с платами ввода/вывода, установленными в системную шину аппаратного средства, на котором запущен монитор. Большинство таких плат после получения запроса требуют временных задержек на оцифровку данных, на переключение внутренних или внешних коммутаторов, на обработку усиления и т.п.

На вкладке задаются следующие параметры:

- время ожидания ответа от УСО – значения этих таймаутов задаются в относительных единицах. Реальная задержка зависит от производительности процессора. Задержка обеспечивается вставленным в тело программы циклом, внутри которого присутствует еще один пустой цикл. В качестве параметра обоих циклов используется вве-

денное значение таймаута (таким образом, шкала таймаутов является квадратичной). Предельное значение таймаута равно 32000;

- **MUX** – номер канала на внешнем мультиплексоре;
- **GAIN** – параметр усиления;
- **Преобразование** – этот параметр задается в дополнение к таймауту ожидания ответа для плат, время оцифровки которых может варьироваться. Он определяет число запросов, которые генерируются монитором по истечении таймаута ожидания ответа к регистру платы, индицирующему завершение оцифровки. В Windows один такой запрос осуществляется за время порядка 10мкс, частота запросов устанавливается монитором. Поскольку все такие запросы находятся внутри цикла монитора, задавать их число избыточным нецелесообразно – при отказе оборудования существенно увеличивается время цикла монитора. По умолчанию значение параметра **Преобразование** равно 2000;
- **Терминал** – таймаут на отображение принятой строки 4- или 8-строчным терминалом с протоколом ADAM (задержка передачи следующей строки после подтверждения терминала о приеме предыдущей строки);
- **WaitStatus 32000** – таймаут для некоторых плат ICP DAS;

Таймауты для плат PCL 711/813:

- **PCL 711/813** – таймаут на коммутацию;
- **PCL GainControl** – таймаут на установку коэффициента усиления;
- **Преобразование PCL 711** – задержка начала проверки окончания оцифровки (таймаут на установку бита в соответствующем регистре платы).

Вкладка ‘Дополнительно’ редактора узла

На этой вкладке конфигурируются дополнительные параметры узла (см. **Группа СИСТЕМНЫЕ**):

- **Период посылки сообщения о присутствии в сети (с)** – аналог **@RTM_Parameter** типа OUTPUT с атрибутом **Параметр = 1, SENDTIME**;
- **Предельное время после последнего приема от удаленного узла (с)** – аналог **@RTM_Parameter** типа OUTPUT с атрибутом **Параметр = 2, Node_UnPresent**;
- **Период проверки состояния удаленных узлов (с)** – аналог **@RTM_Parameter** типа OUTPUT с атрибутом **Параметр = 3, Status_Check**;
- **Время на обнаружение резервов (мс)** – аналог **@RTM_Parameter** типа OUTPUT с атрибутом **Параметр = 4,**

Waite_Start;

- **Таймаут на переход в режим WORK** – аналог **@RTM_Parameter** типа OUTPUT с атрибутом **Параметр = 5, Waite_Chg_Sts;**
- **Маска рассылки регистраторам** – аналог **@IP_parameter** типа OUTPUT с атрибутом **Параметр = 7, Logger_Mask;**
- **Предельное время ожидания ответа на запрос (в циклах)** – если запрашиваемый узел присутствует в сети, а ответ на запрос в течение заданного времени не пришел, запрашивающему каналу устанавливается признак аппаратной недостоверности. Данный параметр задается в циклах монитора, значение по умолчанию – 4;
- **Число буферов для индивидуальных рассылок по сети IP** – аналог **@IP_parameter** типа OUTPUT с атрибутом **Параметр = 32, CountOfIndBlock;**
- **Число считываемых строк ОТ** – число строк ОТ, считываемых при старте / размер буфера чтения ОТ (значение по умолчанию (0) соответствует 256). Этот параметр задается кратным 64 (не более 16320).

В разделе «Приоритеты потоков» данной вкладки приоритеты потоков монитора, заданные по умолчанию, могут быть изменены (см. **Потоки монитора**).

Табличный редактор аргументов

Аргументы компонента проекта и их привязки к атрибутам/аргументам других компонентов задаются в табличном **редакторе аргументов**, в котором параметры каждого аргумента задаются в отдельной строке.

Аргументы могут быть созданы только для шаблона и канала класса **CALL** с ненастроенным свойством **ВЫЗОВ**, при этом редакторы аргументов этих компонентов имеют отличающиеся функции (и, соответственно, набор инструментов – см. **Разновидности редактора аргументов**).

Существует также третий вид редактора аргументов – для компонента (в том числе канала класса **CALL**) с настроенным свойством **ВЫЗОВ**. При задании свойства **ВЫЗОВ** таблица аргументов шаблона копируется в вызывающий компонент (в том числе и привязки аргументов, если они были заданы в редакторе аргументов шаблона), при этом в окне свойств вызывающего компонента появляется вкладка **Аргументы**. С помощью этой вкладки привязки аргументов (теперь уже аргументов компонента, вызывающего шаблон) могут быть изменены (компоненты могут вызывать один и тот же шаблон с передачей различных параметров). Такая перепривязка не отражается на таблице аргументов шаблона. Добавление/удаление аргументов в шаблоне воспроизводится во всех компонентах, вызывающих этот шаблон. Кроме того, задание привязки аргумента в шаблоне воспроизводится во всех компонентах, вызывающих шаблон, в которых данный аргумент не был привязан. Основные параметры аргументов – **имя**, **тип** и **тип данных** – не могут быть изменены в редакторе аргументов компонента с настроенным свойством **ВЫЗОВ**.

Информация об аргументах канала и их привязках отображаются в атрибуте 124, **ArgSize** этого канала (см. **Атрибуты каналов, отображаемые профайлером**).


Если компонент имеет аргументы, редактор аргументов доступен в окне свойств и редакторе этого компонента.

Поля редактора аргументов

Табличные редакторы аргументов (см. **Разновидности редактора аргументов**) имеют следующий набор полей:

- **Имя** – имя аргумента, задается по умолчанию и может быть изменено. Для перехода к редактированию имени аргумента нужно дважды нажать ЛК в данном поле. Имена аргументов одного и того же компонента должны быть уникальными;
- **Тип** – тип аргумента; выбирается в списке, который открывается при двойном нажатии ЛК в данном поле (см. также **Пользова-**

тельные функции Техно ST):

- **INPUT** – передача по значению (упрощенно – для приема);
- **OUTPUT** и **INPUT/OUTPUT** – передача по ссылке (упрощенно – для передачи);
- **Тип данных** – тип данных аргумента (см. **Определение переменных и констант** и **Специальная обработка аргументов HANDLE**). Тип данных выбирается в списке, который открывается при двойном нажатии ЛК в данном поле. Этот параметр должен учитываться при привязке аргументов;
- **[]** – объявление массива, используется только в функциях шаблонов программ (см. **Создание элементов программ с помощью табличных редакторов**);
- **Значение по умолчанию** – значение, указанное в этом поле, используется монитором в случае отсутствия привязки у аргумента. Для перехода к редактированию этого параметра нужно дважды нажать ЛК в данном поле;
- **Привязка** – привязка аргумента. При двойном нажатии ЛК в данном поле на экране появляется диалог выбора компонента/аргумента для привязки (см. **Привязка аргументов вручную**);
- **Принадлежность** – зарезервировано;
- **Флаги** – флаги, установленные для аргумента. В списке, который открывается при двойном нажатии ЛК в данном поле, для аргумента могут быть установлены следующие флаги: **HW, SL, NP, PO**. От флагов зависит результат автопостроения/привязки каналов (см. **Автопостроение и автопривязка каналов из редактора аргументов**);
- **Группа** – номер группы, к которой принадлежит аргумент. Для группирования нужно выделить несколько аргументов (см. **Типовые операции редактирования**) и нажать кнопку  **Группировать выделенные аргументы** – по этой команде выделенной группе аргументов автоматически присваивается номер. От параметра **Группа** зависит результат автопостроения каналов из редактора аргументов (см. **Автопостроение и автопривязка каналов из редактора аргументов**);
- **Единицы измерения** – единицы измерения (или унифицированный сигнал), выбираются в списке, который открывается при двойном нажатии ЛК в данном поле. От этого параметра зависит результат автопостроения/привязки каналов узла из редактора аргументов (см. **Автопостроение и автопривязка каналов из редактора аргументов** и **Флаг HW**).
- **Комментарий** – комментарий к аргументу. Для перехода к редактированию этого параметра нужно дважды нажать ЛК в данном поле.

- **Кодировка** – кодировка; выделяется красным, если задана некорректно. Присваивается каналу, созданному по команде **Создать каналы и привязать к аргументам**.

Основные параметры аргументов – **имя, тип и тип данных** – не могут быть изменены в редакторе аргументов компонента с настроенным свойством **ВЫЗОВ**.

Специальная обработка аргументов HANDLE

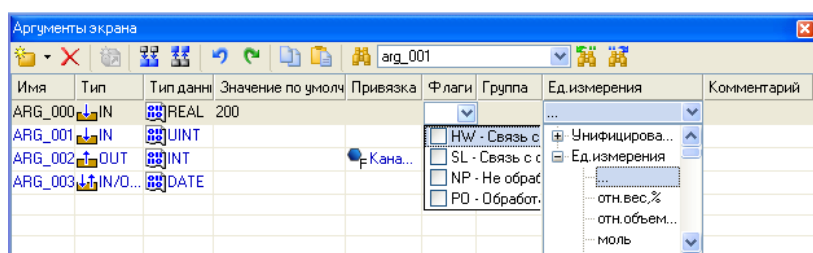
Аргументы с типом данных HANDLE (далее – **harg**) обрабатываются следующим образом:

- **harg**, связанные с атрибутом (89, **LO**) некоторого канала возвращают атрибут 89 как число или в тексте – имя канала, привязанного к нему;
- **harg**, связанные с аргументом/каналом, возвращают привязку аргумента;
- **harg**, связанные с аргументом того же канала вызова экрана/программы, возвращают привязку аргумента к привязке аргумента;
- **harg**, связанные с атрибутом (55, **relink**) некоторого канала **ch** возвращают:
 - если **ch** – канал вызова экрана, – то, что скопировано в экран;
 - если **ch** – CALL.ChGroupReq, – то, что он копирует в экран;
 - в других случаях – атрибут (86, **nAttr**) канала **ch**.

Учитывается только начальный аргумент.


Разновидности редактора аргументов

Редактор аргументов шаблона



Редактор аргументов шаблона снабжен типовыми инструментами создания, удаления, работы с буфером обмена и поиска (см. **Типовые ин-**

струменты редактирования), а также содержит специфические инструменты редактирования:

 **Группировать выделенные аргументы** – по этой команде выделенной группе аргументов автоматически присваивается номер (очередной порядковый). Этот номер отображается в поле **Группа** каждого аргумента группы;


 **Отменить принадлежность выделенных аргументов к группе.**

При задании имен аргументов шаблона нужно учитывать следующее:

- шаблоны экранов и связей с БД используют индексы аргументов – это означает, что при изменении имени аргумента, например, шаблона связи с БД в ИС автоматически изменятся: имя соответствующего аргумента компонента, вызывающего этот шаблон, и SQL-запрос, в котором использован аргумент;
- шаблоны отчетов и программ используют имена аргументов – это означает, что после изменения имени аргумента необходимо соответствующим образом отредактировать шаблон программы или документа.


Редактор аргументов канала CALL с ненастроенным свойством ВЫЗОВ

В отличие от редактора аргументов шаблона, в этом редакторе присутствуют следующие инструменты:

 **Создать каналы и привязать к аргументам** – по этой команде в той же группе каналов, к которой принадлежит и канал CALL, автоматически создаются каналы и привязываются к аргументам. Алгоритм этой процедуры зависит от параметров аргументов (см. **Автопостроение и автопривязка каналов из редактора аргументов**).

Редактор аргументов компонента с настроенным свойством ВЫЗОВ

В отличие от редактора аргумента канала класса CALL с ненастроенным свойством **ВЫЗОВ**, в этом редакторе отсутствует инструмент создания аргументов (а в редакторе аргументов источника/приемника – и инструменты автоматического создания каналов) и присутствует следующий переключатель:

 **Синхронизировать изменения привязок со всеми вызовами шаблона** – если эта опция включена (иконка имеет «утопленный» вид), то изменение привязок аргументов данного

компонента равнозначно изменению привязок всех компонентов, вызывающих тот же шаблон. Для включения/отключения данной опции надо нажать ЛК на переключателе.

Основные параметры аргументов – **имя, тип и тип данных** – не могут быть изменены в редакторе аргументов компонента с настроенным свойством **ВЫЗОВ**.

Редактирование параметров группы аргументов

Чтобы задать некоторый параметр одновременно для всех выделенных аргументов, нужно, удерживая клавишу CTRL, дважды нажать ЛК в соответствующем поле редактора и установить требуемое значение параметра.

Для выделения группы аргументов используются стандартные операции (см. **Типовые операции редактирования**).

Привязка аргументов вручную

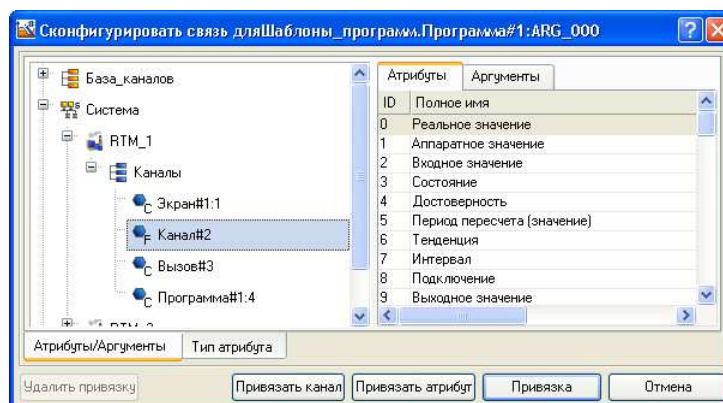
При привязке аргументов вручную (см. также **Связь через аргументы**) необходимо учитывать:

- направление передачи данных (из этого условия выбирается тип аргумента, см. **Поля редактора аргументов**);
- соответствие типа данных аргумента типу данных привязываемой переменной (аргумента или атрибута). Конвертер **FileCnv32.dll** записывает в свой протокол предупреждение о несоответствии типов данных (см. **Файлы узла, создаваемые при экспорте**);
- тип числового канала (при привязке его атрибутов (1, **A**) **Аппаратное значение** и (0, **R**) **Реальное значение** к аргументам канала, вызывающего шаблон программы – см. **Числовые каналы**).

Явная привязка

При двойном нажатии ЛК в поле **Связь** редактора аргументов (см. **Разновидности редактора аргументов**) на экране появляется диалог, в котором задается привязка аргумента.

Атрибут канала (или атрибут источника/приемника) для привязки выбирается на вкладке **Атрибуты** в правом окне диалога (в левом окне должна быть открыта вкладка **Атрибуты/Аргументы**):



К аргументу числового канала, в отличие от аргумента канала CALL, запрещается привязывать атрибут источника/приемника.

Для отработки связи «аргумент – источник/приемник» МРВ создает канал, имя которого формируется аналогично имени канала, который создается по аргументу с привязкой к удаленному каналу (см. **Связь через аргументы**).

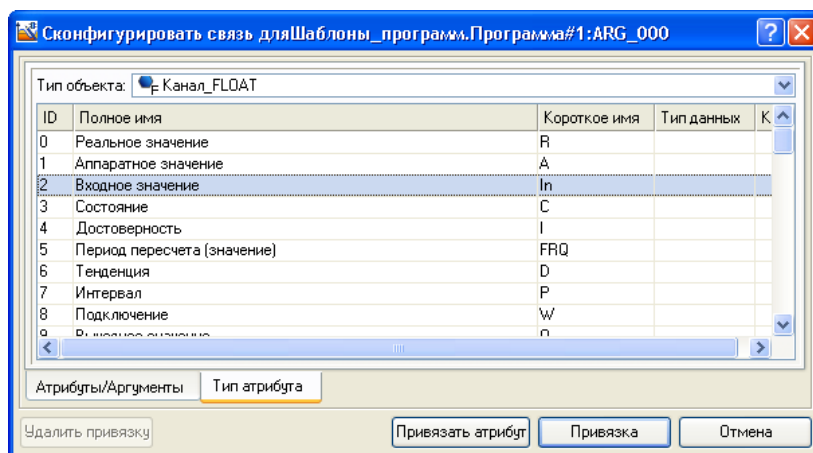
Аргумент для привязки выбирается на вкладке **Аргументы** в правом окне диалога (в левом окне должна быть открыта вкладка **Атрибуты/Аргументы**).

Если в левом окне открыта вкладка **Атрибуты/Аргументы**, диалог снабжен следующими инструментами:

- **Удалить связь** – удалить привязку;
- **Привязать канал** – изменить канал в привязке без изменения атрибута/аргумента (атрибут/аргумент, выделенный правым окном, не имеет значения);
- **Привязать атрибут/аргумент** – изменить атрибут/аргумент в привязке без изменения канала (канал, выделенный левым окном, не имеет значения);
- **Привязать** – привязать указанный атрибут указанного канала (в этом случае имеют значение как канал, выделенный левым окном, так и атрибут, выделенный правым окном);
- **Отмена** – закрыть диалог.

Виртуальная привязка

Вкладка **Тип атрибута** предназначена для привязки к аргументу прото-типа канала:



В верхнем списке для прототипа канала задается класс, в нижнем списке указывается атрибут.

Виртуальная привязка используется в дальнейшем при автоподстройке каналов из редактора аргументов или перепривязке (см. **Автоподстройка и автопривязка каналов из редактора аргументов**).

Автоподстройка и автопривязка каналов из редактора аргументов

В подразделах данного раздела описаны процедуры автоподстройки и автопривязки каналов из редактора аргументов.

Автоподстройка каналов по команде редактора аргументов

Редактор аргументов канала снабжен командой автоподстройки каналов и их одновременной привязкой к аргументам (см. **Разновидности редактора аргументов**).

Операции по аргументам без привязки

При выполнении команды автоподстройки с привязкой, класс канала и его атрибут для привязки определяются соответственно типом данных и типом аргумента (см. таблицы ниже). Автоматически создаваемые каналы имеют тип INPUT и имена, соответствующие именам аргументов.

Тип данных аргумента задается аналогично типу данных переменных программы (см. **Определение переменных и констант**).

В следующей таблице приведено соответствие класса канала, создаваемого при автопостроении, типу данных аргумента:

Тип данных аргумента	Класс канала
BOOL, SINT, USINT, INT, UINT	HEX16
DINT, UDINT	HEX32
TIME, DATE, DATE_AND_TIME	TIME
REAL	FLOAT
LREAL	DOUBLE FLOAT

Выбор атрибута для привязки зависит от типа аргумента следующим образом:

Тип аргумента	Атрибут
INPUT	Реальное значение (0, R)
OUTPUT, INPUT/OUTPUT	Входное значение (2, In)

При автопостроении канал создается для каждого аргумента, который не входит в группу (см. **Разновидности редактора аргументов**) и не имеет привязки.

Если аргументы сгруппированы, канал создается для каждой группы, которая содержит непривязанные аргументы. Непривязанные аргументы группы привязываются к созданному каналу, при этом атрибут, выбираемый для привязки, определяется типом аргумента. Класс канала, автоматически создаваемого для группы, определяется типом данных первого по списку непривязанного аргумента группы.

Операции по аргументам с виртуальной привязкой

Если аргументы имеют виртуальную привязку, при выполнении команды автопостроения с привязкой в группе создаются каналы по заданным в виртуальной привязке прототипам, и их атрибуты, заданные в виртуальной привязке, привязываются к соответствующим аргументам.

Флаг SL

Этот флаг устанавливается для аргумента шаблона программы, имеющего виртуальную привязку (см. **Привязка аргументов вручную**), и задает привязку аргумента к каналу числового класса или каналу T-Factory, вызывающему шаблон (с сохранением привязки к атрибуту, указанному в виртуальной привязке). Подмена в привязке прототипа канала на реальный канал выполняется монитором и не отображается в ИС.

Замена виртуальной привязки аргументов в канале CALL

Для замены всех виртуальных привязок аргументов канала CALL на привязку к некоторому каналу той же группы (например, к каналу **Канал#1**) нужно присвоить атрибуту **(47) iDstr** канала CALL имя или ID канала **Канал#1**.

Замена привязок аргументов канала CALL на привязки аргументов другого канала CALL

Для замены привязок аргументов канала CALL (пусть его имя – **call1**) на привязки аргументов другого канала CALL (пусть его имя – **call2**) нужно присвоить атрибуту **(55) relink** канала **call1** имя или ID канала **call2**.

Аргументы перепривязываются в соответствии с их порядковыми номерами. Привязки для аргументов должны быть заданы изначально.

Флаг HW

Флаг **HW** устанавливается при задании для аргумента единиц измерения (см. **Поля редактора аргументов**) и выборе контроллера (серии модулей), который предполагается использовать в АСУ (см.).

При автопостроении каналов из редактора аргументов (см. **Автопостроение каналов по команде редактора аргументов**), ИС на основе этих параметров подбирает плату для выбранного контроллера (или конкретный модуль из серии). Затем ИС создает источники/приемники выбранного оборудования и соответствующие каналы и далее – цепочки связей **аргумент – канал – источник/приемник**.

Флаги NP и PO

Флаг **NP** означает игнорирование аргумента при автопостроении каналов из редактора аргументов (см. **Автопостроение каналов по команде редактора аргументов**).


Флаг **PO** означает однократное использование аргумента при автопостроении каналов из редактора аргументов – с этой целью при первой операции автопостроения созданные привязки устанавливаются для аргументов вызываемого шаблона.

Автопостроение каналов по аргументам с установленным флагом **PO** зависит от флага **Обрабатывать флаг PO как глобальный** (см. **Настройка параметров редактора аргументов**). Если этот флаг не установлен, каналы создаются и привязываются к аргументам по соответствующим правилам. Если флаг установлен, вначале производится анализ каналов группы, и если в группе уже существует канал, который был бы

создан при автопостроении, то аргумент привязывается к нему, и новый канал не создается.

Автопостроение и автопривязка аргументов

Для автопостроения/автопривязки аргументов может быть использован метод **drag-and-drop**.

При перетаскивании компонента/аргумента (группы компонентов / аргументов) в редактор аргументов курсор принимает вид .

С помощью данного метода могут быть выполнены следующие операции:

- при перетаскивании компонента на выделенную группу аргументов – привязка всех выделенных аргументов к компоненту;
- при перетаскивании группы компонентов на выделенную группу аргументов – привязка каждого выделенного аргумента к компоненту, непосредственно входящему в группу, с соответствующим порядковым номером;
- при перетаскивании группы компонентов на свободное место в случае, когда создание аргументов разрешено и выделено не более одного аргумента, – автопостроение аргумента для каждого компонента, непосредственно входящего в группу, и привязка аргумента к этому компоненту;
- при перетаскивании группы аргументов в том случае, когда создание аргументов разрешено, – копирование аргументов (с привязками).

В операциях автопривязки атрибут канала для привязки выбирается автоматически по типу аргумента (см. **Автопостроение каналов по команде редактора аргументов**). При автопривязке источника/приемника во всех случаях выбирается его атрибут **Значение**.

При автопостроении по каналам созданные аргументы имеют тип INPUT и тип данных, зависящий от класса канала:

Класс канала	Тип данных аргумента
FLOAT, TIME, Событие, M-Ресурс	REAL
HEX16, Единица оборудования, Персонал, Пользователь	UINT
HEX32	UDINT
DOUBLE FLOAT	LREAL
CALL	DINT
D-Ресурс	USINT

При автопостроении по источникам/приемникам аргументы имеют тип INPUT и тип данных REAL.

Настройка параметров редактора аргументов

Параметры редактора аргументов настраиваются на вкладке **Редактор аргументов** диалога **Параметры** (см. **Задание общих настроек ИС**).

Если флаг **Переименовывать аргументы при привязке** установлен, при выполнении как ручной, так и автоматической привязки аргумент автоматически принимает имя следующего вида:

<имя компонента>_<короткое имя атрибута>

где

- **имя компонента** – имя привязываемого компонента, в котором служебные символы и пробелы заменены знаками подчеркивания;
- **короткое имя атрибута** – короткое имя (мнемоническое обозначение) привязываемого атрибута.

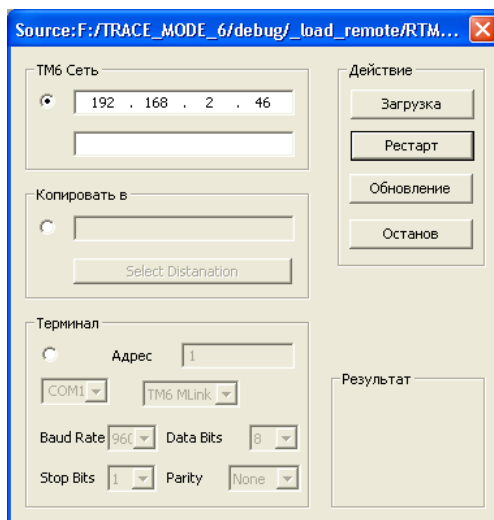
Флаг **Обрабатывать флаг РО как глобальный** влияет на автопостроение каналов по аргументам с установленным флагом **РО** (см. **Флаги NR и РО**).

Если флаг **Показать сетку** установлен, в редакторе отображается сетка.

Корректировка проекта в реальном времени

Перезагрузка узла из ИС

По команде  **Загрузить в контроллер** (см. **Меню и главная панель инструментов навигатора проекта**) на экране появляется следующий диалог:



С помощью переключателей **TM6 Сеть**, **Копировать в** и **Терминал** выбирается раздел конфигурирования интерфейса/механизма, который будет использован для перезагрузки узла.

Раздел **Действия** содержит команды управления перезагрузкой:

- **Загрузка** – копирование файлов *.rtp и *.dbb (см. **Файлы узла, создаваемые при экспорте**);
- **Рестарт** – перезагрузка узла через останов МРВ (требуется, например, в том случае, если в узле сокращено число каналов);
- **Обновление** – зарезервировано;
- **Останов** – останов монитора и выход в ОС.

Перед выполнением команд должен быть выполнен экспорт узлов.

При успешном выполнении команды в поле **Результат** выводится сообщение «ОК», в противном случае – сообщение об ошибке.

Перезагрузка узла по сети

Для перезагрузки узла по сети нужно в разделе **ТМ6 Сеть** задать IP-адрес или имя компьютера/контроллера, на котором запущен МРВ (IP-адрес подставляется автоматически, если он задан для узла при редактировании проекта в ИС).

Далее нужно выполнить команду **Загрузить** (для обновления бинарных файлов узла на удаленном компьютере/контроллере) и одну из команд перезагрузки.

*Экспорт файла *.dbb*

Для экспорта файла ***.dbb** нужно в разделе **Копировать в** указать полный путь к директории размещения и выполнить команду **Загрузить**. Путь может быть задан вручную или выбран с помощью диалога, который открывается при нажатии кнопки **Выбрать папку**.

Перезагрузка узла по RS

Для перезагрузки узла по RS нужно в разделе **Терминал** задать параметры используемого последовательного интерфейса (описание параметров приведено в разделе **Редактор параметров СОМ-порта**), а также указать протокол (ТМ6 MLink или XModem в случае использования модема).

Далее нужно выполнить команду **Загрузить** и одну из команд перезагрузки.

Регистратор аварийных событий

Для регистрации аварийных событий может быть использован канал CALL с типом вызова (57) **EvReg** (см. **Атрибуты канала класса CALL**).

Первые три аргумента канала имеют следующее назначение: **arg0** = **T_{START}**, **arg1** = **T_{END}**, **arg2** = **OFFSET**. Эти аргументы не могут иметь привязок. **OFFSET** (32 бита) задает интервал выборки:

$$T_FROM = T_{START} - \langle OFFSET.слово_0 \rangle [c]$$

$$T_TO = T_{END} + \langle OFFSET.слово_1 \rangle [c]$$

Если $\langle OFFSET.слово_0 \rangle = 0$, $T_FROM = T_{START} - 60c$.

Следующие **P** групп аргументов (**P** = **Параметр**) содержат по 3 аргумента: **arg0_i** – сигнал аварии; **arg1_i** – миллисекунды; **arg2_i** – порядковый номер, присвоенный аварии. Из этих аргументов только **arg0_i** могут иметь привязки.

Следующие аргументы создаются группами по 3 для отслеживания значений необходимых каналов (далее – сопутствующих каналов): **ARG0_k** – привязка канала, **ARG1_k** – значение канала до первой аварии, **ARG2_k** – значение канала после последней аварии. **ARG0_k** должны быть привязаны к каналам, **ARG1_k** и **ARG2_k** не могут иметь привязок.

По умолчанию регистратор деактивирован и **EvReg.R=0**. Команда активации регистратора – **EvReg.In=1** (команды 17-20 также активируют регистратор – см. ниже). Если регистратор активирован, при возникновении первой по времени аварии (**arg0_i=1**) **EvReg.R=2**, фиксируется время старта регистрации аварий (дата и время записываются в **arg0=T_{START}**, миллисекунды – в **arg1_i**) и аварии присваивается порядковый номер (в данном случае **arg2_i=1**). Последний присвоенный порядковый номер записывается в **EvReg.A**.

Активация и фиксация времени старта может быть выполнена вручную по команде **EvReg.In=2**.

После фиксации аварии значения **arg1_i** (число миллисекунд относительно **T_{START}**) и **arg2_i** не меняются при любых изменениях **arg0_i**.

Если число обнаруженных аварий равно **P**, регистрация аварий автоматически завершается, **EvReg.R=4**, а затем **EvReg.R=5**. Для принудительного завершения можно использовать команду **EvReg.In=3**. Время завершения фиксируется в **arg1 = T_{END}**.

Если для регистратора установлен флаг **Запрос времени значения**, времена фиксируются по атрибутам 45 и 88 каналов, привязанных к **arg0**. Если флаг не установлен или привязки нет – по собственному времени изменения.

Для деактивации регистратора нужно выполнить команду **EvReg.In=0**. По этой команде обнуляется **EvReg.A** и **arg2**.

Если регистратор находится в состоянии 5, то с помощью команд **EvReg.In = 6, 7, 8** или 9 его можно перевести соответственно в состояние 6, 7, 8 или 9. Из этих состояний регистратор возвращается в состояние 5 автоматически. Если регистратор активирован командами 17, 18, 19 или 20, то он автоматически переходит из состояния 5 соответственно в состояние 6, 7, 8 или 9. В состоянии 6 в папке узла создается текстовый файл ***.avt**, в состоянии 7 – бинарный файл ***.avb**, в состоянии 8 – файл ***.html** (в кодировке **utf-8**), в состоянии 9 – файл ***.xml**. Эти файлы содержат отчет регистратора. Имена файлов образуются из имени канала CALL и даты и времени создания. Для отображения информации из файла ***.avb** в профайлер с поддержкой графических экранов встроено специальное окно (см. **Профайлер с поддержкой графических экранов**). При генерации файла ***.avt EvReg.R** принимает значение 38, ***.avb** – 39, ***.html** – 40, ***.xml** – 41.

Информацию об ошибках регистратора МРВ записывает в файл **tm6_log.txt** (см. **Файл tm6_log.txt**).

С помощью конфигурационного файла узла можно задать произвольную директорию для сохранения файлов (см. **Задание параметров работы мониторов**).

Если сопутствующий канал архивируется, в отчет регистратора записывается выборка из архива по этому каналу.

Регистратор можно использовать только для выборки данных из архива – для этого его нужно принудительно активировать (**EvReg.In=2**) и затем принудительно остановить (**EvReg.In=3**).

Если для сопутствующего канала ведется индивидуальный архив, то запись в этот архив прекращается (в **arg1** архива пишется большое число), если регистратор переходит в состояние 5, и возобновляется по команде **EvReg.In=0**.

Если в канале CALL_1.57 типа OUTPUT задана привязка к другому каналу CALL_2.57 (локальному), то в состоянии 4 аргументы CALL_1 копируются в аргументы CALL_2, после чего каналы переходят в состояние 5. Канал CALL_2 конфигурируется так же, как CALL_1 (в том числе, аргу-

менты могут иметь привязки по указанным выше правилам).

Если в канале CALL_1.57 (INPUT, узел 1) задана привязка к каналу CALL_2.57. (узел 2), в сопутствующем канале **ch** (узел 1) канала CALL_1.57 задана привязка к сопутствующему каналу **rem_ch** (узел 2) канала CALL_2.57, и для канала **rem_ch** в узле 2 ведется индивидуальный архив, то при CALL_2.57.R > 3 по запросу узла 1 аргументы CALL_2.57 копируются в аргументы CALL_1.57 и, кроме того, запрашивается весь индивидуальный архив канала **rem_ch** и записывается в SIAD, в который архивируется **ch**. Если **rem_ch** привязан непосредственно к аргументу CALL_1.57, в выборку записывается ID канала вместо имени, а если привязка аргумента не задана, извлекаются только 2 значения (соответствующие **T_FROM** и **T_TO**).

Глава 3

**Каналы и
системные
переменные
TRACE MODE 6**

Пересчет базы каналов

В этом разделе описывается алгоритм пересчета базы каналов монитором TRACE MODE. Эту информацию следует учитывать при создании проекта в ИС для оптимизации работы будущей АСУ.

Потоки монитора

Монитор – многопоточный процесс. Приоритеты потоков заданы по умолчанию, однако их можно изменить на вкладке **Дополнительно** редактора узла (см. **Задание параметров узла**) или в файле *.cnf (см. **Задание параметров работы мониторов**).

Приоритет выбирается из следующего списка:

- -1 – Default;
- 0 – IDLE;
- 2 – LOWEST;
- 4 – BELOW_NORMAL;
- 6 – NORMAL;
- 10 – ABOVE_NORMAL;
- 12 – HIGHEST;
- 14 – TIME_CRITICAL;
- 77 – STOP.

Ниже указаны номера потоков и выполняемые в них задачи (см. также **Принцип работы монитора. Канал TRACE MODE 6**):

- 0 – CALC, основной поток, выполняемый монитором циклически. Один цикл включает следующие последовательно выполняемые этапы:
 - последовательный анализ всех включенных каналов узла (по возрастанию ID) и установка флага **SV** (недоступен для пользователя) каналам, требующим пересчета (см. **Период пересчета канала**).

Каналы с типом пересчета **цикл FAST** и **FAST EXE**, **цикл IDLE** и **IDLE EXE**, **цикл TF** анализируются соответственно в потоке FAST, IDLE и TF.

Пересчет всех каналов (кроме каналов CALL) типа INPUT, которые должны пересчитываться в основном потоке, и, в некоторых случаях, – отработка этих каналов:

если канал связан с УСО – установка каналу флага необходимо-

сти обработки **Q** (недоступен для пользователя), обработка (чтение данных) и пересчет (вычисление атрибутов канала – т.е. выполнение внутренних алгоритмов и трансляции). Сброс для канала флагов **SV** и **Q**;

если канал принимает данные асинхронно (сеть, RS) – установка флага **Q** и пересчет. Сброс флага **SV**, установленный флаг **Q** является признаком необходимости обработки канала в соответствующем потоке;

- пересчет и обработка каналов класса CALL основного потока;
- пересчет каналов типа OUTPUT, которые должны пересчитываться в основном потоке, и анализ их выходного значения. Установка флага **Q** каналам, выходное значение которых изменилось. Дальнейшие действия выполняются по следующему алгоритму:

если канал связан с УСО – обработка (передача значения) и сброс флага **Q**;

если канал передает данные асинхронно (сеть, RS), он не обрабатывается и флаг **Q** не сбрасывается.

Не сброшенный в основном потоке флаг **SV** является признаком необходимости пересчета канала в соответствующем потоке.

Если используются приоритеты потоков по умолчанию, и для цикла установлено время (см. **Время цикла монитора**), недостаточное для выполнения всех его задач, система будет работоспособной (будет идти обмен по RS и сети, выполняться программы и т.п.), однако заданные временные характеристики пересчета/обработки каналов будут нарушаться (т.е. быстродействие системы снизится). Например, в такой ситуации при запуске узла под Windows CE может никогда не отработаться вызов графической панели;

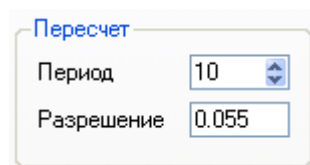
- 1 – IP_RECEIVE, прием по сети IP;
- 2 – IP_SEND, отсылка по сети IP;
- 3 – ACTION, вспомогательный (печать, вывод в файл и т.п.);
- 4 – SDDE, сервер DDE, NetDDE;
- 5 – TF, T-Factory;
- 6 – MODBUS;
- 7 – AR, отчет тревог;
- 8 – копирование отчета тревог;
- 9 – дамп;
- 10 – FAST (быстрые каналы);
- 11 – TCP_CS (поток подключения по TCP);
- 12 – t12;
- 13 – MASTER RS;

- 14 – MLINK;
- 15 – модем;
- 16 – вызов графики;
- 17 – SIAD;
- 18 – IDLE;
- 19 – клиент DDE, NetDDE;
- 20 – GSM, SMS;
- 21 – чтение архива;
- 25 – GPRS;
- 26 – MAKE_UP, короткоживущий поток, создается при необходимости на одну операцию (перепривязка на уровне объектов, генерация файлов **avb**);
- 27 – GPS;
- 28 – сервер MODBUS TCP;
- 29 – MAIN, мониторинг всех остальных потоков;
- 30 – графический поток ОС (рисование);
- 31 – TCP_RS, чтение по TCP.

Некорректная установка приоритетов потоков может привести к неработоспособности системы.

Время цикла монитора

Время цикла CALC (время, отводимое на однократное выполнение задач основного потока) настраивается с помощью двух параметров, которые задаются в разделе **Пересчет** вкладки **Основные** редактора узла. Параметр **Разрешение** задает разрешение таймера в секундах (величина **tick**), параметр **Период** – период пересчета в единицах **tick**:



Произведение этих параметров определяет время цикла CALC в секундах.

Разрешение таймера (**tick**) может варьироваться в следующих пределах:

- в MS Windows – не менее 0.01с;
- в MS Windows CE – не менее 0.001с;

По умолчанию разрешение таймера равно 0.055 с, период – 10.

Для мониторинга/задания циклов CALC, TF, IDLE и FAST используется

системная переменная **@Calc_Loop** (см. **Группа ДИАГНОСТИКА**). Циклы CALC, TF, IDLE и FAST могут быть также заданы в конфигурационном файле узла (см. **Файл CNF** в разделе **Задание параметров работы мониторов**).

Управляя временами циклов и приоритетами потоков, а также временными характеристиками пересчета отдельных каналов (см. **Период пересчета канала**), можно оптимизировать быстродействие системы.

Период пересчета канала

Значение периода пересчета канала может устанавливаться либо в единицах времени (при этом оно должно быть кратно времени цикла CALC), либо в циклах соответствующего потока (см. **Время цикла монитора**).

Если, например, период пересчета канала задан равным 2с, а время цикла CALC при этом равно 5с, то канал будет пересчитываться не чаще, чем 1 раз в 5с.

При редактировании проекта в ИС начальное значение и единицы измерения периода задаются на вкладке **Основные** раздела **Системные** редактора канала:

Период	Единица измерения
1	цикл CALC

В реальном времени период канала можно изменять, модифицируя значения атрибутов **Период пересчета (значение)** (5, FRQ) и **Период пересчета (единицы)** (38, FRQ_D).

При редактировании в ИС единицы измерения периода канала (или **тип пересчета**) выбираются из следующего списка (**recalculation.tmc**, в скобках слева приведено значение, которое в реальном времени нужно послать в атрибут **FRQ_D** для задания соответствующих единиц измерения, в скобках справа – допустимое значение периода для выбранных единиц измерения):

- (0) **цикл CALC** – период в циклах CALC;
- (1) **сек** – период в секундах (1-60);
- (2) **мин** – период в минутах (1-60);
- (3) **час** – период в часах (1-24);
- (4) **флаги** – период в циклах CALC с блокировкой пересчета с помощью канала типа OUTPUT, связанного с системной переменной **Системные/@Recalculation_Flag** (атрибут **Параметр** переменной должен быть равен 0). Для блокировки пересчета надо

послать в такой канал значение 1;

- (5) **F1** – период в циклах CALC с отставанием на 1 цикл;
- (6) **F2** – период в циклах CALC с отставанием на 2 цикла;
- (7) **F3** – период в циклах CALC с отставанием на 3 цикла;
- (8) **F4** – период в циклах CALC с отставанием на 4 цикла;
- (9) **цикл FAST** – период в циклах FAST (см. **Потоки монитора**);
- (10) **на старте** – канал пересчитывается один раз при старте монитора;
- (11) **в нач. часа** – один раз в сутки в начале часа, заданного атрибутом **Период**;
- (12) **в нач. дня** – один раз в месяц в начале суток, номер которых задается атрибутом **Период**;
- (13) **по времени** – в указанное время. Монитор создает в памяти список фиксированных значений времени (32), который при старте заполнен нулевыми значениями (0 соответствует периоду 1 цикл). Для заполнения списка необходимыми значениями времени (в секундах с 01.01.70) используются каналы класса **TIME**, связанные с системными переменными **Системные/@Recalculation_Flag**. Атрибут **Параметр** системной переменной задает индекс значения времени в списке (начиная с 1), значение канала **TIME** – значение времени:
 - если значение канала меньше 86400, оно прибавляется к значению времени в указанной позиции списка;
 - если значение канала больше или равно 86400, оно задает значение времени в указанной позиции списка.

Начальное время пересчета задается с помощью ключа **TM_BYFRQ<m>** в файле *.CNF (**m** – индекс значения времени в списке).

Для выбора времени пересчета канала с единицами измерения периода **По времени** используется его атрибут **Период пересчета (значение)** (5, **FRQ**) – значение этого атрибута указывает индекс времени в списке монитора;

- (14) **однократно** – только для каналов асинхронного обмена. После генерации запроса (**Request=1**) канал ждет ответа; если ответ получен, **Response=1** и канал выключается; если ответа нет (**Request=1, Response=0**) в течение заданного таймаута, канал не выключается, в нем устанавливается флаг аппаратной недостоверности, и в дальнейшем канал периодически генерирует запросы (до получения ответа);
- (15) **цикл IDLE** – период в циклах IDLE;
- (16) **цикл TF** – период в циклах TF;
- (17) **Trace Depth 0** и (18) **Trace Depth <>0** – по команде **Сохранить для MPB** из каналов с такими периодами пересчета экс-

портируются только те, значение периода которых не противоречит значению параметра **Глубина отслеживания источников** в общих параметрах ИС (см. **Задание общих настроек ИС**). Период экспортированных каналов устанавливается в циклах CALC;

- (19) **свой поток** – пересчет в собственном потоке (создается монитором, имеет приоритет **idle**; номер потока задается атрибутом **Период**). Этот тип пересчета задается для каналов CALL.Program, CALL.SQLQuery или CALL.Document(Report) (см. **Атрибуты канала класса CALL**), если эти каналы требуют значительного количества ресурсов. После выполнения канала бит 0 атрибута 46, **QE** (параметр **b14**) принимает значение 1;
- (20) **CALC EXE** – канал OUTPUT с таким типом пересчета анализируется 1 раз за N циклов CALC (N=**Период**), а обрабатывается при изменении входного значения в следующем цикле CALC вне зависимости от атрибута **Период**;
- (21) **FAST EXE** – канал OUTPUT с таким типом пересчета анализируется 1 раз за N циклов FAST (N=**Период**), а обрабатывается при изменении входного значения в следующем цикле FAST вне зависимости от атрибута **Период**;
- (22) **IDLE EXE** – канал OUTPUT с таким типом пересчета анализируется 1 раз за N циклов IDLE (N=**Период**), а обрабатывается при изменении входного значения в следующем цикле IDLE вне зависимости от атрибута **Период**;
Специфика типов 20-22. Если тип канала – OUTPUT, его значение передается приемнику даже в том случае, если приемник имеет такое же значение.
- (23) **день недели** – значение 4 младших битов атрибута **Период** задает день недели (1 – воскресенье, 2 – понедельник и т.д.), ключ **FRQSHF4** в файле *.cnf – время дня (как число минут с начала дня), значение старших 4 битов – тип отработки:
 - 0 – в указанный день в указанное время;
 - 1 – если день рабочий – 1 раз в неделю;
 - 2 – если день выходной – 1 раз в неделю;
 - 3 – каждый рабочий день недели;
 - 4 – каждый выходной день недели;
 - 5 – в заданный день с периодом пересчета;
 - 6 – в заданный день с периодом пересчета 1с;
 - 7 – в заданный день с периодом пересчета 1мин.;
- (24) ***день недели** – аналог предыдущего, но после недельного ожидания обрабатывается чаще до тех пор, пока не считывает данные;
- (25) ***мин** – период в минутах, ключ **FRQSHF0** в файле *.cnf задает сдвиг пересчета относительно начала минуты (см. **Задание параметров работы мониторов**);

- (26) ***час** – период в часах, ключ **FRQSHF1** в файле *.cnf задает сдвиг пересчета относительно начала часа;
- (27) ***день** – период в сутках, ключ **FRQSHF2** в файле *.cnf задает сдвиг пересчета относительно начала суток;
- (28) ***месяц** – период в месяцах, ключ **FRQSHF3** в файле *.cnf задает сдвиг пересчета относительно начала месяца;
- (29) ****мин** – период в минутах.

Специфика типов 25-29. Если ответа на запрос нет, то в канале, для которого задан временной тип пересчета, отличный от 25-29, флаг аппаратной недостоверности устанавливается сразу по истечении заданного таймаута, а последующие запросы генерируются с заданным периодом. В такой же ситуации в канале с типом пересчета 25-29 флаг аппаратной недостоверности не устанавливается до следующего пересчета, а последующие запросы генерируются чаще:

25 – 1 раз за **FRQ** циклов CALC;

26 – 1 раз в минуту;

27 – 1 раз в минуту;

28 – 1 раз в минуту;

29 – задается монитором (в самом быстром варианте – с периодом, заданным для узла, но не чаще, чем 1 раз в секунду).

- (30) **по событию** – канал обрабатывается при наступлении заданного события (для указания события нужно присвоить соответствующее значение атрибуту **Период пересчета (значение)**):
 - 1 – авторизация;
 - 2 – завершение сеанса;
 - 3 – авторизация и завершение сеанса;
 - 4 – спустя 1 мин. после авторизации;
 - 5 – спустя 2 мин. после авторизации;
 - 6 – спустя 3 мин. после авторизации;
 - 7 – спустя 4 мин. после авторизации;
 - 8 – спустя 5 мин. после авторизации;
 - 60 – спустя 1ч после авторизации;
 - 61 – спустя 2ч после авторизации;
 - 62 – спустя 3ч после авторизации;
 - 63 – спустя 4ч после авторизации;
 - 64 – спустя 5ч после авторизации.

При создании канала в ИС ему по умолчанию задается период 1 цикл CALC.

Произвольный поток для отработки может быть задан для

любого канала, кроме CALL.Screen – такой канал всегда обрабатывается в соответствующем потоке вне зависимости от заданного.

Для каналов, период которых задан в циклах CALC (кроме **F1**, ..., **F4**), необходимость пересчета определяется по равенству нулю остатка от деления значения системной переменной **@Recalculation_Index** (индекс пересчета, см. **Группа СИСТЕМНЫЕ**) на период канала.

Для каналов, имеющих тип периода **F1**, ..., **F4**, условие пересчета можно записать следующим образом:

$$(<\text{Индекс пересчета}>\%5 = I) \ \& \ (<\text{Индекс пересчета}>\%FRQ = 0)$$

где

I – задержка в циклах (1, 2, 3 или 4);

FRQ – период.

Исходя из этого условия, значение периода канала **1** и размерность **F1** устанавливают его пересчет один раз в 5 циклов с отставанием на 1 цикл от канала с периодом, равным 5 и размерностью **цикл CALC**. Каналы с периодом **F2** в этих же условиях будут отставать на 2 цикла, **F3** – на 3 цикла, а **F4** – на 4 цикла. Значение периода 2 задает для этих каналов пересчет 1 раз в 10 циклов и т.д.

Быстрые каналы

На быстром RS может работать до 4-х каналов одновременно. Перевод канала в FAST RS задается с помощью ключа **CH_IN_FAST_RS=<номер канала>** в файле *.cnf.

Особенности пересчета и отработки каналов

Канал типа INPUT

При аппаратной недостоверности канала (см. **Атрибуты каналов, отображаемые профайлером** и **Общие атрибуты каналов**) не выполняются его внутренние алгоритмы и трансляция (см. **Числовые каналы**), и атрибуты (0,**R**) и (1,**A**) становятся независимыми переменными. Если канал связан с локальным источником (аргументом, каналом), то значение привязки записывается в атрибут (2,**In**). Если канал связан с удаленным каналом, то атрибут (2,**In**) становится независимой переменной. Атрибуты 0-2 сохраняют значения, сформированные до установки аппаратной недостоверности, и эти значения могут быть изменены.

При (8,**W**) = 1 не выполняются внутренние алгоритмы канала и трансля-

ция, **R=Q**, а атрибуты (1, **A**) и (2, **In**) становятся независимыми переменными. Атрибуты 1 и 2 сохраняют значения, сформированные до установки флага, и эти значения могут быть изменены.

Если канал **ch INPUT** связан с удаленным каналом **rem_ch** с автопосылкой, то **ch.In = rem_ch.R** даже при **ch.W=1**.

Канал типа OUTPUT

Выходное значение (9, **Q**) канала передается во вход (2, **In**) привязанного удаленного канала в следующих случаях:

- **Q** изменилось;
- канал обрабатывается принудительно (39, **EXEC=1**, после такой обработки **EXEC** автоматически обнуляется);
- после сброса признака аппаратной недостоверности канала. Аппаратная недостоверность автоматически устанавливается при отсутствии удаленного узла в следующих случаях:
 - после загрузки МРВ и запуска узла;
 - при изменении **Q** (аппаратная недостоверность канала не отключает его внутренние алгоритмы и трансляцию).

При восстановлении связи с удаленным узлом признак аппаратной недостоверности канала сбрасывается автоматически, если состояние удаленного узла – WORK.

При CHFAULT / All Redundant (см. **Связь канал-канал**) признак аппаратной недостоверности не устанавливается.

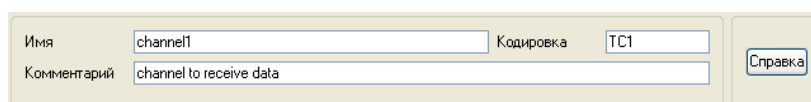
- если (8, **W**) = 0 или этот флаг сбрасывается в первый раз после загрузки МРВ и запуска узла;
- после сброса (8, **W**), если во время (8, **W**) = 1 изменилось **Q** (отключение канала от приемника не отключает его внутренние алгоритмы и трансляцию).

В отсутствие обработки канала вход (2, **In**) привязанного канала является независимой переменной, т.е. его значение может произвольно изменяться в удаленном узле.

Общие атрибуты каналов

В данном разделе описаны типовые инструменты редактора канала, с помощью которых задаются только те атрибуты, которыми обладают каналы всех классов. MPV отображает все необходимые атрибуты каналов – см. **Атрибуты каналов, отображаемые профайлером**.

Верхний раздел редактора канала

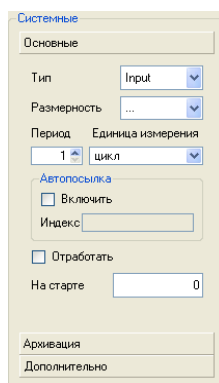


В верхней части редактора канала любого класса содержатся поля для редактирования следующих атрибутов:

- **Имя** – (127, **V_NAME**, MPV – **NAME**) – имя канала (при создании канала задается по умолчанию и может быть изменено в этом поле) (см. **Имена и идентификаторы объектов структуры**);
- **Кодировка** – (79, **CODE**) – кодировка канала (при создании канала задается по умолчанию и может быть изменена в этом поле). Кодировка может содержать буквы, цифры и символы «_», «-», «*», «?» и «@». MPV считывает первые 31 символ кодировки;
- **Комментарий** – (80, **CMNT**, MPV – **COMMNT**) – произвольный текст, MPV считывает первые 39 символов комментария.

Эти атрибуты могут быть также отредактированы в окне свойств (см. **Вкладка 'Информация'**).

Вкладка 'Основные' редактора канала



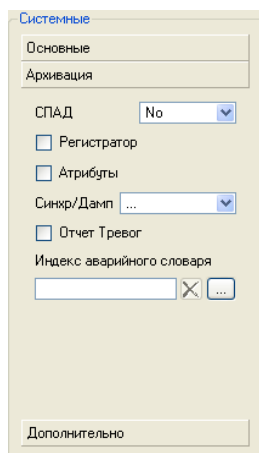
На этой вкладке редактируются следующие атрибуты канала:

- **Тип** – (81, **TYPE**, MPB – **IO**, в окне свойств – флаг **Тип OUTPUT**) – тип канала: INPUT (0 в реальном времени) или OUTPUT (1). Числовые каналы типа INPUT предназначены для приема данных от источников, типа OUTPUT – для передачи данных приемникам. У канала класса FLOAT с обработкой алгоритм обработки зависит от типа канала. Монитор автоматически устанавливает для канала тип, соответствующий привязанному источнику/приемнику;
- **Размерность** – (82, **DIM**) – размерность реального значения канала. Этот параметр выбирается из списка, который хранится в текстовом файле **tmcf/dimension.tmc**. Если требуемая размерность в списке отсутствует, то ее можно добавить, отредактировав указанный файл в редакторе с поддержкой кодировки UTF-8 – например, в MS WORD (редактировать файлы *.tmc в редакторе **Блокнот** ОС WINDOWS не рекомендуется). Текст размерности не может содержать более 8 символов, а количество строк в списке не должно превышать 255;
- **Период** – (5, **FRQ**) – значение периода пересчета канала (см. **Период пересчета канала**);
- **Единица измерения** – (38, **FRQ_D**) – единицы измерения периода пересчета канала, выбирается из списка (см. **Период пересчета канала**);
- **Включить** – (60, **T_NET**, MPB – **ToNet**, в окне свойств – флаг **Разрешить**) – при установке этого флага монитор будет передавать в сеть реальное значение канала при каждом его изменении в виде широковещательного сообщения (см. **Связь канал-канал**). На других узлах такое сообщение принимается каналами, которые связаны с данным;
- **Индекс** – (61, **N_SND**, MPB – 60, **ToNet**, в окне свойств – флаг **ID автопосылки**) – индекс, по которому другие узлы идентифицируют широковещательное сообщение. Использование этого параметра позволяет перенести узел на другое аппаратное средство (например, на компьютер с другим IP-адресом) без модификации проекта;
- **Отработать** – (39, **EXEC**) – установка этого флага является признаком необходимости отработки канала. Установка флага **Отработать** в редакторе задает пересчет и отработку канала при старте монитора;
- **На старте** – (2, **In**) – значение, указанное в этом поле, присваивается атрибуту (2, **In**) канала при старте монитора.

Тип и параметры автопосылки в сеть могут быть заданы также в окне свойств канала (см. **Вкладка 'Флаги'**).

Вкладка 'Архивация' редактора канала


На этой вкладке редактируются следующие атрибуты канала:

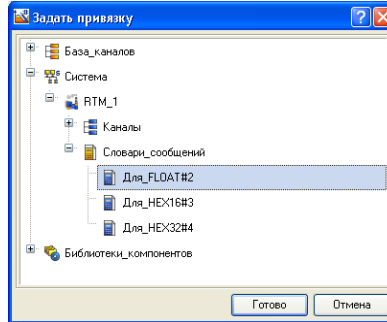



- **СПАД** – (41, **AB**) (в окне свойств – **Номер SIAD**) – признак архивирования атрибутов канала в SIAD, выбирается из следующего списка: **нет** – не архивировать, **1** – архивировать в SIAD1, **2** – в SIAD2, **3** – в SIAD3 (в реальном времени в атрибут нужно послать соответственно 0, 1, 2 и 3). Набор архивируемых атрибутов зависит от флага **Атрибуты**;
- **Регистратор** – (42, **RB**) – признак архивирования атрибутов канала в регистратор. Набор архивируемых атрибутов зависит от флага **Атрибуты**;
- **Атрибуты** – (44, **MB**) – от этого флага зависит набор атрибутов канала, архивируемых в указанный архив SIAD и/или регистратор. Этот флаг влияет также на генерацию сообщений для отчета тревог;
- **Синхр/Дамп** – (58, **DW**, **MPB** – **DumpSync**) – значение этого атрибута выбирается из следующего списка (**dump.tmc**):
 - ... – **DumpSync=0**; канал не использует дамп и игнорируется при синхронизации резервов (см. описание функций синхронизации в разделе **Задание параметров узла**, а также раздел **Резервирование в TRACE MODE 6**);
 - **Синхр.** – **DumpSync=1** (установлен бит 0 **DumpSync**); канал не использует дамп, но участвует в синхронизации резервов;
 - **Дамп** – **DumpSync=2** (установлен бит 1 **DumpSync**); канал использует дамп (чтение при старте/запись), но игнорируется при синхронизации резервов;
 - **Синхр./Дамп** – **DumpSync=3** (установлены биты 0 и 1 **DumpSync**); канал использует дамп и участвует в синхронизации резервов.

В окне свойств канала (см. **Вкладка 'Флаги'**) флаг **Синхронизировать** устанавливает бит 0, а флаг **Дамп** – бит 1 атрибута

DumpSync;

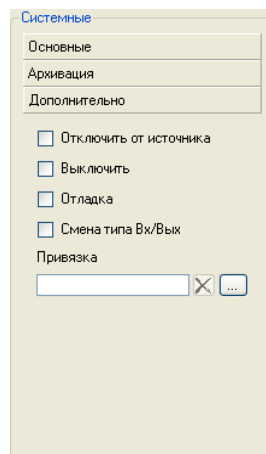
- **Отчет тревог** – (43, **EB**) – признак генерации сообщений для отчета тревог. Критерии генерации сообщений зависят от флага **Атрибуты**;
- **Индекс аварийного словаря** – (57, **ALR_MSG_I**, MPB – **Iala**) – словарь, чьи сообщения будут использованы для событий данного канала (если словарь не задан, генерируются сообщения, заданные по умолчанию). Словарь выбирается в следующем диалоге (для вызова диалога нужно нажать кнопку 




Для удаления привязки канала к словарю нужно нажать кнопку .

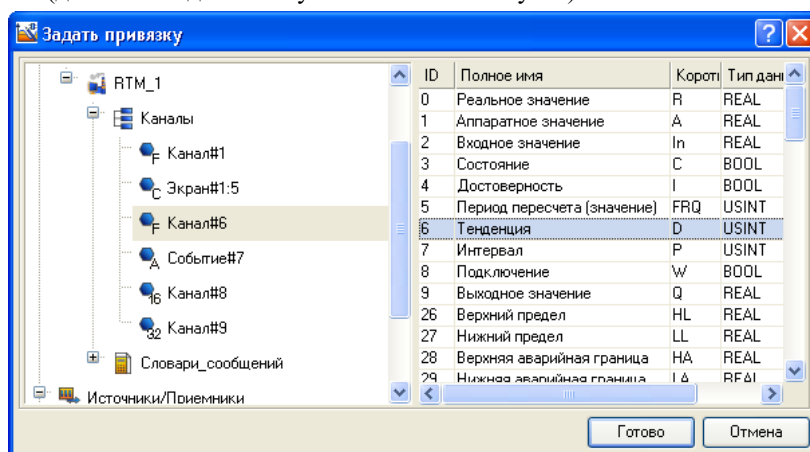
Все перечисленные атрибуты, за исключением индекса словаря, могут быть заданы в окне свойств канала (см. **Вкладка 'Флаги'**).


В реальном времени атрибуты 41, 42, 43, 44 и 57 допускается изменять только в том случае, если соответствующие параметры/режимы были заданы при редактировании проекта.

Вкладка 'Дополнительно' редактора канала

На этой вкладке редактируются следующие атрибуты канала:

- **Отключить от источника** – (8, **W**) **Подключение** – при установке этого флага канал отключается от источника/приемника (см. также **Особенности пересчета и отработки каналов**);
- **Выключить** – (3, **C**) **Состояние** – при установке этого флага пересчет канала не выполняется;
- **Отладка** – (49, **DBG**) **Отладка** – если этот флаг установлен, в отладочный файл профайлера <имя файла prj>_<ordinal>.txt (**ordinal** – порядковый номер узла) будет выводиться информация, определяемая каналом, связанным с системной переменной **@Debug** (группа СИСТЕМНЫЕ);
- **Смена типа Вх/Вых** – (51, **blO**) **Change IO** – смена типа канала на одну отработку;
- **Привязка** (86, **LN_ATTR**, **MPB** – **nAttr**) – задание свойства **связь**. Компонент для привязки выбирается в следующем диалоге (для вызова диалога нужно нажать кнопку ):



Для удаления привязки канала нужно нажать кнопку .

Атрибуты каналов, отображаемые профайлером

В данном разделе описаны общие атрибуты каналов, отображаемые в MPB. В том числе, MPB отображает те общие атрибуты, которые могут быть заданы в редакторе или окне свойств любого канала (см. **Общие атрибуты каналов**). В канале **СОБЫТИЕ** и каналах **T-Factory** общие атрибуты могут иметь другие имена и использоваться специфическим образом.

- (0, **R**) – **Реальное значение**;
- (1, **A**) – **Аппаратное значение** – этот атрибут используется в

числовых каналах (кроме **TIME**) и канале **CALL** при определенных условиях;

- (2, **In**) – **Входное значение**;
- (3, **C**) **Состояние** – установка этого флага означает остановку пересчета канала. Профайлер отображает значение этого атрибута совместно с типом канала и аппаратной достоверностью в виде <тип (I/O)> <состояние (On/Off)>+<достоверность (T/F)> – например, **O On+F** для пересчитываемого канала типа OUTPUT с недостоверным значением;
- (4, **I**) – **Достоверность** – бит 0 этого атрибута индицирует **аппаратную**, а бит 1 – **программную** достоверность значения канала (0 – достоверно. 1 – недостоверно; оба бита допускают запись/чтение, запись в бит 1 возможна только при установленном флаге 85, **BNDR** – см. **Канал класса FLOAT**). Числовым значениям **I** соответствуют следующие строковые:
 - 0 – **_T**
 - 1 – **_F**
 - 2 – **!T**
- (5, **FRQ**) – значение периода пересчета канала;
- (6, **D**) – **Тенденция** – этот атрибут индицирует результат сравнения реальных значений канала на текущем и предыдущем тактах пересчета:
 - в каналах классов **FLOAT** и **DOUBLE FLOAT**:
 - 0 – значение не изменилось;
 - 1 – значение уменьшилось;
 - 2 – значение увеличилось.
 - в каналах классов **HEX16** и **HEX32** каждый бит данного атрибута (соответственно 16- и 32-разрядного) индицирует изменение соответствующего бита реального значения. Если по сравнению с предыдущим тактом пересчета значение бита реального значения канала изменилось, то соответствующий бит атрибута **Тенденция** принимает значение 1, в противном случае – 0;
- (7, **P**) – этот атрибут имеет различный смысл и наименование в каналах различных классов.

В каналах **FLOAT** и **DOUBLE FLOAT** с заданными границами атрибут имеет имя **Интервал** и индицирует номер интервала, в котором находится реальное значение канала.

В канале **Событие** атрибут индицирует возникновение (1) и исчезновение (0) события;
- (8, **W**) **Подключение** – при установке этого флага канал отключается от источника/приемника;

- (9, **Q**) – **Выходное значение**. Данный атрибут не используется в каналах **TIME**, **CALL**;
- (38, **FRQ_D**) – единицы измерения периода пересчета канала;
- (39, **EXEC**) – флаг принудительной отработки канала;
- (41, **AB**) – номер SIAD;
- (42, **RB**) – флаг архивирования канала в регистратор;
- (43, **EB**) – флаг генерации сообщений для отчета тревог;
- (44, **MB**) – флаг архивирования дополнительных атрибутов в SIAD и/или регистратор. Этот флаг влияет также на генерацию сообщений для отчета тревог;
- (45, **T**) **Время изменения** – в этот атрибут записывается время пересчета/отработки канала; атрибут возвращает 4-байтовое число секунд с 01.01.70;
- (46, **QE**) **Защелка** – этот атрибут используется в каналах **FLOAT** и **HEX16**:
 - **b14** – бит 0 **QE**, принимает значение 1 при изменении значения канала (в канале **FLOAT**, в котором вычисляется интервал, – при изменении (7, **P**));
 - **b15** – бит 1 **QE** (бит 8 **QE** в канале **FLOAT**, в котором вычисляется интервал), принимает значение 1 при аппаратной недостоверности канала. **b15** автоматически не обнуляется.

В канале **FLOAT**, в котором вычисляется интервал: после вычисления **b14** и **b15** атрибуту **QE** присваивается следующее значение:

$$QE = P * b14 + 256 * b15$$

Возможна посылка следующих значений в атрибут 46:

- 1 – сброс **b14**;
- 2 – сброс **b15**;
- 5 – сброс **b14** и **b15**;
- 3 или 4 – сброс **b14** и **b15** и квитирование последнего сообщения в ОТ по каналу;
- (47, **iDstr**) – запрос DDE (только чтение). Этот атрибут также используется:
 - в канале **CALL** (см. **Замена виртуальной привязки аргументов в канале CALL**);
 - в канале обмена через драйвер (см. **Шаблоны каналов обмена**);
- (48, **WF**) – номер класса канала;
- (49, **DBG**) **Отладка** – если этот флаг установлен, в отладочный файл профайлера <имя файла prj>_<ordinal>.txt (**ordinal** – порядковый номер узла) будет выводиться информация, определяемая каналом, связанным с системной переменной **@Debug** (группа

СИСТЕМНЫЕ);

- (50) – флаг **Запрос времени значения** (см. Вкладка ‘Флаги’);
- (51, **blO**) **Change IO**:
 - числовой канал типа OUTPUT (сетевой или локальный, подтипы 71 и 70) – если флаг установлен, в начальный момент времени канал считывает значение привязанного канала-приемника и записывает в себя, после чего флаг сбрасывается;
 - канал СОБЫТИЕ – при считывании стека событий по сети флаг устанавливается автоматически;
 - канал CALL.Vector – при считывании удаленного архива по сети флаг устанавливается автоматически;

Отображение состояния флага:

- +/- – флаг установлен/сброшен;
- . – несмотря на то, что флаг установлен, никаких действий не производится (например, при попытке считывания отсутствующего стека канала СОБЫТИЕ);
- (52, **FS**):
 - значение байта 0 (0xFF, **ErCode**) указывает на причину возникновения аппаратной недостоверности:
 - 0 – нет;
 - 1 – стандартная (например, флаг установлен аппаратурой);
 - 2 – ошибка интерфейса (запрос ушел, а ответа нет);
 - 3 – **In=0** при типе сигнала 4-20mA или 1-5V (см. **Канал класса FLOAT**);
 - 4 – по времени;
 - бит 8 (0x100, **Request**) – этот бит устанавливается, когда запрос для данного канала сформирован и поставлен в очередь на исполнение (только для асинхронных каналов);
 - бит 9 (0x200, **Response**) – этот бит устанавливается, когда от устройства пришел корректный ответ (только для асинхронных каналов). Отсутствие ответа в **Response** не фиксируется (т.е. если ответа нет в течение заданного таймаута, устанавливается флаг аппаратной недостоверности, а **Response=0**);
 - бит 12 (0x1000, **DataReady**) – то же, что и атрибут 120, **ACK**;
 - бит 16 (0x10000, **ForceUpdate**) – принудительное обновление графики (0 – не задано, 1 – задано); этот бит сбрасывается автоматически.

Строковое значение атрибута 52,**FS** содержит четыре цифры:

<ForceUpdate>.<Response><Request>.<ErCode>

В атрибут могут быть посланы следующие значения:

- 0 – сбросить **ErCode** и признак аппаратной недостоверности;
- 1, 2, 3, 4 – установить соответствующее значение **ErCode**;
- 8 – установить 39,EXEC и сбросить **Response** и **Request**;
- 0x100 – инвертировать **Request**;
- 0x200 – инвертировать **Response**;
- 0x10000 – установить **ForceUpdate**;
- (55, **relink**) **Перепривязка** – этот атрибут предназначен для перепривязки аргументов канала CALL (см. **Замена привязок аргументов канала CALL на привязки аргументов другого канала CALL**);
- (56, **nBits**) **Число бит** – этот атрибут отображает число бит, заданное для канала в ИС (см. **Канал класса HEX16** и **Канал класса HEX32**);
- (57, **Iala**) – индекс словаря сообщений ОТ для данного канала;
- (58, **DumpSync**):
 - 0 – канал не использует дамп и игнорируется при синхронизации резервов (см. описание функций синхронизации в разделе **Задание параметров узла**, а также раздел **Резервирование в TRACE MODE 6**);
 - 1 – (установлен бит 0 **DumpSync**); канал не использует дамп, но участвует в синхронизации резервов;
 - 2 (установлен бит 1 **DumpSync**) – канал использует дамп (чтение при старте/запись), но игнорируется при синхронизации резервов;
 - 3 (установлены биты 0 и 1 **DumpSync**) – канал использует дамп и участвует в синхронизации резервов.
- (59, **DR**) – при чтении этого атрибута как числа, значение битов 0-30 индицирует номер последнего сообщения в ОТ по каналу. Установленный бит 31 (0x80000000) индицирует, что сообщение квитировано. В канале CALL.ChGroupReq/CALL.AS_DATA, в который записана выборка из архива, атрибут 59 индицирует время начала выборки (T_FROM);
- (60, **ToNet**) – индекс автопосылки («N»), если автопосылка не задана);
- (61, **Nind**) – **QDS**, описатель качества.

Значения при чтении атрибута **Индекс** (61, **Nind**), если в файле *.cnf задан ключ **TM_FA_QDS=DEF**:

- 0x01 – **OV** (выход за пределы) – устанавливается для каналов FLOAT с границами, если интервал отличен от 0;

- 0x10 – **BL** (**is_block**, заблокирован, т.е. не передается);
- 0x20 – **SB** (замещен), **is_sb** | (8,**W**);
- 0x40 – **NT** (**not_actual**, не актуален);
- 0x80 – **IV** (не действителен), по недостоверности ((4,**I**) & 0x1=1).

Значения при чтении атрибута **Индекс** (61, **Nind**), если в файле *.cnf задан ключ **TM_FA_QDS=ON**:

- 0x01 – **OV** (выход за пределы) – устанавливается для каналов FLOAT с границами, если интервал отличен от 0;
- 0x10 – **BL** (**is_block**, заблокирован, т.е. не передается);
- 0x20 – **SB** (замещен), **is_sb**;
- 0x40 – **NT** (**not_actual**, не актуален);
- 0x80 – **IV** (не действителен), по недостоверности ((4,**I**) & 0x1=1).

Для каналов HEX32 с представлением DEC:

- 0x10 – **BL**, только если **TM_FA_QDS=ON**;
- 0x20 – **CY**, было переполнение или нет (сбрасывается сбросом счетчика);
- 0x40 – **CA**, значение было установлено/сброшено (сбрасывается чтением по IEC104 IDT101);
- 0x80 – **IV**.

Сброс счетчика осуществляется SLAVE104 IDT101 или локально посылкой значения 0 в атрибут (7, **P**) **Интервал**.

Ключ **TM_FA_QDS_ACT=OFF** запрещает автоматическое формирование **NT** – по умолчанию для каналов INPUT по протоколам ModBusTCP, ModBusRTU, MitsubishiTCP и UDPOmron). Если за период пересчета канал не был считан и не возведена недостоверность, **NT=1**.

Ключ **TM_FA_QDS_REC=OFF** запрещает автоматическое формирование **NT**, **BL**, **SB** и **OV** для каналов INPUT, полученных по IEC104.

Ключ **TM_FA_QDS_TIME=ON** разрешает обработку бита недостоверности временной отметки, полученной по IEC104. Если получены данные с недостоверностью временной отметки, то в канале атрибут (45,**T**) равен времени получения, а не полученному времени.

Формат записи в атрибут (61, **Nind**):

BBVV

- **BB** – биты для установки;

- **VV** – значения бит;
- (78, **HF**) – 1 для целочисленных каналов (**HEX16**, **HEX32**, **Событие**), 0 – для каналов других классов;
- (79, **CODE**) – кодировка канала;
- (80, **COMMNT**) – комментарий канала;
- (81, **IO**) – тип канала;
- (82, **DIM**) – размерность реального значения канала;
- (83, **SGNL**) **Сигнал** – номер типа сигнала, заданного для источника, с которым связан канал (по списку **tmcf/signal.tmc**).

При выводе канала CALL в генерируемый документ в виде вертикальной таблицы (см. **Номер SubNum**):

- значение битов 2-4 задает формат значений float;
- значение битов 5-7 задает формат временных меток:
 - 0 – короткая дата (2 разряда для года) и время с миллисекундами;
 - 1 – короткая дата (2 разряда для года) и время без миллисекунд;
 - 2 – время без миллисекунд;
 - 3 – время с миллисекундами;
 - 4 – короткая дата (2 разряда для года), часы и минуты;
 - 5 – короткая дата (4 разряда для года) и часы;
 - 6 – короткая дата (4 разряда для года);
 - 7 – длинная дата (наименование месяца сокращенное, 4 разряда для года);
- (86, **nAtrt**) **Привязка** – если компонент вызывает шаблон, **nAtrt**=255. Если компонент связан с некоторым атрибутом **A** другого компонента, **nAtrt** равно индексу атрибута **A**. В остальных случаях **nAtrt**=0;
- (87, **CC**) – этот атрибут используется в канале CALL с типами вызова **Program**, **SQLQuery**, **Document(Report)** и **Screen** (см. **Атрибуты канала класса CALL**);
- (88, **ms**) – в этот атрибут записываются миллисекунды времени изменения значения канала (дополнение к атрибуту 45);
- (89, **IA_L0**, MPB – **L0**) **L0** – значение байтов 0-3 (0x0000FFFFFFFF) удаленного адреса;
- (90, **IA_I0**, MPB – **I0**) **I0** – значение нулевого слова (0x00000000FFFF) удаленного адреса; адрес привязанного аппаратного тега;
- (91, **IA_I1**, MPB – **I1**) **I1** – параметр **A** запроса DDE (чтение/запись, см. **Редактор переменной DDE**); значение первого слова (0x0000FFFF0000) удаленного адреса. Этот атрибут используется также в канале CALL с типом вызова **Screen** (см. **Атрибуты ка-**

нала класса CALL);

- (92, **IA_I2**, MPB – **I2**) **I2** – 1) параметр **B** запроса DDE (чтение/запись); 2) значение второго слова (0xFFFF00000000) удаленного адреса; 3) у каналов обмена с другим узлом – номер узла;
- (93, **IA_C0**, MPB – **C0**) **C0** – значение нулевого байта (0x0000000000FF) удаленного адреса;
- (94, **IA_C1**, MPB – **C1**) **C1** – значение первого байта (0x00000000FF00) удаленного адреса;
- (95, **IA_C2**, MPB – **C2**) **C2** – значение второго байта (0x000000FF0000) удаленного адреса;
- (96, **IA_C3**, MPB – **C3**) **C3** – значение третьего байта (0x0000FF000000) удаленного адреса;
- (97, **IA_C4**, MPB – **C4**) **C4** – значение четвертого байта (0x00FF00000000) удаленного адреса;
- (98, **IA_C5**, MPB – **C5**) **C5** – значение пятого байта (0xFF0000000000) удаленного адреса;
- (116, только MPB – **TVformat**) –
- (118, **ID**) – зарезервировано;
- (119, **extL**) – служебная переменная (номер шаблона, который вызывается данным каналом);
- (120, **ACK**) – переменная **DataReady**, принимает значение 1, если запрошенные данные успешно считаны;
- (123, **CTYPE**, MPB – **What**) – тип вызова (для канала с настроенным свойством **вызов** – см. **Канал класса CALL**);
- (124, **ArgSize**) – число аргументов, их список, текущие значения и привязки. Аргументы также являются атрибутами канала, их индексация начинается с 1000. В ИС 47 аргументов канала (индексы 1000-1046) доступны для привязки через атрибуты 140-186 (привязка к атрибуту 140 в ИС равнозначна привязке к аргументу с индексом 1000);
- (125, **RInd**) – для каналов **CALL**, вызывающих шаблоны, – оценка времени выполнения в миллисекундах; для канала, принимающего сетевую автопосылку, – номер этой автопосылки;
- (126, **TsT**) – этот атрибут индицирует информацию о канале в следующем формате:
 (<class><subtype>.<addition>_p<num>

где

- **class** – обозначение класса канала;
- **subtype** – номер **подтипа** канала (внутренний параметр TRACE MODE – см. **Подтипы каналов**);
- **addition** – номер **дополнения к подтипу** канала (внутренний параметр TRACE MODE – см. **Подтипы каналов**);

- **num** –
- (127, **NAME**) – имя канала;
- (187, **MST**) –
- (240, **ERR**) – в канале CALL – ошибка исполнения канала:
 - CALL.Program, CALL.SQLQuery и CALL.Document(Report);
 - канал выборки из архива;
 - CALL.EMAIL;
 - CALL.VECTOR – атрибут 240 индицирует также ошибки обмена по TCP и RS;
 - CALL.SCREEN;
 - CALL.ChGroupReq (MODBUS-TCP, MODBUS-RS MITSUBISHI);
- (245, **A_OPT**) – имеет различное назначение в различных каналах (см., например, **Особенности вызова графического экрана**);
- (246, **VFRMT**) – формат строкового представления значения канала;
- (252, только MPB – **dT**) – в канале CALL.ChGroupReq / CALL.AS_DATA, в который записана выборка из архива, – временной шаг данных (в секундах);
- (254, **RST**) – при посылке значения в этот атрибут выполняется соответствующая функция:
 - 1 – в каналах **CALL.Screen**, **CALL.Program**, **CALL.SQLQuery** и **CALL.Document(Report)**: перезагрузка шаблона из файла *.res (см. **Файлы узла, создаваемые при экспорте**). Для **CALL.Screen**: если перезагрузка шаблона производится при перерисовке экрана, экран закрывается;
 - 2 – в каналах **CALL.Screen**: сохранение экрана в файл (см. **Сохранение экрана в файл**);
 - произвольное значение – в обобщающем канале СОБЫТИЕ: генерация файла (см. **Канал класса СОБЫТИЕ**).

Назначение атрибута (254, **RST**) в других каналах:

- если (45, **T**) = 0 или канал выключен, **RST** = 0;
- если (45, **T**) <> 0 и канал включен, атрибут (254, **RST**) хранит информацию, указанную в таблице ниже. В таблице используются следующие обозначения: **DEC**, **HEX** – вид представления каналов HEX16 и HEX32; **dT** = **T**_{текущее} – (45, **T**); **Invalid=1** – в канале установлен флаг аппаратной неадекватности.

Класс и конфигурация канала	RST как число (*)	RST как строка (*)
HEX16 или HEX32 DEC, b14=1 или b15=1	4, если R=0 7, если R<>0 и dT<1с 1, если R<>0 и dT>1с	Без словаря: <R> Со словарем: <словарное сообщение с номером R> Если Invalid=1: Error <R>
HEX16 или HEX32 DEC, b14=b15=0	2, если R=0 3, если R<>0	То же
HEX16 или HEX32 HEX, b14=1 или b15=1	7, если dT<1с 1, если dT>1с	Без словаря: <R> Со словарем: <конкатенация словарных сообщений об установленных битах R> Если Invalid=1: Error <R>
HEX16 или HEX32 HEX, b14=b15=0	2	То же
FLOAT или DOUBLE FLOAT P не вычисляется, b14=1 или b15=1	7, если dT<1с 1, если dT>1с	Без словаря: <R> Со словарем: <словарное сообщение с номером R> Если Invalid=1: Error <R>
FLOAT или DOUBLE FLOAT P не вычисляется, b14=b15=0	4	То же
FLOAT или DOUBLE FLOAT P вычисляется, b14=1 или b15=1	4, если P=0 7, если P<>0 и dT<1с 1, если P<>0 и dT>1с	Без словаря: <P> (<R>) Со словарем: <словарное сообщение с номером P> <R> Если Invalid=1: Error <R>
FLOAT или DOUBLE FLOAT P вычисляется, b14=b15=0	2, если P=0 3, если P<>0	То же

Для ГЭ **События**: значения **RST** соответствуют статусам канала **Событие**, сброс **b14** и **b15** интерпретируется как квитирование.

При выводе значений атрибутов/аргументов каналов в окно просмотра компонентов, таблицу, БД или файл MPB преобразует значения в текстовую строку:

- канал FLOAT – реальное значение (в разных форматах);
- канал CALL.ChGroupReq и CALL.TVC – значение числовых и

временных аргументов (в разных форматах). Если аргумент привязан к каналу, выбирается формат, заданный в канале.

Профайлер позволяет изменить значение любого атрибута, однако следует иметь в виду, что некорректные действия с атрибутами могут привести к неработоспособности канала.

Числовые каналы

Обработка данных в числовых каналах

Числовые каналы (кроме каналов DOUBLE FLOAT и TIME – см. **Классификация компонентов**) снабжены внутренними алгоритмами обработки данных, принятых от источников или передаваемых приемникам (см. **Математическая обработка данных**). В этих алгоритмах, в общем случае, задействованы 4 атрибута – **Входное значение** (2, **In**), **Аппаратное значение** (1, **A**), **Реальное значение** (0, **R**) и **Выходное значение** (9, **Q**).

Трансляция

Трансляцией называется вызов программы числовым каналом (это единственное назначение свойства **ВЫЗОВ** числового канала).

При вызове программы числовым каналом (кроме канала TIME) может быть выполнено следующее преобразование его атрибутов:

аппаратное значение (A) <=> программа <=> реальное значение (R)

Направление преобразования зависит от типа канала:

- INPUT: **A=>программа=>R**
- OUTPUT: **R=>программа=>A**

Направление преобразования следует учитывать для корректной привязки атрибутов числового канала к аргументам программы.

Канал класса FLOAT

Тип данных канала класса **FLOAT** – **REAL** (см. **Определение переменных и констант**).

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), и атрибутов, общих для числовых каналов (см. **Атрибуты каналов, отображаемые профайлером**), каналы класса **FLOAT** имеют специфические атрибуты.

К специфическим атрибутам, которые могут быть заданы в редакторе канала **FLOAT** (см. **Редакторы каналов**), относятся следующие:

- раздел «Границы»:
 - флаг **Использовать** – (85, **BNDR**) – установка этого флага в редакторе равнозначна присвоению атрибуту **BNDR** зна-

чения 0, что разрешает монитору анализировать значения шести границ канала (HL, HA, HW, LW, LA и LL – см. **Границы и интервалы канала FLOAT**). При **BNDR=1** (аналог снятия флага **Использовать**) анализ границ запрещен. От этого флага зависит исполнение алгоритма клиппирования в канале OUTPUT (см. **Обработка в канале FLOAT**);

- **ВП** – (26, **HL**) – значение верхнего предела;
- **ВА** – (28, **HA**) – значение верхней аварийной границы;
- **ВГ** – (30, **HW**) – значение верхней предупредительной границы;
- **НГ** – (31, **LW**) – значение нижней предупредительной границы;
- **НА** – (29, **LA**) – значение нижней аварийной границы;
- **НП** – (27, **LL**) – значение нижнего предела;
- **Гистерезис** – (32, **Hyst**) – от этого параметра зависят условия генерации сообщений при переходе реальным значением канала заданных границ (см. **Границы и интервалы канала FLOAT**);
- флаг **Контроль границ** – (53, **SC_F**, MPB – **LMT**) – установка этого флага равнозначна присвоению атрибуту **LMT** значения 1. Действие флага различно для каналов типов INPUT и OUTPUT. В первом случае наличие флага означает разрешение установки каналу признака программной недостоверности (см. **Атрибуты каналов, отображаемые профайлером**) в случае выхода реального значения канала за пределы диапазона [**LL**, **HL**] (если **BNDR=0**). При возврате реального значения в диапазон признак программной недостоверности автоматически сбрасывается. Для типа OUTPUT установка флага **Контроль границ** разрешает клиппирование реального значения канала (если **BNDR=0** – см. **Обработка в канале FLOAT**). При **LMT=0** или **BNDR=1** описанные алгоритмы не исполняются;
- раздел «Обработка»:
 - флаг **Использовать** – (50, **PRS**, недоступен для изменения в реальном времени) – если этот флаг снят, канал является каналом FLOAT без обработки, если флаг установлен – каналом FLOAT с обработкой (см. **Обработка в канале FLOAT**);
 - **Апертура** – (35, **AP**, MPB – **APert**) – этот параметр конфигурирует алгоритм фильтрации малых изменений значения (см. **Обработка в канале FLOAT**). По умолчанию **APert = 0**;
 - **Пик** – (36, **DP**, MPB – **DPic**) – этот параметр конфигурирует алгоритм подавления одиночных пиков в канале INPUT и

алгоритм линейного сглаживания – в канале OUTPUT (см. **Обработка в канале FLOAT**). По умолчанию $DPic=10000$;

- **Сглаживание** – (37, **DS Экспоненциальное сглаживание**, MPB – **DSmoot**) – коэффициент ($0 \leq DSmoot \leq 1$) в стандартном алгоритме экспоненциального сглаживания (см. **Обработка в канале FLOAT**). При $DSmoot=0$ (значение по умолчанию) этот алгоритм не выполняется;
- **Множитель** (33, **KX**) и **Смещение** (34, **Z Дрейф нуля**, MPB – **ZERO**) – параметры масштабирования (см. **Обработка в канале FLOAT**):

$A=In \cdot KX+Z$ в канале типа INPUT;

$Q=(A+Z) \cdot KX$ в канале типа OUTPUT.

По умолчанию $KX=1$, $Z=0$.

Атрибуты **Множитель** и **Смещение** могут быть также рассчитаны в разделе **Масштабирование** (для активизации раздела нужно установить флаг **Масштабирование**). Этот раздел, в зависимости от типа канала (INPUT или OUTPUT), имеет вид соответствующей формулы преобразования:

- для канала типа INPUT:

Масштабирование

Масштабирование

In	Множитель	Смещение	A
Max 200	× 4.54	+ -154	Max 754
Min 100			Min 300

Рассчитать

- для канала типа OUTPUT:

Масштабирование

Масштабирование

A	Смещение	Множитель	Q
Max 200	+ 33.3333	× 2.25	Max 525
Min 100			Min 300

Рассчитать

Для расчета множителя и смещения нужно ввести входной диапазон (диапазон изменения атрибута **In** для канала INPUT или **A** для канала OUTPUT), выходной диапазон (диапазон изменения атрибута **A** для канала INPUT или **Q** для канала OUTPUT) и нажать кнопку **Рассчитать**. Для задания диапазонов используются соответствующие поля **Min** и **Max**.

Особенности установки признака достоверности

Если канал FLOAT типа INPUT связан с источником, для которого задан

тип сигнала 4-20mA или 1-5V, и при этом входное значение канала равно 0, ему устанавливается признак аппаратной недостоверности – для этого монитор анализирует значения атрибутов (2, **In**) и (84, **b11**). (В процессе конвертирования в атрибут **b11** передается значение атрибута 83, **SGNL** – см. **Атрибуты каналов, отображаемые профайлером**).

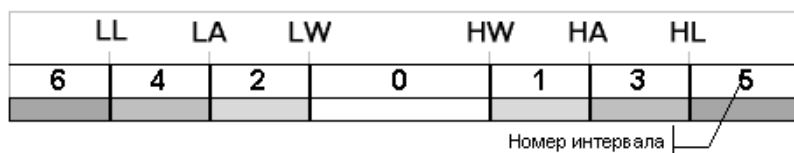
Границы и интервалы канала FLOAT

Границы и интервалы

Для мониторинга состояния техпроцесса (см. **Обработка аварийных ситуаций**) для каналов FLOAT могут быть заданы 6 границ (см. **Канал класса FLOAT**):

Некоторые OEM-версии мониторов и **NanoRTM** имеют ограничения по числу анализируемых границ.

Границы задают диапазоны (интервалы), в которых может находиться значение отслеживаемого параметра:



В случае, когда задано не менее двух ненулевых значений границ и полный перечень заданных значений границ корректен ($LL \leq LA \leq LW \leq HW \leq HA \leq HL$), монитор на каждом цикле пересчета определяет номер интервала, в котором находится реальное значение канала, и записывает этот номер в атрибут **Интервал** (7, **P**).

С помощью флага (53, **LMT**) **Контроль границ** можно задать режим установки каналу типа INPUT признака программной недостоверности в случае выхода его значения за пределы диапазона [**LL**, **HL**]. Этот режим может быть использован в том случае, если в канал записывается некоторая величина, диапазон изменения которой заранее известен, и всякое значение вне этого диапазона, принятое в канал, является следствием какой-либо ошибки или сбоя, поэтому не может быть использовано для анализа.

Если флаг **Контроль границ** установлен для канала типа OUTPUT, реальное значение этого канала при выходе из диапазона [**LL**, **HL**] клиппируется.

Указанные действия выполняются в том случае, если разрешен анализ границ (**BNDR=0**).

Генерация сообщений. Гистерезис

Условия генерации сообщений при пересечении реальным значением канала заданных границ зависят от значения атрибута **Гистерезис (32, Hyst)** (см. **Канал класса FLOAT** и **Сообщения по каналам**):

- переход реального значения канала через границы в сторону развития аварийной ситуации (т.е. в сторону увеличения номера интервала) фиксируется по заданным для границ значениям:
 $<LL, <LA, <LW, >HW, >HA, >HL$
- при обратном переходе реального значения канала через границы (в сторону уменьшения номера интервала) значения границ корректируются в соответствующем направлении на величину гистерезиса **H**:
 $>(LL+H), >(LA+H), >(LW+H), <(HW-H), <(HA-H), <(HL-H)$

По полученным таким образом значениям границ вычисляется номер интервала.

Введение гистерезиса позволяет убрать ненужный поток сообщений в отчет тревог при небольших колебаниях контролируемого параметра вблизи значения одной из границ.

Обработка в канале FLOAT

В измерительном тракте (в общем случае датчик=>УСО=>контроллер) происходит преобразование реальной физической величины (температуры, давления и т.п.) в один из следующих «инженерных» видов:

- в число, соответствующее амплитуде некоторого электрического сигнала (в том числе унифицированного – 0-10V, 4-20mA и т.д.);
- в число, соответствующее проценту от диапазона изменения некоторого электрического сигнала;
- в двоичный код (после АЦП).

В управляющем тракте (в общем случае контроллер=>УСО=>исполнительный механизм) выполняется обратное преобразование.

При обработке данных, поступающих из измерительного тракта или передаваемых в управляющий, необходимо скорректировать различные погрешности трактов. Для отображения поступающих данных требуется переводить «инженерные» данные в реально измеряемые (например, если требуется отображать значение температуры в ее физических единицах – градусах). Управляющий сигнал во многих случаях требуется сглаживать. Для решения подобных задач канал FLOAT снабжен следующими встроенными алгоритмами обработки, параметры которых могут быть заданы как в редакторе, так и в реальном времени (см. **Канал класса FLOAT**):

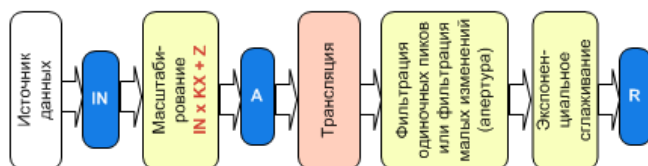
- канал INPUT:
 - масштабирование;
 - фильтрация одиночных пиков;
 - фильтрация малых изменений (апертура);
 - экспоненциальное сглаживание;
- канал OUTPUT:
 - экспоненциальное сглаживание;
 - линейное сглаживание;
 - фильтрация малых изменений (апертура);
 - клиппирование;
 - масштабирование.

При использовании экспоненциального сглаживания фильтрация малых изменений в канале FLOAT не выполняется.

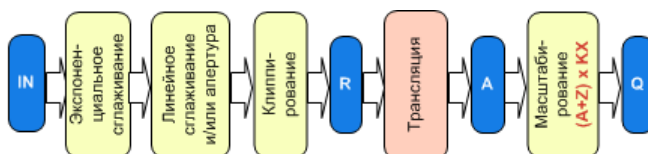
Если встроенных алгоритмов обработки данных недостаточно, в каналах FLOAT может быть использована процедура трансляции (см. **Числовые каналы**) – например, для корректировки нелинейности передаточной характеристики измерительного/управляющего тракта.

Атрибуты **Входное значение** (2, **In**), **Аппаратное значение** (1, **A**), **Реальное значение** (0, **R**) и **Выходное значение** (9, **Q**) задействованы во внутренних алгоритмах канала FLOAT следующим образом:

- канал INPUT:



- канал OUTPUT:



В указанных на рисунке формулах масштабирования **KX** и **Z** являются значениями атрибутов **Множитель** (33, **KX**) и **Смещение** (34, **Z Дрейф нуля**, **MPB – ZERO** – см. **Канал класса FLOAT**).

Все алгоритмы обработки (за исключением клиппирования реального значения в канале OUTPUT) выполняются в том случае, если для канала установлен флаг (50, **PRS**) **Использовать** – такой канал является **каналом FLOAT с обработкой**. В профайлере канал FLOAT с обработкой

индицируется как **C5_FloatExt** (атрибут 126, **TsT**).

Если флаг (50, **PRS**) **Использовать** не установлен, монитор не загружает атрибуты, конфигурирующие соответствующие алгоритмы обработки, и, соответственно, не исполняет эти алгоритмы – в этом случае канал является **каналом FLOAT без обработки**. В профайлере канал FLOAT без обработки индицируется как **C4_Float**.

Клиппирование реального значения канала OUTPUT выполняется в том случае, если установлены флаги (83, **BNDR**) **Использовать** и (53, **LMT**) **Контроль границ** (т.е. **BNDR=0** и **LMT=1**). Выполнение этой процедуры не зависит от флага (50, **PRS**) **Использовать**.

Если флаг **PRS** не установлен и в канале не используется трансляция, монитор не использует аппаратное значение канала.

В нижеприведенных формулах алгоритмов **In**, **A**, **R**, **Q**, **KX**, **Z**, **APert**, **DSmoot** и **DPic** обозначают соответствующие атрибуты канала.

Масштабирование

Масштабирование в канале FLOAT выполняется по следующим законам (при **PRS=1**):

- в канале типа INPUT:
$$A = In * KX + Z$$
- в канале типа OUTPUT:
$$Q = (A + Z) * KX$$

Экспоненциальное сглаживание

При **PRS=1** и **DSmoot** <> 0 (0 < **DSmoot** <= 1) в канале FLOAT выполняется экспоненциальное сглаживание (индекс указывает номер такта пересчета):

- в канале типа INPUT:
$$R_n = Result_n * (1 - DSmoot_n) + DSmoot_n * R_{n-1}$$

Result_n обозначает результат фильтрации пиков;
- в канале типа OUTPUT:
$$RESULT_n = In_n * (1 - DSmoot_n) + DSmoot_n * R_{n-1}$$

Результат экспоненциального сглаживания **RESULT_n** далее обрабатывается процедурой линейного сглаживания.

При **DSmoot** <> 0 фильтрация малых изменений в канале FLOAT не выполняется.

Фильтрация пиков или фильтрация малых изменений в канале INPUT

При **PRS=1** в канале INPUT выполняется фильтрация пиков или фильтрация малых изменений (апертура).

Фильтрация пиков – это алгоритм, позволяющий игнорировать в течение одного такта пересчета изменение значения сверх установленной величины (**DPic**).

Фильтрация малых изменений – это алгоритм, позволяющий игнорировать изменение значения, если это изменение меньше заданной величины (**APert**).

При использовании экспоненциального сглаживания фильтрация малых изменений в канале FLOAT не выполняется.

При обработке значения в канале выполняется один из этих алгоритмов фильтрации (индекс указывает номер такта пересчета):

- если $|\mathbf{RES}_n - \mathbf{R}_{n-1}| \leq \mathbf{DPic}_n$, то выполняется фильтрация малых изменений:
 - если $|\mathbf{RES}_n - \mathbf{R}_{n-1}| < \mathbf{APert}_n$, то $\mathbf{Result}_n = \mathbf{R}_{n-1}$;
 - если $|\mathbf{RES}_n - \mathbf{R}_{n-1}| \geq \mathbf{APert}_n$, то $\mathbf{Result}_n = \mathbf{RES}_n$;
- если $|\mathbf{RES}_n - \mathbf{R}_{n-1}| > \mathbf{DPic}_n$, то выполняется фильтрация одиночных пиков:
 - если на такте (**n-1**) фильтрация пиков не использовалась, $\mathbf{Result}_n = \mathbf{R}_{n-1}$ (первый пик фильтруется);
 - если на такте (**n-1**) фильтрация пиков использовалась, $\mathbf{Result}_n = \mathbf{RES}_n$ (второй подряд пик не фильтруется).

Здесь **RES_n** – результат процедуры трансляции или, в ее отсутствие, аппаратное значение канала; **Result_n** – результат описываемых алгоритмов (на **n**-ом такте пересчета).

Как следствие, $\mathbf{APert}_n > \mathbf{DPic}_n$ не имеет смысла, поскольку не обрабатывается никогда.

Линейное сглаживание и фильтрация малых изменений в канале OUTPUT

При **PRS=1** в канале OUTPUT выполняется линейное сглаживание и/или фильтрация малых изменений (индекс указывает номер такта пересчета):

- если $|\mathbf{RESULT}_n - \mathbf{R}_{n-1}| \leq \mathbf{DPic}_n$, то выполняется фильтрация малых изменений:
 - если $|\mathbf{RESULT}_n - \mathbf{R}_{n-1}| \leq \mathbf{APert}_n$, то $\mathbf{Res}_n = \mathbf{R}_{n-1}$;

- если $|\mathbf{RESULT}_n - \mathbf{R}_{n-1}| > \mathbf{APert}_n$, то $\mathbf{Res}_n = \mathbf{RESULT}_n$;
- если $|\mathbf{RESULT}_n - \mathbf{R}_{n-1}| > \mathbf{DPic}_n$, то вычисляется результат линейного сглаживания $\mathbf{LIN}_n = \mathbf{R}_{n-1} + \mathbf{DPic}_n$ (если $\mathbf{RESULT}_n > \mathbf{R}_{n-1}$) или $\mathbf{LIN}_n = \mathbf{R}_{n-1} - \mathbf{DPic}_n$ (если $\mathbf{RESULT}_n < \mathbf{R}_{n-1}$) и далее выполняется фильтрация малых изменений:
 - если $|\mathbf{LIN}_n - \mathbf{R}_{n-1}| \leq \mathbf{APert}_n$, то $\mathbf{Res}_n = \mathbf{R}_{n-1}$;
 - если $|\mathbf{LIN}_n - \mathbf{R}_{n-1}| > \mathbf{APert}_n$, то $\mathbf{Res}_n = \mathbf{LIN}_n$;

\mathbf{RESULT}_n – результат предыдущей процедуры, \mathbf{Res}_n – результат описываемых алгоритмов на n -ом такте пересчета. \mathbf{Res}_n при $\mathbf{BNDR}=0$ и $\mathbf{LMT}=1$ обрабатывается далее процедурой клипирования.

При использовании экспоненциального сглаживания фильтрация малых изменений в канале \mathbf{FLOAT} не выполняется.

Клипирование в канале \mathbf{OUTPUT}

Ограничение реального значения в канале \mathbf{FLOAT} типа \mathbf{OUTPUT} выполняется по следующим законам (при $\mathbf{BNDR}=0$ и $\mathbf{LMT}=1$):

- если $\mathbf{Res} > \mathbf{HL}$, то $\mathbf{R} = \mathbf{HL}$;
- если $\mathbf{Res} < \mathbf{LL}$, то $\mathbf{R} = \mathbf{LL}$;
- если $\mathbf{LL} \leq \mathbf{Res} \leq \mathbf{HL}$, то $\mathbf{R} = \mathbf{Res}$.

Здесь \mathbf{Res} – результат предыдущей процедуры обработки (линейного сглаживания и/или фильтрации малых изменений).

Канал класса $\mathbf{DOUBLE FLOAT}$

Тип данных канала класса $\mathbf{DOUBLE FLOAT}$ – \mathbf{LREAL} (см. **Определение переменных и констант**).

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), и атрибутов, общих для числовых каналов (см. **Атрибуты каналов, отображаемые профайлером**), каналы класса $\mathbf{DOUBLE FLOAT}$ имеют специфические атрибуты.

К специфическим атрибутам, которые могут быть заданы в редакторе канала, относятся следующие:

- раздел «Границы»:
 - флаг **Использовать** – (85, \mathbf{BNDR}) – установка этого флага в редакторе равнозначна присвоению атрибуту \mathbf{BNDR} значения 0, что разрешает монитору анализировать значения двух границ канала (атрибутов \mathbf{HL} и \mathbf{LL}). При $\mathbf{BNDR}=1$ (аналог снятия флага **Использовать**) границы не анализируются;

- **ВП** – (26, **HL**) – значение верхней границы;
- **НП** – (27, **LL**) – значение нижней границы.

Границы канала задают диапазоны (интервалы), в которых может находиться его реальное значение:

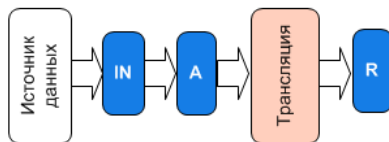


Монитор на каждом цикле пересчета определяет номер интервала, в котором находится реальное значение канала, и записывает этот номер в атрибут **Интервал** (7, **P**).

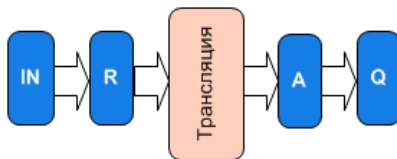
Для каналов DOUBLE FLOAT может быть определена процедура трансляции (см. **Числовые каналы**).

Атрибуты **Входное значение** (2, **In**), **Аппаратное значение** (1, **A**), **Реальное значение** (0, **R**) и **Выходное значение** (9, **Q**) канала DOUBLE FLOAT связаны следующим образом:

- канал типа INPUT:



- канал типа OUTPUT:



В этом канале, в отличие от канала FLOAT, атрибут **Аппаратное значение** (1, **A**) используется во всех случаях.

В профайлере канал DOUBLE FLOAT индексируется как **C6_DFloat** (атрибут 126, **TsT**).

Канал класса HEX16

Тип данных канала класса **HEX16** – **UINT** (см. **Определение переменных и констант**).

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), и атрибутов, общих для числовых каналов (см. **Атрибуты каналов, отображаемые профайлером**), каналы класса HEX16 имеют специфические атрибуты.

К специфическим атрибутам, которые могут быть заданы в редакторе канала, относятся следующие:

- раздел «Параметры»:
 - **Размерность в битах** – этот параметр задает число бит значения канала для некоторых операций по следующему закону:
 - 1 – 1 бит;
 - 2 – 2 бита;
 - 3-8 – 8 бит;
 - 9-16 – 16 бит.

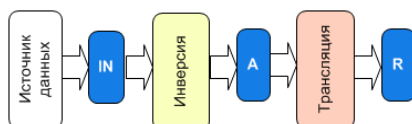
В реальном времени заданное число бит отображает атрибут (56, **nBits**) **Number Bits**.

Операции, в которых используется заданное число бит:

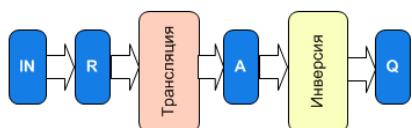
- инверсия;
- 1 бит – см. **MPB как сервер протокола МЭК 60870-104**;
- 1, 2 и 8 бит – для функции **W_Byte(15)** при обмене по MOD-BUS;
- флаг **Инверсия** (40, **NM**) – если этот флаг установлен, инвертирование в канале разрешено;
- флаг **DEC** (84, **HD**) – если этот флаг установлен (**HD=1**), значение канала отображается в профайлере в десятичном виде; если флаг не установлен (**HD=0**) – в шестнадцатеричном. От этого флага зависит также алгоритм записи сообщений в отчет тревог (см. **Сообщения по каналам HEX16 и HEX32**).

Атрибуты **Входное значение** (2, **In**), **Аппаратное значение** (1, **A**), **Реальное значение** (0, **R**) и **Выходное значение** (9, **Q**) канала HEX16 задействованы в его алгоритмах обработки следующим образом (см. также **Числовые каналы**):

- канал типа INPUT:



- канал типа OUTPUT:



В отсутствие процедуры трансляции, в каналах HEX16 не используется

атрибут **Аппаратное значение** (1, **A**).

К специфическим атрибутам канала HEX16, отображаемым в профайлере, относятся также следующие:

- **Бит1...Бит16** (10...25, **b1...b16**) – значение соответственно бита 0...15 реального значения канала;
- **Байт0, Байт1** (100, **Byte0**; 101, **Byte1**) – значение соответственно байта 0 (0x00FF) и байта 1 (0xFF00) реального значения канала.

Следующие специфические атрибуты содержат значения соответствующих групп битов реального значения канала:

- **Биты_0-3** (104, **bits_0-3**) – 0x000F;
- **Биты_4-7** (105, **bits_4-7**) – 0x00F0;
- **Биты_8-11** (106, **bits_8-11**) – 0x0F00;
- **Биты_12-15** (107, **bits_12-15**) – 0xF000;
- **Биты_0-1** (108, **bits_0-1**) – 0x0001 и 0x0002;
- **Биты_2-3** (109, **bits_2-3**) – 0x0004 и 0x0008;
- **Биты_4-5** (110, **bits_4-5**) – 0x0010 и 0x0020;
- **Биты_6-7** (111, **bits_6-7**) – 0x0040 и 0x0080;
- **Биты_8-9** (112, **bits_8-9**) – 0x0100 и 0x0200;
- **Биты_10-11** (113, **bits_10-11**) – 0x0400 и 0x0800;
- **Биты_12-13** (114, **bits_12-13**) – 0x1000 и 0x2000;
- **Биты_14-15** (115, **bits_14-15**) – 0x4000 и 0x8000.

Запись значений в эти атрибуты равнозначна изменению соответствующих битов/байтов входного значения канала.

В профайлере канал HEX16 индицируется как **C2_Hex16** (атрибут 126, **TsT**).

Канал класса HEX32

Тип данных канала класса **HEX32** – **DINT** (см. **Определение переменных и констант**).

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), и атрибутов, общих для числовых каналов (см. **Атрибуты каналов, отображаемые профайлером**), каналы класса HEX32 имеют специфические атрибуты.

К специфическим атрибутам, которые могут быть заданы в редакторе канала, относятся следующие:

- раздел «Параметры»:
 - **Размерность в битах** – этот параметр задает число бит значения канала для некоторых операций по следующему

закону:

- 1 – 1 бит;
- 2 – 2 бита;
- 3-8 – 8 бит;
- 9-16 – 16 бит;
- 17-24 – 24 бита;
- 25-32 – 32 бита.

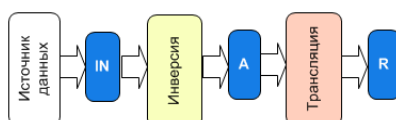
В реальном времени заданное число бит отображает атрибут (56, **nBits**) **Number Bits**.

Операции, в которых используется заданное число бит:

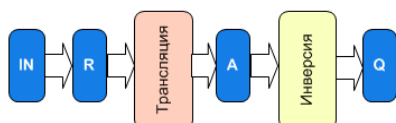
- инверсия;
- 1 бит – см. **MPB как сервер протокола МЭК 60870-104**;
- 1, 2 и 8 бит – для функции **W_Byte(15)** при обмене по MOD-BUS;
- флаг **Инверсия** (40, **NM**) – если этот флаг установлен, инвертирование в канале разрешено;
- флаг **DEC** (84, **HD**) – если этот флаг установлен (**HD=1**), значение канала отображается в профайлере в десятичном виде; если флаг не установлен (**HD=0**) – в шестнадцатеричном. От этого флага зависит также алгоритм записи сообщений в отчет тревог (см. **Сообщения по каналам HEX16 и HEX32**).

Атрибуты **Входное значение** (2, **In**), **Аппаратное значение** (1, **A**), **Реальное значение** (0, **R**) и **Выходное значение** (9, **Q**) канала HEX32 задействованы в его алгоритмах обработки следующим образом (см. также **Числовые каналы**):

- канал типа INPUT:



- канал типа OUTPUT:



В отсутствие процедуры трансляции, в каналах HEX32 не используется атрибут **Аппаратное значение** (1, **A**).

К специфическим атрибутам канала HEX32, отображаемым в профайлере, относятся также следующие:

- **Бит1...Бит16** (10...25, **b1...b16**) – значение соответственно бита 0...15 реального значения канала;
- **Бит17...Бит32** (62...77, **b17...b32**) – значение соответственно бита 16...31 реального значения канала
- **Байт0...Байт3** (100...103, **Byte0...Byte3**) – значение соответственно байта 0 (0x000000FF), байта 1 (0x0000FF00), байта 2 (0x00FF0000) и байта 3 (0xFF000000) реального значения канала.

Запись значений в эти атрибуты равнозначна изменению соответствующих битов/байтов входного значения канала.

В профайлере канал HEX32 индицируется как **C3_Hex32** (атрибут 126, TsT).

Канал класса TIME

Канал класса **TIME** предназначен для работы со значениями даты и времени. Каналы этого класса, как правило, имеют тип INPUT, атрибуты **Аппаратное значение** (1, **A**) и **Выходное значение** (9, **Q**) в них не используются.

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), и некоторых атрибутов, общих для числовых каналов (см. **Атрибуты каналов, отображаемые профайлером**), каналы класса **TIME** имеют специфические атрибуты.

В редакторе может быть задан единственный специфический атрибут этого канала – **Формат представления** реального значения канала (слева в скобках указано значение атрибута (7, **P**), этот список хранится в файле **TIME_def.tmc**):

- (0) **Дата и время ОС** (в профайлере – **Date Time**);
- (1) **Только дата ОС** (в профайлере – **Date**);
- (2) **Только время ОС** (в профайлере – **Time**);
- (3) **Число секунд с 01.01.70** (в профайлере – **As Long**);
- (4) **Дата и время Access** (в профайлере – **Date Time Access**);
- (5) **d..dD hh:mm:ss** (в профайлере – **XDhXmXs**);
- (6) **h...hH:mm:ss** (в профайлере – **Period**).

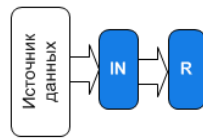
В Windows CE эта функция не работает. Формат даты и времени при работе в этой ОС устанавливается в соответствии с языком (русский – формат соответствует русскому по умолчанию, английский и китайский – формат соответствует английскому (США) по умолчанию).

К специфическим атрибутам канала TIME, отображаемым в профайлере,

относятся также следующие:

- **Секунды** (10, **s**);
- **Минуты** (11, **m**);
- **Часы** (12, **h**);
- **День** (13, **day**);
- **Месяц** (14, **month**);
- **Год** (15, **year**);
- **День года** (16, **day_of_y**);
- **День недели** (17, **day_of_w**);
- **Зимнее/летнее время** (18, **daylight**, 0 – зимнее, 1 – летнее).

Атрибуты **Входное значение** (2, **In**) и **Реальное значение** (0, **R**) канала **TIME** связаны следующим образом:



Значение даты и времени может быть записано в канал **TIME** из аргумента с типом данных **DATE_AND_TIME** или из аргумента с целочисленным 4-байтовым типом данных (см. **Определение переменных и констант**).

Канал **TIME**, привязанный к атрибуту (45, **T**) другого канала, отображает разницу между текущим временем и привязанным атрибутом.

Если канал **TIME** – не сетевой, а в **snf**-файле задан ключ **HANDLE=ON**, то атрибут (9, **Q**) канала – это строка, формируемая из атрибута (0, **R**) в соответствии с форматом в нотации **Si**, заданным в атрибуте (80, **CMNT**) (см. **Формат Si вывода даты и времени**).

В профайлере канал **TIME** индексируется как **C13_Time** (атрибут 126, **TsT**).

Анализ рабочего календаря

Атрибут (6, **D**) **Тенденция** канала класса **TIME** индексирует результат анализа даты из значения этого канала (при обработке канала **TIME** предварительно анализируются каналы **CALL.EXCEPT_DAY** узла – см. **Канал CALL.EXCEPT_DAY**):

- **D=1** – рабочий день;
- **D=2** – выходной день;
- **D=3** – зарезервировано;
- **D=0** – статус дня неизвестен.

Канал класса CALL

Канал этого класса является многофункциональным.

При конфигурировании свойства **ВЫЗОВ** канал CALL настраивается на вызов шаблонов. Канал CALL с ненастроенным свойством **ВЫЗОВ** используется для различных операций – например, для извлечения и обработки архивных данных.

Атрибуты канала класса CALL

В редакторе канала CALL задаются следующие атрибуты (см. также **Общие атрибуты каналов** и **Атрибуты каналов, отображаемые профайлером**):

- **Тип вызова** (123, **What**) (см. **Типы вызова канала CALL**);
- **Параметр** (34, **FPnt**) – уточняет функциональное назначение канала при некоторых типах вызова;
- **Содержимое аргументов** – выбирается из следующего списка:

Атрибуты **Содержимое аргументов** и **Обработка** используются в каналах, содержащих данные (CALL.ChGroupReq, CALL.Vector и т.п.).

- ... –
- **snap** – срез; аргументы представляют собой независимые переменные, хранящие данные на одно и то же время;
- **trend** – аргументы хранят значения одной переменной на разное время (например, выборку из архива в канале CALL.ChGroupReq с момента времени <атрибут 59> с шагом <атрибут 252>);
- **cpu_snap** – аргументы хранят значения привязанных переменных на одно и то же время (привязки аргументов скрыты);
- **Обработка** – выбирается из следующего списка (содержится в `sql_call.tmc`):
 - ... –
 - **sum** – сумма аргументов;
 - **avr** – среднее аргументов;
 - **sum+avr** – сумма и среднее аргументов;

Вычисленные значения используются, например, в канале CALL.ROOT для вывода в документ.

- **Формат вывода** (246, **VFRMT**) – выбирается из следующего списка (содержится в **sql_call.tmc**):
 - **0** – %g;
 - **1** – %.1f;
 - **2** – %.2f;
 - **3** – %.3f;
 - **4** – %.4f;
 - **5** – %8.3f;
 - **6** – %8.2f;
 - **7** – %.3e;
- **Активен** (40, **ACTIVE**) – при обработке свойства **вызов** канала CALL этот атрибут принимает значение 1, в противном случае он равен 0;
- **Запись в файл** (129, **WFile**) – при посылке в этот атрибут неотрицательного значения **V** текущие значения аргументов канала записываются в текстовый файл **<имя канала>_<N>.dat** (**N** – целая часть **V**; если **N=0**, файл имеет имя **<имя канала>.dat**), каждая строка которого содержит обозначение аргумента и его значение.

В NTFS: если имя канала содержит двоеточие (т.е. имеет вид **<строка1>:<строка2>**), то текущие значения аргументов канала записываются в секцию **N** файла с именем **<строка1>**.

- **Чтение из файла** (128, **RFile**) – при посылке в этот атрибут отрицательного значения **V** в аргументы канала считываются значения из файла **<имя канала>_<N>.dat** или из файла **<имя канала>.dat** или из секции **N** файла с именем **<строка1>** без расширения (см. выше описание атрибута 129, **WFile**).

Единовременно может быть обработано чтение из файла или запись в файл только для одного канала.

Начальное значение атрибутов **Тип вызова**, **Параметр**, **Содержимое аргументов**, **Обработка** и **Формат вывода** задается в редакторе канала.

Привязка аргументов каналов CALL имеет особенности (см. **Связь через аргументы**).

В профайлере канал CALL индексируется как **C1_Control** (атрибут 126, **TsT**), подтип – 10, номер дополнения к подтипу соответствует номеру типа вызова (DEC) (см. **Подтип 10**).

Типы вызова канала CALL

Атрибут **Тип вызова** определяет функциональное назначение канала (в ряде случаев уточняется значением атрибута **Параметр**). Ниже при описании каждого типа вызова в скобках слева указан его номер (задан в **call_type.tmc**) – для изменения типа вызова в реальном времени это значение нужно присвоить атрибуту (123, **What**) (работает для типов вызова с номерами больше 14):

- (0) **No** – запрет пересчета канала;
- (1) **Program** – вызов программы, см. **Выполнение программы в реальном времени**;
- (2) **SQLQuery** – вызов связи с БД, см. **Выполнение SQL-запросов в реальном времени**;
- (3) **Document(Report)** – генерация документа (отчета), см. **Использование разработанных шаблонов**;
- (4) **Screen** – вызов графического экрана; см. **Особенности вызова графического экрана**;
- (5) **Panel** – вызов графической панели, см. **Графические панели**;
- (8) **ChGroupReq** – см. **Канал CALL.ChGroupReq**;
- (9) **TableFunction** – см. **Канал CALL.TableFunction**;
- (10) **LongFromBits** – см. **Канал CALL.LongFromBits**;
- (11) **BitsFromLong** – см. **Канал CALL.BitsFromLong**;
- (12) =, (13) <, (14) <= – см. **Канал CALL с типами вызова 12-14**;
- (15) **Sum** – см. **Канал CALL.Sum**;
- (16) **Set** – см. **Канал CALL.Set**;
- (17) **RT_Statistics** – см. **Канал CALL.RT_Statistics**;
- (20) **Move** – см. **Канал CALL.MOVE**;
- (21) **MResource_1** – используется при решении задач T-FACTORY (см. **Канал CALL с типом вызова MRESOURCE_1**);
- (22) **MResource_2** – зарезервировано;
- (23) **LArc0**, (24) **LArc1** – см. **Индивидуальный архив**;
- (25) **DifSnap** – см. **Дифференциальный срез локального архива**;
- (29) **LocalStatistics** – см. **Обработка данных локального архива по каналу**;
- (30) **LocalQuick** – см. **Быстрая выборка данных из локального архива**;
- (31) **LocalList** – см. **Выборка данных из локального архива по каналу**;

- (32) **LocalQuick-T** – см. **Расширенная быстрая выборка данных из локального архива**;
- (33) **MUX** – см. **Канал CALL.MUX**;
- (41) **RemSIADauto** – зарезервировано;
- (42) **RemoteSIAD** – зарезервировано;
- (43) **RemoteStatistics** – зарезервировано;
- (44) **RemoteList** – зарезервировано;
- (45) **RemoteSnap** – зарезервировано;
- (46) **RemoteSIADSnap** – зарезервировано;
- (47) **RemArc0**, (48) **RemArc1** – см. **Запрос удаленного индивидуального архива**;
- (49) **RemArgument** – зарезервировано;
- (50) **AsyncCollection** – см. **Канал CALL.AsyncCollection**;
- (52) **Writer** – **принудительная запись в SIAD** (см. **Канал CALL.Writer**);
- (53) **DI_DO** – см. **Канал CALL.DI_DO**;
- (54) **MLink** – см. **Канал CALL.MLink**;
- (57) **EvReg** – см. **Регистратор аварийных событий**;
- (58) **Exec** – см. **Канал CALL.Exec**;
- (59 и 77) **TVC** – см. **Канал CALL.TVC**;
- (60) **OtherProj** – см. **Канал CALL.OtherProj**;
- (61) **STRING** – см. **Запись длинных строк в канал CALL**;
- (62) **Vector** – см. **Канал CALL.Vector**;
- (64) **ObjModel** – зарезервировано;
- (65) **ObjZ** – зарезервировано;
- (66) **ObjKlapa** – зарезервировано;
- (67) **ObjPReg** – зарезервировано;
- (123) **ROOT** – см. **Канал CALL.ROOT**;
- (124) **AS_DOCUMENT** – см. **Отчеты АСКУЭ**;
- (125) **AS_DATA** – см. **Отчеты АСКУЭ**;
- (126) **EXCEPT_DAY** – см. **Канал CALL.EXCEPT_DAY**;
- (127) **EMAIL** – см. **Канал CALL.EMAIL**.

Канал CALL.Program

Канал вызова программы – см. **Выполнение программы в реальном времени**.

Канал CALL.SQLQuery

Канал вызова связи с БД – см. **Выполнение SQL-запросов в реальном времени.**

Канал CALL.Document(Report)

Канал генерации документа (отчета) – см. **Использование разработанных шаблонов.**

Канал CALL.Screen

Канал вызова графического экрана – см. **Особенности вызова графического экрана.**

Канал CALL.Panel

Канал вызова графической панели – см. **Графические панели.**

Канал CALL.ChGroupReq

Канал CALL.ChGroupReq выполняет различные функции в зависимости от привязки. Помимо компонентов, указанных ниже, к каналу CALL.ChGroupReq может быть привязана любая системная переменная (см. **Системные переменные TRACE MODE 6**).

Значения аргументов канала CALL.ChGroupReq могут быть отображены с помощью ГЭ **‘Тренд’** (в архивном режиме) или ГЭ **‘Архивный тренд’** – для этого к аргументу кривой должен быть привязан атрибут CALL.ChGroupReq.R. Временной диапазон, в котором размещаются значения, задается левой и правой временными границами тренда со следующим выравниванием:

- диапазон меньше 1 часа – на границу часа;
- диапазон меньше 1 дня – на границу дня;
- диапазон меньше 3 месяцев – на границу дня;
- в остальных случаях – на границу года.

Расстояние между точками равно величине **<диапазон>/<число аргументов>**.

Привязка удаленного канала

К CALL.ChGroupReq может быть привязан канал CALL другого узла:

- если привязан атрибут (0, **R**), запрашиваются атрибуты (0, **R**), (45, **T**) и (59, **FS**), а также число аргументов (число аргументов локаль-

ного CALL.ChGroupReq автоматически изменяется);

- если привязан атрибут (2, **In**), запрос посылается в том случае, если у удаленного канала изменилось число аргументов, атрибут (59, **FS**) или атрибут (252, **dT**), а также в том случае, если атрибут (45, **T**) удаленного канала больше, чем атрибут (45, **T**) локального CALL.ChGroupReq;
- привязка атрибута 140-186 означает смещение по аргументам (число аргументов локального CALL.ChGroupReq не изменяется).

Если конфигурируется запись в аргумент внешнего канала, и к **argN** привязан удаленный аргумент, **argN** должен иметь тип OUTPUT, чтобы запись прошла.

Назначение атрибута **Параметр**:

- 0 – READ/WRITE (читает; если изменить аргумент, то однократно запишет);
- 1 – READ (только читает);
- 2 – WRITE (только пишет);
- 3 – аргументы используются парами: при изменении значения **arg_{2n} INPUT** ($n=0,1,\dots$, аргумент может не иметь привязки) это значение передается удаленному каналу, привязанному к **arg_{2n+1} OUTPUT**.

Привязка MODBUS.R_FIFO_Queue

Функция канала CALL.ChGroupReq, к которому привязан источник 9/100.24 **R_FIFO_Queue** (см. Подтипы **9** и **100**), зависит от атрибута **Параметр**.

Параметр=0

Канал INPUT запрашивает до 32 регистров из массива FIFO_Queue устройства. Полученные данные последовательно записываются в аргументы канала, созданные в ИС (не более 32).

Параметр=1

Канал посылает в контроллер команду MODBUS, сформированную по следующим правилам:

- адрес соответствует адресу, заданному в источнике;
- номер функции соответствует аргументу **arg0**;

Если **arg0**>255, то ожидается, что поле ответа контроллера, в котором указывается число байт данных для приема, содержит два байта вместо одного. Дополнение к подтипу такого канала изменяется с 9/100.24 на 9/100.25.

- номер переменной соответствует заданному в источнике (2-байтное целое);
- последние 2 байта (число запрашиваемых переменных) – нулевые.

Параметр=2

Аналог **Параметр=1**, но в последние два байта записывается значение **arg1**.

Параметр=3

Аналог **Параметр=2**, но к команде добавляются 2 байта, в которые записывается значение **arg2**.

Параметр=4

Аналог **Параметр=3**, но к команде добавляются 2 байта, в которые записывается значение **arg3**.

Параметр<0>: расшифровка ответа

Номер первого байта данных в ответе контроллера при **arg0<255 – 3**, при **arg0>255 – 4** (считая с 0).

Байты данных последовательно записываются в аргументы, следующие за **arg0...arg3** (в зависимости от атрибута **Параметр**):

- в аргумент **SINT** или **USINT** пишется 1 байт данных;

В 1-байтовый приемный аргумент пишется 1 байт данных в 2-байтовом формате, при этом значение старшего байта MPB не контролирует.

- в аргумент **INT** или **UINT** пишется 2 байта данных;
- в аргумент **DINT** или **UDINT** пишется 4 байта данных;
- в аргумент **FLOAT** пишется 4 байта данных в формате FLOAT.

Привязка источника MODBUS чтения или записи

Выполняется соответственно групповое чтение или групповая запись (WORD, FLOAT) при обмене с заданным устройством по MODBUS RTU или MODBUS TCP/IP. Устройство и начальный адрес задаются в источнике, количество считываемых/устанавливаемых параметров определяется числом аргументов канала. В зависимости от дополнения к подтипу привязанного источника/приемника:

- **Rout_Byte, Rin_Byte** (1, 2) – 8 бит в аргументе;
- **Rout_Word, Rin_Word** (3, 4) – регистр в аргументе;

- **Rout_Float, Rin_Float** (8, 9) – FLOAT в аргументе.

Если к CALL.ChGroupReq INPUT привязан источник-команда 16 MOD-BUS INPUT, канал шлет значение своего аргумента, если он изменился.

Привязка источника t11/t12

Полученные данные записываются в аргументы канала, если такая функция поддерживается драйвером.

Привязка локального числового канала

Если к каналу CALL.ChGroupReq (**call**) с **Параметр**&0x80 привязан локальный числовой канал (**ch**), в аргументах **call** создается архив реальных значений канала **ch** с периодом обработки **call**.

Канал CALL.TableFunction

Каналы OUTPUT с этим типом вызова задают табличные функции, используемые FBD-блоками **LINTER** (см. **Раздел 'Алгебраические'**).

Канал CALL.LongFromBits

Функция канала класса CALL с типом вызова 10, **LongFromBits** зависит от атрибута **Параметр**:

- **Параметр** = 0 – формирование битов аппаратного значения по значениям аргументов канала (анализируется до 32 аргументов):
 - бит принимает значение 0, если аргумент, соответствующий ему по номеру, равен 0 или отсутствует;
 - бит принимает значение 1, если аргумент, соответствующий ему по номеру, отличен от 0;
- **Параметр** = 2 – биты, привязанные к аргументам типа OUTPUT, упаковываются (операцией сдвига) в число, записываемое в ближайший аргумент типа INPUT. Количество битов в числе – не больше 32;
- **Параметр** = 6 – если аргументы типа OUTPUT привязаны к атрибутам, объединяющим биты (2, 4 и 8), то упаковка в число производится адекватно привязкам. Число записывается в ближайший аргумент типа INPUT. Количество битов в числе – не больше 32.

Параметр=8-11

Аргументы arg0, arg1 и arg2 канала CALL.LongFromBits должны иметь целочисленный тип данных, после arg0-arg2 следуют группы из 14 аргументов (1 группа на устройство).

К CALL.LongFromBits может быть привязан CALL.String (тексты не задаются), число аргументов CALL.String должно быть равно <число групп из 14 аргументов CALL.LongFromBits> + 1 (arg0 в CALL.String не используется).

В группе из 14 аргументов CALL.LongFromBits:

- arg0 – время хода;
- arg1 – время отлипания или период проверки веса;
- arg2 – длительность импульса;
- arg3 – код команды (1 – ОТКРЫТЬ, 2 – ЗАКРЫТЬ);
- arg4 – дискретный вход состояния устройства (1 – открыт, 2 – закрыт, 0 – выключен, <0 – включен);
- arg5 – дискретный выход (в бите 0 генерируется импульс открытия, в бите 1 – импульс закрытия);
- arg6 – цифровой код статуса устройства, соответствующая строка считывается из файла TM_STATUS_NAME.txt и записывается в привязанный CALL.String. Файл TM_STATUS_NAME.txt создается в папке узла при старте узла. Первые 32 строки файла – для устройства типа «клапан», вторые 32 строки – для устройства типа «мотор». Замечание: коды/строки нормального закрытия и закрытия по ошибкам – разные;
- arg7 – число миллисекунд до конца операции;
- arg8 – тип устройства (0 – клапан, 1 – мотор). Установленные биты первого байта arg8 задают следующие режимы:
 - бит 8 (0x100) – автозакрытие по времени (нет сигнала концевого выключателя);
 - бит 9 (0x200) – автозакрытие при отсутствии отлипания;
 - бит 10 (0x400) – автозакрытие при одновременном наличии сигналов концевых выключателей открытия и закрытия;
 - бит 11 (0x800) – автозакрытие при неизменности веса;
 - бит 12 (0x1000) – команда ОТКРЫТЬ обрабатывается устройством, если предыдущее устройство уже открыто;
 - бит 13 (0x2000) – команда ОТКРЫТЬ обрабатывается устройством, если предыдущее устройство было закрыто, открыто и вновь закрыто без ошибок;
- arg9, DATE_AND_TIME – время перехода устройства в текущее состояние;
- arg10, REAL – текущее значение веса;
- arg11, REAL – уставка веса;
- arg12, REAL – предыдущее значение веса;
- arg13, REAL – зарезервировано.

Клапан определяет свое состояние по концевым выключателям:

- бит 0 arg4 – концевой выключатель открытия;
- бит 1 arg4 – концевой выключатель закрытия.

Мотор определяет свое состояние следующим образом:

- если биты 0 и 1 arg4 равны 0 – выключен;
- если биты 0 и 1 arg4 не равны 0 – включен.

Дополнительные команды (arg3):

- бит 4 (0x10) – если этот бит установлен при подаче любой команды, ошибки сбрасываются;
- 5 – одиночный импульс открытия (текущий статус не проверяется);
- 6 – одиночный импульс закрытия (текущий статус не проверяется);
- бит 5 (0x20) – если установлен, состояние индицируется, а команды не исполняются.

При возникновении ошибки генерируется команда возврата в нормальное состояние (нормальным считается следующее состояние: клапан – закрыт; мотор – выключен).

После успешного открытия из закрытого состояния клапан анализирует весовые параметры:

- если уставка веса не задана (arg11=0), клапан ожидает команду (остается открытым);
- если $arg11 < 0$, клапан сам генерирует команду закрытия при $arg10 \geq arg11$.

Нормально включенный мотор с периодом arg1 проверяет, изменился ли вес:

- нет – генерируется ошибка и мотор выключается;
- да – если $arg10 < arg11$ – нормальная работа, если $arg10 \geq arg11$ – мотор выключается.

Канал CALL.BitsFromLong

Функция канала класса CALL с типом вызова 11, **BitsFromLong** зависит от атрибута **Параметр**:

- **Параметр** = 0 – формирование значений аргументов (0 или 1) по аппаратному значению канала;
- **Параметр** = 2 – если аргумент имеет тип INPUT, то он распаковывается в последующие аргументы OUTPUT (до следующего аргумента типа INPUT);
- **Параметр** = 6 – аналог (10) **LongFromBits** с **Параметр**=6;
- **Параметр** = 1, 3, 7 – аналог **Параметр** = 0, 2, 6 со следующим дополнением: если аргументы INPUT и OUTPUT привязаны к атрибутам каналов и аргумент OUTPUT должен изменить привязан-

ный атрибут, то в привязанный к нему канал копируются атрибуты 45 и достоверность канала, привязанного к соответствующему атрибуту типа INPUT.

Параметр=8, 9

Канал представляет собой группу триггеров (не более 8), выходами которых являются аргументы **arg0...arg7** (в этих аргументах вырабатываются импульсы заданной длительности).

Запускающие импульсы подаются:

- **Параметр=8** – в биты атрибута **In** или **R** (бит 0 – запускающий импульс для **arg0** и т.д.);
- **Параметр=9** – в аргументы **arg8...arg15** (**arg8** – запускающий импульс для **arg0** и т.д.).

Особенности атрибутов **Период** (5, **FRQ**) и **Единица измерения** (38, **FRQ_D**):

- заданное значение периода записывается в байт 2 аппаратного значения (1, **A**) и, совместно с заданной единицей измерения, определяет длительность вырабатываемого импульса;
- в то же время заданная единица измерения определяет период пересчета канала – 1 раз в заданную единицу (т.е. невозможно задать период пересчета канала, равный, например, 5с). Таким образом, период пересчета канала никогда не превышает длительность вырабатываемого импульса.

Длительность запускающего импульса не должна превышать длительность вырабатываемого импульса. Рекомендуется задать длительность запускающего импульса, равную периоду пересчета канала.

При запуске триггера устанавливается соответствующий бит **A** (сбрасывается по заднему фронту вырабатываемого импульса).

Если к **CALL.BitsFromLong** привязан канал (**ch**), **CALL.BitsFromLong** упаковывает значения 8-ми аргументов в число и записывает это число в **ch**.

Канал CALL с типами вызова 12-14

При типах вызова (12) =, (13) < и (14) <= значение нулевого аргумента канала **CALL** сравнивается со значениями всех остальных аргументов, а также анализируются привязки аргументов:

- **Параметр=0** – если хотя бы один результат сравнения истинен, устанавливается бит 4 (0x10) **A**. Если все результаты сравнения истинны, дополнительно устанавливается бит 0 (0x1) **A**;
- значение младшего полубайта атрибута **Параметр** равно 1 – биты 0 и 4 **A** имеют то же назначение. Бит 7 (0x80) **A** устанавливается,

если хотя бы в одном из каналов, привязанных к аргументам, установлен признак недостоверности. Бит 9 (0x200) **A** устанавливается, если хотя бы в одном из привязанных каналов установлен признак программной недостоверности. **R** канала типа INPUT индицирует число каналов, которые удовлетворяют условию. При изменении **R** или **A** устанавливается **b14** (бит 0 атрибута 46) и фиксируется время (атрибут 45);

Параметр = 4, 5

Аргументы создаются группами по 3, в группе сравниваются **arg0** и **arg1**. В **arg2** группы пишется число тактов пересчета, в течение которых результат сравнения был соответственно TRUE или FALSE (если тип данных **arg2** – DATE_AND_TIME, то вместо числа тактов пересчета пишется время в секундах).

Если **Параметр=5**, **arg2** не сбрасывается и ведется только при TRUE.

Если **Параметр=4**, **arg2** сбрасывается при изменении результата сравнения.

В аппаратном значении такого канала каждый бит – это результат сравнения соответствующей группы аргументов.

Канал CALL.Sum

При типе вызова (15) **Sum** аппаратное значение канала CALL формируется из значений его аргументов (при изменении **A** устанавливается **b14** (бит 0 атрибута 46) и фиксируется время (атрибут 45)).

Параметр=0

A равно сумме аргументов.

Параметр=1

A равно сумме значений достоверных каналов, привязанных к аргументам (аргумент без привязки достоверен).

Параметр=2

В таком канале MPV последовательно анализирует аргументы.

Если тип аргумента – INPUT:

- Шаг 1:
 - если к аргументу привязан канал, запрашивается его атрибут b1;
 - если аргумент не имеет привязки (константа), берется эта константа;

- Шаг 2:
 - **qds_and=qds_and & atr61** (начальное значение внутренней переменной **qds_and** – 0xFF)
 - **qds_or=qds_or | atr61** (начальное значение внутренней переменной **qds_or** – 0)

Если тип аргумента – OUTPUT:

- Шаг 1:
 - **set_value = (qds_or & qds_or_mask) | (qds_and & qds_and_mask)**
- Шаг 2:
 - **set_value=set_value & write_mask** (где **write_mask** – одна из масок **qds_set_mask_<..>** (см. ниже), в зависимости от типа канала или аргумента)
 - если к аргументу привязан канал, **set_value** записывается в его атрибут 61
 - если аргумент не имеет привязки (константа), **set_value** записывается в эту константу
- Шаг 3:
 - сбрасывается **qds_and**
 - сбрасывается **qds_or**

Необходимые для вычислений параметры задаются с помощью следующих ключей в файле *.cnf (все параметры задаются в формате HEX):

- **MASK_QDS_AND=qds_and_mask**
 - **MASK_QDS_OR=qds_or_mask**

 - **MASK_QDS_SET_FLOAT=qds_set_mask_f (*)**
 - **MASK_QDS_SET_HEX=qds_set_mask_hex (*)**
 - **MASK_QDS_SET_DEC=qds_set_mask_dec (*)**
 - **MASK_QDS_SET_EVT=qds_set_mask_ev (*)**
 - **MASK_QDS_SET_ONE=qds_set_mask_one (*)**
 - **MASK_QDS_SET_DEF=qds_set_mask_def (*)**
- (*) Значение масок **qds_set_mask_<..>** по умолчанию – 0xF0.

Привязка «(CALL.Sum с **Параметр=2**).arg – (другой CALL).arg – канал» равнозначна привязке «(CALL.Sum с **Параметр=2**).arg – канал».

Параметр=3

A равно результату побитового логического сложения аргументов (побитовое OR).

Параметр=4

A равно результату побитового логического умножения аргументов (побитовое AND).

Параметр=5

К каналу **CALL.Sum** (далее – **Sum5**) должен быть привязан канал (далее – **result**), аргумент **Sum5.arg0** должен иметь тип данных DATE_AND_TIME, к последующим аргументам привязываются каналы (**ch_i**). **Sum5** суммирует значения каналов **ch_i** и записывает результат в **result** в том случае, если **ch_i.T >= Sum5.arg0** для любого **i**.

Параметр=6

Аналог предыдущего, но обрабатывает только в том случае, если у всех суммируемых каналов **DataReady=1**.

Параметр=8

CALL.Sum с Параметр=8 (далее – **Sum8**) используется для алгебраического сложения векторов:

- к аргументам **Sum8** должны быть привязаны каналы класса CALL (**call_i**) (привязка к 0, **R** – сложение, к 9, **Q** – вычитание);
- к **Sum8** должен быть привязан канал класса CALL (**call_result**).

В указанной конфигурации аргумент 0 **call_result** равен сумме аргументов 0 каналов **call_i** и т.д. Операция выполняется корректно также в случае, если к аргументам **Sum8** привязаны индивидуальные архивы **CALL.LArc0** или **CALL.LArc1**, сконфигурированные идентично (период пересчета, число аргументов и т.п.). Аргументы, имеющие временной тип данных, игнорируются при суммировании (только копируются).

Параметр=9-14

CALL.Sum с Параметр=9-12 (далее – **Sum9-Sum12**) используется для алгебраического сложения векторов.

Конфигурация **Sum10** и **Sum12**:

- **arg0** должен иметь тип данных DATE_AND_TIME;
- **arg1** и **arg2** должны иметь тип OUTPUT и должны быть привязаны к каналам CALL.ChGroupReq (далее соответственно **ChGroupReq1** и **ChGroupReq2**);
- **arg3** может быть OUTPUT с привязкой к CALL.ChGroupReq (далее соответственно **ChGroupReq3**) или иметь тип INPUT;
- последующие аргументы должны иметь тип INPUT;
- к **Sum10** (**Sum12**) должен быть привязан канал

CALL.ChGroupReq или CALL.TVC (**call_result**).

В этой конфигурации суммируются векторы (CALL.ChGroupReq или CALL.TVC), начиная с **arg3** (**arg4**).

В аргументы **ChGroupReq2** записывается разность между соответствующими аргументами **ChGroupReq1** и **call_result**.

В аргументы **ChGroupReq3** (при его наличии) записываются величины $0.01 * (\text{ChGroupReq1.arg}_k - \text{call_result.arg}_k) / \text{ChGroupReq1.arg}_k$.

Правила суммирования для **Sum9-Sum12**:

- если к аргументу привязан канал, отключенный от источника (**W=1**), он игнорируется;
- если хотя бы один привязанный канал имеет аппаратную недостоверность, флаг аппаратной недостоверности устанавливается и в **CALL.Sum**;
- **CALL.Sum** обрабатывает по времени (когда **45,T** всех суммируемых каналов больше или равно **CALL.Sum.arg0**);
- **CALL.Sum** работает с числовыми аргументами, кроме временных;
- если у аргумента нет привязки, суммируется его значение по умолчанию;
- если привязка к **0,R** канала CALL.ChGroupReq/CALL.TVC, выполняется сложение, если к **9,Q** – вычитание;
- результат суммирования записывается в аргументы **call_result**;
- после отработки в каналах **CALL.Sum** и **call_result** устанавливаются параметры **b14**, **Response** и **DataReady**, а также фиксируется время (атрибут **45,T**). Те же переменные устанавливаются в каналах **ChGroupReq2** и **ChGroupReq3** после успешной записи в их аргументы;
- функции, связанные с **arg1-arg3**, обрабатываются по времени (когда **45,T** канала **ChGroupReq1** больше или равно **CALL.Sum.arg0**);
- каналы-результаты в реальном времени динамически корректируют количество своих аргументов по количеству аргументов векторов, участвующих в операции;
- если в качестве **call_result** используется CALL.TVC, а среди суммируемых каналов нет ни одного CALL.TVC, то в **call_result** устанавливаются **arg0=T_{текущее} - <число точек в call_result>**с и **arg1 = T_{текущее}**;
- если в качестве **call_result** используется CALL.TVC, то суммируются (**call_result.Параметр+1**) точек кривых (времена точек не анализируются). Для CALL.ChGroupReq номер кривой равен 0;
- если в качестве **call_result** используется CALL.ChGroupReq, среди суммируемых каналов должен быть хотя бы один канал CALL.ChGroupReq;

- если к аргументам **CALL.Sum** привязаны атрибуты 0, **R** каналов **CALL.TVC** и в качестве **call_result** используется **CALL.ChGroupReq**, то в **call_result.arg0** записывается результат суммирования **arg0** всех суммируемых **CALL.ChGroupReq** и первых точек всех кривых всех суммируемых **CALL.TVC**, в **call_result.arg1** записывается результат суммирования **arg1** всех суммируемых **CALL.ChGroupReq** и вторых точек всех кривых всех суммируемых **CALL.TVC** и т.д.;
- если к аргументам **CALL.Sum** привязаны атрибуты **x=142, 143...** каналов **CALL.TVC** и в качестве **call_result** используется **CALL.ChGroupReq**, то в **call_result.arg0** записывается результат суммирования **arg0** всех суммируемых **CALL.ChGroupReq** и первых точек кривых (**x-142**) всех суммируемых **CALL.TVC**, в **call_result.arg1** записывается результат суммирования **arg1** всех суммируемых **CALL.ChGroupReq** и вторых точек кривых (**x-142**) всех суммируемых **CALL.TVC** и т.д.

Sum9 (Sum11) – это упрощенный сумматор (нет анализа **arg0**, нет функций, связанных с **arg1-arg3**, все аргументы – INPUT).

CALL.Sum с **Параметр=13, 14** аналогичен **Sum10 (Sum12)**, но обрабатывает только в том случае, если у всех суммируемых каналов **Data-Ready=1**.

Если **CALL.Sum.Параметр=11, 12, 14**, то **arg0=T_{текущее} + 1с**.

В канале **CALL.Sum** векторного суммирования атрибут 120, **ACK** принимает значение 1 после того, как этот атрибут принимает значение 1 во всех суммируемых каналах-векторах.

Параметр=24-26

CALL.Sum с **Параметр=24 (0x18)** – аналог **Sum8**. Число аргументов **call_result** автоматически равно минимальному числу аргументов среди суммируемых **call_i**, **call_result.A** равно сумме аргументов **call_result**.

CALL.Sum с **Параметр=25 (0x19)** – аналог **CALL.Sum** с **Параметр=24 (0x18)**, но **call_i** должны быть **CALL.ChGroupReq**, и **call_i.A** равно сумме аргументов этого **call_i**.

CALL.Sum с **Параметр=26 (0x1A)** – аналог **CALL.Sum** с **Параметр=25 (0x19)**, но суммирует/вычитает разности **arg_i-arg_{i-1}** (при **i=0** суммируются сами аргументы). Далее такие суммы обозначены **sum_main**. Величины, записываемые в **CALL.ChGroupReq**, привязанные к аргументам **call_result** (см. прим. ниже), зависят от аргумента:

- **arg0** – **sum_main**;
- **arg1** – если есть каналы, у которых к аргументу **CALL.Sum** привязан атрибут 9, **Q**, – суммы разностей аргументов каналов, у которых к аргументу **CALL.Sum** привязан атрибут 0, **R (sumR)**;: в про-

тивном случае – ;

- **arg2** – если есть каналы, у которых к аргументу **CALL.Sum** привязан атрибут **0,R**, – сумма разностей каналов, у которых к **CALL.Sum** привязан атрибут **9,Q (sumQ)**; в противном случае – ;
- **arg3** – $100 * (\text{sumQ} - \text{sumR}) / \text{sumQ}$.

Если аргумент **CALL.Sum** типа OUTPUT, то в такую привязку пишется текущая **sumQ** и обнуляется.

Если в **CALL.Sum** установлен флаг **Запрос времени значения** (атрибут 50), то суммирование контролирует атрибут 59.

Если к **CALL.Sum** с **Параметр=24-26** привязан атрибут **call_result.124 (ArgSize, число аргументов)**, то результаты записываются в каналы **CALL.ChGroupReq**, привязанные к аргументам **call_result**.

Канал **CALL.Sum**, в атрибуте **Параметр** которого установлен бит 5, может использоваться на первом уровне канала **CALL.ROOT**.

Канал CALL.Set

При этом типе вызова реальное значение канала (0..65535) присваивается всем его аргументам (тип аргументов должен быть OUT или IN/OUT, тип данных – любой целочисленный). При наличии аргумента типа IN его значение присваивается последующим аргументам типа OUT или IN/OUT.

Канал CALL.RT_Statistics

Параметр=0

При **Параметр=0** канал **RT_Statistics** выполняет статистическую обработку значений каналов. Для корректной работы канала CALL с этим типом вызова его аргументы создаются группами по 6. К первому по порядку аргументу типа IN группы привязывается анализируемый канал (**ch1**), в последующие 5 аргументов типа OUT или IN/OUT при отработке канала CALL записываются результаты статистической обработки:

- сумма значений канала **ch1**;
- число суммирований;
- среднее значение канала **ch1**;
- максимальное значение канала **ch1**;
- минимальное значение канала **ch1**.

Для сброса результата статистической обработки можно присвоить 0 соответствующему аргументу или установить соответствующий бит значения канала **RT_Statistics**:

- бит 0 – сброс суммы и числа суммирований;
- бит 1 – сброс максимального значения;
- бит 2 – сброс минимального значения.

Параметр & 0x20

Такой канал **RT_Statistics** работает по астрономическому времени.

Помимо бита 5, в атрибуте **Параметр** канала **RT_Statistics** устанавливаются биты 0-3, задающие интервал (аналогично каналам выборки из архива – см. **Временной интервал выборки**).

К каналу **RT_Statistics** привязывается локальный **ChGroupReq**, содержащий список обрабатываемых каналов.

Если в канале **RT_Statistics** установлен флаг **Запрос времени значения** (атрибут 50), интервал выравнивается в зависимости от его величины.

На каждый из обрабатываемых каналов создается 2 аргумента.

Значения аргументов зависят от значения первого полубайта атрибута **Параметр (Параметр & 0xF0)**:

- 2 – первый аргумент – текущая сумма, второй – среднее за интервал;
- 6 – первый аргумент – текущая сумма, второй – сумма за интервал;
- 10 – первый аргумент – последнее значение предыдущего интервала, второй – последнее значение интервала, предшествующего предыдущему;
- 14 – первый аргумент – последнее значение предыдущего интервала, второй – разность между последними значениями предыдущего интервала и интервала, предшествующего предыдущему.

Второй аргумент указывает направление передачи результата обработки:

- если к аргументу привязан **ChGroupReq** или **TVC**, то:
 - если флаг **Запрос времени значения** в **RT_Statistics** не установлен, результаты пишутся в конец массива аргументов в режиме стека (сдвиг вверх и запись нового значения в последний аргумент);
 - если флаг **Запрос времени значения** в **RT_Statistics** установлен, аргумент для записи нового значения подбирается по времени;
- к аргументам можно привязать атрибуты 142 и т.д. одного и того же канала **TVC**, настроенного на отображение нескольких кривых, в этом случае результаты формируют эти кривые.

Если к каналу-результату привязан канал CALL.ChGroupReq (**cgr**), то перед сбросом аргументов канала-результата (по истечении интервала) их значения записываются в аргументы **cgr**. Если к каналу-результату привязан числовой канал, в него записывается очередное значение. Чтобы этот механизм работал, нужно установить флаг архивирования в канале **RT_Statistics**.

Канал CALL.MOVE

- **Параметр=0** – присвоение значения аргумента типа IN всем последующим аргументам типа OUT или IN/OUT;
- **Параметр=1** – то же, что **Параметр=0**, но действие происходит, если был встречен аргумент типа DATE_AND_TIME, и значение этого аргумента меньше, чем текущее время, и больше, чем атрибут 45 канала CALL.MOVE;
- **Параметр=4**:
 - если к аргументу типа IN привязан атрибут 0, **R** канала CALL.ChGroupReq (**call_1**), а к последующему аргументу типа OUT или IN/OUT привязан атрибут 0, **R** другого канала CALL.ChGroupReq (**call_2**), то значения аргументов **call_1** копируются в соответствующие аргументы **call_2**;
 - если к аргументу типа IN привязан атрибут 45, **T** канала CALL.TVC (**tvс_1**), а к последующему аргументу типа OUT или IN/OUT привязан атрибут 45, **T** другого канала CALL.TVC (**tvс_2**), то все метки времени из **tvс_1** копируются в **tvс_2**;
 - если к аргументу типа IN привязан атрибут **m** (**m**=142, 143...) канала CALL.TVC (**tvс_1**), а к последующему аргументу типа OUT или IN/OUT привязан атрибут **k** (**k**=142, 143...) другого канала CALL.TVC (**tvс_2**), то все аргументы, задающие кривую (**m**-142) в канале **tvс_1**, копируются в аргументы, задающие кривую (**k**-142) в канале **tvс_2**;
- **Параметр=5** – то же, что **Параметр=4**, но действие выполняется, если у всех каналов-источников, привязанных к аргументам типа IN канала CALL.MOVE, атрибут 45 больше, чем атрибут 45 канала CALL.MOVE;
- **Параметр=6** – то же, что **Параметр=4**, но действие выполняется, если у всех каналов-источников, привязанных к аргументам типа IN канала CALL.MOVE, установлен флаг **DataReady** (атрибут 120, **АСК**);
- **Параметр=8,9** – записывает в аргумент OUT результат конкатенации строковых представлений предшествующих аргументов IN (таких блоков аргументов может быть много). Аргументы IN должны иметь тип данных STRING и не иметь привязки к CALL.STRING, в противном случае конкатенация производится

следующим образом: берется бинарное представление привязки аргумента, и в конкатенации участвуют данные двух младших байтов. Если значение младшего байта атрибута **Параметр** равно 9, старшие 2 байта атрибута задают разделитель:

- 0 – пробел;
- 1 – «=»;
- 2 – «;»;
- 3 – «\».

Если к аргументу OUT привязан атрибут (140, **ARG00**) или (0, **R**) канала CALL, отличного от CALL.STRING, коды символов итоговой строки записывается в аргументы CALL следующим образом:

- в аргумент STRING – вся строка;
- в аргумент SINT/USINT – 1 символ;
- в аргумент INT/UINT – 2 символа;
- в аргумент другого типа данных – 4 символа.

Если привязан атрибут (141, **ARG01**), (142, **ARG02**) и т.д., в аргументы записывается итоговая строка, начиная с ее символа 1, 2 и т.д.

Канал CALL.MResource_1

Этот канал используется при решении задач T-FACTORY (см. **Канал CALL с типом вызова MRESOURCE_1**).

Каналы CALL.LArc0 и CALL.LArc1

См. **Индивидуальный архив**.

Канал CALL.DifSnap

См. **Дифференциальный срез локального архива**.

Канал CALL.LocalStatistics

См. **Обработка данных локального архива по каналу**.

Канал CALL.LocalQuick

См. **Быстрая выборка данных из локального архива**.

Канал CALL.LocalList

См. Выборка данных из локального архива по каналу.

Канал CALL.LocalQuick-T

См. Расширенная быстрая выборка данных из локального архива.

Канал CALL.MUX

Возможные конфигурации канала CALL с типом вызова (33) **MUX** (**p**=**Параметр**, тип всех аргументов – INPUT):

- число аргументов больше (**p**+1). В этом случае первые **p** аргументов (**arg₀-arg_{p-1}**) образуют первую группу, остальные (2 или более) – вторую. Аргументы первой группы могут иметь обычную или виртуальную привязку, а также не иметь привязки. К аргументам второй группы привязываются каналы (в том числе удаленные). В реальном времени:
 - все аргументы первой группы привязываются к каналу (**ch**), который привязан к аргументу с порядковым номером **CALL.MUX.R**<>0 во второй группе (производится привязка к атрибуту, заданному первоначальной привязкой, или к (0, **R**), если привязка не была задана);
 - канал **CALL.MUX** принимает имя и кодировку канала **ch**;

Если **CALL.MUX.R** равно 0 или задает номер несуществующего аргумента, **CALL.MUX.NAME**=«?».

- число аргументов равно (**p**+1). В этом случае первые **p** аргументов (**arg₀-arg_{p-1}**) – перепривязываемые, а последний аргумент (**arg_p**) создается для служебных целей. В реальном времени:
 - при присвоении имени некоторого локального канала атрибуту (127, **NAME**) канала **CALL.MUX** аргументы **arg₀-arg_{p-1}** привязываются к указанному каналу, при этом <имя **CALL.MUX**>=<имя канала>. Если указанный канал не существует, **CALL.MUX.NAME**=«?»;
 - при каждом изменении значения **CALL.MUX** (кроме изменения на 0) МРВ ищет в локальной базе каналов очередной канал, первые символы кодировки которого совпадают с кодировкой **CALL.MUX**, и привязывает аргументы **arg₀...arg_{p-1}** к найденному каналу. При этом <имя **CALL.MUX**>=<имя канала>, а в **arg_p** записывается номер следующего для привязки канала в массиве каналов, кодировка которых совпадает с кодировкой **CALL.MUX**. При изменении значения

CALL.MUX на 0 **arg_p** сохраняет свое значение, **arg₀=arg₁=...=arg_{p-1}=0** и **CALL.MUX.NAME=«?»**.

Каналы **CALL.RemArc0** и **CALL.RemArc1**

См. Запрос удаленного индивидуального архива.

Канал **CALL.AsyncCollection**

Если **Параметр=0**: однократная последовательная обработка каналов **CALL** типа **OUTPUT**, привязанных к аргументам данного канала. Целое значение, посланное в канал **AsyncCollection**, задает номер привязки (считая с 1), с которой нужно начинать последовательную обработку (привязки с меньшими номерами не обрабатываются). Переход к обработке следующей привязки производится после того, как автоматически сбросится атрибут 39, **EXEC** привязки, обрабатываемой в данный момент. Значение **AsyncCollection** индицирует номер обрабатываемой привязки; после окончания цикла значение **AsyncCollection** автоматически сбрасывается в 0.

Если **Параметр=0**:

- если некоторый целочисленный аргумент **argN INPUT** не привязан, его значение передается каналу **CALL**, привязанному к следующему за **argN** аргументу **OUTPUT**. Это необходимо, например, при каскадном запуске **CALL.LocalStatistic**. При этом проверяется b14 (бит 0 атрибута 46);
- если аргумент **OUTPUT** привязан к атрибуту (120, **ACK**) канала **CALL**, то проверяется этот атрибут;
- если аргумент **OUTPUT** привязан к атрибуту (45, **T**) канала **CALL**, то проверяется **CALL.45 > AsyncCollection.45**;
- для каждой проверки существует таймаут по умолчанию, равный примерно 180с. После этого в **AsyncCollection** устанавливается признак аппаратной недостоверности, а в **AsyncCollection.R** сохраняется номер последнего обрабатываемого аргумента;
- для задания таймаута (в мс) используется аппаратное значение **AsyncCollection**. Если **AsyncCollection.2 = 0**, при обработке **AsyncCollection.2 = 180000**;
- если аргумент **OUTPUT** привязан к атрибуту (9, **Q**) канала **CALL**, то этот канал **CALL** запускается и далее осуществляется переход к следующему аргументу;
- аргументы могут быть привязаны не только к каналам **CALL** – например, к числовым каналам **INPUT**, связанным с источниками;
- если у привязанного канала (в т.ч. у канала **CALL**) для периода задано значение **однократно**, проверяется атрибут 8, **W**.

Если **Параметр<>0**, **ARG₀** индицирует время нахождения на текущем

шаге (тип данных REAL – в виде «с.мс», UDINT – число миллисекунд), остальные аргументы (ARG_N) привязываются к каналам CALL или не имеют привязки, AsyncCollection.R индицирует номер начального, конечного, текущего или следующего шага, атрибут 54 – время (мс) процедуры от запуска до достижения последнего шага.

Если аргумент ARG_N не имеет привязки:

- тип данных – BOOL:
 - $ARG_N = 0$ – выход;
 - $ARG_N = 1$ – переход к ARG_{N+1} ;
- тип данных – целочисленный:
 - $ARG_N < 0$ – выход; если тип аргумента OUTPUT, номер шага уменьшается на 1;
 - $ARG_N = 0$ – переход к ARG_{N+1} ;
- тип данных – REAL:
 - $ARG_N = 0$ – переход к ARG_{N+1} ;
 - $ARG_N < 0$ – если предыдущий аргумент привязан к CALL, номер шага уменьшается на 1 и выход, иначе – переход к ARG_{N+1} ;
- тип данных – DATE_AND_TIME: если текущее время больше ARG_N – переход к ARG_{N+1} , иначе – выход;
- тип данных – DATE или TIME: если $ARG_0 > ARG_N$ (мс) – переход к ARG_{N+1} , иначе – выход.

Если аргумент ARG_N привязан к атрибуту канала CALL (кроме R и EXEC) или каналу другого класса:

- $ARG_N = 0$ – выход;
- $ARG_N < 0$ или канал выключен – переход к ARG_{N+1} ;

Если аргумент ARG_N привязан к атрибуту R или EXEC канала CALL OUTPUT:

- если канал включен ($CALL.3 = 0$), он обрабатывается, иначе $b14=1$ (см. описание атрибута 46 в разделе **Атрибуты каналов, отображаемые профайлером**);
- если привязка к EXEC – выход с увеличением номера шага на 1;
- если привязка к R – переход к ARG_{N+1} ;
- если следующий аргумент привязан к атрибуту 46 этого канала CALL, то его ненулевое значение будет соответствовать завершению функции CALL.

Канал CALL.Writer

См. **Принудительная запись в SIAD.**

Канал CALL.DI_DO

Канал CALL с типом вызова (53) **DI_DO** предназначен для быстрого переключения статуса узла, имеющего резерв, при прямой связи резервов.

Бит 1 атрибута ПАРАМЕТР равен 0

В канале создаются два аргумента.

Аргумент **arg0** предназначен для мониторинга статуса резервного узла. **arg0** = 0 означает, что резерв не работает, и через 2 такта данный узел переводится в состояние WORK. **arg0** = 1 означает, что резерв работает, и данный узел переводится в состояние TRACE.

Если узел находится в состоянии WORK, **arg1** = 1, в противном случае **arg1** = 0.

Одна из возможных схем связи для реализации механизма:

```
node1.CALL.arg0-node1.DI-node2.DO-node2.CALL.arg1
node1.CALL.arg1-node1.DO-node2.DI-node2.CALL.arg0
```

Бит 1 атрибута ПАРАМЕТР равен 1

В канале создаются 2 или 4 аргумента, в обоих случаях **arg0** предназначен для мониторинга статуса резервного узла, бит 0 **arg1** равен 0 (в состоянии TRACE) или 1 (в состоянии WORK), а в бите 1 **arg1** генерируется меандр (как в состоянии TRACE, так и в состоянии WORK, ширина импульсов – 1 такт).

В канале CALL.DI_DO, анализирующем меандр, атрибут 98, **C5** представляет собой счетчик пропавших подряд импульсов (при появлении первого импульса **C5**=0). Если меандр отсутствует, узел TRACE переходит в состояние WORK

- спустя 2 такта, если в канале 2 аргумента;
- спустя **arg2** тактов, если в канале 4 аргумента (значение **arg2** по умолчанию – 2).

В отсутствие сигнала от узла TRACE, узел WORK сбрасывает его статус в 0 спустя **arg3** тактов (значение **arg3** по умолчанию – 4). Если впоследствии сигнал появляется, узел WORK восстанавливает статус узла TRACE только после получения извещения по сети (в этой ситуации узел WORK не восстанавливает статус узла TRACE по прямому соединению).

При наличии прямого соединения CALL.DI_DO.R=2 в узле WORK и CALL.DI_DO.R=3 в узле TRACE.

Общие замечания

При старте МРВ входит в режим прослушивания сети, после чего, для

любого CALL.DI_DO, читает DI:

- если 0, МРВ в дальнейшем с прямым соединением не работает (работает через сеть);
- если не 0, МРВ в течение 6 полутактов анализирует сигнал. Если меандр детектируется, узел переключается в состояние TRACE, в противном случае в дальнейшем МРВ с прямым соединением не работает (работает через сеть).

При старте МРВ счетчик пропавших импульсов (**C5**) не работает.

Если бит 0 атрибута **Параметр** равен 0, МРВ читает и пишет непосредственно через порты («поверх» каналов), в противном случае прямая процедура чтения/записи через порты не работает, и МРВ работает через каналы, в связи с чем:

- если бит 0 атрибута **Параметр** равен 0, то **arg0** может быть связан только с каналом **DI**, а **arg1** – только с каналом **DO**;
- если бит 0 атрибута **Параметр** равен 1, то **arg0** и **arg1** могут быть связаны с произвольными каналами или не иметь привязки.

Если канал CALL.DI_DO выключен, нет чтения/записи и нет перехода узла в другое состояние.

Конфигурация прямого соединения для контроллеров Silian

Кабель прямого соединения:

```
Silian0.DO1-Silian1.DI1
Silian0.DO2-Silian1.DI2
Silian0.DI1-Silian1.DO1
Silian0.DI2-Silian1.DO2
Silian0.GND-Silian1.GND
```

Канал чтения – OEM_DI_8 (2.6), порт 304.

Канал записи – OEM_DO_8 (2.22), порт 302.

Атрибут **Параметр** канала CALL.DI_DO:

- бит 0 равен 0;
- бит 1 равен 1.

Канал CALL.MLink

Канал CALL с этим типом вызова в основном предназначен для обмена с мониторами версии 5. Если тип канала – INPUT, он запрашивает по M-LINK **N** значений (**N** – число аргументов канала CALL) у узла с номером, заданным атрибутом **Параметр** канала CALL, по каналу с индексом, заданным значением канала CALL. Полученные значения записываются в аргументы канала CALL.

Если канал CALL имеет тип OUTPUT, в нем может быть создано до 16 пар аргументов. Первый аргумент пары задает индекс канала, в который требуется передать по M-LINK значение, заданное вторым аргументом пары. Номер бита значения канала CALL, равный 1, задает номер пары для отработки. При отработке канала CALL обрабатывается одна пара, заданная самым младшим битом из всех, равных 1, после чего этот бит сбрасывается.

Канал CALL.EvReg

См. **Регистратор аварийных событий**.

Канал CALL.Exec

Канал обеспечивает запуск приложения Windows при выполнении следующих условий:

- значение канала отлично от 0;
- в канале создан хотя бы один аргумент.

Если **Параметр** = 0, имя канала задает имя exe-файла, атрибут 80, **CMNT** – путь к файлу.

Если **Параметр** = 1, имя канала задает путь и имя exe-файла, атрибут 80, **CMNT** – параметры запуска.

Канал CALL.TVC

Канал CALL с типом вызова (59) **TVC** задает (**Параметр**+1) табличных функций $x(t)$.

Аргументы **arg0** и **arg1** создаются для служебных целей.

Последующие аргументы создаются группами, каждая из которых содержит (**Параметр**+2) аргументов. Первые (**Параметр**+1) аргументов группы задают значения соответственно первой, второй и т.д. функций на момент времени, заданный последним аргументом группы.

Табличные функции CALL.TVC могут быть отображены с помощью ГЭ **Тренд** (в архивном режиме) или ГЭ **Архивный тренд**. Для отображения первой (второй и т.д.) функции к аргументу тренда нужно привязать атрибут 142, **ARG02** (143, **ARG03** и т.д.) канала CALL.TVC. Для отображения первой функции к аргументу тренда можно также привязать атрибут 0, **R** канала CALL.TVC. Аргументы **arg0** и **arg1** канала CALL.TVC задают временной диапазон отображения тренда; если эти аргументы равны 0 (не заданы), используется диапазон (**T_{текущее}**-1год, **T_{текущее}**+1год).

Если значения временных аргументов групп меньше числа секунд в сутках (86400), кривые на тренде будут повторяться каждые сутки.

К каналу CALL.TVC (**tvс**) с **Параметр=0** может быть привязан удаленный канал класса CALL (**rem**) при связи по сети или RS. При такой конфигурации в аргументы **tvс** копируются значения аргументов **rem**.

Для некоторых драйверов поддерживается передача полученных данных в канал CALL.TVC – для этого к каналу нужно привязать соответствующий источник.

К каналу CALL.TVC с **Параметр=0** могут быть привязаны системные переменные TRACE MODE (см. **Системные переменные TRACE MODE 6**) со стековым накоплением их информации формата UINT (FLOAT).

Канал CALL.TVC с **Параметр=0** может быть записан в SIAD. При этом атрибут CALL.TVC.41 указывает, какой канал архивировать (допускается ввод ID или имени канала в атрибут).

Канал CALL.TVC с **Параметр=0**, к которому привязан локальный CALL.ChGroupReq, формирует кривую с f(аргументы ChGroupReq) – сумма или среднее. При этом метки времени кривой формируются по текущему времени.

Канал CALL.TVC как архив

Если к каналу CALL.TVC (**call**) с **Параметр=0** привязан локальный числовой канал (**ch**), в аргументах **call** создается архив канала **ch**, формируемый по изменению атрибутов (0, **R**), (45, **T**), (8, **W**) и (3, **C**) канала **ch**. Если флаг **Запрос времени значения** в **call** не установлен, метки времени для архивных значений формируются по **call.45**, в противном случае запрашиваются у **ch**.

Канал CALL.TVC как расписание

Расписание может быть задано с помощью канала CALL.TVC следующей конфигурации:

- в канале не задана привязка;
- начальное значение задает тип расписания:
 - 1 – расписание на сутки (далее – **TVCday**; в таком канале может быть произвольное число групп аргументов после arg0 и arg1);
 - 7 – расписание на неделю (далее – **TVCweek**; минимальное число групп аргументов после arg0 и arg1 – 7).

Дополнение к подтипу **TVCday** и **TVCweek** – 77; arg0 и arg1 – соответственно левая и правая границы временного диапазона.

Для детализации недельного расписания к аргументам групп канала **TVCweek** нужно привязать каналы **TVCday**. Такие **TVCday** должны со-

держат одно и то же число кривых. В MPB атрибут (97, **C4**) привязанного **TVCday** индицирует день недели. Значение из детализированного недельного расписания выводится в обычный канал, к которому привязан атрибут 142...145 (не более четырех кривых) канала **TVCweek**.

Каналы **TVCday** и **TVCweek** (в т.ч. детализированный) выводятся на тренд аналогично CALL.TVC табличных функций.

По умолчанию **TVCweek** настраивается на текущую неделю. Если флаг **Запрос времени значения** установлен, **TVCweek** перенастраивается на следующие недели автоматически.

Значения 1 и 2 атрибута (0, **R**) канала **TVCday** или **TVCweek** (в т.ч. детализированного) являются командами изменения временного диапазона (в т.ч. в привязанных **TVCday**):

- 1 – следующий диапазон;
- 2 – предыдущий диапазон.

Для вывода **TVCday** и **TVCweek** в табличном виде используется **Sub-Num=544-549**.

После описанных выше групп аргументов в **TVCweek** может быть создано произвольное число аналогичных вспомогательных групп, в которых аргументы (за исключением временной метки) должны иметь тип данных BOOL. К этим аргументам привязываются каналы **TVCday**. Если временная метка основной группы совпадает с временной меткой вспомогательной группы, вместо основной обрабатывается вспомогательная группа. Этот механизм позволяет задавать расписание особых дней.

Канал CALL.OtherProj

Канал CALL с типом вызова 60, **OtherProj** обеспечивает запись/чтение переменных произвольного узла произвольного проекта. Атрибут **Комментарий** (80, **CMNT**) канала CALL.OtherProj задает IP-адрес узла, атрибут 92, **I2** – номер узла (в ИС номер узла задается как начальное значение канала). Функция канала CALL.OtherProj зависит от его атрибута **Параметр**:

- **Параметр=0** или **2** – соответственно чтение **0,R** или запись **2,In** каналов указанного узла; аргументы CALL.OtherProj создаются парами (первый аргумент пары – ID канала, второй – значение). Для отработки канала записи нужно присвоить ему значение, каждый бит которого указывает порядковый номер пары аргументов (начиная с 1), которую нужно отработать; после отработки значение канала сбрасывается в 0 автоматически;
- **Параметр=1** или **3** – соответственно чтение или запись аргументов канала CALL указанного узла; **arg0** канала CALL.OtherProj задает ID удаленного канала CALL, остальные аргументы – данные. Для отработки канала записи нужно изменить его значение; после

отработки значение канала сбрасывается в 0 автоматически.

Число одновременно запрашиваемых/записываемых значений не может превышать 64.

При создании запроса значение атрибута (1,**A**) канала CALL.OtherProj увеличивается на 1; при получении ответа от указанного узла – сбрасывается в 0. Если **A** превышает значение, заданное параметром **Предельное время ожидания ответа на запрос (в циклах)** (на вкладке **Дополнительно** редактора узла), в канале CALL.OtherProj устанавливается признак аппаратной недостоверности.

Канал CALL.STRING

См. **Запись длинных строк в канал CALL.**

Канал CALL.Vector

Канал CALL с типом вызова (62) **Vector** предназначен для работы с векторами, он хранит в аргументах данные с единой меткой времени.

Канал CALL.Vector аналогичен каналу CALL.ChGroupReq и ему подобным, но имеет ограничения по привязке.

Метка времени вектора хранится в атрибуте (59, **DR**).

Младшие 4 бита атрибута **Параметр** задают временной интервал аналогично разбиению (T_FROM, T_TO) на интервалы в каналах выборки из архива (см. **Временной интервал выборки**).

Если в канале CALL.Vector установлен флаг архивирования (номер SIAD не имеет значения), то он сохраняет вектор (аргументы) с меткой времени в собственный архив.

Информация по лицензионным ограничениям:

- если в канале CALL.Vector не установлен флаг архивирования, он считается как 2 канала;
- если в канале CALL.Vector установлен флаг архивирования и канал связан с источником/приемником:
 - если мощность МРВ более 1024 каналов, канал CALL.Vector считается как 2 канала;
 - если мощность МРВ менее 1024 каналов, каждый аргумент канала CALL.Vector считается как канал.

В строке **Дополнительно** источника/приемника, привязанного к каналу CALL.Vector, могут быть заданы следующие ключи (разделитель ключей – точка с запятой):

- **TSXEMA**=<0...15> – задает распределение данных по аргументам (по умолчанию – 0). Для использования ключа в папке узла вруч-

ную должен быть создан файл **tsxema<n>.txt** (**n** – двухразрядное число 00...15), содержащий строки. Формат строк:

<номер аргумента> <номер переменной>

- **TSHHDR**=<0...8> – схема вывода вектора (группы векторов) с помощью **SubNum**. Для использования ключа в папке узла вручную должен быть создан файл **TABLSXEMA<n>.txt** (**n** – двухразрядное число 00...08), содержащий строки. Формат строк:

<номер поля> <наименование столбца> <единицы измерения> <формат> CALLMASK=<число HEX без префикса «0x»>

Если поле **Наименование столбца** или **Единицы измерения** содержит знак «@», то заголовок соответствующего столбца выводимой таблицы будет пуст.

В поле **Формат** могут быть заданы следующие значения (аналогичны значениям, задаваемым в атрибуте (6, **D**) канала **FLOAT** – см. **Атрибуты каналов, отображаемые профайлером**):

- 0 – %g;
- 1 – %.1f;
- 2 – %.2f;
- 3 – %.3f;
- 4 – %.4f;
- 5 – %8.3f;
- 6 – %8.2f;
- 7 – %3e;

Установленные биты значения **CALLMASK** задают следующие функции:

- бит 0 – не выводить имя канала;
- бит 1 – не выводить кодировку канала;
- бит 2 – не выводить комментарий канала;
- **REGCOUNT**=<число регистров чтения> – только для протоколов, подобных MODBUS;
- **BYTESREAD**=<сколько байтов читать>;
- **NREAD**=<номер байта, с которого начинаются данные, DEC> – например, для MODBUS должно быть **NREAD**=3;
- **SREAD**=<символ> – если этот ключ задан, считается, что данные следуют после указанного символа;
- **SEPARATOR**=<символ> – указывает разделитель данных (по умолчанию – запятая);
- **CR** или **CRLF** – если этот ключ задан, в конце послышки/приемной строки добавляется соответственно Chr(10) или Chr(10)+Chr(13);

Ключи **SREAD**, **SEPARATOR**, **CR** и **CRLF** – для протоколов ASCII.

- **SENDCMD**=<команда> – набор байтов в формате HEX без префикса «0x» для отправки (для бинарных и ASCII протоколов). Кроме 0..F, набор может содержать следующие predefined выражения/символы:
 - **c<0..5>** – подставляется соответствующий байт удаленного адреса ((93, **C0**)..(98, **C5**));
 - **i<0..2>** – подставляется соответствующее слово удаленного адреса ((90, **I0**)..(92, **I2**));
 - **I<0..2>** – то же, что **i<0..2>**, но в ((90, **I0**)..(92, **I2**)) меняется порядок следования байтов;
 - **z<2,,9>** – подставляется соответственно 2..9 нулей;
 - **x** – подставляется число от 0 до 255, это число меняется;
 - **I** – подставляется (<число байтов в отсылке после I>-1);
 - **L** – подставляется (<число байтов в отсылке после L>+1);
 - **n** – подставляется число байтов в отсылке;
 - **N** – подставляется (<число байтов в отсылке>+2);
 - **t0** – подставляется значение атрибута 59 (4 байта);
 - **t1** – подставляется (<значение атрибута 59>+<период архивирования>);
 - **ts** – подставляется значение атрибута 59, преобразованное в соответствии с **TFRT<n>** (поддерживается только **n**=0, 10, 3 и 13; все форматы поддерживаются в релизе 6.08 и старше);
 - **T<1, 2, 3, 4 или h>** – подставляется значение даты и времени, полученное из текущего времени компьютера по следующему алгоритму.

Шаг 1 – преобразуется считанное время компьютера:

- h** – начало предыдущего часа;
- 1** – начало текущего дня;
- 2** – начало предыдущего дня;
- 3** – начало текущего месяца;
- 4** – начало предыдущего месяца.

Шаг 2 – результат шага 1 преобразуется в соответствии с **TFRT**;

Чтобы в отсылке присутствовали только байты, заданные ключом **SENDCMD**, нужно использовать источник **R_FIFO_Queue** и задать в строке **Дополнительно** этого источника ключ **MDV=NO**. Если ключ **MDV=NO** отсутству-

ет, реализуется MODBUS-подобный протокол (к посылке добавляются 6 байт, разбор ответа производится по MODBUS и т.п.).

- **TFRT**<n> – этот ключ задает правило декодирования полученного значения времени, **n** – двухразрядное число от 00 до 31, расшифровка приведена ниже;
- **VFRF**<n> – задает формат переменных (порядок следования байтов):
 - 0 – 3210 (MODBUS);
 - 1 – 1032;
 - 2 – 0123 (INTEL);
 - 3 – 0123;
 - 7 – из принятой последовательности берутся 3 байта, к ним добавляется 8 нулевых битов, и полученное таким образом 4-байтовое число интерпретируется как FLOAT;
- **VLIF**<n> – формат целочисленных переменных:
 - 0 – MODBUS;
 - 2 – INTEL;
- **VDRF**<n> – формат переменных DOUBLE:
 - 0 – MODBUS, 4 байта;
 - 2 – INTEL, 4 байта;
 - 4 – INTEL, 8 байт;
- **TREC**=<0...99> – время записи считывается из принятой последовательности байтов, начиная с поля (параметра) с номером **TREC**;
- **MREC**=<число DEC <= 4000> – максимальное число записей в архиве;
- **MULT**=<k> – коэффициент, на который умножаются принятые и распределяемые по аргументам данные. **k** может принимать следующие значения: 10, 100, 1000, 0.1, 0.01, 0.001;
- **ECHO** – после отправки считывается то, что послано, а затем считывается столько же байт;
- **-STEP** – из временной метки, переданной устройством, вычитается период архивирования;
- **UTC=ON** – время, принятое от устройства, интерпретируется как UTC. Смещение вычисляется автоматически или задается в файле *.cnf с помощью ключа **UTC_OFFSET**=<число минут>. Ключ **ISDST=OFF** в файле *.chf отключает анализ сезона;
- **UDP** – при обмене с УСО используется UDP вместо TCP.

Для формирования контрольной суммы надо задать ключи:

- **CRCADD** – контрольной суммой является сумма всех символов

команды по модулю 256;

- **CRCXOR** – контрольная сумма типа логическое исключение;
- **CRCNO** – не формировать контрольную сумму.

Во всех остальных случаях используется контрольная сумма, создаваемая в источниках MODBUS.

Процедуры, выполняемые при распределении принятых данных по аргументам, зависят от типа данных аргумента:

- **BOOL** – записывается 0 или 1;
- **USINT (char)** – из приемной строки считывается байт и записывается в аргумент;
- **2-байтный целый** – из приемной строки считываются 2 байта, преобразуются согласно **VFRF** и записываются в аргумент;
- **4-байтный целый** – из приемной строки считываются 4 байта, преобразуются согласно **VFRF**, умножаются на **MULT** и записываются в аргумент;
- **REAL** – если в источнике/приемнике указана команда MODBUS float, из приемной строки считываются 4 байта, преобразуются согласно **VFRF** и записываются в аргумент;
- **LREAL** – при наличии ключа **VDRF4** из приемной строки считываются 8 байтов, преобразуются согласно **VFRF** и записываются в аргумент. Аргумент должен быть типа OUT и должен быть привязан к каналу класса DOUBLE FLOAT;
- **DATE_AND_TIME** – считанное значение преобразуется согласно **TFRT** и записывается в аргумент;
- **STRING** – считывается не более 4 символов.

К каналу CALL.Vector могут быть привязаны следующие источники/приемники:

- **MODBUS** – при следующих назначениях RS:
 - 0, **Master**
 - 15, **E_Meter**
 - 19, **Host Variant**
 - 20, **Host OEM RS**
- **UniversalPLC_user1...UniversalPLC_user4** – при следующих назначениях RS:
 - 15, **E_Meter**
 - 19, **Host Variant**
 - 20, **Host OEM RS**
- **MODBUS**, у которого в строке **IP-адрес** задан один из следующих ключей:
 - **TCP=<IP-адрес>** – подтип и дополнение к подтипу такого

канала CALL.Vector – 9.<код команды Modbus>; реализуется протокол Modbus TCP/IP (ключи конфигурирования описаны в разделе **Обмен по встроенным протоколам по сети**);

- **UDP**; – подтип такого канала CALL.Vector – 30, протокол – USER UDP;
- **IP**; – подтип такого канала CALL.Vector – 31, протокол – USER TCP.

В потоках 30 и 31 реализуется соответственно Modbus UDP/IP и Modbus TCP/IP. Если не задано никаких дополнительных ключей и команда – **3, 4, 8** или **9**, то реализуется истинный Modbus. Если команда – **R_FIFO_Queue**, анализируется **SENDCMD** (см. выше).

Если в строке **Дополнительно** указан ключ «**ASCII**», протокол рассматривается как ASCII (работают ключи **NREAD, SREAD**), в противном случае – как бинарный.

В протоколе ASCII не вычисляется контрольная сумма и не осуществляются подстановки в командную строку числовых значений атрибутов канала.

К каналу CALL.Vector, в котором не установлен флаг **Запрос времени значения**, (далее – **vector**) может быть привязан локальный канал CALL (далее – **call**). В этом случае аргументы **call** копируются в аргументы **vector (**)**, а метка времени копируется по следующим правилам (*):

- если **call** – CALL.Vector, то копируется атрибут 59;
- если **call** – CALL.ChGroupReq или CALL.AS_DATA, то метка времени копируется с проверкой:
 - копируется **call.59**, если значение этого атрибута больше числа секунд в году;
 - копируется **call.45**, если **call.59** меньше числа секунд в году;
- если **call** – любой другой канал CALL, копируется **call.45**.

(*) Если в **vector** задан ключ **TREC**, метка времени копируется из аргумента, заданного этим ключом.

(**) Если привязан канал **CALL.RT_Statistics** (Параметр&0x20), в аргументы CALL.Vector копируются первые интервальные параметры.

К каналу CALL.Vector, в котором флаг **Запрос времени значения** установлен, может быть привязан только локальный канал CALL.Vector. В этом случае тип канала может быть как INPUT (чтение архива привязанного канала), так и OUTPUT (запись в архив привязанного канала).

К каналу CALL.Vector INPUT (**vector1**) может быть привязан атрибут (0, **R**) удаленного канала CALL.Vector (**vector2**). В этом случае **vector1** со своим периодом запрашивает атрибут **vector2.59** и, в случае изменения этого атрибута, копирует аргументы **vector2**. Если в канале **vector1**

установлен флаг архивирования и детектируются пропуски записи в архив (разность (<текущее время> – <атрибут 59>) больше периода записи в архив), недостающие записи запрашиваются из удаленного архива.

Режим работы связи каналов CALL.Vector по сети задается с помощью следующего ключа в файле *.cnf:

- **UNITV_REGIM**=<>

После получения данных и метки времени метка времени выравнивается по периоду и сравнивается со значением атрибута 59. При несовпадении метка времени пишется в атрибут 59, а данные – в архив.

Архивные данные могут быть выведены на тренд – для этого к кривой тренда нужно привязать аргумент данных.

Канал CALL.Vector может быть привязан к самому себе. В этом случае к аргументам канала должны быть привязаны другие каналы CALL.Vector. Для вывода аргументов всех каналов используется **SubNum**=546.

Для вывода канала CALL.Vector используются следующие **SubNum** (в соответствии с **TSHHDR**):

- 544 – текущее значение (таблица, если к аргументам привязаны каналы CALL.Vector);
- 548 – за текущий день или час (определяется периодом сохранения в архив – если больше минуты, то за день);
- 549 – за предыдущий день или час.

Атрибут (240, **ERR**) канала CALL.Vector содержит код ошибки:

- 1 – файл не найден;
- 2 – таймаут;
- 4 – общая ошибка (например, недостаточно памяти);
- 5 – ошибка записи;
- 6 – некорректная дата;
- 7 – не найдены данные.

Расшифровка **n** ключа **TFRT**<**n**> :

- 0, 1, 10 – бинарное число секунд с 1970 года (1 – нотация Intel, 10 – нотация Motorola);
- 3, 13 – (секунды, минуты и часы в BCD) плюс ((3) день и месяц в BCD или (13) месяц и день в BCD) плюс год (2 байта);
- 5 – 2 байта года, месяц, день, часы, минуты, секунды, прямая нотация;
- 11 – 2 байта года, 2 байта месяца, 2 байта дня месяца, пропуск 2 байтов, 2 байта часов, 2 байта минут, 2 байта секунд;
- 2, 12 – 4 байта преобразуются соответственным образом (2 – Intel, 12 – Motorola), далее: старшие 6 бит – год, следующие 4 бита – ме-

сяц, следующие 5 бит – день месяца, следующие 5 бит – часы, следующие 6 бит – минуты, следующие 6 бит – секунды;

- 18 – АВВ: 4 байта переводим во float, печатаем как целое, далее: младшие 3 байта – это секунды (байт 0), минуты и часы в BCD. Следующие 4 байта (после аналогичного преобразования) – год, день и месяц;
- 8 – 18, но год, день, месяц, часы, минуты, секунды;
- 7 – формат времени МЭК-104 (7 байт);

Побайтовые:

- 17 – HMSdmy;
- 6 – ymdHMS;
- 9 – dmySMH;
- 4 – Hdmy;
- 14 – Hdmy в BCD;
- 15 – HMdmy в BCD;
- 16 – SMHdmy в BCD;
- 20 – dmyHM;
- 21 – dmyH.

Если в строке **Дополнительно** указан ключ **FBIN=<имя файла без расширения>**, канал загружает характеристики источников-приемников из указанного файла с расширением *.udf (из папки узла). Если файл отсутствует или некорректен, в канале устанавливается признак аппаратной недостоверности.

CALL.Vector без привязки:

- если флаг **Запрос времени значения** (атрибут 50) установлен, в атрибут 45, **Время изменения** записывается время отработки;
- если флаг **Запрос времени значения** (атрибут 50) не установлен, значение атрибута 45 нужно формировать вручную.

CALL.Vector с привязкой:

- CALL.ROOT (140–186) – в вектор записываются соответствующие аргументы.

Канал CALL.ROOT

Данный канал предназначен для генерации таблиц и вывода их в файл или документ. К аргументам канала CALL.ROOT могут быть привязаны различные атрибуты различных каналов.

Для вставки таблицы в документ используются **SubNum=544, 545, 644, 645** (см. **Номер SubNum**).

Для вывода в файл в выходное значение (9,**Q**) канала CALL.ROOT необ-

ходимо записать число:

- 54/64 – HTML (значения переменной по столбцам);
- 55/65 – HTML (значения переменной по строкам);
- 56/66 – XML (только файл);
- 57/67 – текстовый файл (ТХТ).

Существуют predetermined числа для записи в выходное значение (эти числа автоматически конфигурируют и изменяют канал CALL.ROOT):

- 10, 20, 30 – текущие значения;
- 16, 26, 36 – текущий год по месяцам;
- 14, 24, 34 – предыдущий год по месяцам;
- 19, 29, 39 – текущий месяц по дням;
- 17, 27, 37 – предыдущий месяц по дням;
- 18, 28, 38 – текущий год по месяцам, а каждый месяц по дням.

Параметры генерации:

- **id_class** – значение битов 4-6 (0x70) атрибута **Параметр** (34, **FPnt**);
- **DOC_N** – атрибут (92, **I2**) – счетчик сгенерированных документов (автоматически увеличивается на 1 при каждой генерации);
- **DOC_G** – маска генерации, может быть задана в окне компонентов как **ROOT_HREAD_CLASS<класс>=<hex>**. Например, для класса 0:

ROOT_HREAD_CLASS0=<hex>

Задание архивирования и флаг **Запрос времени значения** (атрибут 50) определяют алгоритм выборки значений для канала:

- **SEL0** – **AV=0 Tf=0**;
- **SEL1** – **AV=0 Tf=1** или **(DOC_G & 3) == 1**;
- **SEL2** – **AV=1 Tf=0** или **(DOC_G & 3) == 2**. При этом алгоритме для числового канала выполняется обращение к архиву SIAD, а для канала CALL.Vector – к его файлу;
- **SEL3** – **AV=1 Tf=1** или **(DOC_G & 3) == 3**. При этом алгоритме выполняется обращение к архиву.

Параметр **Period** – величина интервалов, на которые разбивается диапазон (T_FROM, T_TO), задается значением битов 0-3 (0x0F) атрибута **Параметр** (34, **FPnt**) (см. **Временной интервал выборки**):

- 1 – 1мин.;
- 2 – 1ч;
- 3 – 5мин.;
- 4 – 15мин.;

- 5 – 20мин.;
- 6 – 30мин.;
- 7 – 2ч;
- 8 – 4ч;
- 9 – 6ч;
- 10 – 8ч;
- 11 – 12ч;
- 12 – 24ч;
- 13 – неделя (7 дней);
- 14 – декада (10 дней);
- 15 – месяц (30 дней).

При генерации в файл имя файла создается следующим образом:

`[<id_class>_]<ROOT.name>[_DOC_N]`

Необязательная составляющая **id_class** добавляется к имени, если **id_class** <>0.

Необязательная составляющая **DOC_N** добавляется к имени, если установлен бит 8 (0x100) **DOC_G**.

Реальное значение (0, **R**) канала задает число генераций таблицы (0, 1 – один раз, 2 – два и т.д.).

Таблица содержит значения от T_START до T_STOP (если заданное число генераций таблицы больше 1, то при генерациях после первой (T_START-T_STOP) автоматически меняется):

- T_START – начальное время поиска значений, задается атрибутом (59, **DR**);
- T_STOP – конечное время поиска значений, вычисляется автоматически:
 - для **SEL2** и **SEL3**:
 - если **Period** = 1мин. или 5мин., T_STOP-T_START = 1ч;
 - если **Period** = 15мин., 20мин., 30мин, 1ч, 2ч, 4ч, 6ч, 8ч или 12ч, T_STOP-T_START = 24ч (1 день);
 - если **Period** = 1, 7 или 10 дней, T_STOP-T_START = 1 месяц (30 дней);
 - если **Period** = 1 месяц (30 дней), T_STOP-T_START = 1 год.
 - для **SEL0** и **SEL1**:

В XML- и TXT-файлы имеется возможность выводить все найденные точки за временной интервал выборки – для этого младшие 4 бита атрибута **Параметр** (34, **FPrnt**) должны быть равны 0. Интервал выборки в этом случае определяется атрибутами (59, **DR**) и (252, **dT**) (T_STOP и T_START выравниваются на границу минуты):

- **DR=dT=0:**
 $T_STOP = \langle \text{текущее время} \rangle$
 $T_START = T_STOP - 1\text{ч}$
- **DR<>0, dT=0:**
 $T_START = DR$
 $T_STOP = T_START + 1\text{ч}$
- **DR=0, dT<>0:**
 $T_STOP = \langle \text{текущее время} \rangle$
 $T_START = T_STOP - dT$
- **DR<>0 и dT<>0:**
 $T_START = DR$
 $T_STOP = T_START + dT$

Назначение аппаратного значения (1, **A**):

- значение битов 0-3 (0x000f) – задает итоговое значение (только при выводе каналов CALL):
 - 1 – сумма;
 - 2 – среднее;
- биты 4-7 (0x00f0) – задают формат:
 - бит 4 (0x0010) – нумерация столбцов;
 - бит 5 (0x0020) – двойной/одинарный формат времени;
 - бит 6 (0x0040) – разрешение кодировки и комментария от вектора;
- биты 8-11 (0x0f00):
 - бит 8 (0x0100) – нумерация строк;
 - бит 9 (0x0200) – столбец/строка времени от ROOT;
 - бит 10 (0x0400) – пустая строка или столбец;
- биты 12-13 (0x3000) – вид представления значения (XML);
- значение битов 14-15 (0xC000) – управление генерацией заголовка таблицы:

Расширенный заголовок задается привязкой к ROOT атрибута (245, **A_OPT**) канала CALL.ChGroupReq или CALL.String. В этом случае в заголовке выводится имя канала и его единицы измерения, а при **SubNum** = 64, 65, 66 добавляется кодировка и комментарий.

- 0 – генерировать стандартный и расширенный заголовки;
- 1 – запрет стандартного заголовка;
- 2 – запрет расширенного заголовка;

- 3 – запрет обоих заголовков;
- бит 16 (0x10000) – добавить вывод времени последнего изменения канала.

Особенности привязок атрибутов каналов к аргументам CALL.ROOT:

- при привязке атрибута 127, 80 или 79 генерируется пустой столбец или строка, а в заголовке пишется значение соответствующего атрибута;
- при привязке атрибута 124 канала CALL (здесь имеются в виду канал **CALL.ROOT**, канал **CALL.ChGroupReq**, канал **CALL.Sum** и канал вызова программы (см. **Выполнение программы в реальном времени**)) генерируется новый уровень таблицы (максимальное число уровней – 3);
- при привязке атрибута 140 .. 186 (или аргумента arg0 .. arg46) извлекается этот аргумент.

К аргументу CALL.ROOT может быть привязан канал **CGR_main** (см. **Универсальный механизм обмена с электросчетчиками**). В зависимости от привязанного атрибута, в таблицу выводятся следующие параметры:

- 230, ATR_FEW_SHORT_PARAM – частота, ток, напряжение;
- 231, ATR_FEW_GROOP_PARAM – мощности, коэффициенты мощности (по фазам);
- 233, ATR_FEW_AVR_PARAM – мощности (суммы по фазам);
- 235, ATR_ARC_AVR_PARAM – средние мощности за интервал (из архива);
- 236, ATR_ITOGO_DROPPED_PARAM – энергии от сброса (из архива);
- 237, ATR_ITOGO_DROPPED_MONTH_PARAM – энергии по месяцам.

К аргументу CALL.ROOT может быть привязан канал **CALL.Vector**.

Каналы CALL.AS_DOCUMENT и CALL.AS_DATA

См. **Отчеты АСКУЭ**.

Канал CALL.EXCEPT_DAY

Канал CALL с типом вызова (126) **EXCEPT_DAY** предназначен для корректировки рабочего календаря, тип данных аргументов канала должен быть DATE или DATE_AND_TIME.

Если в узле нет каналов CALL.EXCEPT_DAY, дни с понедельника по субботу – рабочие, воскресенье – выходной.

Функции канала CALL.EXCEPT_DAY, **Параметр=1**:

- установленные биты атрибута 0, **R** объявляют выходными соответствующие дни недели (бит 6 – понедельник, бит 5 – вторник и т.д.);
- даты из значений аргументов канала объявляются выходными.

Функции канала CALL.EXCEPT_DAY, **Параметр=0** (имеет более высокий приоритет по сравнению с **Параметр=1**):

- даты из значений аргументов канала объявляются рабочими.

При останове МРВ значения аргументов канала CALL.EXCEPT_DAY записываются в файл **../<папка узла>/<имя канала>.txt**. При старте МРВ значения аргументов восстанавливаются из этого файла.

Для анализа рабочего календаря используются каналы TIME (см. **Канал класса TIME**).

Канал CALL.EMAIL

Канал класса CALL с типом вызова 127, **EMAIL** отправляет сообщения по электронной почте:

- **Кодировка** (79, **CODE**) – имя сервера исходящих сообщений (SMTP);
- **Комментарий** (80, **CMNT**) – адрес электронной почты;
- **Arg0** (тип данных – любой целочисленный) – для служебных целей;
- **Arg1** (STRING) – тема;
- **Arg2** (STRING) – имя вложенного файла (без пути – из папки проекта);
- **Arg3** (STRING) – сообщение.

Для конфигурирования данной функции используются следующие ключи в файле *.cnf:

- **EMAIL_SRV**=<имя сервера исходящих сообщений (SMTP)> – если этот ключ задан, имя сервера в атрибуте CALL.EMAIL.79 не указывается;
- **EMAIL_DEF**=<адрес **Кому**> – если этот ключ задан, адрес получателя в атрибуте CALL.EMAIL.80 не указывается;
- **EMAIL_FROM**=<адрес **От кого**>;
- **EMAIL_LOGIN**=<имя учетной записи>;
- **EMAIL_PASSW**=<пароль учетной записи>.

Отправка сообщений по электронной почте требует наличия библиотек **tm_email_send.dll** и **tmemailc.dll**.

Если сообщение не удалось отправить, в протокол профайлера записывается соответствующее сообщение, в канале устанавливается признак аппаратной недостоверности, а атрибут (240, **ERR**) канала индицирует код

ошибки:

- 1 – нет библиотек;
- 2 – нет библиотечных функций;
- 3 – отправка сконфигурирована не полностью.

Запись вектора в CALL.ChGroupReq и CALL.TVC

Стандартный механизм последовательной записи набора значений с метками времени в аргументы каналов CALL.ChGroupReq и CALL.TVC (см. например, **Выборка данных из локального архива по каналу**):

- в аргументы CALL.ChGroupReq записываются только значения (начиная с **arg0**);
- аргументы CALL.TVC:
 - **arg0** и **arg1** – границы временного диапазона (вычисляются автоматически);
 - начиная с **arg2**, аргументы создаются парами, в первый записывается значение, во второй – его время.

Запись длинных строк в канал CALL

Длинные (до 64 байт) строки могут быть записаны в аргументы каналов CALL.STRING, CALL.ChGroupReq и CALL.TVC. Аргументы не должны иметь привязок и иметь тип данных STRING.

Для канала CALL.STRING:

- если в канале установлен флаг **Отчет тревог (43,EB)**, при каждом изменении CALL.STRING.R значение соответствующего аргумента (или комментарий) записывается в ОТ (не более 48 символов);
- если в канале задано использование дампа (установлен бит 1 атрибута **Синхр/Дамп (58,DumpSync)**), при останове монитора текущие значения аргументов сохраняются в файле **../<папка узла>/<имя канала>.dmp**;
- при старте монитора значения аргументов считываются из файла **../<папка узла>/<имя канала>.txt**. Формат файла:
 - <число задаваемых аргументов, N>
 - <значение arg0>
 - <значение arg1>
 - ...
 - <значение argN>
 - <пустая строка>

Если такого файла нет и в канале задано использование дампа (установлен бит 1 атрибута **Синхр/Дамп (58,DumpSync)**), значе-

ния аргументов восстанавливаются из файла **./<папка узла>/<имя канала>.dmp**.

Для принудительного считывания/записи значений аргументов в файл дампа **./<папка узла>/<имя канала>.dmp** нужно послать 0/8 в атрибут (41, **AB**).

Для принудительного считывания/записи значений аргументов в файл **./<папка узла>/<имя канала>.txt** нужно послать 2/9 в атрибут (41, **AB**).

Для принудительной записи значений аргументов (только константы) в файл **./<папка узла>/<имя канала_DDMMYYYYhhmm>.txt** нужно послать 15 в атрибут (41, **AB**).

Для принудительного считывания/записи значений аргументов в файл **./<папка узла>/<имя канала_N>.txt** (где **N=1..15**) нужно послать (16-31)/(32-47) в атрибут (41, **AB**).

CALL.STRING с Параметр = 1-8. При старте MPB такой канал копирует привязки в аргументы, затем, если в канале задан дамп, считывает значения аргументов из дампа. Для вывода таких переопределенных строк используется **SubNum=402-409**.

При межкомпонентном взаимодействии:

- для отображения значения аргумента с помощью ГЭ **Текст** нужно к аргументу экрана (тип данных должен быть **STRING**) привязать атрибут **1,A** канала и задать формат **Generic** (см. **Динамизация атрибута ГЭ**). ГЭ отображает значение аргумента с номером **CALL.STRING.R**. Если **R** больше или равно числу аргументов, отображается комментарий канала (атрибут 80, **CMNT**);
- аргументы программы читают и пишут в аргументы **CALL.STRING**, привязанные через атрибуты и аргументы;
- для записи строки из программы в **CALL.STRING** при связи «аргумент-аргумент» аргумент **CALL.STRING** должен иметь тип **IN/OUT**;
- аргумент экрана может быть привязан к **CALL.STRING** через атрибуты;
- аргумент экрана может быть привязан к **CALL.PROGRAM** через аргументы, которые привязаны к **CALL.STRING** через атрибуты и аргументы.

Следующая конфигурация **НЕ РАБОТАЕТ**: аргумент экрана привязан к **CALL.PROGRAM** через атрибуты, которые привязаны к **CALL.STRING** через атрибуты и аргументы.

Перепривязка аргументов канала CALL

Перепривязка аргументов CALL.Screen

Для перепривязки аргументов экрана предназначена конфигурация, в которой к CALL.Screen привязан атрибут (124, **ArgSize**) или атрибут (140, **ARG00**), (141, **ARG01**) и т.д. канала CALL.ROOT или CALL.Program (**root**), к аргументам которого привязаны атрибуты (0, **R**) каналов CALL (**call**).

Для перепривязки анализируется аппаратное значение CALL.Screen (4 байта B3.B2.B1B0).

При перепривязке учитывается значение битов 4 и 5 атрибута 34 канала CALL.Screen:

- значение битов 4 и 5 равно 0 – перепривязка, начиная с arg0;
- значение битов 4 и 5 равно 1 – перепривязка, начиная с arg1;
- значение битов 4 и 5 равно 2 – перепривязка, начиная с arg2;
- значение битов 4 и 5 равно 3 – перепривязка, начиная с arg4.

CALL.Screen.A.B2=255

Данный механизм – это перепривязка через изменение **root.R**.

- К CALL.Screen привязан **root.124**.
Если значение B1B0 не равно **root.R**, к аргументам CALL.Screen привязываются аргументы **call**, который привязан к аргументу с номером **root.R** канала **root**, после чего CALL.Screen.A.B1B0=**root.R**.
- К CALL.Screen привязан **root.140**, **root.141** и т.д.
Если значение B1B0 не равно **root.R**, к аргументам CALL.Screen привязываются аргументы с номером **root.R** всех **call**, после чего CALL.Screen.A.B1B0=**root.R**.

CALL.Screen.A.B2<>255

Данный механизм – это перепривязка через изменение B2 (вне зависимости от **root.R**). Чтобы изменить значение B2, нужно послать значение в атрибут 241. Можно также послать значение непосредственно в атрибут (1, **A**), однако этот способ нежелателен.

- К CALL.Screen привязан **root.124**.
К аргументам CALL.Screen привязываются аргументы **call**, который привязан к аргументу с номером B2 канала **root**, после чего CALL.Screen.A.B1B0=CALL.Screen.A.B2.
- К CALL.Screen привязан **root.140**, **root.141** и т.д.

К аргументам CALL.Screen привязываются аргументы с номером B2 всех **call**, после чего CALL.Screen.A.B1B0=CALL.Screen.A.B2.

Перепривязка аргументов CALL.SQLQuery

Перепривязка аргументов CALL.SQLQuery не поддерживается.

Перепривязка аргументов CALL.Document(Report)

Перепривязка аргументов документа аналогична перепривязке аргументов экрана при CALL.Screen.A.B2=255.

Для перепривязки аргументов документа предназначена конфигурация, в которой к CALL.Document(Report) привязан атрибут (124, **ArgSize**) или атрибут (140, **ARG00**), (141, **ARG01**) и т.д. канала CALL.Screen или CALL.Program (**root**). Перепривязка производится в момент генерации документа и зависит от **root.R**.

Если к CALL.Document(Report) привязан атрибут (124, **ArgSize**) или атрибут (140, **ARG00**) канала **root**, то

```
CALL.Document(Report).arg0=root.arg0  
CALL.Document(Report).arg1=root.arg1  
и т.д.
```

Если к CALL.Document(Report) привязан атрибут (140+N, **ARG<N>**) канала **root**, то

```
CALL.Document(Report).arg0=root.argN  
CALL.Document(Report).arg1=root.arg(N+1)  
и т.д.
```

Документ может быть привязан к аргументу 124 или 140, 141.. экрана (или другого документа). В этом случае при генерации в аргументы документа копируются значения аргументов экрана (другого документа). При привязке к атрибуту 124 аргументы копируются, начиная с arg0, при привязке к атрибуту 140, 141 и т.д. – начиная с аргумента arg0, arg1 и т.д.

Канал класса СОБЫТИЕ

Канал класса **Событие** предназначен для мониторинга объекта с целью фиксирования возникновения/исчезновения на этом объекте некоторого события или ситуации (например, аварии). Канал хранит историю события (в стеке, последнее по времени сообщение имеет номер 0) и допускает квитиование последнего по времени события.

В канале предусмотрены два алгоритма – стандартный (фиксация двух типов событий) и расширенный (фиксация 127 типов событий). Для использования расширенного алгоритма нужно в редакторе канала задать значение **На старте**, большее или равное 16, и установить флаг **Отработать** (см. **Общие атрибуты каналов**).

Для работы с каналами этого класса в РПД предусмотрен специальный графический элемент – таблица каналов класса СОБЫТИЕ (см. **ГЭ ‘События’**).

В профайлере канал **Событие** индицируется как **C12_Event** (атрибут 126, TsT).

События делятся на **короткие** ($T_{\text{исчезновения}} - T_{\text{возникновения}} < \text{Ack_after_off}$) и **длинные** ($T_{\text{исчезновения}} - T_{\text{возникновения}} \geq \text{Ack_after_off}$) – см. описание системной переменной **@RTM_Parameter** в разделе **Группа СИСТЕМНЫЕ**.

С помощью этой же переменной задается таймаут на квитиование (**Ack_after_on**).

Кроме атрибутов, которые имеют каналы всех классов (см. **Общие атрибуты каналов**), каналы класса **Событие** имеют специфические атрибуты.

К специфическим атрибутам, которые могут быть заданы в редакторе канала, относятся следующие:

- раздел «Параметры»:
 - **Размер стека аварий** (17, **Stack_Size**) – глубина стека для хранения истории события (1–255, 0 – стек не используется);
 - флаг **Квитиование в сеть** (84, **ACKtoNET**) – этот флаг используется в случае необходимости квитиования события на удаленных сетевых узлах. Пусть, например, мониторинг объекта выполняет канал **ev_1** узла **node1**, а в узлах **node2...nodeN** существуют соответственно каналы **ev_2...ev_N**, для каждого из которых задана привязка к каналу **ev_1** и установлен флаг **Квитиование в сеть** (**ACKtoNET=1**). При такой конфигурации:

– в каналы **ev_2...ev_N** передается информация о статусе события и его квитировании на узле **node1** (вне зависимости от флага **Квитирование в сеть** канала **ev_1**);

– квитирование события может быть выполнено на любом из узлов **ev_2...ev_N** – уведомление об этом будет получено на узле **node1** и, соответственно, на всех остальных узлах группы **ev_2...ev_N**.

В профайлере доступны также следующие специфические атрибуты канала:

- **Статус события (0, R)** – статус события. Этот параметр автоматически принимает следующие значения:
 - 0 (в мониторе по умолчанию – ...) – событие не возникло;
 - 1 (**E_On**) – событие первого типа возникло и не квитировано;
 - 2 (**E_Off+ACK**) – событие исчезло после своевременного квитирования;
 - 3 (**E_On+ACK**) – событие возникло и квитировано своевременно;
 - 4 (**E_Off+unACK**) – событие не было квитировано и исчезло. Для коротких событий в этом статусе возможно квитирование (и переход, в зависимости от таймаута **Ack_after_on**, в статус 2 или 6). Длинные события в статусе 4 не могут быть квитированы;
 - 5 (**E_On+wACK**) – событие возникло и квитировано несвоевременно;
 - 6 (**E_Off+wACK**) – событие исчезло после несвоевременного квитирования;
 - 7 (**E_On_On**) – событие любого типа, отличного от первого, возникло и не квитировано.
- **Наличие/отсутствие события (7, P)** – этот атрибут индицирует наличие/отсутствие события (соответственно 1 и 0);
- **Дата и время прихода (10, EventIn)** – дата и время возникновения события (с точностью до миллисекунд);
- **Дата и время ухода (11, EventOut)** – дата и время исчезновения события (с точностью до миллисекунд);
- **Дата и время квитирования (12, EventAck)** – дата и время квитирования события (с точностью до секунд). Для квитирования надо послать в данный атрибут ненулевое значение;
- **Число событий (13, ECount)** – счетчик событий, увеличивается на 1 при каждом возникновении события;
- **Миллисекунды прихода (14, MsIn)** – миллисекунды возникновения события;
- **Миллисекунды ухода (15, MsOut)** – миллисекунды исчезновения события;

ния события;

- **Число квитирований** (16, **Ack_Count**) – счетчик квитирований;
- **Число записей в стеке** (18, **InStack**) – число записей, помещенных в стек. Сообщение заносится в стек при исчезновении события и содержит следующую информацию (справа в скобках указан атрибут, в который выводится соответствующая часть сообщения при его выборе из стека с помощью атрибута **Позиция** (19, **Position**)):
 - дата и время прихода с точностью до миллисекунд (128, **Tin**);
 - дата и время ухода с точностью до миллисекунд (129, **Tout**);
 - дата и время квитирования с точностью до секунд (если квитирование было выполнено) (130, **Tack**);
 - миллисекунды прихода (131, **MSin**);
 - миллисекунды ухода (132, **MSout**);
 - статус при уходе (133, **STS**);
- **Позиция** (19, **Position**) – этот атрибут задает номер сообщения в стеке (позиции стека нумеруются с 0) – указанное сообщение отображается атрибутами 128-133. Монитор отслеживает параметр **InStack** и игнорирует попытки присвоить несуществующий номер атрибуту **Position**.

Стандартный алгоритм канала СОБЫТИЕ

Возникновение/исчезновение события канал определяет следующим образом:

- событие первого типа возникло, если входное значение канала (атрибут **In**) изменилось и приняло значение 1;
- событие второго типа возникло, если входное значение канала (атрибут **In**) изменилось и приняло любое целое значение **n** ($n \geq 2$);
- событие исчезло, если входное значение канала (атрибут **In**) приняло нулевое значение после ненулевого.

Расширенный алгоритм канала СОБЫТИЕ

Возникновение/исчезновение события канал определяет следующим образом:

- событие типа **n** возникло, если входное значение канала (атрибут **In**) изменилось и приняло целое значение **n** ($n=1...127$);
- событие исчезло, если входное значение канала (атрибут **In**) при-

няло нулевое значение после ненулевого.

При расширенном алгоритме в ГЭ **События** и отчет тревог выводятся сообщения с указанием типа события (к значению статуса добавляется значение **In**, заключенное в скобки). Для корректной работы ГЭ каналу периодически присваивается значение 7 (на 1 такт).

При расширенном алгоритме квитирование в сеть отключено.

Особенности свойства СВЯЗЬ канала СОБЫТИЕ

Канал класса СОБЫТИЕ может быть привязан к любому атрибуту канала произвольного класса с одним ограничением – привязка к атрибуту **R** другого канала возможна для канала СОБЫТИЕ, в котором не используется расширенный алгоритм обработки.

Если к каналу класса СОБЫТИЕ (**ev**) привязан произвольный атрибут **aaa** некоторого канала **ch** не класса СОБЫТИЕ, то **ev.In = ch.aaa**.

Если к каналу класса СОБЫТИЕ (**ev**) привязан атрибут (7, **P**) некоторого канала **ch**, связанного со словарем, то к сообщениям, генерируемым в канале **ev**, через двоеточие добавляются сообщения канала **ch**, если в **ev** установлен флаг **Атрибуты** (44, **MB**).

Канал СОБЫТИЕ (пусть его имя – **GNRL**) с кодировкой, которая заканчивается знаком звездочки, является **обобщающим** (в таком канале устанавливается атрибут 40).

Если к **GNRL** привязан некоторый атрибут **aaa** канала класса **Class**, то при обработке **GNRL** МРВ анализирует каналы класса **Class** узла, первые символы кодировки которых совпадают с кодировкой обобщающего канала (до знака звездочки). При обнаружении хотя бы одного канала (пусть его имя – **ch**), удовлетворяющего критерию (см. ниже), входу обобщающего канала присваивается 1 (в противном случае входу обобщающего канала присваивается 0):

- **Class** – СОБЫТИЕ:
 - **aaa** – любой атрибут, кроме указанных ниже: если значение **aaa** равно 1 или 7;
 - **aaa** – **Дата и время прихода** (10, **EventIn**): если **ch.R = 1** или **7** и **ch.EventIn > GNRL.EventAck**;
 - **aaa** – **Дата и время квитирования** (12, **EventAck**): если **ch.EventAck > GNRL.EventAck**;
 - **aaa** – **Число событий** (13, **ECount**): если **ch.R = 1, 3, 5** или **7** и **ch.EventIn > GNRL.EventAck**;
 - **aaa** – **Число квитирований** (16, **Ack_Count**): если **ch.ECount > ch.Ack_Count**;

- **aaa** – (45, T) **Время изменения:** если **ch.T** > **GNRL.EventAck**;
- **Class** – числовой (кроме TIME):
 - **aaa** – (45, T) **Время изменения:** если **ch.T** > **GNRL.EventAck**;
 - **aaa** – (59, DR): если не квитируется последнее сообщение ОТ по каналу **ch**;
 - **aaa** – (7, P): если **ch.P** <> 0. Особый случай: если **ch** недостоверен, то **GNRL.In** = 7;
 - **aaa** – (254, RST): если **ch.RST** = 1 или 7;
 - **aaa** – любой другой атрибут: если **ch.aaa** <> 0;
- **Class** – любой другой класс:
 - **aaa** – любой атрибут: если **ch.aaa** <> 0.

При обработке **GNRL**, в котором привязка не задана, МРВ анализирует каналы всех классов, первые символы кодировки которых совпадают с кодировкой обобщающего канала (до знака звездочки). При обнаружении канала (**ch**) статус **GNRL** устанавливается в соответствии с **ch.R**, если класс **ch** – СОБЫТИЕ, и в соответствии с **ch.RST**, если класс **ch** – любой другой. После присвоения **GNRL.In** ненулевого значения поиск и анализ других каналов в текущем цикле прекращается.

Экспорт данных в файл

Запись значения в атрибут (254, **RST**) обычного канала СОБЫТИЕ приводит к генерации файла с именем по умолчанию **event_export** (1 – TXT, 3 – HTML, 4 – XML, кодировка – utf-8). Имя файла может быть изменено с помощью ключа **EXP_NAME** в файле *.cnf (см. **Задание параметров работы мониторов**). В файл экспортируется стек канала (статус, если стек не задан), а также информация о текущем событии, если оно актуально (статус 1, 3, 5 или 7).

Запись значения в байт 0 атрибута (254, **RST**) **GNRL** приводит к генерации файла с именем канала (1 – TXT, 3 – HTML, 4 – XML, кодировка – utf-8). Файл содержит информацию обо всех случаях возникновения событий (по вышеуказанным критериям) в обнаруженных каналах. Если **aaa** – (254, **RST**) числового канала или привязка не задана, значение байта 1 (0xFF00) атрибута (254, **RST**) **GNRL** задает дополнительное условие отбора данных для экспорта:

- 0 – нет дополнительного условия;
- 1 – каналы с **RST** = 1, 3, 5, 7;
- 4 – каналы с **RST** = 4;
- 7 – включенные каналы с аппаратной недостоверностью;

- 17, 20 и 23 (1, 4 и 7, увеличенные на 16) – соответственно 1, 4 и 7 при условии, что время изменения канала больше или равно времени квитирования **GNRL**;
- 45 – время изменения канала больше времени квитирования **GNRL**;
- 46 – время изменения канала больше времени квитирования **GNRL** и атрибут 46 канала не равен 0.

Файлы создаются в директории, заданной с помощью ключа **AVB_PATH** в файле *.cnf (по умолчанию – в папке узла).

Системные переменные TRACE MODE 6

В дочерних группах **Группа Системные** и **Группа Диагностика** группы **Диагностика и сервис** слоя **Источники/Приемники** описываются системные переменные TRACE MODE.

Каналы, связанные с этими переменными, могут в реальном времени управлять различными системными параметрами или отображать их состояние. Подтип таких каналов – 14 или 15.

Системная переменная содержит набор параметров. При ее привязке к каналу CALL.ChGroupReq MPB создает в канале необходимое число аргументов. Если переменная привязана к числовому каналу, в атрибут (2, **In**) канала записывается параметр с номером, заданным атрибутом **Параметр** переменной (атрибутом 90, **IO** канала).

Параметры системных переменных задаются в редакторах, типовых для источников/приемников (см. **Редактор системных переменных TRACE MODE** и **Редакторы источников (приемников)**).

Все системные переменные создаются монитором вне зависимости от их наличия в проекте. Поэтому, например, для распределения пересчета каналов с типом пересчета **F1...F4** на разные циклы монитора создавать в проекте переменную **@Recalculation_Index** (группа СИСТЕМНЫЕ) необязательно.

Создание системных переменных по комментарию

Если в комментарии любой системной переменной (см. **Редактор системных переменных TRACE MODE**) задана описанная ниже строка чисел, то создается соответствующая системная переменная:

<подтип>.<дополнение к подтипу>.<io>.<dop>

где:

- **подтип** – подтип переменной (14 или 15);
- **дополнение к подтипу** – дополнение к подтипу (2 цифры);
- **io** – атрибут (90, **IO**) (в редакторе – **Параметр**), 1 или 2 цифры;
- **dop** – задается для следующих переменных и обозначает:
 - для 14.00 **@Status** – индекс узла;
 - для 15.00 **@e_SIAD** – операция: 1 – подключить копию архива, 2 – создать копию архива;

- для 15.30 **@q_SIAD_Q** – номер архива;
- для 15.13 **@e_TCP_ModBus** – номер встроенного протокола (9 – MODBUS, 10 – SNMP и т.п.).

Группа СИСТЕМНЫЕ

@Status

Подтип и дополнение к подтипу – 14.0.

В зависимости от значения атрибута **Параметр** (см. **sysdiag_status.tmc**), значение данной 2-байтовой переменной индицирует (INPUT) или задает (OUTPUT) соответствующий параметр узла с порядковым номером, заданным с помощью атрибута **Номер узла** (92,12 в реальном времени; порядковый номер узла не следует путать с его индивидуальным номером – см. **Имена и идентификаторы объектов структуры** и **Задание параметров узла**).

Если **Номер узла** =10000, переменная индицирует/задает параметры собственного узла.

- **Параметр** = 0, **ShortStatus** – короткий статус узла (OUTPUT – задание статуса для текущего узла):
 - 0 – статус неизвестен, и узел невидим;
 - 1 – узел запускается;
 - 2 – WORK;
 - 3 – TRACE;
 - 5 – узел останавливается и выгружается;
 - 9 – узел будет перезагружен.

Канал OUTPUT с **Параметр**=0 используется для задания статуса собственного узла;

- **Параметр** = 1, **Last_Exchange** – разность **<текущее время> – <время последнего приема от узла>** в секундах (по любому интерфейсу) (OUTPUT – установка).
- **Параметр** = 2, **Problems** – 32 бита (помеченные звездочкой биты могут быть установлены для собственного узла каналом OUTPUT):
 - (*) бит 0 (0x1) – 1 – узел не может послать широковещательное сообщение о своем присутствии в сети;
 - (*) бит 1 (0x2) – 1 – то же, индивидуально;
 - бит 2 (0x4) – зарезервировано;
 - (*) бит 3 (0x8) – 1 – узел не может подключиться по TCP;
 - бит 4 (0x10) – 1 – в течение таймаута ничего не получено по TCP/IP или UDP/IP;

- (*) бит 5 (0x20) – 1 – узла нет;
- бит 6 (0x40) – 1 – для узла задан резерв, но резерва нет;
- бит 7 (0x80) – зарезервировано;
- бит 8 (0x100) – зарезервировано;
- (*) бит 9 (0x200) – этот бит устанавливается автоматически при установке бита 5 (сбросить бит 9 можно только вручную);
- бит 10 (0x400) – этот бит устанавливается автоматически при установке бита 6 (сбросить бит 10 можно только вручную);
- бит 11 (0x800) – зарезервировано;
- бит 12 (0x1000) – 1 – очередь отсылки по UDP большая;
- бит 13 (0x2000) – 1 – узел шлет слишком много данных;
- бит 14 (0x4000) – 1 – время рассинхронизации слишком велико;
- бит 15 (0x8000) – 1 – блок UDP отправляется узлу слишком долго;
- бит 16 (0x10000) – 1 – ошибка отправки по тому же сокету (для регистратора);
- бит 17 (0x20000) – 1 – узел работает, но не может связаться с текущим узлом;
- биты 18-20 (0x1C0000) – ошибки отправки широковещательного сообщения через указанный адаптер;
- бит 21 (0x200000) – 1 – есть подключение по TCP;
- бит 22 (0x400000) – 1 – узел подключен к текущему узлу;
- бит 23 (0x800000) – 1 – ошибка отправки по TCP;
- бит 24 (0x1000000) – 1 – от узла получен UDP-блок;
- бит 25 (0x2000000) – 1 – от узла получено широковещательное сообщение;
- бит 26 (0x4000000) – 1 – узел «не умеет» отсылать широковещательные сообщения;
- бит 27 (0x8000000) – 1 – узел «не умеет» принимать широковещательные сообщения;
- бит 28 (0x10000000) – 1 – сообщение текущего узла о присутствии в сети отсылается указанному узлу не широковещательно, а индивидуально;
- бит 29 (0x20000000) – 1 – использовать альтернативный порт для прослушивания сети;
- бит 30 (0x40000000) – 1 – использовать альтернативный порт для подключения по TCP;
- бит 31 (0x80000000) – 1 – сокет для прослушивания сети со-

здан;

- **Параметр = 3, FA_counter** – число случаев установки недостоверности при обмене с заданным узлом (OUTPUT – сброс);
- **Параметр = 4, Event_STS** – статус узла как статус события (при привязке переменной к CALL.ChGroupReq; исчезновение узла соответствует возникновению события – см. **Канал класса СОБЫТИЕ**):
 - 0 – статус узла неизвестен;
 - 1 – узел исчез, и это не квитируется;
 - 2 – WORK или TRACE;
 - 3 – узел исчез, и это квитируется.

Атрибут 252 – время исчезновения узла.

Атрибут 253 – время обнаружения узла.

Атрибут 251 – время последнего обмена с узлом.

Для квитирувания нужно выполнить запись (или через атрибут 122);

- **Параметр = 5, IP_Address** – адрес для обмена по UDP/IP;
- **Параметр = 6, TCP_Address** – адрес для обмена по TCP/IP;
- **Параметр = 7, UnSync** – рассинхронизация (в миллисекундах). Вычисляется как разность **<текущее время данного узла> – <текущее время указанного узла>**.
- **Параметр = 8, RS_Number** – каждый из двух байтов указывает на один из двух COM-портов, через которые данный узел взаимодействует с указанным узлом (0 – COM1). Задание несуществующих портов (например, 0xFFFF) фактически запрещает обмен по RS;
- **Параметр = 9, Card_Number** – номер сетевого адаптера, через который данный узел взаимодействует с указанным узлом. Если адаптер автоматически выбирается WINDOWS, **Card_Number** всегда равно 0;
- **Параметр = 10, FullStatus** – значение данной переменной индицирует полный статус узла:
 - биты 0-3 (0xF) – короткий статус узла (см. выше **Параметр= 0, ShortStatus**);
 - биты 4-6 (0x70) – ОС;
 - биты 7-11 (0xF80) – продукт TRACE MODE;
 - бит 12 (0x1000) – 1 – недостоверность при обмене по UDP;
 - бит 13 (0x2000) – 1 – недостоверность при обмене по TCP;
 - бит 14 (0x4000) – 1 – таймаут исчезновения из сети задан для узла вручную;
 - бит 15 (0x8000) – 1 – переключение адаптеров (если для от-

сылки/приема заданы разные адаптеры);

- бит 16 (0x10000) – 1 – блокировка обмена по сети вручную;
- бит 17 (0x20000) –
- бит 18 (0x40000) –
- бит 19 (0x80000) –
- биты 20 и 21 (0x300000) – режим:
 - 0 – инициализация узла;
 - 1 – узел найден;
 - 2 – узла нет;
 - 3 – переход из 2 в 1;
- бит 22 (0x400000) – переключение узла в WORK (используется в регистраторе);
- бит 23 (0x800000) – переключение узла в WORK (используется в регистраторе);
- биты 24-26 (0x7000000) – основной интерфейс обмена с узлом:
 - 0 – интерфейс не найден;
 - 1 – сеть;
 - 2 – сеть;
 - 3 – ;
 - 4 – SMS;
 - 5 – модем;
 - 6 – RS;
 - 7 – GPRS;
- бит 27 (0x8000000) –
- **Параметр=11, NodeLossByT** – таймаут перевода узла в исчезнувшие в отсутствие обмена с этим узлом;
- **Параметр = 12, T_Loss** – время исчезновения узла
- **Параметр = 13, T_Detection** – время обнаружения узла;
- **Параметр = 14, Interface_set** – обнаруженные виды обмена:
 - биты 0-7 – обмен по UDP:
 - бит 0 – автопосылка;
 - бит 1 – запрос CHCOPY;
 - бит 2 – CALL.ChGroupReq;
 - бит 3 – CALL.Vector;
 - бит 4 – зарезервировано;
 - бит 5 – зарезервировано;
 - бит 6 – запросы каналов консоли к другому узлу;

- бит 7 – экран запрашивает аргументы у другого узла;
- биты 8-15 – то же по TCP (автопосылки по TCP не существует, поэтому бит 8 всегда 0);
- биты старше 16 – служебные;
- **Параметр = 15, Interface_view** – основной вид обмена (биты соответствуют битам при **Параметр=14**).

@t_Seconds

Переходит в 14.17 **@t_Set_Date** типа INPUT с **Параметр=4**.

@t_Minutes

Переходит в 14.17 **@t_Set_Date** типа INPUT с **Параметр=5**.

@t_Hours

Переходит в 14.17 **@t_Set_Date** типа INPUT с **Параметр=6**.

@Mode_Control

Подтип и дополнение к подтипу – 14.4.

Данная переменная зарезервирована.

@Recalculation_Flag

Подтип и дополнение к подтипу – 14.5.

- **Параметр = 0** – если значение данной переменной типа OUTPUT равно 0, пересчет каналов узла с типом пересчета **флаги** (см. **Период пересчета канала**) разрешен, если 1 – запрещен;
- **Параметр > 0** – канал класса TIME, связанный с данной переменной, используется для заполнения списка значений времени, создаваемого монитором в памяти (см. **Период пересчета канала**).

@Recalculation_Index

Подтип и дополнение к подтипу – 14.6.

Параметр=0

Значение данной переменной типа INPUT увеличивается на 1 на каждом цикле CALC и сбрасывается в 1 при достижении значения 100. Используется для определения момента пересчета каналов узла, для которых задан тип пересчета **F1...F4** (см. **Период пересчета канала**).

Параметр=1

Время со старта (в секундах).

Параметр=2

IS_RECALC_MASK, только для разработчиков TRACE MODE.

Параметр=3

Зарезервировано.

@Data_from_SIAD

Подтип и дополнение к подтипу – 14.7.

Данная переменная типа OUTPUT выполняет различные функции по работе с архивами (в зависимости от своего значения). Для всех случаев значение атрибута **Параметр** переменной указывает номер архива **N** (0 – **System**, 1 – **SIAD1**, 2 – **SIAD2**, 3 – **SIAD3**).

Значение (**period**) нулевого байта переменной задает интервал (T_FROM, T_TO).

Значение (**snap**) битов 12-15 (0xF000) задает шаг разбиения интервала (T_FROM, T_TO):

- 0 – 1с
- 16#1000 – 5 с
- 16#2000 – 10 с
- 16#3000 – 15 с
- 16#4000 – 1 мин
- 16#5000 – 15 мин
- 16#6000 – 1 ч
- 16#7000 – 1 сутки

Вид экспорта зависит от значения **a** бита 8 и значения **b** битов 9-11 (полубайт 0xF00).

При **a=0** в генерируемые файлы экспортируются все данные из соответствующего архива в заданном интервале:

- **b=0** – экспорт в файл **siad_I<N>.txt**. Если файл существует, данные добавляются в него. Данные в файле сгруппированы по архивируемому каналу, каждая строка содержит дату, время и значение;
- **b=1** – данные по каждому архивируемому каналу экспортируются в отдельный файл **<имя архивируемого канала>.txt**. Если файл существует, данные добавляются в него. Каждая строка файла содержит дату, время и значение;

- **b=2** – аналог **b=1**, файлы перезаписываются при каждом экспорте;
- **b=3** – экспорт в файл **<имя канала @Data_from_SIAD>_N.txt**. Если файл существует, данные добавляются в него. Данные в файле сгруппированы по архивируемому каналу, каждая строка содержит имя архивируемого канала, дату, время и значение.

При **a=1** в генерируемые файлы экспортируются срезы из соответствующего архива (каждый экспорт содержит (интервал/шаг) строк):

- **b = 0** – экспорт в файл **siad_s<N>.txt**; данные по каналам представлены в файле в табличном виде. Если файл существует, данные добавляются в него;
- **b = 1** – аналог **b = 0**, но файл имеет имя **s.txt** и перезаписывается при каждом экспорте;
- **b = 3** – аналог **b = 0**, файл перезаписывается при каждом экспорте.

Указанные файлы создаются в папке узла. Во всех случаях выборка данных из архива производится только по тем каналам, которые хотя бы однажды пересчитаны после старта монитора. При **period=snap=a=b=0** экспорт не выполняется.

В текстовый файл экспортируются только сообщения об изменении атрибута **0, R** канала.

Если значение битов 8-11 переменной равно 8 (16#800), при ее обработке производится принудительная запись значений каналов в указанный архив. Запись производится по каналам, архивируемым в указанный архив, если их время изменения меньше, чем значение (**текущее время – сдвиг**), где **сдвиг** задается значением битов 12-15 (0xF000) значения переменной (см. выше). Время значения в архиве будет равно времени записи. Сдвиг и время значения в архиве можно задать с помощью ключа **SIADWRMT** в файле *.cnf (см. **Задание параметров работы мониторов**).

@Copy_SIAD

@Copy_SIAD с **Параметр=12..15** переходит в 15.0 **@e_SIAD** с **Параметр=12..15** (соответственно).

Имя канала, связанного с переменной **@Copy_SIAD**, не должно начинаться со знака **@**.

@Copy_AR

Переходит в 15.01 **@e_Alarm_Report** типа OUTPUT с **Параметр=7**.

@t_Operation_Time

Переходит в 14.17 **@t_Set_Date** типа INPUT с Параметр=2.

@Rights

Подтип и дополнение к подтипу – 14.11.

Данная переменная может иметь только тип INPUT и не может быть привязана к каналу CALL.ChGroupReq.

Данная переменная индицирует права текущего пользователя. Ниже показано соответствие битов переменной флагам редактора канала **Пользователь** (см. **Канал класса ПОЛЬЗОВАТЕЛЬ**):

- Бит 0 (0x1) – **Изменение** (раздел **Система**);
- Бит 1 (0x2) – **Выход** (раздел **Система**);
- Бит 2 (0x4) – **Останов** (раздел **Система**);
- Бит 3 (0x8) – **Редактирование** (раздел **Пользователи**);
- Бит 4 (0x10) – **Добавление** (раздел **Пользователи**);
- Бит 5 (0x20) – **Удаление** (раздел **Пользователи**);
- Бит 6 (0x40) – **Редактирование** (раздел **SCADA**);
- Бит 7 (0x80) – **Добавление** (раздел **SCADA**);
- Бит 8 (0x100) – **Удаление** (раздел **SCADA**);
- Бит 9 (0x200) – **Редактирование** (раздел **Дополнительно**);
- Бит 10 (0x400) – **Добавление** (раздел **Дополнительно**);
- Бит 11 (0x800) – **Удаление** (раздел **Дополнительно**);
- Бит 12 (0x1000) – **Квотирование** (раздел **Система**);
- Бит 13 (0x2000) – **Запуск** (раздел **Система**);
- Бит 14 (0x4000) – **Рассылка** (раздел **Логин**);
- Бит 15 (0x8000) – **Автоматическое завершение сеанса** (раздел **Логин**).

@Key_Code

Подтип и дополнение к подтипу – 14.12.

В Windows (в т.ч. Windows CE): в канале устанавливается соответствующий бит, если в файле *.cnf задан ключ **KBD_VIRTUAL<0...9>** = **<ASCII-код клавиши>** и клавиша нажата. По умолчанию (в т.ч. при отсутствии ключа) 0...3 соответствует **F1...F4**.

Если тип переменной – OUTPUT:

- если ia.i[0]=0 – посылка в канал ASCII-кода клавиши равнозначна нажатию и отжатию этой клавиши (например, ASCII-код клавиши F1 – 112);
- если ia.i[0]=1 или 2 – то же, но вместе с CTRL или ALT.

ia.i[0] – нулевое слово удаленного адреса, задается с помощью атрибута **Параметр** источника-приемника; в канале – атрибут (90,10).

@Calculate_Cycle

Подтип и дополнение к подтипу – 14.13.

В зависимости от ia.i[0], данная переменная индицирует/задает определенный параметр соответствующего потока (см. **Потоки монитора**).

Числа ниже – это значения ia.i[0], а в случае привязки переменной @Calculate_Cycle к каналу CALL.ChGroupReq – номера аргументов канала.

Параметр \ Цикл	CALC	IDLE	FAST	T_FACTORY
Время счета	0	4	8	12
Реальный цикл, мс	1	5	9	13
Заданный цикл, мс	2	6	10	14
Счетчик	3	7	11	15

- 16 – число превышений времени цикла CALC;
- 17 – последнее время цикла CALC в режиме превышения;
- 18 – время начала последнего превышения цикла CALC;
- 19 –

При ia.i[0]=2,6,10,14 числовой канал может иметь тип OUTPUT.

В данную переменную преобразуются следующие переменные (последнее число – ia.i[0], **Параметр**):

- **@Idle_Loop** => 14.13.5;
- **@Above** => 14.13.16;
- **@Calc_Loop** [0,1,5,6,9,10,18,19] => 14.13.[1,0,13,12,9,8,6,4].

Время циклов CALC, TF, IDLE и FAST может быть задано также в конфигурационном файле узла – см. **Файл CNF** в разделе **Задание параметров работы мониторов**.

@Debug

Подтип и дополнение к подтипу – 14.14.

Биты данной переменной, установленные в 1, индицируют (тип переменной INPUT) или задают (тип переменной OUTPUT) степень детализации информации, выводимой в протокол MPB:

Канал, связанный с переменной **@Debug**, должен быть класса HEX32.

Параметр=0, DEBUG

Для задания значения переменной могут использоваться ключи **DEBUG=**, **DEBUGON=**, **DEBUGOFF=** в файле *.cnf (см. **Задание параметров работы мониторов**).

- бит 0 (0x1) – каналы до загрузки;
- бит 1 (0x2) – воспроизведение звука;
- бит 2 (0x4) – зарезервировано;
- бит 3 (0x8) – SMS;
- бит 4 (0x10) – важнейшие критические ошибки;
- бит 5 (0x20) – временная статистика по графике:
 - **tcreate**=<время>(<ID экрана>) – максимальное время создания экрана;
 - **tdestroy**=<время>(<ID экрана>) – максимальное время уничтожения экрана;
 - **tupdate**=<время>(<ID экрана>) – максимальное время обновления экрана;
 - **tallupdate**=<время> – суммарное время обновления экранов;
 - **twait**=<время> – суммарное время недоступности экранов (экрану нельзя передать данные во время его рисования);
 - **tmax**=<время> – максимальное время графического цикла;
- бит 6 (0x40) – драйверы t12;
- бит 7 (0x80) – узлы в проекте (статусы, IP-адреса и т.п.);
- бит 8 (0x100) – устройства, для обмена с которыми используется универсальный механизм обмена с электросчетчиками;
- бит 9 (0x200) – обмен по RS (ключ **DBG_INFO_RS** в файле *.cnf);
- бит 10 (0x400) – обмен по сети;
- бит 11 (0x800) – SIAD;
- бит 12 (0x1000) – действия пользователя;
- бит 13 (0x2000) – установка флага (49, **DBG**) **Отладка**;
- бит 14 (0x4000) – потоки;
- бит 15 (0x8000) – ОТ, дампы и т.п.;
- бит 16 (0x10000) – ввод/вывод (чтение/запись) (дополнительная информация);
- бит 17 (0x20000) – протоколы, когда MPB – Slave по сети;
- бит 18 (0x40000) – SQL;
- бит 19 (0x80000) – OPC;

- бит 20 (0x100000) – разрешение ведения списка ошибок. Этот список выводится в том числе с помощью **SubNum**;
- бит 21 (0x200000) – каналы, которые включаются/выключаются при изменении статуса узла в резервированной системе;
- бит 22 (0x400000) – регистратор;
- бит 23 (0x800000) – зарезервировано;
- бит 24 (0x1000000) – принимаемые/посылаемые блоки по UDP;
- бит 25 (0x2000000) – принимаемые/посылаемые блоки по TCP;
- бит 26 (0x4000000) – DDE;
- бит 27 (0x8000000) – извлечение данных из архива (подробная информация);
- бит 28 (0x10000000) – шаблоны;
- бит 29 (0x20000000) – общая информация (номер версии, ОС и т.п.);
- бит 30 (0x40000000) – критические ошибки;
- бит 31 (0x80000000) – каналы, работающие по внешним протоколам (t12, Modbus и т.п.) (расширенная информация).

Параметр=1, ERROR_SHOW

Дополнительная отладочная информация:

Для задания значения переменной могут использоваться ключи **ERRSHOW=**, **ERROR_SHOW=**, **ERRSHOWON=**, **ERRSHOWOFF=** в файле *.cnf (см. **Задание параметров работы мониторов**).

- бит 0 (0x1) – о каналах CALL, извлекающих данные из архива (CALL.LocalStatistic и т.п., вне зависимости от флага **Отладка** в самом канале);
- бит 1 (0x2) –
- бит 2 (0x4) **INFO_NET** – о сети (см. **Сообщения при DBG_INFO_NET=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_NET=ON/OFF**;
- бит 3 (0x8) –
- бит 4 (0x10) – о перепривязке (см. **Сообщения при RE-LINK_INFO=ON/OFF**);
- бит 5 (0x20) – об изменении числа аргументов;
- биты 6 (0x40) – о реальном значении экрана;
- биты 7 (0x80) **NET_MARKER** – о сетевых маркерах (см. **Сообщения при DBG_INFO_MARKER=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_MARKER = ON/OFF**;

- биты 8 (0x100) – о **E_Meter** (см. **Универсальный механизм обмена с электросчетчиками**);
- биты 9 (0x200) **NODE_MARKER** – о маркерах узлов (основаны на наборах сетевых маркеров) (см. **Сообщения при DBG_INFO_NODE=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_NODE=ON/OFF**;
- бит 10 (0x400) – для HOST RS / TCP MODBUS / MITSUBISHI / OMRON: за время пересчета INPUT значение не считано;
- бит 11 (0x800) – о записи в SIAD (см. **Сообщения при ARCH_SYNC_DEBUG=ON/OFF**);
- бит12 (0x1000) – только для разработчиков TRACE MODE (при посылке в этот бит единицы в протокол профайлера выводятся счетчики потоков, и бит сбрасывается);
- бит13 (0x2000) – о приеме регистратором (от кого, куда и т.п.);
- бит14 (0x4000) – о вычислении CALL.Sum и CALL.RT_Statistics;
- бит15 (0x8000) – о каналах CALL.Vector;
- бит16 (0x10000) **INFO_TCP** – о встроенных сетевых протоколах (см. **Сообщения при DBG_INFO_TCP=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_TCP=ON/OFF**;
- бит 18 (0x40000) – о блоках синхронизации;
- бит 19 (0x80000) – о синхронизации;
- бит 20 (0x100000) – об архивах;
- бит 21 (0x200000) – о записи в архивы (сообщение, записываемое в архив, записывается и в протокол);
- бит 22 (0x400000) – о встроенных M-Link-протоколах (ищутся каналы, что принято/отослано);
- бит 23 (0x800000) – об обмене по SNMP (ключ **DBG_INFO_SNMP** в файле *.cnf);
- бит 24 (0x1000000) – о CALL.ChGroupReq с Параметр=10-13 и привязкой к каналу удаленного узла (см. **Канал CALL.ChGroupReq**);
- бит 25 (0x2000000) **INFO_IEC104** – см. **Сообщения при DBG_INFO_IEC104=ON**. Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_IEC104 = ON/OFF**;
- бит 27 (0x8000000) **INFO_SLAVE** – о SLAVE-протоколах (см. **Сообщения при DBG_INFO_SLAVE=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_INFO_SLAVE = ON/OFF**;
- бит 28 (0x10000000) **PEVE_TCP** – (см. **Сообщения при DBG_PEVE_TCP=ON**). Для установки/сброса этого бита в файле *.cnf предусмотрены ключи **DBG_PEVE_TCP = ON/OFF**;

- бит 31 (0x80000000) – разрешение записи в протокол профайлера информации об обмене (см. выше).

Параметр=2

Настройка протоколов MPV (<имя файла prj>_<порядковый номер узла>.txt и **tm6_log.txt**) и панели MPV (см. Профайлеры, Файл **tm6_log.txt** и Панель MPV):

Для задания значения переменной может использоваться ключ **GRAPH_STATUS_DISABLE** в файле CNF (а также ключи **GRAPH_STATUS_DISABLE = LOG** и **GRAPH_STATUS_DISABLE = PROF**).

- бит 0 (0x1) – сообщения tm6log.txt не направляются в окно «ПРОТОКОЛ 1» (ключ **GRAPH_STATUS_DISABLE=LOG** в CNF-файле);
- бит 1 (0x2) – сообщения <имя файла prj>_<порядковый номер узла>.txt не направляются в окно «ПРОТОКОЛ 1» (ключ **GRAPH_STATUS_DISABLE=PROF** в CNF-файле);
- бит 2 (0x4) – сообщения PROBLEMADD не направляются в окно «СИСТЕМНЫЕ СООБЩЕНИЯ 1»;
- бит 3 (0x8) – сообщения INFO не направляются в окно «ПРОТОКОЛ 2»;
- бит 4 (0x10) – сообщения tm6log.txt не направляются в файл;
- бит 5 (0x20) – сообщения <имя файла prj>_<порядковый номер узла>.txt не направляются в файл;
- бит 6 (0x40) –
- бит 7 (0x80) – сообщения INFO не направляются в файл <имя файла prj>_<порядковый номер узла>.txt;
- бит 8 (0x100) – сообщения PROBLEMADD не направляются в файл tm6log.txt;
- бит 9 (0x200) – сообщения PROBLEMADD не направляются в файл <имя файла prj>_<порядковый номер узла>.txt;
- бит 10 (0x400) – сообщения PROBLEMADD не направляются в файл;
- бит 11 (0x800) – сообщения PROBLEMADD не запоминаются в буфере;
- бит 12 (0x1000) – запрет записи в файл;
- бит 13 (0x2000) – вывод сообщений DBG_USER: 0 – в файл и в окно «ПРОТОКОЛ 1»; 1 – только в файл;
- бит 14 (0x4000) –
- бит 15 (0x8000) –

- бит 16 (0x10000) – 1 удаляет сообщения протоколов;
- бит 17 (0x20000) – 1 удаляет сообщения INFO;
- бит 18 (0x40000) – 1 удаляет сообщения PROBLEMADD;
- бит 19 (0x80000) – 1 удаляет сообщения DBG_USER;
- бит 20 (0x100000) – 1 удаляет сообщения PROBLEMADD из буфера;
- бит 21 (0x200000) –
- бит 22 (0x400000) –
- бит 23 (0x800000) – 1 удаляет все сообщения (очищаются окна «СИСТЕМНЫЕ СООБЩЕНИЯ 1», «СИСТЕМНЫЕ СООБЩЕНИЯ 2» и вкладки «ПРОТОКОЛ 1» и «ПРОТОКОЛ 2» (аналог команды «tmlog=clear» в панели MPB).

Параметр=3, STRESS

Искусственная нагрузка (останавливается посылкой 0 или по времени – см. **Параметр=7**; для задания теста может использоваться ключ **STRESS_VAL** в файле *.cnf):

В данную переменную переходит переменная **@RTM_Parameter** с **Параметр=80**.

- бит 0 (0x1) – читать каналы INPUT **STRESS_CLC** раз (см. **Параметр=4**);
- бит 1 (0x2) – читать каналы CALL **STRESS_CLC** раз;
- бит 2 (0x4) – читать каналы OUTPUT **STRESS_CLC** раз;
- бит 3 (0x8) –
- бит 4 (0x10) – автопосылка **STRESS_NET** раз (см. **Параметр=5**);
- бит 5 (0x20) – запрос по сети **STRESS_NET** раз;
- бит 6 (0x40) – отсылка по сети **STRESS_NET** раз;
- бит 7 (0x80) – загрузка 100% цикла пересчета;
- бит 8 (0x100) – не посылать команду сброса сторожевого таймера;
- бит 10 (0x400) – нагрузка RS (всегда «готов»);
- бит 11 (0x800) – нагрузка интерфейса t12;
- бит 12 (0x1000) – занять 4Гб ОЗУ(в Windows; в Windows CE – 1Гб). По окончании теста память освобождается;
- бит 13 (0x2000) – создать 1024 bitmap-изображения (не удаляются по окончании);
- бит 13 (0x4000) – создать 1024 окна (не удаляются по окончании);
- бит 14 (0x8000) – освободить память, занятую по биту 12 (0x1000);
- бит 16 (0x10000) – 1 раз в 5с переключать всплывающие экраны (останавливается посылкой 0);

- бит 17 (0x20000) – 1 раз в 5с переключать экраны (останавливается посылкой 0);
- бит 18 (0x40000) – включить статистику экранов (останавливается посылкой 0).

Параметр=4

Значение **STRESS_CLC** (может быть задано ключом **STRESS_CLC** в файле *.cnf).

В данную переменную переходит переменная **@RTM_Parameter** с **Параметр=81**.

Параметр=5

Значение **STRESS_NET** (может быть задано ключом **STRESS_NET** в файле *.cnf).

В данную переменную переходит переменная **@RTM_Parameter** с **Параметр=82**.

Параметр=6

Значение переменной задает процент загрузки цикла пересчета (может быть задано ключом **STRESS_THC** в файле *.cnf).

В данную переменную переходит переменная **@RTM_Parameter** с **Параметр=86**.

Параметр=7

Время, на которое создается искусственная нагрузка (10с по умолчанию).

@Net_DDE

Подтип и дополнение к подтипу – 14.15.

Данная переменная индицирует (INPUT) или задает (OUTPUT) параметры обмена. Биты значения переменной соответствуют следующим функциям (если значение бита равно 0, функция разрешена, 1 – запрещена):

- 0 (0x1) – автопосылка в сеть;
- 1 (0x2) – собственные запросы SMCOPY чтения (см. **Связь канал-канал**);
- 2 (0x4) – собственные запросы SMCOPY записи;
- 3 (0x8) – прием сетевых автопосылок;

- 4 (0x10) –
- 5 (0x20) –
- 6 (0x40) – передача по DDE;
- 7 (0x80) – прием по DDE от внешних приложений;
- 8 (0x100) – изменение входов каналов по DDE;
- 9 (0x200) – изменение границ каналов по DDE (кроме верхнего и нижнего пределов);
- 10 (0x400) – изменение верхнего и нижнего пределов каналов по DDE;
- 11 (0x800) – изменение других изменяемых атрибутов каналов по DDE;
- 12 (0x1000) –
- 13 (0x2000) –
- 14 (0x4000) – внешние запросы CHCOPY чтения/записи;
- 15 (0x8000) – прослушивание модемом входящих вызовов.

@t_Set_Time

При привязке переменной к каналу CALL.ChGroupReq в аргументы (их может быть до 18-ти) и атрибуты канала записываются производные значения временной метки, которая определяется по следующим правилам:

- если **cgr.A** <> 0, временная метка равна **cgr.A**;
- если **cgr.A** = 0, временная метка равна текущему времени.

Перечисленные ниже аргументы и атрибуты содержат значения в формате DATE_AND_TIME (за исключением **arg16** и **arg17**):

- **arg0** – собственно временная метка;
- **arg1** – начало дня;
- **arg2** – начало предыдущего дня;
- **arg3** – начало следующего дня;
- **arg4** – начало текущего месяца;
- **arg5** – начало предыдущего месяца;
- **arg6** – начало следующего месяца;
- **arg7** – начало текущего года;
- **arg8** – начало предыдущего года;
- **arg9** – время перехода на летнее время;
- **arg10** – время перехода на зимнее время;
- **arg11** – время, образованное следующим образом: год и месяц – из **arg0**, день задается атрибутом (98, **C5**);
- **arg12** – аналог **arg11**, предыдущий месяц;
- **arg13** – начало дня год назад;

- **arg14** – время начала сессии (или старта MPB);
- **arg15** – время окончания сессии;
- **arg16** – если тип данных этого аргумента STRING, он содержит трехбуквенную аббревиатуру текущего месяца;
- **arg17** – если тип данных этого аргумента STRING, он содержит трехбуквенную аббревиатуру дня недели, которому соответствует текущий день;
- 94, **C1** – тип текущего дня (1 – рабочий день, 2 – выходной);
- 95, **C2** – номер недели с начала года;
- 96, **C3** – день недели, которому соответствует текущий день (0 – воскресенье, 1 – понедельник и т.д.).

Если тип переменной – INPUT:

- **Параметр** = 0 – переходит в 14.17 **@t_Set_Date** с **Параметр**=1;
- **Параметр** = 1 – переходит в 14.17 **@t_Set_Date** с **Параметр**=3.

@t_Set_Date

Подтип и дополнение к подтипу – 14.17.

Переменная INPUT привязывается к числовому каналу и возвращает:

- **Параметр**=0 – текущие дата и время;
- **Параметр**=1 – float-значение секунд текущего времени ОС с точностью до миллисекунды (в эту переменную переходит переменная **@t_Set_Time** INPUT с **Параметр**=0);
- **Параметр**=2 – число секунд со старта (в эту переменную переходит переменная **@t_Operation_Time**);
- **Параметр**=3 – время со спутника через GPS-приемник или число секунд от начала суток (в эту переменную переходит переменная **@t_Set_Time** INPUT с **Параметр**=1);;
- **Параметр**=4 – число секунд из значения текущего времени ОС (в эту переменную переходит переменная **@t_Seconds**);
- **Параметр**=5 – число минут из значения текущего времени ОС (в эту переменную переходит переменная **@t_Minutes**);
- **Параметр**=6 – число часов из значения текущего времени ОС (в эту переменную переходит переменная **@t_Hours**).

@Logging

Подтип и дополнение к подтипу – 14.18.

Биты значения данной переменной типа OUTPUT соответствуют следующим функциям (если значение бита равно 0, функция разрешена, 1 – запрещена):

- 0 (0x1) – запись во все архивы SIAD;
- 1 (0x2) – отправка сообщений по всем направлениям (**AR**, **G** и т.д.);
- 2 (0x4) – передача в регистратор;
- 3 (0x8) – запись в дампы;
- 4 (0x10) –
- 5 (0x20) – передача данных в терминал (при его наличии). В других случаях бит 5 влияет на формирование запроса к удаленному узлу. Если бит равен 0, узел ждет наличия удаленного узла в сети в течение 10с; если бит равен 1 – не ждет. Механизм генерации запроса:
 - узел есть – запрос посылается на полученный IP-адрес;
 - узла нет, но его IP-адрес задан в ИС – запрос посылается на заданный IP-адрес;
 - узла нет и его IP-адрес не задан в ИС – запрос не посылается;
- 9 (0x200) – открытие файла отчета тревог;
- 11 (0x800) –
- 12 (0x1000) – отправка сообщений по направлению **PRN**;
- 13 (0x2000) – отправка сообщений по направлению **G**;
- 14 (0x4000) – 1 – разрешена отправка сообщений об изменении атрибутов каналов по всем направлениям; 0 – разрешена отправка сообщений об изменении атрибутов каналов только по направлению **AR**;
- 15 (0x8000) – отправка сообщений по направлению **GSM**.

@RS_on_off

Подтип и дополнение к подтипу – 14.19.

Данная переменная представляет собой набор 32-битных переменных, которые индицируют (INPUT) или задают (OUTPUT) параметры обмена по последовательным портам. Номер бита значения соответствует номеру СОМ-порта (0 – СОМ1 и т.д.). В набор входят следующие переменные:

- **Параметр=0** – OUTPUT – ручное включение (бит равен 0) или отключение (бит равен 1); INPUT – индикация;
- **Параметр=1** – 1, если поток работает (только чтение);
- **Параметр=2** – 1, если к порту физически должен быть подключен модем;
- **Параметр=3** – 1, если к порту подключен модем (только чтение);
- **Параметр=4** – 1, если SMS активизирована (только чтение);
- **Параметр=5** – 1, если порт не открыт;

- **Параметр=6** – 1, если порт закрыт в режиме TRACE (ключ **RS_SWTI_TRACE**=<число> в файле *.cnf);
- **Параметр=7** – 1, если поток в режиме TRACE не работает, а порт открыт;
- **Параметр=8** – значение из 15.14. OUTPUT: значение задает номер COM-порта на переинициализацию (1 – COM1); INPUT: номер последнего переинициализированного порта;
- **Параметр=15** – 1, если на RS включена эмуляция (ключ **RS_GOAL_EMULATE**=<0 или 1> в файле *.CNF).

Запреты, устанавливаемые с помощью переменной **@RS_on_off**, а также с помощью битов 8 и 9 переменной **@Input_Output**, для COM-порта, настроенного на Slave M_Link, не работают, т.е. отменить ответ Slave M_Link на полученный им запрос нельзя.

@Input_Output

Подтип и дополнение к подтипу – 14.20.

Данная переменная индицирует (INPUT) или задает (OUTPUT) различные параметры обмена. Биты значения переменной соответствуют следующим функциям (если значение бита равно 0, функция разрешена, 1 – запрещена):

- 0 (0x1) – чтение DI;
- 1 (0x2) – чтение AI;
- 2 (0x4) – запись DO;
- 3 (0x8) – запись AO;
- 4 (0x10) – чтение по OPC;
- 5 (0x20) – запись по OPC;
- 6 (0x40) – пересчет каналов чтения с типом пересчета **цикл FAST** (см. **Период пересчета канала**);
- 7 (0x80) – пересчет каналов записи с типом пересчета **цикл FAST**;
- 8 (0x100) – чтение по RS;
- 9 (0x200) – запись по RS;
- 10 (0x400) – чтение по t12;
- 11 (0x800) – запись по t12;
- 12 (0x1000) –
- 13 (0x2000) – этот бит управляет атрибутом (8, **W**) **Подключение** всех каналов узла при переключении статуса узла (WORK/TRACE) (1 – отключает, 0 – подключает);
- 14 (0x4000) – этот бит управляет атрибутом (8, **W**) **Подключение**

каналов CALL, вызывающих шаблоны, при переключении статуса узла (WORK/TRACE) (1 – отключает, 0 – подключает);

- 15 (0x8000) – этот бит управляет атрибутом (8, **W**) **Подключение** каналов узла, связанных с источниками/приемниками, при переключении статуса узла (WORK/TRACE) (0 – отключает (значение по умолчанию), 1 – подключает).

Запреты, устанавливаемые с помощью переменной **@RS_on_off**, а также с помощью битов 8 и 9 переменной **@Input_Output**, для COM-порта, настроенного на Slave M_Link, не работают, т.е. отменить ответ Slave M_Link на полученный им запрос нельзя.

@IP_parameter

Подтип и дополнение к подтипу – 14.21.

В зависимости от значения атрибута **Параметр** (см. **sysdiag_ipparam.tmc**), данная переменная индицирует (INPUT) или задает (OUTPUT) параметры обмена узла по TCP/IP.

- **Параметр** = 0, **Card_Num** – число сетевых адаптеров.
- **Параметр** = 1, **Receive_Card** – номер сетевого адаптера приема.
- **Параметр** = 2, **Send_Card** – номер сетевого адаптера передачи.
- **Параметр** = 3, **Selected_Bridge** – установленные в 1 биты переменной задают используемый мост:
 - бит 0 (0x1) – первый заданный (**Параметр**=9);
 - бит 1 (0x2) – второй заданный (**Параметр**=10);
 - бит 2 (0x4) – первый резерв;
 - бит 3 (0x8) – второй резерв.

По умолчанию установлены биты 0, 2 и 3 (0xD).

- **Параметр** = 4, **WriteMask** – OUTPUT – маска записи, перенесена в **Параметр**=8 (биты 28-31).

INPUT:

- бит 0 (0x1) .. бит 7 (0x80) – равенство бита единице индицирует наличие проблем с соответствующим адаптером.

Равенство следующих битов 1 индицирует ошибку соответствующих сокетов прослушивания сети:

- бит 8 (0x100) – TCP-обмен (поток main_tcp);
- бит 9 (0x200) – TCP-обмен в регистраторе (поток log_main_tcp);
- бит 10 (0x400) – GPRS-обмен (поток gprs_main_tcp);
- **Параметр** = 5, **Ch_in_move_block** – номер канала в скользя-

щем блоке (см. **@Net_DDE**) при досылке по сети.

- **Параметр** = 6, **UDP_dT** – разность между текущим временем и временем последнего приема по UDP.
- **Параметр** = 7, **Logger_Mask** – зарезервировано.
- **Параметр** = 8, **IpTcpRedirect** – 1 – все каналы с обменом по TCP переводятся на обмен по UDP (перенаправление может конфигурироваться в CNF-файле с помощью ключей **IPTCP_DIR**, **IPTCP_DIR_ON** и **IPTCP_DIR_OFF**);
 - бит 0 (0x1) –
 - бит 1 (0x2) –
 - бит 2 (0x4) –
 - бит 3 (0x8) –
 - бит 4 (0x10) –
 - бит 5 (0x20) –
 - бит 6 (0x40) –
 - бит 7 (0x80) – удаленный SIAD;
 - бит 8 (0x100) – удаленные аргументы;
 - бит 9 (0x200) –
 - бит 10 (0x400) – запись в регистратор;
 - бит 11 (0x800) – экраны;
 - бит 12 (0x1000) –
 - бит 13 (0x2000) –
 - бит 14 (0x4000) –
 - бит 15 (0x8000) –
 - бит 16 (0x10000) –
 - бит 17 (0x20000) –
 - бит 18 (0x40000) –
 - бит 19 (0x80000) – синхронизация по TCP;
 - бит 20 (0x100000) –
 - бит 21 (0x200000) –
 - бит 22 (0x400000) –
 - бит 23 (0x800000) –
 - бит 24 (0x1000000) – CALL.ChGroupReq с **Параметр**=10-13 и привязкой к каналу удаленного узла (см. **Канал CALL.ChGroupReq**);
 - бит 25 (0x2000000) –
 - бит 26 (0x4000000) и бит 27 (0x8000000) –

Следующие биты перенесены из **Параметр** = 4, **WriteMask** – маска записи (биты 0 -3):

- бит 28 (0x10000000) – задает метод отправки сформированного блока в регистратор:
 - 0 – широковещательно;
 - 1 – индивидуально (только заданным регистраторам);
- бит 29 (0x20000000) – флаг RETRANSLATE. Если 1, ретрансляция разрешена, если 0 – запрещена;
- бит 30 (0x40000000) – 1 блокирует всю отправку по IP;
- бит 31 (0x80000000) – 1 – если используется системный адаптер и широковещательная отправка, послать 1 раз;
- **Параметр = 9, Node_Bridge0 – ordinal** первого моста.
- **Параметр = 10, Node_Bridge1 – ordinal** второго моста.
- **Параметр = 11, Node_Logger0 – ordinal** регистратора 0.
- **Параметр = 12, Node_Logger1 – ordinal** регистратора 1.
- **Параметр = 13, Node_Logger2 –**
- **Параметр = 14, UDP_StandBy** – таймаут перехода в спящий режим в отсутствие записи по UDP;
- **Параметр = 15, UDP_adapter** – индекс адаптера отправки по UDP (если задан, в противном случае – -1).
- **Параметр = 16, TCPReconnect** – таймаут переподключения в случае ошибки записи по TCP, должен быть не менее 5с.
- **Параметр = 17, TCPterminate** – таймаут разрыва соединения в отсутствие записи по TCP;
- **Параметр = 18, UDP_Qmax** – максимальный размер очереди записи по UDP;
- **Параметр = 19 – .**
- **Параметр = 20 – .**

- **Параметр = 31, CountOfIndBlock** – число одновременно заданных буферов для индивидуальных рассылок, по умолчанию – 256.

@Program

Подтип и дополнение к подтипу – 14.22.

Данная переменная зарезервирована.

@RTM_Parameter

Если ниже не указано особо, подтип и дополнение к подтипу – 14.23.

В зависимости от значения атрибута **Параметр** (см. **sysdiag_rtmparam.tmc**), данная переменная индицирует (INPUT) или задает (OUTPUT) параметры обмена узла и параметры событий.

- **Параметр = 1, SENDTIME** – период посылки сообщения о своем присутствии в сети (в секундах, по умолчанию – 10с). С этим периодом узел досылает значения каналов по **MoveNet** (см. **@Net_DDE**). Данный параметр может быть задан в конфигурационном файле узла – см. **Файл CNF** в разделе **Задание параметров работы мониторов**.
- **Параметр = 2, Node_UnPresent** – максимально допустимое число секунд с момента последнего приема от узла (по умолчанию – 20 с). При превышении считается, что узла в сети нет.
- **Параметр = 3, Status_Check** – период (в секундах) анализа полученной информации о состоянии узлов в сети, по умолчанию – 5с.
- **Параметр = 4, Wait_Start** – время прослушивания основным узлом сети для обнаружения резервов (в миллисекундах). Для резервов основного узла это время автоматически умножается соответственно на 2 и 3. Данная процедура выполняется при старте системы. По умолчанию **Wait_Start=1000мс**.
- **Параметр = 5, Wait_Chg_Sts** – величина **Wait_Chg_Sts * Wait_Start** определяет таймаут на автоматический переход узла TRACE в состояние WORK, если соответствующий узел WORK вышел из строя. По умолчанию **Wait_Chg_Sts = 4**;
- **Параметр = 10, Ack_after_off** – длительность короткого события в секундах (см. **Канал класса СОБЫТИЕ**). Значение по умолчанию – 60;
- **Параметр = 11, Ack_after_on** – таймаут квитирования события в секундах (см. **Канал класса СОБЫТИЕ**). Значение по умолчанию – 3600;
- **Параметр=12, SyncMask** – маска синхронизации узла (ключ **SYNC_MASK** в файле *.cnf); установленные в 1 биты переменной равнозначны установке следующих флагов в редакторе узла на вкладке **Отчет тревог/Дамп/Параметры** (см. **Задание параметров узла**):
 - Бит 0 (0x1) – **Синхронизация на старте**;
 - Бит 1 (0x2) – **Архивы по началу**;
 - Бит 2 (0x4) – **Архивы по концу**;
 - Бит 3 (0x8) – **Разрешение tm6_log**;
 - Бит 4 (0x10) – **Запрет непрерывной синхронизации глобальных переменных**;
 - Бит 5 (0x20) – **Глобальные переменные**;
 - Бит 6 (0x40) – **Дополнительные атрибуты**;
 - Бит 7 (0x80) – **Принудительная обработка каналов OUTPUT**;
 - Бит 8 (0x100) – **Время**;
 - Бит 9 (0x200) – зарезервировано;

- Бит 11 (0x800) – **Запись в архивы среза по всем каналам.**

При успешной синхронизации архивов бит 13 (0x2000) переменной **Sync_mask** INPUT принимает значение 1.

В эту переменную переходит переменная **@RTM_Parameter** с **Параметр=138**;

- **Параметр = 16, ECOL0** –
- **Параметр = 17, ECOL1** –
- **Параметр = 18, ECOL2** –
- **Параметр = 19, ECOL3** –
- **Параметр = 20, ECOL4** –
- **Параметр = 21, ECOL5** –
- **Параметр = 22, ECOL6** –
- **Параметр = 23, ECOL7** –
- **Параметр = 24, CHGATRXXX** –
- **Параметр = 25, G_DT_SEL_E** –
- **Параметр = 26, G_DT_SEL_P** –
- **Параметр = 27, G_DT_SEL_S** –
- **Параметр = 28, CHGATRNODE** –

В переменные с **Параметр=16 .. 28** переходят в переменные **@RTM_Parameter** с **Параметр=90 .. 102**.

- **Параметр = 59** –
- **Параметр = 60** –
- **Параметр = 61** –
- **Параметр = 62** –
- **Параметр = 63** –
- **Параметр = 64,**
- **Параметр = 65,**
- **Параметр = 66,**
- **Параметр = 67,**
- **Параметр = 68** –
- **Параметр = 69,**
- **Параметр = 70,**
- **Параметр = 71, Memory** – используемая память, МВ;
- **Параметр = 72, Memory Peak** – максимум использования памяти;
- **Параметр = 73, Swap** – использование файла подкачки;
- **Параметр = 74, Swap Peak** – максимум использования файла

подкачки;

- **Параметр** = 75 –
- **Параметр** = 76, **CPU** – загрузка ЦП;
- **Параметр** = 77, **PRCMES** – см. **Файл CNF** в разделе **Задание параметров работы мониторов**;

Переменные с **Параметр**= 59 .. 77 переходят в переменные 15.20 **@e_Connect** с **Параметр**=0 .. 18 (соответственно).

- **Параметр** = 80, **stress** – переходит в переменную 14.14 **@Debug** с **Параметр**=3;
- **Параметр** = 81, **STRESS_CLC** – переходит в переменную 14.14 **@Debug** с **Параметр**=4;
- **Параметр** = 82, **STRESS_NET** – переходит в переменную 14.14 **@Debug** с **Параметр**=5;
- **Параметр** = 83,
- **Параметр** = 84,
- **Параметр** = 85,
- **Параметр** = 86, **STRESS_THC** – переходит в переменную 14,14 **@Debug** с **Параметр**=6;
- **Параметр** = 90, **ECOL0** –
- **Параметр** = 91, **ECOL1** –
- **Параметр** = 92, **ECOL2** –
- **Параметр** = 93, **ECOL3** –
- **Параметр** = 94, **ECOL4** –
- **Параметр** = 95, **ECOL5** –
- **Параметр** = 96, **ECOL6** –
- **Параметр** = 97, **ECOL7** –
- **Параметр** = 98, **CHGATRXXX** –
- **Параметр** = 99, **G_DT_SEL_E** –
- **Параметр** = 100, **G_DT_SEL_P** –
- **Параметр** = 101, **G_DT_SEL_S** –
- **Параметр** = 102, **CHGATRNODE** –

Переменные с **Параметр**=90 .. 102 переходят в переменные 14.23 **@RTM_Parameter** с **Параметр**=16 .. 28.

- **Параметр** = 137, **Max_Sound_Length** – переходит в 14.30 **@Sound_File** с **Параметр**=4;
- **Параметр** = 138, **Sync_mask** – переходит в 14.23 **@RTM_Parameter** с **Параметр**=12;
- **Параметр** = 179, **SMS_DLS** (ключ **SMS_DLS** в CNF-файле) – см.

Редактор параметров COM-порта;

- **Параметр** = 180, **SMS_TOR** – (ключ **SMS_TOR** в CNF-файле) – см. **Редактор параметров COM-порта**;
- **Параметр** = 181, **SMS_TCR** – (ключ **SMS_TCR** в CNF-файле) – см. **Редактор параметров COM-порта**;
- **Параметр** = 182, **SMS_RCN** – (ключ **SMS_RCN** в CNF-файле) – см. **Редактор параметров COM-порта**;
- **Параметр** = 183, **GSM_LOG** – переходит в 15.22 **@e_IO_Error** с **Параметр**=3;
- **Параметр** = 184, **SMS_Parameter** –
- **Параметр** = 185, **GPRS_NWN** –
- **Параметр** = 186, **GPRS_DIN** –
- **Параметр** = 187, **GPRS_SFR** –
- **Параметр** = 188, **GPRS_RZL** –
- **Параметр** = 189, **GPRS_RAL** –
- **Параметр** = 190, **GPRS_CWN** –

Переменная **@RTM_Parameter** с **Параметр**=179-190 переходит в переменную 15.22 **@e_IO_Error** с **Параметр**=0-11.

- **Параметр** = 191, **ANS_ZERO** – данная переменная переходит в переменную 15.00 **@e_SiAD** с **Параметр**=10;
- **Параметр** = 256, **CHKSTSE** –

@Message

Подтип и дополнение к подтипу – 14.24.

Данная переменная используется для генерации сообщений (см. **Генерация сообщений с помощью переменной MESSAGE**).

@AR_Length

Данная переменная с **Параметр**=0..2 переходит в переменную 15.26 **@q_Queue_Alarms** с **Параметр**=16..18.

@Modem_Step

Подтип и дополнение к подтипу – 14.26.

Атрибут **Параметр** переменной задает номер RS (1 – COM1) – как в числовом канале, так и в ChGroupReq.

В числовом канале:

- R – состояние обмена по заданному RS;

- C2 – код ошибки.

Биты атрибута (96, **C3**) (как числового канала, так и канала CALL.ChGroupReq) индицируют следующие параметры:

- бит 0 (0x1) – 1, если к заданному COM-порту подключен модем;
- бит 1 (0x2) – 1, если SMS активизирован на заданном COM-порту;
- бит 2 (0x4) – 1, если GPRS активизирован на заданном COM порту;
- бит 4 (0x10) – 1, если в системе есть хотя бы один модем;
- бит 5 (0x20) – 1, если в системе есть хотя бы один модем с активной SMS;
- бит 6 (0x40) – 1, если в системе есть хотя бы один модем с активной GPRS.

В ChGroupReq создается по 2 аргумента на порт:

- аргумент 1 – состояние обмена по заданному RS:
 - 0 – инициализация;
 - 1 – ожидание входящего звонка в режиме SLAVE;
 - 2 – попытка соединения в режиме SLAVE;
 - 3 – соединение в режиме SLAVE;
 - 4 – попытка соединения в режиме MASTER;
 - 5 – соединение в режиме MASTER;
 - 6 – в режиме SLAVE принят корректный запрос;
 - 7 – в режиме SLAVE сгенерирован ответ, и запись ответа прошла успешно;
 - 8 – реинициализация модема;
 - 9 – поиск каналов обмена в режиме MASTER;
 - 10 – ожидание поиска каналов обмена в режиме MASTER (см. описание ключа **MODEM_GOLNK** файла *.cnf в разделе **Обмен по телефонным линиям**);
- аргумент 2 – код ошибки:
 - в режиме MASTER:
 - 17 – нет ответа на исходящий звонок;
 - 3 – ошибка записи;
 - 4 – ошибка чтения;
 - в режиме SLAVE:
 - 18 – ошибка «снятия трубки»;
 - 41 – нет несущей (off-line);
 - 4 – некорректный запрос;
 - 7 – принято правильное число байт, но запрос некорректен;

3 – ошибка отправки ответа.

@Synchronize

Подтип и дополнение к подтипу – 14.27.

При отработке данная переменная типа OUTPUT корректирует время текущего узла на величину $0.5 * \delta T$ (см. формулу ниже). Для отработки нужно присвоить переменной ненулевое значение; после отработки значение переменной автоматически сбрасывается в 0.

Для WINDOWS:

- **Параметр** = 0 или 1 – синхронизация не производится, если время рассинхронизации меньше периода пересчета базы каналов;
- **Параметр** = 2 или 3 – синхронизация не производится, если время рассинхронизации больше 0.5 мин или меньше 1 с;
- **Параметр** = 4 или 5 – синхронизация производится во всех случаях;
- **Параметр** = 6 или 7 – обнуление миллисекунд времени рассинхронизации, синхронизация не производится;
- **Параметр** = 16 или 17 – аналог 0/1, но в вычислении времени рассинхронизации не учитывается собственный узел.

В данную переменную типа INPUT записывается рассогласование времени с узлами в состоянии WORK/TRACE в миллисекундах (**T** – собственное время):

$$\delta T = \frac{1}{N} \sum_{i=1}^N (T - T_i)$$

Если **Параметр** = 1, анализируются только резервы; если **Параметр** = 0 – все узлы; если **Параметр** = 16, 17 – все узлы, кроме собственного.

Данная переменная не может быть привязана к CALL.ChGroupReq.

@Load

Подтип и дополнение к подтипу – 14.28.

Данная переменная зарезервирована.

@Sound

Подтип и дополнение к подтипу – 14.29.

Данная переменная типа OUTPUT воспроизводит звуковой сигнал на системном динамике. Байт 0 (0x00FF) задает время звучания от 10 мс с точностью 1 мс, байт 1 (0xFF00) – частоту от 100 Гц с дискретностью 10 Гц.

Данная переменная не может быть привязана к CALL.ChGroupReq.

@Sound_File

Подтип и дополнение к подтипу – 14.30.

Данная переменная типа OUTPUT со значением **n=1,...32000** воспроизводит звуковой файл **n.wav**.

Монитор ведет очередь воспроизведения звуковых файлов. Режим воспроизведения может быть задан с помощью атрибута **Параметр** переменной:

- **Параметр=0** – постановка в конец очереди, однократное воспроизведение (OUTPUT);
- **Параметр=1** – постановка в начало очереди без прерывания текущего воспроизведения, циклическое воспроизведение (OUTPUT);
- **Параметр=2** – постановка в начало очереди, прерывание текущего воспроизведения, однократное воспроизведение (OUTPUT);
- **Параметр=3** – постановка в начало очереди, прерывание текущего воспроизведения, циклическое воспроизведение (OUTPUT);
- **Параметр=4** – ограничение длительности воспроизведения заданного звукового файла, INPUT/OUTPUT. Посылка 0 в OUTPUT останавливает воспроизведение. Задается в секундах (проверка – 1 раз в минуту), значение по умолчанию – 600с. В эту переменную переходит переменная **@RTM_Parameter** с **Параметр = 137, Max_Sound_Length;**
- **Параметр=5** – размер очереди воспроизведения (INPUT/OUTPUT); посылка 0 в OUTPUT очищает очередь;
- **Параметр=6** – постановка в конец очереди, однократное воспроизведение (OUTPUT). Имя звукового файла задается в атрибуте 80, **CMNT;**
- **Параметр=7** – при циклическом воспроизведении – время воспроизведения, в противном случае – 0.

Допускается посылка следующих значений в переменную OUTPUT:

- 0 – прерывание текущего воспроизведения (очередь воспроизведения не очищается);
- -1 – полная остановка воспроизведения (очередь воспроизведения очищается).

Данная переменная типа INPUT индицирует текущий размер очереди воспроизведения.

@Exit

Подтип и дополнение к подтипу – 14.31.

- **Параметр=0** – определены следующие значения для отправки в канал OUTPUT:
 - 1 – останавливает MPB;
 - 5 – останавливает и выгружает MPB;
 - 256 – останавливает и выгружает MPB (Windows и Linux);
- **Параметр=1** – время, спустя которое MPB автоматически останавливается и выгружается;
- **Параметр=2** – счетчик времени, если **arg0>0**;
- **Параметр=3** – зарезервировано;

ia.i[0] – нулевое слово удаленного адреса, задается с помощью атрибута **Параметр** источника-приемника; в канале – атрибут (90,10).

Группа ДИАГНОСТИКА

Как правило, отправка положительного значения в переменную типа OUTPUT обнуляет значение аналогичной переменной-счетчика типа INPUT.

@e_SIAD

Подтип и дополнение к подтипу – 15.00.

В зависимости от атрибута **Параметр** (см. **sysdiag_siaddiag.tmc**), данная переменная типа INPUT индицирует результаты соответствующей диагностики архивов SIAD.

- Если **Параметр** равен номеру архива (0 – **System**; 1 – **SIAD1**; 2 – **SIAD2**; 3 – **SIAD3**), то значение переменной индицирует код ошибки соответствующего архива (см. **Коды диагностируемых ошибок**).
- **Параметр** = 4 .. 7, **qSystem_Lost** .. **qSIAD3_Lost** – данная переменная типа INPUT индицирует число сообщений, которые не удалось вставить в очередь на запись в архив, заданный атрибутом **Параметр** (4 – **System**; 5 – **SIAD1**; 6 – **SIAD2**; 7 – **SIAD3**).
В данную переменную переходит переменная **@q_SIAD_Lost** с **Параметр=0** .. 4 (соответственно).
- Если **Параметр** = 8, **Work Mask**, то установленные в 1 биты атрибута (91, 11) канала указывают на работоспособность соответствующих архивов (бит 0 – System, бит 1 – SAID1 и т.д.).
В данную переменную переходит переменная 15.00 **@e_SIAD** с **Параметр=128**.
- Если **Параметр** = 9, **Error Mask**, то установленные в 1 биты атрибута (92, 12) канала указывают на наличие критической ошибки соответствующих архивов (бит 0 – System, бит 1 – SAID1 и т.д.).

В данную переменную переходит переменная 15.00 **@e_SIAD** с **Параметр=129**.

- **Параметр=10, ANS_ZERO** – таймаут (в секундах) принудительной записи блока в архив, если запись в архив не ведется. Значение по умолчанию – 180с. Данная процедура выполняется только в том случае, если после ее предыдущего выполнения запись в архив возобновлялась (ключ **ANS_ZERO** или **ANSZERO** в CNF-файле – см. **Задание параметров работы мониторов**).

Запись в файл архива начинается с 10 записей, проверка – 1 раз в 3 минуты.

В данную переменную переходит переменная **@RTM_Parameter** с **Параметр=191**;

- **Параметр=12..15, copySystem .. copySIAD3** – создание/подключение копии архива (см. ниже; в данную переменную переходит переменная **@Copy_SIAD** с **Параметр=12..15**).
- **Параметр = 128, Work Mask** – данная переменная переходит в переменную 15.00 **@e_SIAD** с **Параметр=8**;
- Если **Параметр = 129, Error Mask** – данная переменная переходит в переменную 15.00 **@e_SIAD** с **Параметр=9**.

Параметр=12..15

Переменная **@e_SIAD** типа OUTPUT создает/подключает копию архива, номер которого задает атрибут **Параметр** переменной (12 – System, 13 – SIAD1, 14 – SIAD2, 15 – SIAD3).

В переменную **@e_SIAD** OUTPUT с **Параметр=12 .. 15** переходит переменная **@Copy_SIAD** OUTPUT с **Параметр=12 .. 15** (соответственно).

Копирование выполняется при (98, **C5**)=2:

- если бит 0 атрибута (91, **I1**) равен 1, копия архива создается и подключается;
- если бит 0 атрибута (91, **I1**) равен 0, копия архива создается, но не подключается;

Если бит 1 атрибута (91, **I1**) равен 1, то старая копия переименовывается, в противном случае – уничтожается.

Значение переменной определяет имя файла резервной копии и его размещение (аргумент DESTINATION – см. **Копирование архивов и отчета тревог**).

Код ошибки копирования записывается в атрибут (92, **I2**).

Подключение копий выполняется при (98, **C5**)=1.

Определены следующие значения для канала, связанного с переменной:

- 1 и 17 – подключение соответственно первой/второй копии архива с запоминанием в файле-копии;
- 2 и 18 – подключение соответственно первой/второй копии архива;
- 3 и 19 – отключение соответственно первой/второй копии архива;

Функции 2/18 и 3/19 не запоминаются в файле-копии.

- если функция выполнена успешно, значение канала обнуляется. В случае ошибки в канале устанавливается признак аппаратной неадекватности.

Для подключения экранов и документов ко второй копии нужно установить признак архивирования для соответствующих каналов CALL (41, AB = 1).

Если считать основной архив нулевой копией, то

<индекс SIAD> = <номер SIAD> + 4 * <номер копии>

CALL.LocalList, CALL.DifSnap, CALL.LocalStatistic и CALL.LocalQuick автоматически работают со второй копией архива.

Ошибка подключения копии архива записывается в атрибут (97, **C4**) канала.

@e_Alarm_Report

Подтип и дополнение к подтипу – 15.01.

В данную переменную типа INPUT записываются результаты диагностики отчета тревог:

- **Параметр** = 0 – файловая ошибка;
- **Параметр** = 1 – индекс последней строки в файле;
- **Параметр** = 2 – число потерянных сообщений из-за нехватки памяти или из-за ошибки записи в файл. В эту переменную переходит переменная **@q_Lost_Alarms** INPUT;
- **Параметр** = 3 – при установке битов значения запрещаются следующие направления (см. **Редактор словарей сообщений**):
 - бит 1 – **AR** (задается только в ИС);
 - бит 2 – **G**;
 - бит 4 – передача в набор **toCash** (см. **@q_Lost_Alarms**);
 - бит 8 – **Net**;
 - бит 10 – набор **toGlist**;
 - бит 20 – прием из набора **toCash**;
- **Параметр** = 4 – биты значения управляют следующими функциями:

- бит 0 – 1 разрешает запрос ОТ на старте;
- бит 1 – 1 разрешает ответ на стартовый запрос ОТ;
- бит 2 – 1 глобально разрешает отсылку сообщений ОТ в сеть;
- бит 3 – 1 разрешает отсылку сообщений ОТ в сеть в состоянии WORK;
- бит 4 – 1 разрешает отсылку сообщений ОТ в сеть в состоянии TRACE;
- бит 5 – 1 разрешает обработку полученного сообщения ОТ (как в состоянии WORK, так и в состоянии TRACE);
- бит 6 – этот бит задает метод отсылки сообщения ОТ с признаком отсылки в сеть:
 - 0 – широковещательно;
 - 1 – индивидуально узлам, заданным в CNF-файле (см. описание ключа **ALANET_SEND_TONODE**);
- **Параметр = 7** – данная переменная типа OUTPUT создает резервную копию отчета тревог. Значение переменной определяет имя файла резервной копии и его размещение (аргумент DESTINATION – см. **Копирование архивов и отчета тревог**). Если значение бита 8 (0x100) переменной установить равным 1, то отчет тревог будет копироваться каждый раз при переполнении. В данную переменную переходит переменная **@Copy_AR**.

@e_Logger

Подтип и дополнение к подтипу – 15.02.

Значение переменной содержит информацию, зависящую от атрибута **Параметр**:

- **Параметр=0**:
 - если узел является регистратором – число активных подключений узлов (именно к регистратору, а не просто к узлу регистратора);
 - если узел не является регистратором – статус регистратора как статус события (см. **@Status** с **Параметр=4**);
- **Параметр=1** – ошибка:
 - 3 – не найден адаптер для отсылки в регистратор;
 - 5 – ошибка добавления в список для отсылки регистратору;
 - >10000 – WSA-ошибка;
- **Параметр=2** – время ошибки, индицируемой при **Параметр=1**;
- **Параметр=3** – маска работы регистраторов:
 - биты 0-3 – регистратор существует и находится в состоянии WORK;

- биты 4-8 – регистратор существует и находится в состоянии WORK или TRACE;
- **Параметр=4** – общее число принятых или переданных значений;
- **Параметр=5** – число принятых или переданных блоков;
- **Параметр=6** – число удаленных значений;
- **Параметр=7** – последнее время отправки регистратору или приема от регистратора;
- **Параметр=8** – текущий размер блока в списке для отправки регистратору;
- **Параметр=9** – среднее число значений в отсылаемом или принимаемом блоке;
- **Параметр=10** – **LOG_CHECK_TIME** (см. **Архивирование в регистратор**);
- **Параметр=11** – **LOG_SEND_SLOW** (см. **Архивирование в регистратор**);
- **Параметр=12** – число ошибок отправки блоков или распаковки принятых блоков;
- **Параметр=13** – число ошибок записи в список;
- **Параметр=14** – время первого сообщения в списке для отсылки регистратору;
- **Параметр=15** – предельный размер списка;
- **Параметр=16**:
 - бит 0 – отсылка в регистратор разрешена (0)/запрещена (1) (аналог ключа **LOG_SEND** в файле *.cnf – см. **Архивирование в регистратор**). Бит 0 не запрещает запись данных в очередь отсылки в регистратор;
 - бит 1 – если 1, данные не записываются в очередь отсылки в регистратор;
- **Параметр=20** – **LOG_STORY_TIME** (см. **Архивирование в регистратор**);
- **Параметр=22** – **LOG_SEND_TIME** (см. **Архивирование в регистратор**);
- **Параметр=23** – время начала следующего сеанса обмена с регистратором (увеличивается в случае успешной отсылки). Если отсылка была неуспешной, переменная сохраняет свое значение.

@Net_Code

Подтип и дополнение к подтипу – 15.04.

В данную переменную типа INPUT записываются коды ошибок обмена по IP операционной системы.

@e_M_LINK_Host

Подтип и дополнение к подтипу – 15.05.

В данную переменную типа INPUT записывается код ошибки при обмене по протоколу M-LINK в режиме MASTER (см. **Коды диагностируемых ошибок**).

@e_DCS

Подтип и дополнение к подтипу – 15.06.

В данную переменную типа INPUT записывается код ошибки при обмене по протоколу DCS (см. **Коды диагностируемых ошибок**).

@e_MODBUS

Подтип и дополнение к подтипу – 15.07.

Переменная (INPUT) содержит следующие параметры диагностики обмена по MODBUS по RS:

- **0** – <номер RS><код ошибки>;
- **1** – <код ошибки от устройства><номер устройства>;
- **2** – время первой ошибки;
- **3** – предыдущий <код ошибки от устройства><номер устройства>.

@e_PLC1Type

Подтип и дополнение к подтипу – 15.08.

Если **Параметр**=0, байт 1 данной переменной типа INPUT индицирует номер COM-порта, байт 0 — код ошибки при обмене через драйвер t11 (см. **Коды диагностируемых ошибок**).

Если **Параметр**=n (n>0), переменная индицирует код ошибки по COM<n>.

@e_PLC2Type

Подтип и дополнение к подтипу – 15.09.

В данную переменную типа INPUT записывается код ошибки при обмене через драйвер t12 (см. **Коды диагностируемых ошибок**).

@Above

Переходит в 14.3 **@Calculate_Cycle** INPUT с **Параметр**=16.

@Modem

Подтип и дополнение к подтипу – 15.11.

Атрибут **Параметр** переменной задает номер RS (1 – COM1).

Числовой канал

Значение 1 бита реального значения канала INPUT индицирует, что к соответствующему порту подключен модем, и этот модем находится в нужном режиме; в противном случае значение бита – 0.

Аппаратное значение – команда:

- 3 – реинициализировать модем с заменой телефона(из кодировки);
- 2 – реинициализировать модем;
- 1 – очистить список отсылки и ошибки.

Биты атрибута (95, **C2**) индицируют активность потоков SMS безотносительно RS (с SMS работают 8 RS – 4 основных и 4 дополнительных).

Биты атрибута (96, **C3**) индицируют следующие параметры:

- бит 0 (0x1) – 1, если к порту подключен модем;
- бит 1 (0x2) – 1, если SMS активизирован;
- бит 4 (0x10) – 1, если в системе есть хотя бы один модем;
- бит 5 (0x20) – 1, если в системе есть хотя бы один модем с активной SMS;
- бит 6 (0x40) – 1, если в системе есть хотя бы один модем с активной GPRS.

Канал CALL.ChGroupReq

Атрибуты канала используются так же, как в числовом канале. Кроме того, используются аргументы:

- **arg0** – 1 в случае ошибки RS (-> последний код ошибки);
- **arg1** – время ошибки RS;
- **arg2** – 1 в случае ошибки модема (-> последний код ошибки);
- **arg3** – время ошибки модема;
- **arg4** – 1) при использовании SMS – счетчик полученных SMS 2) для остальных назначений порта – счетчик запросов;
- **arg5** – 1) при использовании SMS – счетчик отправленных SMS; 2) для остальных назначений порта – счетчик ошибок;
- **arg6** – 1) при использовании SMS – счетчик ошибок отправки SMS; 2) для остальных назначений (исключая назначения **Card/Bar Reader** и **GPS**) – если в канале должен быть автоматически установлен флаг **Отработать** (для обмена; обмен по RS –

асинхронный), а этот флаг уже установлен, то **arg6** увеличивается на 1;

- **arg7** – 1) при использовании SMS – очередь SMS (очередь TRACE MODE + внешняя очередь); 2) для остальных назначений (исключая назначения **Card/Bar Reader** и **GPS**) – среднее время, затрачиваемое на все транзакции;
- **arg8** – 1) время отправки последнего SMS; 2) среднее время, затрачиваемое на одну транзакцию;
- **arg9** – 1) время получения последнего SMS; 2) время последней транзакции.

@SIAD_Synchronize

Подтип и дополнение к подтипу – 15.12.

Данная переменная зарезервирована.

@e_TCP_ModBus

Подтип и дополнение к подтипу – 15.13.

Если **Параметр**=0, в данную переменную типа INPUT записывается код ошибки при обмене по протоколу MODBUS TCP/IP (см. **Коды диагностируемых ошибок**).

Если **Параметр**<>0, он задает номер устройства MODBUS TCP/IP, для которого данная переменная типа INPUT диагностирует ошибки:

- 2, **In** – ошибка WINDOWS обмена по TCP (DEC, >=10000);
- 95, **C2** – ошибка устройства; если бит 7 (0x80) не установлен – ошибка, генерируемая самим устройством, если установлен – ошибка, генерируемая MPB:
 - 0x80 – ошибка соединения;
 - 0x81 – соединение с устройством принудительно разорвано;
- 96, **C3** – число неудачных попыток соединения;
- 97, **C4** – число очисток приемного буфера; буфер принудительно очищается в случае некорректного ответа устройства на запрос (например, если ответ содержит неверное число байтов);
- 98, **C5** – разница между текущим временем ОС и последним обменом (в секундах, <=255).

При посылке значений в данную переменную типа OUTPUT выполняются следующие команды (номер устройства задается атрибутом **Параметр**<>0):

- 3, 5 или 7 – заблокировать обмен с устройством (разрыв соединения);
- 2 – сбросить блокировку обмена;

- 1 – сбросить ошибку TCP;
- 16 – реинициализировать обмен;
- 17 – реинициализировать обмен с инвертированием битов байта XXX.xxx.xxx.xxx IP-адреса;
- 18 – реинициализировать обмен с инвертированием младшего бита байта xxx.xxx.XXX.xxx IP-адреса;
- 19 – реинициализировать обмен с инвертированием 2 младших битов в байтах xxx.xxx.XXX.XXX IP-адреса.

@RS_Reinit

Подтип и дополнение к подтипу – 15.14.

Значение байта 0 (0x00FF) переменной типа OUTPUT задает номер последовательного порта (1 – COM1 и т.д.). Отличное от 0 значение байта 1 (0xFF00) инициализирует обмен через порт. При нулевом значении старшего байта реинициализируется указанный порт, инициализированный ранее.

@e OPC

Подтип и дополнение к подтипу – 15.15.

Установленный бит переменной типа INPUT с **Параметр**=0 означает, что соответствующий OPC-сервер находится не в состоянии 1 и не в состоянии 2.

Атрибут 0, **R** канала, связанного с переменной типа INPUT с **Параметр**<>0 индицирует состояние соответствующего сервера:

- 0 – не подключен;
- 1 – подключается;
- 2 – подключен, нормальная работа;
- 6 – ошибка подключения;
- 17 – подключен, но в процессе обмена возникают ошибки.

Если переменная имеет тип OUTPUT и **Параметр**=0, то посылка в нее ненулевого числа переинициализирует все связи с OPC-серверами. При этом, если в начальный момент времени сервер не был найден, то надо использовать полную реинициализацию (значение больше 255). Если связь была оборвана уже при работе, то можно провести частичную реинициализацию (значение меньше 255).

При посылке ненулевого значения в переменную типа OUTPUT с **Параметр**<>0 выполняется запуск (переинициализация) соответствующего сервера.

Привязка к CALL.ChGroupReq

Переменная @e_OPC может быть привязана к каналу CALL.ChGroupReq, в котором создаются 4 аргумента для каждого OPC-сервера:

- **arg0** – статус;
- **arg1** – число ошибок;
- **arg2** – время старта сессии;
- **arg3** – время окончания сессии.

Запись в аргумент ненулевого числа приводит к реинициализации соответствующего сервера.

Установленный бит аппаратного значения (1, **A**) канала означает, что соответствующий OPC-сервер не работает.

При нормальной работе OPC-серверов реальное значение (0, **R**) равно 0. Если хотя бы один сервер не работает, (0, **R**) равно 1.

@Redundant

Подтип и дополнение к подтипу – 15.16.

При привязке **@Redundant** к CALL.ChGroupReq установленные биты CALL.ChGroupReq.R означают следующее:

- бит 0 – проблемы с сетью;
- бит 1 – проблемы с дублированным RS;
- бит 2 – не проходит синхронизация данных;
- бит 3 – зарезервировано.

Переменная содержит один или несколько параметров, в зависимости от своего атрибута **Параметр**:

- **Параметр=0, status** – статус данного узла (см. **@Status** с **Параметр=0**).

Если резервные узлы WORK и TRACE связаны по сети, то при изменении с помощью **@Redundant** статуса узла (WORK=>TRACE или TRACE=>WORK) вначале ожидается завершение процесса синхронизации данных, затем узел WORK сообщает, что переключается, и после этого статусы узлов изменяются (нет никаких таймаутов).

Если резервные узлы связаны только по RS, по данному механизму возможно только переключение WORK=>TRACE;

- **Параметр=1, 32valOwn** – 32 значения данного узла (только для привязки к CALL.ChGroupReq);
- **Параметр=2, 32valOther** – 32 значения партнера (только для привязки к CALL.ChGroupReq);

- **Параметр=3, 128val** – 128 последних значений (вне зависимости от записавшего их узла), только для привязки к CALL.ChGroupReq;
- **Параметр=4, statusOther** – статус партнера;
- **Параметр=5, secFromExchange** – число секунд с момента последнего обмена с партнером;
- **Параметр=6, msSyncTrace** – число миллисекунд на синхронизацию в узле TRACE;
- **Параметр=7, msSyncWork** – число миллисекунд на синхронизацию в узле WORK;
- **Параметр=8, CycleLostOwn** – число собственных потерянных циклов синхронизации;
- **Параметр=9, CycleLostOther** – число потерянных циклов синхронизации у партнера;
- **Параметр=10, T_error** – время последней ошибки обмена;
- **Параметр=11, wsaExchErr** – WSA-код последней ошибки обмена:
 - 2 – из-за эмуляции ошибки;
 - 9 – сокет не создан;
- **Параметр=12, wsaConnErr** – код обмена:
 - 0 – норма,
 - 1 – старт обмена по времени;
 - 2 – нет памяти;
 - 3 – нет места в очереди (UDP);
 - 8 – синхронизировано не все;
- **Параметр=13, VALs_InSync** – среднее число значений в синхронизации;
- **Параметр=14, BlocksForSync** – число блоков, используемых для синхронизации;
- **Параметр=15, DataBlocksLost** – число потерянных блоков данных.
- **Параметр=16, rtSyncMask** – маска синхронизации в реальном времени (не на старте):
 - бит 0 (0x1) – 1 запрещает синхронизацию каналов CALL.Program;
 - бит 1 (0x2) – 1 запрещает синхронизацию каналов CALL.ChGroupReq;
 - бит 2 (0x4) – 1 запрещает синхронизацию других каналов CALL;
 - бит 3 (0x8) – 1 запрещает синхронизацию каналов CALL.TVC;
 - бит 4 (0x10) – 1 запрещает синхронизацию глобальных пе-

ременных программ;

- бит 5 (0x20) – 1 запрещает синхронизацию каналов CALL.Vector;
- бит 6 (0x40) – 1 запрещает синхронизацию каналов CALL.AS_DATA;
- бит 7 (0x80) –
- биты 8-10 (0x700) –
- бит 11 (0x800) – 1 запрещает в канале CALL синхронизацию аргументов 3-го вида (см. ниже);
- бит 12 (0x1000) – 1 запрещает в канале CALL синхронизацию аргументов 2-го вида (см. ниже);
- бит 13 (0x2000) – 1 – узел WORK не ищет данные для TRACE, разрешены только блоки 64/128 значений (см. **Параметр=1-3**);
- бит 14 (0x4000) – 1 – установленные биты значения переменной запрещают в узле TRACE обмен по встроенным протоколам по TCP/IP (соединение разрывается, запрещены даже блоки 64/128 значений – см. **Параметр=1-3**).
- бит 15 (0x8000) – 1 – узел TRACE не обрабатывает данные, посылаемые узлом WORK;

Каналы CALL синхронизируют вначале атрибуты, а затем – аргументы (в зависимости от их вида):

- аргумент 1 вида – аргумент без привязки. Такой аргумент синхронизируется всегда;
 - аргумент 2 вида – к аргументу привязан канал.
Если в привязанном канале установлен флаг **Синхр.** (см. описание атрибута 58 в разделе **Общие атрибуты каналов**), аргумент не синхронизируется (поскольку синхронизируется канал).
Если в канале не установлен флаг **Синхр.**, и тип канала – INPUT, аргумент синхронизируется (узел WORK посылает узлу TRACE значение привязанного атрибута канала);
 - аргумент 3 вида – к аргументу привязан аргумент. В этом случае синхронизация выполняется так же, как для вида 2, но анализируется канал, привязанный к привязанному аргументу;
- **Параметр=17, IP_built_Prot** – маска TCP/IP для встроенных протоколов (если все биты установлены, протоколы отключены);
 - **Параметр=18, RS_mask1** – маска RS (если все биты установлены, RS отключен);
 - **Параметр=19, TsyncLast** – время последнего действия по сети для синхронизации;

- **Параметр=20, ModeMaskTrace** – маска режимов для состояния TRACE (может быть задана с помощью ключа **DISABLE_IN_TRACE** в файле *.cnf):
 - бит 0 (0x1) – 1 запрещает отсылку каналам OUTPUT подтипов 1 и 2 (см. **Подтип 1** и **Подтип 2**);
 - бит 1 (0x2) – 1 запрещает отсылку каналам OUTPUT по сети;
 - бит 2 (0x4) – 1 запрещает отсылку каналам OUTPUT по RS;
 - бит 3 (0x8) – 1 запрещает отсылку каналам OUTPUT по встроенным протоколам по TCP/IP;
 - бит 4 (0x10) – 1 запрещает отсылку каналам OUTPUT по интерфейсу t12;
 - бит 5 (0x20) – 1 запрещает отсылку каналам OUTPUT по OPC;
 - бит 6 (0x40) – 1 запрещает обмен с OEM-контроллерами по RS;
 - бит 7 (0x80) – 1 запрещает обмен по **E_Meter** (см. **Универсальный механизм обмена с электросчетчиками**);
 - бит 8 (0x100) – 1 запрещает обмен по модему;
 - бит 9 (0x200) –
 - бит 10 (0x400) –
 - бит 11 (0x800) –
 - бит 12 (0x1000) –
 - бит 13 (0x2000) –
 - бит 14 (0x4000) –
 - бит 15 (0x8000) – 1 запрещает отсылку в регистратор в состоянии TRACE;
 - бит 16 (0x10000) – 1 отключает переключение TRACE/WORK для RS с назначением **(10) DRs_Master** (см. **Редактор параметров COM-порта**);
 - бит 17 (0x20000) – 1 запрещает запрос OT;
 - бит 18 (0x40000) – 1 запрещает запрос SIAD;
 - бит 19 (0x80000) –
 - бит 20 (0x100000) – 1 запрещает CHCOPY по сети (см. **Связь канал-канал**);
 - биты 21-26 (0x7E00000) –
 - бит 27 (0x8000000) – 1 запрещает отсылку сообщений OT в сеть;
 - бит 28 (0x10000000) –
 - бит 29 (0x20000000) – 1 запрещает узлу переход в состояние WORK, если узла WORK нет;

- бит 30 (0x40000000) – 1 сохраняет нахождение обоих узлов в статусе WORK (если такая ситуация имеет место);
- бит 31 (0x80000000) – 1 задает узлу статус TRACE (при старте);
- **Параметр=21, PR_LIST** – текущий размер буфера PRESERVE_LIST.
Если в течение времени, которое затрачивается на начальную синхронизацию узла TRACE, в узле WORK происходит архивирование значений каналов, эти значения не участвуют в процессе синхронизации, а записываются в буфер PRESERVE_LIST. По окончании начальной синхронизации буфер досылается узлу TRACE;
- **Параметр=22** –
- **Параметр=23, dTsync** – разность между текущим временем и временем последней синхронизации.

@e_Dump

Подтип и дополнение к подтипу – 15.17.

В данную переменную типа INPUT записываются результаты диагностики дампа узла (см. **Коды диагностируемых ошибок**).

@e_DDE

Подтип и дополнение к подтипу – 15.18.

В данную переменную типа INPUT записывается код ошибки при обмене по DDE (см. **Коды диагностируемых ошибок**).

@e_MLink_Slave

Подтип и дополнение к подтипу – 15.19.

В данную переменную типа INPUT записывается код ошибки при обмене по протоколу M-LINK в режиме SLAVE (см. **Коды диагностируемых ошибок**).

@e_Connect

Подтип и дополнение к подтипу – 15.20.

Эта переменная типа INPUT используется для индикации результатов проверки имеющихся ресурсов и памяти.

В переменные с **Параметр=0 .. 18** переходят в переменные **@RTM_Parameter** с **Параметр=59 .. 77**.

- **Параметр** = 0 –
- **Параметр** = 1 –
- **Параметр** = 2 –
- **Параметр** = 3 –
- **Параметр** = 4 –
- **Параметр** = 5,
- **Параметр** = 6,
- **Параметр** = 7,
- **Параметр** = 8,
- **Параметр** = 9 –
- **Параметр** = 10,
- **Параметр** = 11,
- **Параметр** = 12, **Memory** – используемая память, МВ;
- **Параметр** = 13, **Memory Peak** – максимум использования памяти;
- **Параметр** = 14, **Swap** – использование файла подкачки;
- **Параметр** = 15, **Swap Peak** – максимум использования файла подкачки;
- **Параметр** = 16 –
- **Параметр** = 17, **CPU** – загрузка ЦП;
- **Параметр** = 18, **PRCMES** – см. **Файл CNF** в разделе **Задание параметров работы мониторов**.

@q_SIAD_Lost

Данная переменная типа INPUT с **Параметр**=0 .. 3 переходит в переменную 15.00 **@e_SIAD** с **Параметр** = 4 .. 7 соответственно.

@e_IO_Error

Подтип и дополнение к подтипу – 15.22.

В данную переменную с **Параметр**=0-11 переходит переменная **@RTM_Parameter** с **Параметр**=179-190.

- **Параметр**=0 –
- **Параметр**=1 –
- **Параметр**=2 –
- **Параметр**=3, **GSM_LOG** – степень детализации отчета по GSM-или GPRS-обмену, который профайлер записывает в свой отладочный файл (ключ **GSM_LOG** в CNF-файле – см. **Задание параметров работы мониторов**). Установленные в 1 биты байта 0

переменной задают запись следующих сообщений:

- биты 0 и 1 – модемные сообщения об ошибках;
- бит 2 (0x4) – модемные предупреждения;
- бит 3 (0x8) – модемная информация об обмене.

Установленные в 1 биты байта 1 переменной задают запись сообщений MPB:

- бит 8 (0x100) – ошибки;
- бит 9 (0x200) – краткая информация об обмене;
- бит 10 (0x400) – расширенная информация об обмене.

По умолчанию устанавливаются биты 0 и 1 байта 0 (0x3).

В эту переменную переходит переменная **@RTM_Parameter** с **Параметр=183**;

- **Параметр=4** –
- **Параметр=5** –
- **Параметр=6** –
- **Параметр=7** –
- **Параметр=8** –
- **Параметр=9** –
- **Параметр=10** –
- **Параметр=11** –

@q_Lost_Alarms

При привязке к числовому каналу данная переменная переходит в переменную 15.01 **@e_Alarm_Report** с **Параметр=2** (число потерянных сообщений при записи в память и файл).

Привязка к CALL.ChGroupReq (допускается изменение в реальном времени):

- **arg0 .. arg29** – соответствуют направлениям из destination.tmc (см. **Редактор словарей сообщений**);
- **arg30** – что подчиталось из файла;
- **arg31** – сообщения в ОТ, для которых нет словаря. Каждый бит соответствует направлениям передачи сообщения:
 - бит 0 (0x1) – **AR**;
 - бит 1 (0x2) – **G**;
 - бит 2 (0x4) – набор **toCash** (в кэш: **Net, toDoc, toSync, toRoot**);
 - бит 3 (0x8) – набор **toGlist** (**GSM, PRN, toSQL, toStore, Play**);
 - биты 6 и 7 (0xC0) – **Play, PlayStop, PlayLoopStop**;

- бит 8 (0x100) – **Net**;
- бит 9 (0x200) – **toDoc**;
- бит 10 (0x400) – **toSync**;
- бит 11 (0x800) – **toRoot**;
- бит 12 (0x1000) – **GSM**;
- бит 13 (0x2000) – **PRN**;
- бит 14 (0x4000) – **toSQL**;
- бит 15 (0x8000) – **toStore**;
- бит 16 (0x10000) – **Play**.

@Idle_Loop

Переходит в 14.3 **@Calculate_Cycle** INPUT с **Параметр=5**.

@Graphics_Loop

Подтип и дополнение к подтипу – 15.25.

Переменная содержит следующие параметры:

- реальное время цикла потока 16 вызова графики (значение **@Graphics_Loop**) – см. **Потоки монитора**;
- максимальное время перерисовки;
- заданное время цикла потока 16.

Числовой канал с ia.i0 =2 может иметь тип OUTPUT.

@q_Queue_Alarms

Подтип и дополнение к подтипу – 15.26.

- **Параметр=0, 8 и 16** – соответственно позиция текущего сообщения ОТ, порядковый номер текущего сообщения (от создания ОТ) и максимальное число строк ОТ. В переменную с **Параметр=16** (INPUT/OUTPUT) переходит переменная **@AR_Length** с **Параметр=0**;
- **Параметр=1, 9 и 17** – то же, для **AR** (см. **Редактор словарей сообщений**). В переменную с **Параметр=17** (INPUT/OUTPUT) переходит переменная **@AR_Length** с **Параметр=1**;
- **Параметр=2, 10 и 18** – то же, для **toGlist** (см. 15.23 **@q_Lost_Alarms**). В переменную с **Параметр=18** (INPUT/OUTPUT) переходит переменная **@AR_Length** с **Параметр=2**;
- **Параметр=3, 11 и 19** – то же, для **G**;
- **Параметр=4, 12 и 20 (I_start)** – то же, при старте;

- **Параметр=5, 13 и 21 (l_cash)** – параметры полученного ОТ в кэше (текущая позиция, время начала, максимальное число строк);
- **Параметр=6 – 2 параметра:** счетчик принятых сообщений по сети и счетчик принятых сообщений по сети при старте;
- **Параметр=7 – 2 параметра:** счетчик сообщений в графику и количество сообщений, которое графика должна хранить у себя;
- **Параметр=24 .. 31** – счетчики потерянных сообщений по направлениям:
 - **Параметр=24 – AR;**
 - **Параметр=25 – toGlist;**
 - **Параметр=26 – G;**
 - **Параметр=27 – l_start** (см. выше);
 - **Параметр=28 – toCash** (не хватает памяти);
 - **Параметр=29 – ;**
 - **Параметр=30 – ;**
 - **Параметр=31** – общий счетчик потерь по всем направлениям, кроме **toCash**.

@Calc_Loop

Переходит в 14.3 **@Calculate_Cycle** со следующим соответствием атрибутов **Параметр**:

@Calc_Loop.[0,1,5,6,9,10,18,19] => 14.13.[1,0,13,12,9,8,6,4]

@q_IP_Lost

Данная переменная переходит в переменную 15.29 **@q_IP_Send_Q** с **Параметр=8**.

@q_IP_Send_Q

Подтип и дополнение к подтипу – 15.29.

- **Параметр=0** – переменная INPUT индицирует текущий размер очереди на отправку по UDP/IP (число пакетов);
- **Параметр=1** – переменная INPUT (OUTPUT) индицирует (задает) максимальный размер очереди на отправку по UDP/IP. Данный параметр может быть задан также с помощью ключа **QUEUE** в файле *.cnf;
- **Параметр=2** – текущий размер очереди приема по UDP/IP;
- **Параметр=3** – максимальный размер очереди приема по UDP/IP.
- **Параметр=4** – текущий размер очереди сервера TCP;
- **Параметр=5** – максимальный размер очереди сервера TCP;

- **Параметр=6** – текущий размер очереди на отправку по TCP/IP;
- **Параметр=7** – максимальный размер очереди на отправку по TCP/IP;
- **Параметр=8** – число потерянных пакетов при отправке по UDP/IP. В данную переменную переходит переменная 15.28 **@q_IP_Lost** с **Параметр=0**;
- **Параметр=9** – число потерянных пакетов при приеме по UDP/IP;
- **Параметр=10** – число потерянных пакетов при отправке по TCP/IP;
- **Параметр=11** – число потерянных пакетов при приеме по TCP/IP.

@q_SIAD_Q

Подтип и дополнение к подтипу – 15.30.

Атрибут **Параметр** этой переменной может принимать следующие значения (см. **sysdiag_queuesiad.tmc**):

- 0...11, **Queue System ... Queue SIAD3 Copy2** (функция 0)
- 16...27, **Lost System ... Lost SIAD3 Copy2** (функция 1)
- 32...43, **Used System ... Used SIAD3 Copy2** (функция 2)
- 48...59, **K System ... K SIAD3 Copy2** (функция 3)
- 64...75, **Count System ... Count SIAD3 Copy2** (функция 4)
- 80...91, **S System ... S SIAD 3 Copy2** (функция 5)
- 96...107, **First System...First SIAD 3 Copy2** (функция 6)
- 112...123, **Last System...Last SIAD 3 Copy2** (функция 7)

Значение переменной INPUT индицирует следующие характеристики:

- **Параметр=0...11** – размер текущей очереди на запись в соответствующий архив (в процентах к максимальному числу записей);
- **Параметр=16...27** – число потерянных сообщений для записи в соответствующий архив;
- **Параметр=32...43** – приблизительный процент заполнения соответствующего архива. При старте MPB значение этого параметра соответствует текущему уровню заполнения архива. После достижения значения 100 параметр обнуляется и затем снова растет до 100 по мере затирания старых записей новыми;
- **Параметр=48...59** – коэффициент полезного использования объема соответствующего архива (в процентах);
- **Параметр=64...75** – число сообщений, посланных в соответствующий архив с момента старта монитора;
- **Параметр=80...91** – число запросов на выборку из соответствующего архива в очереди;
- **Параметр=96...107** – время первой записи в соответствующем

архиве;

- **Параметр**=112...123 – время последней записи в соответствующем архиве.

Значения 48-59 и 64-75 атрибута **Параметр** обрабатываются только в профайлере.

В реальном времени номер выполняемой функции (0-7) и индекс архива (0-11) записываются соответственно в атрибуты (90, **I0**) и (92, **I2**) канала.

@Node_Lock

Подтип и дополнение к подтипу – 15.31.

Для удержания связи с узлом **N** при обмене через модем или по GPRS нужно присвоить значение **N+1** переменной **@Node_Lock** типа OUTPUT:

- **Параметр**=0 – по всем RS;
- **Параметр**=<номер RS> – по заданному RS (1 – COM1).

Канал класса ПОЛЬЗОВАТЕЛЬ

Каналы класса **Пользователь** создаются в группе **Пользователи ТМ** узла.

Логин, пароль и права пользователя задаются в соответствующих разделах редактора канала.

В процедурах авторизации/окончания сеанса участвуют каналы **Пользователь** без привязки.

Имя канала, логин и пароль должны иметь не менее 4 символов.

Задание прав пользователей

Раздел **Авторизация:**

- **Логин** – поле задания логина пользователя (не более 32 символов), (210) **Login**;
- **Пароль** – поле задания пароля пользователя (не более 32 символов), (211) **Password**.

Раздел **Логин:**

- **Рассылка** – этот флаг следует установить, если на мобильный телефон пользователя требуется отправлять сообщения отчета тревог;
- **Автоматическое завершение сеанса** – если этот флаг установлен, при остановке монитора логин и пароль не запрашиваются.

Раздел **Система:**

- **Запуск** – разрешение на запуск монитора;
- **Останов** – разрешение на останов монитора;
- **Выход** – разрешение на выход из программы;
- **Изменение** – разрешение на изменение параметров процесса;
- **Квитирование** – разрешение на квитирование (не зависит от разрешения на изменение).

Раздел **Пользователи:**

- **Редактирование** – разрешение на редактирование прав пользователей;
- **Добавление** – разрешение на добавление пользователей;
- **Удаление** – разрешение на удаление пользователей.

В разделе **Доступ** конфигурируется доступ к графическим экранам и функциям управления графических элементов.

- **Экраны** – установка битов атрибута (209) **Screen** и битов байта 0 (0x000000FF) атрибута 207, **Right_Mask** (см. ниже);
- **Формы** – установка битов атрибута (208) **Access** и битов байта 1 (0x0000FF00) атрибута 207, **Right_Mask** (см. ниже).

Раздел **SCADA**:

- **Редактирование** – разрешение на редактирование проекта АСУТП и изменение значений атрибутов из дерева компонентов;
- **Добавление** – разрешение на добавление объектов в проект АСУТП;
- **Удаление** – если этот флаг установлен:
 - разрешено удаление объектов из проекта АСУТП;
 - разрешено дерево компонентов в rtmg32.exe.

Раздел **Дополнительно**:

- **Редактирование** – зарезервировано;
- **Добавление** – если этот флаг установлен, доступ к архивам из графики запрещен;
- **Удаление** – если этот флаг установлен, то, независимо от прав пользователя, ему разрешен доступ к экранам и формам с кодом доступа 0.

16 флагов, для которых не указано соответствие атрибутам канала, формируют биты байтов 2 и 3 (0xFFFF0000) атрибута (207) **Right_Mask** (соответствие битов флагам аналогично соответствию для переменной **@Rights** – см. **Группа СИСТЕМНЫЕ**).

Для управления пользователями и индикации/задания их прав в реальном времени предусмотрены следующие средства:

- системная переменная **@Rights**;
- специфические атрибуты канала **Пользователь** (см. ниже);
- вкладка **Пользователи** панели МРВ (см. **Панель МРВ**);
- канал **Пользователь** INPUT, к которому привязан локальный или удаленный канал **Пользователь** (привязка служит только для идентификации узла). Канал с привязкой имеет следующие свойства:
 - имя канала, права, телефон и адрес электронной почты синхронизируются по авторизованному пользователю заданного узла. Права записываются, в том числе, в атрибут 9, **Q** (32 бита);
 - изменение битов 0-31 атрибута 9, **Q** приводит к изменению соответствующих прав авторизованного пользователя заданного узла. Для такого управления канал с привязкой должен

иметь права на редактирование пользователей;

- канал не участвует в процедуре авторизации;
- канал не отображается в стандартном объекте **8, Users**.

Необходимые права для выполнения функций:

- перезагрузка узла – **Редактирование, Добавление и Удаление** в разделе **SCADA**;
- открытие диалогов rtc.exe **Компоненты, Статус узлов и Пользователи** – **Редактирование, Добавление и Удаление** в разделе **SCADA**;
- изменение атрибутов – **SCADA/Редактирование** и **Система/Изменение**;
- изменение отображения в rtmg32.exe – **Редактирование и Удаление** в разделе **SCADA**;
- изменение привязок – **SCADA/Редактирование**.

Специфические атрибуты канала ПОЛЬЗОВАТЕЛЬ

Помимо атрибутов, общих для каналов всех классов (см. **Общие атрибуты каналов** и **Атрибуты каналов, отображаемые профайлером**), канал класса **Пользователь** имеет следующие специфические атрибуты (в реальном времени эти атрибуты можно изменять):

- 26, **Phone** – номер мобильного телефона;
- 27, **Email** – адрес электронной почты;
- 28, **Message** – при каждом вводе в этот атрибут текстовой строки она отправляется в виде SMS-сообщения на мобильный телефон пользователя. Чтение из этого атрибута невозможно.
Для отправки сообщения по электронной почте нужно ввести в атрибут строку «**email:**<сообщение>» (префикс **email:** указывать не нужно, если в файле *.spf задан ключ **EMAIL_ONLY**);
- 207, **Right_Mask** – права пользователя. Значение, посылаемое в этот атрибут, записывается байты 2 и 3 (0xFFFF0000) атрибута;
- 208, **Access** – права на использование функций управления ГЭ (битовая маска **Формы** раздела **Доступ** редактора, см. также **Функции управления ГЭ**). Эти права индицирует также байт 1 (0x0000FF00) атрибута 207, **Right_Mask**;
- 209, **Screen** – права на доступ к графическим экранам (битовая маска **Экраны** раздела **Доступ** редактора, см. также **Задание параметров графического экрана**). Эти права индицирует также байт 0 (0x000000FF) атрибута 207, **Right_Mask**;
- 210, **Login** – логин;

- 211, **Password** – пароль.

Контролирование действий пользователей

В TRACE MODE предусмотрен контроль следующих действий пользователей:

- посылка числового значения в канал из графики.

Для включения режима контроля в файле *.cnf нужно задать следующий ключ:

- **USER_CFR**=<число DEC>.

Значения ключа могут быть следующие: 1, 2, 3, 4, 32, 33, 34, 35. Значения, меньшие 32, предназначены для контроля действий всех пользователей, значения, большие или равные 32, – для контроля действий тех пользователей, у которых установлен флаг **Атрибуты** (44, **MB**).

Сообщения о контролируемых действиях записываются в файл, имя которого зависит от значения ключа (файл создается в папке проекта):

- 0 или 32 – **USER_CFR21.txt**;
- 1 или 33 – **USER_D**<день месяца>.txt;
- 2 или 34 – **USER_I**<ID пользователя в проекте>.txt;
- 3 или 35 – **USER_Y**<день года>.txt.

Формат сообщения:

<номер действия> <дата и время действия> <имя пользователя в проекте> <посланное значение> (<ID канала вызова экрана> →[<ID канала, в который значение записано>] <имя канала>.<номер атрибута>)
{<N1>}{<N2>}{<N3>}{<N4>}{<N5>}

Нумерация контролируемых действий – сквозная (со старта МРВ вне зависимости от пользователя).

Числа N1, N2, N3, N4 и N5 – это цифровая подпись сообщения.

Для вывода данных из файлов используется **SubNum=560**:

- **USER_CFR=0** или 32 – выводятся все сообщения;
- **USER_CFR=2** или 34 – выводятся сообщения о действиях текущего пользователя.

Формат вывода сообщения:

<результат верификации>=<сообщение без цифровой подписи>

В качестве результата верификации выводится одно из следующих строковых значений:

- **FALSE** – сгенерированный файл был затем исправлен и примерно

известно, что именно было исправлено;

- **WRONG** – сгенерированный файл был затем исправлен, но не известно, что именно было исправлено;
- **VALID** – сгенерированный файл не исправлялся.

Глава 4

Распределенные АСУ

Конфигурирование межкомпонентного взаимодействия

Распределенные системы в ИС разрабатываются методом конфигурирования информационных потоков (см. **Технология разработки проекта в ИС**).

Основной принцип конфигурирования информационного потока в ИС заключается в независимом описании звеньев потока с дальнейшим заданием связей между звеньями.

До тех пор, пока речь идет о связи между компонентами одного узла, не возникает вопрос об аппаратно/программном интерфейсе, который должен быть задействован для обеспечения связи, – в этом случае достаточно выполнить конфигурирование свойств **связь/вызов** компонентов. Если взаимодействующие компоненты относятся к разным узлам, интерфейс связи, как правило, должен быть указан и сконфигурирован. Например, при задании связи двух каналов разных узлов по RS необходимо создать в узлах компоненты **СОМ-порт**, задать для них необходимые параметры и указать для канала-приемника используемый интерфейс связи.

Свойство **связь** может быть задано для компонента при копировании и вставке (см. **Копирование и вставка объекта структуры**), в окне свойств (см. **Вкладка 'Информация'**), а в случае канала – и в его редакторе (см. **Общие атрибуты каналов**)

Свойство **вызов** может быть задано для компонента при копировании и вставке (см. **Копирование и вставка объекта структуры**) и в окне свойств (см. **Вкладка 'Информация'**).

Связь канал-канал

При связи каналов в качестве **интерфейса взаимодействия** могут быть заданы (в скобках указан номер подтипа канала, для которого сконфигурировано свойство **связь**, – см. **Подтипы каналов и Атрибуты каналов, отображаемые профайлером**):

- **СНNET** (71) – по сети;
- **СНMLINK** (65) – по последовательному интерфейсу по протоколу M-LINK.

Информация о номерах подтипа и дополнения к подтипу канала содержится в атрибуте 126, **TsT**.

При взаимодействии с другим узлом во всех случаях (кроме приема сетевых автопосылок) канал с настроенным свойством **СВЯЗЬ** посылает запрос (в случае интерфейса CHNET – на IP-адрес удаленного сетевого узла), содержащий указание удаленному каналу передать свое значение или изменить его, и принимает ответ, содержащий информацию о выполнении запроса. При приеме сетевой автопосылки ответ не генерируется.

Автопосылка – это способность монитора передавать в сеть реальные значения каналов в виде ширококвещательных сообщений (в сети TCP/IP ширококвещательные сообщения отправляются по специальному IP-адресу, все биты которого равны единице). Монитор генерирует такие сообщения по каналам любого типа (как INPUT, так и OUTPUT) с установленным флагом **Автопосылка** (атрибутом **ToNet**, равным 1) при каждом изменении их реального значения. Прием автопосылки в канал **ch** (тип **ch** должен быть INPUT) от канала **rem_ch** с **ToNet=1** конфигурируется следующим образом: если канал **rem_ch** имеет тип INPUT, **ch** должен быть привязан к **rem_ch.R**; если OUTPUT – к **rem_ch.Q** (в обоих случаях **ch.In=rem_ch.R**).

В одном узле не может быть больше 32 каналов с автопосылкой.

Начиная с релиза 6.091, для запрета приема/отправки ширококвещательных сообщений используются следующие ключи в файле *.cnf (см. **Задание параметров работы мониторов**):

- **NOBROADCAST**

Вид взаимодействия каналов задает характеристика **ВИД СВЯЗИ** (в скобках указано дополнительное значение атрибута **Tst** канала, для которого сконфигурировано свойство **СВЯЗЬ**):

- **CHCOPY** (0) – копирование/установка значения удаленного канала (запрос/ответ);
- **CHASEND** (1) – прием сетевой автопосылки (подтверждение приема не генерируется);
- **CHGROUP** (9) – запрос/ответ по групповому номеру;
- **CHFAULT** (2) – запрос/ответ по резервам, кроме **All Redundant** (см. ниже);
- **CHFAULT** (10) – запрос/ответ по резервам **All Redundant** (см. ниже);
- **CHCS** (3) – специфический вид взаимодействия.

При интерфейсе **CHMLINK** виды связи **CHCOPY** и **CHASEND** идентичны.

При конфигурировании связи каналов разных узлов их взаимодействие задается на вкладке **Информация** окна свойств с помощью выбора из

списка одной из следующих опций (см. **Вкладка 'Информация'**):

- **Auto** – CHNET (UDP)/CHMLINK. Если доступно взаимодействие по сети, будет установлен интерфейс CHNET вне зависимости от возможности взаимодействия по RS. При установке интерфейса CHMLINK по умолчанию используется первый обнаруженный COM-порт, который сконфигурирован корректно. Опция **Auto** задается по умолчанию;
- **TCP** – CHNET (TCP), **подтип 71**, дополнение к подтипу 20;
- **RS 1** – CHMLINK, используется первый обнаруженный COM-порт, который сконфигурирован корректно (RS, SMS, модем; **подтип 65**, дополнение к подтипу – 17);
- **RS 2** – CHMLINK, используется второй обнаруженный COM-порт, который сконфигурирован корректно (RS, SMS, модем; **подтип 65**, дополнение к подтипу – 18);
- **GSM SMS 1** – GPRS;
- **GSM SMS 2** – GPRS;
- **Group** – CHNET (UDP), **подтип 71**, дополнение к подтипу 0. По групповым адресам не работает.

При интерфейсе CHMLINK: если вид связи – **Auto**, анализируются оба байта переменной **@Status.RS_number**. Если вид связи – **RS 1** или **GSM SMS 1**, анализируется байт 0, если **RS 2** или **GSM SMS 2** – байт 1 переменной. Если к портам подключены разные устройства, для каналов обмена с ними не следует задавать вид связи **Auto**.

При интерфейсе CHFAULT в этом списке доступна опция **All Redundant** (все резервы), а также опции вида **Net_ind<N>** (**N=0,1,2**), явно указывающие один из узлов (0 – основной, 1 – первый резерв, 2 – второй резерв).

При интерфейсе CHNET характеристики канала-приемника устанавливаются автоматически в зависимости от параметров запрашиваемого/передающего канала:

- если для запрашиваемого канала установлен флаг автопосылки, для запрашивающего канала устанавливается вид связи **CHASEND**.

Особенности связи канал-канал (см. также **Особенности пересчета и отработки каналов**, а также **Запись аргументов канала CALL в SIAD**):

- только для **CHNET\CHCOPY**: при привязке канала (**ch**) к строковому атрибуту 79, 80 или 127 (**attrS**) удаленного канала (**rem_ch**):

$$ch.attrS = rem_ch.attrS$$
- при привязке к числовому атрибуту (**attrN**) удаленного канала:

$$ch.2 = rem_ch.attrN$$

- для передачи времени используется один из следующих механизмов:
 - привязка канала TIME/HEX32 к временному атрибуту удаленного канала;
 - значение атрибута 45 может быть передано с помощью флага **Запрос времени значения** (см. Вкладка 'Флаги').

По **CHMLINK** время передается с точностью до 10 мс;

- если каналу с автопосылкой устанавливается признак аппаратной недостоверности, автопосылки не генерируются, и признак аппаратной недостоверности устанавливается всем каналам, принимающим данную автопосылку;
- если к числовому каналу **ch** (кроме канала TIME) привязан атрибут 35 произвольного канала **call** класса CALL (кроме CALL.TVC), в **ch** записывается значение последнего аргумента **call**;
- если к числовому каналу **ch** (кроме канала TIME) привязан атрибут 35 канала CALL.TVC, в **ch** записывается последнее значение первой кривой (см. Канал **CALL.TVC**).

У всех каналов, связанных с каналами удаленных узлов, в атрибут 90, **I0** записывается ID привязанного канала, а в атрибут 92, **I2** – порядковый номер узла (**ordinal**). Значения обоих атрибутов можно менять в реальном времени, что обеспечивает перепривязку каналов.

Связь через аргументы

К аргументу может быть привязан атрибут канала, аргумент или атрибут источника/приемника.

Методы и особенности привязки аргументов описаны в разделе **Табличный редактор аргументов**.

Связь 'аргумент – удаленный канал'

Такая связь разрешена только для аргумента канала вызова шаблона или канала класса CALL, который предназначен для обращения к удаленному узлу.

Если аргумент канала привязан к каналу не класса CALL другого узла, то на запрашивающем узле монитор создает один или несколько каналов, для которых интерфейс связи – CHNET или CHMLINK – определяется по результатам анализа файла **addr.ind** (см. **Файлы узла, создаваемые при экспорте**).

Имена новых каналов формируются следующим образом:

*_<номер запрашиваемого узла>_<номер запрашиваемого

канала><номер запрашиваемого атрибута><I/O>

Исключением из этого правила являются каналы CALL.8, **ChGroupReq** (см. Канал **CALL.ChGroupReq**).

В некоторых случаях имена создаваемых каналов соответствуют именам удаленных каналов, а имена, указанные выше, записываются в атрибут 80 создаваемых каналов.

Если монитор обнаруживает доступность обмена по сети, он создает один канал с интерфейсом связи CHNET, вид связи для которого устанавливается **CHCS**. Канал CHCS – того же класса, что и запрашиваемый канал, плюс 32 байта (255 разрядов). Номер бита, равного 1, соответствует номеру запрашиваемого атрибута. Каналы CHCS не пересчитываются, а только предоставляют данные для каналов класса CALL. В мониторе с отладкой (профайлере) такие каналы отображаются в стандартном объекте **18, N_Restored** (см. Вкладка 'Интегрированная среда разработки' / Раздел 'Сохранить для MPB'). Тип нового канала соответствует типу аргумента, к которому привязан запрашиваемый канал. На один компонент создается один компонент, а при записи/чтении – два (атрибуты, которые требуется читать/устанавливать, задаются в компоненте).

Если выбирается обмен по M-LINK/RS, то на все найденные сочетания

<id канала><attr><I/O>

создаются компоненты с интерфейсом связи CHMLINK и видом связи CHCOPY, класс которых соответствует запрашиваемым данным (например, если запрашиваемое значение REAL, создается FLOAT).

Значение аргумента устанавливается для соответствующего атрибута локального канала, созданного монитором, – это следует иметь в виду при использовании связи аргумента с такими атрибутами удаленного канала, как, например, (4, I) **Достоверность** и (8, W) **Подключение**.

При связи по сети, если аргумент (**arg**) типа OUTPUT канала (**ch**) привязан к атрибуту **IN** удаленного канала (**rem_ch**), то с периодом, равным нескольким (2-3) периодам пересчета канала **ch**, тип **arg** изменяется на INPUT, у удаленного узла запрашивается значение **rem_ch.IN** и записывается в **arg**, после чего тип **arg** изменяется на OUTPUT. Таким образом, в случае **CHNET\CHCS** не только **rem_ch.IN** изменяется в соответствии с изменением **arg**, но и **arg** изменяется в соответствии с изменением **rem_ch.IN** в удаленном узле.

Передача строк и времени при связи аргумент-канал:

- передача строки (только для **CHNET\CHCS**): у канала CALL должны быть созданы два аргумента (REAL и STRING), и оба этих аргумента должны быть привязаны к строковому атрибуту удален-

ного канала (аргумент REAL – вспомогательный, аргумент STRING содержит полученное строковое значение);

- для передачи времени аргумент привязывается к временному атрибуту удаленного канала.

Связь ‘аргумент – аргумент’

В одном узле разрешена привязка аргументов канала CALL к аргументам другого канала CALL, при этом привязываемый аргумент должен иметь ID в диапазоне 0-511.

При связи аргументов STRING каналов CALL можно передать не более четырех символов, если ни один из этих аргументов не привязан к строковому атрибуту произвольного канала.

При взаимодействии через аргументы каналов CALL, вызывающих шаблоны, привязка может быть задана в любом из каналов, однако проект экспортируется быстрее, если:

- при взаимодействии CALL.Screen и CALL.Program привязки аргументов заданы в CALL.Screen.

В канале CALL, не вызывающем шаблон, не допускается привязка аргументов к аргументам (такие аргументы могут быть привязаны к атрибуту канала либо не иметь привязки). При взаимодействии через аргументы канала CALL, вызывающего шаблон, с каналом CALL, не вызывающим шаблон, привязка аргументов должна быть задана в первом.

Особенности взаимодействия

К особенностям взаимодействия относятся следующие:

- CHGROUP. CHFAULT:
 - посылка N узлам;
 - если запрос, то во вход попадают данные от узла со статусом WORK, а в Q – данные остальных узлов (для CHFAULT – если вид взаимодействия – **All Redundant**);
- в NET отвечают все узлы, в M-Link – только узел со статусом WORK, однако по OUTPUT изменяют значения все (для CHFAULT – если вид взаимодействия – **All Redundant**);
- CHCS может работать на посылку в двух режимах – CHCOPY (запрос/ответ) и CHASEND (broadcast);
- CHASEND не поддерживает работу с текстовыми атрибутами;
- CHMLINK не поддерживает работу с текстовыми атрибутами;
- в узле LOGGER должны быть созданы связи со всеми каналами,

которые имеют флаг **LOGGER**, – для этих узлов нет автопостроения.

При выполнении любой функции TRACE MODE, запрашивающей значения удаленных аргументов (по сети или RS), максимальное число передаваемых данных – 16383 (0x3FFF) (**CALL.RemArc**, **Call.ChGroupReq** и т.п.).

Допустимые цепочки связей

При редактировании проекта в ИС допускается создание цепочек связей – например, канал–канал–канал; источник/приемник-источник/приемник–канал–канал–аргумент и т.п.

В цепочках вида **канал–источник/приемник1–источник/приемник2** можно задать порядковый номер источника/приемника, который будет использован при экспорте узла (в разделе **Сохранить для MPB** диалога настройки параметров ИС – см. **Задание общих настроек ИС**). Это позволяет «обойти» отключенное или удаленное устройство (например, контроллер) без модификации проекта. Кроме того, при использовании в такой цепочке встроенного генератора TRACE MODE можно отлаживать проект без реальной аппаратуры.

В подобной цепочке связей нельзя использовать аргумент (вместо канала).

Создание контура управления

Для создания контура управления нужно передать, в том числе с обработкой, данные из измерительного информационного потока (датчик => УСО => контроллер => операторская станция или, с точки зрения ИС, источник => канал INPUT => канал INPUT => ...) в управляющий (операторская станция => контроллер => исполнительное устройство или ... => канал OUTPUT => приемник). Для этой цели можно использовать только программу или канал CALL с типом вызова MOVE.

Связь [канал INPUT или источник] => [канал OUTPUT] запрещена.

Специальные виды связи

В TRACE MODE 6 предусмотрены специальные виды связи компонентов, описанные в следующих разделах:

- **Атрибуты канала класса CALL** (типы вызова (49) **RemArgument** и (54) **MLink**);

- Канал CALL.ChGroupReq;
- Канал CALL.TVC;
- Канал CALL.DI_DO;
- Канал CALL.Vector;
- Регистратор аварийных событий;
- Универсальный механизм обмена с электросчетчиками (см. подраздел **Конфигурирование обмена/Сеть**).

Обмен по TCP/IP

Полный набор параметров драйвера TCP/IP (Tcpip.sys) задается в реестре Windows (см. **TCP/IP and NBT Configuration Parameters for Windows XP** в MSDN). Оптимальные значения параметров зависят от архитектуры сети и могут быть получены из специальной литературы.

В TRACE MODE начальные параметры обмена по TCP/IP конфигурируются с помощью редактора узла (см. **Задание параметров узла**), ключей команды запуска и ключей в файле *.cnf (см. **Задание параметров работы мониторов**).

Для диагностики и управления обменом в реальном времени используются следующие средства TRACE MODE:

- системные переменные (см. **Группа СИСТЕМНЫЕ** и **Группа ДИАГНОСТИКА**);
- окна **Панель MPV** и **Статусы узлов** (см. **Профайлер с поддержкой графических экранов** и **Профайлер без поддержки графических экранов**);
- некоторые конфигурации канала CALL.ChGroupReq (см. **Канал CALL.ChGroupReq**).

Сеть конфигурируется с помощью ключей в файле *.cnf (см. **Сеть** в разделе **Задание параметров работы мониторов**).

Механизм предотвращения потерь при обмене по сети

HOST по COPY_FROM запрашивает данные по сети у SLAVE (только для числовых каналов и исключая каналы СОБЫТИЕ).

При потере связи и последующем ее восстановлении SLAVE передаст значения запрашиваемых каналов, если они менялись в период отсутствия связи, и, в соответствии с флагами, данные соответствующих COPY_FROM-каналов запишутся в SIAD и/или в ОТ.

Для работы этого механизма необходимо:

- SLAVE:
 - узел – **RTM** или **EmbeddedRTM**;
 - у каналов установлен флаг **Регистратор**;
 - для узла задано **При старте=Выключен** в разделе **Регистратор** вкладки **Основные** редактора узла (см. **Задание параметров узла**). (Если **При старте=Выключен**, обмен с узлом типа **Logger** не производится);
 - отсутствуют ключи **SLVPUIP** (см. **MPB как сервер протоколов поверх TCP/IP**).

Во время отсутствия связи SLAVE хранит данные каналов в буфере, объем которого и глубину задают соответственно следующие ключи:

LOG_SND_QSIZE=<объем, kB (DEC)> – по умолчанию 1024

LOG_STORY_TIME=<число секунд, DEC> – по умолчанию 3600

- HOST:
 - узел – **RTM** или **EmbeddedRTM**;
 - в узле есть каналы, запрашивающие данные у узла SLAVE по COPY_FROM. У каналов установлены флаги записи в архив, ОТ или регистратор.

REQ11=UNSUPPORT:

- если для себя – запрет данного режима ко всем другим узлам;
- если для другого узла – запрет данного режима к этому узлу.

Если со SLAVE в данном режиме работает более одного клиента (HOST), то допускается единовременное пропадание 1 клиента, при этом поддерживается целостность данных.

Если после старта SLAVE не находит ни одного клиента с обращением на данные из буфера, то он чистит буфер каждые 60 секунд.

Для отладки режима может использоваться ключ **DBG_INFO_NODE=ON** (см. соответствующий подраздел раздела **Сообщения при DBG_INFO_NODE=ON**).

Обмен по телефонным линиям

Для узлов TRACE MODE параметры COM-портов и подключенных к ним модемов задаются в ИС (см. **Редактор параметров COM-порта и Задание параметров узла**).

Для обмена по телефонным линиям с устройствами, в которых нет TRACE MODE, в узле TRACE MODE используются каналы DCS, MODBUS и t11 (подтипы 3, 9 и 11). MPB идентифицирует такие устройства по

двум параметрам, которые задаются в редакторе источника/приемника, – номеру устройства и его номеру телефона, что позволяет конфигурировать обмен с очень большим числом устройств. Номер телефона устройства указывается в поле **Комментарий** в редакторе источника/приемника в следующем формате:

tel: <номер в международном формате>

Информация об устройствах, сгруппированных по RS, сохраняется в текстовых файлах **rs<номер RS>** в папке узла (здесь номер RS равен 0 для COM1, 1 – для COM2 и т.д.). Формат строк таких файлов:

<номер устройства> <номер телефона>

Номер устройства индицируется атрибутом (94, **C1**), номер RS – атрибутом (93, **C0**) канала обмена.

Поиск каналов обмена в базе производится по устройствам – т.е., вначале MPB ищет и обрабатывает каналы обмена с устройством 1, затем – с устройством 2 и т.д. Данные функции выполняются в потоке 15, **Модем** (см. **Потоки монитора**).

Обмен конфигурируется в файле *.cnf (см. **Задание параметров работы мониторов**).

Для диагностики обмена используется переменная **@Modem_Step**, для управления – **@Input_Output** (см. **Группа СИСТЕМНЫЕ**).

Если в узле существует хотя бы один канал (**ch**) обмена с устройством, в котором нет TRACE MODE, для узла запрещается режим M-LINK SLAVE по тому RS, который указан в **ch** (в том числе для обмена с узлом TRACE MODE). Если в узле нет каналов обмена по RS, а RS с назначением **Modem** есть, для узла запрещается режим M-LINK MASTER.

Если в узлах TRACE MODE заданы RS с назначением **Modem**, но модемов на самом деле нет, и номер телефона для RS не задан, то между такими узлами будет идти обычный обмен по RS (не будет инициализации модема, набора номера и т.п.).

Если узлам TRACE MODE явно задан статус (одному – MASTER, другому – SLAVE), то поиск каналов обмена в базе меняется на поиск по узлам. Если статусы узлов не заданы явно, узлы будут запрашивать друг друга, и с помощью таймаутов операции записи и чтения можно корректно распределить по времени.

Резервирование в TRACE MODE 6

TRACE MODE 6 поддерживает **структурное резервирование**:

- узлов (с кратностью 1:1 (дублирование) и 2:1);
- УСО;
- распределенного УСО.

Резервы узла создаются при разработке проекта в ИС (см. **Меню и главная панель инструментов навигатора проекта, а также Модификации ИС TRACE MODE 6**). В документации TRACE MODE 6 узел, для которого в ИС создаются резервы, иногда называется основным.

При разработке проекта автоматически обеспечивается идентичность баз каналов основного узла и его резервов, индивидуально могут быть заданы только параметры узлов.

В реальном времени обеспечивается выполнение следующих функций:

- отслеживание состояния и ряда других параметров узлов (см. **@Status** в разделе **Группа СИСТЕМНЫЕ**, а также описание команды **Статусы узлов** профайлеров);
- при выходе из строя узла в состоянии WORK – автоматический переход из TRACE в WORK резерва с наименьшим таймаутом **Waite_Chg_Sts * Waite_Start** (см. **Вкладка 'Дополнительно' редактора узла** в разделе **Задание параметров узла** и **@RTM_Parameter** в разделе **Группа СИСТЕМНЫЕ**). Это свойство может быть использовано для локального и/или удаленного управления статусами резервов;
- автоматический переход на обмен через резервный сетевой адаптер в случае отсутствия обмена в течение времени, заданного ключом **NET_SWTNTIME** в файле ***.cnf** (в WINDOWS XP). Для контроля работоспособности линий связи используется канал **CALL.ChGroupReq**, к которому привязана системная переменная **@Net_Code** (см. **Канал CALL.ChGroupReq**);
- механизмы синхронизации (см. **Синхронизация резервов**). Атрибуты каналов узла TRACE могут синхронизироваться как однократно (при старте узла), так и непрерывно;
- механизмы подключения/отключения каналов при переключении статуса узла (WORK/TRACE) (см. **Синхронизация резервов**);
- автоматический перевод резерва с младшим ID в состояние WORK, а остальных резервов – в TRACE, если в состоянии WORK оказались одновременно несколько резервов;
- резервирование COM-портов (см. **Редактор параметров COM-порта**).

При проектировании резервированных систем следует также учитывать особенности, изложенные в следующих разделах:

- Особенности пересчета и отработки каналов
- Связь канал-канал
- Особенности взаимодействия
- Редактор параметров СОМ-порта;

При наличии между резервами прямой связи может использоваться канал CALL с типом вызова (53) DI_DO – см. Атрибуты канала класса CALL.

Синхронизация резервов

Синхронизация при старте описана в разделе **Задание параметров узла / Вкладка 'Отчет тревог/Дамп/Параметры'** редактора узла.

Для управления синхронизацией резервов используются следующие ключи в файле *.cnf: (ключи, отмеченные звездочкой, работают в реальном времени):

- **SYNC_MASK**=<маска синхронизации> – @RTM_Parameter с Параметр=12;
- **SYNC_BY_IP** или **SYNC_BY_TCP** – протокол синхронизации (соответственно UDP или TCP), по умолчанию используется TCP (по IP синхронизация истории отсутствует);
- **SYNC_ADJUST**=OFF/ON – отсылка данных сразу или после цикла пересчета. По умолчанию данные отсылаются поле цикла пересчета;
- (*) **SYNC_DEBUG**=OFF/ON – режим отладки выключен или включен, по умолчанию выключен;
- (*) **SYNC_EMUL_ERR**=<число DEC> – эмуляция ошибки:
 - 1 – не отсылать завершающий блок;
 - 2 – не отсылать некоторые блоки данных (TCP);
 - 4 – игнорировать запрос соединения.
 - 8, 16 – замедление работы WORK на отсылку данных не в начальный момент (10, 100, 500);
- (*) **SYNC_ARCH**=ON/OFF – включение/отключение синхронизации архивов;
- **SYNC_ARCH_LAST_DEPTH**=<число секунд> – временной интервал синхронизации архива от текущего времени, когда у SLAVE архив есть (по умолчанию 3600с);
- **SYNC_ARCH_START_DEPTH**=<число секунд> – временной интервал синхронизации архива от текущего времени, когда у SLAVE

- архива нет (по умолчанию 3600с);
- **SYNC_ARCH_DISABLE_A1=ON** – запрет синхронизации 1-го архива;
 - **SYNC_ARCH_DISABLE_A2=ON** – запрет синхронизации 2-го архива;
 - **SYNC_ARCH_DISABLE_A3=ON** – запрет синхронизации 3-го архива;
 - **SYNC_CPY_ARCH_BYFLAG=ON/OFF** – при синхронизации архива передаются/не передаются значения каналов, не отмеченных для синхронизации (не установлен бит 0 атрибута 58, **DumpSync**);
-
- (*) **SYNC_HISTORY=ON/OFF** – включение/отключение синхронизации отчетов тревог;
 - **SYNC_HISTORY_TIME_LIVE=<число секунд>** – время жизни сообщений в отчет тревог для синхронизации (по умолчанию 3600с);
 - **SYNC_CPY_ALA_BYFLAG=ON/OFF** – при синхронизации отчета тревог передаются/не передаются значения каналов, не отмеченных для синхронизации (не установлен бит 0 атрибута 58, **DumpSync**);
-
- (*) **SYNC_SLOW_WORK=ON/OFF**(по умолчанию) – включение/отключение замедленной синхронизации;
 - **SYNC_FAST_START=ON**(по умолчанию)/**OFF** – включение/отключение быстрого старта (по пингу);
 - **SYNC_DISABLE_SYNC=<число HEX>** – отключение синхронизации каналов:
 - 0 – CALL
 - 1 – CALL
 - 2 – HEX16
 - 3 – HEX32
 - 4 – FLOAT без обработки
 - 5 – FLOAT с обработкой
 - 6 – DOUBLE FLOAT
 - 12 – СОБЫТИЕ
 - 13 – TIME
 - ...
 - 16 – CALL вызова программ
 - 17 – CALL.ChGroupReq, CALL.Vector, CALL.TVC, CALL.AS_DATA
 - 18 – других CALL

- **SYNC_DISABLE_OTHER_THEN** и **SYNC_ENABLE_OTHER_THEN** – отключение/включение синхронизации каналов CALL, для которых отдельные ключи синхронизации не предусмотрены;
- **SYNC_UNIT_TCP_CONNECT=DISABLE** (по умолчанию) / **ENABLE** – отключение/включение обмена и соединения по сети в статусе TRACE; бит задает протокол;
- **SYNC_UNIT_RS_COM=DISABLE** – **DISABLE** (по умолчанию) / **ENABLE** – отключение/включение обмена и соединения по RS в статусе TRACE; установленный бит задает порт (бит 1 – COM1);
- **SYNC_CONTROLLER_OEM_RS_COM=DISABLE/ENABLE** – запрет/разрешение обращений к порту в узле TRACE;
- **SYNC_EXTRA_DEBUG=OFF/ON** – запрет/разрешение вывода в протокол дополнительной информации о синхронизации;
- **SYNC_FIND_ERROR=ON** – вывод в протокол дополнительной информации уровня 2 о синхронизации;
- **SYNC_SWITCH_MODE=ON/OFF** – зарезервировано;
- (*) **SYNC_ARCH_START_DEPTH=ALL/<число секунд>/OFF** – на основании 3 параметров (время последней записи в собственном архиве, время последней записи в архиве узла WORK и значение данного ключа) узел TRACE запрашивает последние архивные данные у узла WORK (ALL соответствует 1 году);
- **TM_FT_IP_ADDR_<номер узла>=<IP-адрес>** – IP-адрес узла, который синхронизируется с данным узлом;
- **SYNC_NO_NODE=<число секунд>** – таймаут синхронизации, для резерва 10с по умолчанию (таймаут проверки наличия резерва).

Если в канале CALL **Зарезервировано=yes**, то начальные значения атрибутов (8, **W**) и (3, **C**) задаются в ИС для режима TRACE (в режиме WORK (8, **W**)=(3, **C**)=0).

Информация о синхронизации доступна на вкладке **Диагностика** панели MPB (см. **Панель MPB**).

Для диагностики синхронизации используется системная переменная 15.16 **@Redundant** (см. **Группа ДИАГНОСТИКА**).

В синхронизации принимают участие только каналы с установленным битом 0 атрибута (58, **DumpSync**). Узел TRACE на каждом цикле (в том числе при старте до пересчета каналов) посылает запрос узлу WORK и получает данные. Если узел WORK не успевает ответить в течение цикла (т.е. если данные от узла WORK не получены), узел TRACE повторяет запрос спустя таймаут. Подключение и переключение каналов отсутствует.

Структурное резервирование и надежность

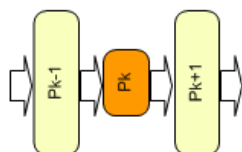
Структурное резервирование не всегда приводит к повышению надежно-

сти системы, поэтому до принятия решения о таком резервировании необходимо выполнить расчет надежности АСУ.

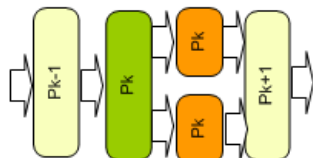
Терминология и методы расчета, использованные в данном разделе, соответствуют ГОСТ 27.002-89.

Рассмотрим в качестве примера некорректного структурного резервирования нагруженное дублирование некоторого передающего устройства в последовательном тракте.

Начальная схема тракта:



Пусть дублируется устройство p_k (таким же устройством), при этом для обеспечения переключения на резерв в структурную схему добавляется переключатель, который имеет то же значение p_k вероятности безотказной работы, что и само устройство:



При таком резервировании коэффициент выигрыша надежности системы по вероятности безотказной работы составит:

$$G_p = p_k(2 - p_k)$$

Поскольку неравенство $p_k(2 - p_k) > 1$ не имеет решений, рассмотренное резервирование приводит к снижению надежности системы.

Резервирование УСО

Для переключения на резервную плату ввода/вывода, установленную в системную шину, нужно в реальном времени присвоить ее адрес соответствующему атрибуту канала обмена (см. **Атрибуты каналов, отображаемые профайлером**). При таком переключении монитор, в случае необходимости, реинициализирует обмен с УСО.

Обмен с базами данных

Основными функциями, выполняемыми TRACE MODE 6 при работе с базами данных (далее – БД), являются следующие:

- подключение к БД по протоколу ODBC3;
- создание новых источников данных ODBC (при редактировании проекта);
- отображение информации о структуре БД (при редактировании проекта);
- создание и редактирование запросов к БД, в том числе при помощи мастера (при редактировании проекта);
- выполнение запросов к БД (при редактировании проекта и в реальном времени).

Для программирования связей с БД в интегрированную среду разработки проекта встроен **редактор связей с базами данных**. Этот редактор снабжен **мастером**, позволяющим конфигурировать основные операции по взаимодействию с базами данных без глубокого изучения языка SQL.

Синтаксис SQL

TRACE MODE 6 не накладывает никаких ограничений на запросы к базам данных; эти ограничения определяются установленной системой управления базами данных (СУБД).

Описание диалектов SQL разных СУБД (в том числе типов данных и форматов констант) выходит за рамки данного руководства. Эту информацию можно получить из документации СУБД и драйверов ODBC. В данном руководстве рассматривается построение простейших запросов к базе данных Microsoft Access 2000.

Настройка параметров редактора связей с БД

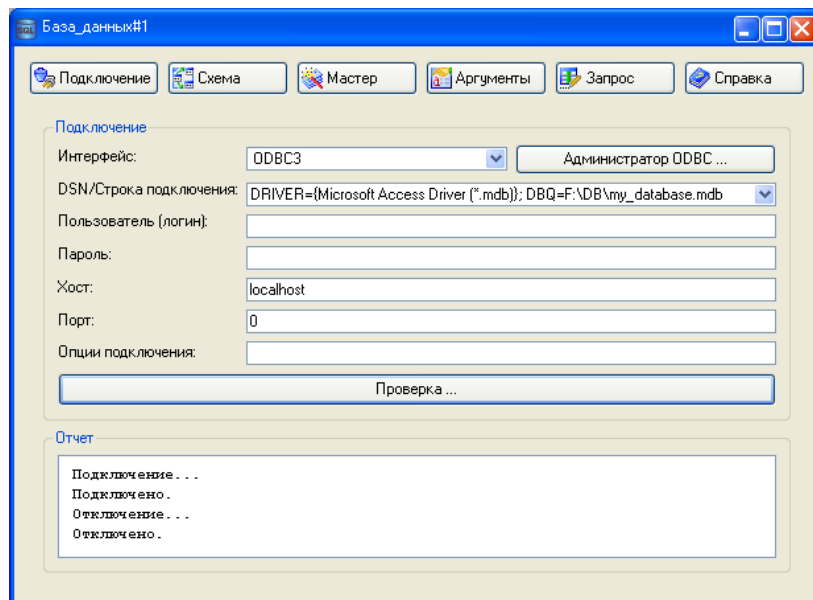
Параметры редактора связей с БД настраиваются на вкладке **Редактор SQL** (см. **Задание общих настроек ИС**):

Эта вкладка содержит типовые инструменты для задания таких стандартных характеристик шрифта, как семейство, размер, вид начертания (жирный, курсив, подчеркнутый) и цвет. Данные параметры могут быть заданы отдельно для обычного текста запросов SQL, ключевых слов и подстановок. Для каждого из этих объектов может быть также задан цвет фона.

Вкладка содержит кнопку восстановления параметров, заданных по умолчанию.

Подключение к базе данных

При открытии редактора связей с БД по умолчанию открывается окно **Подключение**:



Для подключения к БД можно использовать два способа:

- зарегистрировать БД в ОС как источник данных ODBC и подключиться к этому источнику;
- подключиться без регистрации БД как источника ODBC.

Чтобы открыть диалог ОС регистрации БД как источника данных ODBC, надо нажать кнопку **Администратор ODBC**.

Для подключения к источнику данных ODBC в поле **Интерфейс** нужно выбрать протокол (ODBC3), в поле **DSN/Строка подключения** – имя зарегистрированного источника данных. В поле **Пароль** нужно ввести пароль, заданный для БД. Поля **Пользователь**, **Хост** и **Порт** надо оставить пустыми.

Вместо DSN можно ввести **строку подключения** (в этом случае регистрация БД как источника данных ODBC не требуется). Синтаксис строки подключения указан в документации драйверов ODBC. На рисунке выше показана строка подключения к базе данных Microsoft Access (в простейшем случае).

В поле **Опции подключения** могут быть заданы дополнительные опции подключения к БД, если эти опции поддерживаются СУБД (описание опций выходит за рамки данного руководства).

Для проверки подключения к БД нужно нажать кнопку **Проверка** – в нижней части окна **Подключение** появится соответствующее сообщение.

Окно ‘Схема’

Если подключение к БД выполнено корректно, в окне **Схема** редактора связей с БД отображается структура базы данных.

В столбце **Тип** этого окна содержится информация о типе данных полей таблиц БД. В столбце **Обяз.поле** указывается, являются ли поля столбцов БД обязательными для заполнения (**нет** – могут принимать значение **NULL**, т.е. быть пустыми, **да** – поля обязательны для заполнения). В столбце **Сортировка** указывается порядок сортировки данных (**asc** – по возрастанию, **desc** – по убыванию).

Окно ‘Аргументы’

В окне **Аргументы**, представляющем собой редактор аргументов (см. **Табличный редактор аргументов**), задаются аргументы редактируемого шаблона связи с БД.


При задании типа аргумента следует учитывать направление передачи данных (**IN** – в БД, **OUT** или **IN/OUT** – из БД).

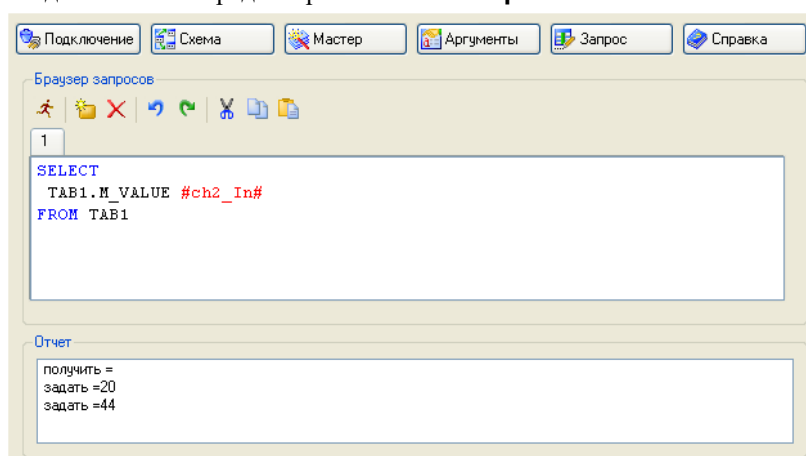
Имена аргументов в виде подстановок (см. **Подстановки в SQL-запросе**) могут быть использованы в SQL-запросе. Механизм подстановок обеспечивает запись данных БД в атрибуты каналов, а также запись значений атрибутов каналов в БД.

Создание и выполнение SQL-запросов

Создание SQL-запросов

В шаблоне связи с БД может быть создан единственный SQL-запрос. Запрос может быть создан:

- вручную в окне **Запросы** – для этого нужно щелкнуть кнопку  (по этой команде создается вкладка **1**) и ввести на вкладке запрос с помощью клавиатуры;
- с помощью мастера (для его запуска нужно нажать кнопку **Мастер**). Запрос, построенный с помощью мастера, может быть в дальнейшем отредактирован в окне **Запрос**:



Данное окно снабжено типовыми инструментами создания/удаления запроса и работы с буфером обмена (см. **Типовые инструменты редактирования**).

Подстановки в SQL-запросе

В зависимости от направления передачи данных, подстановки в SQL-запросе имеют различный синтаксис.

Вставка значения аргумента в базу данных

При вставке значения аргумента в базу данных в подстановке используются одинарные кавычки:

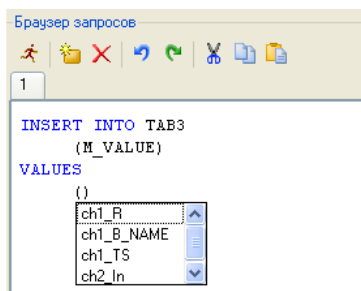
```
'#<имя аргумента>#'
```

Например:

```
INSERT INTO TAB1
      (M_VALUE)
VALUES
      ('#ch1_R#')
```

При построении запроса с помощью мастера подстановки создаются в правильном формате автоматически.

Если запрос создается вручную, при вводе знака # на экране появляется диалог выбора аргумента:



После выбора аргумента создается подстановка без одинарных кавычек, и их необходимо добавить вручную.

Запись в аргумент значения из базы данных

При записи в аргумент значения из базы данных одинарные кавычки в подстановке не используются:

#<имя аргумента>#

Например:

```
SELECT
      TAB1.M_VALUE #ch2_In#
FROM TAB1
```

Особенности использования аргумента в условии

Если значение аргумента – это значение даты и времени, то при использовании такого аргумента в предложении WHERE следует учитывать синтаксис Microsoft Access для временных меток.

Мастер запросов строит предложение WHERE в следующем формате:

```
WHERE mTime > #ARG_000#
```

Если сравнение должно быть произведено как по времени, так и по дате, предложение следует отредактировать следующим образом:

```
WHERE mTime > {ts '#ARG_000#'}
```

Модификация запросов в реальном времени

Подстановки могут использоваться вместо имен таблиц и полей БД, что позволяет модифицировать запросы в реальном времени. В таких подстановках одинарные кавычки не используются.

Пример с оператором INSERT:


```
INSERT INTO #ARG_000#  
  ( #ARG_002# )  
VALUES  
  ( '#ARG_001#' )
```

Пример с оператором SELECT:

```
SELECT m_val #ARG_003#  
FROM #ARG_000#
```

Две подстановки подряд использовать нельзя, поэтому, например, запрос **SELECT #ARG_002# #ARG_003# FROM #ARG_000#** не будет выполнен.

Выполнение SQL-запросов из ИС

Окно **Запросы** снабжено кнопкой  запуска текущего SQL-запроса. При выполнении запроса из ИС сообщение о результатах выводится в нижнюю часть окна.

Если в запросе используется аргумент, для которого задано значение по умолчанию (см. **Поля редактора аргументов**), это значение будет использовано при выполнении запроса из ИС.

Выполнение SQL-запросов в реальном времени

Для выполнения SQL-запросов к подключенной БД в реальном времени в узле должен быть создан канал класса CALL с типом вызова **SQLQuery**, настроенный на вызов шаблона связи с БД. При данном типе вызова тип канала CALL не имеет значения (см. **Атрибуты канала класса CALL**).

С помощью атрибута **Параметр** канала CALL можно задать номер возвращаемой строки из выборки, полученной из таблицы БД в результате запроса. Если **Параметр=0** (значение по умолчанию), возвращается последняя строка.

Для выполнения запроса в канал CALL нужно записать 1. После отработки значение канала автоматически сбрасывается в 0.

Если в запросе используется аргумент, для которого задано значение по умолчанию (см. **Поля редактора аргументов**), то при выполнении за-

проса монитор использует привязку аргумента, а если она не задана – значение по умолчанию.

Если SQL-запрос по каким-либо причинам выполнить невозможно, то по истечении 600 секунд с момента инициализации запроса каналу CALL устанавливается признак аппаратной недостоверности.

В атрибут 87, **СС** канала CALL с данным типом вызова записывается время (в миллисекундах) между началом передачи SQL-запроса драйверу БД и готовностью драйвера БД к приему следующего SQL-запроса (см. **Атрибуты каналов, отображаемые профайлером**). В графической оболочке МРВ значение атрибута 87, **СС** отображается в виде sss.mmm. Пока драйвер не подтвердил готовность, в атрибуте 87 хранится время начала передачи.

МРВ не управляет транзакциями и не контролирует их; т.е., за целостность БД отвечает СУБД.

В атрибут 91, **I1** записывается число строк в выборке, полученной из таблицы БД в результате запроса.

В атрибут 92, **I2** записывается код ошибки (см. **Коды диагностируемых ошибок**).

Для записи в протокол профайлера информации о взаимодействии с БД используется системная переменная **@Debug** (см. **Группа СИСТЕМНЫЕ**).

По умолчанию, МРВ выполняет одновременно только один SQL запрос. Этот режим можно отменить с помощью ключа **SQLMANY** в файле **TMcom_<ordinal>.cnf** (см. **Задание параметров работы мониторов**).

Для принудительного прерывания сессии в канале CALL.SQLQuery нужно установить флаг **Запрос времени значения** (атрибут 50).

Извлечение столбцов

Допустима следующая конфигурация:

- канал **CALL.SQLQuery (INPUT, Параметр=0)** содержит запрос с конструкцией **SELECT**;
- к аргументу канала **CALL.SQLQuery**, который используется в качестве подстановки в запросе, привязан атрибут 0, **R** канала **CALL.ChGroupReq**.

При **CALL.SQLQuery.In=0xFFFF** все извлеченные значения столбца последовательно записываются в аргументы канала **CALL.ChGroupReq**.

При групповом извлечении строк и столбцов к аргументам

канала **CALL.SQLQuery** можно привязывать каналы **CALL.STRING**.

Если к аргументам канала **CALL.SQLQuery** привязаны атрибуты (124, **ArgSize**) каналов **CALL.ChGroupReq**, при **CALL.SQLQuery.In=0xFFFF** извлеченные значения столбца записываются в канал **CALL.ChGroupReq**, привязанный к аргументу с порядковым номером **CALL.SQLQuery.R**.

Извлечение строк

Допустима следующая конфигурация:

- канал **CALL.SQLQuery (INPUT, Параметр<>0)** содержит запрос с конструкцией **SELECT**.

В этом случае по команде **CALL.SQLQuery.In=0xFFFF** выполняется запрос, после чего в **CALL.SQLQuery.In** может быть многократно записано число, в котором должен быть установлен бит 15 (0x8000), а биты 0-12 задают номер строки, извлекаемой из полученной выборки. Значения извлеченной строки записываются в аргументы канала **CALL.SQLQuery**, используемые в виде подстановок в запросе.

При групповом извлечении строк и столбцов к аргументам канала **CALL.SQLQuery** можно привязывать каналы **CALL.STRING**.

Групповая запись

Допустима следующая конфигурация:

- канал **CALL.SQLQuery (OUTPUT)** содержит запрос с конструкцией **INSERT**;
- к аргументу канала **CALL.SQLQuery**, который используется в качестве подстановки в запросе, привязан атрибут 0, **R** канала **CALL.ChGroupReq** или **CALL.TVC**.

В этом случае в столбец записывается **CALL.SQLQuery.R** аргументов канала **CALL.ChGroupReq/CALL.TVC**, начиная с нулевого (к таким аргументам могут быть привязаны только атрибуты (0, **R**) других каналов).

Формирование кривых в CALL.TVC

Если таблица БД содержит столбец временных меток и столбцы значений, допустима следующая конфигурация:

- канал **CALL.SQLQuery (INPUT, Параметр=0)** содержит запрос с конструкцией **SELECT**;
- к аргументу (**DATE_AND_TIME**) канала **CALL.SQLQuery**, кото-

рый используется в качестве подстановки в запросе (в операторе извлечения столбца временных меток), привязан атрибут 45, **T** канала **CALL.TVC**;

- к аргументу канала **CALL.SQLQuery**, который используется в качестве подстановки в запросе (в операторе извлечения столбца значений), привязан атрибут (142, **ARG02**), (143, **ARG03**)... канала **CALL.TVC**.

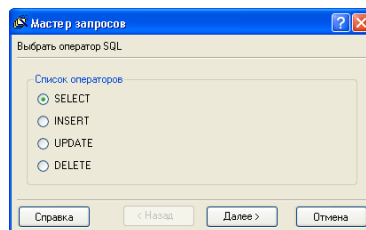
При отработке запроса в **CALL.TVC** формируется соответствующая кривая (см. **Канал CALL.TVC**).

Создание SQL-запросов с помощью мастера

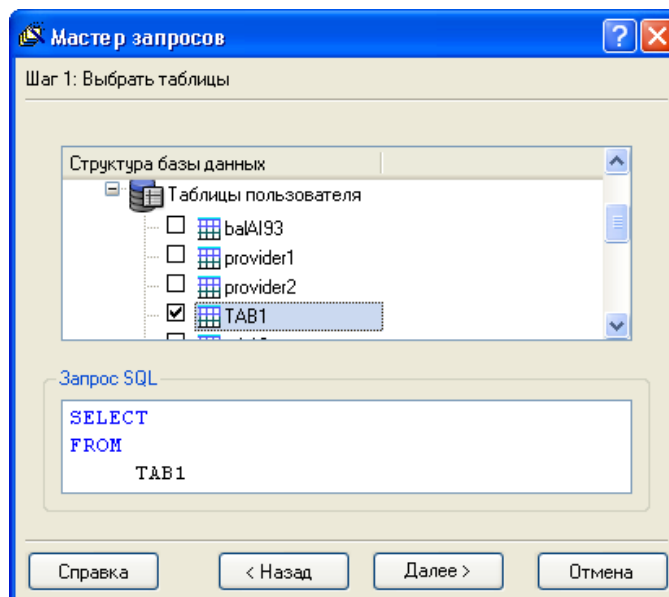
Для построения простейших SQL-запросов редактор связей с БД снабжен мастером, для запуска которого нужно нажать кнопку **Мастер** редактора (см. **Создание SQL-запросов**).

Построение запроса SELECT с помощью мастера

Для построения запроса с оператором SELECT нужно выбрать этот оператор в окне мастера:



После нажатия кнопки **Далее** на экране появится диалог, в котором нужно выбрать таблицы БД:

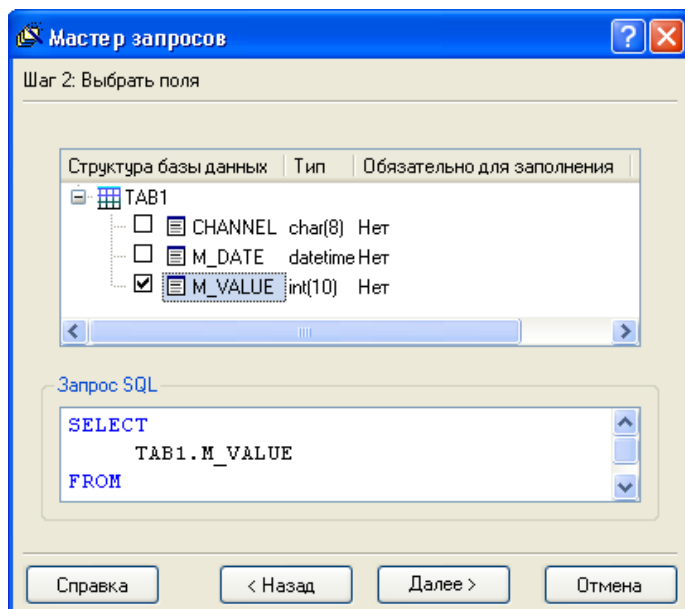


При построении запроса с помощью мастера текущий листинг запроса отображается в нижнем окне диалогов.

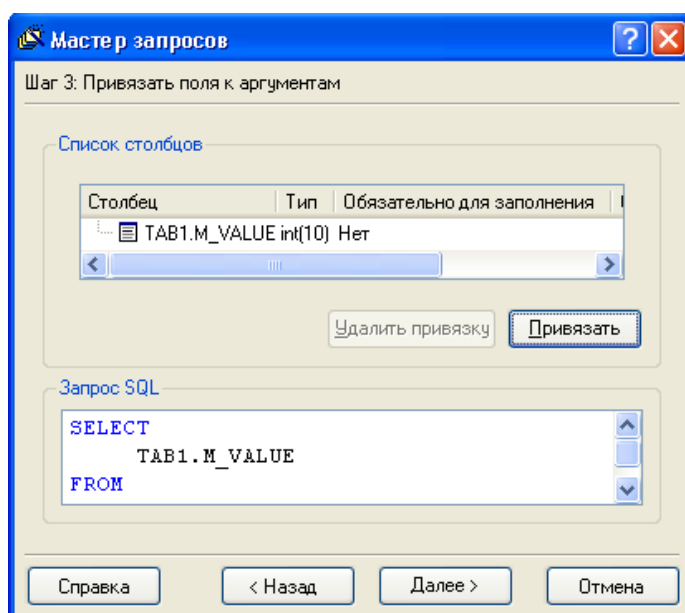
Все диалоги мастера снабжены кнопкой **Назад** для возврата

к предыдущему шагу.

После нажатия кнопки **Далее** на экране появится диалог, в котором нужно выбрать поля таблиц БД:

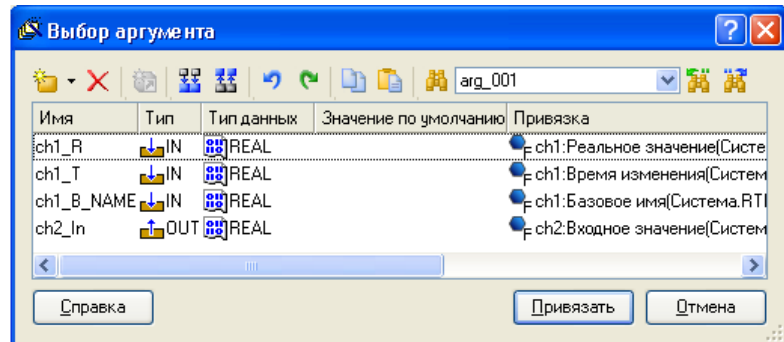


После нажатия кнопки **Далее** на экране появится следующий диалог:



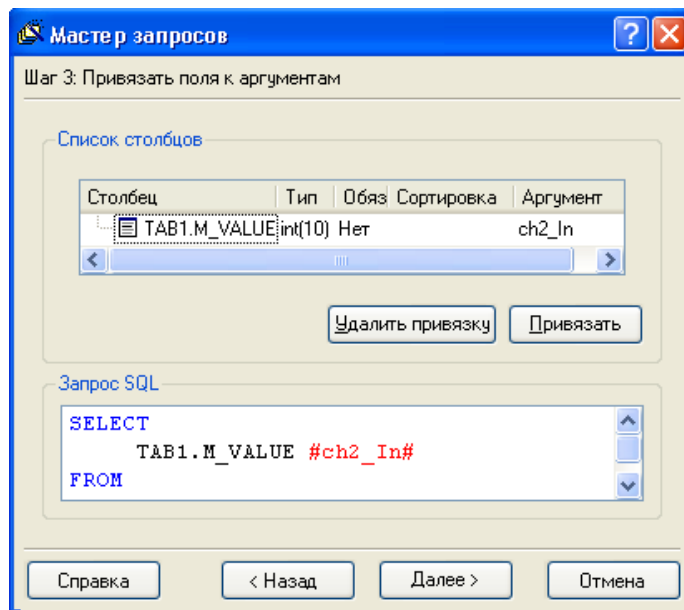
С помощью этого диалога задается привязка полей таблицы БД к аргу-

ментам – для этого нужно выделить поле и нажать кнопку **Привязать**. По этой команде на экране появится диалог выбора аргумента данной связи с БД (см. **Окно ‘Аргументы’**):



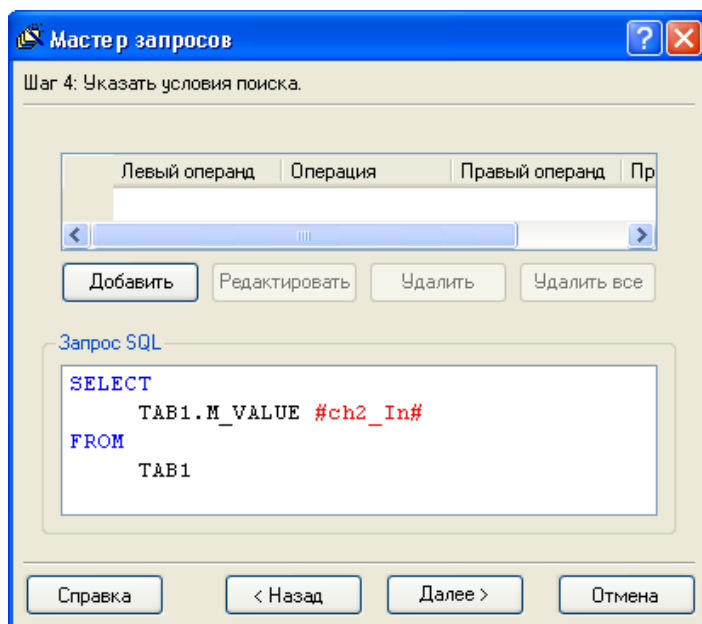
Данное окно представляет собой редактор аргументов связи с БД (см. **Табличный редактор аргументов**).

В этом окне нужно выбрать аргумент (в данном случае есть только один аргумент OUT для привязки) и нажать кнопку **Привязать** – в диалого мастера отобразится заданная связь:



Для удаления привязки выделенного поля нужно нажать кнопку **Удалить связь** данного диалога, для изменения привязки – кнопку **Связь**.

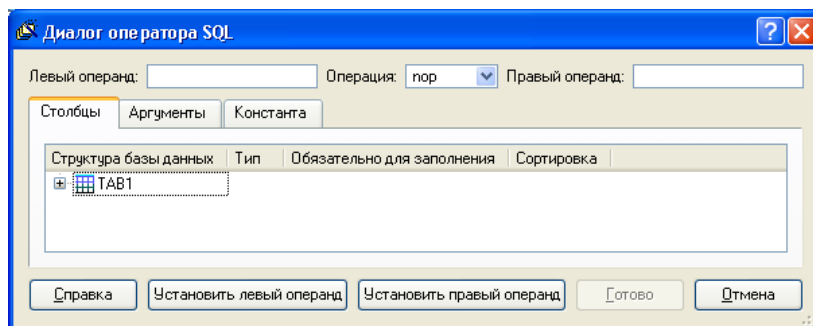
После нажатия кнопки **Далее** на экране появится следующий диалог:



Если условия выборки данных задавать не требуется, нужно в этом диалоге нажать кнопку **Готово** – созданный запрос отобразится на вновь созданной вкладке окна **Запросы** (см. **Создание SQL-запросов**).

Задание условий

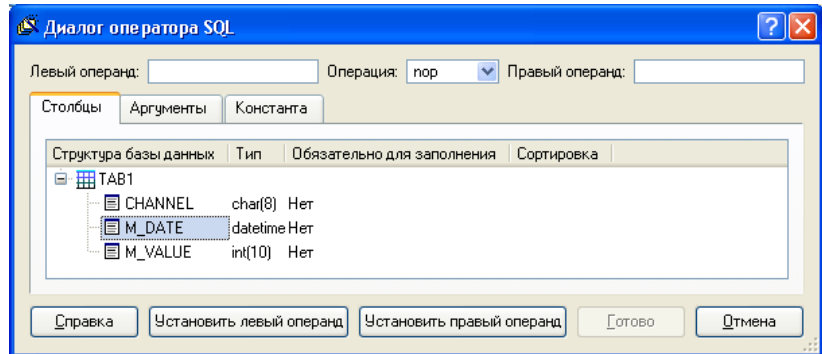
Чтобы задать условия выборки (предложение WHERE), в диалоге, показанном на рисунке выше, нужно нажать кнопку **Добавить** – по этой команде на экране появится следующий диалог:



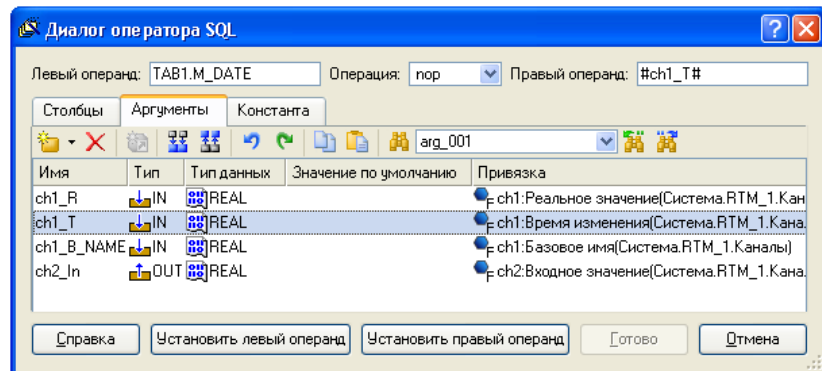
Для конфигурирования условия в этом диалоге нужно выполнить следующие действия:

- выбрать поле таблицы БД (на вкладке **Поля**), аргумент (на вкладке **Аргументы**) или задать константу (на вкладке **Константа**) и нажать кнопку **Установить левый операнд** – операнд отобра-

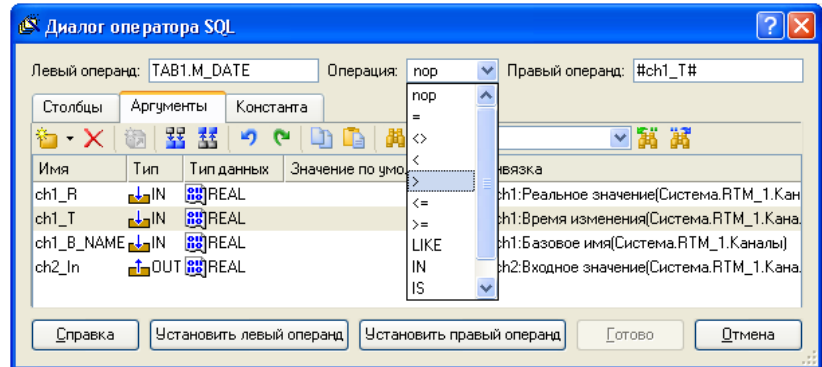
зится в поле **Левый операнд** диалога:



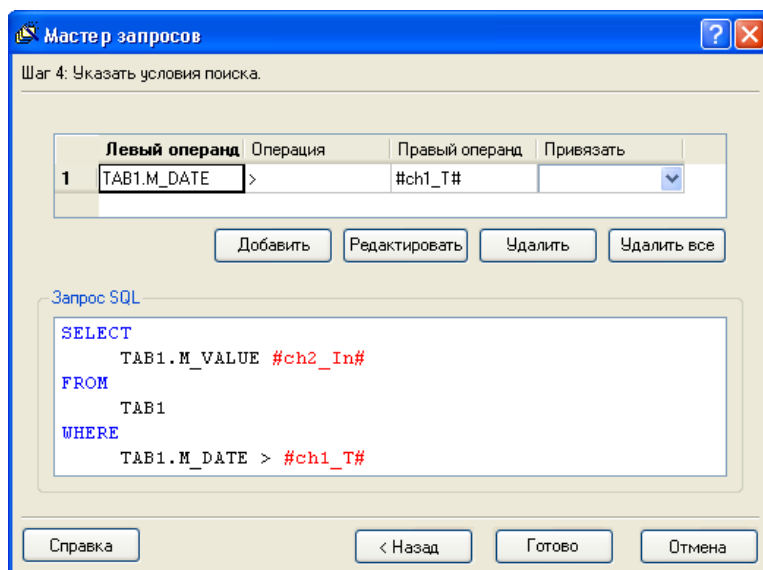
- выбрать поле таблицы БД (на вкладке **Поля**), аргумент (на вкладке **Аргументы**) или задать константу (на вкладке **Константа**) и нажать кнопку **Установить правый операнд** – операнд отобразится в поле **Правый операнд** диалога:



- в списке **Операция** выбрать оператор сравнения:

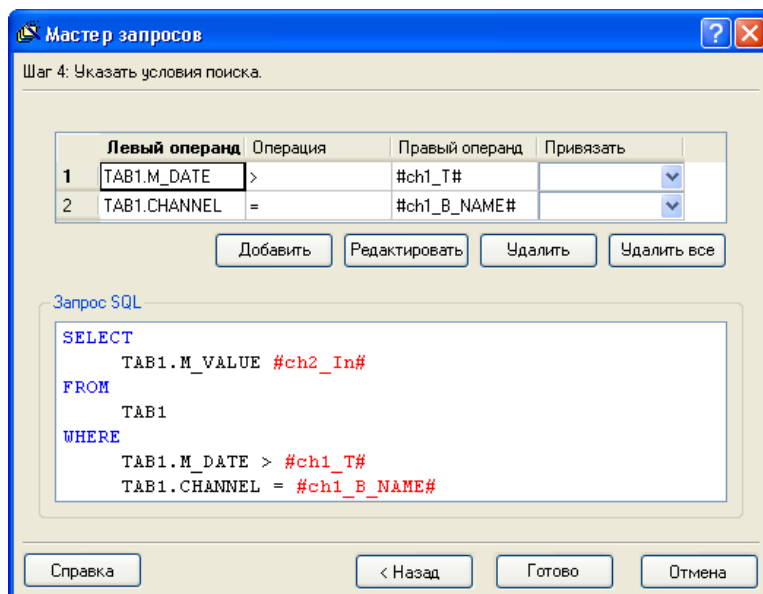


После нажатия кнопки **ОК** созданное предложение отобразится в диалоге мастера:

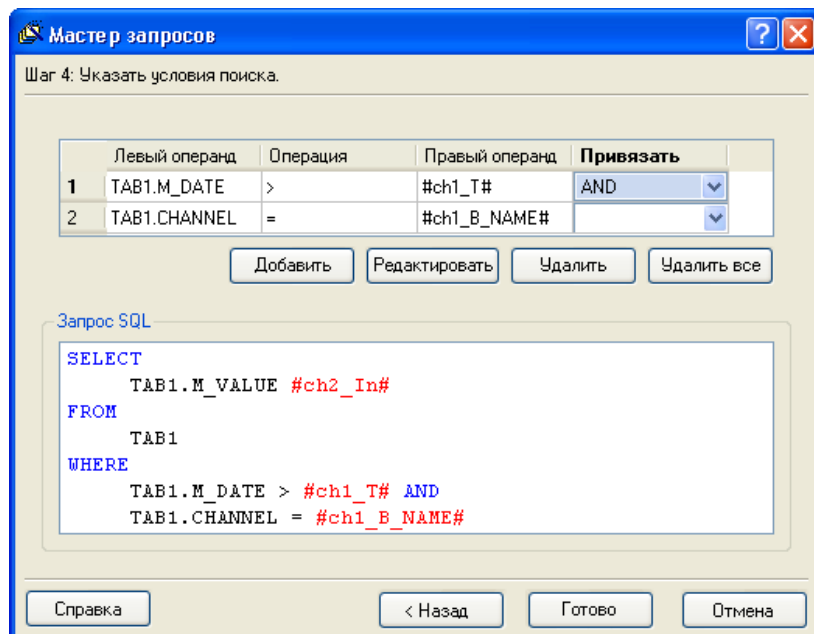


Для перехода к редактированию выделенного условия нужно нажать кнопку **Редактировать**. Чтобы удалить выделенное условие, нужно нажать кнопку **Удалить**, чтобы удалить все условия – кнопку **Удалить все**.

Чтобы добавить еще одно условие, нужно вновь нажать кнопку **Добавить** и повторить описанные выше действия:



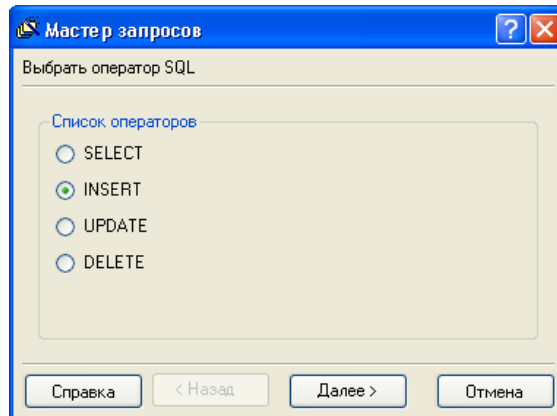
Если задано более одного условия, в списке **Привязать** необходимо выбрать оператор связи условий (**AND** или **OR**):



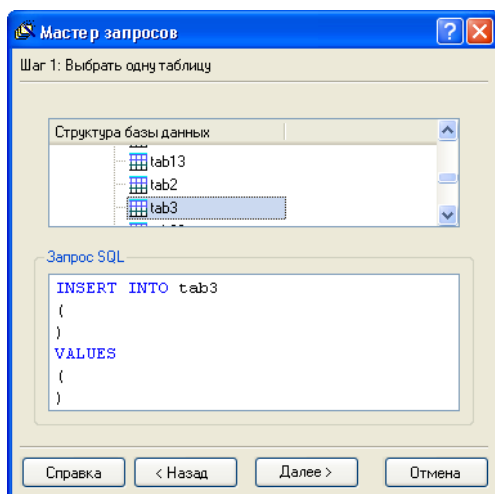
Для завершения построения запроса нужно нажать кнопку **Готово**.

Построение запроса INSERT с помощью мастера

Для построения запроса с оператором INSERT нужно выбрать этот оператор в окне мастера:



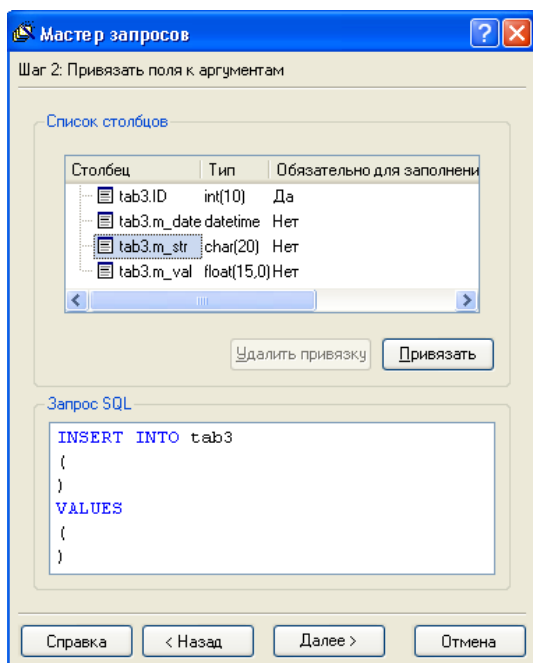
После нажатия кнопки **Вперед** на экране появится диалог, в котором нужно выбрать таблицу БД:



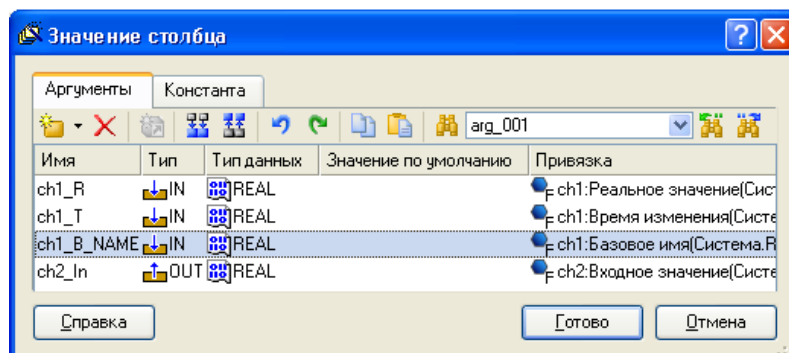
При построении запроса с помощью мастера текущий листинг запроса отображается в нижнем окне диалогов.

Все диалоги мастера снабжены кнопкой **Назад** для возврата к предыдущему шагу.

После нажатия кнопки **Вперед** на экране появится диалог, с помощью которого задаются привязки полей БД:

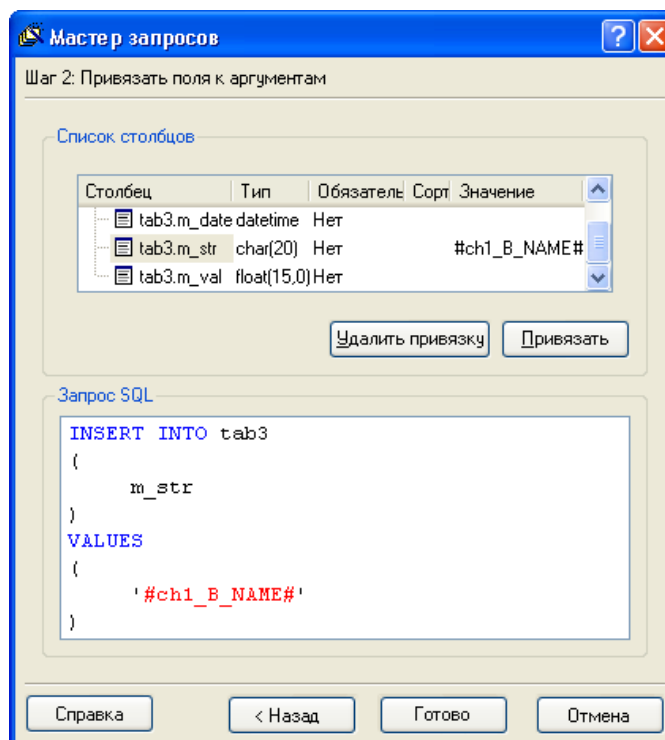


Для привязки поля БД нужно выделить его и нажать кнопку **Привязать** – по этой команде на экране появится следующий диалог:



Для привязки поля БД к аргументу нужно выбрать аргумент на вкладке **Аргументы** (эта вкладка представляет собой редактор аргументов связи с БД – см. **Табличный редактор аргументов**), для привязки к константе – ввести значение этой константы на вкладке **Константа**.

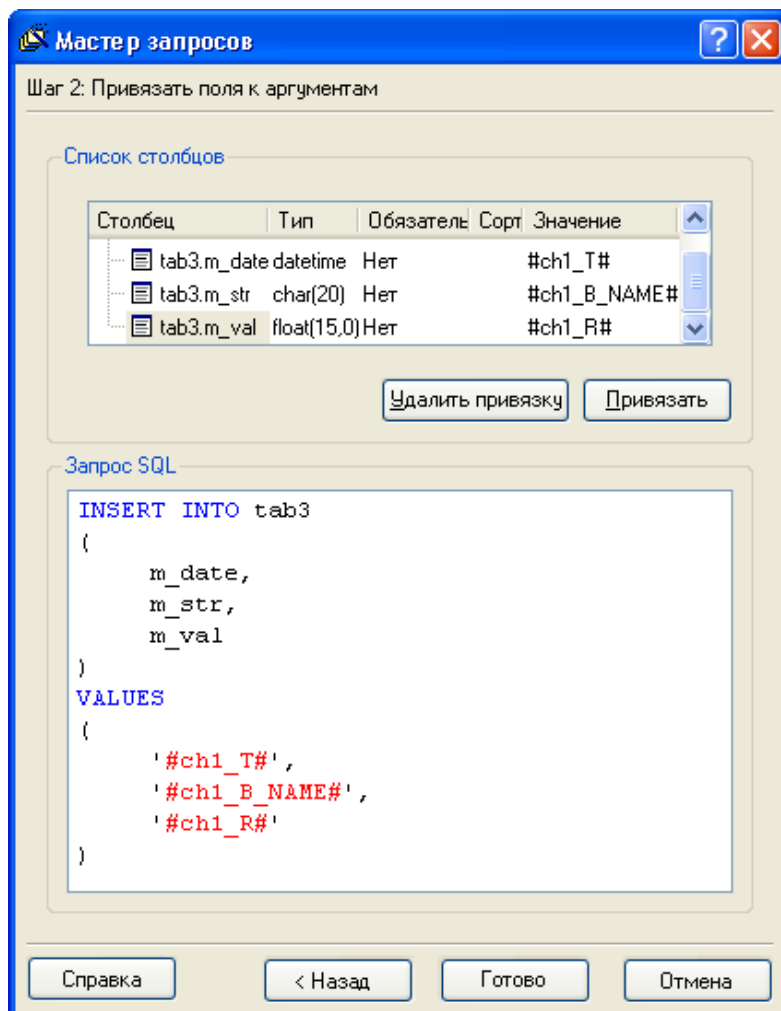
После нажатия кнопки **ОК** созданная привязка отобразится в диалог:



Для удаления привязки выделенного поля БД нужно нажать

кнопку **Удалить связь**.

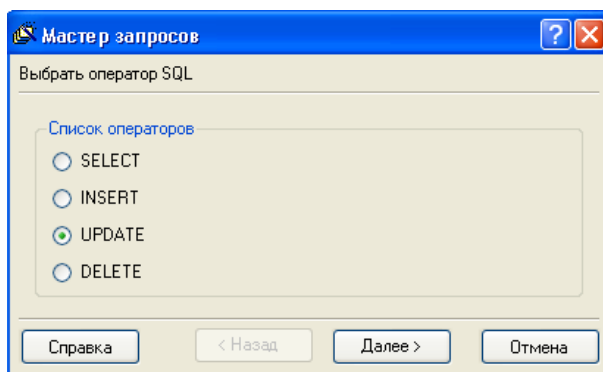
Для задания привязки других полей БД нужно повторить описанные выше действия:



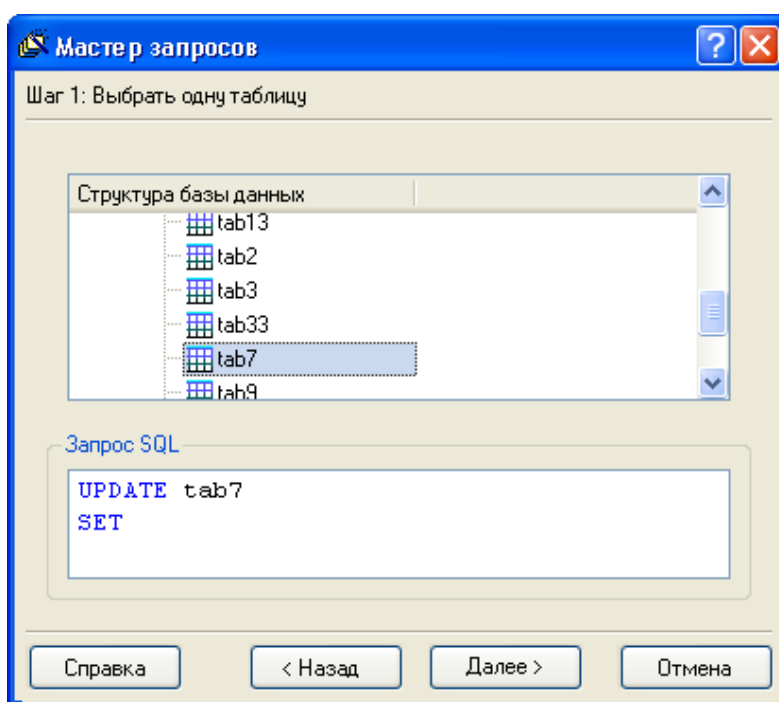
Для завершения построения запроса нужно нажать кнопку **Готово** – созданный запрос отобразится на вновь созданной вкладке окна **Запросы** (см. **Создание SQL-запросов**).

Построение запроса UPDATE с помощью мастера

Для построения запроса с оператором UPDATE нужно выбрать этот оператор в окне мастера:



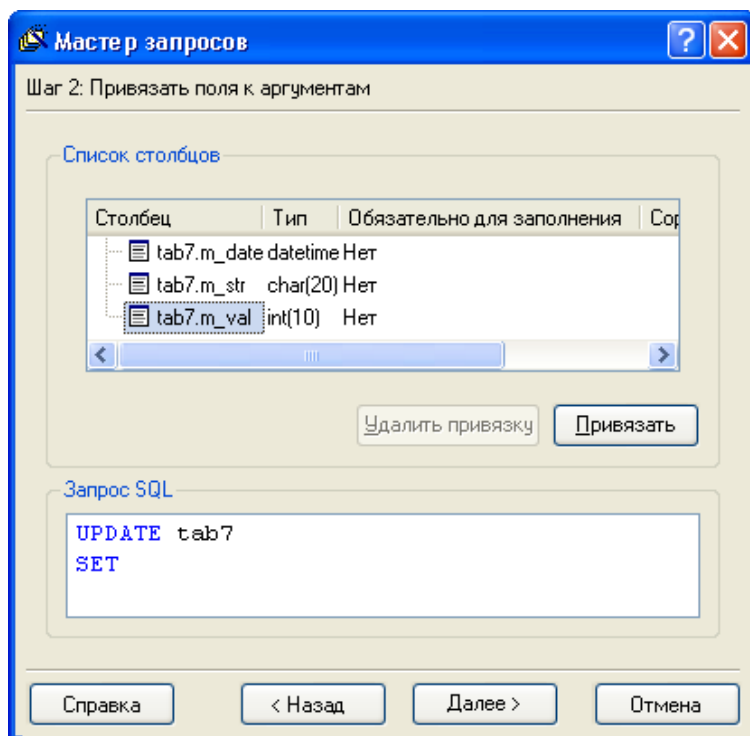
После нажатия кнопки **Вперед** на экране появится диалог, в котором нужно выбрать таблицу БД:



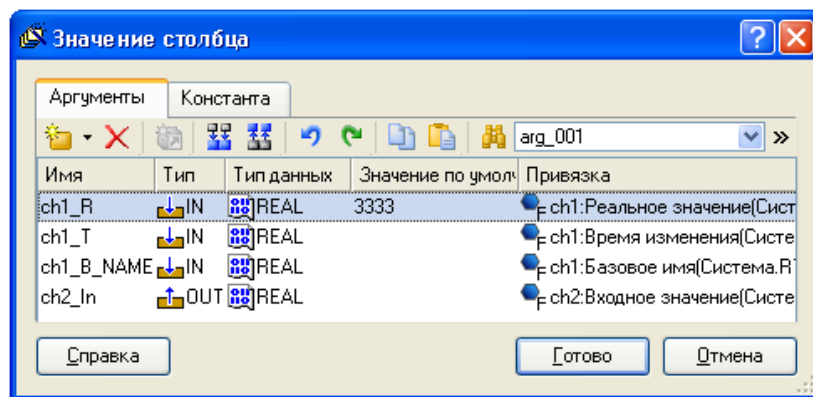
При построении запроса с помощью мастера текущий листинг запроса отображается в нижнем окне диалогов.

Все диалоги мастера снабжены кнопкой **Назад** для возврата к предыдущему шагу.

После нажатия кнопки **Вперед** на экране появится диалог, с помощью которого задаются привязки полей БД:

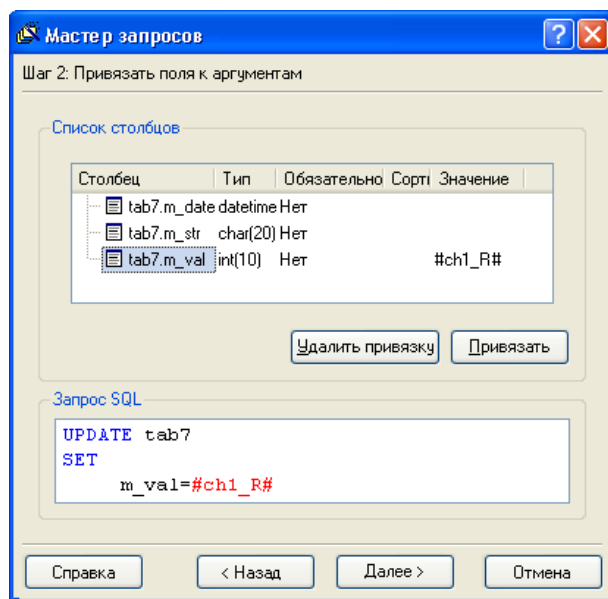


Для привязки поля БД нужно выделить его и нажать кнопку **Связь** – по этой команде на экране появится следующий диалог:



Для привязки поля БД к аргументу нужно выбрать аргумент на вкладке **Аргументы** (эта вкладка представляет собой редактор аргументов связи с БД – см. **Табличный редактор аргументов**), для привязки к константе – ввести значение этой константы на вкладке **Константа**.

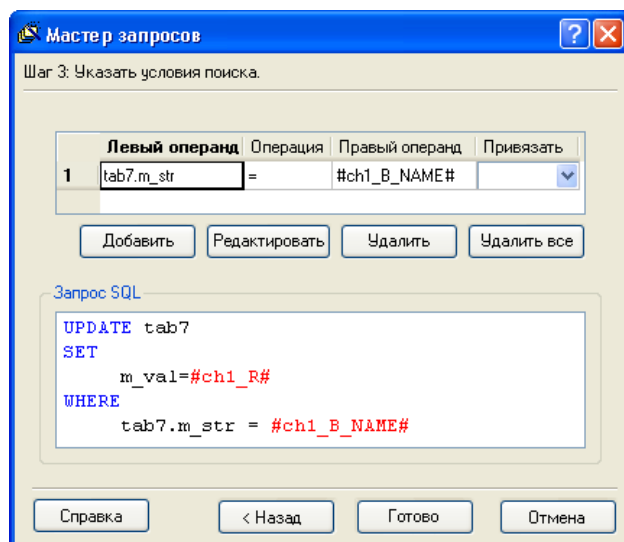
После нажатия кнопки **ОК** созданная привязка отобразится в диалоге:



Для удаления привязки выделенного поля БД нужно нажать кнопку **Удалить связь**.

После нажатия кнопки **Вперед** на экране появится диалог, с помощью которого задаются условия отбора строк для модификации (предложение WHERE – см. **Задание условий** в разделе **Построение запроса SELECT с помощью мастера**).

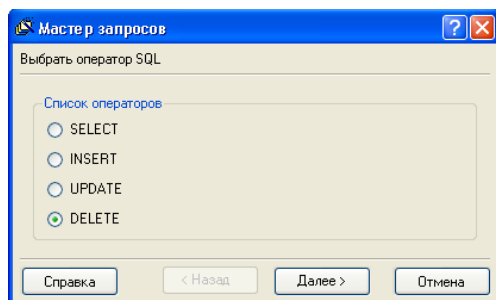
Заданные условия отобразятся в диалоге:



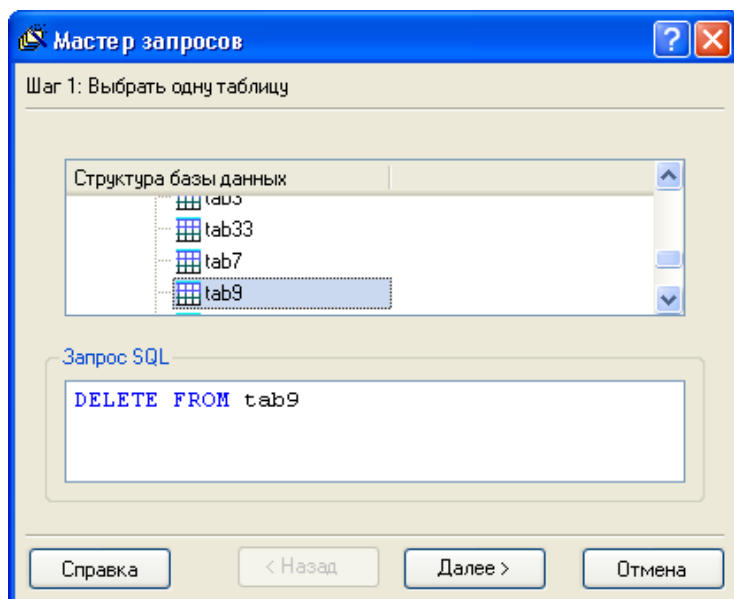
Для завершения построения запроса нужно нажать кнопку **Готово** – созданный запрос отобразится на вновь созданной вкладке окна **Запросы** (см. **Создание SQL-запросов**).

Построение запроса DELETE с помощью мастера

Для построения запроса с оператором DELETE нужно выбрать этот оператор в окне мастера:



После нажатия кнопки **Вперед** на экране появится диалог, в котором нужно выбрать таблицу БД:

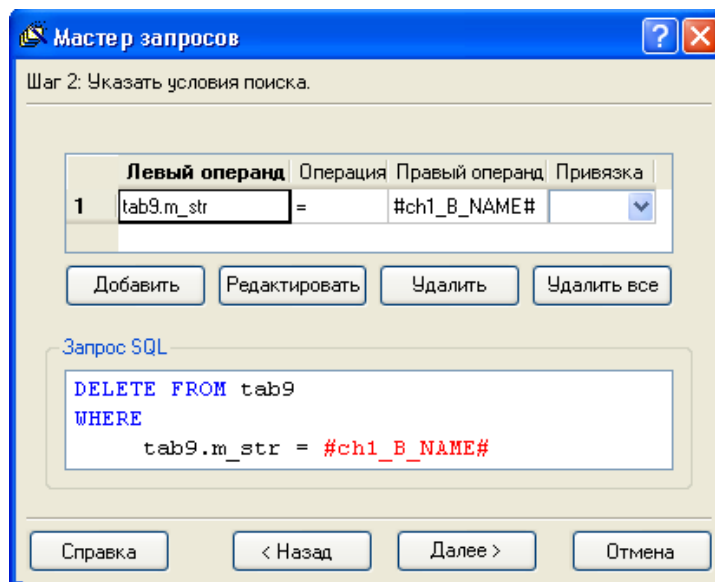


При построении запроса с помощью мастера текущий листинг запроса отображается в нижнем окне диалогов.

Все диалоги мастера снабжены кнопкой **Назад** для возврата к предыдущему шагу.

После нажатия кнопки **Вперед** на экране появится диалог, с помощью которого задаются условия отбора строк для удаления (предложение WHERE – см. **Задание условий** в разделе **Построение запроса SELECT с помощью мастера**).

Заданные условия отобразятся в диалоге:



Для завершения построения запроса нужно нажать кнопку **Готово** – созданный запрос отобразится на вновь созданной вкладке окна **Запросы** (см. **Создание SQL-запросов**).

Сочетания клавиш в редакторе связей с БД

В редакторе:

- **SPACE** – открыть вкладку (окно), на которую установлен фокус;
- **ALT+C** – открыть вкладку **Подключение**;
- **ALT+S** – открыть вкладку **Схема**;
- **ALT+W** – запустить мастер запросов;
- **ALT+A** – открыть вкладку **Аргументы**;
- **ALT+Q** – открыть вкладку **Запрос**;
- **ALT+H** – открыть контекстную справку вкладки;
- **ALT+T** – выполнить команду проверки подключения к БД (на вкладке **Подключение**).

В мастере запросов:

- **SPACE** – выбрать оператор, на который установлен фокус;
- **ALT+N (ALT+➡)** – перейти в следующий диалог;
- **ALT+B (ALT+⬅)** – перейти в предыдущий диалог;
- **ALT+C** – закрыть главное окно мастера;
- **ALT+H** – открыть контекстную справку;
- **ALT+L** – команда **Привязать**;
- **ALT+U** – команда **Удалить привязку**;
- **ALT+F** – команда **Готово**.

Обмен по DDE и NetDDE

Мониторы поддерживают обмен по DDE/NetDDE между собой и с приложениями WINDOWS, выступая одновременно в качестве сервера и клиента.

Клиент инициирует обмен с сервером и задает один из следующих режимов обмена:

- **POKE** – изменение значения указанного параметра на сервере;
- **REQUEST** – запрос значения указанного параметра от сервера;
- **ADVISE** – режим, при котором сервер посылает клиенту значение указанного параметра при его изменении.

Примеры, приведенные в этом разделе, выполнены в ОС Windows XP с использованием Excel 2000.

При обмене с сервером DDE/NetDDE в узле следует предусмотреть хотя бы один канал, запрашивающий данные у этого сервера в режиме REQUEST. Только при наличии такого канала монитор реинициализирует обмен, если связь с сервером нарушается, а затем восстанавливается. Только такие каналы отслеживают состояние сервера (если связь с сервером нарушается, им устанавливается признак аппаратной недостоверности; если связь восстанавливается, признак недостоверности сбрасывается).

Конфигурирование пользователей в проекте TRACE MODE не влияет на обмен по DDE, при этом на права пользователя ОС, инициирующего обмен с монитором из DDE-клиента, не накладывается никаких ограничений.

Для обмена по **NetDDE** узлы не требуют никакого конфигурирования.

MPB может выступать в качестве NetDDE-клиента только по отношению к другому MPB (NetDDE-серверу).

Обмен по DDE

Обмен приложение - MPB как DDE-сервер

При обмене с локальным MPB из приложения – DDE-клиента – возможно:

- чтение атрибута (0, **R**) каналов (в режиме REQUEST или ADVISE);
- чтение (REQUEST) и запись (POKE) любых других атрибутов каналов.

Запись значений в атрибуты, вычисляемые в канале, не имеет смысла.

Если в узле существуют каналы с одинаковым именем, то выполнение запроса влияет только на канал с наименьшим ID.

Если клиентом DDE является Excel, то DDE-обмен с MPB может быть сконфигурирован с помощью формул Excel и/или макросов VBA. Формула Excel может быть использована только для запроса значения (в режиме ADVISE или REQUEST) – в этом случае она имеет следующий формат:

`=<server>|<topic>!<item>`

где

server – имя сервера в формате **RTM<k>**, где **k** – индивидуальный номер узла;

topic – тема запроса (см. примеры ниже);

item – имя канала или уточненное имя атрибута.

Уточненное имя атрибута имеет следующий формат:

`<имя канала>.<номер атрибута>`

Любой из перечисленных параметров DDE-формулы Excel следует заключить в одинарные кавычки в следующих случаях:

- если параметр содержит пробелы или служебные символы, которые используются в формулах Excel (двоеточие, минус и т.п.);
- если параметр имеет вид ссылки на ячейку (примером такого параметра может служить имя канала **ch1**).

В приведенных ниже примерах узел имеет индивидуальный номер 2.

Запрос значения атрибута R

Для запроса реального значения канала **pila** в режиме ADVISE в ячейку таблицы Excel нужно записать следующую формулу:

`=RTM2|GET!pila`

В этом режиме значение в ячейке обновляется автоматически.

В режиме ADVISE монитор посылает клиенту значение канала при каждом его пересчете.

Для запроса реального значения канала **pila** в режиме REQUEST в ячейку

таблицы Excel нужно записать одну из следующих формул:

```
=RTM2|PUT!pila
=RTM2|PUT!pila.0
```

Во второй формуле тема запроса может быть произвольной. В этом режиме значение канала запрашивается и записывается в ячейку однократно при исполнении формулы.

Запросить значение атрибута (0, **R**) канала в режиме REQUEST можно также с помощью одного из следующих макросов VBA Excel (во втором макросе тема запроса может быть произвольной):

```
Sub read_pila_R()
    'PUT or GET to read R
    chNum = Application.DDEInitiate("RTM2", "GET")
    Worksheets("Sheet1").Range("F1") = Application.DDERequest(chNum, "pila")
    Application.DDETerminate chNum
End Sub
```

```
Sub read_pila_R()
    'Arbitrary topic
    chNum = Application.DDEInitiate("RTM2", "LL")
    Worksheets("Sheet1").Range("F1") = Application.DDERequest(chNum, "pila.0")
    Application.DDETerminate chNum
End Sub
```

Запрос значений других атрибутов

Для запроса значения атрибутов (3, **C**), (4, **I**), (7, **P**); (8, **W**), (26, **HL**), (27, **LL**), (28, **HA**), (29, **LA**), (30, **HW**), (31, **LW**), (79, **CODE**) и (127, **NAME**) могут быть использованы запросы двух видов:

- **topic** – короткое имя атрибута, **item** – имя канала;
- **topic** – произвольное значение (из допустимых), **item** – уточненное имя атрибута.

Для запроса значений других атрибутов могут быть использованы запросы только второго вида.

Таким образом, для запроса, например, верхнего предела (26, **HL**) канала **ch1** в ячейку таблицы Excel нужно записать одну из следующих формул:

```
=RTM2|HL!'ch1'
=RTM2|PUT!ch1.26
```

Запросить значение атрибута из приведенного выше перечня (ниже в примерах запрашивается кодировка) можно также с помощью одного из

следующих макросов (для запроса значений других атрибутов нужно использовать второй макрос):

```
Sub read_ch1_CODE()  
    'Short name to read attribute  
    chNum = Application.DDEInitiate("RTM2", "CODE")  
    Worksheets("Sheet1").Range("F1") = Application.DDERequest(chNum, "ch1")  
    Application.DDETerminate chNum  
End Sub
```

```
Sub read_ch1_CODE()  
    'Arbitrary topic  
    chNum = Application.DDEInitiate("RTM2", "LL")  
    Worksheets("Sheet1").Range("F1") = Application.DDERequest(chNum, "ch1.79")  
    Application.DDETerminate chNum  
End Sub
```

Задание значений атрибутов

Для задания значения атрибутов (2, **In**), (3, **C**), (4, **I**), (7, **P**); (8, **W**), (26, **HL**), (27, **LL**), (28, **HA**), (29, **LA**), (30, **HW**), (31, **LW**), (79, **CODE**) и (127, **NAME**) могут быть использованы макросы двух видов:

- **topic** – короткое имя атрибута, **item** – имя канала;
- **topic** – произвольное значение (из допустимых), **item** – уточненное имя атрибута.

Для задания значений других атрибутов могут быть использованы макросы только второго вида.

Таким образом, для записи в атрибут (2, **In**) канала **ch1** значения из ячейки C3 таблицы Excel можно использовать один из следующих макросов:

```
Sub write_ch1_In()  
    'PUT or GET to write into In  
    chNum = Application.DDEInitiate("RTM2", "PUT")  
    Application.DDEPoke chNum, "ch1", Worksheets("Sheet1").Range("C3")  
    Application.DDETerminate chNum  
End Sub
```

```
Sub write_ch1_In()  
    'Arbitrary topic  
    chNum = Application.DDEInitiate("RTM2", "HA")
```

```
Application.DDEPoke chNum, "ch1.2", Work-  
sheets("Sheet1").Range("C3")  
Application.DDETerminate chNum  
End Sub
```

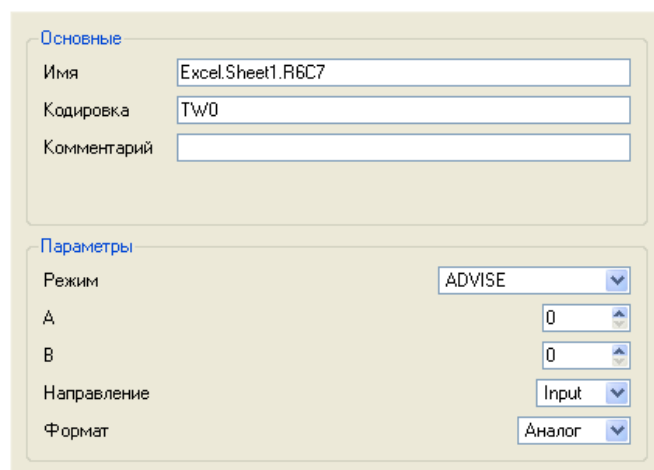
Обмен MPB как DDE-клиент - приложение

Если монитор выступает в роли DDE-клиента, для конфигурирования обмена используются переменные DDE (см. **Редактор переменной DDE**).

Рассмотрим обмен на одном компьютере между монитором и Excel, выступающим в качестве DDE-сервера.

Запрос значения в режиме ADVISE

Создадим в узле канал **ch1** и свяжем его с переменной DDE, параметры которой зададим следующим образом:

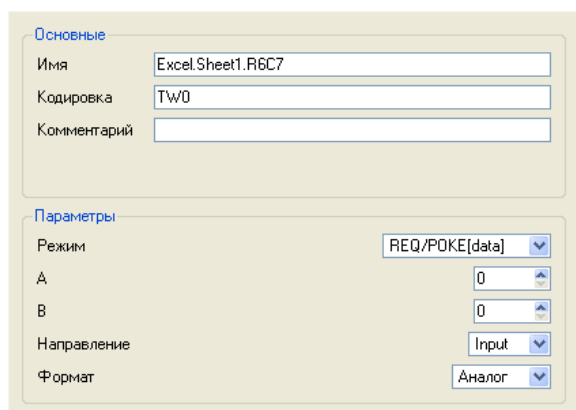


Откроем Excel, создадим новую книгу и запустим узел – установится связь монитора с книгой Excel, активной в данный момент, в режиме ADVISE. В этом режиме DDE-сервер (Excel) будет посылать DDE-клиенту (монитору) значение (при каждом его изменении) ячейки **G6** (Row=6, Column=7) листа **Sheet1**, и это значение будет записано во вход канала **ch1**.

Запрос значения в режиме REQUEST

Для запроса в режиме REQUEST тип переменной DDE должен быть INPUT.

Создадим в узле канал **ch2** и свяжем его с переменной DDE, параметры которой зададим следующим образом:



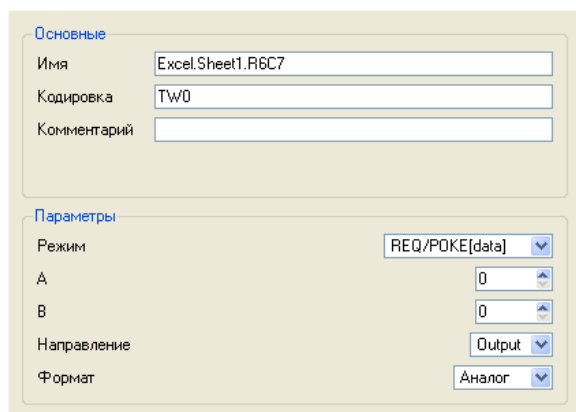
Для запроса в режиме REQUEST параметру **Режим** переменной DDE можно также задать значение **REQ/POKE[data/r]** или **REQ/POKE[data/n]**.

Откроем Excel, создадим новую книгу и запустим узел – установится связь монитора с книгой Excel, активной в данный момент, в режиме REQUEST. В этом режиме монитор генерирует в соответствующем потоке запросы на чтение значения ячейки **G6** листа **Sheet1** и записывает это значение во вход канала **ch2**.

Задание значения в режиме POKE

Для передачи в указанную ячейку таблицы Excel выходного значения канала (атрибут 9, **Q**), привязанная к этому каналу переменная DDE должна иметь тип OUTPUT.

Создадим в узле канал **ch_p** и свяжем его с переменной DDE, параметры которой зададим следующим образом:



Для записи значения в Excel параметру **Режим** переменной DDE можно

также задать значение **REQ/POKE[data/r]** или **REQ/POKE[data/n]**.

Откроем Excel, создадим новую книгу и запустим узел – установится связь монитора с книгой Excel, активной в данный момент, в режиме POKE. В этом режиме монитор генерирует в соответствующем потоке запрос на запись выходного значения (при каждом его изменении) канала **ch_p** в ячейку **G6** листа **Sheet1**.

Обмен по NetDDE

Настройка ОС для обмена по NetDDE

Запуск служб NetDDE

При обмене по NetDDE используются следующие службы Windows (должны быть запущены как на клиенте, так и на сервере):

- NT DDE-сервер (clipsrv.exe);
- NETWORK DDE – обеспечение обмена по NetDDE (netdde.exe);
- NETWORK DDE DSDM (Network DDE Service Data Manager) – совместный доступ к данным DDE.

Доступ к разделяемым данным DDE

Если необходимые службы запущены, монитор, в случае его запуска пользователем с правами администратора, автоматически создает разделяемый DDE-ресурс **RTM<k>\$** (**k** – индивидуальный номер узла). Без конфигурирования доступа к такому ресурсу могут обращаться пользователи, которые идентичны пользователю сервера (логин и пароль) и имеют права администратора. Для других пользователей доступ к ресурсу должен быть сконфигурирован.

При остановке монитор удаляет разделяемый DDE-ресурс **RTM<k>\$**.

Обмен приложение - MPB как NetDDE-сервер

При обмене с монитором – сервером NetDDE – из приложения возможен запрос значений атрибута (0, **R**) (ADVISE или REQUEST), а также запрос (REQUEST) и задание (POKE) значений других атрибутов каналов узла (как удаленного, так и локального).

Имя сервера задается в следующем формате:


```
\\<name>\NDDE$
```

где **name** – имя компьютера, на котором запущен МРВ.

Тема запроса задается как **RTM<k>\$**, где **k** – индивидуальный номер узла.

Если в узле существуют каналы с одинаковым именем, то выполнение запроса влияет только на канал с наименьшим ID.

В приведенных ниже примерах узел имеет индивидуальный номер 3.

Запрос атрибута R

Для запроса реального значения канала **ch1** узла **nodeA** в режиме ADVISE в ячейку таблицы Excel нужно записать следующую формулу (параметр **Update** в диалоге **Links** должен иметь значение **Automatic**):

```
='\nodeA\NDDE$'|'RTM3$.OLE'!'ch1'
```

Запросить значение атрибута (0, **R**) канала **ch1** в режиме REQUEST из Excel можно с помощью следующего макроса:

```
Sub read_ch1_R()  
    chNum = Application.DDEInitiate("\\nodeA\NDDE$",  
    "RTM3$")  
    Worksheets("Sheet1").Range("C1") = Application.  
    DDERequest(chNum, "ch1")  
    Application.DDETerminate chNum  
End Sub
```

Запрос других атрибутов

Для запроса значения произвольного атрибута в режиме REQUEST можно использовать макрос, в котором в качестве **item** указано уточненное имя атрибута.

В следующем макросе запрашивается значение атрибута **32** канала **ch1** узла **nodeA**:

```
Sub read_attr32()  
    chNum = Application.DDEInitiate("\\nodeA\NDDE$",  
    "RTM3$")  
    Worksheets("Sheet1").Range("C1") = Application.  
    DDERequest(chNum, "ch1.32")  
    Application.DDETerminate chNum  
End Sub
```

Задание атрибутов

Для задания значения произвольного атрибута можно использовать мак-

рос, в котором в качестве **item** указано уточненное имя атрибута.

В следующем макросе в атрибут **32** канала **ch1** узла **nodeA** записывается значение из ячейки C3 таблицы Excel:

```
Sub write_attr32()  
    chNum = Application.DDEInitiate("\\nodeA\NDDE$",  
    "RTM3$")  
    Application.DDEPoke chNum, "ch1.32", Work-  
    sheets("Sheet1").Range("C3")  
    Application.DDETerminate chNum  
End Sub
```

Для записи в атрибут (2, **In**) указание его номера не обязательно:

```
Sub write_ch1_In()  
    chNum = Application.DDEInitiate("\\nodeA\NDDE$",  
    "RTM3$")  
    Application.DDEPoke chNum, "ch1", Work-  
    sheets("Sheet1").Range("C3")  
    Application.DDETerminate chNum  
End Sub
```

Обмен между MPB по NetDDE

Обмен по NetDDE между мониторами конфигурируется в узле, выступающем в роли клиента, – для этого используются переменные DDE (см. **Редактор переменной DDE**).

Для корректного выполнения запросов при обмене между MPB в режимах REQUEST и POKE для параметра **Режим** следует задавать значение **REQ/POKE[data/n]** или **REQ/POKE[data/r]**.

Если в узле-сервере существуют каналы с одинаковым именем, то выполнение запроса влияет только на канал с наименьшим ID.

В приведенных ниже примерах предполагается, что узел, выступающий в роли NetDDE-сервера, имеет индивидуальный номер 1 и запущен на компьютере с именем **nodeA**.

Запрос R

Для запроса в режиме ADVISE значения атрибута **R** создадим в узле – клиенте NetDDE – канал **ch2** и свяжем его с переменной DDE, параметры которой зададим следующим образом:

Запустим мониторы – между ними установится связь в режиме ADVISE. В этом режиме сервер (**nodeA**) будет посылать клиенту значение канала **ch1**, и это значение будет записываться во вход канала **ch2**.

В режиме ADVISE монитор посылает клиенту значение канала при каждом его пересчете.

Для запроса в режиме REQUEST значения атрибута **R** создадим в узле – клиенте NetDDE – канал **ch3** и свяжем его с переменной DDE, параметры которой зададим следующим образом:

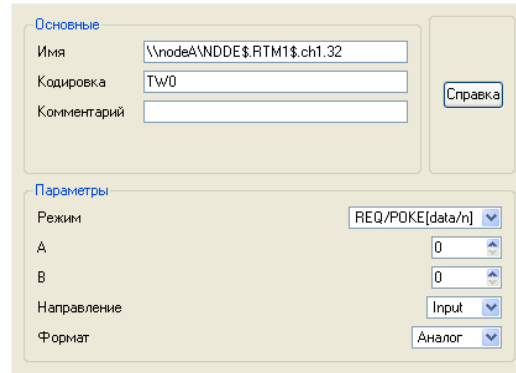
Для запроса в режиме REQUEST параметру **Режим** переменной DDE можно также задать значение **REQ/POKE[data/r]**.

Запустим мониторы – между ними установится связь в режиме REQUEST. В этом режиме клиент генерирует в соответствующем потоке запросы на чтение значения атрибута (0, **R**) канала **ch1** сервера и записывает это значение во вход канала **ch3**.

Запрос других атрибутов в режиме REQUEST

Для запроса в режиме REQUEST значения произвольного атрибута (в

примере сконфигурировано чтение атрибута 32) создадим в узле – клиенте NetDDE – канал **ch4** и свяжем его с переменной DDE, параметры которой зададим следующим образом:



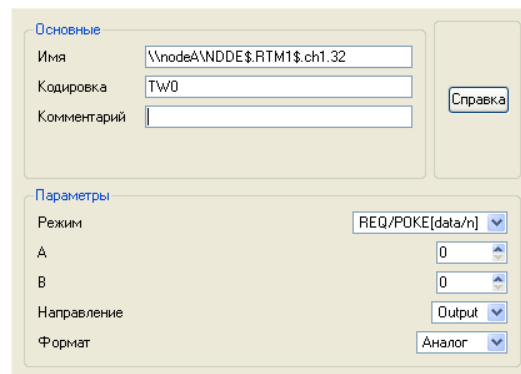
Для запроса в режиме REQUEST параметру **Режим** переменной DDE можно также задать значение **REQ/POKE[data/r]**.

Запустим мониторы – между ними установится связь в режиме REQUEST. В этом режиме клиент генерирует в соответствующем потоке запросы на чтение значения указанного атрибута (32) канала **ch1** сервера и записывает это значение во вход канала **ch4**.

Задание произвольных атрибутов

Для реализации режима POKE переменная DDE должна иметь тип OUTPUT.

Для задания значения произвольного атрибута (в примере сконфигурирована запись в атрибут 32) создадим в узле – клиенте NetDDE – канал **ch5** и свяжем его с переменной DDE, параметры которой зададим следующим образом:



Запустим мониторы – между ними установится связь в режиме POKE. В

этом режиме клиент генерирует в соответствующем потоке запрос на запись выходного значения (при каждом его изменении) канала **ch5** в указанный атрибут (32) канала **ch1** сервера.

Для записи значения в атрибут (2, **In**) указание его номера не обязательно:

Для данного режима параметру **Режим** переменной DDE можно также задать значение **REQ/POKE[data/r]**.

Обмен по OPC

Мониторы поддерживают обмен с серверами OPC, в том числе с серверами OPC HDA.

Для функционирования OPC-клиента в узле **Logger** нужно в файле *.cnf задать ключ **OPCSMSK=1**.

Метод поиска OPC-серверов задается в браузере OPC ИС (см. **Раздел 'Браузер OPC'** в разделе **Вкладка 'Базовый редактор'**).

Специализированный монитор – **OPC-сервер TRACE MODE 6** – обеспечивает OPC-клиентам, поддерживающим спецификацию OPC DA 2.0, чтение/запись атрибутов каналов исполняемого узла.

Для мониторинга/управления обменом в МРВ, выступающим в роли OPC-клиента, используется переменная **@e OPC**, для вывода таблицы об OPC-серверах – **SubNum=645**.

OPC-сервер TRACE MODE 6

Для доступа OPC-клиента к каналам узла узел должен исполняться под управлением специализированного монитора – OPC-сервера TRACE MODE 6.

Регистрация OPC-сервера TRACE MODE 6

Для регистрации OPC-сервера TRACE MODE 6 в ОС нужно выполнить следующие процедуры (указанные ниже файлы находятся в директории установки монитора):

- запустить командный файл **register_opc_proxy_stub.cmd** (однократная процедура, после которой сервер можно регистрировать/разрегистрировать). Для отмены нужно запустить командный файл **unregister_opc_proxy_stub.cmd**;
- запустить командный файл **register_server.cmd** (для отмены регистрации нужно запустить файл **unregister_server.cmd**).

Запуск OPC-сервера TRACE MODE 6

Для задания узла и запуска OPC-сервера нужно выполнить одну из следующих команд:

```
%OPC-сервер%/tm6opc.exe %узел%/*.dbb  
%OPC-сервер%/tm6opc.exe %узел%/*.dbb /RUN
```

В первом случае сервер только загружает узел, во втором – запускает.

Если узел был однажды указан, для его запуска в дальнейшем достаточно выполнить следующую команду:

```
%ОПС-сервер%/tm6opc.exe
```

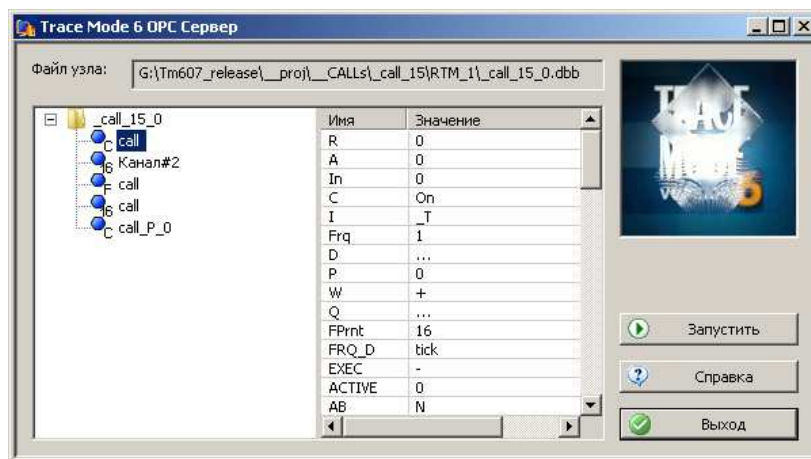
При запросе клиента ОПС-сервер запускается автоматически.

ОПС-сервер поддерживает спецификации OPC DA 2.0 и OPC HDA.

Для передачи ОПС-клиенту архивных данных необходимо в канале ОПС-сервера, получающего данные от архивируемого канала в узле МРВ, установить флаг **Запрос времени изменения** и задать признак архивирования в СПАД. В настройках узла ОПС-сервера архив задавать не надо.

Интерфейс оператора ОПС-сервера TRACE MODE 6

Аналогично оболочке профайлера без поддержки графических экранов (см. **Профайлер без поддержки графических экранов**), графическая оболочка ОПС-сервера отображает каналы узла:



В поле **Файл узла** отображается полный путь к файлу *.dbb.

Кнопка **Запустить** (после запуска – **Остановить**) предназначена для запуска и останова узла.

При нажатии кнопки **Выход** сервер останавливается и выгружается.

Для вывода информации об обмене в протокол профайлера нужно установить бит 19 (0x80000) **@Debug**.

Настройка безопасности ОС

Если необходим удаленный доступ к ОПС-серверу TRACE MODE 6, то на компьютере, где сервер должен быть запущен, может потребоваться кон-

фигурирование параметров ряда служб (в том числе DCOM, брандмауэра и самого сервера).

MPB как клиент сервера OPC HDA

В качестве канала обмена по OPC HDA (см. **Подтип 3**) может выступать:

- канал CALL, для которого не задан тип вызова и задана привязка к переменной OPC HDA (см. **Редактор переменной OPC HDA**). Такому каналу CALL MPB автоматически устанавливает тип вызова 55, **OPC_HDA**;

При конфигурировании в ИС тип вызова 55, **OPC_HDA** каналу CALL задать нельзя (см. **Атрибуты канала класса CALL**).

- числовой канал, привязанный к переменной OPC HDA. Подтип такого канала – 3.2, атрибут 123 также принимает значение 55, **OPC_HDA**.

Обмен по OPC HDA с помощью канала CALL

В канале CALL должны быть созданы нулевой и первый аргументы, задающие временной интервал (**T_FROM=arg0**, **T_TO=arg1**), за который запрашиваются исторические данные.

Полученные данные записываются в аргументы канала: в четный аргумент (начиная со второго) – значение, в последующий нечетный – время значения. Необходимое количество пар аргументов создается монитором.

Обмен по OPC HDA с помощью числового канала

Временной интервал, за который запрашиваются исторические данные, задается следующим образом: **T_FROM=канал.45** (см. **Атрибуты каналов, отображаемые профайлером**), **T_TO=<текущее время>**.

Если в выборке существуют значения с метками времени, большими, чем **канал.45**, то максимальная метка времени присваивается атрибуту 45 канала, а соответствующее значение – атрибуту 2 канала.

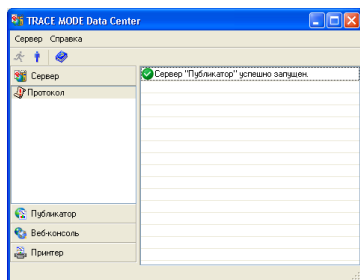
Полученная выборка исторических данных записывается в архив SIAD, если канал архивируется, и может быть отображена с помощью ГЭ/еГЭ **Тренд** (см. **ГЭ 'Тренд'**), а также выведена в генерируемый документ (на тренд – см. **Вставка тренда**).

TRACE MODE Data Center

При запуске в сети, TRACE MODE Data Center обеспечивает выполнение следующих функций:

- вывод генерируемых документов на принтер, используемый по умолчанию (см. **Генерация файлов документов**), мониторинг очереди печати и управление печатью документов (с помощью сервера печати, входящего в состав Data Center);
- управление публикациями на веб-сервере **Публикатор** (входит в состав Data Center), и доступ удаленных пользователей к публикациям;
- выполнение узлов **EmbeddedConsole** в веб-браузерах (обеспечивается сервером **Веб-консоль**, входящим в состав Data Center);
- выполнение узлов **EmbeddedConsole** в мобильных устройствах.

В состав Data Center входит графическая консоль:






Хранилище Data Center

По умолчанию, хранилищем Data Center является директория **%Data Center%\acs**. Эта директория имеет следующую структуру:

- папка **wgc** – сайт, доступ к которому обеспечивает сервер **Веб-консоль**;
- папка **printer**;
- папка **publisher** – сайт, доступ к которому обеспечивает сервер **Публикатор**.

Консоль Data Center



После запуска Data Center его иконка  отображается в области уведомлений панели задач (консоль Data Center свернута). Иконка снабжена контекстным меню, содержащим следующие команды (меню открывается только в том случае, если консоль Data Center свернута):

- **Открыть** – развернуть консоль Data Center (консоль разворачивается также при нажатии ЛК на иконке);
-  **Старт** – запустить Data Center;
-  **Стоп** – остановить Data Center;
- **Настройки** – открыть диалог конфигурирования Data Center (эта команда доступна, если Data Center остановлен);
- **О программе** – информация о Data Center и состоянии серверов;
- **Выход** – завершить работу Data Center.


Собственно консоль Data Center содержит меню **Сервер** и **Справка**, главную панель инструментов и вкладки **Сервер**, **Публикатор**, **Веб-консоль** и **Принтер**. Каждая вкладка содержит меню **Правка**, панель инструментов и контекстное меню.

В консоли поддерживаются типовые сочетания клавиш (см. **Сочетания клавиш в ИС**).

Меню и главная панель инструментов консоли

Меню **Сервер** и главная панель инструментов консоли включают команды  **Старт**,  **Стоп**, **Настройки** и **Выход**, аналогичные одноименным командам контекстного меню иконки Data Center на панели задач (см. **Консоль Data Center**).


Меню **Справка** и главная панель инструментов консоли включают следующие команды:

-  **Справка (F1)** – вызов контекстной справки;
- **О программе** – информация о Data Center и состоянии серверов.

Диалог конфигурирования Data Center

Диалог конфигурирования Data Center содержит вкладки **Общие**, **Публикатор** и **Веб-консоль**.

На вкладке **Общие** задаются следующие параметры:

- **Базовая директория** – хранилище Data Center;
- **Сетевой порт** – порт сетевого взаимодействия Data Center с мониторами; при выборе опции **авто** порт назначается операционной системой;
- **Минимизировать при закрытии окна** – если этот флаг установлен, по команде закрытия окна  консоль Data Center сворачивается; в противном случае Data Center завершает работу.

На вкладке **Публикатор** задаются следующие параметры:

- **Порт веб-сервера** – порт доступа к сайту по HTTP (по умолча-

нию – 80, стандартный порт HTTP);

- флаг **Перезаписывать при совпадении имен** – если этот флаг установлен, ресурс (публикация) перезаписывается, если он создается для узла и документа с теми же именами;
- флаг **Разрешить просмотр по Bluetooth** – если этот флаг установлен, публикации доступны для просмотра на устройстве, подключенном по Bluetooth;
- **Скрыть фильтр по умолчанию** – если этот флаг установлен, в браузере не отображается фильтр по умолчанию (см. **Вкладка 'Публикатор'**).

На вкладке **Веб-консоль** задаются следующие параметры:

- **Порт веб-сервера** – порт доступа к сайту по HTTP (по умолчанию – 81).
- **Таймаут отключения прокси-сервера** – таймаут отключения прокси-сервера (входит в состав Data Center) в отсутствие обмена данными (1...60 мин, **бесконечный** – не отключать).

Если для веб-сервера задан HTTP-порт, отличный от стандартного (80), явное указание порта в адресе сайта обязательно: **http://192.168.2.46:81**.

Для корректной работы апплетов в браузере рекомендуется установить JRE Sun Microsystems.

В разделах **Разрешенные адреса** и **Супервизоры** вкладки конфигурируются списки соответственно разрешенных и привилегированных пользователей (последние имеют приоритет по доступу к ресурсам сайта). Для управления списками используются соответствующие кнопки **✗** (удалить выделенную позицию) и **+** (добавить пользователя). При добавлении пользователя на экране появляется диалог, в котором нужно задать DNS-имя, IP-адрес или адрес Bluetooth. Если заданный адрес некорректен, он не добавляется в список, а на экране появляется соответствующее сообщение.







Вкладка 'Сервер'

На этой вкладке отображается протокол Data Center.


Вкладка 'Публикатор'

Эта вкладка в виде дерева отображает публикации документов на сервере **Публикатор**.

Меню **Правка** и панель инструментов вкладки содержат следующие команды:

-  **Новый фильтр** – создать новый фильтр; список фильтров отображается на первой странице сайта. Публикатор содержит встроенный **фильтр по умолчанию**, в соответствии с которым публикации группируются по источнику. Имя группы соответствует имени экспортированного узла TRACE MODE 6 (**<имя файла prj>_<ordinal>** – см. **Файлы узла, создаваемые при экспорте**). Если в дереве не создана группа для публикаций узла, сервер создает ее автоматически. Для публикаций в качестве описания отображается дата и время.
При создании нового фильтра на экране появляется диалог его конфигурирования. В диалоге задаются следующие параметры:
 - **Имя** – имя фильтра;
 - **Скрыть документы без категории** – если этот флаг установлен, в дереве не отображаются документы без категории (см. **Использование разработанных шаблонов**);
 - **Защита** – если этот флаг установлен, для доступа к данной группе (фильтру) в браузере потребуется ввести логин и пароль, заданные в соответствующих полях диалога. Защита может быть также установлена для фильтра по умолчанию;
-  **Удалить фильтр** – удалить выделенный фильтр (при удалении фильтра публикации не удаляются). Встроенный фильтр по умолчанию удалить нельзя;
-  **Редактировать фильтр** – редактировать параметры выделенного фильтра;
-  **Новый объект (Ins)** – создать подгруппу в выделенной группе дерева. По этой команде на экране появляется диалог, в котором задаются следующие параметры:
 - **Публичное имя** – имя для отображения в дереве;
 - **Истинное имя** – уникальное истинное имя группы. Если задается уже используемое истинное имя, группа не создается, а на экране появляется соответствующее сообщение.
Если истинное имя не задано, фильтр для создаваемой группы отсутствует. Если **Истинное имя = <имя файла prj>_<ordinal>**, данная группа будет содержать публикации указанного узла. Если в качестве истинного имени задано натуральное число, группа будет содержать публикации соответствующей категории.
 -  **Изображение** – открыть диалог для выбора пиктограммы из библиотеки изображений публикатора;
-  **Удалить объект (Del)** – удалить выделенный объект дерева.
При удалении группы на экране появляется диалог, в котором можно задать удаление, в том числе, публикаций группы (**Удалить**) или удаление с сохранением публикаций (**Удалить, но сохра-**

нить документы);

-  **Редактировать объект (Enter)** – редактировать выделенный объект дерева. При редактировании истинное имя **<имя файла prj>_<ordinal>** группы, которая уже содержит публикации, изменить нельзя;
- **Редактировать библиотеку изображений** – открыть редактор библиотеки изображений публикатора.

Команды создания, удаления и редактирования объекта могут быть также выполнены из контекстного меню дерева.



Для перегруппировки объектов дерева используется метод drag-n-drop, а также следующие команды:


- **CTRL+↓** и **CTRL+↑** – переместить группу на одну позицию вниз/вверх в списке групп того же уровня (перемещение возможно только в пределах группы старшего уровня);
- **CTRL+→** – если выделена группа1, за которой следует группа2 того же уровня, то в результате выполнения данной команды группа1 становится подгруппой группы2;
- **CTRL+←** – повысить уровень группы на 1.

Вкладка 'Веб-консоль'




Эта вкладка отображает ресурсы, сконфигурированные из узлов **EmbeddedConsole**.

Меню **Правка**, панель инструментов и контекстное меню вкладки содержат следующие команды:

-  **Новая папка (Ctrl+Ins)** – создать группу дерева. По этой команде на экране появляется диалог, в котором для группы задаются имя и пиктограмма из библиотеки изображений веб-консоли;
-  **Новый узел (Ins)** – добавить ресурс в выделенную группу дерева. По этой команде открывается диалог, в котором задаются следующие параметры:

- **Имя** – имя ресурса для отображения;
-  **Узел** – полный путь к файлу узла (*.dbb);

После редактирования проекта ресурс должен быть создан заново.

-  **Изображение** – пиктограмма из библиотеки изображений веб-консоли;
-  **Удалить объект (Del)** – удалить выделенный объект;
-  **Редактировать объект (Enter)** – редактировать параметры

выделенного объекта;

- **Редактировать библиотеку изображений** – открыть редактор библиотеки изображений веб-консоли.

Для перегруппировки объектов дерева предусмотрены те же команды, что и для вкладки **Публикатор** (см. **Вкладка 'Публикатор'**).


Узлы, в том числе узлы **EmbeddedConsole**, должны быть сконфигурированы в ИС так же, как для сетевого обмена. Для узлов, от которых узел **EmbeddedConsole** получает данные, необходимо задать IP-адрес.

Вкладка 'Принтер'

Эта вкладка в табличном виде отображает очередь печати.

Для каждого документа отображается номер, имя, источник (имя компьютера, от которого документ поступил на печать; отображается только в течение загрузки документа) и состояние.

Меню **Правка**, панель инструментов и контекстное меню вкладки содержат следующие команды:

-  **Отменить (Del)** – отменить печать выделенного документа (документ удаляется из очереди).

Особенности выполнения узла в браузере

После авторизации, для запуска/останова выполнения java-апплета нужно нажать кнопку, которая добавляется на графическую панель при создании ресурса.

еГЭ 'Строка ОТ' и **еГЭ 'ОТ узла'** не поддерживаются.

При использовании **еГЭ 'Ввод значения'** значение задается в диалоге, который появляется на экране после двойного нажатия ЛК на еГЭ. Ввод значений даты/времени и строк не поддерживается.

еГЭ 'Тренд' имеет следующие отличия от **ГЭ 'Тренд'**:

- ориентация – только горизонтальная;
- не поддерживаются псевдонимы кривых; в легенде всегда отображаются имена каналов и номера атрибутов;
- не поддерживается передача значений левой и правой временных границ тренда в аргументы;
- на тренде отображается ось значений той кривой аналоговой панели, которая выделена в легенде, хотя заданные для каждой кривой диапазоны отрабатываются;
- масштабирование не поддерживается;

- в легенду одновременно выводятся кривые аналоговой и дискретной панелей;
- сохранение буфера в файл не поддерживается;
- не используется атрибут **Масштаб дискрет**;
- интерполирование – всегда по периоду реального времени;
- аргумент с типом данных с плавающей запятой и атрибутом **Интерпретировать как = Статус** отображается на дискретной панели;
- формат чисел на оси значений – всегда `%.1f`;
- стиль кривой в случае отсутствия узла, от которого принимаются данные, – пунктир. Не используются атрибуты **Стиль при I<>0 и W=0**, **Стиль при I=0 и W=1** и **Стиль при I<>0 и W=1**;
- в реальном времени можно изменить следующие параметры кривых:
 - атрибут **Интерпретировать как** (в диалоге – **Тип**); таким образом можно перенести аналоговую кривую на дискретную панель;
 - привязку;
- добавлять/удалять кривые в реальном времени нельзя;
- не поддерживается вызов экрана подачей «2» в канал вызова экрана.

Запуск узла в мобильном устройстве


Для запуска узла в мобильном устройстве (смартфоне и т.п.) нужно выполнить следующие действия:

- установить **TRACE MODE 6 Mobile Console** (Java-приложение);
- скопировать файл узла **EmbeddedConsole (*.dbb)**;
- открыть и запустить узел в приложении.

Процедуры установки Java-приложений и загрузки файлов описаны в документации устройства.

Сервер печати TRACE MODE 6

При запуске в сети, сервер печати TRACE MODE 6 (**tmacs_1.exe**) обеспечивает вывод генерируемых документов на принтер, используемый по умолчанию (см. **Использование разработанных шаблонов**).

После запуска сервера его иконка  отображается в области уведомлений панели задач. Иконка снабжена контекстным меню, содержащим следующие команды:

- **О программе** – показать информацию о сервере;
- **Выход** – завершить работу сервера.

Глава 5

**Обмен с
контроллерами и
УСО**

Средства поддержки оборудования

Для обмена с различными устройствами (см. **Обеспечение работы распределенных АСУ**) в TRACE MODE используются следующие механизмы:

- обмен по протоколам, встроенным в мониторы. К таким протоколам относятся:
 - DCS – протоколы контроллеров/серий модулей LAGOON, ROBO, NuDAM-6000, I-7000, ADAM-4000, ADAM-5000/485, RIO-2000 и т.п. при связи по последовательному интерфейсу;
 - MODBUS RTU при связи по последовательному интерфейсу;
 - MODBUS TCP/IP при связи по сети;
 - протоколы плат ввода/вывода (УСО), устанавливаемых в системные шины компьютеров/контроллеров;
 - IEC 60870-101 (по RS и сети);
 - другие протоколы, описанные в разделе **Обмен по встроенным протоколам по сети**;
- обмен через поставляемые драйверы;
- обмен через драйверы, разработанные и подключенные пользователем по правилам, описанным в разделе **Разработка драйверов. Интерфейс TCOM**.

Обмен через драйверы реализуется при связи как по стандартным интерфейсам (в том числе полевым шинам), так и при использовании дополнительных устройств, реализующих необходимые интерфейсы, – коммуникационных плат, преобразователей интерфейсов и т.п.

Для всех видов обмена в ИС предусмотрены соответствующие шаблоны каналов (см. **Шаблоны каналов обмена**).

Шаблоны каналов обмена

Для конфигурирования обмена с поддерживаемым оборудованием в TRACE MODE предусмотрены шаблоны числовых каналов, которые различаются **подтипом** и **дополнением к подтипу** (см. **Подтипы каналов** и **Атрибуты каналов, отображаемые профайлером**).

В большинстве случаев шаблоны каналов обмена с платами/модулями создаются в навигаторе автоматически (см. **Назначение групп источников (приемников)**), при этом для них задается ряд параметров. Эти параметры могут быть изменены в соответствующих редакторах. Пример-

ный вид одного из таких редакторов показан на рисунке (см. также **Редакторы источников (приемников)**):

Основные

Имя AI_Kruiz#1

Кодировка TW0

Комментарий

Справка

Параметры

Базовый адрес 0x100

Канал 0

Мультиплексор 1

GAIN/REG 0

ALT/REG 0

Тип сигнала ...

Направление Input

Формат Аналог

Дополнительно

Параметры аппаратных тегов задаются в соответствии с документацией на устройство, образуя **удаленный адрес** (см. **Удаленный адрес и разновидности драйверов** и **Атрибуты каналов, отображаемые профайлером**).

Для шаблона канала обмена через драйвер (см. **Обмен через драйверы**): строка, заданная в поле **Дополнительно**, записывается в атрибут (47, **iDstr**) канала обмена.

Групповое редактирование параметров шаблонов каналов возможно в базовом редакторе (см. **Базовый редактор группы компонентов**).

IP-адрес может задаваться в редакторе в нескольких форматах (см. **Формат IP-адреса**).

Если в канале обмена по RS номер COM-порта задан равным 255, MPB использует для обмена первый COM-порт, который удалось инициализировать.

Обмен по встроенным протоколам

Обмен по DCS

Для связи с модулями распределенного УСО (см. **Группа 'Распределенные УСО'**) в ИС предусмотрены специальные шаблоны каналов (см. **Подтип 8**).

Пользовательские SMS

Если GSM-модем подключен к COM-порту, назначение COM-порта должно быть **GSM-SMS** или **G_SMS** (см. **Редактор параметров COM-порта**).

Формат запроса значения атрибута канала:

```
?{ (<имя канала>), (#<номер канала>)} [.<номер атрибута>] [=]
```

Чтобы канал был в запросе, знак '=' обязателен.

Формат ответа, если канал есть в запросе:

```
<имя канала>.<номер атрибута>=<значение>
```

Формат ответа, если канала нет в запросе:

```
=<значение>
```

Ответ в случае ошибки:

```
!NAC
```

Формат задания значения атрибута канала:

```
{ (<имя канала>), (#<номер канала>)} [.<номер атрибута>]=<значение>
```

Формат группового задания атрибутов канала CALL, начиная с указанного атрибута (атрибут 1000 – это аргумент arg0):

```
{ (<имя канала>), (#<номер канала>)} [.<номер начального атрибута>]=<значение>, =<значение>, ..
```

Формат группового задания значений каналов (R – разделитель):

```
<имя канала>=<значение>R<имя канала>=<значение>  
R<имя канала>=<значение> ..
```

Формат группового задания аргументов канала CALL, начиная с arg0 (R – разделитель):

<имя канала>=<значение>R=<значение>R=<значение>

..

Ответ в случае ошибки:

!NAC

Ответ в случае успешного задания значений:

!ACK

В некоторых случаях ответы !NAC и !ACK не содержат восклицательного знака.

SMS-сообщение не может начинаться с символов «0n0», где n – 0,...,9,r,w,R,W,e,E,t,T,l,L, а также с символа @.

В файле *.cnf используются следующие ключи:

- **SMS_DLS**=<число> (@RTM_Parameter с Параметр=179) – период (в секундах) проверки прихода SMS-сообщения;
- **SMS_TOR**=<число> (@RTM_Parameter с Параметр=180) – максимальное время ожидания ответа при обмене по GSM. Задается в секундах, значение по умолчанию – 60с. При превышении этого таймаута каналу, инициализировавшему передачу SMS, устанавливается признак аппаратной недостоверности;
- **SMS_TCR**=<число> (@RTM_Parameter с Параметр=181) – период проверки наличия ответа после посылки SMS. Задается в секундах, значение по умолчанию – 5с;
- **SMS_RCN**=<> (@RTM_Parameter с Параметр=182) – зарезервировано;
- **SMS_FIND_DELIMITER**=ON (по умолчанию)/OFF – разрешение запрет поиска разделителя (ищется /, \, ;);
- **SMS_DELIMITER_IS**=<один символ> – задание символа разделителя;
- **SMS_ALL_INCOMMING**=ON (по умолчанию)/OFF – включение/отключение обработки всех SMS;
- **SMS_WRITE_INCOMMING**=ON (по умолчанию)/OFF – включение/отключение обработки SMS запроса данных;
- **SMS_READ_INCOMMING**=ON (по умолчанию)/OFF – включение/отключение обработки SMS записи данных;
- **SMS_ACK**=ON (по умолчанию)/OFF – включение/отключение ответа на запись данных и сообщения «!NAC» при ошибке;
- **SMS_ALARME_PHONE_LIST**=<phone>;..<phone>; – список телефонов для рассылки отчета тревог (в этом случае не участвуют пользователи – см. Канал класса ПОЛЬЗОВАТЕЛЬ);
- **SMS_CHECK_PHONE_LIST**=<phone>; <phone>; – список телефонов, с которых разрешен прием SMS (наличие этих телефонов

у пользователей не проверяется);

- **SMS_CHECK_PHONE_LIST_ADD_USER=ON** (по умолчанию) / **OFF** – включение/отключение проверки телефона из пришедшего SMS-сообщения (проверяется, что у пользователя есть такой номер и права на запись). Ключ используется совместно с ключом **SMS_CHECK_PHONE_LIST**;

Обмен по MODBUS

Для конфигурирования шаблонов каналов обмена по некоторым модификациям протокола MODBUS в ИС встроен редактор (см. **Шаблоны каналов обмена, Группа 'MODBUS-группа', Группа 'Распределенные УСО', Подтипы 9 и 100 и Обмен по встроенным протоколам по сети**).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), в редакторе задаются следующие атрибуты:

- **Номер порта** – номер последовательного интерфейса (0 – COM1, ..., 31 – COM32);
- **Адрес** – адрес контроллера в формате HEX (0, ..., FF);
- **Канал** – номер переменной в формате HEX;
- **Тип протокола** – модификация протокола:
 - **ModBus** – стандартный протокол;
 - **Эмикон** – это значение следует установить при конфигурировании обмена с контроллерами ЭМИКОН через драйвер (см. **Обмен через плату C05, шина ISA, Обмен через плату C06, шина PCI, Обмен через плату C06PnP, шина PCI**);
 - **ModBus (No group write)** – стандартный протокол без поддержки функции групповой записи для дискретных сигналов;
 - **TCP** – протокол MODBUS TCP/IP.
- **IP-адрес** – дополнительные параметры обмена (см. **Обмен по встроенным протоколам по сети**).

При обмене по MODBUS по последовательному интерфейсу в узле должен быть создан и настроен компонент COM-порт (см. **Редактор параметров COM-порта**).

Ключи конфигурирования обмена по RS задаются в файле *.cnf (перед знаком равенства не должно быть пробела):

- **MDB_FRMT_ALL=<число HEX>** – то же, что и ключ **MDB_FLOATCNV=<число DEC>**;
 - биты 0 и 1 – если установлен бит 0, используется обычный

- формат FLOAT, если установлен бит 1 – используется перестановка (только MODBUS RTU);
- биты 10 и 11 – то же, что и **FRMT=TIME_1** и **FRMT=TIME_2** (только MODBUS RTU, см. **Обмен по встроенным протоколам по сети**);
 - бит 14 – то же, что **FRMT=REGSIZE=4**;
 - бит 15 – то же, что **FRMT=SWAP_LONG**;
- следующие ключи аналогичны соответствующим ключам **FRMT=** (см. **Обмен по встроенным протоколам по сети**), по умолчанию используется **перестановка_1**:
 - **MDB_FRMT_FLOAT**=<номер порта, DEC> – аналог **FRMT=SWAP_FLOAT**;
 - **MDB_FRMT_REGSIZE**=<номер порта, DEC>, 4 – аналог **FRMT=REGSIZE=4**;
 - **MDB_FRMT_WORD**=<номер порта, DEC> – аналог **FRMT=SWAP_LONG**;
 - **MDB_FRMT_TIME**=<номер порта, DEC> – аналог **FRMT=SWAP_TIME**;
 - **MDB_FRMT_LEN_TIME**=<номер порта, DEC>, <число DEC, 0,1,2> – формат времени TIME_1/TIME_2;
 - **MDB_FRMT_SWAP**=<номер порта, DEC> – аналог **FRMT=SWAP_DOUBLE**.

Для формирования команд MODBUS может также использоваться канал CALL.8 (см. **Канал CALL.ChGroupReq**).

Если в строке **IP-адрес** источника-приемника MODBUS, привязанного к каналу CALL.ChGroupReq, задан ключ **«BYARG;»**, то количество запрашиваемых байт (N) равно сумме длин всех аргументов в байтах, исключая аргументы BOOL и STRING. Число запрашиваемых регистров – это N, округленное до ближайшего большего четного числа и деленное пополам.

Прибор 'Взлет МР'

Для считывания данных архивов контроллера УРСВ **«ВЗЛЕТ МР»** в ИС предусмотрены шаблоны каналов **Vzlet1(2,3)** (см. **Подтипы 9 и 100**). Эти шаблоны конфигурируются в редакторе переменной MODBUS (см. **Обмен по MODBUS**).

Вид – 1, 2 или 3 – канала выбирается в соответствии с типом архива (соответственно часовой, суточный и месячный).

Канал типа **OUTPUT** с настройкой **Канал = 0** своим значением указывает номер строки в архиве, а значения настроек **Канал** каналов типа **INPUT** – номер соответствующего параметра:

1 – объем в прямом направлении для первого канала;

- 2 – объем в обратном направлении для первого канала;
- 3 – объем в прямом направлении для второго канала;
- 4 – объем в обратном направлении для второго канала.

Величины объема часового, суточного и месячного архивов прибора определяют диапазон допустимых значений индекса записи для каналов **OUTPUT** с настройкой **Канал=0**:

- допустимый диапазон значений канала **Vzlet1**, соответствующий диапазону индексов записей часового архива: 0...671;
- допустимый диапазон значений канала **Vzlet2**, соответствующий диапазону индексов записей суточного архива: 0...63;
- допустимый диапазон значений канала **Vzlet3**, соответствующий диапазону индексов записей месячного архива: 0...23.

При использовании каналов **Vzlet1(2,3)** в проекте должен быть задан **СПАД**-архив.

При работе с данными каналами необходимо учитывать особенности:

- заполнение архива в приборе осуществляется циклически, по сетке фиксированных временных меток. При этом расположение записей в архиве может быть непоследовательным;
- при отсутствии записей по запрошенному индексу прибор возвращает данные с нулевой меткой времени, которая в **СПАД**-архиве интерпретируется как **01.01.1970 03:00:49**.

Контроллеры ROC 827 и FloBoss 107

Для обмена по MODBUS RTU, MODBUS TCP/IP контроллеры ROC 827 и FloBoss 107 предварительно конфигурируются с помощью утилиты RO-CLINK 800 в соответствии с документацией на устройства.

В качестве шаблонов каналов обмена с контроллерами используются следующие шаблоны группы **Источники/Приемники/MODBUS: W_SingleCoil(5), Rin_Byte(2), Rin_Float(4), W_Float(16)** и **R_FIFO_Queue** (см. Подтипы 9 и 100). **R_FIFO_Queue**, привязанный к каналу **CALL.ChGroupReq**, обеспечивает чтение архива контроллера.

В ИС указанные шаблоны конфигурируются в редакторе (см. **Обмен по MODBUS**). Значение атрибута **Канал** (номер переменной в формате HEX) должно соответствовать номеру регистра, указанному в столбце **Start Register** таблицы регистров контроллера (для групповых запросов указывается номер регистра первой переменной).

Для обмена по MODBUS RTU в узле необходимо создать и настроить компонент **COM-порт** (в соответствии с настройками используемого COM-порта контроллера).

Для обмена по MODBUS TCP/IP для узла необходимо сконфигурировать

файл **tcp_modBus** (см. **Обмен по встроенным протоколам по сети**).

*Особенности запроса переменных **FLOAT***

В контроллерах EMERSON под переменную **FLOAT** выделяется 1 регистр, поэтому для MODBUS-запроса одной переменной **FLOAT** в качестве количества запрашиваемых слов необходимо указывать не 2 слова (что соответствует стандартному протоколу MODBUS), а одно. Для этого в файле *.cnf узла должен быть задан ключ **FRMT=REGSIZE=4**.

Порядок следования байтов

При конфигурировании контроллеров с помощью утилиты ROCLINK 800 необходимо обратить внимание на настройку порядка следования байтов, который устанавливается отдельно для каждого порта контроллера. Соответствующий порядок следования байтов должен быть задан в файле *.cnf узла TRACE MODE (см. **Обмен по встроенным протоколам по сети**).

*Чтение архива контроллера **FloBoss 107***

Для запроса архива используется MODBUS-функция с номером 3. В данных ответа содержатся времена, даты и значения архивируемых переменных. Все переменные ответа имеют формат **FLOAT**.

Для запроса архива переменная **R_FIFO_Queue** привязывается к каналу **CALL.ChGroupReq** типа **INPUT** с атрибутом **Параметр=2**.

Канал посылает в контроллер команду MODBUS, сформированную по следующим правилам:

- адрес соответствует адресу, заданному в источнике;
- **arg0** задает номер функции (3);
- номер переменной соответствует заданному в источнике (2-байтовое целое);
- **arg1** задает номер запрашиваемой строки архива.

Дата и время строки архива записывается в **arg2**.

Значения параметров, заданных при конфигурировании FloBoss 107 на вкладке **History Setup** утилиты ROCLINK 800, записываются в последующие аргументы.

Обмен по встроенным протоколам по сети

Используемые шаблоны каналов:

- обмен по **TCP**:
 - MODBUS TCP/IP – **Источники/Приемники / MODBUS / <шаблон-команда Modbus (см. Подтипы 9 и 100)>**.
Протокол MODBUS TCP/IP также используется, если указанный шаблон привязан к каналу CALL.Vector. В строке **IP-адрес** шаблона в этом случае должен быть задан ключ '**TCP=<IP-адрес>**' (см. **Канал CALL.Vector**);
 - MITSUBISHI – **Источники/Приемники / PLC / Mitsubishi_Group / Mitsubishi**;
 - IEC 60870-104 – см. **Обмен по IEC 60870-104**;
 - TCP USER – шаблон **Источники/Приемники / MODBUS / <шаблон-команда Modbus (см. Подтипы 9 и 100)>**, привязанный к каналу CALL.Vector. В шаблоне должен быть задан ключ '**IP;**' (см. **Канал CALL.Vector**);
- обмен по **UDP**:
 - OMRON – **Источники/Приемники / PLC / Omron_IP_Group / Omron_IP**;
 - SNMP – **Источники/Приемники / Пользовательские драйверы / TYPE12 / UniversalPLC_type4** (в строке IP-адрес нужно обязательно указать «**SNMP:**»). В реальном времени каналу, к которому привязан источник/приемник, присваивается подтип 108 с дополнением 10;
 - UDP USER – шаблон **Источники/Приемники / MODBUS / <шаблон-команда Modbus (см. Подтипы 9 и 100)>**, привязанный к каналу CALL.Vector. В шаблоне должен быть задан ключ '**UDP;**' (см. **Канал CALL.Vector**).

Для конфигурирования обмена по встроенным протоколам в специальных файлах или в строке **Дополнительно/IP-адрес** источника/приемника задаются ключи. Конфигурационные файлы в некоторых случаях создаются автоматически при экспорте узла и во всех случаях могут быть созданы вручную. Конфигурационные файлы не имеют расширения и имеют текстовый формат.

Имена конфигурационных файлов:

- MODBUS TCP/IP – **tcp_modbus**;
- OMRON – **udp_omron**;
- MITSUBISHI – **tcp_mitsubishi**;
- SNMP – **udp_snmp**.

Допускается групповой запрос при подключении источника к каналу **ChGroupReq**.

Ключи:

- **UNIT**=<адрес устройства>

- **PORT**=<номер порта>
- **TIMEOUT**=<число>:
 - **число**=<индекс таймаута> – для MODBUS TCP/IP – индекс встроенного в RTM таймаута MODBUS (в протоколе определено несколько таймаутов);
 - **число**=<число секунд> – для всех остальных протоколов;
- **CSC_ON** или **CSC_OFF** (по умолчанию) – генерировать или нет контрольную сумму при посылке (ModBus, TCP USER);
- **IDNT_OFF** или **IDNT_ON** (по умолчанию) – не анализировать или анализировать ID пакетов (ModBus, TCP USER);
- **MDVRTU_ON** или **MDVRTU_OFF** (по умолчанию) – генерировать или нет контрольную сумму пакета для отправки и принимать или нет с контрольной суммой (считается, что контрольная сумма должна прийти) (ModBus, TCP USER);
- **AUTO_ON** – для всех встроенных протоколов по сети. Используется в следующих случаях:
 - 2 резерва, узел WORK работает, в узле TRACE обмен запрещен. Если ключ задан, для проверки связи узел TRACE периодически посылает команду **ping** по 1-му и 2-му IP-адресу;
 - 2 контроллера-резерва, обмен с одним из них. Если ключ задан, для проверки связи со вторым контроллером периодически посылается команда **ping**;
- **PING_ON** – до попытки соединения выполнить команду **ping**, и в случае неудачи не подключаться (если заданы 2 IP-адреса, то выбирается тот, на который команда **ping** проходит) (ModBus, Mitsubishi, TCP USER);
- **FAST_ON** – (ModBus, Mitsubishi, TCP USER);
 На ошибке отправки переключиться на соединение по второму адресу с незначительной задержкой.
 При ошибке соединения попробовать подключиться на второй адрес;
- **TH<IND>** (**IND** может принимать значение **14** или **15**) – работать с УСО в дополнительном потоке 14 или 15 (для всех встроенных сетевых протоколов).
 Транзакции для разных контроллеров могут быть распределены по 3 потокам:
 - основной поток (ключ **TH<IND>** не задается);
 - поток 14 (задается ключ **TH14**);
 - поток 15 (задается ключ **TH15**);
 - только для **UDP**: для основного потока может осуществляться процедура вынесения транзакций контроллера, кото-

рый достаточно долго не отвечает на обращения, в отдельные потоки BAD (могут задаваться ключи **THBAD1**, **THBAD2** и **THBAD3**). В потоке BAD транзакции осуществляются редко и с низким приоритетом. При появлении ответа от контроллера его транзакции возвращаются в основной поток;

- **THONE** – режим «одно устройство в одном потоке»;
- **TCP=<IP-адрес>** – первый IP-адрес для TCP;
- **TCP2=<IP-адрес>** – второй IP-адрес для TCP;
- **IP=<IP-адрес>** – первый IP-адрес для UDP;
- **IPADDR=<IP-адрес>** – первый IP-адрес для UDP;
- **IP2=<IP-адрес>** – второй IP-адрес для UDP;
- **IPADDR2=<IP-адрес>** – второй IP-адрес для UDP;
- **CLOSE_BY_SEND_ERR_ON** –
- **FRMT=REGSIZE=4** – этот ключ надо задать в том случае, если FLOAT занимает 1 регистр (MODBUS TCP/IP). Данный ключ нельзя задать в строке **Дополнительно**;
- **FRMT=TIME_1** и **FRMT=TIME_2** – для тех же контроллеров, для которых **FRMT=REGSIZE=4**. Данные ключи задают формат времени. Данные ключи нельзя задать в строке **Дополнительно**.

Ключи перестановки байтов для MODBUS TCP/IP:

- для всех классов каналов:
 - **FRMT=DEFAULT** – стандартная перестановка байтов (**перестановка_1**);
 - **FRMT=INTEL** – перестановка байтов Intel (**перестановка_2**);
 - **FRMT=SWAP0** – перестановка_0;
 - **FRMT=SWAP3** – перестановка_3;
- для определенных классов каналов (из перечисленных ниже ключей в строке **Дополнительно** может быть задан только ключ **FRMT=SWAP_FLOAT**):
 - **FRMT=SWAP_FLOAT** – перестановка_2 для FLOAT;
 - **FRMT=SWAP_LONG** – перестановка_2 для HEX32;
 - **FRMT=SWAP_TIME** – перестановка_2 для TIME;
 - **FRMT=SWAP_DOUBLE** – перестановка_2 для DOUBLE FLOAT;
 - **FRMT=SWAP_FLOAT0**, **FRMT=SWAP_FLOAT1**, **FRMT=SWAP_FLOAT2** и **FRMT=SWAP_FLOAT3** – соответственно перестановка_0, перестановка_1, перестановка_2 и перестановка_3 для FLOAT;
 - **FRMT=SWAP_LONG0**, **FRMT=SWAP_LONG1**,

FRMT=SWAP_LONG2 и **FRMT=SWAP_LONG3** – соответственно **перестановка_0**, **перестановка_1**, **перестановка_2** и **перестановка_3** для HEX32;

- **FRMT=SWAP_TIME0**, **FRMT=SWAP_TIME1**, **FRMT=SWAP_TIME2** и **FRMT=SWAP_TIME3** – соответственно **перестановка_0**, **перестановка_1**, **перестановка_2** и **перестановка_3** для TIME;
- **FRMT=SWAP_DOUBLE0**, **FRMT=SWAP_DOUBLE1**, **FRMT=SWAP_DOUBLE2** и **FRMT=SWAP_DOUBLE3** – соответственно **перестановка_0**, **перестановка_1**, **перестановка_2** и **перестановка_3** для DOUBLE FLOAT.

Обмен с контроллерами OMRON по UDP

Чтобы использовать встроенный драйвер, надо в поле **IP-адрес** записать строку вида

```
UDP=<адрес модуля в сети OMRON>; <IP-адрес контроллера>:<порт>; [AREA=<hh>;] [LEN1;] [LEN2;] [LEN4;] [BCD;] [TMOUT=n;]
```

- адрес модуля в сети OMRON – 3 символа;
- **hh** – код области в HEX-формате без префикса '0x' (см. **Обмен с контроллерами OMRON по сети Ethernet по протоколу UDP**);
- ключи **LEN1**, **LEN2**, **LEN4** могут устанавливаться при наличии ключа **AREA**; они определяют длину поля данных – соответственно 1, 2 или 4 байта (по умолчанию – 2 байта);
- при наличии ключа **BCD** ответ контроллера интерпретируется как двоично-десятичное число (1 или 2 байта);
- значение **n** таймаута ожидания ответа (**TMOUT**) задается в секундах.

Обмен по SNMP

Обмен по SNMP не работает Windows CE.

Атрибут **Дополнительно** шаблона канала должен содержать следующие ключи:

```
SNMP:AGENT=<IP-адрес>; [COMMUNITY=<область>;]OID=<ID>;
```

Параметры строки:

- **область** – **PRIVATE** (для записи) или **PUBLIC**. Если подстрока **COMMUNITY=<область>** отсутствует, то значение параметра устанавливается автоматически в зависимости от команды:
 - **GET_NEXT** или **GET** – **PUBLIC**;
 - **SET** – **PRIVATE**;

- **ID** – ID объекта в дереве данных SNMP (дерево представляет собой последовательность вида **.число.число...**).

Кроме того, строка **Дополнительно** может содержать следующие параметры:

- **tmout=<n>**; – таймаут в секундах ожидания соединения;
- **retries=<m>**; – число попыток соединения, после которых в канале устанавливается признак недостоверности.

Атрибут **C0** задает команду:

- 0 – **GET_NEXT** (устанавливается автоматически, если шаблон привязан к каналу **CALL.ChGroupReq**);
- 1 – **GET** (устанавливается автоматически, если шаблон привязан к каналу типа INPUT);
- 2 – **SET** (устанавливается автоматически, если шаблон привязан к каналу типа OUTPUT);

Атрибут **C5** содержит код ошибки:

- 1 – у агента не хватает места;
- 2 – несуществующее имя;
- 3 – некорректное значение;
- 4 – обнаружена попытка записи, когда запись запрещена;
- 5 – общая ошибка;
- 16 – ошибка обмена;
- 17 – не открыт сеанс.

По команде **GET** считывается одна переменная и записывается в атрибут (2,In).

По команде **SET** выполняются следующие действия:

- считывается текущее значение переменной;
- значение атрибута (9,Q) записывается в переменную.

По команде **GET_NEXT** в аргументы канала **CALL.ChGroupReq** записывается структура данных SNMP, начиная с заданного объекта.

Тип данных канала/аргумента должен соответствовать типу данных SNMP. Специфические типы данных SNMP:

- **counter64** – 64-разрядное число, записывается либо в канал **DOUBLE FLOAT**, либо в 2 аргумента канала **CALL.ChGroupReq** (в аргумент с младшим номером пишутся байты 4-7, в следующий аргумент – байты 0-3);
- IP-адрес, **counter32** – тип данных канала/аргумента должен быть **DINT (long)**;
- **octet string** – записывается в аргументы канала **CALL.ChGroupReq**. Если в считываемой строке есть непечатаемый

символ, все байты преобразуются в формат HEX и записываются через запятую. Если тип аргумента – STRING или DINT, аргумент меняет тип на длинную строку (ср. **Запись длинных строк в канал CALL**);

- **bit** – записывается в атрибут/аргумент с типом данных **DINT (long)** если длина строки меньше или равна 4 байтам;
- **OID** – записывается в аргумент с типом данных **STRING**;
- **sequence** – не поддерживается;
- **OPAQUE** – записывается так же, как **bit**.

Если в канале установлен флаг **Отладка** (49, **DBG**), в протокол профайлера (файл **<имя файла prj>_<порядковый номер узла>.txt**) записывается следующая строка:

<OID>=<считанное значение>

В драйвере может быть создано до 32 потоков. Номер потока (число от 0 до 31) задается в атрибуте (94, **C1**). Можно объединять в один поток запросы к устройствам с разными IP-адресами. Можно также разделить на несколько потоков трафик к одному IP-адресу.

Обмен по Modbus TCP/IP

Данный обмен настраивается типовым для ряда встроенных протоколов способом (см. **Обмен по встроенным протоколам по сети**).

Обмен по SNMP

Данный обмен настраивается типовым для ряда встроенных протоколов способом (см. **Обмен по встроенным протоколам по сети**).

Обмен с устройствами Mitsubishi

Данный обмен настраивается типовым для ряда встроенных протоколов способом (см. **Обмен по встроенным протоколам по сети**).

Обмен с устройствами OMRON

Данный обмен настраивается типовым для ряда встроенных протоколов способом (см. **Обмен по встроенным протоколам по сети**).

Обмен по IEC 60870-101

В качестве шаблонов каналов для обмена с устройствами по стандарту IEC 60870-101 используются шаблоны **UniversalPLC_type3** (группа **Источники/Приемники.Пользовательские_драйверы.TYPE12**).

Для чтения создаются: 1 служебный канал, 1 канал запроса данных и необходимое количество каналов INPUT приема данных.

Для записи создаются: 1 служебный канал и необходимое количество каналов OUTPUT записи данных.

Конфигурация служебного канала:

- **Параметр1** – номер RS;

Если в узле существует указанный COM-порт с назначением **IEC 60870-101**, инициализируется обмен по RS, в противном случае – по TCP/IP (подтип и дополнение к подтипу канала – соответственно 109.6 и 108.6).

- **Параметр2** – номер устройства. При записи в прибор с 2-байтовым номером второй байт передается нулевым;
- **Параметр3** = **Параметр4** = 0;
- **Параметр5** – таймаут (в секундах) послышки команды **reset_link** (значение таймаута записывается в атрибут 92, **I2** канала);

При старте обмена выполняется цепочка действий **link_status** – ответ – **reset_link** – **link_status** – ответ, и с заданным таймаутом сравнивается временной интервал между ответами прибора.

- **Дополнительно** – для обмена по RS:

IEC;

Для обмена по TCP/IP (если задан только IP-адрес, используется порт 2404):

IEC; IP=<IP-адрес>:<порт>;

Строка **Дополнительно** может также использоваться для задания следующих параметров:

IEC; ASDU=<1 или 2>; COT=<1 или 2>; IOA=<1, 2 или 3>; IP=<IP-адрес>:<порт>; LAD=<1 или 2>; TCA=<0, 1, 2 или 3>;

где

- **ASDU** – размер (число байт) общего адреса ASDU (значение по умолчанию – 1). При записи в прибор с **ASDU** = 2 второй байт передается нулевым;
- **COT** – размер (число байт) причины передачи (значение по умолчанию – 1). При записи в прибор с **COT** = 2 второй байт передается нулевым;
- **IOA** – размер (число байт) адреса объекта информации (значение по умолчанию – 2). При записи в прибор с **IOA** = 3 третий байт передается нулевым;

- **LAD** – размер (число байт) адреса контроллера (значение по умолчанию – 1);
- **TCA** – аналог задания значения битов 6 и 7 в канале OUTPUT (см. ниже).

Атрибут 0, **R** служебного канала может принимать следующие значения:

- 0 – обмена не было;
- 1 – команда **link_status** прошла, данных класса 1 нет;
- 2 – команда **link_status** прошла, данные класса 1 есть;
- 3 – формируется команда **reset_link**.

Конфигурация канала запроса данных:

- **Параметр1** и **Параметр2** – такие же, как в служебном канале;
- **Параметр3** – 0xF1 для запроса данных класса 1 или 0xF2 для запроса данных класса 2. Канал с **Параметр3=0xF2** запрашивает данные со своим периодом. Канал с **Параметр3=0xF1** со своим периодом проверяет сообщения устройства; если устройство сообщает, что данных нет, они не запрашиваются, в противном случае канал запрашивает данные быстро, игнорируя заданный период пересчета;
- **Дополнительно** – для обмена по RS и TCP/IP:
IEC;

В атрибут 0, **R** канала запроса данных записывается значение младшего полубайта управляющего слова устройства (например, 9 означает отсутствие данных).

Конфигурация канала приема (записи) данных:

- **Параметр1** и **Параметр2** – такие же, как в служебном канале;
- **Параметр3** – идентификатор типа данных (IDT) – >0;
- **Параметр 4** – в канале записи (OUTPUT) младший полубайт этого атрибута должен содержать причину передачи, а биты 6 и 7 должны задавать общий адрес ASDU.

Если значение младшего полубайта не задано (нулевое), МРВ подставляет следующие значения:

- 6 – для всех IDT, кроме 102;
- 5 – для IDT=102.

Задание общего адреса ASDU:

- биты 6 и 7 не установлены – общий адрес ASDU равен 0;
- установлен бит 6 (0x40) – общий адрес ASDU равен адресу устройства;
- установлен бит 7 (0x80) – общий адрес ASDU равен адресу, указанному в последней посылке устройства;
- установлены биты 6 и 7 (0xC0) – общий адрес ASDU равен

0xFF;

- **Параметр5** – адрес объекта информации (IOA);
- **Дополнительно** – для обмена по RS и TCP/IP:

IEC;

Строка **Дополнительно** может также использоваться для задания параметров IDT и IOA в десятичном формате (**Параметр3** и **Параметр5** имеют более низкий приоритет):

IEC; IDT=<число DEC>; IOA=<число DEC>;

Все ответы, полученные от устройства, распределяются по трем спискам:

- список 1 – ответы приоритета A1; список 1 обрабатывается 1 раз за цикл пересчета;

Все IDT имеют буквенные обозначения, два последних символа которых определяют приоритет: A1 – высший приоритет, далее A2 и т.д.

- список 2 – ответы на запросы чтения (кроме ответов приоритета A1); список 2 обрабатывается 1 раз за секунду или реже в потоке ACTION;
- список 3 – ответы на запросы записи (кроме ответов приоритета A1); список 3 обрабатывается 1 раз за секунду или реже в потоке ACTION.

Чтение

При чтении поддерживаются стандартные идентификаторы типов данных без времени, а также со временем Sp24time2a и Sp56time2a (типы 1-40).

Значения, полученные с помощью канала запроса данных, записываются в атрибуты 2, **In** каналов приема данных, а если запрошены и метки времени, то они записываются в атрибуты 45, **T** (без миллисекунд, миллисекунды записываются в атрибуты 88, **ms**).

В профайлере: если в канале запроса данных установлен флаг **Отладка** (49, **DBG**), то при записи данных в каналы приема в протокол профайлера (файл <имя файла prj>_<порядковый номер узла>.txt) записываются следующие строки:

```
ADDR=<адрес устройства> COT=<причина передачи>
IOA=<адрес объекта информации> VAL=<полученное значение>
QV=<байт описателя> [<полученная метка времени>] IDT=<идентификатор типа данных>
```

Поиск каналов приема: MPV последовательно анализирует все каналы узла и ищет канал, в котором адрес устройства, IDT и IOA совпадают с аналогичными параметрами в ответе устройства. При поиске следующие IDT считаются идентичными:

- 1, 2, 30
- 3, 4, 31
- 5, 6, 32
- 7, 8, 33
- 9, 10, 34
- 11, 12, 35
- 13, 14, 36
- 15, 16, 37

Шаблон канала запроса может быть привязан к каналу CALL.ChGroupReq:

- для незарезервированных IDT: если IOA в ответе устройства больше, чем IOA в канале, но меньше, чем IOA в канале плюс число аргументов канала, то полученное значение записывается в соответствующий аргумент (номер аргумента равен разности принятого IOA и IOA в канале). Если к такому аргументу привязан канал (**ch**), и принимается значение с меткой времени, то значение записывается в **ch.In**, а метка времени – в **ch.T**;
- для зарезервированных IDT: IOA в канале должно быть равно принятому IOA. В зависимости от типа данных аргумента, из полученных данных в него записывается:
 - SINT, USINT – 1 байт;
 - INT, UINT – 2 байта;
 - DINT, UDINT – 4 байта (long);
 - REAL, LREAL – 4 байта (float);
 - TIME, DATE, DATE_AND_TIME – 4 байта (дата и время).

Шаблон канала запроса может быть привязан к каналу CALL.TVC, в котором **Параметр**=0 и IOA совпадает с принятым IOA. В этом случае принятые значения и их метки времени последовательно записываются соответственно в **arg_{2k}** и **arg_{2k+1}** (**k**=1,2...). Если метки времени не запрашивались, в **arg_{2k+1}** записывается текущее время. Аргументы **arg₀** и **arg₁** вычисляются в канале автоматически.

Запись

При установке какого-либо параметра (передается значение (9,**Q**)) поддерживаются следующие идентификаторы типов данных (время передается в формате Cp56time2a):

- 113, 47, 46, 45 – 1 байт;
- 111, 110, 49, 48 – 2 байта;
- 112, 50 – FLOAT (4 байта);
- 51 – LONG (4 байта);
- 58, 59, 60 – 1 байт + время;

- 61, 62 – 2 байта + время;
- 63 – FLOAT + время;
- 64 – LONG + время.

Общее управление:

- 100 – управляющая команда опроса, **QOI**=(9,**Q**) (адрес объекта информации должен быть 0);
- 101 – команда опроса счетчиков, **QCC**=(9,**Q**) (адрес объекта информации должен быть 0);
- 102 – команда чтения (адрес объекта информации используется, передача данных одной переменной);
- 103 – синхронизация времени устройства (время посылается в формате Cr5btime2a);
- 104 – тестовая команда (посылается 0x55AA, устройство сообщает, что обмен прошел успешно и возвращает 0x55AA, либо сообщает, что переменная не существует (ошибочный IOA));
- 105 – установка процесса в исходное состояние, **QRP**=(9,**Q**) (адрес объекта информации должен быть 0);
- 106 – задержка опроса, 2 байта (посылается (9, **Q**) как число миллисекунд).

Чтобы разрешить использование бита FCB при записи, в строке **Дополнительно** канала OUTPUT нужно указать «**FCBON**».

Управление и диагностика

Для управления и диагностики используются унифицированные механизмы, описанные в разделе **Управление и диагностика обмена**.

Обмен по IEC 60870-104

В качестве шаблонов каналов для обмена с устройствами по стандарту IEC 60870-104 используются шаблоны **UniversalPLC_type3** (группа **Источники/Приемники.Пользовательские_драйверы.TYPE12**).

Для обмена создаются: 1 служебный канал (в строке **Дополнительно** источника служебного канала не должно быть ключа IDT) и необходимое количество каналов INPUT приема данных и каналов OUTPUT записи данных.

Конфигурация служебного канала:

- **Параметр2** – номер устройства;
- **Параметр3** = **Параметр4** = 0;
- **Дополнительно** – задается в следующем виде (если задан только IP-адрес, используется порт 2404):

IEC104; IP=<IP-адрес>:<порт>;

Строка **Дополнительно** может также использоваться для задания следующих параметров:

```
IEC104; ASDU=<1 или 2>; COT=<1 или
2>; IOA=<1, 2 или 3>; IP=<IP-адрес>:<порт>;
UTC=ON; CADDR=<число DEC>;
```

где

- **ASDU** – размер (число байт) общего адреса ASDU (значение по умолчанию – 2);
- **COT** – размер (число байт) причины передачи (значение по умолчанию – 2);
- **IOA** – размер (число байт) адреса объекта информации (значение по умолчанию – 3);
- **UTC=ON** – прием временных меток UTC;
- **CADDR**=<число DEC> – общий адрес ASDU сервера (по умолчанию номер устройства).

Атрибут 0, **R** служебного канала может принимать следующие значения (DEC):

- 0 – обмена не было;
- 2 – обмен идет;
- 20 – послана команда общего опроса IDT=100 (20);
- 21 – послана команда общего опроса счетчиков IDT=101 (5);
- 64 – послана команда **testfr(act)**.

Каналы приема и записи данных конфигурируются так же, как для обмена по стандарту 60870-101 (см. **Обмен по IEC 60870-101**, допустимые команды указаны в стандарте 60870-104).

Если в строке **Дополнительно** источника не указан **IDT**=<число DEC>, для автоматического формирования IDT нужно указать «**IDT;**» или «**IDT=?;**».

Канал передачи (OUTPUT):

- HEX16/HEX32 ((84, **HD**)=0, (56, **nBits**)=1) – IDT=45;
- HEX16/HEX32 ((84, **HD**)=0, (56, **nBits**)=2) – IDT=46;
- HEX16/HEX32 ((84, **HD**)=0, (56, **nBits**)=16/32) – IDT=51;
- FLOAT – IDT=50.

Каналы приема (INPUT):

- FLOAT – прием IDT=13/36;
- HEX16/HEX32 ((84, **HD**)=0, (56, **nBits**)=1) – прием IDT=1/30;
- HEX16/HEX32 ((84, **HD**)=0, (56, **nBits**)=2) – прием IDT=3/31;
- HEX16/HEX32 ((84, **HD**)=1, (56, **nBits**)=16/32) – прием

IDT=15/37;

- HEX16-HEX32 ((84, **HD**)=0, (56, **nBits**)=16/32)) – прием IDT=7/33.

Канал HEX16 с числом бит 16 работает аналогично HEX32 при наличии ключа **IEC_ENABLE_HEX16=ON**.

Для служебного канала может быть задано начальное значение атрибута (2, **In**) (например, в ИС). При старте МРВ установленный бит младшего байта **In** обозначает запрет совершения действия кроме первого соединения, аналогичный бит старшего байта обозначает запрет совершения действия при каждом соединении:

- бит 7 – очистить списки посылки;
- бит 0 – послать команду общего опроса (IDT=100 (20));
- бит 1 – послать команду общего опроса счетчиков (IDT=101 (5)).

Если шаблон служебного канала привязан к каналу CALL.ChGroupReq INPUT, МРВ создает в канале 8 аргументов:

- **arg0** –
- **arg1** – при задании следующих значений этого аргумента формируются команды:
 - 20-36 – 20 – команда общего опроса (IDT=100 (20)), 21-36 – команда запроса группы (IDT=100 (21..36));
 - 1-5 – 5 – команда общего опроса счетчиков (IDT=101 (5)), 1-4 – запрос группы счетчиков командой (IDT=101 (1..4)).

МРВ-сервер (SLAVE) обрабатывает такие команды и посылает соответствующий стандарту ответ, содержащий данные. В текущей версии TRACE MODE клиент (HOST) принимает данные из ответа сервера, однако анализирует ответ сервера не полностью.

- **arg2** –
- **arg3** –
- **arg4** –
- **arg5** – разность между текущим временем и временем последнего обмена;
- **arg6** и **arg7** – ошибка WSA (сокета) и время этой ошибки.

Если создано 16 аргументов:

- **arg12...arg13** – число информационных фреймов, посланных и принятых узлом-инициатором обмена;
- **arg14...arg15** – число информационных фреймов, посланных и принятых отвечающим узлом.

В атрибут (97, **C4**) служебного канала можно записать следующие значения:

- 2 – переподключиться;
- 3 или 7 – остановить опрос;
- 16 – остановить опрос;
- 17 – переподключиться с изменением младшего бита IP-адреса;
- 18 – переподключиться с изменением младшего бита тетрады 3 IP-адреса;
- 19 – переподключиться с изменением младших битов тетрад 3 и 4 IP-адреса (или использовать другой IP-адрес, если он задан);
- 21 – послать **stopdt(act)**;
- 24 – послать **startdt(act)**;
- 32 – принудительно послать S-фрейм **testfr(act)** (число 22);
- 33 – послать команду общего опроса (IDT=100 (20)) (число 23).

Для задания необходимых таймаутов используются следующие ключи в файле *.cnf:

- **IEC_TESTFR**=<n, DEC> – посылка команды **testfr**, если не было принято данных в течение **n** секунд. Если результат выполнения команды отрицательный, сокет закрывается. По умолчанию, **IEC_TESTFR**=20с;
- **IEC_FAOUT**=<число секунд, DEC> – таймаут на подтверждение команды управления (по умолчанию 10с).

Другие ключи:

- **IEC_HOST_ACK_COT**=ANY – по умолчанию **host**, в ответ на команду с IDT=45, 46, 50, 51, ожидает две посылки от **slave** с тем же IDT и COT=7, затем – с COT=10. Для подтверждения команды любым COT (COT=7 или COT=10) однократной посылкой используется данный ключ.
- **IEC_KFROM**= – остановка посылки после не подтвержденных блоков;
- **TCP_DISCONN07**=<число секунд, DEC> – если спустя заданное время после **testfr(act)** не получена **testfr(con)**, сокет закрывается.

Ключи для тестирования:

- **IEC_HOST_SFRAME**=OFF/ON – запретить/разрешить S-фрейм;
- **IEC_HOST_FIRST_TESTFR** – после соединения послать TESTFR(act);
- **IEC_HOST_CLOSE_SOC** – закрыть сокет;
- **IEC_HOST_TESTFR(CON)**=ON/OFF – разрешение/запрет TESTFR(CON).

Команда синхронизации времени IDT=103

Для отсылки IDT=103 нужно создать источник для IEC-104 с IOA=0 и привязать его к каналу TIME OUTPUT. При изменении значения канала

посылается команда синхронизации времени.

Для контроля можно создать источник для IEC104 с IOA=0 и привязать его к каналу TIME INPUT. В такой канал запишется время, посланное серверу (по IDT=100 с COT=7 после проведения необходимых процедур).

Групповые запросы

Для запроса группы нужно создать OUTPUT/INPUT канал с IDT=100 и послать во вход такого канала 21-36.

Для операции с группой счетчиков нужно создать OUTPUT/INPUT канал с IDT=101 и послать во вход такого канала величину (**frozen** | 1..4), где **frozen** может принимать значения 0xC0 (сброс **R**), 0x80 (заморозка **R**), 0x40 (сброс и заморозка **R**) или 0x00 (запрос); 1..4 – номер группы счетчиков.

Если тип канала – OUTPUT, запрос выполняется однократно (**R**=0 после выполнения).

Если тип канала – INPUT, запрос выполняется многократно с периодом канала.

Для канала INPUT с IDT101 предусмотрено задание **R**=0x0n4n, 0x0n8n, где n=1..4 – номер группы счетчиков. Канал генерирует команду (заморозки/заморозки и сброса группы), после корректного ответа генерирует команду опроса группы счетчиков, и далее команда генерируется с периодом пересчета канала.

Чтение штрих-кодов и магнитных носителей

При подключении к последовательному порту устройства чтения штрих-кода или магнитного носителя в проекте нужно создать COM-порт с назначением **Card/Bar Reader** (см. **Редактор параметров COM-порта**).

В качестве шаблонов каналов чтения используются шаблоны **UniversalPLC_type4** (группа **Источники/Приемники / Пользовательские драйверы / TYPE11**).

Атрибут **Порт** (93, **C0**) должен быть равен номеру COM-порта с назначением **Card/Bar Reader**.

Для чтения магнитного носителя атрибут **Параметр4** (98, **C5**) должен быть равен 0, для чтения штрих-кода – отличен от 0.

Шаблон может быть привязан к числовому каналу, а также, если **Параметр4** = 0 или 1, – к каналу CALL.ChGroupReq или CALL.TVC. Корректно сконфигурированный канал имеет подтип и дополнение к подтипу 105.0.

Если **Параметр4** = 0 или 1 и шаблон привязан к числовому каналу, счи-

танный код записывается как строка в атрибут (47, **iDstr**) канала (до 60 байт). В случае успешного считывания, значение канала изменяется в зависимости от атрибута **Параметр1** (95, **C2**):

- если **Параметр1** = 0 – увеличивается на 1;
- если **Параметр1** = 1 – уменьшается на 1 (до 0);
- если **Параметр1** = 2 – обнуляется;
- если **Параметр1** = 3 – устанавливается в 1.

Если **Параметр4** = 0 или 1 и шаблон привязан к каналу CALL.ChGroupReq или CALL.TVC, поля считанного кода последовательно записываются в аргументы канала (текущее поле записывается в аргумент с наименьшим номером после предварительного сдвига ранее записанных полей на один аргумент). Тип данных аргументов приема должен быть STRING (см. также **Запись вектора в CALL.ChGroupReq и CALL.TVC**).

Если **Параметр4** = 0x16...0x20, после считывания кода производится поиск всех каналов подтипа 105 с **Параметр4** = 0x16...0x20, и при успешном считывании у таких каналов значение меняется в зависимости от атрибута **Параметр1** (см. выше). Данные считываются, если удовлетворяются соответствующие условия; для всех перечисленных ниже случаев длина считываемого кода должна быть больше, чем **Параметр2** (96, **C3**):

- **Параметр4** = 0x16 – если в любом месте, начиная с байта, заданного **C3**, в коде обнаружена кодировка канала;
- **Параметр4** = 0x17 – если кодировка канала располагается в коде с байта, заданного **C3**;
- **Параметр4** = 0x18 – если кодировка канала располагается в коде с байта, заданного **C3**, и после кодировки код ничего не содержит;
- **Параметр4** = 0x19 – если в любом месте кода обнаружена кодировка канала (только начиная с релиза 6.07);
- **Параметр4** = 0x20 – если кодировка канала располагается в коде с байта, заданного **C3**, и в любом месте кода обнаружен комментарий канала (только начиная с релиза 6.07).

В случае считывания штрих-кода:

- если **C5**=0x3 (дополнительно установлен бит 1) – если реальная длина считанного кода не равна значению первых двух байтов этого кода, код не записывается в канал;
- если **C5**=0x5 (дополнительно установлен бит 2) – запись кода в канал зависит от атрибута (97, **C4**):
 - **C4**=0 – код записывается в любом случае;
 - **C4**<>0 – код записывается только в том случае, если значение его первого байта равно **C4**;
- если **C5**=0x9 (дополнительно установлен бит 3) – запись кода в

канал зависит от атрибута (94, **C1**):

- **C1**=0 – код записывается в любом случае;
- **C1**=<ASCII-код символа> – код записывается только в том случае, если он начинается с заданного символа.

В **C5** не рекомендуется устанавливать биты 1 и 2 одновременно с битом 3.

В случае считывания магнитного носителя:

- если бит 0 (0x1) **C2** равен 0, считывается дорожка 1, в противном случае – дорожка 3.

Если установлен бит 1 (0x2) **C2**, дополнительно считывается дорожка 2 и записывается во вход канала. Дорожка 2 считывается до конца, если бит 4 (0x10) **C2** не установлен, или до пробела, если бит 4 (0x10) **C2** установлен.

Верификация дорожек по стандарту:

- дорожка 1 – начало – “%”, конец – “?”;
- дорожка 2 – начало – “;”, конец – “?”;
- дорожка 3 – начало – “;”, конец – “?”.
- если устройство отвечает, что кода нет, во вход канала пишется 0, а в атрибут (47, **iDstr**) – (**NULL**);
- если устройство сообщает об ошибке чтения, во вход канала пишется 1, в атрибут (47, **iDstr**) – (**ERROR**), и в канале устанавливается признак аппаратной недостоверности;
- если в канале установлен флаг **Отладка** (49, **DBG**), при успешном чтении кода в протокол профайлера (файл <имя файла **prj**>_<порядковый номер узла>.txt) записывается следующая строка:

TRACK=<считанный код>

Обмен по произвольному протоколу по RS

Обмен по произвольному протоколу типа t11 по RS конфигурируется следующим образом:

- в узле создается соответствующий COM-порт с назначением **MAS-TER**;
- в качестве шаблона канала обмена используется любой из шаблонов **UniversalPLC_type1...4** (группа **Источники/Приемники / Пользовательские драйверы / TYPE11**).

Конфигурация удаленного адреса шаблона:

- **C0** – номер COM-порта (обязательно);
- **C1** – номер устройства (<>0);

- **C2** и **C3** ($\langle \rangle 0$) – заголовок запроса;
- **C4** ($\langle \rangle 0$) – количество байтов в ответе.

Атрибут **Комментарий** шаблона должен содержать следующую строку:

UDEF

Шаблон привязывается к каналу **CALL.ChGroupReq**;

- атрибут **Параметр** канала **CALL.ChGroupReq** определяет количество аргументов, используемых для формирования запроса, эти аргументы не должны иметь привязки. Последующие аргументы канала предназначены для приема ответа.

В реальном времени каналу **CALL.ChGroupReq** присваивается подтип 109 с дополнением 27, 28, 29 или 30 в зависимости от типа шаблона (1, 2, 3 или 4).

В удаленном адресе канала **CALL.ChGroupReq**:

- 93, **C0** – номер COM-порта;
- ненулевые значения байтов **C2**, **C3**, **C1** (атрибуты 95, 96 и 94) передаются в запросе перед **arg0** в указанном порядке;
- 97, **C4** $\langle \rangle 0$ – количество байтов в ответе. Если **C4** = 0, количество байтов, ожидаемых в ответе, определяется по количеству приемных аргументов и их типу данных.

Формат запроса определяется типом данных соответствующих аргументов. Для аргумента с типом данных STRING количество передаваемых байтов определяется количеством символов в его значении (≤ 4).

Количество байтов, копируемых из ответа в приемный аргумент, соответствует типу данных этого аргумента. В приемный аргумент с типом данных STRING копируется 1 символ (1 байт).

В 1-байтовый приемный аргумент пишется 1 байт данных в 2-байтовом формате, при этом значение старшего байта MPB не контролирует.

Для декодирования меток времени используется ключ **TFRT<n>** – см. **Канал CALL.Vector**.

Для обмена по произвольному протоколу может быть также использован канал **CALL.Vector** (см. **Канал CALL.Vector**).

Обмен по протоколу HydraLink

В качестве шаблонов каналов для обмена с устройствами по протоколу HydraLink используются шаблоны **UniversalPLC_type2** (группа **Источники/Приемники.Пользовательские_драйверы.TYPE11**).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона задаются следующие атрибуты:

- **Порт** – номер COM-порта, уменьшенный на 1;
- **Адрес** – адрес прибора в сети;
- **Параметр1** – номер записи в архиве, уменьшенный на 1 (для **Параметр3** = 5; если **Параметр1** = 0, запрашивается последняя запись);
- **Параметр2** – номер теплосистемы;
- **Параметр3** – тип команды:
 - 1 – запрос значений параметров наработки;
 - 2 – запрос значений параметров наработки со временем;
 - 3 – запрос текущих значений параметров;
 - 4 – запрос текущих значений параметров со временем;
 - 5 – чтение архива;
 - 8 – чтение заголовка архива со значениями;
 - 10 – текстовая команда;
 - 16 – чтение времени (для каналов типа OUTPUT – коррекция времени);
 - 17 – чтение даты и времени.
 - **Дополнительно** – задается в следующем формате:
Hydralink: [PWD=xxxxxx; COMMAND]
PWD=xxxxxx – пароль, необходимый для коррекции времени;
COMMAND – текстовая команда.

При **Параметр3** = 1, 2, 3, 4, 5 или 8 шаблон должен быть привязан к каналу Call.ChGroupReq. Назначение аргументов Call.ChGroupReq:

- **Параметр3** = 1:
 - ARG000 (REAL) – нарастающий итог времени наработки;
 - ARG001 (REAL) – нарастающий итог подающего объема;
 - ARG002 (REAL) – нарастающий итог обратного объема;
 - ARG003 (REAL) – нарастающий итог объема подпитки;
 - ARG004 (REAL) – нарастающий итог обратной массы;
 - ARG005 (REAL) – нарастающий итог подающей массы;
 - ARG006 (REAL) – нарастающий итог массы подпитки;
 - ARG007 (REAL) – нарастающий итог тепловой энергии;
- **Параметр3** = 2:
 - ARG000 (DATE_AND_TIME) – время;
 - ARG001 (REAL) – нарастающий итог времени наработки;
 - ARG002 (REAL) – нарастающий итог подающего объема;

- ARG003 (REAL) – нарастающий итог обратного объёма;
- ARG004 (REAL) – нарастающий итог объёма подпитки;
- ARG005 (REAL) – нарастающий итог обратной массы;
- ARG006 (REAL) – нарастающий итог подающей массы;
- ARG007 (REAL) – нарастающий итог массы подпитки;
- ARG008 (REAL) – нарастающий итог тепловой энергии;
- **Параметр3 = 3:**
 - ARG000 (REAL) – расход подающий объёмный;
 - ARG001 (REAL) – расход обратный объёмный;
 - ARG002 (REAL) – расход подпитки объёмный;
 - ARG003 (REAL) – расход подающий массовый;
 - ARG004 (REAL) – расход обратный массовый;
 - ARG005 (REAL) – расход подпитки массовый;
 - ARG006 (REAL) – температура подающая;
 - ARG007 (REAL) – температура обратная;
 - ARG008 (REAL) – температура подпитки;
 - ARG009 (REAL) – температура окружающей среды;
 - ARG010 (REAL) – давление подающее;
 - ARG011 (REAL) – давление обратное;
 - ARG012 (REAL) – давление подпитки;
 - ARG013 (REAL) – тепловая мощность;
 - ARG014 (UDINT) – маска ошибок (см. расшифровку ниже);
- **Параметр3 = 4:**
 - ARG000 (DATE_AND_TIME) – время;
 - ARG001 (REAL) – расход подающий объёмный;
 - ARG002 (REAL) – расход обратный объёмный;
 - ARG003 (REAL) – расход подпитки объёмный;
 - ARG004 (REAL) – расход подающий массовый;
 - ARG005 (REAL) – расход обратный массовый;
 - ARG006 (REAL) – расход подпитки массовый;
 - ARG007 (REAL) – температура подающая;
 - ARG008 (REAL) – температура обратная;
 - ARG009 (REAL) – температура подпитки;
 - ARG010 (REAL) – температура окружающей среды;
 - ARG011 (REAL) – давление подающее;
 - ARG012 (REAL) – давление обратное;
 - ARG013 (REAL) – давление подпитки;
 - ARG014 (REAL) – тепловая мощность;

- ARG015 (UDINT) – маска ошибок (см. расшифровку ниже);
- **Параметр3 = 5:**
 - ARG000 (DATE_AND_TIME) – время выборки;
 - ARG001 (REAL) – время наработки;
 - ARG002 (REAL) – объем подающий;
 - ARG003 (REAL) – объем обратный;
 - ARG004 (REAL) – объем подпитки;
 - ARG005 (REAL) – масса подающая;
 - ARG006 (REAL) – масса обратная;
 - ARG007 (REAL) – масса подпитки;
 - ARG008 (REAL) – средняя температура подающая;
 - ARG009 (REAL) – средняя температура обратная;
 - ARG010 (REAL) – средняя температура подпитки;
 - ARG011 (REAL) – средняя температура окружающей среды;
 - ARG012 (REAL) – среднее давление подающее;
 - ARG013 (REAL) – среднее давление обратное;
 - ARG014 (REAL) – среднее давление подпитки;
 - ARG015 (REAL) – тепловая энергия;
 - ARG016 (UDINT) – маска ошибок (см. расшифровку ниже);
 - ARG017 (REAL) – время расхода меньше минимального;
 - ARG018 (REAL) – время расхода больше максимального;
 - ARG019 (REAL) – время, в течение которого delta_T меньше минимального значения;
 - ARG020 (REAL) – время отсутствия электропитания;
 - ARG021 (REAL) – общее учтённое время;
- **Параметр3 = 8:**
 - ARG000 (REAL) – флаг изменения содержимого;
 - ARG001 (REAL) – контрольная сумма заголовка;
 - ARG002 (REAL) – номер виртуального прибора;
 - ARG003 (REAL) – тип архива;
 - ARG004 (REAL) – набор в основе архива;
 - ARG005 (REAL) – частота занесения данных;
 - ARG006 (REAL) – состав архива;
 - ARG007 (REAL) – число записей в архиве;
 - ARG008 (REAL) – индекс следующей записи;
 - ARG009 (REAL) – предельное число записей в архиве;
 - ARG010 (DATE_AND_TIME) – дата и время обновления архива;

- ARG011 (REAL) – нарастающий итог подающего объёма;
- ARG012 (REAL) – нарастающий итог обратного объёма;
- ARG013 (REAL) – нарастающий итог объёма подпитки;
- ARG014 (REAL) – нарастающий итог обратной массы;
- ARG015 (REAL) – нарастающий итог подающей массы;
- ARG016 (REAL) – нарастающий итог массы подпитки;
- ARG017 (REAL) – нарастающий итог тепловой энергии.

ARG011-ARG017 выдаются на момент последней записи в архив.

Расшифровка маски ошибок:

- бит 0 – расход теплоносителя меньше минимально допустимого;
- бит 1 – расход теплоносителя больше максимально допустимого;
- бит 2 – обрыв или неисправность датчика температуры;
- бит 3 – температура теплоносителя меньше минимально допустимой;
- бит 4 – температура теплоносителя больше максимально допустимой;
- бит 5 – обрыв или неисправность датчика давления;
- бит 6 – давление теплоносителя меньше минимально допустимого;
- бит 7 – давление теплоносителя больше максимально допустимого;
- бит 24 – разность активных температур меньше минимально допустимой;
- бит 25 – математическая ошибка вычисления тепловой энергии;
- бит 26 – обрыв или неисправность датчика окружающей температуры;
- бит 27 – температура окружающей среды меньше минимально допустимой;
- бит 28 – температура окружающей среды больше максимально допустимой;
- бит 29 – несанкционированный перезапуск ПО прибора (включение/выключение электропитания не контролируется);
- бит 30 – ошибка вычисления тепловой энергии (вспомогательный параметр).

При **Параметр3** = 10 шаблон привязывается к любому каналу; запрос обрабатывается при отработке канала.

При **Параметр3** = 16 или 17 шаблон типа INPUT привязывается к каналу TIME.

При **Параметр3** = 16 шаблон типа OUTPUT привязывается к каналу FLOAT.

Универсальный механизм обмена с электросчетчиками

В данном разделе описан универсальный механизм обмена по RS с произвольным электросчетчиком из числа поддерживаемых данным механизмом (номенклатура таких электросчетчиков постоянно расширяется). При организации обмена со счетчиками с помощью данного механизма должны быть отменены запросы тех параметров, которые не поддерживаются в конкретной модели счетчика.

Ниже используются следующие обозначения:

- **I** – сила тока,
- **U** – напряжение;
- **<вектор_1>V<вектор_2>** – угол между векторами;
- φ – сдвиг фаз между **U** и **I**;
- **P**, **Q**, и **S** – соответственно мгновенная активная, реактивная и полная мощность (суммарная по всем фазам, нижний индекс **i** – импортируемая мощность, нижний индекс **e** – экспортируемая мощность);
- **<P>**, **<Q>** и **<S>** – те же мощности, усредненные в счетчике (интервал усреднения задается в счетчике);
- **dP** – мгновенная мощность потерь, суммарная по всем фазам;
- **P/S** – коэффициент мощности, суммарный по всем фазам;
- **EP**, **EQ** и **ES** – соответственно активная, реактивная и полная энергия (текущие значения, нижний индекс **i** – импортируемая энергия, нижний индекс **e** – экспортируемая энергия, верхний числовой индекс **T** – номер тарифа (**T=1...7**));
- **f** – частота сети;
- **THD** – коэффициент нелинейных искажений (КНИ);
- **HRMn** – коэффициент n-ой гармоники напряжения.

Нижние числовые индексы обозначают номера фаз.

Конфигурирование обмена

В узле должен быть создан соответствующий COM-порт с назначением **E_Meter** (см. Редактор параметров COM-порта).

В группе **Источники/Приемники / Пользовательские драйверы / TYPE11** создается любой из шаблонов **UniversalPLC_type1...4**. С помощью редактора шаблона задаются байты удаленного адреса:

- 93, **C0** (по умолчанию **Порт** в редакторе) – номер COM-порта (0 – COM1);

- 94, **C1** (по умолчанию **Адрес** в редакторе) – тип прибора (см. ниже перечень поддерживаемых приборов), может быть изменен в реальном времени;
- 97, **C4** – номер группы;
- 98, **C5** – номер фидера, если **C5** <> 0;
- 91, **I1** – адрес прибора (если адрес прибора не задан в атрибуте **Дополнительно**), может быть изменен в реальном времени.

Шаблон привязывается к каналу CALL.ChGroupReq (далее – **CGR_main**, подтип и дополнение к подтипу такого канала в MPB – 102.0).

К аргументам **CGR_main** (тип данных аргументов должен быть REAL) привязываются каналы CALL.ChGroupReq, CALL.TVC или CALL.AS_DATA (далее – **CGR**). Подтип **CGR** в MPB – 103.

В **CGR** в ИС должен быть создан хотя бы 1 аргумент.

Начальное значение **CGR**, заданное в ИС, определяет группу параметров, записываемых в его аргументы (значение, заданное в поле **На старте** редактора канала, в MPB хранится в атрибуте 86, **nAtrt**, и это значение может быть изменено в реальном времени). При необходимости MPB корректирует число аргументов **CGR**.

Каналы **CGR** чтения архивных данных с одним и тем же значением **CGR.86** нумеруются (инкрементируется атрибут 98, **C5**). Это используется, например, для устранения двойного управления атрибутом **CGR.59** (со стороны сумматора и со стороны ГЭ – см. ниже).

Атрибут **Дополнительно** предназначен для задания ряда служебных параметров в следующем формате:

```
<обозначение параметра 1>=<значение>; ... <обозначение параметра N>=<значение>;
```

Параметры:

- **ID** – идентификатор прибора;
- **PWD** – пароль;
- **ACCL** – уровень доступа (**ACCL**=256 – альтернативный доступ, без пароля);
- **SPD** – скорость обмена, бод (допустимо задание только значащих цифр, например, 96 вместо 9600);
- **CTRL** – контроль четности (например, 8,1,e);
- **SLP** – время бездействия (мс) после изменения скорости обмена;
- **TRFG** – число тарифов прибора;
- **PHASE** – число фаз прибора (1 или 3);
- **PCNT**=<n1>, <n2>, ... <n8> – номера регистров, в которых хра-

няться соответственно P_e , P_i , Q_e , Q_i , S_e , S_i , dP_e и dP_i ;

- **ECNT**=<канал0>,<канал1>... – описание каналов (две запятые подряд указывают, что соответствующий канал не сконфигурирован; разделителем может быть также точка или знак подчеркивания):
 - **P**, **P+** или **PE** – канал будет измерять активную мощность;
 - **P-** или **PI** – канал будет измерять обратную активную мощность;
 - **Q+** – канал будет измерять реактивную мощность;
 - **Q-** – канал будет измерять обратную реактивную мощность;
- **ECFG**=<буква (*)> – управление чтением значений энергии:
 - **A** – активная энергия;
 - **Q** – реактивная энергия;
 - **I** – обратная энергия;
 - **M** – в приборе есть архив месячных значений энергии нарастающим итогом;
 - **D** – в приборе есть архив дневных значений энергии нарастающим итогом;

После **M** и **D** может следовать 0 (ноль), означающий разрешение месячного/дневного архива, формируемого по значениям энергии нарастающим итогом.

 - **G** – месячные значения энергии представлены нарастающим итогом (**g** – с начала месяца);
 - **K** – дневные значения энергии представлены нарастающим итогом (**k** – с начала дня);
 - **T** – энергия нарастающим итогом;
 - **R** – энергия на старте;
- **PCFG**=<буква (*)> – управление чтением:
 - **A** – активная мощность;
 - **Q** – реактивная мощность;
 - **S** – полная мощность;
 - **L** – мощность потерь;
 - **I** – обратная мощность;
 - **D** – в приборе есть архив значений средней мощности;
 - **0** (ноль) – МРВ вычисляет среднюю мощность по считанным значениям мгновенной мощности;
 - **R** – средняя мощность на старте;
 - **G** – МРВ вычисляет суммарную мощность по считанным значениям мощности по фазам (суммарная мощность не считывается из прибора);

- **K** – МРВ вычисляет суммарную мощность и коэффициент мощности по считанным значениям активной и реактивной мощности;
- **H** – характеристики нелинейных искажений;
- **+** – инверсия знака значений $\langle P_e \rangle$ (используется в том случае, если считанные значения $\langle P_e \rangle$ отрицательны);
- **CRCT**=<буква (*)> – управление чтением служебных параметров:
 - **S** – константы из аргументов;
 - **H** – константы из прибора;
 - **N** – серийный номер прибора с записью его в атрибут (80,**CMNT**) канала подтипа 102;
 - **E** – чтение конфигурации прибора (анализируется **PCNT**, после чего предпринимается попытка чтения параметров из прибора);
 - **D** – разрешение переключения на резервный COM-порт (должен быть задан);
 - **V** – разрешение включения прибора в список;
- **TAVR**=<число минут> – период усреднения;
- **RGRP**=<число HEX без префикса «0x»> – биты числа разрешают (0) или запрещают (1) чтение соответствующих параметров:
 - бит 1 (0x2) – **U_N**, **I_N**, **U_NV_I_N**, **U_{NM}** и **U_NV_U_M** (**CGR.86**=0x11-0x16, см. ниже **Расшифровка значений CGR.86**);
 - бит 2 (0x4) – мгновенные значения мощности;
 - бит 7 (0x80) – все данные по нелинейным искажениям;
- **FILE**=<**N**> – указание файла, содержащего описание команд чтения счетчика. Файл должен размещаться в директории проекта и иметь имя **CMD_FILE<N>.txt** (**N**=1...9). В файле должно быть 24 строки, номер каждой строки определяет команду чтения (строка, начинающаяся со знака «;», интерпретируется как комментарий, но тоже нумеруется):
 - 0 – время;
 - 1 – дата;
 - 2 – «;» (зарезервировано);
 - 3 – серийный номер счетчика;
 - 4 – текущий статус счетчика;
 - 5 – **EP_e** от сброса;
 - 6 – **EP_i** от сброса;
 - 7 – **EQ_e** от сброса;
 - 8 – **EQ_i** от сброса;
 - 9 – **EP_e** за предыдущий месяц;
 - 10 – **EP_i** за предыдущий месяц;

- 11 – E_{Q_e} за предыдущий месяц;
- 12 – E_{Q_i} за предыдущий месяц;
- 13 – E_{P_e} за предыдущий день;
- 14 – E_{P_i} за предыдущий день;
- 15 – E_{Q_e} за предыдущий день;
- 16 – E_{Q_i} за предыдущий день;
- 17 – f ;
- 18 – U по фазам;
- 19 – I по фазам;
- 20 – P по фазам;
- 21 – Q по фазам;
- 22 – S по фазам;
- 23 – dP по фазам.

Параметр **FILE** может использоваться только для приборов, поддерживающих стандарт IEC 62056-21 (2002-05);

- **TEL:**<номер телефона> – указание модема для связи;
- **EMUL**=<число> – коэффициент, на который умножаются считанные из прибора значения **E**;
- **UMUL**=<число> – коэффициент, на который умножаются считанные из прибора значения **U**;
- **AMUL**=<число> – коэффициент, на который умножаются считанные из прибора значения **<P>**;
- **QMUL**=<число> – коэффициент, на который умножаются считанные из прибора значения **<Q>**;
- **UNOMINAL**=<число>, **UNOMINAL**=<число>**kV** и **UNOMINAL**=<число>**kV** – номинальное **U** фазы (если единицы не указаны – в вольтах).

(*). Большая буква разрешает выполнение функции, аналогичная маленькая – запрещает.

Архивы

Для каждого счетчика создается 3 архива, в которые записываются все параметры, считанные из счетчика. Файлы архивов имеют разное расширение:

- **.bin** – архив констант и журналов событий;
- **.pwr** – архив значений мощности;
- **.enr** – архив значений энергии.

Архивные данные извлекаются при изменении атрибута **CGR.59**, который задает начальное время выборки (день, месяц или год). В ряде случаев атрибут 59 возвращает округленное значение даты и времени. Атрибут

CGR.252 индицирует расстояние между точками (интервал усреднения) как число секунд.

Архивные данные записываются в аргументы **CGR** (см. **Запись вектора в CALL.ChGroupReq и CALL.TVC**). Эти данные могут быть выведены в графику и в документ (см. ниже).

Замечания

Если изменять конфигурацию обмена в реальном времени не предполагается, к аргументам **CGR** могут быть привязаны каналы.

При выключении **CGR_main** ((3, **C**) = 1) обмен со счетчиком прекращается, ни один из привязанных **CGR** не получает данных, а уже полученные данные недоступны для передачи в графику, программу и т.п.

При отключении **CGR_main** от источника ((8, **W**) = 1), обмен также прекращается, но полученные данные (в т.ч. из архива) доступны для просмотра.

Сеть

Чтобы передать данные **CGR_main** по сети, в удаленном узле нужно создать аналогичный канал (**CGR_main_rem**) и привязать к нему **CGR_main**. При конфигурировании **CGR_main_rem** в ИС нужно присвоить 100 его атрибуту **Параметр** (34, **FPrnt** в MPB). В MPB **CGR_main_rem.C0** = 0xFF (номер RS), и установится признак работы по сети. В остальном **CGR_main_rem** идентичен **CGR_main** (в т.ч. создаст архивы).

Расшифровка значений CGR.86

Отладка

Конфигурация (86, **nAttr**) = 0xFE используется как отладочная, в этом случае в аргументы **CGR** (создаются автоматически группами по 5) записывается большинство параметров, вычисляемых в счетчике, доступа к архиву нет. Полный аналог этой конфигурации: тип данных аргумента **CGR_main** – SINT, **CGR.86** = 0.

Глобальные параметры

Для установки глобальных параметров предназначена конфигурация (86, **nAttr**) = 0xEE. Такой **CGR** (далее – **CGR_global**, может быть один в узле) должен быть привязан к аргументу OUTPUT произвольного **CGR_main**. В атрибут (0, **R**) **CGR_global** могут быть записаны следующие число-команды (**CGR_global.R**=0 автоматически после отработки канала):

- 8, 9, 10 – прекратить обмен со всеми приборами соответственно на

- 1, 10 и 30 минут;
- 2, 3 – синхронизировать время всех подключенных приборов по времени компьютера (если прибор поддерживает широковещательную команду синхронизации времени);
- 1 – для этой команды **CGR_global** должен иметь 8 или 16 аргументов.

Первые 8 аргументов должны быть целочисленными 32-разрядными, эти аргументы определяют, какая информация выводится в predetermined отчеты (см. ниже **Вывод в графику и документ**):

- установленный бит с номером **N** аргумента **argM** запрещает генерацию столбца **N** в predetermined отчете **M**;
- установленный бит 30 (0x40000000) **argM** запрещает вывод фидера в отчет **M**;
- установленный бит 29 (0x20000000) **argM** запрещает вывод сумматора в отчет **M**.

Аргументы 8-15 задают дополнительные параметры:

- **arg12** – сутки, за которые выбираются значения **<P>** для отчета. Значение аргумента может быть задано собственно как дата или как одно из зарезервированных значений:
 - 0 – за предыдущие сутки;
 - 2 – за текущие сутки;
 - 5 – за предыдущие 24 часа;
- **arg13** – интервал, за который выбираются значения **E** для отчета. Значение аргумента может быть задано только как одно из зарезервированных значений:
 - 0 – за предыдущие 12 месяцев;
 - 10 – за каждый день предыдущего месяца;
 - 11 – за каждый день текущего месяца.

В **CGR_global.R** может быть также записана любая команда с кодом больше 16 из описанных в этом разделе. Результат зависит от значения старшего байта **CGR_global.R**:

- если старший байт **CGR_global.R** равен 0, команда (например, 0x11, соответственно **CGR_global.R=0x11**) записывается в атрибут (86, **nAtrt**) всех **CGR**, у которых (86, **nAtrt**)=0xEF;
- если старший байт **CGR_global.R** равен 1, команда (например, 0x11, соответственно **CGR_global.R=0x111**) записывается в атрибут (86, **nAtrt**) всех **CGR**, у которых (86, **nAtrt**)=0xED.

Временные и служебные параметры

Конфигурация (86, **nAtrt**) = 0xE1 предназначена для чтения различных

временных и служебных параметров. Полный аналог этой конфигурации: тип данных аргумента **CGR_main** – DATE_AND_TIME, **CGR.86** = 0. Число аргументов **CGR** – 4, если в ИС создано не более 4, в противном случае – 24:

- **arg0** – текущие дата и время счетчика;
- **arg1** – разность текущих времен компьютера и счетчика;
- **arg2** – среднее время сессии обмена;
- **arg3** – среднее время между сессиями за все время работы;
- **arg4** – месяц, за который считана месячная энергия;
- **arg5** – день, за который считана дневная энергия;
- **arg6** – последний день, за который считана средняя мощность в случае **CGR.86=0x31...0x34** (см. ниже);
- **arg7** – последний день, за который считана средняя мощность в случае **CGR.86=0x35**;
- **arg8** – минуты записей с начала суток;
- **arg9** – число ошибок соединения;
- **arg10** – число ошибочных транзакций;
- **arg11** – коэффициент повторов;
- **arg13...arg22** – время последнего чтения соответственно группы параметров 1...10 (номер группы – это значение полубайта 1 **CGR.86**);
- **arg23** – время последнего чтения накопленной энергии.

Для чтения служебных параметров предназначена также специальная конфигурация – тип данных аргумента **CGR_main** – BOOL, **CGR.86** = 0:

- **arg0** – интервал усреднения мощности;
- **arg1** – **AConst**;
- **arg2** – коэффициент трансформации по U;
- **arg3** – коэффициент трансформации по I;
- **arg4** – **ECNT**;
- **arg5** – **ECFG**.

Напряжение и ток

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x1	По фазе 1	10 (arg0 – частота, arg1 – U₁ , arg2 – I₁ , arg3 – U₁V_{I₁} , arg4 – P₁ , arg5 – P₁/S₁ , arg6 – Q₁ , arg7 – S₁ , arg8 – dP₁ , arg9 =0
0x2	По фазе 2	То же по фазе 2
0x3	По фазе 3	То же по фазе 3
0x4	По всем фазам	То же, суммарные параметры (по всем фазам)

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x11	U_N	4 (arg0= f , arg1= U_1 , arg2= U_2 , arg3= U_3)
0x12	I_N	4 (arg1= I_1 , arg2= I_2 , arg3= I_3)
0x13	$U_N V I_N$	4 (arg1= $U_1 V I_1$, arg2= $U_2 V I_2$, arg3= $U_3 V I_3$, в arg0 записывается суммарный угол между U и I)
0x15	U_{NM}	4 (arg0= f , arg1= U_{12} , arg2= U_{23} , arg3= U_{13})
0x16	$U_N V U_M$	4 (arg1= $U_1 V U_2$, arg2= $U_2 V U_3$, arg3= $U_3 V U_1$)
0x17	Характеристики U_N ; если $C4=0$ – за предыдущий день, если $C4=1$ – за текущий день	30 (3 группы по 10 аргументов на фазу). В группе: arg0 – U_{MAX} , arg1 – время U_{MAX} (левое); arg2 – U_{MIN} , arg3 – время U_{MIN} (левое); arg4= $\langle U \rangle$; arg5 – счетчик случаев $U > BNR_1$; arg6 – суммарное время, в течение которого $U > BNR_1$; arg7 – счетчик случаев $U > BNR_2$; arg8 – суммарное время, в течение которого $U > BNR_2$; arg9= $(U_{MAX}-U_{MIN})/\langle U \rangle$ $BNR_1=1.05 \cdot UNOMINAL$ $BNR_2=0.95 \cdot UNOMINAL$

Мгновенная мощность

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x21	P, P_N	4 (arg0= P , arg1= P_1 , arg2= P_2 , arg3= P_3)
0x22	Q, Q_N	4 (arg0= Q , arg1= Q_1 , arg2= Q_2 , arg3= Q_3)
0x23	S, S_N	4 (arg0= S , arg1= S_1 , arg2= S_2 , arg3= S_3)
0x25	dP, dP_N	4 (arg0= dP , arg1= dP_1 , arg2= dP_2 , arg3= dP_3)
0x26	$P/S, P_N/S_N$	4 (arg0= P/S , arg1= P_1/S_1 , arg2= P_2/S_2 , arg3= P_3/S_3)
0x24	P, Q, S, dP	4 (arg0= P , arg1= Q , arg2= S , arg3= dP)
0x28	Пик мощности: если $C4=0$ – за предыдущий день, если $C4=1$ – за текущий день	8 (arg0 – время arg1, arg1 – утренний (до 12ч) максимум P_e , arg2 – время arg3, arg3 – утренний максимум Q_e , arg4 – время arg5, arg5 – вечерний максимум P_e , arg6 – время arg7, arg7 – вечерний максимум Q_e)

Средняя мощность

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x35	Последние $\langle P \rangle$ и $\langle Q \rangle$	4 (arg0= $\langle P_e \rangle$, arg1= $\langle P_i \rangle$, arg2= $\langle Q_e \rangle$, arg3= $\langle Q_i \rangle$)
0x31 при (97,C4)=0	Все $\langle P_e \rangle$ за день (*)	Число аргументов зависит от интервала усреднения счетчика

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x32 при (97, C4)=0	То же, <P _i > (*)	То же
0x33 при (97, C4)=0	То же, <Q _e > (*)	То же
0x34 при (97, C4)=0	То же, <Q _i > (*)	То же
0x31...0x34 при (97, C4)=1	Соответственно <P _e >, <P _i >, <Q _e > и <Q _i > (*)	То же, по считанной средней мощности вычисляется соответствующая энергия в кВт-ч (значение счетчика делится на соответствующее число).
0x31...0x34 при (97, C4)=2	То же (*)	24, по считанной средней мощности вычисляется соответствующая энергия за каждый час (в кВт-ч)
0x3B – задание желаемого графика нагрузок	(*)	Число аргументов зависит от интервала усреднения счетчика. В день перехода: если час «пропадает», соответствующие аргументы обнуляются (значение этих аргументов восстанавливается, если задается другой день). Если час добавляется, МРВ создает недостающие аргументы. В ИС такие аргументы также могут быть заданы – они должны размещаться после аргумента BOOL. Если в ИС после некоторого аргумента arg создан аргумент UINT с начальным значением N, МРВ создает N аргументов, идентичных arg .
0x3A при CGR.R =0 – коррекция архива	(*)	4 (arg0=<P _e >, arg1=<P _i >, arg2=<Q _e >, arg3=<Q _i >) Алгоритм: - при изменении CGR.59 значения на заданное время считываются из архива; - корректируются значения аргументов; - при изменении CGR.41 значения аргументов записываются в архив со временем, заданным CGR.59 .
0x3A при CGR.R =1 – принудительное чтение из прибора с записью в архив	(*)	4 (arg0=<P _e >, arg1=<P _i >, arg2=<Q _e >, arg3=<Q _i >) При изменении CGR.59 значения на заданное время считываются из прибора и записываются в аргументы и в архив.

* Глубина архивных данных – 3 года.

Текущая энергия

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x81	Текущие значения энергии (в т.ч. по тарифам) с начала работы	$(T_{MAX}+1)*4$ (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i , arg4= EP¹_e и т.д.)
0x85	Текущие значения суммарной энергии с начала работы	4 (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i)

Энергия за день

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x82	EP_e за каждый день месяца (*) нарастающим итогом, в т.ч. по тарифам	Число аргументов равно числу дней в месяце, CGR.A задает номер тарифа (CGR.A =0 – сумма по всем тарифам)
0x86	То же, EP_i (*)	То же
0x8A	То же, EQ_e (*)	То же
0x8E	То же, EQ_i (*)	То же
0x93	То же, EP_e с начала дня (*)	То же
0x96	То же, EP_i (*)	То же
0x99	То же, EQ_e (*)	То же
0x9C	То же, EQ_i (*)	То же
0x90	EP_e , EP_i , EQ_e и EQ_i за указанный день, в том числе по тарифам (с начала дня) (*)	32 (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i , arg4= EP¹_e и т.д.) Если CGR.R =0, при CGR.41 =1 все аргументы записываются в архив. Если CGR.R =1, данные принудительно считываются из прибора и записываются в архив (в архив записываются значения с начала дня даже в том случае, если прибор возвращает данные нарастающим итогом).
0x9D	То же, нарастающим итогом (*)	То же, нарастающим итогом
0x92	Текущие значения энергии (в т.ч. по тарифам) с начала текущего дня	$(T_{MAX}+1)*4$ (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i , arg4= EP¹_e и т.д.)
0x95	Текущие значения суммарной энергии с начала текущего дня	4 (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i)

* Глубина архивных данных – 3 года.

Энергия за месяц

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x83	EP_e за каждый месяц года (*) по заданному тарифу (нарастающим итогом)	12, CGR.A задает номер тарифа (CGR.A=0 – сумма по всем тарифам)
0x87	То же, EP_i (*)	То же
0x8B	То же, EQ_e (*)	То же
0x8F	То же, EQ_i (*)	То же
0x91	То же, EP_e с начала месяца (*)	То же
0x94	То же, EP_i (*)	То же
0x97	То же, EQ_e (*)	То же
0x9A	То же, EQ_i (*)	То же
0x9F	EP_e, EP_i, EQ_e и EQ_i за указанный месяц, в том числе по тарифам (нарастающим итогом) (*)	32 (arg0= EP_e , arg1= EP_i , arg2= EQ_e , arg3= EQ_i , arg4= EP_e¹ и т.д.). Если CGR.R=0 , при CGR.41=1 все аргументы записываются в архив. Если CGR.R=1 , данные принудительно считываются из прибора и записываются в архив.
0x9E	То же, с начала месяца (*)	То же

* Глубина архивных данных – 10 лет.

Нелинейные искажения

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x71	HRM1...HRM40 по каждой фазе	120 (arg0= HRM1₁ , arg1= HRM2₁ и т.д.)
0x72	THD напряжения по каждой фазе	4 (arg0 не используется, arg1= THD_{u1} , arg2= THD_{u2} , arg3= THD_{u3})
0x73	THD тока по каждой фазе	4 (arg0 не используется, arg1= THD_{i1} , arg2= THD_{i2} , arg3= THD_{i3})
0x74	THD межфазного напряжения	4 (arg0 не используется, arg1= THD_{u12} , arg2= THD_{u23} , arg3= THD_{u13})
0x76	U	4 (arg0 не используется, arg1 – U прямой последовательности основной частоты, arg2 – U обратной последовательности основной частоты, arg3 – U нулевой последовательности основной частоты)

CGR.86	Группа параметров	Число и расшифровка аргументов CGR
0x78	Коэффициент несинусоидальности напряжения	4 (arg0 не используется, arg1 – коэффициент несинусоидальности U прямой последовательности основной частоты, arg2 – коэффициент несинусоидальности U обратной последовательности основной частоты, arg3 – коэффициент несинусоидальности U нулевой последовательности основной частоты)

Журналы

Журналы имеют одинаковую структуру, сообщение содержит время фиксации события, время окончания события (или время перехода к другому событию), код события (32 бита) и субкод события (дополнительная информация о событии).

CGR.86	Группа параметров	Число и расшифровка аргументов CGR (*)
0x60	Журнал событий связи с прибором	Коды событий: 1 – прибор выключен (аналог 3, C =1 у канала счетчика); 2 – индикация запуска MPB; 3 – прибор отключен от обмена (аналог 8, W =1 у канала); 4 – не удалось установить канал связи с прибором; 5 – ошибка чтения параметра из прибора; 6 – ошибка чтения параметра из прибора (зарезервировано); 7 – ошибка чтения параметра из прибора на старте.
0x62	Журнал событий средней мощности. Канал OUTPUT: 4 младших бита атрибута (1, A) разрешают (0) или запрещают (1) анализ соответствующей мощности (журнал не ведется).	Значения групп битов кода события: – биты 0-3 – номер интервала TRACE MODE (**), в котором находится <P_e> ; – биты 4-7 – то же, <P_i> ; – биты 8-11 – то же, <Q_e> ; – биты 12-15 – то же, <Q_i> .
0x63	Журнал событий I по фазам. Канал OUTPUT: 3 младших бита атрибута (1, A) разрешают (0) или за-	Значения групп битов кода события: – биты 0-3 – номер интервала TRACE MODE (**), в котором находится значение I по фазе 1; – биты 4-7 – то же по фазе 2;

CGR.86	Группа параметров	Число и расшифровка аргументов CGR (*)
	прещают (1) анализ соответствующей фазы (журнал не ведется).	- биты 8-11 – то же по фазе 3; - биты 12-15 – каждый бит индицирует состояние соответствующей фазы (0 – включена, 1 – выключена). Если код события равен 0x8000 (установлен только бит 15), все параметры находятся в норме.
0x64	То же, U Если из архива в канал считаны данные, которые были получены из прибора (а не вычислены монитором), в атрибуте (1, A) канала устанавливается бит 15 (0x8000).	То же, U (**).
0x61	То же, P/S, f Канал OUTPUT: 3 младших бита атрибута (1, A) разрешают (0) или запрещают (1) анализ соответствующей фазы (журнал не ведется).	То же, P/S (**). Значение битов 12-15 кода события – номер интервала TRACE MODE, в котором находится f .
0x65	Журнал статуса устройства	Код и субкод события зависят от прибора (см. документацию прибора).
0x6F	Последние сообщения журналов 0x60-0x65	6 пар аргументов (arg0 – последнее сообщение журнала 0x60, arg1 – метка времени последнего сообщения журнала 0x60 и т.д.) Журнал 0x6F напрямую с архивом не взаимодействует; при изменении его атрибута 59 MPB ищет все другие журналы и изменяет их атрибут 59, после чего обнуляет атрибут 59 журнала 0x6F.

(*) Число аргументов равно числу сообщений в журнале с указанного времени. Глубина журналов 0x60-0x65 – 512 сообщений.

(**) Границы задаются аргументами OUTPUT канала **CGR** (в данном случае нужно использовать CALL.ChGroupReq). Допустимое число аргументов для **<P_e>**:

- 4 (одна граница на каждую мощность, интервал 5 или 0);

- 8 (две границы на каждую мощность, интервал 0, 5 или 3).

Допустимое число аргументов для **I**:

- 1 – интервал 5 или 0;
- 2 – интервал 5, 3 или 0;
- 3 – интервал 5, 3, 1 или 0.

Допустимое число аргументов для **U**:

- 2 – интервал 6, 5 или 0 (если значения аргументов не заданы, аргументы инициализируются значениями $1.05 \cdot \mathbf{UNOMINAL}$ и $0.95 \cdot \mathbf{UNOMINAL}$);
- 4 – интервал 6, 5, 4, 3 или 0 (если значения аргументов не заданы, аргументы инициализируются значениями $1.1 \cdot \mathbf{UNOMINAL}$, $1.05 \cdot \mathbf{UNOMINAL}$, $0.95 \cdot \mathbf{UNOMINAL}$ и $0.9 \cdot \mathbf{UNOMINAL}$);
- 6 – все интервалы.

Допустимое число аргументов для журнала 0x61 (**P/S** и **f**):

- 1 – только сравнение **P/S** по каждой фазе (интервалы 0 и 6);
- 2 – только сравнение **f** (интервалы 0, 6 и 5);
- 3 – $\text{arg0}=\mathbf{f_{MAX}}$, $\text{arg1}=\mathbf{f_{MIN}}$ (интервалы 0, 5 и 6), $\text{arg2}=(\mathbf{P/S})_{\text{MIN}}$ (интервалы 0 и 6);
- 4 – границы **f** (интервалы 0, 3-6);
- 6 – $\text{arg0}-\text{arg3}$ – границы **f** (интервалы 0, 3-6), arg4 и arg5 – границы **P/S** (интервалы 0, 4 и 6).

В случае двух границ для **f**: если значения таких аргументов не заданы, аргументы инициализируются значениями $\pm 2\%$ от размерности (50Гц по умолчанию).

В случае четырех границ для **f**: если значения таких аргументов не заданы, аргументы инициализируются значениями $\pm 2\%$ и $\pm 4\%$ от размерности (50Гц по умолчанию).

В случае одной границы для **P/S**: если значение такого аргумента не задано, аргумент инициализируется значением 0.65.

В случае двух границ для **P/S**: если значения таких аргументов не заданы, аргументы инициализируются значениями 0.65 и 0.95.

Реальное значение канала (как INPUT, так и OUTPUT) индицирует текущее сообщение для соответствующего журнала.

Каналы OUTPUT

Кроме указанных выше, допустимы **CGR** OUTPUT при следующих значениях **CGR.86**:

- 0x3A, 0x90, 0x9D, 0x9F, 0x9E – данные, записанные в аргументы вручную; записываются в архив с временем **CGR.45**;
- 0x11 – у такого канала должно быть 8 аргументов, значения кото-

рых задаются вручную (в архив не записываются):

- $\text{arg}0$ – частота;
- $\text{arg}1 \dots \text{arg}3$ – **U** по фазе 1...3;
- $\text{arg}4=0$;
- $\text{arg}5 \dots \text{arg}7$ – **I** по фазе 1...3.

При наличии такого канала МРВ не считывает указанные параметры из прибора;

- $0x21 \dots 0x24$, $0x26$ – у такого канала должно быть 4 аргумента, которые задают суммарный параметр и параметр по фазам 1-3 (соответственно для **P_e**, **P_i**, **Q_e**, **Q_i** и **P/S**). При наличии такого канала МРВ не считывает указанные параметры из прибора.

Поддерживаемые приборы

Универсальный механизм может быть использован для обмена со следующими приборами (число слева – это задаваемый в TRACE MODE тип прибора (**CGR_main.94**)):

- 1 – Меркурий 230 (*);
- 2 – СЭТ-4ТМ2 (*);
- 3 – СЭТ-4ТМ1 (*);
- 4 – СЭТ-4ТМ3 (*);
- 5 – ПСЧ-3М, ПСЧ-4М;
- 6 – Iskra_MI;
- 7 – Iskra_MT;
- 8 – Iskra_MC;
- 9 – Iskra_WS;
- 10 – CVM_NGR;
- 11 – CVM_BD;
- 12 – CVM_K2;
- 13 – СЭБ-х.ТМ;
- 14 – Энергомера 102;
- 16 – Энергомера 6850;
- 17 – Энергомера 301;
- 18 – Энергомера 303;
- 19 – Энергомера 304;
- 20 – Энергомера 201;
- 30 – UPM 215, 310, 315;
- 31 – UPM 303;
- 32 – UPM 304;
- 33 – UPM 305;

- 40 – прибор, поддерживающий стандарт IEC 62056-21 (2002-05), чтение – обычное (используется команда R1) (данная опция доступна в релизе 6.07.1 и старше);
- 42 – прибор, поддерживающий стандарт IEC 62056-21 (2002-05), чтение – форматированное (используется команда R2) (данная опция доступна в релизе 6.07.1 и старше);
- 44 – SATEC BFM 136;
- 45 – SATEC PM 175;
- 50 – СЭБ-2А.07, СЭБ-2А.08, ПСЧ-ЗТА.07;
- 51 – ПСЧ-ЗАРТ.07, ПСЧ-ЗАРТ.08.

(*) Реализована блокировка команд чтения счетчика, если в ответе содержится недопустимый параметр или команда.

Кроме того, могут быть заданы следующие значения **CGR_main.94**:

- 125 – мультиплексор (**CGR_mux**). Для подключения к прибору (каналу **CGR_main**) нужно атрибуту **CGR_mux.55** присвоить ID или имя канала **CGR_main**. Если **CGR_mux** не подключен, **CGR_mux.CMNT=0**, первый символ имени **CGR_mux** – «?», а во всех каналах, привязанных к **CGR_mux**, устанавливается признак аппаратной недостоверности. К аргументам **CGR_mux** могут быть привязаны любые каналы-результаты (запрещена привязка каналов-журналов типа OUTPUT).
- 126 – эмулятор прибора (модель 1);
- 127 – эмулятор прибора (модель 2);
- 128 – эмулятор прибора (пустая модель);
- 129...133 – сумматоры (см. ниже).

Сумматоры

Сумматором (далее – **CGR_sum**) является канал **CGR_main** с **CGR_main.94=129...133** (соответственно **CGR_sum1...CGR_sum5**). К аргументам **CGR_sum** привязываются каналы **CGR** (далее – **cgr**).

Сумматоры не взаимодействуют с приборами, они только вычисляют определенные характеристики по текущим или архивным данным. По текущим данным характеристики вычисляются непрерывно, архивные данные обрабатываются по следующему алгоритму:

Если в сумматоре задан корректный COM-порт (атрибут 93, **C0**), сумматор пересчитывается с циклом RS (как прибор). Если **.CGR_sum.C0=0xFF**, номер COM-порта не анализируется, а сумматор пересчитывается со своим периодом.

- при изменении **cgrN.59** производится поиск каналов **CGR_main**, у которых **CGR_main.C5 = 0**;

Если **CGR_main.C5** = <номер фидера> <> 0, канал является каналом счетчика на входе фидера.

- в найденных **CGR_main** производится поиск соответствующих **CGR**, у которых **CGR.86=cgr_N.86** (образуется группа **CGR**):
 - **CGR_sum1** обрабатывает те **CGR**, у которых **CGR.C4 = CGR_sum1.C4** (номер группы);
 - **CGR_sum2** обрабатывает те **CGR**, у которых **CGR.C5 = CGR_sum1.C4**;
 - **CGR_sum3** аналогичен **CGR_sum2**, но выбирает из найденных каналов только **CGR_sum1** и **CGR_sum2**;
 - **CGR_sum4** обрабатывает каналы **CGR_mux (CGR_main с CGR_main.94 = 125)**;
 - **CGR_sum5** обрабатывает каналы **CGR_main**, размерность (атрибут (82, **DIM**)) которых совпадает с размерностью **CGR_sum5**;
- в каждом **CGR** группы устанавливается **CGR.59=cgr_N.59**;
- после того, как во все **CGR** группы будут считаны архивные данные (**DataReady=1**), сумматор обрабатывает их; результат обработки записывается в **cgr_N**.

Атрибут **cgr.86** может принимать следующие значения (в дополнение к описанному выше):

- 0xD1 – разность между **P_e** фидера и суммой **P_e** группы (число аргументов равно числу интервалов в архиве);
- 0xD2 – то же, **P_i**;
- 0xD3 – то же, **Q_e**;
- 0xD4 – то же, **Q_i**;
- 0xD5 – результат 0xD1, деленный на **P_e** фидера и умноженный на 100;
- 0xD6 – результат 0xD2, деленный на **P_i** фидера и умноженный на 100;
- 0xD7 – результат 0xD3, деленный на **Q_e** фидера и умноженный на 100;
- 0xD8 – результат 0xD4, деленный на **Q_i** фидера и умноженный на 100;
- 0xD9 – 4 аргумента содержат результаты 0xD1...0xD4, но вычисленные не по архивным, а по текущим данным (атрибут 59 не используется);
- 0xDA – 4 аргумента содержат результаты 0xD5...0xD8, но вычисленные не по архивным, а по текущим данным;
- 0xA0 – абсолютный небаланс **EP_e** (разность между **EP_e** фидера и суммой **EP_e** группы) за каждый день указанного месяца (число ар-

- гументов равно числу дней в месяце);
- 0xA1 – то же, **EP_i**;
 - 0xA2 – то же, **EQ_e**;
 - 0xA3 – то же, **EQ_i**;
 - 0xA4 – относительный небаланс **EP_e** за каждый день указанного месяца (результат 0xA0, деленный на **EP_e** фидера и умноженный на 100);
 - 0xA5 – то же, **EP_i** (результат 0xA1, деленный на **EP_i** фидера и умноженный на 100);
 - 0xA6 – то же, **EQ_e** (результат 0xA2, деленный на **EQ_e** фидера и умноженный на 100);
 - 0xA7 – то же, **EQ_i** (результат 0xA3, деленный на **EQ_i** фидера и умноженный на 100);
 - 0xA8...0xAF – аналог 0xA0...0xA7 за каждый месяц указанного года (12 аргументов);
 - 0xF1...0xFC – при этих значениях МРВ автоматически создает аргументы **сgr** (в ИС аргументы не создаются) и привязывает к ним каналы-источники данных (каналы группы) или каналы-результаты:
 - arg0...argN – каналы-источники (каналы группы);
 - arg(N+1) – канал-результат (сумма по группе);
 - arg(N+2) – канал-фидер;
 - arg(N+3) – канал-небаланс абсолютный;
 - arg(N+4) – канал-небаланс относительный.

Отличия:

- 0xF1 – привязка канала-результата последнего по времени суммирования;
 - 0xF2 – привязка канала-результата при условии, что суммировалась **PE, Pi, QE** или **Qi**;
 - 0xF3...0xF6 – привязка канала-результата при условии, что суммировалась соответственно **PE, Pi, QE** и **Qi**;
 - 0xF8 – привязка канала-результата при условии, что суммировалась **EP_e, EP_i, EQ_e** или **EQ_i** (*);
 - 0xF9...0xFC – привязка канала-результата при условии, что суммировалась соответственно **EP_e, EP_i, EQ_e** или **EQ_i** (*).
- (*) Вне зависимости, дневная или месячная.

Диагностика в каналах подтипа 102

Установленные в 1 биты атрибута 2,А канала подтипа 102 индицируют следующие ошибки:

- биты 0...2 – **U** по фазе 1...3 не в норме;

- биты 4...6 – то же, **I**;
- биты 8...10 – то же, **P/S**.

Значение битов 12-15 индицирует следующие ошибки:

- 1 – ошибка открытия канала связи;
- 2 – ошибка чтения прибора;
- 3 – ошибка чтения значений средней мощности;
- 8 – ошибка чтения дополнительных параметров (время прибора).

Значение битов **A** в **CGR_sum** формируется как логическая сумма (OR) соответствующих битов **A** обрабатываемых **CGR_main**.

Глобальная диагностика

Для глобальной диагностики используется **CGR** с **CGR.86 = 0xEC**, привязанный к произвольному **CGR_main** (далее такой **CGR** обозначается как **CGR_gdiag**).

CGR_gdiag отбирает для анализа приборы – каналы **CGR_main** (за исключением каналов **CGR_mux**) – по следующему алгоритму:

- **CGR_gdiag.C4 = CGR_gdiag.C5 = 0xFF** – все;
- **CGR_gdiag.C4 <> 0xFF** или **CGR_gdiag.C5 <> 0xFF**:
 - **CGR_gdiag.C4 = 0xFF** – по условию **CGR_main.C5 = CGR_gdiag.C5**;
 - **CGR_gdiag.C5 = 0xFF** – по условию **CGR_main.C4 = CGR_gdiag.C4**;
- **CGR_gdiag.C4 <> 0xFF** и **CGR_gdiag.C5 <> 0xFF** – по условию **CGR_main.C4 = CGR_gdiag.C4** и **CGR_main.C5 = CGR_gdiag.C5**.

В отобранных каналах **CGR_main** анализируется атрибут (2, **A**), результаты записываются в 16 аргументов **CGR_gdiag**:

- **arg0** – общее число приборов;
- **arg1** – число неинициализированных приборов;

Если все приборы инициализированы, **CGR_gdiag.R = 1**, в противном случае **CGR_gdiag.R = 0**.

- **arg2** – число приборов с проблемами чтения **<P>**;
- **arg3** – число приборов с общими проблемами чтения;
- **arg4** – число приборов, в которых **f** вне границ;
- **arg5** – число приборов с отключением фаз;
- **arg6** – число приборов, в которых **U** или **I** вне границ;
- **arg7** – число приборов, в которых **P/S** вне границ;

- **arg8** – число приборов, в которых **P** вне границ;
- **arg9** – число приборов, чье время отличается от времени компьютера больше, чем на 30с;
- **arg10** – самая левая временная метка;
- **arg11** – самая левая временная метка **<P>**;
- **arg12** – самая левая временная метка **E** от сброса;
- **arg13** – временная метка, ближайшая к текущему времени (из всех журналов всех приборов);
- **arg14** – зарезервировано;
- **arg15** – зарезервировано.

Вывод в документ

Каналы 103

Левая и правая границы архивного тренда документа могут быть привязаны соответственно к атрибутам 500 и 501 **CGR**. **CGR.500** возвращает значение **CGR.59**, **CGR.501** – время последней точки.

Чтобы сравнить две кривые за разные промежутки времени, для одной из кривых можно сконфигурировать отображение, которое не зависит от диапазона, заданного левой и правой границами тренда:

- сбросить флаг **Запрос времени значения** в **CGR**;
- после того, как архивные данные считаны в аргументы **CGR**, отключить **CGR** от источника ((8,**W**)=1).

Каналы 103 - журналы

Журналы могут быть выведены в документ с помощью **SubNum=552-559** (если привязан журнал 0x6F, в таблицу в виде столбцов выводятся все журналы (0x60-0x65)).

При выводе нескольких журналов их сообщения размещаются в таблице в хронологическом порядке (за исключением сообщений в пределах часа).

В таблице индицируется «+», если фаза включена, и «-» в противном случае (только для журнала 0x64 (**U**)).

При выводе интервала:

- если к **CGR** привязан словарь для канала **FLOAT**, вместо номеров интервалов индицируются сообщения, заданные в словаре (только для журналов 0x62-0x64 (**<P>**, **I** и **U**));
- вне зависимости от интервала индицируется «...», если фаза отключена.

Ниже показан один из predetermined отчетов:

29.03.2010 17:01:48

ES Number	ES Time	few sts	Reading Time	Err open	Err read	Link ev	Link sts	F-kp ev	power ev	ampere ev	volts ev	unit sts ev
serial_number	ES Time	few sts	Reading Time	Err open	Err read	Link ev	Link sts	F-kp ev	power ev	ampere ev	volts ev	unit sts ev
Name	ES Number	ES Time	few sts	dt read	freq	Phase sts	I	U	du	anglu		
EMULATOR TC1	serial_number 220B	29.03.10 17:01:47	00000070	29.03.10 17:01:47	0	0	0	0	0	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:47	00000000	29.03.10 17:01:47	0	0	0	0	0	0	0	0
sum TC1	400B	29.03.10 17:01:47	00040070	29.03.10 17:01:32	0	0	0	0	0	0	0	0
EMULATOR TC1	serial_number 220B	29.03.10 17:01:47	00000070	29.03.10 17:01:47	50	I	10	220	0	120		
EMULATOR2 TC1	380B	29.03.10 17:01:47	00000000	29.03.10 17:01:47	50	II	11,7	220	0	120		
sum TC1	400B	29.03.10 17:01:47	00040070	29.03.10 17:01:32	50	III	21,7	220	0	120		
EMULATOR TC1	serial_number 220B	29.03.10 17:01:47	00000070	29.03.10 17:01:31	50	I	2,1978	-0,0022	2,1978	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:47	00000000	29.03.10 17:01:32	50	II	2,1978	-0,0022	2,1978	0	0	0
sum TC1	400B	29.03.10 17:01:47	00040070	29.03.10 17:01:32	50	III	6,5934	-0,0066	6,5934	0	0	0
EMULATOR TC1	serial_number 220B	29.03.10 17:01:47	00000070	29.03.10 17:01:31	50	I	2,1978	-0,0022	2,1978	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:47	00000000	29.03.10 17:01:32	50	II	2,1978	-0,0022	2,1978	0	0	0
sum TC1	400B	29.03.10 17:01:47	00040070	29.03.10 17:01:32	50	III	6,5934	-0,0066	6,5934	0	0	0
EMULATOR TC1	serial_number 220B	29.03.10 17:01:47	00000070	29.03.10 17:01:31	50	I	2,1978	-0,0022	2,1978	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:47	00000000	29.03.10 17:01:32	50	II	2,1978	-0,0022	2,1978	0	0	0
sum TC1	400B	29.03.10 17:01:47	00040070	29.03.10 17:01:32	50	III	6,5934	-0,0066	6,5934	0	0	0
EMULATOR TC1	serial_number 220B	29.03.10 17:01:31	0	0	0	0	0	0	0	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:32	0	0	0	0	0	0	0	0	0	0
sum TC1	400B	29.03.10 17:01:32	0	0	0	0	0	0	0	0	0	0
EMULATOR TC1	serial_number 220B	29.03.10 17:01:31	0	0	0	0	0	0	0	0	0	0
EMULATOR2 TC1	380B	29.03.10 17:01:32	0	0	0	0	0	0	0	0	0	0
sum TC1	400B	29.03.10 17:01:32	0	0	0	0	0	0	0	0	0	0

Каналы 102

Каналы 102 могут быть выведены в документ с помощью следующих SubNum (канал привязывается так же, как канал 103):

- 544 – **E** (предопределенный отчет 1);
- 545 – **P** (предопределенный отчет 3);
- 546 – **I** и **U** (предопределенный отчет 2);
- 547 – статус, журналы (предопределенный отчет 0);
- 548 – архив <**P**> (предопределенный отчет 4);
- 549 – архив **E** (предопределенный отчет 5).

SubNum=644... задают аналогичный вывод, но без заголовка.

Если канал 102 – сумматор, то в predeterminedенные отчеты выводится:

- значения из всех приборов, с которыми сумматор работает;
- значения из фидера (если найден). Вывод фидера можно запретить с помощью **CGR_global**;
- значения из самого сумматора. Вывод сумматора можно запретить с помощью **CGR_global**.

Наименования столбцов для predeterminedенных отчетов считываются из файла <директория проекта>**ES_COLUMN_NAME.txt** (если при старте MPV файл не обнаружен, он генерируется с именами столбцов по умолчанию). В файле допустимы комментарии (строки, начинающиеся с символа «;»). Общее число строк в файле – 128+8, последние 8 строк задают форматы вывода параметров (в нотации Си):

- строка 128 – **U**;
- строка 129 – **I** и **f**;
- строка 130 – углы;
- строка 131 – **P** и **P/S**;
- строка 132 – **<P>**;
- строка 133 – **<Q>**;
- строка 134 – **E** от сброса;
- строка 135 – **E** из архива.

MPB как сервер протоколов поверх TCP/IP

MPB может выступать в качестве сервера при обмене по различным протоколам поверх сетевых протоколов. Для конфигурирования такого режима в файле *.cnf используются следующие ключи:

- **SLAVE_TCPADDR**=<IP-адрес сервера> (эквивалентен ключу **SLAVE_UDPADDR** – см. **Управление и диагностика обмена**);
- **SLAVE_TCPPORT**=<порт сервера> (эквивалентен ключу **SLAVE_UDPPORT** – см. **Управление и диагностика обмена**);
- **SLVPUIP**=<число> – запуск соответствующего сервера:
 - **SLVPUIP**=0 – сервер MODBUS TCP/IP (существует во всех модификациях MPB);
 - **SLVPUIP**=1 – сервер протокола МЭК 3;
 - **SLVPUIP**=6, 7, 8 или 9 – сервер протокола МЭК 60870-104 (существует только в профессиональном MPB):
 - 6 или 7 – по команде общего опроса без меток времени;
 - 8 или 9 – по команду общего опроса с метками времени.

При обмене по сети поддерживается только один протокол. Для его смены нужно отредактировать файл *.cnf и перезапустить MPB.

В узлах **RTM** и **Embedded_RTМ** для каналов, доступных по серверным протоколам, необходимо установить флаг **Регистратор** на вкладке **Архивация**.

MPB как сервер MODBUS TCP/IP

Параметры настройки в Trace Mode 6

Необходимо указать флаги **Прием** и **Посылка** для сетевых адаптеров текущего узла на вкладке **Основные** редактора узла. Имя компьютера или IP-адрес указывать не обязательно.

Обмен в режиме ModBus TCP/IP Slave не работает для нулевых узлов проекта. Нулевой узел – узел, созданный самым первым в слое **Система** и имеющий индекс 0.

Особенности настройки клиентов, подключаемых к Trace Mode 6 по протоколу ModBus TCP/IP

Параметры подключения:

- **IP-адрес** – определяется сетевыми настройками ПК, на котором МРВ исполняет узел проекта;
- **Номер порта** – по умолчанию 502;
- **Номер регистра** – ID канала в базе узла;
- **Номер функции** – обрабатываются функции 3 и 4, иначе ответ с кодом 1.
Если канала нет – ответ с кодом 2.
Функция 3 – для запроса числовых каналов.
Функция 4 – для запроса аргументов каналов CALL.

По умолчанию, при чтении MODBUS SLAVE в МРВ-клиенте код преобразования установлен в 0. Чтобы установить код преобразования в 1, нужно использовать следующий ключ в файле *.cnf:

- **MDB_SLAVE_SWAP_FLOAT=ON**

МРВ как сервер протокола МЭК 60870-104

Помимо общих ключей (см. **МРВ как сервер протоколов поверх TCP/IP**), в файле *.cnf для МРВ-сервера могут быть заданы те же ключи, что и для МРВ-клиента (см. **Обмен по IEC 60870-104**).

Для обмена МРВ по умолчанию использует порт 2404.

Необходимым условием участия канала в формировании списков посылки является установленный в этом канале флаг **Регистратор**, при этом в регистратор информация о канале не передается (см. также **Уменьшение записей канала FLOAT в SIAD** в разделе **Архивы SIAD**).

Для инициализации обмена клиент должен подключится к сокету МРВ и послать команду **startdt(act)**. В ответ МРВ посылает команду **startdt(con)**.

Стандартный формат посылки:

- общий адрес ASDU – 2 байта;
- причина передачи – 2 байта;
- адрес объекта – 3 байта.

Реакция МРВ на ошибку в принятом ASDU:

- неизвестный IDT – посылается COT=44;
- неверный IOA – посылается COT=47;
- неверный общий адрес ASDU – посылается COT=46.

Для задания общего адреса ASDU используется ключ **SP104_CASDU_ADDR=<адрес. DEC>** в файле *.cnf. Если ключ не задан, общий адрес ASDU равен номеру узла.

MPV поддерживает следующие IDT:

- в направлении управления:
 - 100 (QOI=20 / 21..36) – команда общего опроса / команда опроса группы;
 - 101 (5/1-4) – команда общего опроса счетчиков / команда опроса группы счетчиков;
 - 45, 58* – one-position command; команда записи 0 или 1 в атрибут (2, **In**) канала, заданного полученным ASDU;
 - 46, 59* – то же, команда записи 1 или 2 (dual-position command);
 - 50, 63* – то же, команда записи short float;
 - 51, 64* – то же, команда записи 32 бит;
 - * При IDT=58, 59, 63 или 64 передается также время послыки (в формате CP562A), однако MPV не анализирует этот параметр.
 - 102 – команда чтения. MPV посылает значение атрибута (0, **R**) канала, заданного полученным IOA. С помощью данной команды реализуется также запись аргументов канала CALL.ChGroupReq, CALL.TVC или CALL.AS_DATA в аргументы аналогичного канала;
- в направлении контроля:
 - IDT = 1 (30) – если изменился бит в канале HEX16 или HEX32, который сконфигурирован следующим образом:
 - сброшен флаг **DEC** ((84, **HD**)=0);
 - **Размерность в битах** (56, **nBits**) = 1;
 - IDT = 7 (33) – если изменилось значение в канале HEX16 или HEX32, который сконфигурирован следующим образом:
 - сброшен флаг **DEC** ((84, **HD**)=0);
 - **Размерность в битах** (56, **nBits**) > 1;
 - IDT = 15 (37) – если изменилось значение в канале HEX16 или HEX32, который сконфигурирован следующим образом:
 - установлен флаг **DEC** ((84, **HD**)=1);
 - IDT = 13 (36) – если изменилось значение в канале FLOAT или DOUBLE FLOAT.

Если в канале установлен флаг **Отчет тревог** (43, **EB**), в список послыки заносится значение со временем ((45, **T**) и (88, **ms**)).

Канал HEX16 с числом бит 16 работает аналогично HEX32 при наличии ключа **IEC_ENABLE_HEX16=ON**.

MPB формирует списки посылки на 100(QOI=20) по следующему алгоритму:

- если для запуска MPB-сервера используется ключ **SLVPUIP=6** или 7, в списки заносятся значения каналов без меток времени (IDT=7, 13, 15);
- если для запуска MPB-сервера используется ключ **SLVPUIP=8** или 9, в списки заносятся значения каналов и их метки времени.

MPB формирует описатель качества значения канала в зависимости от значения ключа **TM_FA_QDS** в файле *.cnf (см. описание атрибута (61, Nind) **QDS** в разделе **Атрибуты каналов, отображаемые профайлером**).

Другие ключи в файле *.cnf:

- **IEC_TESTFR=<n, DEC>** – посылка команды **testfr**, если не было принято данных в течение **n** секунд. Если результат выполнения команды отрицательный, сокет закрывается. По умолчанию, **IEC_TESTFR=20с**;
- **IEC_SLAVE_TESTFR=<n, DEC>** – посылка команды **testfr**, если не было принято данных в течение **n** секунд;
- **_DONT_CLOSE_BY_ERROR=ON/OFF** – этот ключ управляет состоянием клиентского сокета в случае ошибки отсылки **testfr(act)** (**OFF** – закрыть сокет);
- **SP104_STOPDT=CLOSE** – сокет закрывается после получения **STOPDT(act)** и отправки **STOPDT(con)**;
- **SP104_TWO_PORT=ON** – подключение на прослушивание второго порта с номером на 1 больше, чем основной;
- **SLAVE_TCPPORT=<номер порта, DEC>** – основной порт прослушивания (по умолчанию 2404);
- **SP104_BLOCK_COT3=<число HEX (0 по умолчанию)>** – блокировка передачи IDT по изменению (см. выше описание поддерживаемых IDT в направлении контроля):
 - 1 – IDT = 1 (30) (1 бит HEX16/HEX32);
 - 2 – IDT = 7 (33) (значение HEX16/HEX32 с представлением HEX);
 - 4 – IDT = 13 (36) (FLOAT);
 - 8 – зарезервировано;
 - 16 (0x10) – IDT = 15 (37) (значение HEX16/HEX32 с представлением DEC);
- **SP104_BLOCK_OUT_COT3=<число HEX (0 по умолчанию)>** – аналог **SP104_BLOCK_COT3** для каналов OUTPUT;
- **SP104_BLOCK_OUT_COT20=<число HEX (0 по умолчанию)>** – не включать/включать на команду общего опроса каналы OUTPUT (значения ключа – те же, что в **SP104_BLOCK_COT3**);

- **IEC_COUNTER_LIST_ON_START=OFF/ON** – счетчики не включены/включены на команду общего опроса;
- **SP104_IOA_CODE=ON** – **IOA** берется из кодировки;
- **SP104_START_IOA**=<число DEC> – знакоместо в кодировке, где начинается **IOA**;
- **SP104_MIN_IOA**=<число DEC> – минимальное значение **IOA** в проекте, не обязательно вычисляется;
- **SP104_MAX_IOA**=<число DEC> – максимальное значение **IOA** в проекте, не обязательно вычисляется;
- **SP104_USE_103=ON** – разрешение обработки IDT103 синхронизации времени;
- **SP104_USE_103**=<число секунд> – предельное время ожидания получения времени синхронизации;
- **SP104_START_GRP**=<номер группы, DEC> – опрос группы; номер группы считывается из кодировки канала CALL.ChGroupReq.

Каналы, привязанные к аргументам CALL.ChGroupReq, включаются в ответ на команду общего опроса, если в них установлен флаг **Регистратор** (42, **RB**) или аргумент имеет тип, отличный от OUTPUT.

Если в канале CALL.ChGroupReq установлен флаг **Отчет тревог** (43, **EB**), в список заносятся значения каналов с временными метками.

IDT определяется по типу данных первого аргумента канала CALL.ChGroupReq:

- отличен от **DINT** – группа, которая отвечает на IDT=100 (значения в кодировке – 1..16);
- **DINT** – группа счетчиков, которая отвечает на IDT=101 (значения в кодировке – 1..4);

В атрибут (0,**R**) таких каналов CALL.ChGroupReq можно записать следующие значения:

- 0x1010 или 0x1000 – соответственно установка или сброс бита 4 (0x10) **BL** атрибута (61, **Nind**) **QDS** у всех каналов группы (см. **Атрибуты каналов, отображаемые профайлером**);
 - 0xC0, 0x80 или 0x40 – соответственно сброс **R** (**R=0**), заморозка **R** или сброс и заморозка **R** у всех каналов группы (локально).
- **IEC_USE_PN=OFF** – по умолчанию **slave** отвечает с **COT** = 44, 45, 46, 47 и устанавливает **PN** (0x40 в **COT**), то есть байт **COT=0x40** (44, 45, 46, 47). Для отмены установки **PN** используется данный ключ.

- **IEC_SLAVE_KFROM=** – остановка посылки после не подтвержденных блоков.
- **TCP_DISCONN00**=<число секунд, DEC> – если спустя заданное время после **testfr(act)** не получена **testfr(con)**, сокет закрывается.

Ключи для тестирования:

- **IEC_SLAVE_COT10=OFF/ON** – запрет/разрешение COT=10;
- **IEC_SLAVE_COT7=OFF/ON** – запрет/разрешение COT=7;
- **IEC_SLAVE_TESTFR(CON)=ON/OFF** – разрешение/запрет/TESTFR(CON);
- **IEC_SLAVE_FIRST_TESTFR** – после соединения послать TESTFR(act);
- **_SLAVE_CLOSE_SOC** – закрыть сокет;
- **IEC_SLAVE_MANY=COT20** – увеличить на порядок число переменных, передаваемых по команде опроса;
- **IEC_SLAVE_MANY=COT3** – увеличить на порядок число переменных, передаваемых по COT3;
- **IEC_SLAVE_MANY=0** – сбросить увеличение.

Для вычислений используется канал **CALL.Sum** с **Параметр=2**.

Команда синхронизации времени IDT=103

Для обработки IDT=103 в файле *.cnf нужно задать ключ **SP104_USE_103=ON** или создать канал TIME INPUT, в котором установить флаг **Регистратор**. К каналу должен быть привязан источник для IEC104 с IOA=0.

В канал запишется время, которое клиент (HOST) хочет установить ((92,12) – миллисекунды, (91,11) – разница в секундах между устанавливаемым и текущим временем).

Если канал отключен от источника, устанавливаемое время в него запишется, но корректировка времени произведена не будет, и клиенту будет отослано IDT=103 с COT=0x40 | 7.

Поддержка протокола NMEA

MPV поддерживает протокол NMEA-0183.

В узле должен быть создан COM-порт с назначением **GPS** (см. **Редактор параметров COM-порта**). В узле может быть несколько COM-портов с назначением GPS.

Для чтения/управления используются шаблоны **UniversalPLC_type2** или **UniversalPLC_type4** (группа **Источники/Приемники / Пользо-**

вательские драйверы / TYPE11) соответственно типа INPUT/OUTPUT. Шаблон привязывается к каналу CALL.ChGroupReq (далее **cgr**).

Для приема данных от GPS-приемника могут использоваться переменные **@t_Set_Time** и **@t_Set_Date** (см. раздел **Группа СИСТЕМНЫЕ**).

Чтение

Формат ответа прибора:

```
$<talking_ID, 2 байта><sentence, 3 байта><sentence_ID, 3 байта>, <данные>* <контр. сумма, 2 байта><CR><LF>
```

MPB записывает данные из ответа в соответствующий буфер, **cgr** читает свой буфер (для этого в строке **Дополнительно** шаблона указывается **sentence_ID**) и записывает данные в аргументы. MPB распознает следующие **sentence_ID**: ZDA, RMC, RMA, VHW, DPT, XDR, R\$A, R\$D, GGA, GLL, HDG, HDT, MTW, MWV, RMB и G\$A, однако буфер создается и для неизвестного **sentence_ID** (таких буферов в узле может быть создано не более 8).

Значение, заданное в строке **Дополнительно**, записывается в атрибут (47, **IDstr**) **cgr**; и это значение может быть изменено в реальном времени, что обеспечивает настройку канала на чтение ответа с другим **sentence_ID**.

Интерпретация данных зависит от типа данных аргумента; например, для записи в аргумент REAL значение преобразуется в формат float.

Допустимые типы данных аргумента: REAL, STRING (буква, содержащаяся в данных), SINT (ASCII-код буквы, содержащейся в данных), DINT (целочисленное значение), UDINT (значение в формате HEX), DATE (дата), DATE_AND_TIME (дата и время без миллисекунд) и TIME (время без миллисекунд).

Для приема времени с десятками и сотнями миллисекунд аргумент должен быть REAL, а четвертый байт удаленного адреса (C4) шаблона должен быть равен 1. В этом случае, например, значение аргумента 71749.00 означает 7ч 17мин. 49с 00мс.

Время прихода ответа записывается в атрибут (45, **T**) **cgr**.

Если **@Debug & 0x200 = 1**, при приеме ответа с неизвестным **sentence_ID** в отладочный файл профайлера записывается следующее сообщение:

```
UNKNOWN SENTENCE <принятая строка>
```

В **сgr** устанавливается признак аппаратной недостоверности, если:

- в **сgr** установлен флаг **Запрос времени значения**;
- ответ считывается повторно (не в первый раз);
- время ответа более чем на 1 мин. меньше текущего.

Чтение параметров цели

Если прибор (радар) передает параметры многих целей, то запись в **сgr** параметров нужной цели конфигурируется следующим образом:

- в строке **Дополнительно** шаблона указывается **TTM**;
- (91, I1) = <номер цели> (цели нумеруются с 1) или (80, CMNT) = <имя цели>.

Если номер цели задан нулевым, **сgr** читает (80, CMNT), среди всех TTM-предложений выбирает предложение, соответствующее заданной цели, и записывает данные в аргументы. При этом в атрибут (0, R) **сgr** записывается номер цели.

Управление

Формат строки **Дополнительно** шаблона типа OUTPUT:

```
$<talking_ID, 2 байта><listener_ID, 2 байта>Q,<sentence_ID, 3 байта><данные (если требуется)>
```

MPV дополняет эту строку следующим образом и посылает в прибор:

```
*<контр. сумма, 2 байта><CR><LF>
```

Для управления NMEA-совместимым прибором, имеющим собственные коды, строка **Дополнительно** задается в следующем виде:

```
$P<ID производителя, 3 байта><параметры в соответствии с описанием прибора>
```

Например, следующая строка является командой остановки передачи для приборов GARMIN:

```
$PGRMO, PGRMT, 0
```

Управление и диагностика обмена

В данном разделе описывается управление и диагностика обмена по встроенным протоколам TRACE MODE 6.

Если в комментарии канала задана строка «**error**» (любыми буквами), то канал превращается в 15.73, при этом в **C5** записывается номер протокола, а в **I0** – адрес устройства.

Следующие каналы, созданные в ИС, преобразуются в MPV, если в комментарии канала задано MODBUS, IEC104, SNMP, OMRON, USER UDP,

USER TCP или MITSUBISHI:

- 15.13 (@e_TCP_ModBus) => 15.73;
- 15.9 (@e_PLC2Type) => 15.73.

Для числового канала:

- во входе статус только для INPUT, для OUTPUT – код управления устройством (сбрасывается в 0 после исполнения);
- **I1** – код ошибки от устройства с добавлением 0x8000, если нет связи со 2-м адресом (команда **ping** не проходит);

В конфигурации устройства может быть задан ключ **AUTO_ON**, запускающий регулярную проверку связи с обоими сетевыми адресами с помощью команды **ping** (только для INPUT).

- (*) кодировка – текущий IP-адрес;
- (*) комментарий – текстовый статус устройства (с учетом **AUTO_ON**);
- 116.x **I2** – код сетевой ошибки;
- 15.73 **C4** – счетчик любых обменов с устройством (не более 255);
- (*) изменение кодировки на **?xxx.xxx.xxx.xxx** приводит к поиску устройства с этим адресом сначала по заданному протоколу, а потом по всем протоколам;
- (*) изменение кодировки на **=xxx.xxx.xxx.xxx** приводит к изменению IP-адреса устройства;
- если в канале установлен флаг запроса времени значения (атрибут 50), то в атрибут 45, **T** записывается последнее время обмена.
(*) Если **ALR_COMMENT_ON** (без звездочки – если **ALR_COMMENT_OFF**).

Источник может быть привязан к каналу CALL.ChGroupReq (номер протокола записывается в **C5**):

- **Параметр=1,2** – число аргументов равно удвоенному числу устройств с первым IP адресом, отличным от нуля. На устройство – 2 аргумента: номер, время **ping** или код ошибки;
- **Параметр=3,4** – число аргументов равно удвоенному числу устройств со вторым IP адресом, отличным от нуля. На устройство – 2 аргумента: номер, время **ping** или код ошибки;
- **Параметр=5,6** – число аргументов равно максимальному числу устройств. На устройство – 1 аргумент: если команда **ping** проходит на второй адрес, в аргумент записывается время **ping** или код ошибки, в противном случае – 1;
- **Параметр=7,8** – число аргументов равно максимальному числу устройств. На устройство – 1 аргумент: если команда **ping** прохо-

дит на первый адрес, в аргумент записывается время **ping** или код ошибки, в противном случае – 1;

- **Параметр=9,10** – число аргументов равно равно максимальному числу устройств. На устройство – 1 аргумент:
 - 1 – команда **ping** не проходит на первый адрес;
 - 2 – команда **ping** не проходит на второй адрес.

Для контроля и управления обменом по встроенным протоколам предусмотрены следующие ключи в файле *.cnf:

- **DISABLEUNITTCP** – запрет запуска обмена по встроенным протоколам;
- **IEC_KFROM=<число>** – для протокола 104, узел SLAVE. Ключ задает число фреймов ожидания подтверждения обмена (по истечении обмен останавливается);
- **IEC_WFROM=<число>** – то же, что и **IEC_KFROM**, для узла MASTER;
- **OEMTUSE= <>** – биты конфигурации SLAVE-протоколов;
- **SLAVE_UDPPORT=** – порт для сокета узла SLAVE (эквивалентен ключу **SLAVE_TCSPORT** – см. **MPB как сервер протоколов поверх TCP/IP**);
- **SLAVE_UDPADDR=** – IP-адрес для сокета узла SLAVE (эквивалентен ключу **SLAVE_TCPADDR** – см. **MPB как сервер протоколов поверх TCP/IP**);
- **SLVPCOM=** – SLAVE-протокол, подобный протоколу 101;
- **HOSTTCP=** – маска включения TCP-протоколов на хосте;
- **SETCARDHOST=1** и **SETCARDHOST=2** – задание сетевого адаптера; только для обмена по произвольному протоколу (29) или OMRON (30) по UDP.
- **TCP_DIFCONN<nn>=<число секунд>** – таймаут переподключения. В случае ошибки соединения попытка подключения предпринимается спустя время, заданное данным ключом (значение по умолчанию – 30с, **nn** – номер протокола);
- **TCP_DISCONN<nn>=<число секунд>** – таймаут разрыва соединения. Если в ходе нормального обмена с устройством наступает момент, когда не проходит посылка, то сокет не уничтожается, а в течение времени, заданного данным ключом, примерно 1 раз в 10с предпринимается попытка повторной отправки посылки. Если отправить посылку не удастся, по истечении таймаута сокет уничтожается. Значение данного таймаута по умолчанию – 0, что соответствует уничтожению сокета через 1 минуту;

Обмен через драйверы

Шаблоны каналов обмена с PLC через драйверы создаются в группе **PLC** слоя **Источники/Приемники** (см. **Назначение групп источников (приемников) и Группа 'PLC'**).

Драйверы, подключенные к TRACE MODE, перечислены в файле **%Trace Mode%\Drivers_with_Setup\drivers.txt**.

Обмен с контроллером ЛОМИКОНТ

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данным устройством задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Тип переменной** – тип переменной для чтения/записи. Тип выбирается из списка, содержащего названия переменных в соответствии с документацией на контроллер. Если тип переменной начинается с символов <>, то переменная будет управляться данным каналом в режиме пульта. При этом значение канала задает следующие действия: 0 – увеличить, 1- уменьшить, 2 – увеличить быстро, 3 – уменьшить быстро;
- **Группа** – номер группы, в которую входит данная переменная;
- **Переменная** – номер переменной для чтения/записи.

При обмене с контроллером МРВ по возможности реализует групповые запросы. Если запрашиваются дискретные переменные, то они всегда записываются по 8 в канал (одна группа). При управлении дискретными переменными каждый канал формирует значение только одной переменной.

Формат записи и приема в каналы переменных типа ТМ (значения таймеров контроллера):

- при запросе полученное значение записывается в секундах,
- при передаче в контроллер число записывается шестью десятичными разрядами: 2 старших - секунды, 2 следующих - минуты, 2 младших - часы (для миллисекундного таймера – соответственно десятые доли секунд, секунды, минуты).

Для обмена необходимо настроить последовательный порт (см. **Редактор параметров СОМ-порта**).

Обмен с контроллером РЕМИКОНТ-130

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данным устройством задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – СОМ1, ..., 31 – СОМ32). Этот параметр задается в формате HEX;
- **Модуль** – номер контроллера в сети ТРАНЗИТ;
- **Переменная** – тип переменной для чтения/записи (см. ниже);
- **Алгоблок** – номер алгоблока в пользовательской программе контроллера для переменных ВХ и ВЫХ; для переменных УПР, РУЧ, УВ и РЕГ – номер контура регулирования;
- **Номер ввода/вывода** – номер запрашиваемой или управляемой переменной для переменных СЕТЬ, УПР, ВХ и ВЫХ. Номера индексов каналов;
- **Описатель** – вид представления переменной.

Тип переменной выбирается из следующего списка:

СЕТЬ – опрос сетевых переменных. Для данного типа команды в настройке **Номер ввода/вывода** задается номер контролируемой переменной;

КОНФ – опрос конфигурации сети (16-битовое целое число, каждый бит которого является идентификатором наличия в сети контроллера с соответствующим номером);

ВХ – вход алгоблока;

ВЫХ – выход алгоблока;

УПР – опрос состояния контура. Для данного типа команды в настройке **Алгоблок** указывается номер контура, а величина настройки **Номер ввода/вывода** определяет контролируемую характеристику контура:

- 1 – величина ручного задания;
- 2 – текущее значение задания;
- 3 – вход регулятора;
- 4 – рассогласование;
- 5 – выход регулятора;
- 6 – режим работы контура:

13,14 биты:

00 – 3ДЛ отсутствует;

01 – локальный режим;

10 – каскадный режим;

1 бит – признак ошибки контура;

2 бит – наличие алгоритма РУЧ;

3 бит:

0 – локальный или каскадный;

1 – дистанционный режим;

4 бит:

0 – автоматический режим;

1 – ручной режим;

5,6 биты:

00 – ЗДН отсутствует;

01 – внешнее задание;

10 – программное задание;

11 – ручное задание;

РУЧ – управление значением ручного задания;

У_В – управление выходным сигналом;

РЕГ – управление режимом контура. Режим задается значением канала:

- 07 – установка режима ВЗ;
- 09 – установка режима ПЗ;
- 0В – установка режима РЗ;
- 0D – переход на автоматический режим;
- 0F – установка режима РУ;
- 11 – установка режима ДУ;
- 13 – отмена дистанционного управления;
- 15 – установка режима КУ;
- 17 – установка режима ЛУ.

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров СОМ-порта**), задав скорость обмена 4800 и контроль – 8-2-е.

Обмен с контроллерами Ш711

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данным устройством задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Канал** – номер измерительного канала в преобразователе (от 1 до 60 – для Ш711/1, от 1 до 80 – для Ш711/1.1).

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров COM-порта**).

Обмен с контроллером Ш9327

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данным устройством задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 - COM1, ..., 31 - COM32). Этот параметр задается в HEX формате;
- **Адрес** – сетевой номер преобразователя;
- **Параметр** – номер переменной, начиная с 1. Канал с атрибутом **Параметр=0** является инициатором обмена – запросы выполняются с его периодом, но в него никакие данные не записываются;
- **Команда** – выполняемое действие, которое выбирается из следующего списка:
 - **AI** – опрос аналогового входа;
 - **RL** – если канал типа INPUT, то запрос значений 8 дискретных выходов. Если канал имеет тип OUTPUT, то он формирует значение 8 дискретных выходов в буфере для передачи в МИП;
 - **AO** – если канал типа INPUT, то запрос аналогового выхода. Если канал имеет тип OUTPUT, то он формирует значение соответствующего выхода в буфере для передачи в МИП;
 - **AIP** – код признаков неисправности данного канала или результат сравнения с уставками:
 - 0 - канал исправен, результат в пределах уставок;
 - 1 - "НДАТ", неисправность датчика;
 - 2 - "НМИП", неисправность МИП;
 - 4 - "ПЕРЕГР", перегрузка АЦП, т.е. значение входного сигнала превышает диапазон АЦП;
 - 5 - "<<", результат ниже нижнего предела измерений датчика;
 - 6 - ">>", результат выше верхнего предела измерений датчика;
 - 7 - "<>", результат не в допуске (для служебных каналов);
 - 9 - "<", результат ниже уставки;
 - 10 - ">", результат выше уставки;
 - **AIK** – код результата опроса датчика типа «сухой контакт»:
 - 1 - «ЗАМКН», контакт замкнут;
 - 2 - «РАЗОМ», контакт разомкнут;
 - 0 - датчик не запрограммирован;
 - **wRL** – отработка этого канала запускает передачу в МИП

- буфера со значениями дискретных выходов;
- **wAO** – отработка этого канала запускает передачу в МИП буфера со значениями аналоговых выходов;
 - **AlpEx** – дополнительный код признаков неисправности данного канала или результата сравнения с уставками. Принимает значение 2 в степени **Alp**.

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров СОМ-порта**).

Обмен по протоколу HART

Драйвер обмена с устройствами по протоколу HART выполнен как драйвер t11.

Для последовательного интерфейса должны быть заданы следующие настройки (см. **Редактор параметров COM-порта**):

- скорость передачи – 1200 бод;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – проверка на нечетность.

Рекомендуемое значение таймаута ожидания ответа – 1-2с, управление передатчиком – **HART_ER**.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер COM-порта, уменьшенный на 1;
- **Устройство** – номер прибора (DEC), используется в начальном запросе ID прибора. В TRACE MODE предусмотрено 2 варианта запроса ID прибора с одновременным заданием типа опрашиваемого устройства **Master_SCADA** (система управления):
 - **Устройство=0...15**;
 - **Устройство=0...15+n*16** (n=1,2,3) с одновременным указанием переменной в атрибуте **Комментарий** в следующем формате:

HART=<имя переменной>

Номера **Устройство=128...143** (0...15, увеличенные на 128) задают тип опрашиваемого устройства **Handheld_Terminal** (ручной коммуникатор). Тип опрашиваемого устройства влияет на формирование заголовка запроса;

- **Запрос** – тип запроса к прибору, значение выбирается из меню:
 - **ManufacturerCode** – код изготовителя;
 - **DeviceTypeCode** – код типа устройства;
 - **UniversalCommVer** – версия универсальных команд;
 - **SpecificCommVer** – версия специфических команд;
 - **SoftwareVer** – версия программного обеспечения;
 - **HardwareVer** – версия аппаратного обеспечения;
 - **DeviceFlags** – флаги функций устройства;
 - **DeviceID** – идентификационный номер устройства;

- **PVUnitsCode** – код единиц измерения первичной переменной;
- **PV** – первичная переменная;
- **Current** – ток (mA);
- **Range Percentage** – процент диапазона;
- **SensorSerialNum** – серийный номер датчика;
- **SensorUnitsCode** – код единиц измерения датчика;
- **SensorUpperLimit** – верхний предел измерения датчика;
- **SensorLowerLimit** – нижний предел измерения датчика;
- **SensorMinSpan** – минимальный интервал датчика;
- **DynCurrent** – блок динамических переменных: ток (mA);
- **DynPVUnitsCode** – блок динамических переменных: код единиц измерения первичного параметра;
- **DynPV** – блок динамических переменных: первичный параметр;
- **DynSVUnitsCode** – блок динамических переменных: код единиц измерения второго параметра;
- **DynSV** – блок динамических переменных: второй параметр;
- **DynTVUnitsCode** – блок динамических переменных: код единиц измерения третьего параметра;
- **DynTV** – блок динамических переменных: третий параметр;
- **DynFVUnitsCode** – блок динамических переменных: код единиц измерения четвертого параметра;
- **DynFV** – блок динамических переменных: четвертый параметр;
- **(12)Message**
- **(C,60*)AO_Level**
- **(C,60*)AO_Percent**
- **(61)PVAnalogOutputLevel**
- **(61)AO_PV_Variable**
- **(61)AO_SV_Variable**
- **(61)AO_TV_Variable**
- **(61)AO_FV_Variable**
- **(110)PV_Variable**
- **(110)SV_Variable**
- **(110)TV_Variable**
- **(110)FV_Variable**
- **(15)AlarmSelectCode**
- **(15)TFC**

- **(15)PVRangeCode**
- **(15)Upper-range**
- **(15)Lower-range**
- **(15,W34)Dumping Value**
- **(C,33*)V_Slot**
- **(C,54*)V_Transmitted**
- **(C,63*)_R_AO_Inf**
- **R_Unicum**
- **R_Tag**
- **(Wc3)Write_Byte** – запись в предустановленную переменную, атрибут (96,**C3**) канала OUTPUT задает код команды;
- **(Wc3)Write_Short** – запись в предустановленную переменную, атрибут (96,**C3**) канала OUTPUT задает код команды;
- **(Wc3)Write_Long** – запись в предустановленную переменную, атрибут (96,**C3**) канала OUTPUT задает код команды;
- **(Wc3)Write_Float** – запись в предустановленную переменную, атрибут (96,**C3**) канала OUTPUT задает код команды;
- **(Wvalue*)Func_0** – тип канала – OUTPUT, атрибут (96,**C3**) задает код команды, в качестве данных используется значение канала (1 байт);
- **(Wvalue)Func_1** – тип канала – OUTPUT, атрибут (96,**C3**) задает код служебной команды, данные не передаются. Запрос обрабатывается при изменении значения канала;
- **Status** – канал INPUT считывает байт статуса из ответа прибора. Канал OUTPUT управляет обменом с заданным устройством:
 - (9,**Q**)=3, 5, 7 – отмена опроса;
 - (9,**Q**)=16 – отмена опроса с последующей реинициализацией обмена;
 - (9,**Q**)=2 – разрешение обмена;
- **Главное устройство** – не используется.

Шаблон канала обмена может быть привязан к каналу CALL.ChGroupReq.

Назначение символов, указываемых в начале запроса в круглых скобках:

- **C** – шаблон канала обмена должен быть привязан к каналу CALL.ChGroupReq;
- число – код команды HART;
- **W** – допускается запись;
- ***** – при выполнении запроса используется атрибут (96, **C3**).

Обмен по протоколу COMM2H

Данный протокол используется для обмена данными с контроллерами HITACHI, Sprecher&Schuh, Samsung.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Модуль** – номер модуля в формате HEX (0-FF);
- **Адрес** – десятичный адрес переменной (диапазон соответствует протоколу обмена);
- **Канал** – тип переменной (соответствует протоколу обмена).

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров COM-порта**).

Обмен по протоколам SNP, SNP-X

Данные протоколы используются для обмена данными с контроллерами Cegelec, GE Fanuc.

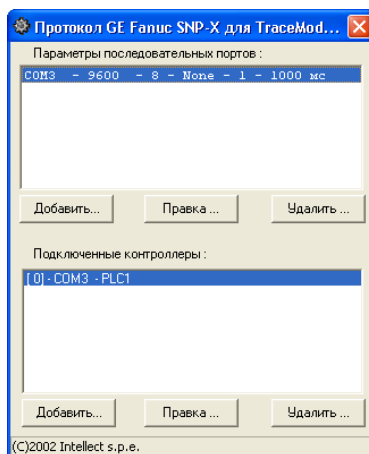
Драйвер обеспечивает обмен данными с ПЛК серии 90 фирмы GE Fanuc. Драйвер поддерживает чтение регистров **I, Q, G, M, SA, SB, SC, T, S, R** и **AQ** и запись регистров **I, Q, G, M, T, R, AI** и **AQ**. Для обмена данными используется последовательный порт компьютера. Драйвер оформлен в виде драйвера t12. Для корректной работы драйвера нужно правильно произвести конфигурацию порта в ПЛК и назначить контроллеру уникальный идентификатор **SNP-ID**. Протокол **SNP-X** является подразделом протокола **SNP** (Series Ninety Protocol).

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Последовательные порты, используемые при обмене, должны быть занесены в список **SNPX.cfg** с указанием **SNP-ID** тех ПЛК, с которыми будет осуществляться связь. Для этого нужно воспользоваться утилитой **SNPXconfig.exe**. Каждый порт в списке будет автоматически открыт драйвером.

Порт, к которому подключены устройства SNP-X, не нужно настраивать в TRACE MODE.

Утилита **SNPXconfig.exe** предназначена для создания списка портов, которые могут использоваться при обмене. Список сохраняется в файл **SNPX.cfg** в папке узла.

Основное окно программы имеет следующий вид:

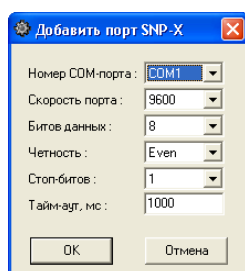


Каждая строка списка портов в основном окне утилиты состоит из шести параметров:

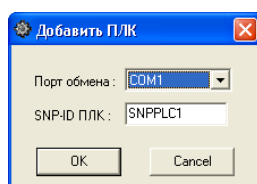
- номер порта;
- скорость передачи данных (300-115200 bps), значение по умолчанию – 9600;
- число битов данных (8 по умолчанию);
- контроль четности передачи, может принимать значения **None**, **Odd**, **Even**, Значение по умолчанию – **Even**;
- количество стоп-битов (1 или 2). Значение по умолчанию – 1.
- таймаут. Если в течение этого времени ПЛК не возвращает правильный ответ, каналу выставляется признак недостоверности. Значение этого параметра зависит от скорости передачи данных (обычно 500-1000 мс).

Каждому запросу по сети **SNP-X** предшествует посылка команды **Attach** (подключение к ПЛК), поэтому обработка каждого запроса занимает до 100-200 мс.

Для создания новой записи в списке COM-портов нажмите кнопку **Добавить** в верхней части основного окна утилиты, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



Нижняя половина окна конфигуратора позволяет редактировать список подключенных ПЛК и их идентификаторы. В данный список должны быть занесены все ПЛК, с которыми может осуществляться связь. Для создания новой записи в списке ПЛК нажмите кнопку **Добавить**, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



В окне добавления нового ПЛК нужно указать номер последовательного порта, к которому подключен контроллер, и идентификатор **SNP-ID** контроллера. При этом нужно следить, чтобы указанный COM-порт присут-

ствовал в списке портов. Идентификатор должен быть уникальным для каждого контроллера; он устанавливается в системе программирования ПЛК (Versa Pro для ПЛК GE Fanuc).

Если при запуске утилиты файл **SNPX.cfg** уже существует, его содержимое после первичного анализа загружается в список портов. Процедура анализа отбирает все записи, которые имеют правильный номер порта. Если несколько записей имеют одинаковый номер порта, то первая из них (самая верхняя) появится в списке зарегистрированных портов, а остальные будут проигнорированы и удалены. Список зарегистрированных портов сохраняется при каждом выходе из утилиты, список проигнорированных портов не сохраняется.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **SNP идентификатор** – номер ПЛК, по которому производится обмен для этого канала. Введенное здесь число должно соответствовать номеру ПЛК в списке контроллеров в утилите **SNPXconfig**. Если в списке утилиты нет порта с заданным номером или порт настроен в TRACE MODE как порт для обмена любыми данными, то канал будет отключен при запуске системы;
- **Тип регистра** – тип адресуемого регистра, выбирается из следующего списка:
 - **(I)DInputs(R/W)** – элементы данных типа **Input**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **(Q)DOutputs(R/W)** – элементы данных типа **Output**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **(M)InCoils(R/W)** – элементы данных типа **Internal**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **(T)TempCoils(R/W)** – элементы данных типа **Temporary**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **(G)DGlobals(R/W)** – элементы данных типа **Global**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **(S)StatusCoils(R)** – элементы данных типа **Status**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **(SA)Descretes(R)** – элементы данных типа **StatusDiscrete**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **(SB)Descretes(R)** – элементы данных типа **StatusDiscrete**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **(SC)Descretes(R)** – элементы данных типа **StatusDiscrete**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **(R)Registers(R/W)** – элементы данных типа **Register**, 16 бит/регистр, чтение/запись;

- **(AI)AnalogIn(R/W)** – элементы данных типа **AnalogIn**, 16 бит/регистр, чтение/запись;
- **(AQ)AnalogOut(R/W)** – элементы данных типа **AnalogOut**, 16 бит/регистр, чтение/запись.
- **Номер регистра** – адрес выбранного регистра в массиве регистров ПЛК. Значение параметра нужно вводить в десятичном виде, начиная с 0. Для регистров, чтение которых невозможно, флаг **Достоверность** устанавливается в 1.

Каналы для всех типов регистров могут иметь тип **I**, т.е. использоваться для чтения. Для записи могут быть применены каналы с типом регистра **I, Q, G, M, T, R, AI, AQ** (ограничение стандарта). Каналы типа **O** с типом регистра, отличным от приведенных выше, будут отключены при запуске.

Элементы с типом регистра **R, AI, AQ** имеют размерность 2 байта (1 слово), поэтому они читаются/пишутся по одному элементу в канал. Нумерация начинается с 0 и указывается в десятичной форме (0 соответствует R1, 1 – R2 и т.д.).

Элементы **I, Q, S, SA, SB, SC, G, M, T** имеют размерность 1 бит, поэтому читаются/пишутся по 16 элементов в 1 канал. Нумерация начинается с 0 и указывается в десятичном виде, значение начального адреса указывает на номер байта (пакет из 8 регистров). Таким образом, если указать **RegNum=0**, то канал будет содержать в себе элементы 1(мл. бит) – 16(ст. бит), а если указать **RegNum=1**, то канал будет содержать в себе элементы 9(мл. бит) – 24(ст. бит).

Если регистры привязаны в ПЛК к реальным входам, то команда записи в них не будет отработана на уровне ПЛК.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. При каждой отправке ответ ожидается в течение 1 с. Если ответа за это время нет, то каналу выставляется признак недостоверности, связь с ПЛК разрывается и предпринимается попытка восстановить ее заново. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Если восстановления связи не произошло, следующая попытка установки связи будет при следующем обращении к этому порту. Необходимо обратить внимание, что при безуспешной попытке записи значения канала TRACE MODE будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

Обмен с контроллерами Mitsubishi и Delta

Обмен по RS

Драйвер обеспечивает обмен данными, используя транспортный протокол **FXComputerLink** и реализацию протокола **Protocol Format 1** для обмена с ПЛК серии FX фирмы MITSUBISHI. Драйвер поддерживает чтение регистров **X, Y, M, S, TN, CN** 16 и 32 бит, **TS, CS, D** и запись регистров **X, Y, M, S, TN, D** и **CN** 16 бит. Для обмена данными используется последовательный порт компьютера. Драйвер оформлен в виде драйвера t12. Для корректной работы драйвера нужно правильно произвести конфигурацию порта в ПЛК (см. ниже). Протокол **ComputerLink Format 1** поддерживается в ПЛК серий **FX, FX0N, FX1N, FX1S, FX2C, FX2N, FX2NC**.

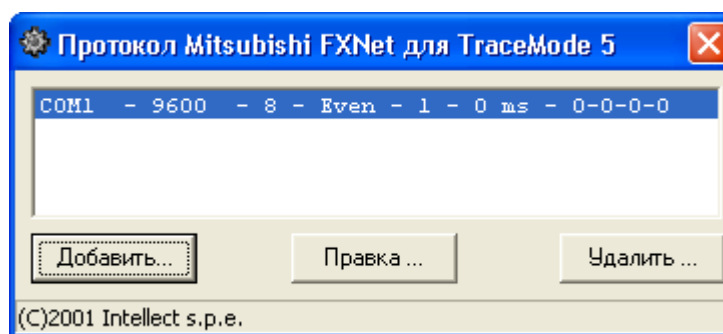
Для функционирования драйвера необходимо установить и сконфигурировать коммуникационный адаптер серий **232-BD, 232-ADP, 485-BD** или **485-ADP**. О поддержке протокола в других сериях ПЛК проконсультируйтесь с поставщиком этого ПЛК.

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Последовательные порты, используемые при обмене, должны быть занесены в список **FXNet.cfg**. Для этого нужно воспользоваться утилитой **FXconfig.exe**. Каждый порт в списке будет автоматически открыт драйвером.

Порт, к которому подключены устройства FXNet, не нужно настраивать в TRCAE MODE.

Утилита **FXconfig.exe** предназначена для создания списка портов, которые могут использоваться при обмене. Список сохраняется в файл **FXNet.cfg**.

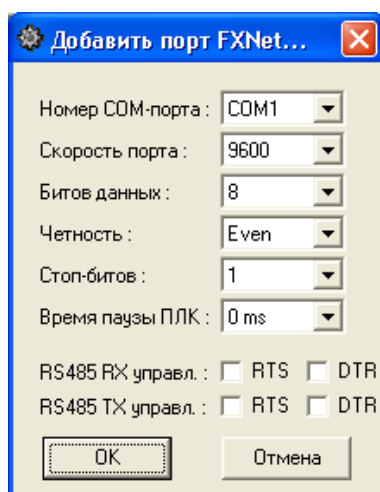
Основное окно программы имеет следующий вид:



Каждая строка списка портов в основном окне утилиты состоит из семи параметров:

- номер порта. Этот параметр, уменьшенный на 1, затем нужно указать в качестве настройки **Порт** канала, который связывается с данным портом;
- скорость передачи данных (300-115200 bps), значение по умолчанию – 9600;
- число битов данных (8 по умолчанию);
- контроль четности передачи, может принимать значения **None**, **Odd**, **Even**, значение по умолчанию – **Even**;
- количество стоп-битов (1 или 2). Значение по умолчанию – 1;
- время переключения конвертора RS485. Этот параметр передается в ПЛК, который задерживает передачу ответа на указанное время с тем, чтобы конвертор успел переключиться в режим приема. Этот параметр лежит в диапазоне 0-150 мс, по умолчанию – 0 мс;
- параметры **RTS** и **DTR** для приема и передачи. Если конвертор интерфейсов требует переключения сигналов для индикации приема и передачи данных, то необходимо указать, какие сигналы и как должны работать при приеме (**RX**) и при передаче (**TX**).

Для создания новой записи нажмите кнопку **Добавить**, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



Если при запуске утилиты файл **FXNet.cfg** уже существует, его содержимое после первичного анализа загружается в список портов. Процедура анализа отбирает все записи, которые имеют правильный номер порта. Если несколько записей имеют одинаковый номер порта, то первая из них (самая верхняя) появится в списке зарегистрированных портов, а остальные будут проигнорированы и удалены. Список зарегистрированных портов сохраняется при каждом выходе из утилиты, список проигнорированных портов не сохраняется.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер порта (0 – COM1, ... 31 – COM32). Настройки порта должны быть предварительно заданы в утилите **FXconfig**. Если в списке утилиты нет порта с заданным номером или порт настроен в TRACE MODE как порт для обмена любыми данными, то канал будет отключен при запуске системы;
- **Номер станции** – адрес ПЛК («станции» в терминологии MITSUBISHI), с которым обменивается данный канал. Адрес лежит в диапазоне 0-15 и выбирается из списка;
- **Номер регистра** – адрес выбранного регистра в массиве регистров ПЛК (в десятичном виде);
- **Тип регистра** – тип адресуемого регистра. Тип регистра выбирается из следующего списка:
 - **[X]Inputs(I/O)** – элементы данных типа **Input**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **[Y]Outputs(I/O)** – элементы данных типа **Output**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **[M]AuxRelays(I/O)** – элементы данных типа **AuxiliaryRelay/SpecialAuxiliaryRelays**, 1 бит/регистр, запись/чтение блоками по 16 элементов;
 - **[S]States(I/O)** –
 - **[TS]TimerContacts(I)** – элементы данных типа **TimerContacts**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **[CS]CounterContacts(I)** – элементы данных типа **CounterContacts**, 1 бит/регистр, только чтение блоками по 16 элементов;
 - **[TN]TimerValue(I/O)** – элементы данных типа **TimerValue**, 16 бит/регистр, запись/чтение;
 - **[CN]Counter16bit(I/O)** – элементы данных типа **CounterValue** в диапазоне C0-C198, 16 бит на регистр, запись/чтение;
 - **[CN]Counter32bitHi(I)** – элементы данных типа **CounterValue** в диапазоне C199-C255, старший байт, 32 бита на регистр, только чтение старшего байта регистра;
 - **[CN]Counter32bitLo(I)** – элементы данных типа **CounterValue** в диапазоне C199-C255, младший байт, 32 бита на регистр, только чтение младшего байта регистра;
 - **(D)Data/FileRegisters(I/O)** – элементы данных типа **Data, File, RAMFile** и **SpecialData**, 16 бит/регистр, чтение/запись.

Каналы для всех типов регистров могут иметь тип **I**, т.е. использоваться

для чтения. Для записи могут быть применены каналы с типом регистра **X, Y, M, S, TN, CN16, D** (ограничение протокола). Каналы типа **O** с типом регистра, отличным от приведенных выше, будут отключены при запуске.

Элементы с типом регистра **CN16, TN, D** имеют размерность 2 байта (1 слово), поэтому они читаются/пишутся по одному элементу в канал. Элементы **X, Y, S, TS, CS, M** имеют размерность 1 бит, поэтому читаются/пишутся по 16 элементов в 1 канал.

Таким образом, если указать **RegNum=0**, то канал будет содержать в себе элементы 0-15, а если указать **RegNum=1**, то канал будет содержать в себе элементы 16-31.

Регистры **CN32** имеют размерность 4 байта, поэтому в TRACE MODE для их чтения требуются 2 канала. В один считывается значение старшего слова, в другой – младшего (тип данных **CN32Hi** и **CN32Lo** соответственно). По этой причине запись в регистры такого типа невозможна. Если регистры привязаны в ПЛК к реальным входам, то команда записи в них не будет отработана на уровне ПЛК.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. При каждой посылке ответ ожидается в течение 1 с. Если ответа за это время нет, то каналу выставляется признак недостоверности, связь с ПЛК разрывается и предпринимается попытка восстановить ее заново. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Если восстановления связи не произошло, следующая попытка установки связи будет при следующем обращении к этому порту. Необходимо обратить внимание, что при безуспешной попытке записи значения канала TRACE MODE будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

Обмен по протоколу Modbus RTU с контроллером DELTA фирмы Electronics конфигурируется следующим образом:

- в узле необходимо создать и настроить нужный COM-порт;
- в строке **Дополнительно** нужно указать «**Delta**».

Для связи с контроллерами DELTA нельзя использовать COM-порты, которые используются драйверами t12 для связи с другими контроллерами по последовательному интерфейсу.

Информация о программе в ПЛК

Сведения, изложенные в этом разделе, являются частью инструкции по установке, программированию и наладке сетей передачи данных на базе ПЛК Mitsubishi FX, «**mitsubishi MELSEC-F, User's Manual, FX Communications (RS-232C, RS485)**», JY992D69901-C, 25 Jan 2001. Все изло-

женные данные не могут претендовать на полноту и точность изложения. Пожалуйста, обращайтесь за более подробной информацией к службе техподдержки фирмы Mitsubishi и соответствующим руководствам пользователя.

Для связи с ПЛК драйвер поддерживает протокол передачи **Computer Link**, поэтому проверьте, какой версии Ваш ПЛК и поддерживает ли он этот протокол:

Items		FX2N, FX2NC	FX1N, FX1S	FX0N	FX, FX2c
N:N network		All versions	All versions	V2.00 or more	No sport
Parallel link				All versions	All versions
Computer link				V1.20 or more	V3.30 or more
No protocol communication	Use RS instruction Use FX2N-232IF			All versions	V3.00 or more
Not supported.					

Для обеспечения правильной передачи данных от компьютера к ПЛК не нужно описывать процедуры передачи данных в программе ПЛК, однако необходимо произвести настройку порта и инициализацию протокола **Computer Link**. Для этого используются регистры D8120 – D8129.

Special data registers	Description
D8120	Communication format (RS instruction, computer link)
D8121	Station No. setting (computer link)
D8122	Number of remaining data to be transmitted (RS instruction)
D8123	Number of receive data (RS instruction)
D8124	Data header <Initial value: STX> (RS instruction)
D8125	Data terminator <Initial value: ETX> (RS instruction)
D8127	On-demand head device register (computer link)
D8128	On-demand data length register (computer link)
D8129	Data network Time-out timer value (RS instruction, computer link)

Наиболее важным является регистр D8120, в котором необходимо указать физические параметры передачи данных и выбрать протокол **Computer Link**. Пример фрагмента программы приведен ниже (подробнее см. руководство ПЛК):

```

M8002 ┌──┴──┐ [ MOV H6086 D8120 ]
      └──┬──┘
          b15      b0
D8120 = [ 0110 1000 1000 0110 ]
          6      8      8      6

```

Data length	7 bits
Parity	Even
Stop bit	1 bit
Baud rate	9600 bps
Protocol	Link
Header	No used
Terminator	No used
Control line	Normal mode 1

Помните, что должен быть указан протокол **Computer Link** и формат протокола **Protocol Format 1** (регистр должен иметь вид 0x60xx, младший байт должен соответствовать выбранным параметрам передачи). После изменения параметров ПЛК должен быть выключен и снова включен.

Обмен данными будет работать только при режиме **Run** контроллера.

Регистр D8120 имеет следующий формат:

Bit No.	Name	Description	
		0 (bit = OFF)	1 (bit = ON)
b0	Data length	7 bit	8 bit
b1 b2	Parity	(b2, b1) (0, 0) : None (0, 1) : Odd (1, 1) : Even	
b3	Stop bit	1 bit	2 bit
b4 b5 b6 b7	Baud rate(bps)	(b7, b6, b5, b4) (0, 0, 1, 1) : 300 (0, 1, 1, 1) : 4,800 (0, 1, 0, 0) : 600 (1, 0, 0, 0) : 9,600 (0, 1, 0, 1) : 1,200 (1, 0, 0, 1) : 19,200 (0, 1, 1, 0) : 2,400	
b8*1	Header	None	Effective (D8124) Default : STX (02H)
b9*1	Terminator	None	Effective (D8125) Default : ETX (03H)
b10 b11 b12	Control line	No protocol (b12, b11, b10) (0, 0, 0) : No use <RS232C interface> (0, 0, 1) : Terminal mode <RS232C interface> (0, 1, 0) : Interlink mode <RS232C interface > (FX2N V2.00 or more) (0, 1, 1) : Normal mode 1 <RS232C interface>. <RS485 (422) interface>*3 (1, 0, 1) : Normal mode 2 <RS232C interface> (FX, FX2C only)	
		Computer link (b12, b11, b10) (0, 0, 0) : RS485 (422) interface (0, 1, 0) : RS232C interface	
b13*2	Sum check	Sum check code is not added	Sum check code is added automatically
b14*2	Protocol	No protocol	Dedicated protocol
b15*2	Transmission control protocol	Protocol format 1	Protocol format 4

Кроме того, очень важно правильно указывать номера запрашиваемых типов регистров, которые могут иметь разные диапазоны для различных типов ПЛК. Для некоторых типов контроллеров информация о диапазонах регистров приведена ниже.

Для 16-битовых регистров:

Device	Device specification characters				Decimal/ octal expression
	FX0N	FX1S	FX, FX2C	FX1N, FX2N, FX2NC	
Timer current value (T)	TN000–TN063		TN000–TN255		Decimal
Counter current value(C)	CN000–CN031 CN235–CN254		CN000–CN255		
Data registers (D)	D0000–D0255		D0000–D0999	D0000–D7999	
File registers (D)	D1000–D2499	—	D1000–D2999	—	
RAM file registers (D)	—		D6000–D7999	—	
Special data registers(D)	D8000–D8255		D8000–D8255		

Следует еще раз отметить, что регистры C200-C255 являются 32-битными.

Для 1-битных регистров:

Device		Device specification characters					Decimal/octal expression
		FX _{0N}	FX _{1s}	FX, FX _{2c}	FX _{1N}	FX _{2N} , FX _{2NC}	
Inputs	(X)	X0000–X0177	X0000–X0017	X0000–X0337	X0000–X0177	X0000–X0267	Octal
Outputs	(Y)	Y0000–Y0177	Y0000–Y0015	Y0000–Y0337	Y0000–Y0177	Y0000–Y0267	
Auxiliary relays	(M)	M0000–M0511		M0000–M1535	M0000–M3071		Decimal
States	(S)	S0000–S0127		S0000–S0999			
Special auxiliary relays	(M)	M8000–M8254		M8000–M8255			
Timer contacts	(T)	TS000–TS063		TS000–TS255			
Counter contacts	(C)	CS000–CS031 CS235–CS254		CS000–CS255			

В данном случае особое внимание уделяется тому, что адреса для регистров X и Y указаны в восьмеричной форме, а для остальных – в десятичной. Помните, что в проекте TRACE MODE Вы должны указывать адреса в десятичной системе счисления.

Обмен по сети

Для конфигурирования обмена с контроллерами Mitsubishi MELSEC FX3U и System Q по сети используются те же шаблоны каналов (создаются в группе Mitsubishi_Group), что и для конфигурирования обмена с ПЛК серии FX по RS.

Особенности конфигурирования (см. также **Обмен по встроенным протоколам по сети**):

- IP-адрес и порт контроллера задаются в поле **Дополнительно** в формате <IP-адрес>:<порт> (например, 192.168.2.84:5002);
- **Порт** – номер станции (атрибут **Номер станции** не используется);
- если **Номер регистра**=0, регистр для обмена может быть указан в поле **Комментарий** как строка "**=<регистр>**" (например, "**=SD210**");
- типы регистров **[CN]Counter32bitHi(I)** и **[CN]Counter32bitLo(I)** не используются.

Подтип и дополнение к подтипу канала обмена – 108.11.

Поддерживается привязка шаблона канала обмена к каналу CALL.ChGroupReq (см. **Канал CALL.ChGroupReq**). Для аргумента BOOL запрашивается один бит, для аргумента USINT – один байт.

Обмен по протоколу NAIS-FP

Данный протокол используется для обмена с контроллерами MATSUSHITA.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена с контроллером (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Адрес** – номер устройства в формате HEX (0-FF);
- **Параметр** – десятичный адрес переменной (диапазон соответствует протоколу обмена);
- **Тип** – тип переменной (соответствует протоколу обмена).

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров COM-порта**).

Обмен по протоколу RKC

Данный протокол используется для обмена с контроллерами RKC Instruments.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена с контроллером (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Адрес** – номер устройства в десятичном формате (0-255);
- **Переменная** – команда в виде 2-символьной строковой переменной (в соответствии с протоколом);
- **Тип** – тип контроллера (в соответствии с протоколом).

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров COM-порта**).

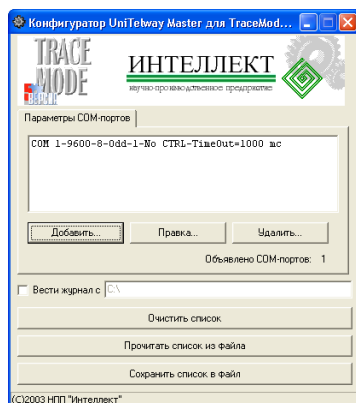
Обмен по протоколу Uni-Telway (Master)

Этот драйвер обеспечивает поддержку протокола **Uni-Telway** для обмена с ПЛК серий **TSX** концерна **Schneider Electric**. Драйвер поддерживает чтение/запись переменных **MW** и констант **KW** в режиме ведущего (MASTER) устройства. Для обмена данными используется последовательный порт компьютера и конвертер RS-232/RS-485. Драйвер оформлен в виде драйвера t12. Для корректной работы драйвера необходимо правильно произвести конфигурацию порта **TER** или **AUX** в ПЛК и последовательного порта компьютера. Протокол **Uni-Telway** поддерживается в ПЛК серий **TSX Micro**, **TSX Nano**, **TSX Premium**. О поддержке протокола в других сериях ПЛК необходимо проконсультироваться с поставщиком ПЛК.

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Конфигурирование драйвера представляет собой создание списка используемых последовательных портов ПК. Для настройки драйвера используется утилита **UTWMconfig.exe** (см. ниже). Каждой порт, указанный в списке, будет автоматически открыт драйвером.

Порт, к которому подключены устройства сети Uni-Telway, не следует настраивать в самой системе TRACE MODE.

Страница конфигурации портов имеет следующий вид:



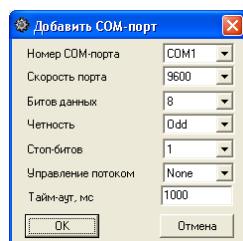
В списке портов каждая строка состоит из семи параметров.

- Номер COM-порта. Повторное объявление одного и того же порта приведет к сообщению об ошибке при попытке сохранить конфигурацию.
- Скорость передачи данных (Baud Rate), от 300 bps до 115200 bps. Для устройств сети Uni-Telway используется 9600 bps или 19200 bps. По умолчанию принимается 9600 bps.

- Число битов данных (Data Bits). По умолчанию установлено 8 бит.
- Контроль четность передачи (Parity), может принимать значения **None**, **Odd** или **Even**. По умолчанию для устройств сети Uni-Telway принимается **Odd**.
- Количество стоп-битов (Stop Bits), 1 или 2. По умолчанию установлен 1 стоп-бит.
- Время таймаута для данного последовательного порта (в мс) По умолчанию – 1000 мс.
- Управление потоком. Используемый конвертер может требовать управления потоком. Для его корректной работы необходимо правильно указать сигналы (**RTS**, **DTR**), которые будут поданы перед каждой посылкой и сняты после ее отправки. Рекомендуется использовать конвертеры интерфейсов, не требующие внешнего управления направлением передачи данных (например i-7520 от ICP DAS), во избежание возможных проблем с задержками на переключение, порядком следования управляющих сигналов и т.д.

Заданные параметры последовательного порта должны совпадать с соответствующими параметрами всех остальных устройств сети Uni-Telway. В противном случае драйвер не сможет вести обмен данными, или полученные данные не будут соответствовать действительности и могут повлечь непредсказуемые сбои в системе

Для создания новой записи нажмите кнопку **Добавить**, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



Драйвер выступает в сети в роли ведущего устройства (Master). Его адрес всегда равен 00.FE.00.00.00. Драйвер не поддерживает обмен данными между подчиненными узлами (Slave-to-Slave) и запросы от подчиненного к ведущему узлу (Slave-to-Master). Работа сети осуществляется в режиме запросов от ведущего узла к ведомым (Master-to-Slave). Данное ограничение введено для повышения быстродействия системы и улучшения ее характеристик по надежности. По вопросам настройки сети Uni-Telway и адресации в ней обращайтесь, пожалуйста, к соответствующим руководствам, в службу технической поддержки Schneider Electric или к Вашему поставщику оборудования.

Для сохранения конфигурации нажмите кнопку **Сохранить в файл**. Если в конфигурации отсутствуют явные ошибки, то будет выдано сообщение об успешном сохранении файла конфигурации (**UTWM.cfg**). Однако это не говорит о гарантированном отсутствии ошибок в настройках драйвера, т.к. большинство логических ошибок не могут быть распознаны утилитой на этапе настройки. Ответственность за создание правильной конфигурации полностью возлагается на пользователя.

Опция **Вести журнал событий** предоставляет возможность удобной отладки работы системы. По указанному пути будут созданы 2 файла: **UTWMmedia.log** и **UTWMproto.log**, в которых будет сохраняться протокол работы драйвера, сообщения о сбоях и их возможных причинах. Указанный каталог должен существовать еще до запуска проекта TRACE MODE. После успешной настройки системы эту опцию можно отключить, снизив затраты времени и дискового пространства.

Файл конфигурации должен располагаться в папке проекта. Не пытайтесь редактировать его вручную и не удаляйте его, т.к. это может привести к непредсказуемым сбоям в системе. В случае отсутствия файла конфигурации, драйвер не будет запущен, и профайлер сохранит в файл отладочной информации строку **PROTOCOL:Open error = -1**. При этом все каналы, относящиеся к данному протоколу, не будут обновляться, однако достоверность для всех них будет равна нулю.

Для открытия существующей конфигурации воспользуйтесь кнопкой **Прочитать из файла**. Для удаления текущей конфигурации (без сохранения в файл) воспользуйтесь кнопкой **Очистить**.

Драйвер реализует обмен данными в сети Uni-Telway в режиме ведущего (MASTER) узла. Сеть Uni-Telway представляет собой т.н. **Polling Network** (Сеть с опросом). Такая архитектура позволяет повысить надежность передачи данных при несколько более низких показателях скорости обмена. Обмен данными происходит по следующей схеме:

- запрос Master-узлом данных у Slave-узла;
- подтверждение получения запроса Slave-узлом;
- опрос Master-узлом всех Slave-узлов на предмет готовности требуемых данных;
- ответ Slave-узла, содержащий запрашиваемые данные или сведения об ошибках при выполнении запроса.

Драйвер реализует обмен в указанном режиме с каждым из подчиненных узлов. При этом время ожидания ответа на любой из запросов лимитировано заданным в конфигураторе значением тайм-аута. Значение тайм-аута по умолчанию равно 1000 мс, однако при уверенном обмене по последовательному порту на скорости 9600 бод оно может быть снижено до 150-

200 мс. Оптимальное значение подбирается на практике. Кроме того, в драйвере предусмотрена система повторных запросов (RETRY-система). Время обмена данными с подчиненными устройствами зависит от скорости и качества линии связи, числа подчиненных устройств, подключенных к одному последовательному порту, и количества запрашиваемых каналов. Поскольку в данном протоколе используется 4-шаговая схема обмена, повышающая надежность системы, скорость передачи данных несколько возрастает и составляет около 120мс на запрос.

Некоторые ПЛК (например, TSX Micro) для работы в режиме Slave требуют не только программной (с помощью PL7), но и аппаратной конфигурации (с помощью подачи логического 0 или 1 на соответствующий вход порта TER/AUX). Для уточнения используйте документацию по данному ПЛК. В режиме Slave ПЛК не должен самопроизвольно управлять никакими запросами в сеть. Это можно проконтролировать по состоянию соответствующего индикатора и с помощью любой терминальной программы на ПК. В противном случае обмен выполняться не будет, т.к. сеть поддерживает наличие только одного ведущего узла.

За более подробной информацией по принципам функционирования сети Uni-Telway обращайтесь к соответствующим руководствам пользователя, в службу технической поддержки Schneider Electric или к Вашему поставщику оборудования.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Порт** – номер последовательно порта (0 – COM1, ... 31 – COM 32). Настройки порта должны быть предварительно описаны в файле конфигурации **UTWM.cfg**, в противном случае используемые каналы будут нефункциональными. Номер порта задается в десятичной системе счисления;
- **Адрес** – адрес ведомого устройства в сети UniTelway. В соответствии с принципами функционирования данной сети, каждое подчиненное устройство должно иметь свой уникальный адрес. Дублирование строго запрещено. По умолчанию в режиме Slave ПЛК имеет адрес 1. Адрес задается в десятичной системе счисления;
- **Номер элемента** – номер слова. Допустимые значения зависят от типа используемого оборудования. Пределы – 0 .. 65535. Номер слова указывается в десятичной системе счисления;
- **Тип данных** – тип данных, с которыми будет производиться обмен. Это могут быть переменные (**MW**) или константы (**KW**). Все данные доступны для чтения и записи и имеют размер 16 бит. Тип данных выбирается из выпадающего списка.

Обращение к неверно настроенному ПЛК, адресация к не-

существующим переменным или константам может привести к непредсказуемым сбоям в системе.

Для обмена данными необходимо создать каналы HEX16 или FLOAT. При использовании каналов FLOAT дробная часть не учитывается, т.к. данные, с которыми оперирует протокол UniTelway, являются целыми (WORD). Также следует избегать использования отрицательных значений, поскольку при этом должны учитываться особенности представления целых отрицательных чисел внутри ПК. Тип канала (I или O) определяет его назначение – чтение данных или запись.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. При каждой посылке ответ ожидается в течение заданного таймаута. При сбоях в обмене применяется система лимитированного количества повторных запросов. Если ответа за это время нет, то каналу выставляется признак недостоверности. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Необходимо обратить внимание на то, что при безуспешной попытке записи значения канала, TRACE MODE будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

Обмен с контроллером RE508

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данным устройством задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Адрес** – номер контроллера в формате HEX (0-FF);
- **Параметр** – десятичный адрес переменной (диапазон соответствует протоколу обмена);
- **Тип** – тип переменной (соответствует протоколу обмена).

Для обмена необходимо настроить последовательный порт (см. **Редактор параметров COM-порта**).

Обмен по протоколу ModbusPlus

Драйвер обмена работает с любыми платами (PCI, ISA, PCMCIA) ModbusPlus производства Schneider Electric. Для функционирования драйвера нужна библиотека NETLIB.DLL. Эта библиотека автоматически устанавливается в процессе установки Windows-драйвера (MBX Driver) платы ModbusPlus. Windows-драйвер входит в комплект поставки плат ModbusPlus Schneider Electric.

Возможна работа с платами других производителей. Для этого производитель должен поставлять 32-битную библиотеку NETLIB.DLL. Библиотека должна находиться в системном каталоге Windows (\WINDOWS\SYSTEM или \WINDOWS\SYSTEM32).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена по данному протоколу задаются следующие атрибуты:

- **Адрес** – однобайтовое число (задается в десятичном виде). 6 младших битов этого числа задают адрес контроллера в сети ModbusPlus (0 соответствует адресу 1 в сети ModbusPlus, 63 – адресу 64). Мосты и маршрутизация не поддерживаются. Старший бит настройки **Адрес** задает номер используемого порта адаптера (сетевой платы ModbusPlus) (0 или 1). Например, для чтения из контроллера с адресом 3 через порт 1 адаптера настройка **Адрес** должна быть равна 130 (128+2);
- **Область** – команда ModbusPlus из числа реализованных в драйвере (1 байт). Команда выбирается из списка (см. ниже);
- **Смещение** – смещение первого читаемого регистра относительно начала группы (2 байта). Этот параметр задается в десятичном виде (0 - 65535). Например, если нужно прочитать 10 out-регистров с 00001 по 00010, **Смещение**=0, **Счетчик**=10. Если нужно прочитать один регистр 40002, **Смещение**=1, атрибут **Счетчик** игнорируется;
- **Счетчик** – количество регистров для чтения/записи (1 байт). Этот параметр имеет смысл только для команд чтения/записи дискретных (однобитовых) регистров 0x и 1x. Регистры 3x и 4x читаются (или в них записывается значение) всегда по одному. Настройка **Счетчик** задается в десятичном виде (1 – 16).

В драйвере реализованы следующие команды ModbusPlus (число в скобках соответствует десятичному коду команды; в командах 03, 04, 06 атрибут **Счетчик** игнорируется):

- **Read0x (01)** – считать состояние группы дискретных out-регистров. По этой команде можно получить состояние от 1 до 16 дискретных (однобитовых) регистров 0x. Самый младший разряд

полученного числа соответствует первому регистру. При запросе менее 16 регистров неиспользуемые разряды заполняются нулями. Если запрошено более 16 регистров, получены будут только первые 16;

- **Read1x (02)** – то же, что Read0x, но для in-регистров 1x (однобитовых);
- **Write0x (15)** – записать в 0x однобитовые out-регистры (правила такие же, как для команды Read0x);
- **Read4x (03)** – считать один 4x out-регистр (16 бит);
- **Read3x (04)** – считать один 3x in-регистр (16 бит);
- **Write4x (06)** – записать в один 4x out-регистр (16 бит);
- **Force0x (05)** – установить/сбросить один 0x out-регистр (однобитовый). Нулевое значение, записываемое в канал, соответствует сбросу регистра. Отличное от 0 значение соответствует установке регистра;
- **ReadExcpt (07)** – считать out-регистры (однобитовые) 00001-00008, при этом полученное число лежит в диапазоне 0-255. Младший бит соответствует регистру 00001, старший – регистру 00008.

Обмен с весовыми терминалами ЗАО 'Тензо-М'

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с данными устройствами задаются следующие атрибуты:

- **Порт** – номер COM порта, к которому подключен терминал (0 – COM1, 1 – COM2 и т.д.). Порт должен быть настроен в TRACE MODE в соответствии с аппаратными настройками терминала (см. документацию на терминал);
- **Адрес** – сетевой адрес терминала (1...32). Этот параметр должен соответствовать адресу, установленному в терминале;
- **Команда** – команда терминала (выбирается из меню);
- **Аргумент** – аргумент команды (число 0-255), используется некоторыми командами.

Каналы обмена позволяют считывать с терминалов значения веса БРУТТО и НЕТТО, устанавливать ноль весов, а также считывать значения суммарных счетчиков терминала.

N	Команда	Тип канала	Как используется аргумент	Описание
1	GET_BRUTTO	IN, FLOAT	не используется	Получить вес брутто в кг
2	GET_NETTO	IN, FLOAT	не используется	Получить вес нетто в кг
3	SET_ZERO	OUT, HEX, FLOAT	не используется	Установить ноль. При записи в этот канал ненулевого значения происходит установка нуля весов. *
4	GET_COUNTER_HI	IN, HEX, FLOAT	значение аргумента равно номеру счетчика 0...15**	Получить старшее слово (16 бит) значения суммарного счетчика
5	GET_COUNTER_LO	IN, HEX, FLOAT	значение аргумента равно номеру счетчика 0...15**	Получить младшее слово (16 бит) значения суммарного счетчика

* Т.к. TRACE MODE не производит запись в канал, если входное значение канала не изменилось, то для повторной установки нуля необходимо сначала записать в канал 0, потом записать в него ненулевое значение.

** Количество доступных счетчиков зависит от модели терминала.

СВЯЗЬ С СЕТЯМИ LONWORKS

Драйвер LONWORKS предназначен для организации связи TRACE MODE с сетями LonWorks.

Драйвер работает на уровне сетевых переменных, он позволяет читать и записывать сетевые переменные стандартных типов (Standard Network Variables Types или SNVT). Для переменных структурированных типов предусмотрена возможность чтения/записи отдельных полей.

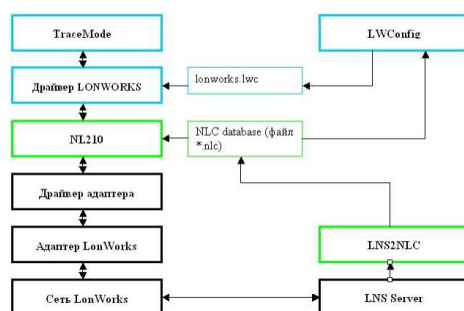
Необходимое программное и аппаратное обеспечение

Для работы драйвера необходимы:

- библиотека NL210, которую можно приобрести в фирме-производителе NEWRON SYSTEM (<http://www.newron-system.com>). Библиотека доступна в нескольких версиях, поддерживающих различное число тегов;
- LNS Server фирмы Echelon™ (<http://www.echelon.com>). Это стандартное программное обеспечение для конфигурирования и настройки сетей LonWorks. Во время работы готового проекта наличие работающего LNS Server не требуется;
- сетевой адаптер для доступа к сети LonWorks. Драйвер тестировался с адаптерами фирмы Echelon™. Драйвер работает также с адаптерами других производителей, если библиотека NL210 работает с этими адаптерами (подробности можно уточнить в фирме NEWRON SYSTEM).

Принцип работы и архитектура

На следующем рисунке представлена схема работы драйвера.



Синим цветом выделены компоненты Trace Mode и драйвера. Зеленым цветом выделены компоненты библиотеки NL210. Тонкими рамками обозначены файлы данных. Стрелки показывают направление передачи данных.

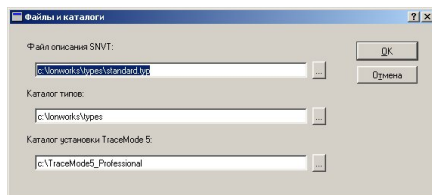
Работа с драйвером

Перед началом работы необходимо убедиться, что сеть LonWorks установлена и сконфигурирована. Кроме того, необходимо установить библиотеку NL210.

Из подкаталога **Distributed Files** каталога установки NL210 (по умолчанию **C:\Program Files\Newron System\NL210**) нужно скопировать все файлы в каталог установки TRACE MODE.

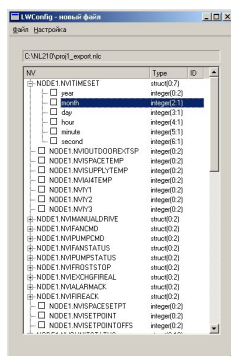
При помощи утилиты LNS2NLC, входящей в состав NL210, надо создать базу данных NLC – например, **c:\nl210\proj1.nlc**.

Далее надо запустить утилиту **LWConfig** из комплекта драйвера. Если утилита запускается первый раз, и LNS сервер (или TRACE MODE) установлен в нестандартный каталог, то появится предупреждение, а после нажатия **OK** – диалоговое окно, изображенное на рисунке:



В этом окне необходимо ввести пути к файлу описания стандартных типов переменных LonWorks (обычно это **c:\lonworks\types\standard.typ**), к каталогу, где находится этот файл, и к каталогу TRACE MODE. Это нужно сделать только один раз, после чего эти пути будут сохранены в реестре, и при дальнейшей работе с утилитой их вводить не придется.

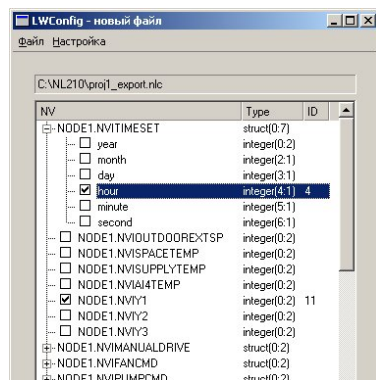
Далее надо выбрать команду **Открыть *.NLC** меню **Файл** и выбрать файл базы данных NLC, созданный с помощью утилиты LNS2NLC. В главном окне программы появится дерево сетевых переменных.



Узлы, рядом с которыми нет квадратиков, являются переменными (или полями переменной структурного типа), которые можно связать с канала-

ми Trace Mode. Узлы первого уровня – это сами сетевые переменные. Узлы следующих уровней – поля структурных переменных. Во второй колонке показан тип переменной или поля. В скобках показано смещение поля относительно начала переменной и длина поля в байтах. В третьей колонке – идентификатор, который нужно использовать в Trace Mode для обращения к переменной/полю.

В этом диалоге нужно с помощью мыши поставить галочки рядом с переменными/полями, которые предполагается использовать в Trace Mode.



Далее надо сохранить конфигурацию в файл **%TRACE MODE%\lonworks.lwc** (команда **Сохранить конфигурацию** из меню **Файл**). Следует иметь в виду, что TRACE MODE работает только с этим файлом. Если файл NLC изменяется, то необходимо заново создать файл ***.lwc** (идентификаторы полей могли измениться). Для просмотра идентификаторов выбранных полей сохраненный файл ***.lwc** можно открыть командой **Файл\Открыть конфигурацию**.

Для каждого выбранного поля нужно создать канал связи.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**, а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Идентификатор** – идентификатор, который назначен соответствующему полю утилитой **LWConfig**;
- **Раздел** – принимает значения **HI** или **LO**, предназначен для работы с полями типа **SIGNED_QUAD**. Поля этого типа являются 32-битовыми, поэтому в Trace Mode передаются частями: **LO** – младшие 16 бит, **HI** – старшие 16 бит.

Особенности записи полей переменных структурных типов

Если канал типа **OUTPUT** связан с каким-то полем сетевой переменной структурного типа, то при записи в этот канал значения вся переменная

будет считана из сети в память, после чего будет модифицирована та ее часть, которая относится к данному полю. После этого переменная будет записана обратно в сеть. То же самое касается переменных типа SIGNED_QUAD.

Протоколирование работы драйвера

В драйвере предусмотрена возможность протоколирования работы. Для ее включения необходимо в каталоге проекта Trace Mode создать файл с именем **_logger** (без расширения). В файле должны быть следующие строки:

```
;=====
;0-все сообщения
;1-только предупреждения и ошибки
;2-только ошибки
    level=0
;=====
;максимальный размер файла в Мб
    max_size=50
;=====
;вкл/выкл протоколирование
    enabled=true
```

Строки, начинающиеся с точки с запятой (";"), являются комментариями.

Следует отметить, что протоколирование очень замедляет работу системы.

Протоколы работы заносятся в файлы **media.log** и **t12.log**. Первый относится к работе функций описания носителя, а второй – к работе функций протокола (см. раздел **Разработка драйверов. Интерфейс TCOM**), хотя в данном драйвере такое разделение достаточно условно.

Обмен по протоколу S-BUS

Драйвер предназначен для связи с контроллерами по протоколу **S-BUS DataMode** через последовательный порт. Протокол S-BUS разработан фирмой SAIA-burgess и поддерживается всеми контроллерами этой фирмы.

Драйвер позволяет читать и записывать регистры, флаги, дискретные выходы, счетчики, таймеры. Драйвер позволяет читать дискретные входы, регистр дисплея (Display Register), состояние PCD. Также драйвер позволяет осуществлять горячий и холодный перезапуск контроллера, останавливать и запускать программу в контроллере, синхронизировать часы контроллера с часами компьютера, обнулять регистры, счетчики, таймеры, флаги и дискретные выходы.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 соответствует COM1);
- **Адрес** - адрес контроллера в сети s-bus (0..255). 255 означает широковещательную рассылку и может применяться с некоторыми командами;
- **Аргумент** - адрес регистра, счетчика, таймера, дискретного входа, дискретного выхода или процессора (для контроллеров с несколькими процессорами);
- **Команда** - команда протокола.

Список команд

	Название в спецификации протокола	Назначение настройки АРГУМЕНТ	Описание
RR_int	Read Register	адрес регистра	Чтение регистра в формате целого со знаком.
RR_float	Read Register	адрес регистра	Чтение регистра в формате с плавающей точкой.
RC	Read Counter	адрес счетчика	Чтение счетчика.
RT	Read Timer	адрес таймера	Чтение таймера.
RF	Read Flag	адрес флага	Чтение 16-ти идущих подряд флагов. Канал должен иметь представление HEX.
RI	Read Input	адрес дискретного	Чтение 16-ти идущих подряд дискретных входов. Канал должен иметь

	Название в спецификации протокола	Назначение настройки АРГУМЕНТ	Описание
		входа	представление HEX.
RO	Read Output	адрес дискретного выхода	Чтение 16-ти идущих подряд дискретных выходов. Канал должен иметь представление HEX.
RD_lo	Read Display Register	не используется	Чтение младших 16 бит регистра дисплея (Display Register). Канал должен иметь представление HEX.
RD_hi	Read Display Register	не используется	Чтение старших 16 бит регистра дисплея (Display Register). Канал должен иметь представление HEX.
RS	Read PCD Status	0..6 – номер ЦПУ, 7 – свой ЦПУ	Чтение состояния контроллера. Канал получает значение ASCII кода одного из следующих символов: 'S'(83), 'H'(72), 'R'(82), 'C'(67), 'D'(68). 'S' – программа в контроллере остановлена. 'R' – программа в контроллере выполняется.
WR_int	Write Register	адрес регистра	Запись регистра в формате целого со знаком.
WR_float	Write Register	адрес регистра	Запись регистра в формате с плавающей точкой.
WC	Write Counter	адрес счетчика	Запись счетчика.
WT	Write Timer	адрес таймера	Запись таймера.
WF	Write Flag	адрес флага	Запись 16-ти идущих подряд флагов. Канал должен иметь представление HEX.
WO	Write Output	адрес дискретного выхода	Запись 16-ти идущих подряд дискретных выходов. Канал должен иметь представление HEX.
WK	Write Real Time Clock	не используется	При записи любого значения в этот канал происходит синхронизация часов контроллера с часами компьютера..
CA	Clear All	не используется	При записи любого значения в этот канал происходит обнуление всех регистров, счетчиков, таймеров, флагов, дискретных выходов
CF	Clear Flags	не используется	При записи любого значения в этот канал происходит обнуление всех флагов

	Название в спецификации протокола	Назначение настройки АРГУМЕНТ	Описание
CO	Clear Outputs	не используется	При записи любого значения в этот канал происходит обнуление всех дискретных выходов
CR	Clear Registers	не используется	При записи любого значения в этот канал происходит обнуление всех регистров
CT	Clear Timers	не используется	При записи любого значения в этот канал происходит обнуление всех таймеров
SW	Restart Warm	0..5 – номер ЦПУ, 7 – свой ЦПУ	При записи любого значения в этот канал происходит горячий перезапуск контроллера
SC	Restart Cold	0..5 – номер ЦПУ, 7 – свой ЦПУ	При записи любого значения в этот канал происходит холодный перезапуск контроллера
SP	Stop Procedure	0..5 – номер ЦПУ, 7 – свой ЦПУ, 8 – все ЦПУ	При записи любого значения в этот канал происходит остановка выполнения программы в контроллере
GP	Run Procedure	0..5 – номер ЦПУ, 7 – свой ЦПУ, 8 – все ЦПУ	При записи любого значения в этот канал происходит запуск выполнения программы в контроллере

Замечание. Для всех контроллеров, оснащенных единственным ЦПУ (это все контроллеры кроме РСД6), настройку АРГУМЕНТ тех каналов, где она означает номер ЦПУ, рекомендуется установить в 0.

Протокол S-BUS предусматривает два формата для чисел – 32-битное целое и 32-битное с плавающей точкой. Протокол S-BUS ограничивает числа с плавающей точкой диапазоном $(-9.22337177 \cdot 10^{18}, 9.22337177 \cdot 10^{18})$. По этой причине не рекомендуется передавать из Trace Mode в контроллеры числа с плавающей точкой, выходящие за этот диапазон, поскольку это может приводить к ошибочным значениям в регистрах контроллера.

В драйвере предусмотрена возможность протоколирования работы. Для ее включения необходимо в каталоге проекта TraceMode создать файл с именем **_logger** (без расширения). В файле должны быть следующие строки:

```
;=====
;0-все сообщения
```



```
;1-только предупреждения и ошибки
;2-только ошибки
level=0
;=====
;максимальный размер файла лога в Мб
max_size=50
;=====
;вкл/выкл лог
enabled=true
```

Строки, начинающиеся с точки с запятой (;), являются комментариями и необязательны. Назначение остальных строк ясно из комментариев.

Следует отметить, что протоколирование сильно замедляет работу системы. Также при включенном протоколировании размер файла протокола растет очень быстро, особенно при интенсивном обмене данными с контроллерами.

Протокол работы заносится в файл **t11.log**.

Обмен с контролерами Honeywell DPR3000, DPR250

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер COM-порта, по которому производится обмен (начиная с 0);
- **Адрес** – адрес контроллера;
- **Канал** – номер канала, который будет считываться;
- **Тип** – тип обмена:
 - **ALR_STS** – считывание тревог (номер бита задает номер считываемой тревоги);
 - **PROC_VAL** –
 - **RELAY_STS** –
- **CRC** – признак использования контрольной суммы.

Обмен с контроллером STEC2000

Для корректной работы драйвера нужно скопировать файл **%TRACE MODE 6%\Drivers_with_Setup\Stec2000\STEC2000_driver.ini** в директорию MPB и отредактировать его, задав следующие параметры:

- список номеров приборов **DevicesList** через запятую (например, "DevicesList = 100,101");
- для каждого номера прибора создать секцию "[Dev_X]", где X – номер прибора из списка. Каждая секция должна содержать строковый параметр IP, содержащий IP-адрес прибора (например, "IP = 192.168.0.157"). Кроме того, в секции можно задать дополнительные параметры:
 - ThreadCycle – время рабочего цикла потока группового запроса (мс), по умолчанию – 500 мс;
 - ConnectTimeout – время ожидания соединения (мс), по умолчанию – 1000 мс;
 - ConnectInterval – период между попытками соединения, по умолчанию – 3000 мс;
 - MaxSendTime – максимальное время ожидания отправки пакета в сеть, по умолчанию – 1000 мс;
 - MaxRecvTime – максимальное время ожидания входящего пакета, по умолчанию – 1000 мс;
 - ErrorsCount – счетчик ошибок, после превышения заданного значения соединение разрывается и устанавливается заново, по умолчанию – 5.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Идентификатор** – идентификатор тега, 2 байта в десятичном формате;
- **Номер устройства** – номер прибора, 1 байт в десятичном формате;
- **Команда** – тип команды, задается выбором из меню одного из следующих значений:
 - **Время**
 - **Управление**
 - **БлокЗапрос** (групповой, все каналы этого типа запрашиваются одним запросом; запрос выполняется в отдельном потоке, который не связан с потоками MPB)
 - **ЕдиничЗапрос** (запрос одного канала)
- **Тип** – тип значения канала; указывается для правильной интер-

претации значения в контроллере (только для каналов **Управление**); задается выбором из меню одного из следующих типов: **Float, Integer, Boolean**.

Все каналы **Управление** должны иметь в Trace Mode тип OUTPUT, а **БлокЗапрос** и **ЕдиницаЗапрос** – тип INPUT.

Для использования команды **Время** нужно создать 2 канала обмена – один типа INPUT, другой – OUTPUT. Для созданных каналов надо задать номер устройства (например, 100), в меню **Команда** выбрать **Время**, остальные параметры значения не имеют. До синхронизации во входной канал значение не записывается, и устанавливается недостоверность. После записи в выходной канал любого числа посылается сигнал синхронизации, после него во входной канал записывается разница значений времени компьютера и прибора.

Обмен с приборами фирмы 'Дана-Терм'

Помимо драйвера (**t11s34.dll**), в директории MPB должен находиться следующий текстовый файл **dt_driver.ini**:

```
[Settings]
N_group = 255;
<пустая строка>
```

Согласно документации на приборы «Дана-Терм», для последовательного интерфейса должны быть заданы следующие настройки (см. **Редактор параметров COM-порта**):

- скорость передачи – 9600;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – НЕТ.

Время ожидания ответа зависит от модели прибора; рекомендуемые значения этого таймаута по каждому прибору даны в соответствующих описаниях. При выборе таймаута меньше рекомендованного значения значительно увеличивается вероятность появления ошибок приема.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого COM-порта (0 – COM1, 1 – COM2 и т.д.);
- **Номер прибора** – номер прибора в группе, один байт в десятичном формате (0-255);
- **Прибор** – тип прибора, выбирается из следующего списка:
 - ИТ 2512 (4506) – 2-канальный измеритель температуры;
 - ИТ 2513 (4505) – 2-канальный измеритель температуры и разницы температур;
 - МУИ 2516 – прецизионный измеритель температуры;
 - ИТ 2518 – 10-канальный измеритель температуры;
 - ИТР 2521-2 – 2-канальный регулирующий измеритель температуры;
 - ИТВР 2522 – 2-канальный регулирующий измеритель температуры и влажности;
 - ИТР 2523 – регулирующий измеритель температуры с регулированием по ПИД-закону;
 - ИТР 2525 (2508) – 2-канальный регулирующий измеритель температуры;
 - ИТР 2528 – 10-канальный регулирующий измеритель тем-

- пературы;
- ИТР 2529 – регулирующий измеритель температуры;
 - ИТВ 2605 – измеритель температуры и влажности;
 - ИТВ 2605/8 – измеритель температуры и влажности 8-канальный;
 - ИТВР 2606 – регулирующий измеритель температуры и влажности;
 - 3501...3504 – группа цифровых таймеров, отличающихся количеством каналов (1...4);
 - ЦТ 3603 – цифровой тахометр;
 - 3701...3703 – группа цифровых счетчиков.
- **Команда** – тип запроса к устройству, один байт в десятичном формате (0-255). Запросы, определенные для конкретных устройств, приведены в таблицах (см. таблицы команд ниже). При попытке реализовать неподдерживаемый запрос драйвер ничего не делает (данные не отправляются и не принимаются);
 - **Параметр** – дополнительный параметр к запросу, один байт в шестнадцатеричном формате (0-FF). Назначение этой настройки зависит от типа прибора (см. ниже особенности настройки обмена).

Таблицы команд

В описании команд указано, для какого типа канала они могут быть использованы (INPUT/OUTPUT).

Код команды	ИТ 2512 (4506)	ИТ 2513 (4505)	МУИ 2516	ИТ 2518
0	Температура (Input)	Температура (Input)	Температура (Input)	Температура (Input)
1	Верхний порог срабатывания аварийной сигнализации (Input, Output)	Разница температур (Input, Output)	Результат проверки на превышение предела (Input)	Верхний порог срабатывания аварийной сигнализации (Input, Output)
2	Нижний порог срабатывания аварийной сигнализации (Input, Output)	Тип датчика (Input, Output)	Сопrotивление термометра (Input)	Нижний порог срабатывания аварийной сигнализации (Input, Output)

Код ко-манды	ИТ 2512 (4506)	ИТ 2513 (4505)	МУИ 2516	ИТ 2518
3	Тип датчика (Input)		Разность температур между выбранным каналом и заданной величиной (Input)	
4			Минимальный предел температуры (Input, Output)	
5			Максимальный предел температуры (Input, Output)	
6			Базовая температура (Input, Output)	
7			Число усреднений (Input, Output)	

Код ко-манды	ИТР 2521-2	ИТВР 2522	ИТР 2523	ИТР 2525 (2508)
0	Температура (Input)	Температура в камере (Input)	Режим регулирования (Input, Output)	Температура (Input)
1	Температура задания (Input, Output)	Влажность в камере (Input)	Температура задания (Input, Output)	Температура задания (Input, Output)
2	Температурный гистерезис (Input, Output)	Температура в сердцевине (Input)	Коэффициент пропорциональности P (Input, Output)	Аларм (Input, Output)
3	Верхний порог срабатывания аварийной сигнализации (Input, Output)	Время с начала регулирования (Input)	Блокировка интеграла A (Input, Output)	Температурный гистерезис (Input, Output)
4	Нижний порог срабатывания аварийной сигнализации (Input, Output)	Состояние (Input, Output)	Коэффициент интеграла I (Input, Output)	Тип датчика (Input)

Код ко-манды	ИТР 2521-2	ИТВР 2522	ИТР 2523	ИТР 2525 (2508)
5	Температура компенсатора "холодных концов" термпары (температура окружающей среды) (Input)	Температура регулирования в камере (Input, Output)	Коэффициент дифференциала d (Input, Output)	
6	Параметр шага - начальная точка для температуры (Input, Output)	Влажность регулирования в камере (Input, Output)	Период ШИМа (Input, Output)	
7	Параметр шага - конечная точка для температуры (Input, Output)	Температура выключения по сердцевине (Input, Output)	Максимум выходной мощности PH (Input, Output)	
8	Параметр шага - температурный гистерезис (Input, Output)	Время регулирования (Input, Output)	Время линейной развертки температуры (Input, Output)	
9	Параметр шага - верхний порог срабатывания аварийной сигнализации для температуры (Input, Output)	Байт регулирования по каналам (Input, Output)	Время регулирования (Input, Output)	
10	Параметр шага - нижний порог срабатывания аварийной сигнализации для температуры (Input, Output)		Температура первого аларма AL1 (Input, Output)	
11	Длительность шага (Input, Output)		Температура второго аларма AL2 (Input, Output)	
12	Начальный шаг (Input, Output)		Текущая температура (Input)	
13	Статус (состояние) ИТВР (Input, Output)		Состояние прибора (Input, Output)	

Код ко-манды	ИТР 2521-2	ИТВР 2522	ИТР 2523	ИТР 2525 (2508)
14	Текущий шаг (Input)		Текущий номер шага (Input)	
15	Время, прошедшее с начала текущего шага (Input)		Тип датчика (Input)	
16			Количество шагов программы регулирования Pn (Input, Output)	
17			Критерий достижения заданной температуры АС (Input, Output)	
18			Гистерезис первого аларма Г.AL1 (Input, Output)	
19			Гистерезис второго аларма Г.AL2 (Input, Output)	
20			Режим работы первого аларма (Input, Output)	
21			Режим работы второго аларма (Input, Output)	
22			Минимум выходной мощности PL (Input, Output)	
23			Минимальная длительность импульса управления PULS (Input, Output)	
24			Режим управления пуском (Input, Output)	

Код ко-манды	ИТР 2528	ИТР 2529	ИТВ 2605	ИТВ 2605-8
0	Температура (Input)	Температура (Input)	Температура (Input)	Температура (Input)

Код ко-манды	ИТР 2528	ИТР 2529	ИТВ 2605	ИТВ 2605-8
1	Температура задания (Input, Output)	Температура задания (Input, Output)	Влажность (Input)	Влажность (Input)
2	Температурный гистерезис (Input, Output)	Температурный гистерезис (Input, Output)	Температура компенсатора "холодных концов" термопары (температура окружающей среды) (Input)	
3		Верхний порог срабатывания аварийной сигнализации (Input, Output)		
4		Нижний порог срабатывания аварийной сигнализации (Input, Output)		
5		Температура компенсатора "холодных концов" термопары (температура окружающей среды) (Input)		

Код ко-манды	ИТВР 2606	3501_3504	ЦТ 3603	3701_3703
0	Температура (Input)	Состояние (Input)	Положение десятичной точки, количество знаков после запятой (Input, Output)	Количество пакетов (Input)
1	Температура задания (Input, Output)	Остаток отсчета (Input)	Уставка реле 1 (Input, Output)	Количество групп пакетов (Input)
2	Температурный гистерезис (Input, Output)	Уставка (Input, Output)	Уставка реле 2 (Input, Output)	Состояние счетчика (Input)

Код ко-манды	ИТВР 2606	3501_3504	ЦТ 3603	3701_3703
3	Верхний порог срабатывания аварийной сигнализации (Input, Output)	Положение точки, количество цифр после запятой (Input, Output)	Гистерезис реле 1 (Input, Output)	Уставка счета (Input)
4	Нижний порог срабатывания аварийной сигнализации (Input, Output)	Состояние в момент пуска отсчета (Input)	Гистерезис реле 2 (Input, Output)	
5	Температура компенсатора "холодных концов" термопары (температура окружающей среды) (Input)	Остаток в момент пуска отсчета (Input)	Значение измеренное тахометром (Input)	
6	Влажность (Input)	Уставка в момент пуска отсчета (Input)		
7	Точка задания для влажности (Input, Output)	Положение точки в момент пуска отсчета (Input)		
8	Влажностный гистерезис (Input, Output)			
9	Параметр шага - начальная точка для температуры (Input, Output)			
10	Параметр шага - конечная точка для температуры (Input, Output)			
11	Параметр шага - температурный гистерезис (Input, Output)			
12	Параметр шага - верхний порог срабатывания аварийной сигнализации для температуры (Input, Output)			

Код ко-манды	ИТВР 2606	3501_3504	ЦТ 3603	3701_3703
13	Параметр шага - нижний порог срабатывания аварийной сигнализации для температуры (Input, Output)			
14	Параметр шага - начальная точка для влажности (Input, Output)			
15	Параметр шага - конечная точка для влажности (Input, Output)			
16	Параметр шага - влажностный гистерезис (Input, Output)			
17	Параметр шага - верхний порог срабатывания аварийной сигнализации для влажности (Input, Output)			
18	Параметр шага - нижний порог срабатывания аварийной сигнализации для влажности (Input, Output)			
19	Длительность шага (Input, Output)			
20	Начальный шаг (Input, Output)			
21	Статус (состояние) ИТВР (Input, Output)			
22	Текущий шаг (Input)			
23	Время, прошедшее с начала текущего шага (Input)			
24	Верхний порог сигнализации для влажности (Input, Output)			
25	Нижний порог сигнализации для влажности (Input, Output)			

Особенности настройки обмена

ИТ 2512

Параметр – номер канала (1 или 2).

Рекомендуемое значение таймаута – 150 мс.

ИТ 2513

Параметр – номер канала (1 или 2).

Рекомендуемое значение таймаута – 150 мс.

МУИ 2516

Параметр – номер канала (от 1 до 8).

Рекомендуемое значение таймаута – 150 мс.

ИТР 2518

Параметр – номер канала (от 1 до 0A), используется только для запросов температуры, для остальных запросов должен быть равен 0.

Рекомендуемое значение таймаута – 150 мс.

ИТР 2521-2

Параметр – передает номер шага (этапа) регулирования и канал, на каждое из значений выделено 4 бита, т.е. в шестнадцатеричном виде первый разряд – номер шага (0..9), нулевой – номер канала (1 или 2). Номер шага должен быть равен 0 для всех запросов, кроме запросов параметров шага. Например, для запроса восьмого шага второго канала параметр будет равен 82.

Рекомендуемое значение таймаута – 200 мс.

ИТВР 2522

Дополнительных параметров запросов нет. Рекомендуемое значение таймаута – 150 мс.

ИТР 2523

Параметр – номер шага (этапа) регулирования (1..9), должен быть равен 0 для всех запросов, кроме запросов параметров шага.

Рекомендуемое значение таймаута – 200 мс.

ИТР 2525

Параметр – номер канала (1 или 2).

Рекомендуемое значение таймаута – 150 мс.

ИТР 2528

Параметр – номер канала (от 1 до 0A).

Рекомендуемое значение таймаута – 150 мс.

ИТР 2529

Дополнительных параметров запросов нет. Рекомендуемое значение таймаута – 150 мс.

ИТВ 2605

Дополнительных параметров запросов нет. Рекомендуемое значение таймаута – 150 мс.

ИТВ 2605-8

Параметр – номер канала (от 1 до 8).

Рекомендуемое значение таймаута – 150 мс.

ИТР 2606

Параметр – номер шага (этапа) регулирования (1..9), должен быть равен 0 для всех запросов, кроме запросов параметров шага.

Рекомендуемое значение таймаута – 200 мс.

Значение температуры компенсатора термопары учитывается только при использовании в качестве датчика термопар ХА или ХК.

3501...3504

Параметр – номер канала (от 1 до 4). Рекомендуемый таймаут – 150 мс.

ЦТ 3603

Дополнительных параметров запросов нет. Рекомендуемый таймаут – 150 мс.

3701...3703

Дополнительных параметров запросов нет. Рекомендуемый таймаут – 150 мс.

Обмен с дозатором ДОЗА-4ТМ

Драйвер обмена с дозатором ДОЗА-4ТМ оформлен в виде драйвера t11.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого СОМ-порта, десятичное число (0 – СОМ1, 1 – СОМ2, и т.д.);
- **Адрес** – адрес прибора ДОЗА-4ТМ, десятичное число (0...255);
- **Команда** – команда; выбирается из списка (см. таблицу ниже);
- **Смещение** – смещение читаемого/записываемого слова (16 бит) в блоке читаемых/записываемых данных.

Название команды в TRACE MODE	Название в спецификации протокола	Назначение настройки СМЕЩЕНИЕ	Тип канала	Краткое описание
control_word	Слово управления	не используется	О,Н	<p>Выполнить одну из команд:</p> <p>D7 – Pause – временный останов подачи по всем компонентам и останов работы системных таймеров.</p> <p>D6 – Continue – выход из состояния, вызванного командой Pause и продолжение цикла дозирования.</p> <p>D5 – Setting zero – принудительная установка нуля системы; используется только для выхода из нештатных ситуаций или при организации рабочего цикла под управлением Trace Mode</p>

Название команды в TRACE MODE	Название в спецификации протокола	Назначение настройки СМЕЩЕНИЕ	Тип канала	Краткое описание
				<p>5.</p> <p>D4 – Start comp4 – запуск дозирования только 4-го компонента.</p> <p>D3 – Start comp3 – запуск дозирования только 3-го компонента.</p> <p>D2 – Start comp2 – запуск дозирования только 2-го компонента.</p> <p>D1 – Start comp1 – запуск дозирования только 1-го компонента.</p> <p>При получении команд Start comp N «Доза-4ТМ» выполнит дозирование по N-ному компоненту вплоть до завершения работы таймера t и перейдет в фазу ожидания следующей команды (фазы - 6,12,18,24).</p> <p>D0 – Unload – разгрузка бункера.</p> <p>D0-D7 обозначают биты младшего байта канала</p>
read_input	Read input	не используется	I,F	Считать оцифрованное и нормированное значение сигнала тензодатчика.

Название команды в TRACE MODE	Название в спецификации протокола	Назначение настройки СМЕЩЕНИЕ	Тип канала	Краткое описание
read_current_dose	Read current dose	0 – доза 1-го компонента; 1 – доза 2-го компонента; 2 – доза 3-го компонента; 3 – доза 4-го компонента;	I,F	Считать текущие результаты цикла дозирования одного компонента.
read_status_word	Read word status	0 – байты 1 и 2 (1 – старший) 1 – байты 3 и 4 (3 – старший)	I,H	Считать слово состояния цикла дозирования.
read_cal_data	Read cal data	0 – байты 1 и 2 (1 – старший) калибровочной информации; 1 – байты 3 и 4 (3 – старший) и т. д.	I,F	Считать калибровочные настройки.
load_out_rg	Load OUT RG	не используется	O,H	Запись в выходной регистр « Доза4ТМ ».
load_recipe_comp	Load recipe comp[1-4]	0 – 1-й компонент; ... 3 – 4-й компонент	O,F	Загрузка рецептурного значения N-го компонента (16-бит).
load_limit_comp	Load limit comp[1-4]	0 – 1-й компонент; ... 3 – 4-й компонент	O,F	Загрузка порога перехода на пониженную скорость для N-го компонента (16-бит).
load_timers_t_T	Load timers t,T	не используется	O,H	Загрузка констант таймеров t,T. t – старший байт канала, T – младший байт канала

Название команды в TRACE MODE	Название в спецификации протокола	Назначение настройки СМЕЩЕНИЕ	Тип канала	Краткое описание
load_timers_U_F	Load timers U,F	не используется	O,H	Загрузка констант таймеров U,F. U – старший байт канала, F – младший байт канала
load_timers_S_D	Load timers S,D	не используется	O,H	Загрузка констант таймеров S,D. S – старший байт канала, D – младший байт канала
load_timers_R	Load timer R	не используется	O,F	Загрузка константы таймера R (16-бит).
load_timers_M	Load timer M	не используется	O,F	Загрузка константы таймера M (16-бит).
load_timers_K	Load timer K	не используется	O,F	Загрузка константы таймера K (16-бит).
start_cycle	Start cycle	не используется	O,H,F	Запуск процесса дозирования, происходит при записи отличного от 0 значения.
load_cal_data	Load cal data	не используется	O,H,F	Загрузить калибровочные настройки; настройки всегда загружаются из файла Wx.txt , находящегося в каталоге проекта; x – сетевой адрес «Доза4ТМ» в диапазоне [0..255], загрузка происходит при записи отличного от 0 значения.
read_last_error	нет	не используется	I,H	Последняя ошибка, которую вернул прибор, 0, если ошибок не было
clear_error	нет	не используется	O,F,H	Сбросить в 0 последнюю ошибку,

Название команды в TRACE MODE	Название в спецификации протокола	Назначение настройки СМЕЩЕНИЕ	Тип канала	Краткое описание
		ся		сброс происходит при записи отличного от 0 значения.
store_cal_data	нет	не используется	O,F,H	При записи ненулевого значения в этот канал происходит сохранение калибровочных настроек в файл Wx.txt , находящийся в каталоге проекта, где x – сетевой адрес «Доза4ТМ» в диапазоне [0..255]. Если калибровочные настройки не были считаны каналом с командой read_cal_data , то будет выставлен флаг аппаратной неадекватности.

В каталоге проекта может находиться текстовый файл **doza4tm.cfg**, содержащий единственную строку вида:

```
transaction_delay=<число>
```

Число в этом файле определяет, сколько миллисекунд МРВ будет ждать после получения ответа от дозатора до отправки очередной команды. Если такого файла в каталоге проекта нет, используется задержка в 50 мс.

В драйвере предусмотрена возможность протоколирования работы. Для включения этой функции необходимо в каталоге проекта Trace Mode создать файл с именем **_logger** (без расширения). В файле должны быть следующие строки:

```

;=====
;0-все сообщения
;1-только предупреждения и ошибки
;2-только ошибки
    level=0
;=====
;максимальный размер файла в Мб
    max_size=50
;=====

```

```
; вкл/выкл протоколирование  
enabled=true
```

Строки, начинающиеся со знака ";", являются комментариями и необязательны.

Следует отметить, что протоколирование сильно замедляет работу системы. Кроме того, при включенном протоколировании размер файла протокола растет очень быстро, особенно при интенсивном обмене данными с контроллерами.

Протокол работы заносится в файл **t11.log**.

Обмен с преобразователями Tosvert фирмы TOSHIBA

Драйвер выполнен как драйвер t11.

Для последовательного интерфейса должны быть заданы следующие настройки (см. **Редактор параметров COM-порта**):

- скорость передачи – в соответствии с настройками прибора, параметр F800;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – в соответствии с настройками прибора, параметр F801.

Рекомендуемое значение таймаута ожидания ответа – 300 мс.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер COM-порта, уменьшенный на 1;
- **Устройство** – номер прибора, число от 0 до 99 в десятичном формате, в приборе задается параметром F802;
- **Номер** – communication number, идентификатор параметра в соответствии с таблицей параметров, число от 0 до FFFF в формате hex (см. описание протокола обмена по последовательному интерфейсу с преобразователями Toshiba);
- **Запрос** – тип запроса к прибору, значение выбирается из меню:
 - **Запись_ER_ASCII** – запись значения в оперативную память прибора (RAM) и в перезаписываемое ПЗУ (EEPROM), передача данных в текстовом формате;
 - **Запись_ER_Binary** – запись значения в RAM/EEPROM, передача данных в двоичном формате;
 - **Запись_RAM_ASCII** – запись значения в оперативную память прибора, передача данных в текстовом формате;
 - **Запись_RAM_Binary** – запись значения в оперативную память прибора (RAM), передача данных в двоичном формате;
 - **Чтение_ASCII** – чтение значения параметра из прибора, передача данных в текстовом формате;
 - **Чтение_Binary** – чтение значения параметра из прибора, передача данных в двоичном формате;
 - **Чтение_G_Binary** – чтение значения параметра из прибора, передача данных в двоичном формате (для двухпроводных линий связи).

Обмен с контроллерами TREI

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с контроллером по RS задаются атрибуты конфигурирования обмена в соответствии с документацией на контроллер.

Обмен с контроллерами TCS

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена с контроллером задаются атрибуты конфигурирования обмена в соответствии с документацией на контроллер.

Обмен с контроллерами ADAM по интерфейсу CAN

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер канала интерфейса CAN;
- **MAC адрес** – идентификатор запрашиваемого прибора;
- **Команда** – запрос;
- **Канал** – номер канала;
- **Слот** – номер слота.

Обмен с электросчетчиками СЭТ-4ТМ

Заводские значения параметров последовательного порта (см. **Редактор параметров СОМ-порта**) для данных электросчетчиков – 8-1-0:

- скорость передачи – 9600 бод;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – проверка на нечетность.

Рекомендуемое значение таймаута ожидания ответа – 300 мс.

Драйвер принадлежит к типу t11 и имеет 5 изменяемых настроек:

- **Port** – номер СОМ-порта, уменьшенный на 1;
- **Device** – номер прибора в десятичном формате, на запрос с адресом 0 отвечают все приборы;
- **Query** – тип запроса к прибору, значение выбирается из меню, список возможных запросов приведен в таблице;
- **Parameter** – параметр запроса, число в шестнадцатеричном формате, зависит от типа запроса, см. таблицу (у некоторых запросов нет параметров и нужно оставить нулевое значение параметра);
- **DataControl** – контроль за передаваемыми данными, значение соответствует конфигурации прибора и выбирается из меню:
 - **CRC** – добавление в конце пакета 2-байтной контрольной суммы по алгоритму MODBUS;
 - **ControlSum** – добавление в конце пакета 1-байтной контрольной суммы;
 - **NoControl** – без проверки целостности пакета.

По умолчанию, по заводским настройкам прибором проверяется CRC.

Для использования драйвера его библиотеку нужно скопировать в корневой каталог Trace Mode 6 под именем **t11s30.dll**. Для связи со счетчиком в среде разработки в слое **Источники/Приемники** создаются шаблоны каналов в группе **Пользовательские драйверы / TYPE11 / тип 4** (6 однобайтных полей настройки). В настройках канала используются первые 5 полей аналогично настройкам в 5-й версии Trace Mode, числовые значения типов запросов (3-е поле) приведены в таблице, контроль за передаваемыми данными (5-е поле): 0 – **CRC**, 1 – **ControlSum**, 2 – **NoControl**.

Код за-проса (для Trace Mode 6)	Тип запроса	Описание	Чтение/запись	Параметр
0	DevAddr	сетевой адрес	Input	—
1	NetworkFreq	частота сети	Input	—
2	Uphase	фазное напряжение	Input	номер фазы
3	Iphase	фазный ток	Input	номер фазы
4	SinDeformCoeff	коэффициент искажения синусоидальности кривой фазного напряжения	Input	номер фазы
5	TransformCoeff	коэффициенты трансформации по напряжению и току	Input	0 – коэф. транс. по напряжению, 1 – коэф. транс. по току
6	Power	мгновенная мощность	Input	байт параметра разбит на два полубайта: старший полубайт (первая цифра в hex) - вид мощности (0- актив., 1- реактив., 2- суммарная), младший полубайт - номер фазы (0 - по сумме фаз)
7	DateTime	дата и время	Input	0 – секунды, 1 – минута, 2 – час, 3 – день недели, 4 – число, 5 – месяц, 6 – год, 7 – признак зима/лето (0-лето)
8	EnergyAll	энергия от сброса	Input	0 – активная прямая (A+), 1 – активная обратная (A-),

Код за-проса (для Trace Mode 6)	Тип запроса	Описание	Чтение/запись	Параметр
				2 – реактивная прямая (R+), 3 – реактивная обратная (R-)
9	EnergyCurYear	энергия за текущий год	Input	как в предыдущем запросе
10	EnergyPrevYear	энергия за предыдущий год	Input	как в предыдущем запросе
11	EnergyMonth	энергия за месяц, номер месяца задается записью параметра 16	Input	как в предыдущем запросе
12	EnergyCurDay	энергия за текущие сутки	Input	как в предыдущем запросе
13	EnergyPrevDay	энергия за предыдущие сутки	Input	как в предыдущем запросе
14	Tariff	номер тарифа в запросах энергии, по умолчанию 1	I, O	—
15	Aconst	константа А счетчика, используется при расчете энергии	I, O	—
16	Month	номер месяца для запроса энергии за месяц	I, O	—

Обмен с контроллерами Vantage

С контроллерами Vantage возможны два типа обмена – через последовательный порт и по сети Ethernet. В последнем случае используется конвертер Ethernet/RS232, подключаемый к порту контроллера.

Для связи через последовательный порт в среде разработки Trace Mode 6 в слое **Источники /Приемники** создаются каналы группы **Пользовательские драйверы / Туре11 / тип 4** (6 однобайтных полей настройки). При этом первое поле – номер используемого СОМ-порта, уменьшенный на 1 (т.е. 0 – СОМ1, 1 – СОМ2 и т.д.). Второе поле – тип запроса (см. таблицу). Значение остальных полей зависит от типа запроса и также приведено в таблице. Для работы необходимо создать и настроить СОМ-порт в узле. Параметры настройки (см. **Редактор параметров СОМ-порта**): скорость 19200 бод, 8 бит данных, 1 стоповый бит, без контроля четности.

Для связи по сети Ethernet в среде разработки Trace Mode 6 в слое **Источники /Приемники** создаются каналы группы **Пользовательские драйверы / Туре12 / тип 4** (6 однобайтных полей настройки). При этом первое поле – идентификатор устройства в сети, каждому номеру должна однозначно соответствовать пара значений **IP-адрес:порт**, задаваемых в строке **Дополнительно** редактора шаблона канала. Второе поле – тип запроса (см. таблицу). Значение остальных полей зависит от типа запроса и также приведено в таблице. Пример – запрос уровня 1-й нагрузки 1-й станции у 1-го master-контроллера – приведен на рисунке:

Основные	
Имя	get_load1
Кодировка	TW0
Комментарий	
Параметры	
Параметр#1 (B)	0x0
Параметр#2 (B)	0x3
Параметр#3 (B)	0x1
Параметр#4 (B)	0x1
Параметр#5 (B)	0x1
Параметр#6 (B)	0x0
Направление	Input
Дополнительно	
192.168.11.122:3040	

Запрос из примера обращается к конвертеру с адресом 192.168.11.122, порт 3040, все запросы к этому же преобразователю должны иметь первый параметр (идентификатор устройства) равный 0 и такую же строку **Дополнительно**.

Тип запроса – 2-й байт	Параметры	Примечания
0 - Запрос версии прошивки	1 - master 2 - выбор параметра: 0 - версия; 1 - подверсия; 2 - день; 3 - месяц; 4 - год.	Только чтение
1 - Запрос/установка состояния светодиода	1 - master; 2 - station; 3 - LED.	
2 - Запрос/установка уровня нагрузки модуля	1 - master; 2 - enclosure; 3 - module; 4 - load.	
3 - Запрос/установка уровня нагрузки станции	1 - master; 2 - station; 3 - load.	
4 - Запрос/установка уровня нагрузки контактора	1 - contactor.	
5 - Запрос/установка параметров термостата	1 - master; 2 - station; 3 - выбор параметра: 5 - Heat Setpoint Day; 6 - Heat Setpoint Night, 7 - Cool Setpoint Day; 8 - Cool Setpoint Night; 9 - Day/Night Status, 0=Day, 1=Night; 10 - Fan Mode, 0=Auto, 1=On; 11 - Operation Mode, 0=Off, 1=Cool, 2=Heat, 3=Auto; 12 - Outdoor Temp; 13 - Indoor Temp; 48 - High Range Indoor Sensor; 49 - Low Range Indoor Sensor; 50 - Hysteresis Indoor Sensor;	

Тип запроса – 2-й байт	Параметры	Примечания
	51 - High Range Outdoor Sensor; 52 - Low Range Outdoor Sensor; 53 - Hysteresis Outdoor Sensor.	
6 – Запрос/установка состояния переключателя	1 – master; 2 – station; 3 – switch.	<state> Description 0 - Execute the switch with the off state. 1 - Execute the switch with the on state. 2 - Learn the current load values for the switch function. This is only valid for switch function types Dim, Preset_on, and Preset_toggle. 3 - Start dimming cycle on loads of a Dim function. 4 - Execute the switch emulating a switch press. 5 - Execute the switch emulating a switch release. 6 - Execute the switch emulating a switch press and release.
7 – Установка профиля нагрузки модуля	1 – master; 2 – enclosure; 3 – module; 4 – load.	Только запись
8 – Установка профиля нагрузки станции	1 – master; 2 – station; 3 – load.	Только запись
9 – Запрос/установка задержки изменения нагрузки		fade

Параметр 1 – 3-й байт в полях настройки пользовательского драйвера типа 4 (6 однобайтных полей), параметр 2 – 4-й, и т.д.

Обмен по протоколу BACnet/IP

Драйвер обмена с устройствами по протоколу **BACnet/IP** выполнен как драйвер t12.

Шаблоны каналов обмена по протоколу **BACnet/IP** создаются в группе **PLC** слоя **Источники /Приемники** (конфигурирование обмена в группе **Пользовательские драйверы** запрещено). Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), в редакторе задаются следующие параметры:

- **Тип объекта** – тип объекта по спецификации **BACnet**, значение выбирается из меню:
 - **analog-input** – аналоговый вход;
 - **analog-output** – аналоговый выход;
 - **analog-value** – аналоговое значение;
 - **binary-input** – цифровой вход;
 - **binary-output** – цифровой выход;
 - **binary-value** – цифровое значение;
 - **calendar** – календарь;
 - **command** – команда;
 - **device** – устройство;
 - **event-enrollment** – событие;
 - **file** – файл;
- **Номер объекта** – номер объекта **BACnet** в десятичном формате;
- **Код свойства** – код свойства объекта по спецификации **BACnet** в десятичном формате. Коды некоторых свойств приведены ниже в таблице;
- **Тип данных** – тип данных свойства по спецификации **BACnet**; используется при записи значения в контроллер, поэтому задается только для каналов типа OUTPUT;
- **Адрес устройства** – строка с составным адресом прибора в следующем формате:

```
<Router_IP>:<Router_Port>-  
<SubNetwork>.[+]<DeviceNumber>
```

где

- **Router_IP** – IP-адрес маршрутизатора BACnet/IP в формате **n1.n2.n3.n4**;
- **Router_Port** – номер порта; для BACnetIP обычно равен 47808 (BAC0_{hex});
- **SubNetwork** – 0 (номер внутренней подсети присваивается маршрутизатором автоматически);

- **DeviceNumber** – номер прибора в своей подсети, обычно задается с помощью набора переключателей на корпусе прибора. Если номеру прибора предшествует знак «+», то
 $\langle \text{номер объекта} \rangle = \langle \text{Номер объекта} \rangle + 256 * \langle \text{номер прибора} \rangle$

После IP-адреса и порта могут следовать следующие выражения (в качестве разделителя следует использовать точку с запятой):

- **OBJINST**=<номер объекта>
- **OBJADD**=<константа, прибавляется к номеру объекта>

Некоторые коды свойств объектов ВАСnet

Код (DEC)	Название свойства
59	low-limit
45	high-limit
85	present-value
81	out-of-service

Типы данных ВАСnet и их поддержка в текущей версии драйвера

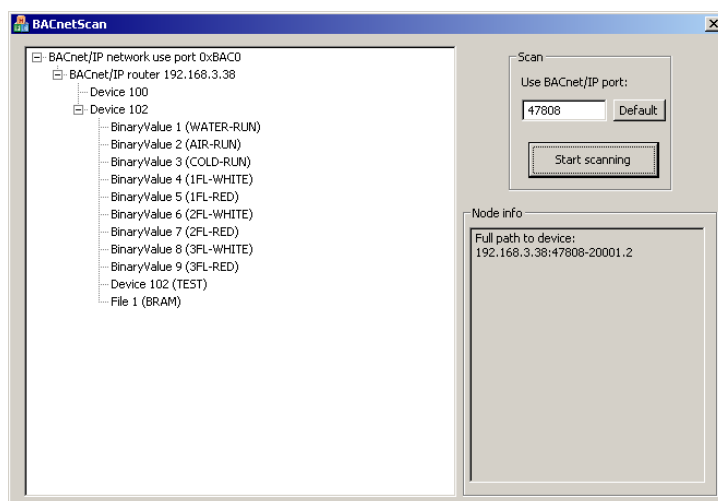
Код (DEC)	Тип данных	Поддерживается в текущей версии драйвера
0	Null	—
1	Boolean	чтение и запись
2	Unsigned Integer	чтение и запись
3	Signed Integer	чтение и запись
4	Real	чтение и запись
5	Double	чтение и запись
6	Octet String	—
7	Character String	—
8	Bit String	—
9	Enumerated	чтение и запись
10	Date	—
11	Time	—
12	BACnetObjectIdentifier	—

Наборы и типы данных свойств у каждого типа объектов различны, полные списки свойств и более подробную информацию о протоколе **BACnet** см. в «BACnet. A Data Communication Protocol for Building Automation and Control Networks.» – ANSI/ASHRAE Standard 135-2001, ISO 16484-5:2003, ISSN 1041-2336.

Утилита *BACnetScan.exe*

Для нахождения в сети устройств, поддерживающих обмен по протоколу **BACnet/IP**, и просмотра свойств этих устройств можно использовать утилиту **BACnetScan.exe**, входящую в состав Trace Mode 6.

Окно утилиты имеет следующий вид:



Перед началом процедуры сканирования сети нужно задать номер порта для опроса, по умолчанию – 47808 ($BAC0_{hex}$). Нажатием кнопки **Start scanning** запускается процесс сканирования, занимающий обычно 3-10 секунд. После окончания сканирования в левом окне утилиты строится дерево обнаруженных устройств, состоящее из списка найденных маршрутизаторов, подключенных к ним приборов и списков свойств приборов. Отдельные элементы дерева можно выбирать, при этом справа в информационном окне утилиты отображается доступная для выбранного элемента дерева информация. Так, для маршрутизатора можно просмотреть список внутренних подсетей, для контроллера – название прибора, производителя и тип модели, а также полный путь к устройству. Для отдельных свойств прибора в информационном окне также показан полный путь к прибору. Так как сканирование сети выполняется с помощью широковещательных IP-пакетов, компьютер с утилитой сканирования и приборы BACnet/IP должны находиться в одной подсети, или между разными подсетями должен быть шлюз, транслирующий широковещательные пакеты в другую подсеть.

Обмен с электросчетчиками Меркурий 230

Драйвер обмена выполнен как драйвер t11 (**t11s59.dll**).

В качестве шаблонов каналов обмена используются шаблоны **UniversalPLC_type4** (группа **Источники/Приемники / Пользовательские драйверы / TYPE11**). В МРВ корректно сконфигурированные каналы обмена имеют подтип 11 и дополнение к подтипу 59.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), в редакторе шаблона задаются следующие параметры:

- **Порт** – номер последовательного порта (0 – COM1 и т.д.);
- **Адрес** – адрес счетчика;
- **Параметр1** (95, **C2** в МРВ) – выполняемая функция (см. ниже);
- **Параметр2–Параметр4** (96, **C3–98, C5** в МРВ) – дополнительные параметры для некоторых функций;
- **Дополнительно** – задается в следующем формате:

M230 : [PWD=<pwd1>;] [PWD1=<pwd1>;] [PWD2=<pwd2>;] [ECHO]

- **pwd1** – пароль для уровня доступа 1;
- **pwd2** – пароль для уровня доступа 2;
- **ECHO** – отменяет в счетчике возврат полученного запроса перед отправкой данных (если такой режим в счетчике сконфигурирован).

Выполняемые функции (DEC):

- **0** – проверка соединения (запрос-ответ до открытия сеанса обмена, запрос содержит адрес счетчика, ответ идентичен запросу);
- **1** – команда открытия сеанса обмена, **C3** задает уровень доступа (1 или 2). В случае успешного открытия сеанса счетчик возвращает 0, в противном случае – код ошибки (записывается в канал INPUT). Если сеанс открыт, канал с **C2=1** посылает команду каждые 15 секунд (т.к. если сеанс открыт, но обмена данными нет, счетчик прерывает сеанс через 20с). Если сеанс не открыт, канал с **C2=1** посылает команду на каждом такте пересчета;
- **2** – чтение даты и времени из счетчика;
- **3** – чтение записи по заданному параметру из журнала событий счетчика. **C3** задает номер параметра в счетчике, **C4** – номер записи по указанному параметру (если **C4=0**, считывается последняя запись по параметру). Если **C3=1-6**, запись содержит две временные метки, если **C3=0x7-0x2F** – одну (см. табл. 4 «Краткого описания и принципа действия электросчетчика Меркурий 230»);
- **5** (3 в описании счетчика) – запись параметра, заданного **C3**, см. табл. 2 описания счетчика, поддерживаются параметры 0x0, 0x8,

0x2C, 0x2E (для 0x2E **C4** задает номер тарифа), 0x2F, 0x30, 0x31. Тип канала записи должен быть OUTPUT;

- **6** – запись в ОЗУ счетчика, конфигурируется аналогично **23**. Если канал имеет класс FLOAT или HEX32, записывается 4 байта, если HEX16 – 2 байта (во всех случаях тип канала должен быть OUTPUT);
- **8** – чтение параметра, заданного **C3** (см. табл. 6 описания счетчика, поддерживаются параметры 0x0, 0x2, 0x3, 0x4, 0x9, 0xA, 0xB, 0x17, 0x18, 0x1D, 0x1E и 0x20);
- **9** – чтение варианта исполнения (см. описание счетчика). Ответ содержит 9 чисел, последнее – постоянная счетчика (число импульсов в 1 кВт*ч);
- **10** (8, 0x14 в описании счетчика) – аналог **21**, но считываются зафиксированные значения;
- **11-16, 20** – чтение потребленной энергии по заданному тарифу, ответ содержит суммарное потребление и потребление по каждой из трех фаз, **C3** задает номер тарифа (если **C3=0**, считываются значения по всем тарифам):
 - **11** – энергия от сброса;
 - **12** – энергия за текущий год;
 - **13** – энергия за предыдущий год;
 - **14** – энергия за месяц с номером **C4**;
 - **15** – энергия за текущий день;
 - **16** – энергия за предыдущий день;
 - **20** – энергия за предыдущий месяц;
- **17** – чтение активной энергии по каждой из трех фаз (ответ содержит 3 числа);
- **18** – чтение утренних и вечерних пиков потребления, ответ содержит 8 чисел, **C4** задает номер месяца;
- **19** – чтение параметров последней записи в журнале событий счетчика, ответ содержит 3 числа:
 - **число1** – адрес записи;
 - **число2** – время записи;
 - **число3** – статус записи;
- **21** (8, 0x11 в описании счетчика) – чтение параметра, заданного **C3** (ответ, как правило, содержит одно число):
 - **0x0** – мощность;
 - **0x1** – напряжение;
 - **0x2** – ток;
 - **0x3** – коэффициент мощности;
 - **0x4** – частота;

- **0x5** – угол между напряжением и током;
- **0xE** – время фиксации энергии;
- **0xF** – зафиксированная энергия.

C4 задает дополнительный параметр (см. описание **BWRI**, рис.28-29 в описании счетчика);

- **22** (8, 0x16 в описании счетчика) – аналог **21**, но по всем фазам (ответ содержит несколько чисел);
- **23** – чтение ОЗУ счетчика, в атрибут 92, **I2** должен быть записан адрес начальной ячейки памяти, деленный на 2. **C3** задает дополнительные параметры:
 - значение битов 0-3 – номер памяти;
 - значение битов 4-6 – вид энергии (см. табл. 7 описания счетчика):
 - 0 – все виды (A+, A-, R+, R-), считывается 15 байт;
 - 1 – A+, считывается 4 байта;
 - 2 – A-, считывается 4 байта;
 - 3 – R+, считывается 4 байта;
 - 4 – R-, считывается 4 байта;
 - бит 7 – бит 17 адреса ячейки памяти.

Если постоянная счетчика отлична от 1000 имп./($\text{kВт}\cdot\text{ч}$), для корректного считывания ОЗУ необходим канал 11.59 с **C2=9**.

Полученные данные записываются в канал CALL.ChGroupReq или в канал 11.59 (в последнем случае в канал записывается значение, которое имеет номер **C5** в полученной выборке).

После 10 неудачных попыток открытия сеанса обмена в каналах чтения/записи данных устанавливается признак аппаратной недостоверности.

Обмен с TwinCAT (Beckhoff)

Драйвер взаимодействия МРВ с **TwinCAT** (см. <http://www.beckhoffautomation.ru>) выполнен как драйвер t12 (**t12s114.dll**), эта библиотека должна располагаться в папке установки МРВ).

Для обеспечения корректной работы должны быть установлены и зарегистрированы **TsAdsDll.dll** и другие библиотеки TwinCAT ADS API.

В качестве шаблонов каналов для связи с TwinCAT используются шаблоны **UniversalPLC_type1** (группа **Источники/Приемники.Пользовательские_драйверы.TYPE12**).

Шаблоны могут быть привязаны к числовым каналам или к каналам CALL.ChGroupReq (далее – **cgr**). Такие каналы в МРВ имеют подтип и дополнение к подтипу 12.114.

В строке **Дополнительно** шаблона обязательно должно присутствовать слово **“БЕКHOFF”**. С помощью строки **Дополнительно** может быть задана конфигурация канала:

- канал с доступом по имени:
BECKHOFF:Name=<имя_параметра>;
- канал с доступом по адресу:
BECKHOFF:ID=NN.NN.NN.NN.mm.mm;Port=<номер порта>;
или
BECKHOFF:ID=NN.NN.NN.NN.mm.mm;Port=<номер порта>;Name=<имя_параметра>;

После слова **“БЕКHOFF”** должно быть двоеточие, в качестве разделителя параметров должна использоваться точка с запятой.

Если **ID** и/или **Port** не заданы, используется локальный параметр и порт 801.

Назначение остальных атрибутов шаблона (кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**)):

- **Параметр1** – тип операции:
 - 255 – чтение кода ошибки, возвращаемого API (1 целое);
 - 16 – чтение статусов **nAdsState** и **nDeviceState** (2 целых).
 Если шаблон привязан к числовому каналу (тип INPUT/OUTPUT задает соответственно операцию чтения/записи):
 - 0 (с доступом по имени – не обязательно) – одна 4-байтовая переменная. Формат (с плавающей запятой или целочисленный) задается классом канала (соответственно

FLOAT/HEX32);

- 1, 2, 3, 4 (с доступом по адресу – не обязательно) – длина данных; для этих значений актуальны атрибуты **Параметр3** и **Параметр4**. Размер элемента – 1, 2 или 4 байта.

Если шаблон привязан к **cgr**:

- 0 – при экспорте проекта превращается в 5, 6, 7 или 8, в зависимости от типа данных аргумента;
- 5, 6, 7, 8, 9 – чтение/запись массива (канал должен быть с доступом по имени или по адресу). Длина данных – (**Параметр1 – 4**)*<**число аргументов**>. Размер элемента – 1 или 2 байта (тип данных аргумента REAL) или 4 байта (REAL или DINT);
- **Параметр2** – номер контроллера (от 0 до 255);
- **Параметр3** – **indexGroup** в интерфейсе ADS-сервера;
- **Параметр4** – **indexOffset** в интерфейсе ADS-сервера.

Ошибки, диагностируемые TRACE MODE:

- 4, ERR_RT_READ – ошибка чтения;
- 7, ERR_RT_FORMAT – доступ по имени, а имя не определено;
- 12, ERR_RT_NOTFOUND – ошибка преобразования имени в адрес;
- 17, ERR_RT_CONNECT – библиотека **TsAdsDll.dll** не загружена.

Обмен по DeviceNet

Описание драйверов DeviceNet

Сеть DeviceNet представляет собой один или несколько приборов, каждый из которых однозначно идентифицируется в сети по присвоенному ему адресу – MAC ID. С каждым из приборов возможно два типа обмена – обмен явными (explicit) сообщениями и обмен I/O пакетами.

При explicit-обмене контроллер представляется в виде объектно-ориентированной модели, и состоит из набора классов, объектов этих классов, атрибутов и методов объектов. Атрибуты могут быть как простыми (BYTE, WORD, REAL) так и составными из простых типов. Драйвер поддерживает два метода классов: `Get_Attribute_Single()` и `Set_Attribute_Single()`. Подробнее про объектную модель см. документацию DeviceNet.

В протоколе DeviceNet также предусмотрен специальный тип обмена – I/O Messaging. При данном типе обмена после установки соединения master и slave устройства обмениваются блоками данных произвольной длины – размеры принимаемого и отправляемого буфера задаются при программировании контроллера. Блок данных, посланный ведущим компьютером, интерпретируется контроллером согласно заложенной программе, он может иметь нулевую длину или игнорироваться. В ответ на запрос контроллер формирует свой блок данных и отправляет ведущему.

Файлы драйвера

Список файлов драйвера протокола DeviceNet:

- `Canbus_stub.dll` – интерфейс CAN-bus;
- `DN_Editor.dll` – редактор объектной модели устройств DeviceNet;
- `canXXmacYY.adn` – файлы с объектной моделью подключенных приборов, XX – номер интерфейса шины CAN, YY – MAC ID прибора;
- `DeviceNetSettings.dat` – двоичный файл с настройками параметров DeviceNet;
- `device_net_editor_plugin.dll` – библиотека для работы в среде разработки ТМб;
- `media9.dll` – модуль описания носителя;
- `t12s9.dll` – библиотека драйвера, обеспечивают обмен между MPB

и устройствами DeviceNet.

Файлы должны находиться в корневой директории Trace Mode.

Интерфейс CAN-bus

Библиотека интерфейса шины CAN `canbus_stub.dll` должна находиться в директории Trace Mode. Эта библиотека должна экспортировать следующий набор функций:

```
int CAN_Init(int channel, eCIBR baud_rate);
int CAN_Send(int channel, unsigned short id,
             unsigned char* data, unsigned char data_len);
int CAN_Recv(int channel, unsigned short* pId,
             unsigned char* data, unsigned char* pLen);
void CAN_Close(int channel);
```

Первым параметром каждой функции является номер канала CAN, в функции отправки и приема пакетов передаются/принимаются также идентификатор пакета, массив байт и его длина (до 8 байт). При инициализации скорость обмена задается в виде константы, согласно перечислению:

```
enum eCIBR
{
    CIBR_10K, // 0
    CIBR_20K, // 1
    CIBR_50K, // 2
    CIBR_125K, // 3
    CIBR_250K, // 4
    CIBR_500K, // 5
    CIBR_800K, // 6
    CIBR_1000K // 7
};
```

Драйвер поставляется с библиотекой для платы – интерфейса шины CAN производства ООО «Марафон» (<http://can.marathon.ru/>).

При использовании других плат пользователь должен написать библиотеку с реализацией перечисленных функций самостоятельно.

DeviceNet. Обмен explicit-сообщениями

Настройки параметров обмена

В структуре проекта контроллер DeviceNet представлен как группа, для

редактирования которой вызывается специальная форма. Все каналы, созданные в этой группе, считаются входящими в соответствующий контроллер.

Настройки устройства

CAN

Общие настройки канала CAN

Таймаут открытия соединения 3000

Таймаут закрытия соединения 1000

Общий таймаут чтения 1000

Интервал между пакетами 500

MAC ID мастера 0

Скорость 250K

MAC ID прибора 1

Добавить в группу каналы DeviceNet

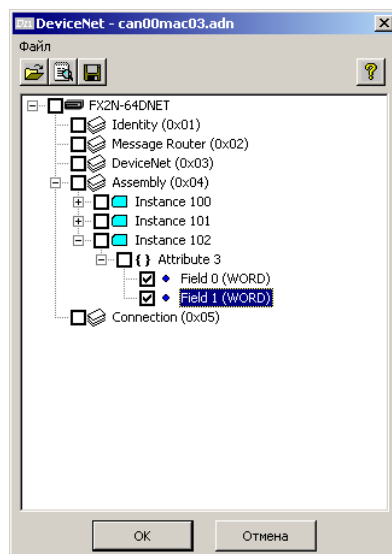
Справка

В верхней части формы расположены общие настройки интерфейса CAN. Номер используемой шины CAN выбирается в списке, в правой части задаются все параметры настройки обмена: скорость, таймауты, MAC ID инициатора обмена. Всего возможно использование до 256 каналов CAN.

В поле «MAC ID прибора» задается адрес контроллера, с которым осуществляется обмен, этот адрес и номер выбранного канала CAN автоматически передаются в настройки каждого канала группы.

Настройки интерфейсов CAN общие для всех контроллеров и всех проектов, они сохраняются в файле DeviceNetSettings.dat расположенном в директории Trace Mode.

Используя кнопку «Добавить в группу каналы DeviceNet» есть возможность создавать каналы для обмена с контроллером, при этом вызывается окно редактора объектной модели прибора, на которой можно выбрать одно или несколько полей атрибутов. Пример группы представлен на рисунке:



Настройка канала

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

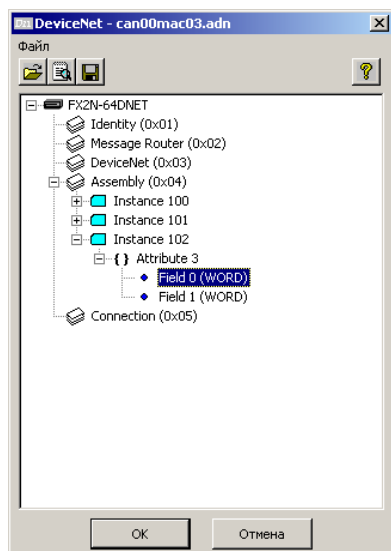
- **Номер канала CAN** – номер канала интерфейса CAN, однобайтное десятичное число;
- **MAC ID прибора** – адрес запрашиваемого прибора, однобайтное десятичное число;
- **Путь** — строка, текстовое представление пути в объектной модели контроллера, для автоматического создания пути справа от текстового поля есть кнопка «Выбрать» открывающая редактор объектной модели DeviceNet для выбора поля данных;
- **Тип данных** — номер поля в списке полей данных атрибута, значение из редактора объектной модели, однобайтное десятичное число;
- **Направление** — тип направления обмена, чтение Input или запись Output.

Поля номера канала CAN и адреса прибора недоступны для редактирования, значения приходят из настроек родительской группы (контроллера).

Редактор объектной модели устройств DeviceNet

Представление контроллера в виде объектной модели используется при обмене явными (explicit) сообщениями. Для каждого контроллера создается дерево узлов, сохраняемое в файле с именем canXXmacYY.adn, где

XX – номер интерфейса шины CAN, YY – MAC ID прибора. При использовании нескольких однотипных приборов можно создать одну модель и копированием файлов под нужными именами получить остальные.



В окне редактора изображено дерево DeviceNet, добавление и удаление элементов дерева производится через контекстное меню, вызываемое правой кнопкой мыши.

В общем случае путь к элементу выглядит так: Прибор – Класс – Экземпляр класса – Номер атрибута в экземпляре класса. Привязка каналов производится к полям атрибутов, т.е. необходимо чтобы атрибут имел как минимум одно поле данных.

В редакторе доступны функции импорта из файлов EDS, сохранение и загрузка дерева из бинарных файлов.

DeviceNet. Обмен I/O пакетами

Настройки в конфигурационном файле для I/O обмена

Конфигурационный файл для I/O обмена называется DeviceNet.ini, он должен быть расположен в корневой директории Trace Mode. Файл состоит из одной или нескольких секций, по количеству каналов шины CAN.

Имя каждой секции – CAN_n, где n – номер канала. Полный список параметров приведен ниже в таблице.

Параметр	Описание
OpenConnectionTimeout	Максимальное время ожидания соединения, мс

Параметр	Описание
CloseConnectionTimeout	Максимальное время закрытия соединения, мс
ReadTotalTimeout	Общее время приема ответа, мс
ReadIntervalTimeout	Максимальный интервал между фрагментами, мс
Master_MAC_ID	Идентификатор компьютера – инициатора запросов, 0...63
Baudrate	Скорость обмена - константа, соответствующая перечислению в описании функции CAN_Init() (см. выше)

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Описание драйверов DeviceNet** и **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Номер CAN-канала** – номер канала интерфейса CAN, однобайтное десятичное число;
- **MAC-адрес** - идентификатор запрашиваемого прибора, однобайтное десятичное число;
- **Запрос** – тип запроса к прибору, значение выбирается из меню:
 - **DeviceExchange** – канал этого типа инициирует сеанс обмена с контроллером, может быть как Input так и Output;
 - **ReadBuffSize** – размер приемного буфера, в этот буфер приходит ответ от прибора;
 - **WriteBuffSize** – длина посылаемого блока данных;
 - **ReadValue** – каналы этого типа служат для доступа к элементам приемного буфера;
 - **WriteValue** – каналы этого типа служат для доступа к элементам отправляемого буфера;
- **Смещение** – смещение переменной относительно начала блока. Настройка имеет смысл только для каналов с атрибутом **Запрос** типа ReadValue и WriteValue;
- **Тип данных** – тип переменной в блоке, настройка имеет смысл только для каналов с атрибутом **Запрос** типа ReadValue и WriteValue. Значение выбирается из меню:
 - SINT – short integer, однобайтное знаковое целое;
 - USINT – unsigned short integer, однобайтное беззнаковое целое;
 - INT – integer, знаковое двухбайтное целое;
 - UINT(WORD) – unsigned integer, беззнаковое двухбайтное целое;

- DINT – double integer, знаковое 4-байтное целое;
- UDINT(DWORD) – unsigned double integer, беззнаковое 4-байтное целое;
- LINT – long integer, знаковое 8-байтное целое;
- ULINT – unsigned long integer, беззнаковое 8-байтное целое;
- REAL – число с плавающей точкой, float;
- LREAL(double) – число с плавающей точкой двойной точности, double.

Для обмена с прибором необходимо:

- при редактировании проекта:
 - создать Output-каналы типов ReadBufferSize и WriteBufferSize;
 - создать каналы, адресующиеся к элементам обоих буферов;
 - создать канал–инициатор обмена (один);
- при работе MPB:
 - задать размеры буферов для отправки и приема;
 - заполнить буфер для отправки, если его длина больше 0;
 - обработка канала – инициатора обмена вызовет посылку контроллеру блока данных, прием ответа и запись ответа в буфер приема;
 - каналы, связывающие буфер приема с переменными нужных типов, отображают ответ контроллера.

Ограничения драйвера DeviceNet

Драйвер использует для обмена только первую группу идентификаторов DeviceNet, которая имеет максимальный приоритет. Соединение устанавливается заново при каждом запросе, и после окончания приема ответа закрывается.

Пример. Настройка обмена с контроллером Mitsubishi FX2N-16MR по протоколу DeviceNet

В примере показано, как использовать для обмена явные (explicit) сообщения и I/O пакеты.

Настройка контроллера

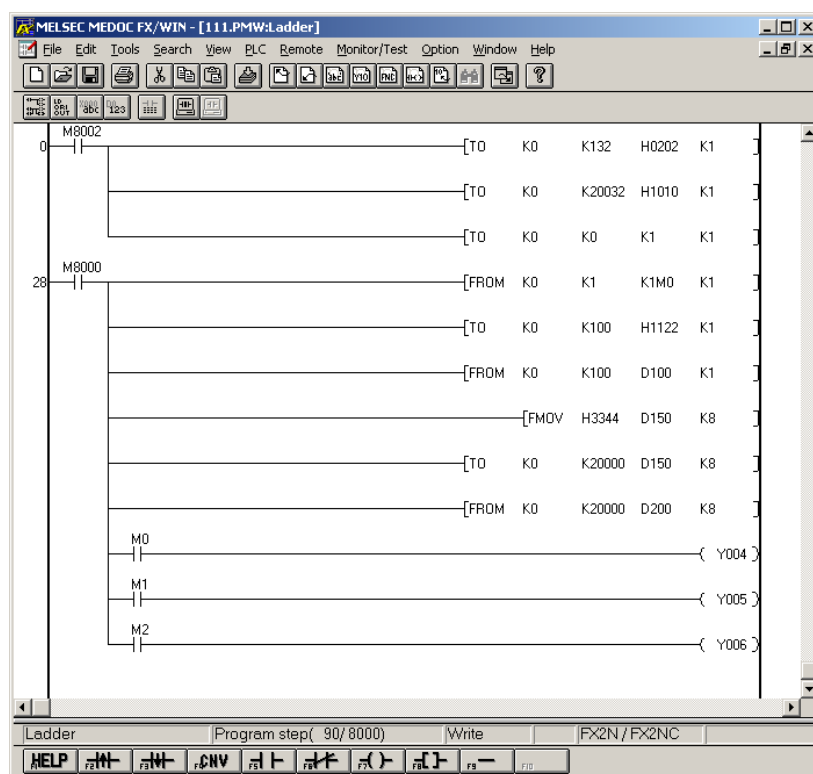
Контроллеру FX2N-16MR для обмена по сети DeviceNet необходим интерфейсный модуль FX2N-64DNET. Порядок подключения интерфейсного модуля описан в руководстве по эксплуатации «FX2N-64DNET DeviceNet Interface Block. USER'S MANUAL» (файл FX2N-64DNET E.pdf).

Не забудьте поставить терминатор на конец линии CAN (резистор 100-150 Ом), в модуль FX2N-64DNET он не включен.

Кроме физического подключения интерфейсный модуль необходимо сконфигурировать. Установим переключателями скорость обмена 125 Кбод, и MAC ID модуля – 1.

Программирование контроллера

Для того, чтобы контроллер мог обмениваться данными через интерфейсный модуль, модуль нужно активизировать. Пример программы, активизирующей интерфейс DeviceNet, приведен ниже (программа в виде лестничной диаграммы).



В программе-примере при старте контроллера задаются размеры области обмена: 16 байт для explicit сообщений, по 2 байта для приемного и передающего буфера I/O messaging, и активизируется интерфейсный модуль. При работе стадии подключения протокола DeviceNet показываются на выходах 4..6 контроллера. В регистр данных D100 записывается пришедшее значение из I/O пакета, в ответ посылается значение 1122_{hex}. В регистры данных D200..D207 записываются пришедшие по explicit обмену

значения, буфер для обратной отправки заполняется значениями 3344_{hex}. Запись программы в контроллер и просмотр содержимого регистров данных осуществляется в программе Mitsubishi MELSEC-FX.

Другие примеры программ (в т.ч. для связи между контроллерами) смотрите в руководстве по эксплуатации интерфейсного модуля - «FX2N-64DNET DeviceNet Interface Block. USER'S MANUAL». Подробное описание синтаксиса лестничных диаграмм дано в руководстве по программированию контроллеров Mitsubishi серии FX.

Настройка драйвера

Заполнение конфигурационного файла для I/O обмена **DeviceNet.ini**:

```
; Baudrate constants:
; 0 - 10K
; 1 - 20K
; 2 - 50K
; 3 - 125K
; 4 - 250K
; 5 - 500K
; 6 - 800K
; 7 - 1000K

[CAN_0]
OpenConnectionTimeout = 2000
CloseConnectionTimeout = 500
ReadTotalTimeout = 1000
ReadIntervalTimeout = 500
Master_MAC_ID = 0
Baudrate = 3
```

В данном конфигурационном файле заданы настройки канала с номером 0 интерфейса шины CAN:

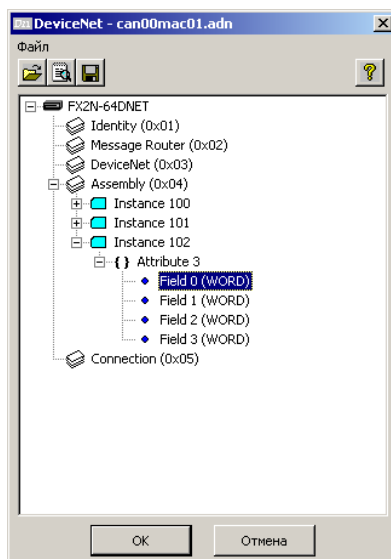
- скорость – 125 Кбод;
- идентификатор master-компьютера в сети – 0;
- максимальное время открытия соединения – 2 сек (2000 мс);
- максимальное время закрытия соединения – 500 мс;
- максимальное время принятия всего ответа – 1000 мс;
- максимальный интервал между пакетами в составном ответе – 500 мс.

В настройках группы – контроллера DeviceNet зададим параметры по

аналогии с ini-файлом, и укажем адрес прибора = 1.

Согласно документации Mitsubishi, для обмена явными (explicit) сообщениями предназначена сборка (Assembly) с номером 66_{hex} (102_{dec}). По стандарту DeviceNet атрибут с данными, содержащимися в сборке, имеет номер 3. В группе создадим 4 канала Input, адресующиеся к полям 3-го атрибута, и затем 4 канала Output для записи в те же поля.

Созданная по перечисленным выше условиям объектная модель представлена на рисунке:



Записывая значение в Output каналы, на следующем цикле пересчета мы получим те же числа в каналах Input, также пришедшие в регистры данных D200..207. Значения можно проверить в программе Mitsubishi MELSEC-FX в режиме монитора.

Для обмена I/O пакетами создадим в базе 5 каналов – 2 Output канала, задающих размеры буфера отправки и приема (на старте MPB заполним их значением 2); 1 Input канал, обращающийся к буферу приема, к значению типа WORD и смещением 0; 1 аналогичный Output канал для буфера отправки; и 1 канал – инициатор I/O обмена.

После сеанса I/O обмена, мы можем наблюдать посланное WORD значение в регистре D100, в ответ мы получаем от контроллера число 1122_{hex}.

Обмен с приборами фирмы МЗТА

Обмен с контроллером KM800 [КОНТАР]

Драйвер обмена с контроллером KM800 (КОНТАР) оформлен в виде драйвера t11.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого COM-порта (0 – COM1, 1 – COM2 и т.д.);
- **Устройство** – номер прибора (0...31 в десятичном формате);
- **Запрос** – тип запроса к прибору, выбирается из следующего меню (`plc_mzta_kontar.tmc`):
 - **Режим_PLС** – режим работы контроллера:
 - 0 – нормальный режим, функционирует алгоритм;
 - 1 – приложение отключено пользователем;
 - 2 – приложение отключено операционной системой после трехкратного подряд сбоя функционирования.

Для запуска/остановки приложения нужно записывать в канал значения 0 или 1 соответственно;

- **Режим_DO** – режим цифрового выхода, может быть автоматическим (значение 0) или ручным (значение 1). Номер выхода передается в параметре **Номер**;
- **Режим_AO** – режим аналогового выхода (номер выхода передается в параметре **Номер**):
 - 0 – автоматический режим, выход по току;
 - 1 – ручной режим, выход по току;
 - 2 – автоматический режим, выход по напряжению;
 - 3 – ручной режим, выход по напряжению.
- **AI** – значение аналогового входа; номер входа передается в параметре **Номер** и может принимать значение от 1 до 8 для контроллера MC8 или от 1 до 5 для MC5;
- **AO** – значение аналогового выхода (в процентах от максимального значения); номер выхода передается в параметре **Номер**;
- **DI** – значение цифрового входа (номер входа передается в параметре **Номер**):
 - 0 – вход разомкнут;

1 – вход замкнут.

- **DO** – значение цифрового выхода с учетом его режима. Номер выхода передается в параметре **Номер**. Значение 1 означает, что выход включен, 0 – отключен;
- **Число_Списков** – количество списков параметров алгоритма, параметр доступен только для чтения;
- **Размер_Списка** – размер списка параметров алгоритма, номер списка задается параметром **Номер**. Размер списка доступен только для чтения;
- **Знач_Параметра** – значение параметра алгоритма, номер списка задается параметром **Номер**, позиция параметра в списке задается в поле **Позиция**;
- **Формат_Параметра** – формат параметра алгоритма, доступен только для чтения. Номер списка и положение в списке задается аналогично запросу значения параметра. Расшифровка кода формата в соответствии с документацией:
 - 0 — битовый параметр, размер 1 байт;
 - 1 — знаковый целочисленный параметр, 2 байта;
 - 2 — параметр в формате с плавающей точкой, одинарная точность (float);
 - 3 — параметр времени, 2 байта (часы, минуты);
 - 4 — параметр даты, 2 байта (число, месяц);
- **Параметр_Доступ** – доступность параметра, возможен только запрос на чтение. 0 – параметр доступен только для чтения, 1 – для чтения и для записи. Номер списка и положение в списке задается аналогично запросу значения параметра;
- **Тревога** – ;
- **Номер** – в зависимости от типа запроса задает номер входа (выхода) или номер списка. Число в десятичном формате, диапазон значений зависит от прибора. Нумерация каналов ввода/вывода начинается с 1, нумерация списков – с 0;
- **Позиция** – номер параметра в списке, число в десятичном формате, нумерация начинается с 0.

Все каналы могут иметь тип INPUT и OUTPUT.

Атрибут **Номер** не используется при запросе режима работы контроллера и запросе количества списков параметров, значение его должно быть равно 0.

Атрибут **Позиция** не используется во всех запросах, кроме запросов параметров алгоритма, его значение должно быть равно 0.

Для инициализации драйвера необходимо создать хотя бы один канал INPUT с любым запросом к нужному прибору.

Обмен с приборами серии МИНИТЕРМ

Универсальный драйвер обмена с приборами серии МИНИТЕРМ оформлен в виде драйвера t11.

Обращение к параметрам ОЗУ прибора осуществляется на основе адресов параметров, поэтому в директории RTM и директории проекта необходимо создать дополнительные текстовые файлы.

Список приборов с указанием соответствия номеров приборов их типу (к одному и тому же COM-порту можно подключить до 16 приборов) хранится в файле **Miniterm_devices.tbl**, расположенном в папке проекта и имеющем следующий формат:

```

C1    N1    T1
C1    N2    T2
...     ...     ...
Сi    Nj    Tk

```

где **C** – номер COM-порта, к которому подключен прибор (0-COM1, 1-COM2 и т.д. как в настройках канала), **N** – номер прибора (0..15), **T** – тип прибора (0..255). Строки, не соответствующие указанному формату, не анализируются.

Таблица коэффициентов преобразования для каждого типа прибора задается в файле **Miniterm_typeX.tbl**, где **X** – номер типа (0..255) (если, например, все подключенные приборы имеют один и тот же тип, потребуется один файл **Miniterm_typeX.tbl**). Файлы с таблицами коэффициентов должны быть расположены в корневой директории Trace Mode. Файлы имеют следующий формат (в качестве разделителей полей используются пробелы):

```

Addr Coeff_write Coeff_read Param_size

```

где

- **Addr** – адрес параметра (HEX);
- **Coeff_write** и **Coeff_read** – коэффициенты для записи и чтения соответственно (DEC);
- **Param_size** – количество памяти для хранения параметра во внешнем ОЗУ.

Если параметр доступен только для записи или только для чтения, вместо соответствующего коэффициента указывается число -1. При размере параметра, равном 6 (троированный формат), запрос к ОЗУ выполняется со смещением на 2 байта. При размере параметра, равном 2, смещения нет.

Все строки, не соответствующие указанному формату, не анализируются.

Используемый последовательный порт должен иметь следующие параметры (см. **Редактор параметров СОМ-порта**):

- скорость обмена – 1200 бод;
- количество информационных битов – 8;
- количество стоповых битов – 2.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер СОМ-порта (0 – СОМ1, 1 – СОМ2 и т.д.), один байт в десятичном формате;
- **Значение** – номер прибора, один байт в десятичном формате;
- **Адрес** – адрес в приборе, два байта в формате HEX. Для внутреннего ОЗУ прибора старший байт равен 0x00;
- **Тип запроса** – тип запроса, выбирается из следующего меню:
 - однобайтовое значение из внутреннего ОЗУ;
 - двухбайтовое значение из внешнего ОЗУ;
 - переключение режима прибора (только для каналов OUTPUT).

Обмен с приборами ДАСУ

БОИ-М2Г

Драйвер обмена с контроллером ДАСУ БОИ-М2Г оформлен в виде драйвера t11.

Для используемого порта должны быть заданы следующие настройки (см. **Редактор параметров СОМ-порта**):

- скорость передачи – 2400/4800/9600 (в зависимости от параметров настройки БОИ в меню уставок);
- число информационных бит = 8;
- число стоповых бит = 1;
- четность – НЕТ.

Рекомендуемое значение таймаута ожидания ответа – 300 мс. При выборе таймаута меньше рекомендованного значения значительно увеличивается вероятность появления ошибок приема.

Протокол обмена с приборами ДАСУ предусматривает использование пароля для доступа к контроллеру. Если необходимо использовать пароли, их список задается в файле **DASU_BOI_passwords.txt**. В файле анализируются строки вида:

```
Abonent1 Password1
Abonent2 Password2
...
AbonentN PasswordN
```

где **Abonent** и **Password** — числа в десятичном формате. Строки, не соответствующие данному формату, не анализируются. Если пароль не задан, абоненту ставится в соответствие пустой пароль (значение 0).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер СОМ-порта, уменьшенный на 1;
- **Пустой** – незначащее поле, введено для совместимости с моделью внешних драйверов TraceMode, значение в этом поле всегда должно быть 0;
- **Абонент** – номер абонента, двухбайтовое число в десятичном формате;
- **Запрос** – код запроса к прибору; значение этого параметра выбирается из следующей таблицы:

NN запроса	Описание	Тип канала (Input/Output)
0	Текущий день	I
1	Текущий месяц	I
2	Текущий год	I
3	Текущий час	I
4	Текущая минута	I
5	Текущая секунда	I
6	Тип системы учета	I
7	Текущее состояние (код неисправности)	I
8	Тип абонента	I
9	Номер версии программы	I
10	Глубина часового архива	I
11	Номер сервера	I
12	Минимальная единица архивирования	I
13	Количество попыток дозвона	I
14	Активные температурные (резистивные) каналы по битам	I
15	Активные (токовые) каналы давления по битам	I
16	Глубина суточного архива	I
17	Отчетный час для суточного архива	I
18	Метрологические параметры - тип каналов по битам	I
19	Метрологические параметры - 1 канал	I
20	Метрологические параметры - 2 канал	I
21	Метрологические параметры - 3 канал	I
22	Метрологические параметры - 4 канал	I
23	Метрологические параметры - 5 канал	I
24	Метрологические параметры - 6 канал	I
25	Метрологические параметры - 7 канал	I
26	Метрологические параметры - 8 канал	I
27	Метрологические параметры - 9 канал	I
28	Метрологические параметры - 10 канал	I
29	Метрологические параметры - 11 канал	I
30	Метрологические параметры - 12 канал	I
31	Метрологические параметры - 13 канал	I
32	Метрологические параметры - 14 канал	I

NN запроса	Описание	Тип канала (Input/Output)
33	Текущий параметр - приведенный расход газа G	I
34	Текущий параметр - избыточное давление газа P _{изб}	I
35	Текущий параметр - температура газа t	I
36	Текущий параметр - плотность газа	I
37	Текущий параметр - перепад давления на диафрагме	I
38	Текущий параметр - код неисправности	I
39	Суммарный объем газа	I
40	Суммарное время наработки в исправном состоянии	I
41	Суммарное время во включенном состоянии	I
42	День для получения часового архива	I, O
43	Месяц для получения часового архива	I, O
44	Год для получения часового архива	I, O
45	Час для получения часового архива	I, O
46	Объем газа за час	I
47	Среднее избыточное давление за час	I
48	Средняя температура газа за час	I
49	Время наработки в исправном состоянии за час	I
50	Время во включенном состоянии за час	I
51	Код ошибки за час	I
52	День для получения суточного архива	I, O
53	Месяц для получения суточного архива	I, O
54	Год для получения суточного архива	I, O
55	Объем газа за сутки	I
56	Среднее избыточное давление за сутки	I
57	Средняя температура газа за сутки	I
58	Время наработки в исправном состоянии за сутки	I
59	Время во включенном состоянии за сутки	I
60	Код ошибки за сутки	I
61	Номер записи в архиве	I, O
62	День архива, соответствующий записи	I
63	Месяц архива, соответствующий записи	I

NN запроса	Описание	Тип канала (Input/Output)
64	Год архива, соответствующий записи	I
65	Час архива, соответствующий записи	I
66	Минута архива, соответствующая записи	I
67	Секунда архива, соответствующая записи	I
68	Значение CO ₂ из архива	I
69	Значение N ₂ из архива	I
70	Значение ρ _c из архива	I
71	Значение φ _c из архива	I

ТВА-1

Драйвер обмена с контроллером ДАСУ ТВА-1 оформлен в виде драйвера t11.

Для используемого порта должны быть заданы следующие настройки (см. **Редактор параметров СОМ-порта**):

- скорость передачи – 1200;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – НЕГ.
- тип управления передатчиком – **DTR,RTS=on**

Рекомендуемое значение таймаута ожидания ответа – 3500 мс. Устойчивый обмен с прибором при запросе части параметров возможен и при меньших значениях данного таймаута. Указанное значение соответствует максимально возможному времени формирования ответа в приборе.

Протокол обмена с приборами ДАСУ предусматривает использование пароля для доступа к контроллеру. Если необходимо использовать пароли, их список задается в файле **DASU_TVA_passwords.txt**. В файле анализируются строки вида:

```

Abonent1 Password1
Abonent2 Password2
...
AbonentN PasswordN

```

где **Abonent** и **Password** — числа в десятичном формате. Строки, не соответствующие данному формату, не анализируются. Если пароль не задан, абоненту ставится в соответствие пустой пароль (значение 0).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**

на), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер COM-порта, уменьшенный на 1;
- **Тип** – тип запроса к прибору, значение выбирается из меню. Возможны запросы итоговых, часовых и суточных значений. При запросах текущих параметров прибора (время, текущее состояние и т.п.) необходимо указать тип запроса **Итоговый**;
- **Абонент** – номер абонента, двухбайтовое число в десятичном формате;
- **Запрос** – код запроса к прибору, значение выбирается из следующей таблицы:

NN запроса	Описание	Итоговые	Часовые	Суточные
0	День	I	I, O	I, O
1	Месяц	I	I, O	I, O
2	Год	I	I, O	I, O
3	Час	I	I, O	-
4	Минуты	I	-	-
5	Секунды	I	-	-
6	Тип системы учета	I	-	-
7	Текущее состояние (код неисправности)	I	-	-
8	Тип абонента	I	-	-
9	Номер версии программы	I	-	-
10	Глубина архива	I	-	-
11	Номер сервера	I	-	-
12	Средняя температура ПВ	-	I	I
13	Средняя температура ОВ	-	I	I
14	Средняя температура ГВС	-	I	I
15	Масса воды в ПВ	I	I	I
16	Масса воды в ОВ	I	I	I
17	Масса воды на ГВС	I	I	I
18	Общее количество тепла	I	I	I
19	Количество тепла на отопление	I	I	I
20	Время в исправном состоянии	I	I	I
21	Время с перегрузкой по расходу	I	I	I
22	Код неисправности	-	I	I

Пояснения к таблице.

Для запросов с 0 по 11 для получения текущих значений параметр **Type** задается как **Итоговый**. Параметр **Type** со значением **Часовой** или **Суточный** используется для чтения и

установки даты архива – соответственно часового или суточного. Если существует хотя бы один канал с кодом запроса 0-11 и типом **Итоговый** (текущие параметры прибора), то дата архива задается по умолчанию как текущие дата и время.

Обмен с контроллерами фирмы ОВЕН

Контроллеры ПКП1, СИ8, ТРМ101, ТРМ138

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер СОМ-порта, уменьшенный на 1 (93, **СО**);
- **Адрес** – адрес прибора (92, **I2**). Используется 11-битовая адресация, если **Адрес** > 256 или **Адрес** & 0x1000 = 1, в противном случае используется 8-битовая адресация;
- **Тип** – тип данных переменной, значение выбирается из меню:
 - **1,2-int, 3-6-float** – интерпретация ответа на запрос этого типа зависит от длины полученных данных:
 - 1 или 2 байта – соответственно 1- или 2-байтовое целое;
 - 3 или 5 байт – float PIC-формата (3 байта);
 - 4 или 6 байт – float IEEE-формата (4 байта);

Запись:

- канал HEX32 типа OUTPUT – 2 байта (целое);
- канал FLOAT типа OUTPUT – float PIC-формата;

- **1-6 байт - int** – 1-4 – соответственно 1-, 2-, 3- или 4-байтовое целое, 5 – 3-байтовое целое, 6 – 4-байтовое целое (только чтение);
- **Время** – время (только чтение);
- **float** – чтение:
 - 1 или 2 байта – как **Bin float** (см. ниже);
 - 3 или 5 байт – float PIC-формата;
 - 4 или 6 байт – float IEEE-формата.

Запись:

- канал FLOAT типа OUTPUT – float IEEE-формата;

- **Байт0 ... Байт3** – при чтении из ответа извлекается соответственно байт 0 ... байт 3 (как целое).

Запись (**Байт0**):

- канал HEX32 типа OUTPUT – 6 бит;
- канал FLOAT типа OUTPUT – 1 байт float.

Запись (**Байт1** или **Байт2**):

- канал HEX32 типа OUTPUT – 12 бит;

- канал FLOAT типа OUTPUT – 2 байта float.

Запись (**Байт3**):

- канал HEX32 типа OUTPUT – 8 бит;

- **Bin-Dec float** – знаковое число с односторонней десятичной точкой, мантисса двоично-десятичная (только чтение);
- **Bin float** – знаковое число с односторонней десятичной точкой, мантисса двоичная (только чтение);
- **Чтение ошибки** – чтение ошибки;
- **Систем. адрес** – только для разработчиков TRACE MODE;
- **Символ** – чтение символьной строки (записывается в атрибут 80, **CMNT**);
- **Код ошибки** – код последней ошибки;
- **Идентификатор** – идентификатор переменной. Значение этого атрибута должно быть 0 при адресации по имени переменной (задается в поле **Дополнительно**);
- **Дополнительно** – имя переменной или **<имя переменной>:<индекс>**.

Контроллеры УКТ38, TPM-32, 33, 34, 38

Драйвер оформлен как драйвер t12.

В директории MPB должен быть размещен файл **OWEN-RS232.ini**, в котором для каждого используемого COM-порта создаются 2 строки вида:

```
COM1_init_string = baud=9600 parity=n data=8
stop=1
COM1_read_timeout = 1000
```

Строка инициализации задается аналогично параметрам команды **mode** ОС Windows, таймаут задается в миллисекундах.

Для задания номера библиотеки используется параметр **MediaNumber**, по этому номеру библиотеку описания носителя загружают драйверы t12.

Все вышеописанные параметры находятся в секции **Config** INI-файла.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер COM-порта, уменьшенный на 1;
- **Запрос** – тип запроса к прибору (см. ниже);
- **КаналАС2** – канал преобразователя АС-2, значение от 1 до 8;
- **Старт/поз** – стартовый адрес или позиция в массиве, в зависимости от типа запроса;
- **Атрибут** – атрибут, тип представления числа – знаковое или без-

знаковое, нужное значение выбирается из меню.

Доступные варианты запросов:

- **Порт** – чтение состояния портов приборов ТРМ, номер порта задается в поле **Старт/поз** и принимает значения от 0 до 3;
- **Байт** – чтение байта из памяти прибора, начальный адрес задается в поле **Старт/поз**, адрес однобайтный в hex;
- **Слово** – чтение слова (2 байт) из памяти прибора, начальный адрес задается в поле **Старт/поз**, адрес однобайтный в hex;
- **buff16** – чтение из прибора массива из 16 байт, начальный адрес массива задается в поле **Старт/поз**, адрес однобайтный в hex. Массив читается во внутренний буфер драйвера и его элементы доступны в виде 2-байтных числа через запрос **buff16_val**;
- **buff32** – чтение из прибора массива из 32 байт, начальный адрес массива задается в поле **Старт/поз**, адрес однобайтный в hex. Массив читается во внутренний буфер драйвера и его элементы доступны в виде 2-байтных чисел через запрос **buff32_val**;
- **buff16_val** – получение значения из считанного 16-байтного массива, значение 2-байтное, индекс элемента массива задается в поле **Старт/поз** и принимает значения от 0 до 7;
- **buff32_val** – получение значения из считанного 32-байтного массива, значение 2-байтное, индекс элемента массива задается в поле **Старт/поз** и принимает значения от 0 до 15;

Для всех запросов кроме **buff16** и **buff32** можно выбрать представление числа (знаковое или беззнаковое) в поле **Атрибут**.

Все каналы могут быть только типа **INPUT** – значения доступны только для чтения.

Соответствующие внутренним переменным приборов адреса памяти можно найти в документации к каждому прибору.

Для правильной работы драйвера необходимо указать номер библиотеки описания носителя в конфигурационном файле **OWEN-RS232.ini**.

Пример файла **OWEN-RS232.ini**:

```
[Config]
MediaNumber = 1
COM1_init_string = baud=96 parity=n data=8
stop=1
COM1_read_timeout = 1000
COM2_init_string = baud=9600 parity=e data=8
stop=2
COM2_read_timeout = 2000
```

Контроллер МПР-51

Драйвер оформлен как драйвер t12.

В директории MPB должен быть размещен файл **OWEN-RS232.ini**, в котором для каждого используемого COM-порта создаются 2 строки вида:

```
COM1_init_string = baud=9600 parity=n data=8  
stop=1  
COM1_read_timeout = 1000
```

Строка инициализации задается аналогично параметрам команды **mode** ОС Windows, таймаут задается в миллисекундах.

Для задания номера библиотеки используется параметр **MediaNumber**, по этому номеру библиотеку описания носителя загружают драйверы t12.

Все вышеописанные параметры находятся в секции **Config** INI-файла.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Номер порта** – номер COM-порта, уменьшенный на 1;
- **КаналАС2** – канал преобразователя АС-2, значение от 1 до 8;
- **Запрос** – тип запроса к прибору, значение выбирается из меню:
 - **Тпрод** – температура продукта;
 - **Тсух** – температура сухого термометра;
 - **Твлаж** – температура влажного термометра;
 - **Влажность** – влажность.

Все значения доступны только для чтения.

Для правильной работы драйвера необходимо указать номер библиотеки описания носителя в конфигурационном файле **OWEN-RS232.ini**.

Обмен с контроллерами фирмы ВИБРАТОР

Обмен с контроллерами Ф1765, Ф1766

Тип драйвера обмена с контроллерами Ф1765 и Ф1766 – t11.

Для последовательного интерфейса должны быть заданы следующие настройки (см. **Редактор параметров СОМ-порта**):

- скорость обмена с прибором – 19200 бод;
- число информационных бит – 8;
- число стоповых бит – 1;
- четность – НЕТ;
- таймаут рекомендуется указывать не меньше 3000 мс, так как при записи значений в прибор время задержки ответа может быть достаточно большим (порядка 1 с).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер СОМ-порта, уменьшенный на 1, 1 байт DEC;
- **Номер устройства** – номер прибора, 1 байт в формате DEC;
- **Тип устройства** – тип прибора (F1765 или F1766);
- **Запрос** – тип запроса, значение выбирается из меню:
 - **Результат** – результат измерения;
 - **Ниж_граница** – уставка У1;
 - **Верх_граница** – уставка У2;
 - **Состояние*** – состояние канала (0-выкл, 1-вкл);
 - **Тип сигнала*** – признак вида измеряемого сигнала (1 - напряжение);
 - **Тип прибора*** – обозначение исполнения прибора (1 или 2);
 - **Номер_диапазона*** – код диапазона измерения (1..6);
 - **Настройка*** – вид уставки;
- **Канал** - номер канала, от 1 до 8 (для Ф1765 - только 1), 1 байт в формате DEC.

Каналы с запросами **Ниж_граница**, **Верх_граница**, **Номер_диапазона** могут быть типа **OUTPUT** (**Номер_диапазона** - только для Ф1766)

* – только для прибора Ф1766.

Обмен с устройствами фирмы Danfoss

Обмен с частотными преобразователями

Обмен данными с частотными преобразователями **VLT2800**, **ULT5000**, **VLT6000** реализуется через последовательный интерфейс **RS-485** по протоколу **FC protocol**. Для настройки протокола используются параметры 500-514 преобразователей.

Драйвер позволяет считывать и управлять значениями всех параметров частотных преобразователей с 1 по 640. Подробное описание этих параметров изложено в руководстве пользователя соответствующего частотного преобразователя.

При настройке COM-порта (см. **Редактор параметров COM-порта**) необходимо указать управление приемо-передатчиком **DTR=on,RTS(auto)**. Используемый конвертор RS-232/RS-485 должен поддерживать этот режим (I7520 не поддерживает).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), в редакторе шаблона канала обмена надо указать номер используемого порта в формате HEX (атрибут **Порт**: 0 – COM1, 31 – COM32), адрес контроллера (атрибут **Адрес**) и номер параметра (атрибут **PNU**).

Для чтения или записи статуса и **reference**, а также получения информации об ошибке преобразователя используются каналы с **PNU=0**. Для работы со статусом значение атрибута **IND** надо установить равным 0, для **reference** – 1 (необработанное значение **reference**) или 2 (значение в процентах), а для контроля ошибки – 3.

Запросы в преобразователь по каналам с **PNU=0** не формируются. Информация о статусе, **reference** или ошибке берется из запросов других параметров. Поэтому в базе каналов должен присутствовать хотя бы еще один канал, запрашивающий значение любого параметра преобразователя, кроме статуса, **reference** или ошибки.

Обмен с тепловычислителями Эксперт-Z

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;

- **Код операции** – выполняемая операция:
 - 0 – считать текущие данные (значение **Номер** должно быть равно 0);
 - 1 – считать конфигурацию устройства (значение **Номер** должно быть равно 0);
 - 2 – управление чтением из архива устройства по значению канала (тип канала должен быть OUTPUT):
 - 1, 0, 1 – соответственно предыдущая, текущая или следующая запись;
 - DD + 256 * (MM + 256 * YY)** – из архива считывается первая запись, время которой равно или больше времени, заданного данным значением. Канал должен быть класса FLOAT;
 - 3, 4 – считать соответственно среднечасовое/среднесуточное значение указанного параметра из архива (если каналы пишутся в архив TRACE MODE, то время в них – от считанной записи).

Общее количество обрабатываемых каналов, по каждой из настроек **Код операции**, составляет:

- 60 для каналов с **Код операции**=0;
 - не более 134 для каналов с **Код операции**=1;
 - 139 для каналов с **Код операции**=3, **Код операции**=4.
- **Модуль** – адрес устройства (1-FFh);
 - **Номер** – номер переменной (полный перечень см. ниже).
- Количество и нумерация параметров, считываемых каналами с соответствующими настройками **Номер**, соответствуют перечню параметров, указанному в документации на протокол прибора, за исключением:
- значения части параметров конфигурации, имеющих представление в виде ASCII-символов, а также параметров, зарезервированных разработчиком для последующей модификации, заменяются значениями индекса собственной настройки **Номер**;
 - значения настройки **Номер** для каналов считывания часового (настройка **Код операции**=3) и дневного (настройка **Код операции**=4) архивов, задаются значениями, равными **удвоенным номерам параметров**. Таким образом, допустимыми значениями настройки **Номер** для данных каналов являются только четные значения;
- **CNV** – дополнительная настройка для выбора способа представления параметров, имеющих 6-байтовый формат:
 - при **CNV**=0 преобразование осуществляется по схеме:

$$I < 256, v = I + D/65536;$$

$$255 < I < 65536, v = I + \text{High-Byte}(D) / 256;$$

$$I > 65535, v = I;$$

где:

I – 4-байтовое целое значение старших байтов числа,
D – 2-байтовое целое значение младших байтов числа,
V – результат преобразования;

- при **CNV=3** преобразование осуществляется по схеме:
 $V = I // 4\text{-байтовое целое}$
- при **CNV=4** преобразование осуществляется по схеме:
 $V = D // 2\text{-байтовое целое}$

Запись в архив TRACE MODE меток времени архивных данных (для каналов с настройками **Код операции**=3, 4) осуществляется с сезонным часовым смещением. Ввиду отсутствия аналогичного преобразования в самом приборе, во избежание смещение часа, подъем архива следует осуществлять глубиной до начала текущего временного сезона.

Ниже приводится полный перечень параметров по каждой из функций (символ '**' рядом с индексом параметра указывает на использование для представления значений данного канала настройки **CNV**):

- при ненулевом значении настройки **Номер** в канал считывается следующий параметр:
 - 1* – V1общ. (м³);
 - 2* – V1мгнов. за 10 сек. (м³/час);
 - 3* – V2общ. (м³);
 - 4* – V2мгнов. за 10 сек. (м³/час);
 - 5* – V3общ. (м³);
 - 6* – V3мгнов. за 10 сек. (м³/час);
 - 7* – V4общ. (м³);
 - 8* – V4мгнов. за 10 сек. (м³/час);
 - 9* – V5общ. (м³);
 - 10* – V5мгнов. за 10 сек. (м³/час);
 - 11* – Gпод. мгнов. за 1 мин. (т/час);
 - 12* – Gпод. общее (т);
 - 13* – Gобр. мгнов. за 1 мин. (т/час);
 - 14* – Gобр. общее (т);
 - 15 – Rпод. мгнов. за 1 мин.;
 - 16 – Rпод. среднее за 1 час (МПа);
 - 17 – Rобр. мгнов. за 1 мин.;

- 18 – Робр. среднее за 1 час (МПа);
 - 19 – Тпод. мгнов.;
 - 20 – Тпод. среднечасовое за час;
 - 21 – Тобр. мгнов.;
 - 22 – Тобр. среднечасовое за час;
 - 23 – ТЗ мгнов.;
 - 24 – ТЗ среднечасовое;
 - 25* – накопленная энергия;
- для функции считывания данных часового (каналы с настройкой **Код операции=3**) и дневного (каналы с настройкой **Код операции=4**) архивов при ненулевом значении настройки **Номер** в канал считывается следующий параметр:
 - 2 – .byte=0xF0 (константа);
 - 4 – тип считанной записи;
 - 6 – время записи (ss);
 - 9 – время записи (mm);
 - 10 – время записи (hh);
 - 12 – время записи (DD);
 - 14 – время записи (MM);
 - 16 – время записи (YY);
 - 18 – время работы прибора в секундах с момента разрешения архива;
 - 20 – зарезервированный параметр (не используется);
 - 22 – зарезервированный параметр (не используется);
 - 24 – температура холодной воды;
 - 26 – зарезервированный параметр (не используется);
 - 28 – P1 (в МПа);
 - 30 – P2 (в МПа);
 - 32* – счетчик массы воды (подача) M1 (в тоннах);
 - 34* – счетчик массы воды (обратка) M2 (в тоннах);
 - 36* – счетчик подпиток (M2-M1);
 - 38* – счетчик утечек (M1-M2);
 - 40* – счетчик объема воды в м³ 3-го расходомера;
 - 42* – счетчик объема воды в м³ 4-го расходомера;
 - 44* – счетчик объема воды в м³ 5-го расходомера;
 - 46 – код ошибки системы теплоснабжения за час;
 - 48 – время <V1_min (подача);

- 50 – время $>V1_max$ (подача);
 - 52 – время $<V2_min$ (обратка);
 - 54 – время $>V2_max$ (обратка);
 - 56* – сумма m_0T_0 ;
 - 58* – сумма m_0 ;
 - 60* – сумма m_1T_1 ;
 - 62* – сумма m_1 ;
 - 64* – энергия (в ГДж/Гкал);
 - 66 – зарезервированный параметр (не используется);
 - 64 – код ошибки за сутки;
 - 70* – V1 счетчика (в м³);
 - 72* – V2 счетчика (в м³);
- для функции считывания параметров конфигурации (**Код операции=1**) при ненулевом значении настройки **Номер** в канал считывается следующий параметр:
 - 1 – температура замещения (подача);
 - 2 – конфигурация канала температуры (подача);
 - 3 – тип датчика (подача);
 - 4 – температура замещения (обратка);
 - 5 – конфигурация канала температуры (обратка);
 - 6 – тип датчика (обратка);
 - 7 – температура замещения (холодная вода);
 - 8 – конфигурация канала температуры (холодная вода);
 - 9 – тип датчика (холодная вода);
 - 10* – V1_min мин. допустимый расход подачи (куб/ч);
 - 11* – V1_max макс. допустимый расход подачи (куб/ч);
 - 12* – V1_min мин. допустимый расход обр. (куб/ч);
 - 13* – V1_max макс. допустимый расход обр. (куб/ч);
 - 14* – V1_min мин. допустимый расход 3-го (куб/ч);
 - 15* – V1_max макс. допустимый расход 3-го (куб/ч);
 - 16* – V1_min мин. допустимый расход 4-го (куб/ч);
 - 17* – V1_max макс. допустимый расход 4-го (куб/ч);
 - 18* – V1_min мин. допустимый расход 5-го (куб/ч);
 - 19* – V1_max макс. допустимый расход 5-го (куб/ч);
 - 20* – цена импульса расходомера в литрах (подача);
 - 21 – максимально допустимое число периодов отсутствия им-

- пульсов (подача);
- 22* – цена импульса расходомера в литрах (обратка);
- 23 – максимально допустимое число периодов отсутствия импульсов (обратка);
- 24* – цена импульса расходомера в литрах (3-й);
- 25 – максимально допустимое число периодов отсутствия импульсов (3-й);
- 26* – цена импульса расходомера в литрах (4-й);
- 27 – максимально допустимое число периодов отсутствия импульсов (4-й);
- 28* – цена импульса расходомера в литрах (5-й);
- 29 – максимально допустимое число периодов отсутствия импульсов (5-й);
- 30 – конфигурация канала расходомера (подача);
- 31 – конфигурация канала расходомера (обратка);
- 32 – конфигурация канала расходомера (3-й);
- 33 – конфигурация канала расходомера (4-й);
- 34 – конфигурация канала расходомера (5-й);
- 35 – конфигурация системы;
- 36* – P1_max – диапазон давления датчика (в МПа);
- 37* – P2_max – диапазон давления датчика (в МПа);
- 38* – P1_min – диапазон давления датчика (в МПа);
- 39* – P2_min – диапазон давления датчика (в МПа);
- 40 – интервал измерения в секундах;
- 41 – интервал интегрирования в интервалах измерения;
- 42 – интервал архивирования в интервалах интегрирования;
- 43 – индекс (не используется);
- 44 – индекс (не используется);
- 45 – номер клеммы расходомера, отражаемый на выход F ;
- 46 – конфигурация канала давления;
- 47 – .byte=0;
- 48 – канал для выхода 4–20 мА;
- 49 – маска ошибок для срабатывания реле;
- 50 – индекс (не используется);
- 51 – .word=0;
- 52 – индекс (не используется);
- 53 – .word=0;

- 54 – индекс (не используется);
- 55 – индекс (не используется);
- 56 – индекс (не используется);
- 57 – .word=0 – архив разрешен, 1 – очистка архива/архив запрещен;
- 58 – индекс (не используется);
- 59 – .byte=0h (конст.);
- 60 – .byte=0h (конст.);
- 61 – .byte=0h (конст.);
- 62 – .byte=55h (конст.);
- 63 – текущее время прибора (ss);
- 64 – текущее время прибора (mm);
- 65 – текущее время прибора (hh);
- 66 – текущая дата прибора (DD);
- 67 – текущая дата прибора (MM);
- 68 – текущая дата прибора (YY).

Для инициализации обмена должен быть создан канал с **Номер=0**.

Для обмена необходимо настроить последовательный порт (см. **Редактор параметров СОМ-порта**).

Обмен с теплосчетчиками FP-93

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – СОМ1, ..., 31 – СОМ32). Этот параметр задается в формате HEX;
- **Модуль** – номер контроллера в формате HEX (0-FF);
- **Канал** – десятичный адрес переменной (диапазон соответствует протоколу обмена);
- **Переменная** – адрес переменной в формате HEX.

Для обмена необходимо настроить последовательный порт (см. **Редактор параметров СОМ-порта**).

Обмен с контроллерами Siemens

Протокол Profibus DP

Драйвер обеспечивает обмен данными TRACE MODE с устройствами, поддерживающими протокол **Profibus-DP**. Для работы в сети Profibus компьютер должен быть оснащен специализированной коммуникационной картой. Драйвер предназначен для использования со следующими Profibus-картами Siemens A&D:

- CP-5611;
- CP-5412;
- CP-5613;
- CP-5614.

По вопросам совместимости коммуникационных карт других производителей обращайтесь к поставщику оборудования. Драйвер поддерживает чтение входных данных и запись выходных данных.

Драйвер оформлен в виде драйвера t12. Для его корректной работы необходимо:

- установить на компьютер пакет **SIMATIC NET**, обеспечивающий связь Windows-приложений с коммуникационной картой;
- установить на компьютер пакет **STEP7** или **COMProfibus**, необходимый для настройки коммуникационной карты и программирования устройств сети Profibus.;

При отсутствии авторизации пакетов **SIMATIC NET**, **STEP7** или **COMProfibus**, Вы используете их на свой страх и риск. При отсутствии самого пакета **SIMATIC NET** драйвер не сможет работать, так как использует библиотеки API, предоставляющие доступ к ресурсам коммуникационного процессора.

- создать проект с конфигурацией аппаратной части всей сети при помощи пакета **STEP7** или **COMProfibus**;
- скомпилировать и выгрузить созданный проект в коммуникационную карту;
- выполнить конфигурирование и программирование всех устройств сети Profibus-DP;
- выгрузить созданные проекты в устройства сети Profibus-DP;
- указать для точки входа (Access Point) CP_L2_1: протокол Profibus-DP;
- в программе конфигурации драйвера задать список активных уз-

лов сети и адрес ведущего устройства.

При неверно указанной точке входа (Access Point) OPC-сервер будет успешно обмениваться данными с устройствами, а драйвер не сможет получить доступ к ресурсам коммуникационного процессора.

Правильность выполнения конфигурации можно проверить с помощью OPC-сервера, поставляемого Siemens на диске SIMATIC NET. OPC-Scout должен читать и записывать данные в устройство без сообщений об ошибках или некачественной связи. Результаты чтений должны отражать текущее состояние входов устройства, а результаты записи отражаться на выходах. По вопросам настройки и программирования устройств в сети Profibus, а также связи OPC-сервера с ними обращайтесь в службу технической поддержки Siemens A&D или к разработчику оборудования.

Успешное использование драйвера возможно только после проверки связи с удаленными устройствами через OPC-сервер и правильного указания точки входа (Access Point).

Протокол Profibus-DP предназначен для высокоскоростного (до 12Мбод) обмена данными с удаленной периферией. В соответствии с его идеологией, в сети есть ведущий узел (Master) и ведомые узлы (Slave). Обмен данными – циклический. Данные, предназначенные для обмена, разделены на «входные» и «выходные». Входные данные представляют собой состояния входов удаленных узлов. Выходные данные отсылаются к подчиненным узлам для записи их в выходы. Обращение к данным побайтное. Т.е. DP-Master может иметь N байт входных данных и M байт выходных данных. Каждый байт этих данных единственным образом связан с каким-либо Slave-устройством. Эта связь устанавливается при создании конфигурации коммуникационной карты и подчиненных устройств. Более подробную информацию по сети Profibus-DP можно получить на сайте или у поставщика оборудования. Информация на русском языке по сетям Profibus (www.automation-drives.ru/as/products/09_net/04_man/index.shtml).

Драйвер имеет доступ на чтение к входным данным и доступ на запись к выходным.

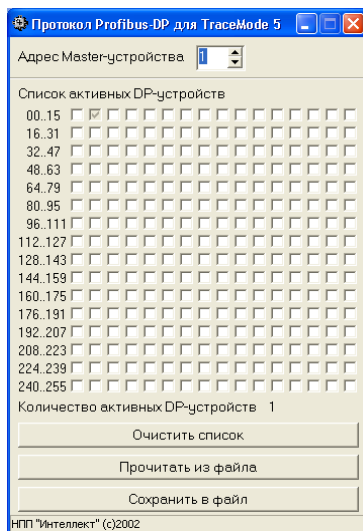
Входные и выходные данные не связаны между собой. Посылка значения X в выходной байт номер N не говорит о том, что это же значение будет иметь входной байт номер N или любой другой байт адресного пространства.

Всю необходимую информацию по конфигурации сети драйвер получает с помощью средств SIMATIC NET. Для его работы необходимо указать только адрес ведущего узла и адреса ведомых узлов, участвующих в обмене данными. Для этих целей служит утилита конфигурации **DPcon-**

fig.exe. Результаты конфигурирования сохраняются в файле **ProfiDP.cfg**.

Файл конфигурации **ProfiDP.cfg** должен быть расположен в папке узла.

Окно программы имеет следующий вид:



В нем необходимо указать адрес ведущего (Master) узла, а также отметить адреса всех узлов, присутствующих в конфигурации коммуникационной карты. После этого необходимо сохранить результаты (кнопка **Сохранить в файл**). Также есть возможность прочитать существующую конфигурацию из файла и очистить текущую конфигурацию.

Не редактируйте и не удаляйте файл конфигурации! Неверные данные, а также отсутствие файла конфигурации могут привести к общему сбою системы.

Каналы должны быть класса HEX16 или FLOAT. Тип канала (**I** или **O**) определяет его назначение – входные данные (**I**) или выходные данные (**O**).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются специфические атрибуты.

Для обращения к данным необходимо указать адрес ведомого (Slave) устройства, с которым будет происходить обмен, тип данных (**Byte**, **Word** или **Float**) и номер стартового байта. Последовательность байт, начиная со стартового байта входных/выходных данных может воспри-

ниматься драйвером как байт, слово или число с плавающей точкой. Для обращения к данным как к байтам (1 байт) необходимо указать тип данных **Byte(1)** и класс канала HEX16. Для обращения к данным как к словам (2 байта, сначала младший, потом старший), необходимо указать тип данных **Word(2)** и класс канала FLOAT. Для обращения к данным как к числам с плавающей точкой (4 байта, стандартное преобразование) необходимо указать тип данных **Float(4)** и класс канала FLOAT.

Тип данных (входные/выходные) будет определен на основе типа канала (**I/O**).

Выходные данные доступны **только для записи**, а входные – **только для чтения**. Вы не сможете прочитать значение выходного байта 00 для узла 04, создав канал для чтения (I) и указав адрес и номер байта. Вместо этого система прочитает значение байта 00 входных данных для узла 04, т.к. входные и выходные данные не связаны между собой.

При побайтной адресации (**Тип элемента=Byte(1)**) попытка записи в выходной канал значения, превышающего 255, приведет к записи 8 младших бит, т.е. младшего байта в слове.

Указание несуществующего адреса или номера байта может привести к непредсказуемым ошибкам в системе.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. Коммуникационная карта может обмениваться данными с Profibus-устройствами со значительно большей скоростью, поэтому входные данные читаются, а выходные записываются в ее внутренний буфер. Обмен данными коммуникационной карты с устройствами происходит асинхронно по отношению к обмену данными с драйвером Profibus-DP для TraceMode. Обмен данными с внешними устройствами происходит постоянно, даже при отсутствии запросов от драйвера или самого драйвера. При сбоях в обмене данными каналу выставляется признак недоверности. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Необходимо обратить внимание, что при безуспешной попытке записи значения канала Trace Mode будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

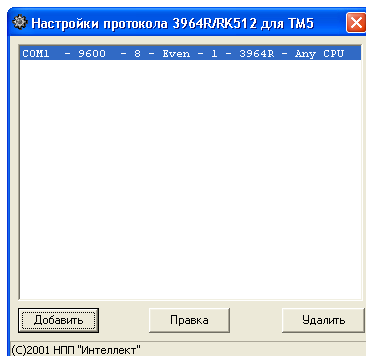
Протокол 3964R / RK512

Драйвер обеспечивает обмен данными, используя транспортный протокол **3964(R)** и реализацию протокола **RK512** для обмена с ПЛК. Драйвер поддерживает чтение регистров **I/E, Q/A, M/F, C/Z, T, DB** и запись регистров **DB**. Для обмена данными используется последовательный порт компьютера. Драйвер оформлен в виде драйвера t12.

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Последовательные порты, используемые при обмене, должны быть занесены в список **r3964.cfg**. Для этого нужно воспользоваться утилитой **R3964cfg.exe**. Каждый порт в списке будет автоматически открыт драйвером.

Порт, к которому подключены устройства **3964(R)**, не нужно настраивать в Trace Mode.

Утилита **R3964cfg.exe** предназначена для создания списка портов, которые могут использоваться при обмене:



Список сохраняется в файл **r3964.cfg**

Файл конфигурации **r3964.cfg** должен быть расположен в папке узла.

Каждая строка списка портов в основном окне утилиты состоит из семи параметров:

- номер порта. Этот параметр, уменьшенный на 1, затем нужно указать в качестве настройки **Номер** канала, который связывается с данным портом;
- скорость передачи данных (300-115200 bps). Значение по умолчанию – 9600;
- число битов данных (8 по умолчанию);
- контроль четности передачи, может принимать значения **None**, **Odd**, **Even**, Значение по умолчанию – **Even**;
- количество стоп-битов (1 или 2). Значение по умолчанию – 1.
- тип протокола: **3964** (без контрольной суммы BCC) или **3964R** (с контрольной суммой BCC). По умолчанию установлен тип **3964R**;
- процессор, с которого получать данные (для многопроцессорных систем): **AnyCPU** (любой процессор) или **CPU1...CPU4**. По умолчанию устанавливается **AnyCPU**.

Кнопки **Добавить**, **Удалить** и **Правка** основного окна утилиты предназначены для редактирования списка портов.

Если при запуске утилиты файл **r3964.cfg** уже существует, его содержимое после первичного анализа загружается в список портов. Процедура анализа отбирает все записи, которые имеют правильный номер порта. Если несколько записей имеют одинаковый номер порта, то первая из них (самая верхняя в файле) появится в списке зарегистрированных портов, а остальные будут проигнорированы и удалены. Список зарегистрированных портов сохраняется при каждом выходе из утилиты, список проигнорированных портов не сохраняется.

Каналы должны иметь класс HEX16. Тип канала (**I** или **O**) определяет его назначение – чтение данных или запись.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Номер** – номер порта (0 – COM1, ... 31 – COM32). Настройки порта должны быть предварительно заданы в утилите **R3964cfg.exe**. Если в списке утилиты нет порта с заданным номером или порт настроен в TRACE MODE как порт для обмена любыми данными, то канал будет отключен при запуске системы;
- **Тип регистра** – тип адресуемого регистра, выбирается из следующего списка:
 - **DInputs(I/E)** – элементы данных типа **Input**;
 - **DOutputs(Q/A)** – элементы данных типа **Output**;
 - **MemoryFlags(M/F)** – элементы данных типа **Memory** или **Flags**;
 - **CounterCells(C/Z)** – элементы данных типа **CounterValue**;
 - **TimeCells(T)** – элементы данных типа **TimerValue**;
 - **DataBlocks(DB)** – элементы данных типа **DataBlocks**;
- **Номер регистра** – адрес выбранного регистра в массиве регистров ПЛК. Значение параметра нужно вводить в десятичном виде (0-254 для I, Q, M, 0-255 для C, T, DB);
- **DB-номер** – номер блока **DB**, внутри которого находится запрашиваемый регистр. Влияет на работу только каналов с типом регистра **DB**. Первый блок имеет номер 1. Каналы с типом регистра **DB** и адресом 0 будут отключены при запуске. Параметр должен вводиться в десятичном виде, он не влияет на другие регистры и может иметь для них любое значение.

Каналы для всех типов регистров могут иметь тип **I**, т.е. использоваться для чтения. Для записи могут быть применены только каналы с типом регистра **DB** (ограничение протокола). Каналы типа **O** с типом регистра, отличным от **DB**, будут отключены при запуске.

Элементы с типом регистра **C, T, DV** имеют размерность 2 байта (1 слово), поэтому они читаются/пишутся по одному элементу в канал. Элементы **I, Q, M** имеют размерность 1 бит, поэтому читаются/пишутся по 2 элемента в 1 канал. Таким образом, если указать **Номер регистра=0**, то канал будет содержать в себе элементы 0 (ст. байт) и 1 (мл. байт), а если указать **Номер регистра=1**, то канал будет содержать в себе элементы 1 и 2.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. При каждой посылке ответ ожидается в течение 1 с. Если ответа за это время нет, то каналу выставляется признак недостоверности, связь с ПЛК разрывается и предпринимается попытка восстановить ее заново. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Если восстановления связи не произошло, следующая попытка установки связи будет при следующем обращении к этому порту. Необходимо обратить внимание, что при безуспешной попытке записи значения канала Trace Mode будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

Протокол Siemens S7

Драйвер обеспечивает обмен данными TraceMode с устройствами, поддерживающими протокол **Siemens S7 communications** (далее **S7**). Связь может быть организована по сети Profibus или по локальной сети Ethernet. Для работы в сети Profibus компьютер должен быть оснащен специализированной коммуникационной картой. Драйвер предназначен для использования со следующими Profibus-картами Siemens A&D:

- CP-5611;
- CP-5412;
- CP-5613;
- CP-5614.

По вопросам совместимости коммуникационных карт других производителей обращайтесь к поставщику оборудования.

Для работы по локальной сети Ethernet компьютер должен быть оснащен сетевой картой Siemens A&D или другого производителя (например, NE 2000 и т.п.)

Со стороны ПЛК должен присутствовать соответствующий коммуникационный модуль (с интерфейсом Profibus или Ethernet)

Драйвер поддерживает чтение и запись переменных типа **MW**.

Драйвер оформлен в виде драйвера t12. Для его корректной работы необходимо:

- установить на компьютер пакет **SIMATIC NET**, обеспечивающий связь Windows-приложений с коммуникационной картой (Profibus/Ethernet);
- установить на компьютер пакет **STEP7**, необходимый для настройки коммуникационной карты и программирования устройств сети Profibus;

При отсутствии авторизации пакетов **SIMATIC NET** или **STEP7**, Вы используете их на свой страх и риск. При отсутствии самого пакета **SIMATIC NET** драйвер не сможет работать, так как использует библиотеки API, предоставляющие доступ к ресурсам коммуникационного процессора.

- создать проект с конфигурацией аппаратной части всей сети при помощи пакета **STEP7**;
- скомпилировать и выгрузить созданный проект в коммуникационную карту;
- выполнить конфигурирование и программирование всех устройств сети **SIMATIC NET**;
- выгрузить созданные проекты в устройства сети **SIMATIC NET**;
- указать для точки входа (Access Point) CP_L2_1: протокол S7 через сетевую карту или карту Profibus.

При неверно указанной точке входа (Access Point) OPC-сервер, возможно, будет успешно обмениваться данными с устройствами, а драйвер не сможет получить доступ к ресурсам коммуникационного процессора.

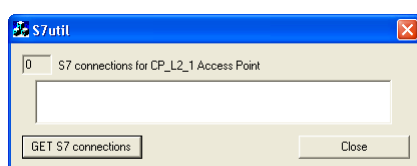
Правильность выполнения конфигурации можно проверить с помощью OPC-сервера, поставляемого Siemens на диске SIMATIC NET Software. OPC-Scout должен читать и записывать переменные **MW** без сообщений об ошибках или некачественной связи. По вопросам настройки и программирования устройств в сети **SIMATIC NET**, а также связи OPC-сервера с ними обращайтесь, пожалуйста, в службу технической поддержки Siemens A&D или к разработчику Вашего оборудования.

Успешное использование драйвера возможно только после проверки связи с удаленными устройствами через OPC-сервер и правильного указания точки входа (Access Point).

Протокол **S7 communications** предназначен для обмена данными с контроллерами Siemens S7 и любым другим оборудованием, поддерживающим данный протокол. Обмен может происходить как по сети Profibus, так и по локальной сети Ethernet. Драйвер использует универсальный SA-PI, поэтому не требуется вносить в него никаких изменений при изменении носителя. Более подробную информацию по сети SIMATIC NET

можно получить на сайте или у поставщика оборудования. Драйвер имеет доступ на чтение/запись к переменным **MW**, которые могут использоваться программой ПЛК для произвольных целей. Размерность переменных – 16 бит.

Всю необходимую информацию по конфигурации сети драйвер получает с помощью средств **SIMATIC NET**. При проектировании сети пользователь создает так называемые **S7-соединения** для каждого узла сети (например, **S7 connection_1**). Драйвер в процессе работы обращается к этим соединениям по их индексам. Если в проекте есть только одно соединение, то его индекс равен 0. Если число соединений больше 1, то распределение индексов зависит от **SIMATIC NET Software**. В этом случае для определения соответствия имен соединений и их индексов необходимо использовать утилиту **S7Util.exe**:



После нажатия на кнопку **GET S7 connections** утилита представит все имеющиеся S7-соединения для точки входа **CP_L2_1** и их индексы. Если точка входа не сконфигурирована или невозможно получить список, то будут выданы соответствующие сообщения. Если ни одного соединения не найдено, то в списке появится строка **–None–**.

Драйвер, утилита **S7util** и OPC-сервер используют одни и те же ресурсы для связи. Поэтому в каждый момент времени данными может обмениваться только одно приложение. Остальные приложения выдадут ошибку при подключении.

Каналы должны быть класса HEX16. Тип канала (**I** или **O**) определяет его назначение – чтение или запись значения переменной **MW**.

Для обращения к данным необходимо указать индекс соединения и номер переменной **MW**. При наличии только одного соединения его индекс равен 0, в остальных случаях используйте утилиту **S7util.exe**. Возможные значения номера переменной **MW** зависят от параметров используемого ПЛК.

Указание несуществующего соединения или номера переменной **MW** может привести к непредсказуемым ошибкам в системе.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **O**. Связь с удаленными объектами осуществляется по сети Profibus или по локальной сети Ethernet в зависимости от конфигурации рабо-

чей станции. При сбоях в обмене данными каналу выставляется признак недостоверности. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление MPB. Необходимо обратить внимание, что при безуспешной попытке записи значения канала Trace Mode будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

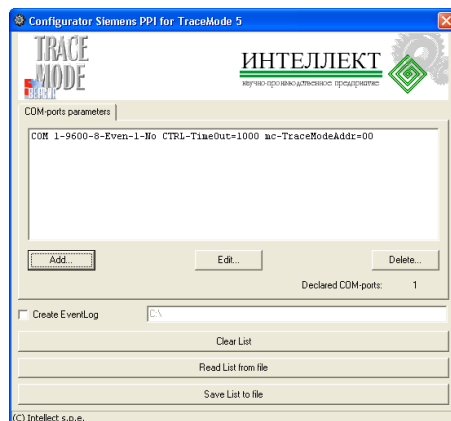
Протокол Siemens PPI

Этот драйвер обеспечивает поддержку протокола **PPI** (Point-to-Point Interface) для обмена с ПЛК Siemens S7-200. Драйвер поддерживает чтение/запись данных во все области памяти контроллера – Stage, System Memory, Analog Input, Analog Output, Counter, Timer, High Speed Counter, Discrete Input, Discrete Output, Marker, Variable Memory. Для обмена данными используется последовательный порт компьютера и конвертер RS-232/RS-485. Драйвер оформлен в виде драйвера t12. Для корректной работы драйвера необходимо правильно произвести конфигурацию порта PPI в ПЛК и последовательного порта компьютера. Протокол PPI поддерживается в ПЛК серий S7-200. О поддержке протокола в других сериях ПЛК нужно получить консультации поставщика ПЛК.

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Конфигурирование драйвера представляет собой указание параметров связи по последовательным портам. Для настройки драйвера используется утилита **PPIconfig.exe**. Каждой порт, указанный в списке, будет автоматически открыт драйвером.

Порт, к которому подключены устройства сети PPI, не должен быть настроен в самой системе Trace Mode.

Страница конфигурации портов имеет следующий вид:

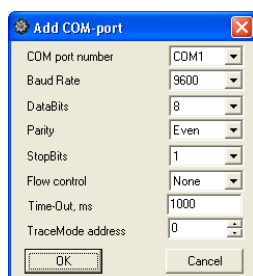


В списке портов каждая строка состоит из восьми параметров:

- номер СОМ-порта. Повторное объявление одного и того же порта приведет к сообщению об ошибке при попытке сохранить конфигурацию;
- скорость передачи данных (Baud Rate), от 300 bps до 115200 bps. Для устройств сети PPI по умолчанию принимается 9600 bps;
- число битов данных (Data Bits). По умолчанию установлено 8 бит;
- контроль четности передачи (Parity), может принимать значения **None**, **Odd** или **Even**. По умолчанию для устройств сети PPI принимается **Even**;
- количество стоп-битов (Stop Bits), 1 или 2. По умолчанию установлен 1 стоп-бит;
- время тайм-аута для данного последовательного порта (в мс) По умолчанию – 1000 мс;
- управление потоком. Используемый конвертер может требовать управления потоком. Для его корректной работы необходимо правильно указать сигналы (**RTS**, **DTR**), которые будут поданы перед каждой посылкой и сняты после ее отправки;
- адрес Trace Mode в сети PPI. Согласно принципам обмена данными в сети PPI, каждое устройство должно иметь уникальный адрес.

Заданные параметры последовательного порта должны совпадать с соответствующими параметрами всех остальных устройств в данном сегменте сети PPI. В противном случае драйвер не сможет вести обмен данными или полученные данные не будут соответствовать действительности и могут повлечь непредсказуемые сбои в системе.

Для создания новой записи нажмите кнопку **Добавить**, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



Для сохранения конфигурации нажмите кнопку **Сохранить в файл**. Если в конфигурации отсутствуют явные ошибки, то будет выдано сообщение об успешном сохранении файла конфигурации (**PPI.cfg**). Ответственность за создание правильной конфигурации полностью возлагается на пользователя.

Файл конфигурации **PPI.cfg** должен быть расположен в папке узла. Не пытайтесь редактировать его вручную и не удаляйте его, т.к. это может привести к непредсказуемым сбоям в системе.

Для открытия существующей конфигурации воспользуйтесь кнопкой **Прочитать из файла**. Для удаления текущей конфигурации (без сохранения в файл) воспользуйтесь кнопкой **Очистить**.

Опция **Вести журнал событий** предоставляет возможность удобной отладки работы системы. По указанному пути будут созданы 2 файла – **PPImedia.log** и **PPIproto.log**, – в которых будет сохраняться соответственно протокол работы драйвера и сообщения о сбоях и их возможных причинах. Указанный каталог должен существовать еще до запуска Trace Mode. После успешной настройки системы эту опцию можно отключить, снизив затраты времени и дискового пространства.

Драйвер реализует обмен данными в сети Siemens PPI в режиме ведущего (Master) узла. Наличие нескольких ведущих узлов не поддерживается. Все остальные ПЛК должны быть ведомыми узлами. Архитектура сети – шина. Каждый узел в сети имеет свой уникальный адрес от 0 до 255. Режим работы сети – Polling Network. Ведущий узел отправляет запрос к ведомому, получает подтверждение о получении и через определенный промежуток времени запрашивает данные. Протокол PPI основан на электрическом интерфейсе RS-485. Поэтому подключение к стандартному последовательному порту ПК (RS-232) возможно только через конвертер RS-485/RS-232 согласно документации по коммуникационным портам ПЛК. За более подробной информацией по принципам функционирования сети Siemens PPI обращайтесь к соответствующим руководствам пользователя, в службу технической поддержки Siemens A&D или к поставщику оборудования.

Каналы могут иметь класс HEX16 или FLOAT. Возможен вариант, при котором будут прочитаны данные типа FLOAT в канал HEX, при этом дробная часть будет отброшена. Тип данных, которые читаются/записываются в контроллер, определяется атрибутом **Область**. Тип канала (**1** или **0**) определяет его назначение – чтение данных или запись.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта. Значение 0 соответствует COM1, 1 – COM2 и т.д. Все используемые порты должны быть описаны на странице настроек портов в **PPIconfig**, и настройки сохранены в файл конфигурации. Номер порта задается в DEC-формате;
- **Адрес** – адрес удаленного ПЛК в сети PPI. Уникальный параметр, определяющий устройство в сети. Диапазон значений – 0 .. 255. По

умолчанию контроллер имеет адрес 2. Изменить эту настройку можно с помощью пакета программирования **STEP7**. Адрес ПЛК задается в DEC-формате;

- **Смещение** – смещение внутри области памяти контроллера. Смещение задается в байтах. Для получения правильных данных необходимо четко представлять принципы адресации в памяти контроллера и способы хранения данных.

Все данные, состоящие из нескольких байт, хранятся в памяти контроллера в формате Big Endian. Т.е. самый старший байт размещается первым, а самый младший – последним:

FLOAT					
1s	ЭКСПОНЕНТА			МАНТИССА	
31	30		23	22	0
WORD					
СТАРШИЙ БАЙТ			МЛАДШИЙ БАЙТ		
15		8	7		0

Любая область памяти, кроме Timer, Counter и High Speed Counter, может рассматриваться как набор слов или чисел с плавающей точкой. Это определяется параметром **Область** данного канала. В соответствии с этой настройкой, драйвер запрашивает 2 или 4 байта из указанной области, преобразует их в формат Trace Mode и передает в систему в виде HEX или FLOAT. При этом 4 байта, прочитанные из памяти контроллера как FLOAT, могут рассматриваться далее в системе как число с плавающей точкой (FLOAT) или как целое (HEX). Во втором случае дробная часть будет отброшена. Аналогично, прочитанные как WORD 2 байта могут далее рассматриваться в Trace Mode как целое (HEX) или как число с плавающей точкой (FLOAT), однако дробная часть его всегда будет равна 0. Во избежание ошибок следует задавать класс HEX для данных типа WORD и класс FLOAT для данных типа FLOAT.

Настройка **Смещение** указывает порядковый номер байта внутри указанной области, который является первым в представлении перемешанной. При этом могут иметь место пересечения данных.

		WORD #11	WORD #13	WORD #15	WORD #17				
	WORD #10	WORD #12	WORD #14	WORD #16					
OFFSET	10	11	12	13	14	15	16	17	18
	FLOAT #10			FLOAT #14					
	FLOAT #11			FLOAT #15					
	FLOAT #12								
		FLOAT #13							

Как видно из рисунка, изменение значения слова со смещением 10 может повлечь за собой непредвиденное изменение слова со смещением 11. Аналогично и для данных с плавающей точкой, но в

этом случае затрагиваются 4 последующих значения. Во избежание ошибок, рекомендуется обращаться к целочисленным данным с четным смещением, а к данным с плавающей точкой – со смещением, кратным 4.

Области Timer, Counter и High Speed Counter являются структурированными. Смещение в этом случае определяет порядковый номер элемента. Элементы не пересекаются. Кроме значения, элемент содержит бит статуса. Для доступа к этому биту используются настройки Timer(Status), Counter(Status) и High Speed Counter(Status). Для High Speed Counters он всегда равен 0.

Смещение задается в HEX-формате;

- **Область** – область памяти контроллера. Драйвер позволяет получить доступ ко всем существующим областям памяти контроллера: Stage, System Memory, Analog Input, Analog Output, Counter, Timer, High Speed Counter, Discrete Input, Discrete Output, Marker, Variable Memory. Кроме того, любая область памяти, кроме Timer, Counter и High Speed Counter, может рассматриваться как массив целых 2-байтных чисел (WORD) или 4-х байтных чисел с плавающей точкой (FLOAT). Это определяется выбранным параметром **Область** и не зависит от класса канала (HEX или FLOAT). Данные в областях Timer, Counter хранятся только в целочисленном формате. Данные в области High Speed Counter хранятся как 4-байтные двойные слова, поэтому для получения максимального количества значащих цифр необходимо использовать канал FLOAT. Кроме того, перечисленные области имеют бит статуса, но в области High Speed Counter он не используется, поэтому всегда равен 0. Некоторые области доступны только для чтения, другие – только для записи.

AREA	WORD		FLOAT		STATUS	
	READ	WRITE	READ	WRITE	READ	WRITE
Stage	+	+	+	+	X	X
System Memory	+	+ *	+	+ *	X	X
Analog Input	+	-	+	-	X	X
Analog Output	-	+	- **	- **	X	X
Counter	+	+	- **	- **	+	-
Timer	+	+	- **	- **	+	-
High Speed Counter	+	-	- **	- **	+ ***	-
Discrete Input	+	+	+	+	X	X
Discrete Output	+	+	+	+	X	X
Markers	+	+	+	+	X	X
Variable Memory	+	+	+	+	X	X

* - SM0 .. SM29 доступны только для чтения

** - эти области рассматриваются только как целые

*** - бит статуса HighSpeedCounter всегда равен 0

Обращение к неверно настроенному ПЛК, адресация к несуществующим переменным или константам может приве-

сти к непредсказуемым сбоям в системе.

В режиме STOP области данных контроллера доступны только для чтения. Для получения возможности записи необходимо перевести ПЛК в режим RUN с помощью переключателя на корпусе или из **STEP7** (при этом переключатель должен быть в положении **TERM**).

Любой последовательный порт в каждый момент времени может использоваться только одним приложением. Таким образом, Trace Mode не сможет обмениваться данными с ПЛК при запуске **STEP7**.

Чтение данных происходит по запросу МРВ в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **О**. При каждой отправке ответа ожидается в течение заданного таймаута. Если ответа за это время нет, то каналу выставляется признак недостоверности. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление МРВ. Необходимо обратить внимание на то, что при неуспешной попытке записи значения канала Trace Mode будет пытаться повторить запись до тех пор, пока она не пройдет успешно.

В случае возникновения ошибок рекомендуется включить в конфигураторе опцию **Вести журнал событий** и проанализировать протокол обмена.

Обмен с контроллерами фирмы ЭМИКОН

Обмен через плату C05, шина ISA

Для обмена в среде WINDOWS 2000/XP с контроллерами фирмы **ЭМИКОН** (ЭК-2000, CPU-11A) через плату **C05** по протоколу **MODBUS** необходимо установить драйвер **EmicC05**.

Одновременная работа плат **C05** и **C06** невозможна. Если ранее на компьютере использовалась плата **C06**, ее необходимо удалить и деинсталлировать ее драйвер.

После инсталляции работу драйвера можно проверить, запустив программу **c05test.exe**.

Шаблоны каналов обмена конфигурируются в редакторе (см. **Обмен по MODBUS**). Настройка **Тип протокола** должны быть **Эмикон**.

МРВ может обмениваться с контроллерами по каналу **А** или по каналу **Б**. Для обмена по каналу **А** настройка **Номер порта** канала должна быть равна 0, а для канала **Б** – 1.

Для настройки параметров обмена по каналам **А** и **Б** платы **C05** используются конфигурационные файлы **MODE.A** и **MODE.B** соответственно (см. **Файлы mode.a** и **mode.b**). Они имеют текстовый формат и редактируются вручную. Эти файлы с требуемыми настройками следует разместить в директории проекта.

Обмен через плату C06, шина PCI

Для обмена в среде WINDOWS 2000 с контроллерами фирмы **ЭМИКОН** (ЭК-2000, CPU-11A) через плату **C06** по протоколу **MODBUS** необходимо установить драйвер **EmicC06**.

Одновременная работа плат **C05** и **C06** невозможна. Если ранее на компьютере использовалась плата **C05**, ее необходимо удалить. Кроме того, нужно деинсталлировать драйвер платы **C05** при помощи программы **C05Setup.exe**. Эта программа поставлялась вместе с драйвером.

После инсталляции работу драйвера можно проверить, запустив программу **c06test.exe**.

Шаблоны каналов обмена конфигурируются в редакторе (см. **Обмен по MODBUS**). Настройка **Тип протокола** должны быть **Эмикон**.

МРВ может обмениваться с контроллерами по каналу **А** или по каналу **Б**.

Для обмена по каналу **А** настройка **Номер порта** канала должна быть равна 0, а для канала **Б** – 1.

Для настройки параметров обмена по каналам **А** и **Б** платы **С06** используются конфигурационные файлы **MODE.A** и **MODE.B** соответственно (см. **Файлы mode.a и mode.b**). Они имеют текстовый формат и редактируются вручную. Эти файлы с требуемыми настройками следует разместить в директории проекта.

Обмен через плату C06PnP, шина PCI

Для обмена в среде WINDOWS XP с контроллерами фирмы **ЭМИКОН** (ЭК-2000, CPU-11A) через плату **С06PnP** по протоколу **MODBUS** необходимо установить драйвер **EmicC06**, используя мастер установки оборудования или файл **EmicC06.inf**.

Одновременная работа плат **С05** и **С06** невозможна. Если ранее на компьютере использовалась плата **С05**, ее необходимо удалить. Кроме того, нужно деинсталлировать драйвер платы **С05** при помощи программы **С05Setup.exe**. Эта программа поставлялась вместе с драйвером.

Шаблоны каналов обмена конфигурируются в редакторе (см. **Обмен по MODBUS**). Настройка **Тип протокола** должны быть **Эмикон**.

MPB может обмениваться с контроллерами по каналу **А** или по каналу **Б**. Для обмена по каналу **А** настройка **Номер порта** канала должна быть равна 0, а для канала **Б** – 1.

Для настройки параметров обмена по каналам **А** и **Б** платы **С06PnP** используются конфигурационные файлы **MODE.A** и **MODE.B** соответственно (см. **Файлы mode.a и mode.b**). Они имеют текстовый формат и редактируются вручную. Эти файлы с требуемыми настройками следует разместить в директории проекта.

Файлы mode.a и mode.b

Файлы **mode.a** и **mode.b** содержат параметры настройки соответственно каналов **А** и **Б** платы **С05/С06**. Оба файла после установки имеют следующий вид:

```
Mode: 2
SubMode: 0
Coding: 0
StopBitNumber: 0
Parity: 0
DataBitNumber: 0
```

```
BaudRate: 14  
RepetitionNumber: 3  
TimeOut: 10
```

где

- **BaudRate** – скорость обмена данными. Значение этого параметра задает следующие скорости обмена:

- 0 – зарезервировано;
- 1 – 300 бод;
- 2 – 600 бод;
- 3 – 1200 бод;
- 4 – 2400 бод;
- 5 – 4800 бод;
- 6 – 9600 бод;
- 7 – 19200 бод;
- 8 – 38400 бод;
- 9 – 48000 бод;
- 10 – 57600 бод;
- 11 – 115200 бод;
- 12 – 144000 бод;
- 13 – 192000 бод;
- 14 – 288000 бод;
- 15 – 576000 бод;

- **RepetitionNumber** – число попыток передачи сообщения модулем C05/C06 контроллеру (1...255).
- **TimeOut** – число 10-миллисекундных интервалов (1...255), в течение которых модуль C05/C06 дожидается ответа от контроллера; Рекомендуется задавать значение **TimeOut** не меньшее, чем рассчитанное по эмпирической формуле:

$$\text{TimeOut} = [38400/R] + 2 * [N/16] + 2$$

где

- **R** = 300...576000 (бод) - скорость обмена данными;
- **N** - число считываемых (записываемых) переменных;
- **[]** - выделение целой части;
- **Mode** – режим:
 - 0 – протокол ModBus, асинхронный режим;
 - 2 – протокол SDLC;
- **SubMode** – подрежим:
 - 0 – работа в режиме "ведущего";
 - 1...255 – адрес "ведомого";

- **Coding** – кодирование. Для асинхронного режима задается 0, в режиме SDLC не используется).

Следующие 3 параметра имеют разный смысл для режимов **MODBUS** и **SDLC**:

Параметр	MODBUS	SDLC
StopBitNum	количество стоп-битов (1 или 2)	режим CRC (0 или 1);
Parity	контроль четности: 0 - без проверки 1 - проверка нечетности 2 - проверка четности	интервал между RR (0...255)
DataBitNum	количество битов данных (5...8)	количество RR(0...6)

За информацией о режиме связи следует обращаться к документации по конкретному контроллеру.

Обмен с контроллерами Allen Bradley

Обмен по протоколу Ethernet/IP Rockwell Automation

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Номер устройства** — идентификатор контроллера в таблице, однобайтное число в десятичном формате;
- **Тип файла** — тип файла в контроллере, значение выбирается из меню:
 - **(S)Status** – статус;
 - **(B)Binary** – бинарный файл;
 - **(T)Timer** – таймеры;
 - **(C)Counter** – счетчики;
 - **(R)Control** – управление;
 - **(N)Integer** – целые числа;
 - **(F)Floating** – числа с плавающей точкой;
 - **(O)Output** – выход;
 - **(I)Input** – вход;
- **Номер файла** — номер файла, однобайтное число в десятичном формате;
- **Номер элемента** — номер элемента в файле, однобайтное число в десятичном формате;
- **Имя поля** — тип элемента файла, значение выбирается из меню:
 - **WORD** – 2-байтное целое;
 - **REAL** – число с плавающей точкой;
 - **ACC** – Accumulator;
 - **PRE** – Preset;
 - **LEN** – Length, длина битового массива или файла;
 - **POS** – Position, битовый указатель или позиция;
 - **Control** – 2-байтный массив флагов.

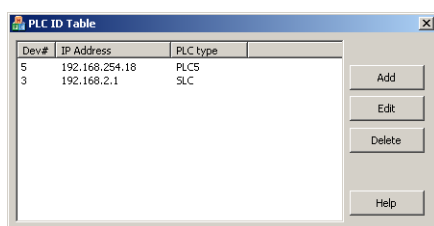
При создании каналов для чтения и записи одного и того же элемента файла все каналы OUTPUT должны иметь идентификаторы в Trace Mode больше, чем максимальный идентификатор у каналов INPUT этого элемента.

- **IP-адрес** – IP-адрес устройства.

При экспорте в папке узла создается двоичный файл **RA_PLC_table.dat**

с таблицей контроллеров. Этот файл необходим для корректной работы драйвера.

Для корректировки файла **RA_PLC_table.dat** можно также использовать редактор **RAPLCEditor.exe**, который в табличном виде представляет список доступных из драйвера контроллеров, для каждого прибора задается уникальный номер (**Dev#**), IP адрес (**IP Address**) и тип контроллера (**PLC type**). В редакторе доступны три функции – добавление строки в таблицу, изменение параметров прибора, и удаление строки из таблицы. Таблица сохраняется в двоичный файл **RA_PLC_table.dat** автоматически при выходе из программы. Кнопка **Help** вызывает раздел справочной системы с описанием драйвера. Вид редактора таблицы контроллеров представлен на рисунке.



Некоторые файлы (**Timer, Counter, Control**) содержат в качестве элементов составные типы данных, тогда в настройке канала указывается соответствующее поле структуры.

Поле структуры **Control** для элементов файлов **Timer, Counter** и **Control** является битовым массивом флагов, назначение отдельных битов описано в руководстве пользователя соответствующего контроллера.

Пример списка файлов контроллера с типами элементов и возможностями доступа приведен в таблице.

Тип файла	Тип элемента файла	Номер файла	Поля элемента	Доступ
Output	WORD	0	-	Input only
Input	WORD	1	-	Input only
Status	WORD	2	-	Input/Output
Binary	WORD	3	-	Input/Output
Timer	структура	4	Control, PRE, ACC	Input/Output
Counter	структура	5	Control, PRE, ACC	Input/Output
Control	структура	6	Control, LEN, POS	Input/Output
Integer	WORD	7	-	Input/Output
Floating	REAL	8	-	Input/Output

За более подробным описанием структуры памяти контроллеров обра-

щайтесь к инструкции по установке, программированию и наладке сетей Allen Bradley DF1 – «DF1 Protocol and Command Set, Reference Manual 1770-6.5/16» и руководствам пользователя соответствующего контроллера.

Обмен по протоколу DF1

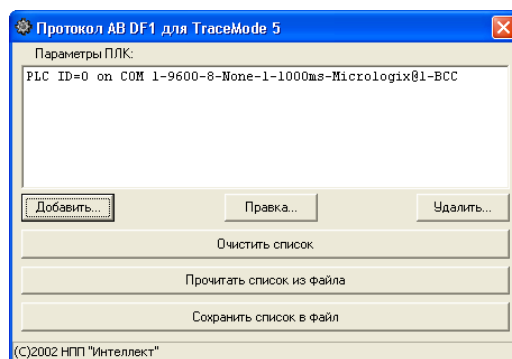
Драйвер обеспечивает обмен данными, используя транспортный протокол **DF1** для обмена с ПЛК серий Micrologix, SLC и PLC фирмы Allen-Bradley. Драйвер поддерживает чтение/запись файлов Output, Input, Status, Binary, Timer (все поля), Counter (все поля), Control (все поля), Integer. Для обмена данными используется последовательный порт компьютера. Драйвер оформлен в виде драйвера t12. Для корректной работы драйвера необходимо правильно произвести конфигурацию порта в ПЛК (см. приложение ниже). Протокол DF1 поддерживается в ПЛК серий Micrologix, SLC 500 Fixed I/O, SLC 5/01, SLC 5/02, SLC 5/03, SLC 5/04, PLC5. О поддержке протокола в других сериях ПЛК проконсультируйтесь с поставщиком ПЛК.

Драйвер обеспечивает обмен данными как в режиме чтения, так и в режиме записи. Для настройки портов, по которым Trace Mode будет обмениваться данными, необходимо занести их в список портов (файл **DF1.cfg**). Для выполнения этого пункта необходимо воспользоваться утилитой **DF1config.exe** (см. ниже). Каждый порт в списке будет автоматически открыт драйвером.

Порт, к которому подключены устройства AB DF1, не должен быть настроен в самой системе Trace Mode.

Утилита **DF1Configurator** предназначена для создания списка портов, по которым может осуществляться обмен данными. Сам список сохраняется в файле **DF1.cfg**, который должен быть перенесен в папку узла.

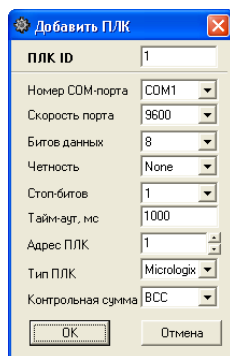
Основное окно программы имеет следующий вид:



В списке портов каждая строка состоит из десяти параметров:

- идентификатор ПЛК. Этот параметр должен быть уникальным, совпадения могут привести к непредсказуемым последствиям. Этот параметр необходимо будет указать в поле **Номер устройства** при создании каналов;
- номер СОМ-порта. При указании последующих параметров последовательных портов необходимо обратить внимание на их идентичность для одного и того же последовательного порта ПК. В противном случае порт будет открыт с последними указанными параметрами;
- скорость передачи данных (Baud Rate), от 300 bps до 115200 bps. По умолчанию принимается 9600 bps;
- число битов данных (Data Bits). По умолчанию установлено 8 бит;
- контроль четность передачи (Parity), может принимать значения None, Odd или Even. По умолчанию установлено None;
- количество стоп-битов (Stop Bits), 1 или 2. По умолчанию установлено 1 бит;
- время таймаута для данного ПЛК (в мс);
- адрес ПЛК в сети DF1;
- тип ПЛК. Различные серии ПЛК используют для доступа к данным различные команды;
- тип проверки целостности пакета. Проверка CRC является более надежной и включает 2 байта. Проверка ВСС включает 1 байт и не определяет выпадение или появление нулевых байтов.

Для создания новой записи нажмите кнопку **Добавить**, кнопка **Удалить** удалит запись, кнопка **Правка** или двойной щелчок по элементу списка вызовет окно редактирования параметров записи:



При нарушении уникальности ID ПЛК сохранение такой конфигурации будет невозможно. Также нельзя будет сохранить конфигурацию без объявления хотя бы одного ПЛК.

Каналы должны быть класса HEX16. Тип канала (**I** или **O**) определяет его назначение – чтение данных или запись.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Номер устройства** – уникальный идентификатор ПЛК, с которым будет проводиться обмен данными для этого канала. Настройки порта должны быть ранее внесены в программе **DF1Configurator**. Если в списке утилиты **DF1Configurator** нет строки с соответствующим ID, то канал будет отключен при запуске системы;
- **Тип файла** – имя файла в ПЛК, с которым данный канал будет обмениваться данными. Для обеспечения совместимости со всеми сериями ПЛК драйвер поддерживает обмен данными со следующими файлами: Output, Input, Status, Binary, Timer, Counter, Control, Integer. Тип доступа к файлам (R/W или R) определяется серией используемого ПЛК (см. приложение);
- **Имя поля** – имя поля в выбранном файле и слове в ПЛК. Если указанный файл не имеет полей, то необходимо указать фиктивное поле WORD. Необходимо учесть, что для каждой серии ПЛК те или иные поля/файлы могут быть доступны только для чтения. Каналы с неверной конфигурацией получают признак аппаратной недостоверности и не опрашиваются. Допустимые имена файлов и соответствующие поля, а также уровень доступа указаны в приложении;
- **Номер файла** – номер файла. Допустимые значения лежат в пределах 0..255 и определяются серией ПЛК (см. приложение);

Неверная ссылка на номер файла может привести к непредсказуемым ошибкам в системе.

- **Номер слова** – номер слова. Допустимые значения лежат в пределах 0..255 и определяются серией ПЛК (см. приложение).

При обращении к элементам файлов необходимо убедиться, что эти элементы присутствуют в текущей конфигурации контроллера. Так, при изначальной конфигурации в файлах N, B и т.п. объявлено только по одному элементу (B3:0, N7:0)

Чтение данных происходит по запросу MPB в соответствии с фазой и периодом работы канала. Запись – при изменении выходного значения канала типа **О**. При каждой посылке ответ ожидается в течение 1 секунды. Если ответа за это время нет, то каналу выставляется признак недостоверности. Вне зависимости от того, была ли попытка успешной или нет, драйвер возвращает управление MPB. Необходимо обратить внимание, что при безуспешной попытке записи значения канала Trace Mode будет пытаться повторить запись до тех пор, пока она не пройдет успешно. Обмен

данными между подчиненными узлами драйвером не поддерживается.

Посылка значений в файлы, доступные только для чтения, приведет к установке канала признака недостоверности.

Приложение. Данные, доступные для драйвера

Сведения, изложенные в этом разделе, являются частью инструкции по установке, программированию и наладке сетей AB DF1 – **DF1 Protocol and Command Set, Reference Manual 1770-6.5.16**. Все изложенные данные не могут претендовать на полноту и точность изложения. Пожалуйста, обращайтесь за более подробной информацией к службе техподдержки фирмы Allen-Bradley и соответствующим руководствам пользователя.

MICROLOGIX

Тип файла	Тип данных	File	MAX word	Доступ	Примечания
Output	word	0	0	read	
Input	word	1	1	read	
Binary	word	3	31	read/write	
Status	word	2	32	read/write	
Timer	word	4	39	read/write	поля: ACC, PRE
Timer	Boolean	4	39	read	поля: DN, TT, EN
Counter	word	5	31	read/write	поля: ACC, PRE
Counter	Boolean	5	31	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6	31	read/write	поля: LEN, POS
Control	Boolean	6	31	read	поля: FD, IN, UL, ER, EM, DN, EU, EN
Integer	word	7	104	read/write	

SLC 500 Fixed I/O

Тип файла	Тип данных	File	MAX word	Доступ	Примечания
Output	word	0	1	read	
Input	word	1	1	read	
Binary	word	3, 9-255	255	read/write	
Status	word	2	96	read/write	

Тип файла	Тип данных	File	MAX word	Доступ	Примечания
Timer	word	4, 9-255	255	read/write	поля: ACC, PRE
Timer	Boolean	4, 9-255	255	read	поля: DN, TT, EN
Counter	word	5, 9-255	255	read/write	поля: ACC, PRE
Counter	Boolean	5, 9-255	255	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6, 9-255	255	read/write	поля: LEN, POS
Control	Boolean	6, 9-255	255	read	поля: FD, IN, UL, ER, EM, DN, EU, EN
Integer	word	7, 9-255	255	read/write	

SLC 5/01

Тип файла	Тип данных	File	Диапазон Slot	MAX word	Доступ	Примечания
Output	word	0	1-30	xxx	read	
Input	word	1	1-30	xxx	read	
Binary	word	3, 9-255	-	255	read/write	
Status	word	2	-	96	read/write	
Timer	word	4, 9-255	-	255	read/write	поля: ACC, PRE
* Timer	Boolean	4, 9-255	-	255	read	поля: DN, TT, EN
Counter	word	5, 9-255	-	255	read/write	поля: ACC, PRE
* Counter	Boolean	5, 9-255	-	255	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6, 9-255	-	255	read/write	поля: LEN, POS
* Control	Boolean	6, 9-255	-	255	read	поля: FD, IN, UL, ER, EM, DN, EU, EN
Integer	word	7, 9-255	-	255	read/write	

SLC 5/02

Тип файла	Тип данных	File	Диапазон Slot	MAX word	Доступ	Примечания
Output	word	0	1-30	xxx	read	

Тип файла	Тип данных	File	Диапазон Slot	MAX word	Доступ	Примечания
Input	word	1	1-30	xxx	read	
Binary	word	3,9-255	-	255	read/write	
Status	word	2	-	96	read/write	
Timer	word	4,9-255	-	255	read/write	поля: ACC,PRE
* Timer	Boolean	4,9-255	-	255	read	поля: DN, TT, EN
Counter	word	5,9-255	-	255	read/write	поля: ACC, PRE
* Counter	Boolean	5,9-255	-	255	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6,9-255	-	255	read/write	поля: LEN, POS
* Control	Boolean	6,9-255	-	255	read	поля: FD, IN,UL, ER, EM, DN, EU, EN
Integer	word	7,9-255	-	255	read/write	

SLC 5/03

Тип файла	Тип данных	File	Диапазон Slot	MAX word	Доступ	Примечания
Output	word	0	1-30	xxx	read	
Input	word	1	1-30	xxx	read	
Binary	word	3,9-255	-	255	read/write	
Status	word	2	-	96	read/write	
Timer	word	4,9-255	-	255	read/write	поля: ACC,PRE
* Timer	Boolean	4,9-255	-	255	read	поля: DN, TT, EN
Counter	word	5,9-255	-	255	read/write	поля: ACC, PRE
* Counter	Boolean	5,9-255	-	255	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6,9-255	-	255	read/write	поля: LEN, POS
* Control	Boolean	6,9-255	-	255	read	поля: FD, IN,UL, ER, EM, DN, EU, EN
Integer	word	7,9-255	-	255	read/write	

SLC 5/04

Тип файла	Тип данных	File	Диапазон Slot	MAX word	Доступ	Примечания
Output	word	0	1-30	xxx	read	
Input	word	1	1-30	xxx	read	
Binary	word	3,9-255	-	255	read/write	
Status	word	2	-	96	read/write	
Timer	word	4,9-255	-	255	read/write	поля: ACC,PRE
* Timer	Boolean	4,9-255	-	255	read	поля: DN, TT, EN
Counter	word	5,9-255	-	255	read/write	поля: ACC, PRE
* Counter	Boolean	5,9-255	-	255	read	поля: UA, UN, OV, DN, CD, CU
Control	word	6,9-255	-	255	read/write	поля: LEN, POS
* Control	Boolean	6,9-255	-	255	read	поля: FD, IN,UL, ER, EM, DN, EU, EN
Integer	word	7,9-255	-	255	read/write	

PLC 5

Тип файла	Тип данных	File	MAX word	Доступ	Примечания
Output	word	0	277octal	read/write	
Input	word	1	277octal	read/write	
Binary	word	3-999	999	read/write	
Status	word	2	127	read/write	
Timer	word	3-999	999	read/write	поля: ACC,PRE
* Timer	Boolean	3-999	999	read	поля: DN, TT, EN
Counter	word	3-999	999	read/write	поля: ACC, PRE
* Counter	Boolean	3-999	999	read	поля: UA, UN, OV, DN, CD, CU
Control	word	3-999	999	read/write	поля: LEN, POS
* Control	Boolean	3-999	999	read	поля: FD, IN,UL, ER, EM, DN, EU, EN
Integer	word	7-	999	read/write	

Тип файла	Тип данных	File	MAX word	Доступ	Примечания
		999			

Примечания:

1. Для серии SLC 5/01, 5/02, 5/03, 5/04 значение MAX word для файлов Output и Input зависит от числа подключенных слотов. Номер адресуемого слота не указывается, но номер K слова в слоте N при числе слов M в каждом слоте определяется как $K+(N-1)*M$
2. Вследствие объективных ограничений возможна адресация к номерам файлов и номерам слов, не превышающим 255.

Обмен с контроллерами OMRON

Обмен по протоколу HOST LINK

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Модуль** – номер устройства в формате HEX (0-FF);
- **Канал** – десятичный адрес переменной (диапазон соответствует протоколу обмена);
- **Тип** – команда:

Тип	Описание
IR	команда чтения/записи регистров области "Input registers" (или CIO)
HR	команда чтения/записи регистров области "Hold registers"
AR	команда опроса области "Auxiliary memory"
LR	команда чтения/записи регистров области "Link registers"
TC	команда чтения статусов регистров "Timer/Counters Status"
DM	команда чтения/записи регистров области "Data Memory"
PV	команда чтения/записи текущих значений регистров "Timer/Counters PV"

При работе в режиме "RUN" контроллером обеспечивается только чтение регистров. Для обеспечения возможности записи регистров контроллер должен работать в режиме "Monitor mode".

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров COM-порта**).

Обмен с контроллерами OMRON по сети Ethernet по протоколу UDP

Для обмена с контроллерами OMRON по UDP используется встроенный драйвер (см. **Обмен по встроенным протоколам по сети**).

Типы областей памяти

№	Название области	Название в списке	Код области (Memory area code) (HEX)	Начальный адрес (HEX, не для пользователей)	Замечания
0	CIO	CIO	80	0000	2 байта
1	CIO(forced status)	CIO_frc	85	0000	2 байта – каждый бит определяет статус (forced/not forced) соответствующего бита из области CIO бит = 0 – не форсирован (not forced) бит = 1 – форсирован (forced)
2	TR	TR	80	09FF	младшие 8 бит – состояние регистров TR0-TR7, offset всегда должен быть 0
3	G	G	80	0A00	2 байта
4	G(forced status)	G_frc	85	0A00	2 байта – каждый бит определяет статус (forced/not forced) соответствующего бита из области G бит = 0 – не форсирован (not forced) бит = 1 – форсирован (forced)
5	A	A	80	0B00	2 байта
6	TIM(completion flag status)	TIM_cmpl	01	0000	1 бит значение канала = 0 – бит сброшен значение канала = 1 – бит установлен
7	TIM(PV)	TIM_PV	81	0000	2 байта – значение таймера
8	CNT(completion flag status)	CNT_cmpl	01	0800	1 бит значение канала = 0 – бит сброшен значение канала = 1 – бит установлен
9	CNT(PV)	CNT_PV	81	0800	2 байта – значение счетчика
10	DM	DM	82	0000	2 байта – значение слова

№	Название области	Название в списке	Код области (Memory area code) (HEX)	Начальный адрес (HEX, не для пользователей)	Замечания
11	TN(flag status)	TN_fs	03	0000	1 бит значение канала = 0 – флаг сброшен значение канала = 1 – флаг установлен
12	ST(flag status)	ST_fs	04	0000	1 бит значение канала = 0 – флаг сброшен значение канала = 1 – флаг установлен
13	IR	IR	9C	0000	2 байта
14	DR	DR	9C	0003	2 байта
15	LR	LR	80	03E8	2 байта
16	HR	HR	80	0428	2 байта
17	AR	AR	80	048C	2 байта
18	Expansion DM bank 0	EDM0	90	0000	2 байта
19	Expansion DM bank 1	EDM1	91	0000	2 байта
20	Expansion DM bank 2	EDM2	92	0000	2 байта
21	Expansion DM bank 3	EDM3	93	0000	2 байта
22	Expansion DM bank 4	EDM4	94	0000	2 байта
23	Expansion DM bank 5	EDM5	95	0000	2 байта
24	Expansion DM bank 6	EDM6	96	0000	2 байта
25	Expansion DM bank 7	EDM7	97	0000	2 байта
26	Expansion DM current bank	EDM	98	0000	2 байта
27	AC(flag status)	AC	1B	0000	1 бит значение канала = 0 – флаг сброшен значение канала = 1 – флаг установлен

Примечание 1.

В контроллерах **C**-серии области **TIM** и **CNT** – это одна область, а в контроллерах **CV**-серии это две разные области,

поэтому для обращения к области **CNT** контроллера **C**-серии нужно использовать тип области **TIM**.

Примечание 2.

Вне зависимости от области памяти, заданной атрибутом **Тип области**, в атрибут (97, **C4**) канала записывается значение, заданное в строке **IP-адрес** в следующем виде:

AREA=<HEX-код области без префикса '0x'>

Например, для доступа к переменной типа HR в контроллере OMRON CJ1W-PA202 в строке **IP-адрес** следует задать следующее выражение:

AREA=B2

Обмен с контроллерами OMRON по сети Controller Link

Для обмена с контроллерами **OMRON** по сети **Controller Link** необходимо установить драйвер **OMRON CLK** и плату **Controller Link Support Board** вместе с прилагающимся к ней программным обеспечением. Также необходимо соответствующим образом настроить все контроллеры OMRON, подключенные к сети (сетям) и плату **Controller Link Support Board**. Настройка контроллеров включает в себя настройку таблиц ввода/вывода (IO Table), установку номеров узлов (Node Address) и номеров модулей (Unit Number), а также настройку таблиц маршрутизации. За подробными инструкциями по установке и настройке контроллеров и сетей OMRON следует обращаться к документации OMRON.

Обмен с контроллерами происходит путем отправки и приема FINS-сообщений. (FINS – Factory Interface Network Service – протокол обмена сообщениями, поддерживаемый большинством контроллеров и сетей OMRON). Это позволяет обмениваться данными не только с контроллерами, подключенными к сети **Controller Link**, но и с контроллерами в других сетях OMRON (**Ethernet**, **SYSMAC Link**, **SYSMAC Net**). Для осуществления такого обмена необходимо, чтобы контроллер, подключенный к Controller Link, был также подключен и к другой сети OMRON (например, Ethernet). Такой контроллер будет выполнять функции маршрутизатора для FINS-сообщений, давая возможность обмениваться данными с контроллерами в сети Ethernet (SYSMAC Link, SYSMAC Net).

При экспорте в папку узла создается файл **omron_clk_addr_tab**, необходимый для корректной работы драйвера (см. ниже **Файл omron_clk_addr_tab**).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

Атрибут	Длина, байт	Диапазон	Функция
---------	-------------	----------	---------

Атрибут	Длина, байт	Диапазон	Функция
Номер устройства	1	1-255	Индекс в таблице адресов; таблица адресов находится в текстовом файле omron_clk_addr_tab . Файл читается при загрузке драйвера.
Область	1	выбор из списка	Определяет область памяти для чтения/записи регистра.
Смещение	2	0 – 65535	Смещение от начала области памяти
Дополнительно			Адрес модуля OMRON в сети Controller Link (см. ниже Файл omron_clk_addr_tab)

Все числа в таблице – десятичные.

Каналу выставляется признак аппаратной недостоверности в следующих случаях:

- канал настроен на устройство, номер которого отсутствует в файле **omron_clk_addr_tab** на момент загрузки драйвера;
- задан неверный адрес регистра или была попытка записи в регистр только для чтения;
- в течение секунды не пришел ответ от контроллера.

Файл omron_clk_addr_tab

Назначение файла **omron_clk_addr_tab** – установить в соответствие каждому значению настройки **Номер устройства** (которая используется внутри TRACE MODE) реальный адрес модуля OMRON в сети Controller Link.

Файл состоит из двух или более строк вида:

1 1.12.10

Каждая такая строка устанавливает в соответствие одному значению настройки **Номер устройства** адрес модуля контроллера в сети. Для каждого модуля, с которым планируется осуществлять обмен данными, должна быть своя строка. Также обязательно должна присутствовать строка, определяющая адрес самого компьютера в сети OMRON. С точки зрения контроллеров, компьютер с TRACE MODE является контроллером с одним модулем, поэтому он должен иметь свой адрес.

Все строки, начинающиеся с ‘;’ игнорируются. Все строки неправильного формата также игнорируются.

Первое число каждой строки – идентификатор модуля (**Номер устройства** в диапазоне [0, 255]), установленного в контроллер. По этому числу модуль идентифицируется в TRACE MODE. Далее через пробел или символ табуляции (‘\t’) записывается адрес модуля в сети OMRON. Первая часть адреса – адрес сети (0 для локальной сети, т.е. той, в которую вхо-

дит сетевая плата). Вторая часть – адрес узла (контроллера). Третья часть – номер модуля (этот номер должен соответствовать номеру, заданному для данного модуля аппаратно). Диапазоны всех частей адреса определяются типом сети (сетей) (см. документацию OMRON).

В файле обязательно должна присутствовать строка для **Номер устройства = 0**. Эта строка определяет адрес компьютера с Trace Mode в сети Controller Link.

Типы областей памяти описаны в разделе **Обмен с контроллерами OMRON по сети Ethernet по протоколу UDP**. В этом же разделе описана организация протоколирования, которая справедлива и для данного драйвера. Файлы отчетов имеют имена **omron_t12_clk.log** и **media_clk.log**.

Обмен с контроллерами фирмы ЭНЕРГОМЕРА

Обмен со счетчиком ЦЭ6823

Данный драйвер оформлен в виде драйвера t12.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого для обмена последовательного порта (0 - COM1, ..., 31 - COM32). Этот параметр задается в HEX формате;
- **Параметр** – физическая величина, значение которой считывается из счетчика (выбирается из списка):
 - **Exyzt** – энергия в кВт*ч;
 - **GP30t** – мощность, усредненная по 30-минутному интервалу;
- **Доп_параметр** – число **0..255** (задается в формате DEC). Отдельные биты этого числа определяют вариант физической величины, установленной настройкой **Параметр**. Расшифровка значений дополнительного параметра для каждой из физических величин приведена ниже;
- **Скорость0** – число **0..8** (задается в формате DEC). Определяет начальную скорость обмена по последовательному порту (в соответствии со стандартом МЭК 1107-96 скорость должна быть 300 бод, однако счетчик позволяет задать более высокую скорость):
 - 0 – 300 бод
 - 1 – 600 бод
 - 2 – 1200 бод
 - 3 – 2400 бод
 - 4 – 4800 бод
 - 5 – 9600 бод
 - 6 – 19200 бод
 - 7 – 38400 бод
 - 8 – 57600 бод

Для задания рабочей скорости обмена предназначена настройка **Скорость1**;

- **Скорость1** – число **0..8** (задается в формате DEC). Определяет рабочую скорость обмена по последовательному порту: Соответствие значений этого параметра скоростям обмена аналогично при-

веденному выше.

Для параметра **Exyzt** (энергия в кВт*ч) биты дополнительного параметра имеют следующее значение:

- Биты 0 и 1:
 - 00 (0 в формате DEC) – нарастающим итогом;
 - 01 (1) – за сутки;
 - 10 или 11 (2 или 3) – за месяц.
- Биты 2 и 3:
 - 00 (0) – текущий день или месяц;
 - 01 (4) – предыдущий день или месяц;
 - 10 или 11 (8 или 12) – два дня или месяца назад.
- Бит 4:
 - 0 (0) – потребленная энергия (I);
 - 1 (16) – отпущенная энергия (E).
- Биты 5-7:
 - 000 (0) – суммарная энергия (S);
 - 001 (32) – по ночному (льготному) тарифу (N);
 - 010 (64) – по дневному тарифу (D);
 - 011 (96) – по пиковому тарифу (P);
 - 100 (128) – по резервному тарифу (R).

Пример.

Если нужно получить отпущенную энергию за предыдущий месяц по дневному тарифу, дополнительный параметр должен быть равен следующему числу:

$$01010110 \text{ (BIN)} = 0x56 \text{ (HEX)} = 86 \text{ (DEC)}$$

Чтобы быстро получить это число, нужно сложить соответствующие десятичные числа, указанные в расшифровке в скобках: $2+4+16+64=86$.

Для параметра **GP30t** (мощность, усредненная по 30-минутному интервалу) биты дополнительного параметра имеют следующее значение:

- Бит 4:
 - 0 (0 в формате DEC) – потребленная мощность;
 - 1 (16) – отпущенная мощность.

Канал с параметром **GP30t** предназначен для передачи значений мощности из памяти счетчика в архив TRACE MODE по следующему алгоритму.

Значения мощности за текущие и двое предыдущих суток 1 раз за 30 минут заносятся в архив счетчика (всего 144 значения). При первом включе-

нии канала все 144 значения записываются в буфер драйвера. Далее, при каждом пересчете канала одно значение из буфера драйвера заносится в архив TRACE MODE и присваивается каналу. Время последнего изменения канала при этом устанавливается равным концу соответствующего интервала вычисления средней мощности. После считывания всех данных из буфера каналу присваивается последнее занесенное в архив значение. Далее, через каждые 2 часа из счетчика в буфер драйвера считываются 4 значения, которые затем записываются в архив TRACE MODE по одному значению за такт пересчета канала.

Обмен со счетчиком ЦЭ6823М

Данный драйвер оформлен в виде драйвера t12.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – десятичное число, номер используемого для обмена последовательного порта (0 - COM1, 1 – COM2, ..., 15 – COM16);
- **Номер прибора** – десятичное число, задающее идентификатор прибора, каждому номеру соответствует одна строка – идентификатор (см. атрибут **ID прибора**);
- **Стартовая скорость** – значение выбирается из меню, определяет начальную скорость обмена по последовательному порту (в соответствии со стандартом МЭК 1107-96, скорость должна быть 300 бод, однако счетчик позволяет задать более высокую скорость);
- **Параметр** – десятичное число, отдельные биты этого числа определяют вариант считываемой физической величины, расшифровка отдельных битов:

Биты 0 и 1:

- 00 (0 в формате DEC) – нарастающим итогом;
- 01 (1) – за сутки;
- 10 или 11 (2 или 3) – за месяц.

Биты 2 и 3:

- 00 (0) – за текущий день или месяц;
- 01 (4) – за предыдущий день или месяц;
- 10 (8) – за день два дня назад или за месяц два месяца назад;
- 11 (12) – за день три дня назад или за месяц три месяца назад.

Бит 4:

- 0 (0) – отпущенная энергия;
- 1 (16) – потребленная энергия.

Пример

Если нужно получить потребленную энергию за предыдущий месяц, параметр должен быть равен следующему числу:

$$00010110_{\text{BIN}} = 0 \times 16_{\text{HEX}} = 22_{\text{DEC}}$$

Чтобы быстро получить это число, нужно сложить соответствующие десятичные числа, указанные в расшифровке в скобках: $2+4+16=22$.

- **Тариф** – десятичное число от 0 до 5, номер тарифа: 0 – суммарная энергия, 1 – энергия по первому тарифу, ..., 5 – энергия по 5 тарифу;
- **ID прибора** – строка-идентификатор прибора в сети с несколькими устройствами, обычно равна серийному номеру, указанному на корпусе прибора. Каждому номеру прибора (атрибут **Номер прибора**) соответствует свой ID. Если создается группа каналов с одинаковым ID (одинаковым номером прибора), то строку ID достаточно заполнить для одного (первого) создаваемого канала.

Обмен со счетчиком ЦЭ6822

Драйвер предназначен для обмена по протоколу **MT-4** и позволяет работать с электросчетчиками **ЦЭ6827M**, **ЦЭ6827M1**, **ЦЭ6822**.

Данный драйвер оформлен в виде драйвера t11.

Для последовательного интерфейса должны быть заданы следующие настройки (см. **Редактор параметров СОМ-порта**):

- скорость – 2400 бод;
- число информационных бит – 8;
- число стоповых бит – 2;
- четность – нет.

Рекомендуемое значение таймаута ожидания ответа – 500 мс.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого СОМ-порта, уменьшенный на 1; однобайтное десятичное число;
- **Номер подсети** – номер подсети (часть адреса прибора), однобайтное десятичное число;
- **Номер прибора** – номер прибора в подсети, однобайтное десятичное число;
- **Запрос** – тип запроса к прибору, значение выбирается из меню (см. таблицу ниже);
- **Параметр 1** – первый параметр запроса, значение зависит от типа

запроса (см. таблицу), однобайтное десятичное число;

- **Параметр 2** – второй параметр запроса, значение зависит от типа запроса (см. таблицу), однобайтное десятичное число.

Запрос	Описание	Тип	Параметр 1	Параметр 2
DeviceInfo	конфигурация счетчика	I	0 – код числа тарифов (0-T1; 1-T1, T2; 2-T1, T2, T3; 3-T1, T2, T3, T4) 1 – тарификация (0-автопереход на зимнее/летнее время; 1-использовать внешнюю тарификацию) 2 – Тип EEPROM 3 – флаг архива (0-нет получасовок, 1-есть получасовки) 4 – положение десятичной точки (0 - целые; 1 - до тысячных долей; 2 -до сотых; 1 - до десятых) 5 – тарификация выходных дней (1-есть)	-
MaxP_1/2Hour Day	максимум получасовой мощности за день	I	-	-
MaxP_1/2Hour Month	максимум получасовой мощности за месяц	I	-	-
Power	мгновенная мощность	I	-	-
PowerHalf-HourOdd	получасовые мощности (для нечетного месяца)	I	номер дня 1..31	номер получаса 1..48
PowerHalf-HourEven	получасовые мощности (для четного месяца)	I	номер дня 1..31	номер получаса 1..48
Tariff	показания по тарифам T1, T2, T3, T4	I	номер тарифа 1..4	-
TariffArchive	тарифные суммы по месяцам	I	номер тарифа	номер месяца

Запрос	Описание	Тип	Параметр 1	Параметр 2
ByteRAM	чтение/запись байта в ОЗУ	I, O	1-байтный адрес в ОЗУ	–
WordRAM	чтение/запись 2 байт в ОЗУ	I, O	1-байтный адрес в ОЗУ	–
ByteEPROM	чтение/запись байта из EPROM	I, O	старший байт адреса	младший байт адреса
WordEPROM	чтение/запись 2 байт из EPROM	I, O	старший байт адреса	младший байт адреса
DateTime	чтение/запись даты и времени	I, O	0 – дата 1 – время	при работе с датой: 0 – год 1 – месяц 2 – число 3 – день недели (0 - вс, 1- пн.,...) при работе со временем: 0 – часы 1 – минуты 2 – секунды

Дополнительная информация о значении флагов конфигурации и содержимом памяти приведена в описании протокола обмена с счетчиками фирмы **Энергомера** на основе модулей МТ-4.

Например, для запроса текущего показания по тарифу Т1 у прибора с адресом 1.02 нужно задать следующие настройки: **Номер подсети=1, Номер прибора=2, Запрос=Tariff, Параметр 1=1, Параметр 2=0.**

Обмен со счетчиком ЦЭ6850М

Драйвер оформлен в виде драйвера t12, протокол соответствует IEC 62056-21 (2002-05) в части режимов А-D (режим Е не поддерживается).

Протокол обмена со счетчиком встроен в Micro TRACE MODE 6 для Windows CE. Если COM-порт с назначением **Host OEM RS** задан, Микро МРВ использует встроенный драйвер с полной поддержкой МЭК (только для профессионального формата). Если COM-порт не задан, Микро МРВ

использует драйвер t12 с поддержкой МЭК посредством ручного конфигурирования переменных (запросы 0-36).

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер используемого COM-порта, уменьшенный на 1, однобайтное десятичное число;
- **Номер прибора** – уникальный для каждого прибора номер, однобайтное десятичное число. Каждому номеру должна соответствовать своя строка-идентификатор;
- **ID прибора** – идентификатор прибора в сети, строка длиной до 16 символов;
- **Стартовая скорость** – начальная скорость обмена, значение выбирается из меню в соответствии с настройкой прибора. Если выбрано значение с окончанием «+», команды окончания сеанса и инициализации не посылаются (в начале очередного сеанса ID прибора посылается только в том случае, если он изменился или была ошибка обмена). Для значений без плюса команды инициализации и разрыва связи посылаются всегда;
- **Запрос** – тип запроса к прибору, значение выбирается из меню (см. описание запросов ниже);
- **Параметр 1** – первый параметр запроса, значение зависит от типа запроса (см. ниже);
- **Параметр 2** – второй параметр запроса, значение зависит от типа запроса (см. ниже).

Типы запросов и их параметры (номер запроса – в соответствии с plc_ce6850m2.tmc):

- 0, **STIME** – широковещательная команда коррекции хода часов, канал должен быть типа OUTPUT;
- 1, **Exyst** – запрос количества учтенной энергии, параметры имеют следующее значение:
 - **Параметр 1** – набор битовых флагов, задающий условия чтения: значение битов 0 и 1 – интервал (0 – нарастающим итогом, 1 – за сутки, 2 или 3 – за месяц); бит 2 – активная (0) или реактивная (1) энергия; бит 3 – отпущенная (0) или потребленная (1) энергия. При запросах энергии за день и за месяц используется дополнительный параметр для указания номера дня (месяца) – см. **EnerNparam**;
 - **Параметр 2** – номер тарифа, 0 – суммарная энергия, 1 – энергия по первому тарифу, 2 – по второму тарифу, ..., 5 – энергия, накопленная в общем регистре;
- 2, **Graph** – запрос конкретного значения из графика нагрузки, дата

для чтения графика задается с помощью запросов **GraYear**, **GraMonth** и **GraDay**. Значение параметров:

- **Параметр 1** – набор битовых флагов; бит 0 – активная (0) или реактивная (1) энергия; бит 1 – отпущенная (0) или потребленная (1) энергия;
- **Параметр 2** – номер архивной записи в сутках;
- 3, **Gra25** – график нагрузки потерянного часа при переходе с летнего на зимнее время. Значение параметров:
 - **Параметр 1** – набор битовых флагов, бит 0 – активная (0) или реактивная (1) энергия; бит 1 – отпущенная (0) или потребленная (1) энергия;
 - **Параметр 2** – номер записи в архиве;
- 4, **Ener3** – энергия за последний заверченный трехминутный интервал; **Параметр 1** – тип энергии (0 – активная потребленная, 1 – активная отпущенная, 2 – реактивная потребленная, 3 – реактивная отпущенная);
- 5, **Power** – мгновенная мощность, значение параметров:
 - **Параметр 1** – тип мощности (0 – полная, 1 – активная, 2 – реактивная);
 - **Параметр 2** – фаза (0 – фаза А, 1 – фаза В, 2 – фаза С, 3 – суммарная мощность по всем фазам);
- 6, **Curre** – значение тока; **Параметр 1** – номер фазы (0 – фаза А, 1 – фаза В, 2 – фаза С);
- 7, **Volta** – значение напряжения; **Параметр 1** – номер фазы (0 – фаза А, 1 – фаза В, 2 – фаза С);
- 8, **Cos_f** – коэффициент активной мощности; **Параметр 1** – номер фазы (0 – фаза А, 1 – фаза В, 2 – фаза С, 3 – ABC);
- 9, **Sin_f** – коэффициент реактивной мощности; **Параметр 1** – номер фазы (0 – фаза А, 1 – фаза В, 2 – фаза С, 3 – ABC);
- 10, **CorUU** – углы между векторами напряжений; **Параметр 1** – тип вектора (0 – между векторами А и В, 1 – ВС, 2 – СА);
- 11, **CorIU** – углы между векторами тока и напряжения; **Параметр 1** – фаза (0 – по фазе А, 1 – по фазе В, 2 – по фазе С);
- 12, **Frequ** – частота сети в герцах, дополнительные параметры не используются и должны быть равны 0;
- 13, **Stat** – состояние счетчика, 16-битное число, расшифровку значений отдельных битов см. в руководстве по эксплуатации счетчика электрической энергии ЦЭ6850М. Дополнительные параметры не используются и должны быть равны 0;
- 14, **Elos** – энергия потерь; **Параметр 1** – тип потерь (0 – потери при отпуске, 1 – потери при потреблении);
- 15, **EnerNparam** – запрос для передачи номера дня (месяца) в драйвер (для запроса **Exyst**). Дополнительные параметры не ис-

ID=<ID прибора>;N=<имя команды или параметра>;[TMOUТ=<время бездействия после посылки команды, мс, DEC>;][FORMAT=<формат времени>;][PASSW=<пароль>]

Строка формата времени может содержать следующие символы:

- **D** – день;
- **M** – месяц;
- **Y** – год;
- **h** – часы;
- **m** – минуты;
- **s** – секунды;
- **x** – пропустить значение времени;
- **v** – интерпретировать как значение;
- **t** – инициализировать приемную структуру (создается в памяти) текущим временем (**DMY**, символ «**t**» может быть указан только в нулевой позиции строки формата);
- **T** – инициализировать приемную структуру текущим временем (**DMYhms**, символ «**T**» может быть указан только в нулевой позиции строки формата);
- **+** – интерпретировать полученное значение месяца 0...11 как 1...12 (символ «**+**» может быть указан только в нулевой, первой или второй позиции строки формата);
- **G** – запрет коррекции с учетом зимнего/летнего времени (символ «**G**» может быть указан только в нулевой, первой или второй позиции строки формата).

Если возвращается массив, то для передачи его данных в каналы TRACE MODE можно использовать следующие способы:

- создать необходимое число одинаково сконфигурированных каналов (в том числе с одинаковым значением атрибута **Параметр 1**). В каждый такой канал записывается число, которое в массиве имеет порядковый номер, равный значению атрибута **Параметр 2** канала;
- создать единственный канал и привязать его к каналу CALL.ChGroupReq – в этом случае данные массива записываются в аргументы CALL.ChGroupReq. Если элемент массива содержит как значение, так и метку времени, аргументы создаются парами (в первый записывается значение, во второй – метка времени). Если **Параметр 2** равен 0, в аргументы записываются все данные массива, в противном случае – только те элементы, для которых остаток от деления их номера в массиве на **Параметр 2** равен 0. Номер аргумента, с которого начинается запись, равен сумме значений атрибутов **Номер прибора** и 34; атрибут 34 задает шаг записи по аргументам (если 0, шаг равен 1 и т.д.).

Для запроса 60 выражение «**N**=...» задается следующим образом:

N=<имя параметра> ([<команда>])

Команда представляет собой строку подстановок:

%<символ>%<символ>...

В канале INPUT могут использоваться следующие подстановки (временные параметры – из структуры времени (см. ниже)):

%P – <день месяца>.<номер месяца>.<2 цифры года>.<номер 30-минутного интервала с начала суток>;

%p – <день месяца>.<номер месяца>.<2 цифры года>.<номер 5-минутного интервала с начала суток>;

%E – <день месяца>.<номер месяца>.<2 цифры года>.<Параметр 2>;

%G – <Параметр 2>;

%g – <Параметр 1>;

%M – минуты;

%H – часы;

%S – секунды;

%d – день месяца;

%m – номер месяца;

%n – номер предыдущего месяца;

%y – 2 цифры года;

%Y – 4 цифры года;

%K – если **Параметр 2** больше номера месяца – 2 цифры предыдущего года, в противном случае – 2 цифры текущего года;

%k – подстановка зависит от номера появления **%k** в строке:

- первое появление – <Параметр 1>;
- второе появление – значение младшего полубайта атрибута **Параметр 2**;
- третье появление – 2 цифры года;

%N – <день месяца>.<номер месяца>.<2 цифры года>.<номер Параметр2-минутного интервала с начала суток>;

%B – зарезервировано;

%b – зарезервировано;

%<любой символ, кроме перечисленных выше> – указанный символ.

В канале OUTPUT могут использоваться: %G, %g, %M, %H, %S, %n, %m, %y, %Y, %d, %B, %b, %<любой символ, кроме перечисленных выше> и %P (в отличие от канала INPUT, вместо %P в канале OUTPUT подставляется выходное значение канала (9, Q)).

Для запроса 60 в отсутствие команды (скобки пусты) подставляется <Параметр 1>.<Параметр 2> (как в канале чтения, так и в канале записи).

Для запросов, отличных от 60, выражение «N=...» также может быть задано как N=<имя параметра>(<команда>) (как в канале чтения, так и в канале записи). При этом скобки могут быть пусты или содержать %G или %<любой символ, кроме перечисленных выше>. Если скобки пусты или содержат %G, подставляется выходное значение канала (9, Q).

Структура времени по умолчанию инициализируется текущим временем. Для изменения используется команда «%~», результат которой зависит от значения атрибута **Параметр 1**:

- 0 – <текущее время> - 1ч
- 1 – <текущее время> - 1 день
- 2 – <текущее время> - 86400*<текущий день>
- 3 – <текущее время> - 0.5ч
- 4 – <текущее время> - 10мин
- 5 – <текущее время> - 5мин
- 6 – <текущее время> - 1мин
- 16 – <текущее время> - (Параметр 2 + 1) ч
- 17 – <текущее время> - (Параметр 2 + 1) дней
- 33 – смещение на начало дня и на Параметр 2 дней назад

Запрос 63

Запрос 63, **USER_DEF_15** используется, если прибор переключен в режим программирования.

Формат строки **ID прибора** для запроса 63, **USER_DEF_15**:

ID=<ID прибора>;N=<имя параметра>;...;N=<имя параметра>

В соответствии с заданным числом параметров, МРВ генерирует запросы к прибору и записывает полученные данные в каналы аналогично запросам 48...62.

Запросы 128...131

Запросы 128 (**REQ_READ**), 129 (**PARSE_READ**), 130 (**MULTI_PARSE**) и 131 (**REQ_READ(file)**) предназначены для обычного режима работы прибора. В этом режиме в приемную структуру (создается в памяти) считываются параметры, заданные производителем прибора.

Запрос 128 или 131 инициализирует обычный режим и считывает все данные (запрос 131, кроме того, записывает данные в файл с именем прибора).

Если канал с запросом 128, **REQ_READ** привязан к каналу CALL.ChGroupReq, то, кроме инициализации и считывания данных из прибора, из приемной структуры в аргументы CALL.ChGroupReq записываются параметры, заданные в строке **ID прибора**. Формат строки:

ID=<ID прибора>;N=<имя параметра>;...;N=<имя параметра>

Если канал с запросом 130, **MULTI_PARSE** привязан к каналу CALL.ChGroupReq, то из приемной структуры в аргументы CALL.ChGroupReq записываются параметры, заданные в строке **ID прибора**. Формат строки:

ID=<ID прибора>;N=<имя параметра>;...;N=<имя параметра>

Формат строки **ID прибора** для запроса 129, **PARSE_READ**:

ID=<ID прибора>;N=<имя параметра>;[FORMAT=<формат времени>]

Формат времени задается так же, как для запросов 48...63.

Если канал с запросом 129 привязан к CALL.ChGroupReq, данные записываются в аргументы CALL.ChGroupReq, в противном случае производится поиск каналов для записи.

Запросы 40...43

Формат строки **ID прибора** для запросов 40...43 (**USER_ARC_00...USER_ARC_03**):

ID=<ID прибора>;N=<имя, по которому считываются значения>;T=<имя, по которому считываются метки времени>;TF=<формат времени>

Строка формата времени может содержать те же символы, что и для запросов 48...63, за исключением символа «V».

Запросы 40 и 41 используются в режиме программирования прибора, 42 и 43 – в обычном режиме (для выборки из приемной структуры).

Канал должен быть привязан к CALL.ChGroupReq. Номер аргумента, с

которого начинается запись данных, задается атрибутом **Номер прибора** (четный аргумент – значение, нечетный – метка времени).

Если число N_T считанных меток времени меньше числа N_V считанных значений, то выполняется коррекция меток времени:

$$T_i \approx T_0 + \frac{T_{N_T} - T_0}{N_V - N_V / N_T} i$$

Для всех запросов в атрибут 45, **T** канала CALL.ChGroupReq записывается время последнего успешного приема, а в атрибут 1, **A** – число считанных значений.

Обмен с контроллерами фирмы ВОЛМАГ

Обмен с контроллерами серии КОНТРАСТ

Параметры шаблонов каналов обмена с контроллерами серии КОНТРАСТ (КР-500, МК-500, КР-300ИШ, КР-300ИП, КР-300Ш, КР-300М, РК-131/300) задаются в соответствующем редакторе.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Порт** – номер последовательного порта обмена (0 – COM1, ..., 31 – COM32). Этот параметр задается в формате HEX;
- **Модуль** – номер устройства в формате HEX (0-FF);
- **Алгоблок** – десятичный номер алгоблока (0-255);
- **Номер** – десятичный номер входа/выхода алгоблока (0-255);
- **Тип** – тип переменной (в соответствии с протоколом).

Для обмена необходимо настроить последовательный порт (см. раздел **Редактор параметров СОМ-порта**).

Для обмена с алгоблоками типа FLOATир и HEXир атрибуты **Алгоблок** и **Номер** обозначают соответственно номер буфера и номер входа ИНР, и отсчет обоих параметров начинается с нуля, а не с единицы, как для остальных алгоблоков. При этом ID канала с номером 0 входа ИНР должен быть меньше ID всех остальных каналов, настроенных на получение данных от того же алгоблока.

Настройка каналов опроса и управления контурами регулирования контроллера КР-300

Настройка **Тип** у этих каналов должна быть КОНТУР.

Настройка **Алгоблок** должна задавать номер контура.

Группа каналов для ЗАПРОСА ПАРАМЕТРОВ одного контура регулирования какого-либо контроллера должна быть сформирована так, чтобы между ними (по индексам каналов) не было других каналов обращения к тому же контроллеру.

Значения настройки **Номер** для этих каналов должны быть следующими (по порядку возрастания индексов каналов):

Назначение	НОМЕР
Значение ручного задания	0
Текущее значение задания	1

Назначение	НОМЕР
Значение входа	2
Значение рассогласования	3
Значение выхода	4
Режим работы контура	5

Канал выдачи команд УСТАНОВКИ РЕЖИМА контура (с кодами команд от 01 до 09) должен иметь настройку **Номер**=5.

Значение канала соответствует коду команды.

Назначение	Значение канала
Установка режима ВЗ (внешнее задание)	01
Установка режима ПЗ (программное задание)	02
Установка режима РЗ (ручное задание)	03
Переход в автоматический режим	04
Установка режима РУ (ручное управление)	05
Установка режима ДУ (дистанционное управление)	06
Отмена дистанционного управления	07
Установка режима КУ (каскадное управление)	08
Установка режима ЛУ (локальное управление)	09

Каналы УСТАНОВКИ СИГНАЛА РУЧНОГО ЗАДАНИЯ и ВЫХОДНОГО СИГНАЛА в контуре передают эти сигналы своими значениями.

Значения настройки **Номер** для этих каналов должны быть следующими:

Назначение	НОМЕР
Установка сигнала ручного задания	0
Установка выходного сигнала	4

Для обращения к входам/выходам алгоблоков с номерами алгоблоков более 255 необходимо использовать в настройке **Тип** типы Б_Вход(256-1023) и Б_Выход(256-1023). При этом номер алгоблока будет формироваться как сумма параметра, задаваемого в настройке **Алгоблок** канала, и 256.

Для обращения к переменным языка ПРОТЕКСТ используются типы переменных ПроТекст(0-255), ПроТекст (256-511), ПроТекст (512-767), ПроТекст (768-1023).

Тип канала (I/O) определяет производимую операцию – чтение/запись.

Формат переменной задается в настройке **Алгоблок** числом, соответствующим кодировке формата в протоколе МАГИСТР (согласно таблице п.9.9.1. описания КОНТ. 421457.001 РЭ):

- целая короткая ЦК – 0;
- целая стандартная ЦС – 1;

- целая длинная ЦД – 2;
- вещественная ВЦ – 3;
- таймер секунднй ТМС – 4;
- таймер десяти миллисекунднй ТММ – 5;
- дискретная групповая ДГ – 6;
- дискретная словная ДС – 7;
- дискретная длинная ДД – 8.

Номер переменной формируется как сумма параметра настройки **Номер** и смещения, соответствующего выбранному типу переменной: 0, 256, 512, 768.

Обмен с КР-500 по IP

Драйвер оформлен в виде драйвера t12, в его состав входят библиотеки t12s41.dll и media41.dll.

Шаблоны каналов обмена с контроллером создаются в группе **Источники/Приемники – PLC – KR_500**.

Кроме атрибутов, общих для всех источников/приемников (см. **Редакторы источников (приемников)**), а также **Шаблоны каналов обмена**), в редакторе шаблона канала обмена задаются следующие атрибуты:

- **Протокол** – протокол обмена (TCP или UDP);
- **Контроллер** – номер контроллера (0...31, как для контроллеров серии КОНТРАСТ – см. **Обмен с контроллерами серии КОНТРАСТ**);
- **Алгоблок** – то же, что и для контроллеров серии КОНТРАСТ;
- **Номер** – то же, что и для контроллеров серии КОНТРАСТ;
- **Тип** – то же, что и для контроллеров серии КОНТРАСТ;
- **Дополнительно** – адрес прибора в следующем формате:

```
<Device_IP>:<Device_Port>-<Timeout>;PWD=<n>
```

где

- **Device_IP** – IP-адрес контроллера;
- **Device_Port** – номер порта, обычно равен 5000;
- **Timeout** – таймаут в мс для операции сетевого обмена; если не указан, то по умолчанию принимается равным 1с.

Работа с таймер-календарем контроллера (**ТМК**):

- **Номер** для каналов чтения:
 - 0 – час;
 - 1 – минуты;
 - 2 – секунды;
 - 3 – день;
 - 4 – месяц;
 - 5 – год;
 - 6 – день недели;
- **Номер** для каналов записи:
 - 0 – установка времени во всех контроллерах;
 - 1 – установка времени в текущем контроллере.

Чтение ошибок контроллера (**Err_KR_300I**):

- **Алгоблок** задает номер ошибки;
- **Номер** определяет считываемую информацию:
 - 0 – код ошибки (мл. байт) и флаг ошибки (ст. байт);
 - 1 – уточнение ошибки;
 - 2 – день (мл.) и месяц (ст.) начала ошибки;
 - 3 – минута (мл.) и час (ст.) начала ошибки;
 - 4 – день (мл.) и месяц (ст.) конца ошибки;
 - 5 – минута (мл.) и час (ст.) конца ошибки;
 - 6 – секунда начала (мл.) и секунда конца (ст.) ошибки.

На рисунке выше показаны настройки канала для чтения значения ИНР 2 из прибора с номером 1, IP-адрес – 192.168.0.100, порт – 5000, таймаут – 500мс.

Попытки подключения к одному и тому же устройству производятся не чаще, чем 1 раз в 10 с. Если число ошибочных попыток превышает заданное по умолчанию, то период попыток увеличивается до 1 минуты.

Канал INPUT с типом переменной ОШИБКА работает в зависимости от значения атрибута НОМЕР:

- НОМЕР=0 – возврат кода ошибки в ответе контроллера;
- НОМЕР=1 – время с последнего успешного обмена;

- НОМЕР=2 – индикация состояния обмена: 0 – есть, 1 – нет;
- НОМЕР=8 – альтернативное подключение.

Канал OUTPUT с типом переменной ОШИБКА обрабатывает при любом изменении значения канала в зависимости от значения настройки НОМЕР:

- НОМЕР=0 – сброс кода ошибки в канале INPUT с типом переменной ОШИБКА;
- НОМЕР=1 – запрет обмена;
- НОМЕР=2 – разрешение обмена;
- НОМЕР=3 – инверсия запрета/разрешения обмена.

Контроллеры могут резервироваться с помощью канала OUTPUT с типом переменной ОШИБКА. Попытка подключения производится ближайшим активным каналом после канала OUTPUT с типом переменной ОШИБКА:

- НОМЕР=8 – задать подключение по альтернативному IP-адресу;
- НОМЕР=9 – задать подключение по альтернативному IP-адресу и разрешить обмен.

Смысл значений канала:

- 16 – принудительное переподключение;
- 17 – переподключение с изменением младшего бита в 4-й тетраде IP-адреса;
- 18 – переподключение с изменением младшего бита в 3-й тетраде IP-адреса;
- 19 – переподключение с изменением младших битов в 3-й и 4-й тетрадах IP-адреса.

Особенности устройств

Устройства ICP-DAS

Контроллеры XPAC, WinPac, LinPac и WinCon

Модули в основном крейте

Для работы с модулями **i-87xxx**, установленными в основной крейт контроллера **WinCon**, используются каналы подтипа 8 (см. **Подтип 8**). При этом назначение COM-порта для обмена с УСО должно быть **Host OEM RS** (см. **Редактор параметров COM-порта**), а для инициализации внутреннего COM-порта (COM0) контроллера **WinCon** нужно создать порт COM1 с назначением **Host OEM RS** и параметрами 115200 8-1-n.

Следующие типы каналов в реальном времени читают/пишут свои значения в зависимости от оборудования, на котором запускается монитор реального времени.

Назначение атрибутов каналов (для всех контроллеров):

- **Мультиплексор** (96, **C3**) – номер слота, считая с 1.
- **Канал** (95, **C2**) – номер канала, считая с 0.

Для работы с модулями **i-87xxx**, в которых одновременно есть аналоговые сигналы и дискретные или частотные и дискретные (например, 87005 или 87016), номер слота должен быть увеличен на 128.

Для контроллеров предусмотрено автоматическое определение модулей.

Ниже после наименования источника/приемника в скобках указан подтип и дополнение к подтипу (см. также **Подтип 1** и **Подтип 2**):

- **OEM_AI** (1.9) (**WinCon_AI** (1.32)) – для модулей AI таких, как 8017/8014/87017 (в WinCon – только для платы 8017);

Здесь и везде далее каналы с префиксом WinCON используются только в контроллерах WinPAC/XPAC/WinCon/LinPac.

Атрибут канала **GAIN/REG** (97, **C4**) задает входной диапазон (например, для 8017 **C4=0** задает диапазон +/- 10V);

- **OEM_AO** (1.23) (**WinCon_AO** (1.33)) – для модулей AO таких, как 8024/87024 (в WinCon – только для платы 8024). Если источник привязан к каналу INPUT, выполняется чтение READBACK.

Атрибут канала **GAIN/REG** (97, **C4**) задает диапазон/тип вывода (для 8024: **C4**=0 – напряжение, **C4**=1 – ток);

- **A_I/O** (1.14) – чтение/управление для счетчиков/частотомеров 8084 (8080 WinCon) – 87084 (WinPAC/XPAC) – 87xxx (DI/DIO, счетчики в платах дискретного ввода).

Канал INPUT.

Для **i-80xx** атрибут канала **GAIN/REG** (97, **C4**) задает режим (например, для 8084: **C4**=3 – Frequency, **C4**=4 – Up Counter). Если **GAIN/REG** (97, **C4**)=0, считывается текущий режим платы.

Для **i-87xxx** (DI/DIO, счетчики в платах дискретного ввода) номер канала (95, **C2**) начинается с 32 (32 – 0, 33 – 1 и т.д.) (изменяется автоматически).

Канал OUTPUT:

- запись любого значения приводит к сбросу счетчика.

Если источник привязан к целочисленному каналу то:

- при переполнении счетчика в атрибуте **QDS** (61, **Nind**) устанавливается бит переполнения;
- запись 0 в атрибут **Интервал** (7, **P**) приводит к сбросу счетчика и сбросу бита переполнения;
- **OEM_Command** (2.25) – предназначен для чтения/записи дополнительных параметров плат счетчиков 8084/8080.

Чтение: значение целочисленное LowPassFilter (**GAIN/REG** {ia.c[4]}=0/1 – фильтр выключен/включен).

Запись:

- значение>0 – установить LowPassFilter=значение и активировать фильтр;
- значение=0 – деактивировать фильтр;
- **OEM_Register** (1.26) – преобразуется в канал 1.14 **A_I/O**, если обнаружена плата 8084, 87084 или обнаружена карта 87xxx или C5=80 или C5=84 (если обнаружена плата 8093, канал **OEM_Register** не преобразуется в **A_I/O**);
- (*) **OEM_DI_8** (2.6), **OEM_DI_16** (2.7), **OEM_DI_32** (2.8) (**WinCon_DI_8** (2.32), **WinCon_DI_16** (2.34), **WinCon_DI_32** (2.36)) – для модулей DI;
- (*) **OEM_DO_8** (2.22), **OEM_DO_16** (2.23), **OEM_DO_32** (2.24) (**WinCon_DO_8** (2.33), **WinCon_DO_16** (2.35), **WinCon_DO_32** (2.37)) – для модулей DO (если привязан к каналу INPUT – чтение READBACK).

(*) Если детектируется модуль **8050** (поддерживается только в **WinPac**), то для инициализации обмена и задания направления работы битов атрибут **Состояние (STATE)** (97, **C4**)

должен иметь одно из следующих значений (в скобках указан номер значения в списке):

I-I (1) – все биты – IN;

I-O (2) – биты 0-3 – OUT, биты 4-15 – IN;

I-O-I (3) – биты 0-7 – OUT, биты 8-15 – IN;

O-O-I (7) – биты 0-11 – OUT, биты 12-15 – IN;

O-O-O (8) – все биты – OUT.

Если автоопределение модуля по каким-либо причинам не срабатывает, анализируется атрибут канала **ALT/REG** (98, **C5**), который задает номер платы (например, 14 для работы с 8014). Чтобы указать, что обмен с платой производится по последовательному интерфейсу (плата может выпускаться с параллельным или последовательным интерфейсом или только с последовательным), нужно прибавить 100 к **ALT/REG** (для каналов подтипа 1 и подтипа 2: если базовый адрес $\geq 0x100$, то плата работает по последовательному интерфейсу).

Модули в крейте расширения

Для всех контроллеров: для работы с модулями **I-87xxx**, установленными в крейт расширения (i87k), используются стандартные каналы подтипа 8 (см. **Подтип 8**), COM-порт – мастер.

Модули **i-8xxx** не могут работать во внешней корзине (и не должны).

Модули i-70xx / i-87xxx

Для обмена используются каналы подтипа 8 (см. **Подтип 8**).

Устаревшие механизмы поддержки устройств ICP DAS

Плата A-826

Для работы с аналоговыми 16-битовыми входами платы **ICP DAS A-826** используются каналы **PCL_channel** (см. **Подтип 1**) со значениями настройки **Канал (CH)** в диапазоне от 128 до 143. Нумерация каналов последовательная, начиная со 128.

Контроллеры i-7188

В контроллерах серий i-7188 используются каналы **DI_port**, **DO_port** и **OEM_AI** (в том числе для чтения/записи через шину расширения, см. **Подтип 2** и **Подтип 1**).

Атрибут **Канал** канала **DI_port** задает чтение следующих сигналов:

- 1 – в младшие 3 бита канала считываются значения соответственно DO1-DO3;
- 2 – DI2;
- 3 – DI3;
- 4 – DIO4;
- 9 – DIO9;
- 14 – DIO14;
- 16 – порт 0xff74;
- 0 – байт из порта с номером (**Базовый адрес**).

Атрибут **Канал** канала **DO_port** задает запись по следующим направлениям:

- 1-3 – соответственно в DO1-DO3;
- 4, 9, 14 – соответственно в DIO4, DIO9 и DIO14;
- 16 – в порт 0xff74;
- 0 – байт в порт с номером (**Базовый адрес**).

Для чтения аналоговых данных через плату расширения используется канал **OEM_AI**. Атрибут **Канал** канала **OEM_AI** задает номер канала на плате.

Устройства фирмы ТЕКОН

Модуль D40Im

Модуль импульсного вывода **D40Im** (ТЕКОН) в режиме ШИМ имеет следующие особенности:

- вывод осуществляется каналами **АО_MFC** (см. **Подтип 1**);
- значение канала (от 1 до 60000) задает длительность импульса в мс; 65535 – установка выхода в состояние статической единицы, 0 – установка выхода в состояние статического нуля;
- атрибут **Мультиплексор** = 1,
- каждый выход модуля может управляться каналом **DO_MFC** (см. **Подтип 2**) до тех пор, пока на него не будет подан сигнал от канала **АО_MFC**. После этого сигналы от канала **DO_MFC** на конкретный дискретный выход не проходят.

Модуль F24

При работе с платой ввода частотно-импульсных сигналов **F24**, канал **AI_MFC** (см. **Подтип 1**) с атрибутом **Мультиплексор**=3 используется для считывания 32-разрядных регистров счетчиков. Канал **АО_MFC** с атрибутом **Мультиплексор**=3 используется для обнуления счетчика.

Модуль L16i

Каналы **AI_MFC** (см. **Подтип 1**) с атрибутом **Мультиплексор**=10 используются для считывания 16-ти аналоговых каналов платы L16i и одного канала измерения температуры окружающей среды. Каналы адресуются значениями атрибута **Канал** с 0 по 16. Последний канал с атрибутом **Канал**=16 считывает значение температуры окружающей среды с цифрового датчика, установленного на плате.

Сконфигурированному диапазону канала соответствует диапазон значений 0...16383. Полный (предельный) диапазон, с учетом допустимого 5-процентного превышения сконфигурированного для канала диапазона, составляет от -1024 до 17408.

Для данных каналов диагностируется обрыв подключения датчика – в этом случае каналу устанавливается признак аппаратной недостоверности.

Устройства ADVANTECH

Платы аналогового ввода/вывода контроллера ADAM5510

Платы аналогового ввода/вывода контроллера ADAM5510 надо настроить на диапазон или тип сигнала. Для этого можно воспользоваться либо утилитами производства ADVANTECH, либо установить соответствующие значения атрибутов **GAIN/REG** каналов **ADAM5510_AI**, **ADAM5510_AO** (см. Подтип 1). По умолчанию **GAIN/REG=0**. Такое значение этого атрибута не изменяет установленного ранее диапазона. Чтобы изменить конфигурацию любого модуля, надо хотя бы в одном канале, созданном для обмена с этим модулем, установить соответствующее значение **GAIN/REG** (см. приведенные ниже таблицы). Значение атрибута задается в десятичном формате.

Для платы ADAM 5017 настройка **ALT/REG** должна быть равна 17; для платы ADAM 5017H – 81.

ADAM-5017. Диапазоны ввода

Код диапазона (DEC)	Диапазон ввода
9	$\pm 10 \text{ V}$
10	$\pm 5 \text{ V}$
11	$\pm 1 \text{ V}$
12	$\pm 500 \text{ mV}$
13	$\pm 150 \text{ mV}$
14	$\pm 20 \text{ mA}$

ADAM-5018. Диапазоны ввода

Код диапазона (DEC)	Диапазон ввода
1	$\pm 15 \text{ mV}$
2	$\pm 50 \text{ mV}$
3	$\pm 100 \text{ mV}$
4	$\pm 500 \text{ mV}$
5	$\pm 1 \text{ V}$
6	$\pm 2.5 \text{ V}$
7	$\pm 20 \text{ mA}$
8	Не используется

Код диапазона (DEC)	Диапазон ввода
15	Термопара J-типа 0-760 °C
16	Термопара K-типа 0-1000 °C
17	Термопара T-типа -100-400 °C
18	Термопара E-типа 0-1000 °C
19	Термопара R-типа 500-1750 °C
20	Термопара S-типа 500-1750 °C
21	Термопара B-типа 500-1800 °C

ADAM-5017H. Диапазоны ввода

Код диапазона (DEC)	Диапазон ввода
1	± 10 V
2	0 - 10 V
3	± 5 V
4	0 - 5 V
5	± 2.5 V
6	0 - 2.5 V
7	± 1 V
8	0 - 1 V
9	± 500 mV
10	0 - 500 mV
11	4 - 20 mA
12	0 - 20 mA

Значения, приведенные в этой таблице, оценочные; реальные значения могут несколько отличаться от них.

ADAM-5024. Диапазоны вывода

Код диапазона (DEC)	Диапазон вывода
49	0 - 20 mA
50	4 - 20 mA
51	0 - 10 V

Для кодов 49 и 50 значение, посылаемое в канал **ADAM5510_AO**, переводится в физические величины тока и напряжения на соответствующих выходах платы по следующим формулам:

$$\langle \text{Величина тока} \rangle = \langle \text{значение канала} \rangle / 1000 \text{ [mA]}$$

$$\langle \text{Величина напряжения} \rangle = \langle \text{значение канала} \rangle / 2000 \text{ [V]}$$

Для кода 51 значение, посылаемое в канал **ADAM5510_AO**, переводится в физические величины тока и напряжения на соответствующих выходах платы по следующим формулам:

$$\langle \text{Величина тока} \rangle = \langle \text{значение канала} \rangle / 500 \text{ [mA]}$$

$$\langle \text{Величина напряжения} \rangle = \langle \text{значение канала} \rangle / 1000 \text{ [V]}$$

ADAM-5013. Диапазоны ввода терморезисторов

Код диапазона (DEC)	Диапазон ввода
33	100 Ом Pt терморезистор -100 – 100 °C a=0.00385
34	100 Ом Pt терморезистор 0 – 100 °C a=0.00385
35	100 Ом Pt терморезистор 0 – 200 °C a=0.00385
36	100 Ом Pt терморезистор 0 – 600 °C a=0.00385
37	100 Ом Pt терморезистор -100 – 100 °C a=0.00392
38	100 Ом Pt терморезистор 0 – 100 °C a=0.00392
39	100 Ом Pt терморезистор 0 – 200 °C a=0.00392
40	100 Ом Pt терморезистор 0 – 600 °C a=0.00392
41	120 Ом Ni терморезистор -80 – 100 °C
42	120 Ом Ni терморезистор 0 – 100 °C

Плата ADAM-5080 контроллера ADAM5510

Для обмена с платой счетчиков **ADAM-5080** контроллера **ADAM5510** создаются каналы **OEM_AI** типа INPUT и каналы **OEM_AO** типа OUTPUT (см. **Подтип 1**).

Атрибут **ALT/REG** этих каналов должен быть равен 80 (DEC). Данный атрибут указывает тип платы.

Функция канала зависит от атрибута **Канал** следующим образом.

Для канала **OEM_AI**:

- если **Канал** < 64, значение канала равно текущему значе-

нию счетчика с номером **Канал**;

- если **Канал** ≥ 64 , канал принимает значение 1 при переполнении счетчика с номером **Канал-64**, в противном случае – 0.

Для канала **OEM_AO**:

- если **Канал** < 64 , канал управляет запуском/остановом счетчика с номером **Канал**. При послыке в канал 0 счетчик запускается, при послыке 1 – останавливается;
- если **Канал** ≥ 64 , значение, посланное в канал, задает начальное значение счетчика с номером **Канал-64**. При изменении значения хотя бы одного такого канала все остальные счетчики платы принимают свои начальные значения.

Для модуля **ADAM-5080**, работающего в режиме определения частоты входного сигнала, значение соответствующего канала **Frq/Counter** (см. **Подтип 8**) необходимо для корректного отображения делить на 100 (этот канал получает от модуля данные, завышенные в 100 раз – сотни герц вместо единиц герц).

Плата ADAM-5050

Для шаблона канала **ADAM5510_DI_16** (см. **Подтип 2**) нужно выбрать **Состояние=I-I-I**; для шаблона канала **ADAM5510_DO_16** – **Состояние=по**.

Модуль ADAM-5013

Для каналов **Ai8**, **Ain** (см. **Подтип 8**) в модуле **ADAM-5013** номер канала надо задавать со смещением на 0x20 (например, вместо 2 установить 22).

Плата PCL-839

Для работы с платой **PCL-839** (3-х координатное управление шаговым двигателем) используются следующие каналы (см. **Подтип 1** и **Подтип 2**):

- **OEM_Register** типа I и O;
- **OEM_Status** типа I;
- **OEM_Command** типа O;
- **DI_port** типа I;
- **DO_port** типа O.

Каналами **OEM_Register** типа OUTPUT производится запись, а типа INPUT – опрос текущих значений регистров R0...R7. При этом адрес каналов платы задается атрибутом **Мультиплексор (MUX)**, а адрес реги-

стра – атрибутом **Канал (CH)**. Схема настройки данных каналов приводится в таблице 1.

Таблица 1

Тип	Канал	MUX	CH	Канал	Регистр
I/O	OEM_Register	0	0	1	R0: Счетчик
I/O	OEM_Register	0	1	1	R1: регистр FL
I/O	OEM_Register	0	2	1	R2: регистр FL
I/O	OEM_Register	0	4	1	R4: Регистр увеличения/ уменьшения скорости
I/O	OEM_Register	0	6	1	R6: Регистр перехода пилы
I/O	OEM_Register	0	7	1	R7: Регистр множителя
I/O	OEM_Register	1	0	2	R0: Счетчик
I/O	OEM_Register	1	1	2	R1: регистр FL
I/O	OEM_Register	1	2	2	R2: регистр FL
I/O	OEM_Register	1	4	2	R4: Регистр увеличения/ уменьшения скорости
I/O	OEM_Register	1	6	2	R6: Регистр перехода пилы
I/O	OEM_Register	1	7	2	R7: Регистр множителя
I/O	OEM_Register	2	0	3	R0: Счетчик
I/O	OEM_Register	2	1	3	R1: регистр FL
I/O	OEM_Register	2	2	3	R2: регистр FL
I/O	OEM_Register	2	4	3	R4: Регистр увеличения/ уменьшения скорости
I/O	OEM_Register	2	6	3	R6: Регистр перехода пилы
I/O	OEM_Register	2	7	3	R7: Регистр множителя

В канал **OEM_Register** типа INPUT с атрибутом **Канал=14** считывается старший байт регистра **R6&Status**. Схема настройки данных каналов приводится в таблице 2.

Таблица 2

Тип	Канал	MUX	CH	Канал	Регистр
I	OEM_Register	0	14	1	R6&Status (BASE+3)
I	OEM_Register	1	14	2	R6&Status (BASE+7)
I	OEM_Register	2	14	3	R6&Status (BASE+11)

В канал **OEM_Status** типа INPUT считывается регистр статуса канала. Схема настройки данных каналов приводится в таблице 3.

Таблица 3

Тип	Канал	MUX	CH	Канал	Регистр
I	OEM_Status	0	0	1	Статус канала (BASE+0) 1
I	OEM_Status	1	0	2	Статус канала (BASE+4) 2
I	OEM_Status	2	0	3	Статус канала (BASE+8) 3

Канал **OEM_Command** типа INPUT взаимодействует с командным буфером. Адрес буфера каждого канала задается атрибутом **Мультиплексор (MUX)** (0..2), а атрибут **Канал (CH)** используется для выбора соответствующей команды. При этом, если **CH=0**, значение канала воспринимается как прямая команда. При значениях **CH=1..4** значение канала воспринимается как данные предопределенной команды. Схема настройки данных каналов приведена в таблице 4.

Таблица 4

Тип	Канал	MUX	CH	Канал	Буфер команд	Команда
O	OEM_Command	0	0	1	WR0 (BASE+0)	Данные (b0...b7)
O	OEM_Command	1	0	2	WR4 (BASE+4)	Данные (b0...b7)
O	OEM_Command	2	0	3	WR8 (BASE+8)	Данные (b0...b7)
O	OEM_Command	0	1	1	WR0 (BASE+0)	Запуск/останов выбора команды (b0...b5)
O	OEM_Command	0	2	1	WR0 (BASE+0)	Команда выбора режима работы (b0...b5)
O	OEM_Command	0	3	1	WR0 (BASE+0)	Команда выбора регистра (b0...b5)
O	OEM_Command	0	4	1	WR0 (BASE+0)	Команда выбора выходного режима (b0...b5)
O	OEM_Command	1	1	2	WR4 (BASE+4)	Запуск/останов выбора команды (b0...b5)
O	OEM_Command	1	2	2	WR4 (BASE+4)	Команда выбора режима работы (b0...b5)
O	OEM_Command	1	3	2	WR4 (BASE+4)	Команда выбора регистра (b0...b5)
O	OEM_Command	1	4	2	WR4 (BASE+4)	Команда выбора выходного режи-

Тип	Канал	MUX	CH	Канал	Буфер команд	ко-ма	Команда
							ма (b0...b5)
O	OEM_Command	2	1	3	WR8 (BASE+8)		Запуск/останов выбора команды (b0...b5)
O	OEM_Command	2	2	3	WR8 (BASE+8)		Команда выбора режима работы (b0...b5)
O	OEM_Command	2	3	3	WR8 (BASE+8)		Команда выбора регистра (b0...b5)
O	OEM_Command	2	4	3	WR8 (BASE+8)		Команда выбора выходного режима (b0...b5)

Каналы **DI port** и **DO port** используются для считывания обычных портов дискретного ввода-вывода. Схема настройки каналов приводится в таблице 5.

Таблица 5

Тип	Канал	MUX	Базовый адрес (BASE)	CH	Данные (регистр)
I	DI_Port	0	BASE+C	0	DIGITAL INPUT 0-7 (BASE+12)
I	DI_Port	0	BASE+C	1	DIGITAL INPUT 8-15 (BASE+13)
O	DO_Port	0	BASE+C	0	DIGITAL OUPUT 0-7 (BASE+12)
O	DO_Port	0	BASE+C	1	DIGITAL OUPUT 8-15 (BASE+13)

Устройства FASTWEL

Плата FASTWEL AI16-5A

Для аналогового ввода используется канал **LA_AI16** типа INPUT (см. **Подтип 1**).

Коэффициент усиления для 4-х последовательных каналов – 0...3, 4...7 и т.д. – задается в атрибуте **Канал** того канала из этой группы, который имеет максимальный ID.

При этом коэффициенты усиления каналов группы занимают по 2 бита в 8-битовом числе:

Биты	7	6	5	4	3	2	1	0
Каналы	3		2		1		0	

Например, число 228= 0xE4 задаст коэффициенты усиления для каналов с номерами 0...3 с кодами соответственно 0, 1, 2, 3.

Каналы с **Канал (CH)= 0...7** работают в режиме дифференциального ввода, с **Канал (CH) = 32...47** – в режиме однопроводного ввода.

Для аналогового вывода (выходы DAC0 и DAC1) надо использовать каналы **AO(L,H)PCL** (см. **Подтип 1**) типа OUTPUT с адресом **Базовый_адрес+E** и атрибутом **Канал (CH)= 0**. При программном формировании выходного сигнала для DAC1 необходимо обеспечить установку 1 в 12-м разряде числа (например, в процедуре трансляции прибавить к выходному значению 4096).

Для дискретного вывода надо использовать канал **DO_port** (см. **Подтип 2**) с адресом **Базовый_адрес+3**.

Платы FASTWEL UNIOxx-5

В данном разделе приводится описание настроек каналов для работы с платами **UNIOxx-5** от **Fastwel** для прошивки G11.

Канал **AI_RWH** (аналоговый ввод) (см. **Подтип 1**):

- **Базовый адрес (BASE)** – базовый адрес платы (типа (адрес матрицы) + (базовый адрес по перемычкам));

адрес матрицы:

FGPA1=0xA000, FGPA2=0xA400, FGPA3=0xA800, FGPA4=0xAC00

- **Канал (CH)** – 0 или 1 (канал модуля GrayHill);
- **Мультиплексор (MUX)** – номер посадочного места модуля GrayHill (от 1 до 12) на плате TBI;

- **GAIN/REG** – не используется;
- **ALT/REG** – не используется.

Канал **DI_port** (дискретный ввод) (см. **Подтип 2**):

- **Базовый адрес (BASE)** – базовый адрес платы (типа (адрес матрицы) + (базовый адрес по переключкам));
адрес матрицы:

```
FGPA1=0xA000, FGPA2=0xA400, FGPA3=0xA800, FGPA4=0xAC00
```

- **Канал (CH)** – номер банка (1 - посадочные места с M0 по M3, 2 – места с M4 по M11 на плате TBI);
- **Мультиплексор (MUX)** – всегда 0;
- **Тип сигнала (Type)** – «...»;
- **Состояние (State)** – для мест M0-M3 – **+0<-FF**, для мест M4-M11 – **+1<-FF**.

Канал **AO_RWH** (аналоговый вывод) (см. **Подтип 1**):

- **Базовый адрес (BASE)** – базовый адрес платы (типа (адрес матрицы) + (базовый адрес по переключкам));
адрес матрицы:

```
FGPA1=0xA000, FGPA2=0xA400, FGPA3=0xA800, FGPA4=0xAC00
```

- **Канал (CH)** – 0 или 1 (канал модуля GrayHill);
- **Мультиплексор (MUX)** – номер посадочного места модуля GrayHill (от 1 до 12) на плате TBI;
- **GAIN/REG** – не используется;
- **ALT/REG** – не используется.

Канал **DO_port** (дискретный вывод) (см. **Подтип 2**):

- **Базовый адрес (BASE)** – базовый адрес платы (типа (адрес матрицы) + (базовый адрес по переключкам));
адрес матрицы:

```
FGPA1=0xA000, FGPA2=0xA400, FGPA3=0xA800, FGPA4=0xAC00
```

- **Канал (CH)** – номер банка (1 - посадочные места M0-M3, 2 - места M4-M11 на плате TBI);
- **Мультиплексор (MUX)** – всегда 64;
- **Тип сигнала (Type)** – «...»;
- **Состояние (State)** – для мест M0-M3 – **+0<-FF**, для мест M4-M11 – **+1<-FF**.

Драйвер платы RWH.EXE должен запускаться перед стартом Микро МРВ.

Модули GrayHill AI и DO на одной плате ТВІ не работают.
Необходимо разносить по разным матрицам FPGAx.

Контроллеры КРУИЗ

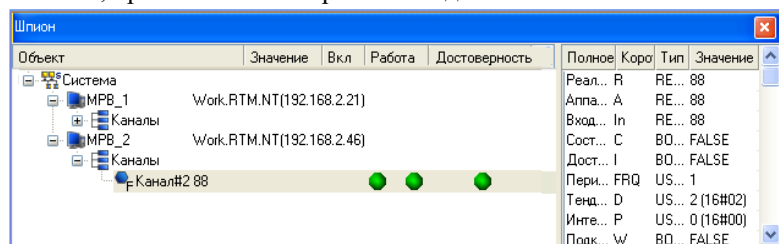
При резервировании процессорного модуля в контроллере КРУИЗ под Windows CE в УСО передается значение того канала АО, у которого вручную установлен бит 0 (вне зависимости от того, к какому узлу – WORK или TRACE – канал принадлежит). В отсутствие резервирования Микро МРВ устанавливает бит автоматически.

Глава 6

Отладка проекта

Для отладки проекта ИС снабжена следующими механизмами:

- автономная отладка шаблонов – см. разделы, посвященные разработке шаблонов;
- запуск выделенного узла (слоя **Технология**) под управлением одного из отладочных мониторов (профайлеров) из интегрированной среды по команде **Отладка** (см. **Меню ‘Файл’** и **главная панель инструментов ИС** и **Задание общих настроек ИС**) с протоколированием работы в текстовый файл;
- использование функции **шпион** – эта функция обеспечивает получение в режиме редактирования реальных данных с работающих узлов проекта. Для ее использования нужно выполнить команду **Шпион**, при этом в ИС открывается одноименное окно:



Функция **шпион** конфигурируется в диалоге **Настройки** (см. **Вкладка ‘Интегрированная среда разработки’**). Данная функция не работает при связи с узлом по RS.

- отладка в критических условиях (см. **@Debug** с **Параметр=3** в разделе **Группа СИСТЕМНЫЕ**).

Профайлеры

Для отладки узла (слоя **Технология**) его можно запустить (в том числе из ИС) под управлением одного из следующих отладочных мониторов:

- профайлера с поддержкой графических экранов (**rtc.exe**);
- профайлера без поддержки графических экранов (**rtmg32.exe**).

При конфигурировании ИС можно указать профайлер, который должен запускаться из ИС (см. **Задание общих настроек ИС**).

Профайлеры записывают протокол своей работы в файл **<имя файла rj>_<порядковый номер узла>.txt**, который сохраняется в папке узла. Степень детализации отладочной информации, выводимой в файл, может быть задана. Кроме того, все мониторы записывают информацию о своей работе в файл **tm6_log.txt** (см. **Файл tm6_log.txt**).

В протокол профайлера не выводятся счетчики байтов, транзакций и т.п.

Если ОС поддерживает переход на летнее время, в протокол профайлера записывается **Daylight enabled** (если переход включен) или **Daylight disabled** (если переход отключен).

В списке профайлера каналы отображаются по возрастанию ID, а аргументы канала – по возрастанию порядкового номера.

При выключении МРВ возможно наличие незавершенных потоков (соответствующее сообщение записывается в протокол профайлера). В этом случае МРВ будет остановлен с задержкой (таймаут – 1 минута).

Если при редактировании проекта в ИС одному из пользователей узла установлен флаг **Отработать**, в реальном времени права этого пользователя действуют после команды окончания сеанса.

Если для узла сконфигурирован дамп и автозапуск, после рестарта МРВ действуют права последнего пользователя даже в том случае, если новый сеанс открыт другим пользователем.

В произвольный атрибут канала могут быть записаны следующие предопределенные строки:

- **debug544...debug549** – в папке узла генерируется файл **t<544...549> <день> <часы> <минуты> <секунды>.html**, содержащий таблицу, соответствующую **SubNum=544...549** (см. **Универсальный механизм обмена с электросчетчиками, Канал CALL.Vector и Номер SubNum**);
- **debug=<число HEX без префикса «0x»>** – задание значения **@Debug**;
- **debugon=<битовая маска, HEX>** – установить биты **@Debug** по маске;
- **debugoff=<битовая маска, HEX>** – сбросить биты **@Debug** по маске.

Профайлер с поддержкой графических экранов

Монитор генерирует сообщения об ошибках загрузки узла (см. **Группа СИСТЕМНЫЕ / @Load**).

Графическая оболочка этого профайлера содержит меню, панель инструментов и рабочее поле, в котором отображаются графические экраны.



При нажатии ПК в области меню или панели инструментов отображается всплывающее меню, содержащее следующие команды:

- **Узел** – скрыть/показать панель инструментов.

В файле *.cnf предусмотрены ключи управления отображением окон МРВ (см. **Окна МРВ** в разделе **Задание параметров работы мониторов**).

Команды меню ФАЙЛ

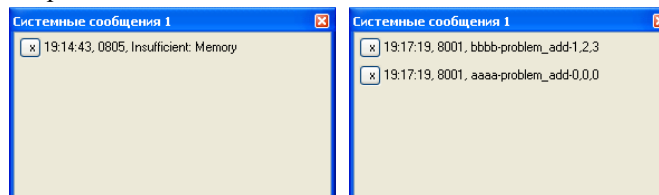
Меню **Файл** и панель инструментов содержат следующие команды:

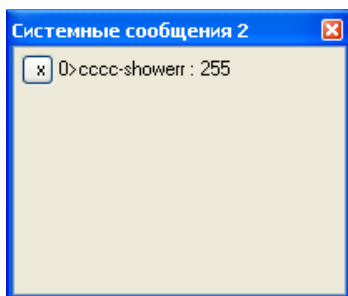
-  **Открыть**, CTRL+O – открыть файл узла;
-  **Запуск/Останов**, CTRL+R – запустить/остановить открытый узел;
- **Выход** – выйти из программы.

Команды меню ВИД

Меню **Вид** содержит следующие команды:

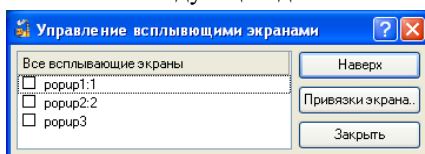
- **Полный экран** (CTRL+F) – переключение вида отображения графический экранов (в окне/полноэкранный). При переходе в полноэкранный режим по умолчанию видимо окно меню; чтобы закрыть/открыть это окно, нужно нажать CTRL+T;
- **Панель МРВ** – открыть панель МРВ;
- **Системные_сообщения_1** и **Системные_сообщения_2** – открыть окно системных сообщений 1 или 2:





В окно «СИСТЕМНЫЕ СООБЩЕНИЯ 2» выводятся, например, сообщения команды **ping**;

- **Окно событий** (CTRL+E) – по этой команде открывается окно событий (см. описание ниже);
- **Всплывающие экраны** (CTRL+K) – по этой команде на экране появляется следующий диалог:



Чтобы отобразить выделенный экран, нужно нажать кнопку **Наверх**, чтобы открыть диалог изменения привязок (см. ниже) – кнопку **Привязки экрана**.

Всплывающие экраны в диалоге и в меню **Вид** группируются по кодировке, образуя дерево. Каналы с кодировкой вида

$$S_1N_1S_2N_2\dots S_{M-1}N_{M-1}S_M$$

или

$$S_1N_1S_2N_2\dots S_{M-1}N_{M-1}S_MX$$

(S_i – строки, N_i – числа, X – произвольное число) отображаются в подгруппе

$$S_1N_1-S_2N_2-\dots-S_{M-1}N_{M-1}-S_M$$

Если у канала нет «партнеров» для группировки, он отображается на самом верхнем уровне иерархии.

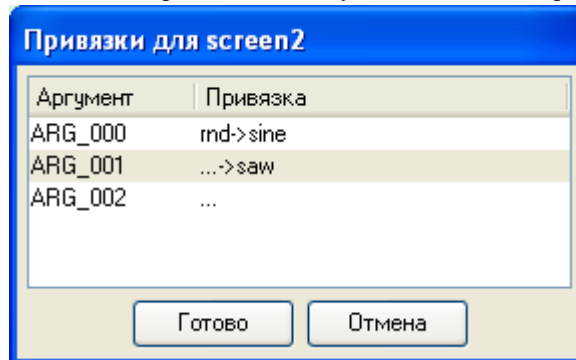
Команды меню ЭКРАНЫ

Меню и панель инструментов **Экраны** содержат команду перехода на стартовый экран (**Стартовый экран**, CTRL+Home), а также команды перехода на экраны, для которых установлен флаг **Показать в меню экранов** (см. **Задание параметров графического экрана**). Имя команды соответствует имени канала вызова экрана.

Команды меню **ДЕЙСТВИЯ**

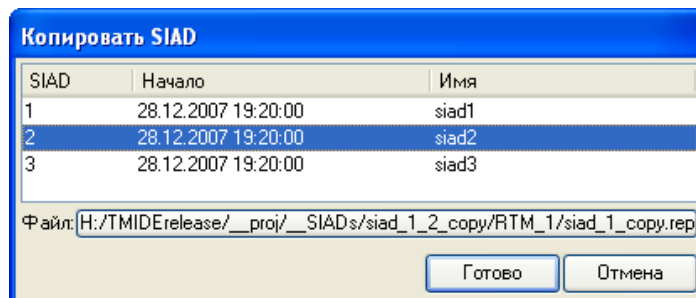
Меню **Действия** содержит следующие команды

- **Вход/Выход** (CTRL+SHIFT+ALT+P) – команды авторизации/окончания сеанса;
- **Ввести комментарий** (CTRL+W) – команда отправки сообщения в отчет тревог;
- **Привязки экрана** (CTRL+B) – команда открытия таблицы, содержащей список аргументов текущего экрана (столбец **Аргумент**) и их привязок (столбец **Привязка**). С помощью команды **Удалить привязку** контекстного меню можно удалить привязку выделенного аргумента. При двойном нажатии ЛК в поле **Привязка** открывается дерево каналов загружаемых объектов, с помощью которого привязку соответствующего аргумента можно изменить. Таблица отображает начальную и последнюю привязку:



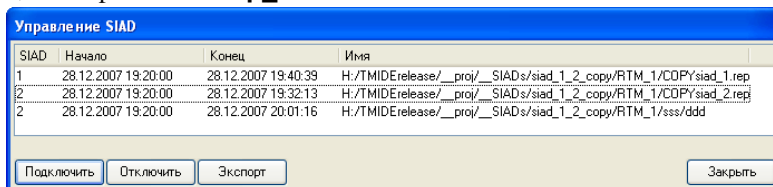
- **Копировать SIAD** (CTRL+U) – копирование SIAD; при выполнении команды открывается список архивов узла.

Для копирования архива нужно выделить его в списке, задать имя копии (задается в стандартном диалоге Windows, который открывается при нажатии кнопки **Файл**) и нажать кнопку **Готово**.



Информация о копиях архивов (номер SIAD, время начала и конца, файл) сохраняется в файле **backup_siad.txt** в папке узла.

- **Управление SIAD (CTRL+M)** – управление архивами; при выполнении команды открывается диалог, отображающий информацию из файла **backup_siad.txt**:



Диалог содержит следующие кнопки:

- **Подключить** – подключить выделенную копию;
- **Отключить** – отключить выделенную копию (подключить соответствующий основной архив);
- **Экспорт** – экспортировать выделенную копию в файл **<имя архива>.html**;
- **Заккрыть** – закрыть диалог.

В диалоге отображается информация об успешности/ошибке выполнения команд.

- **Скрыть все скрываемые слои (CTRL+D)** – закрыть все окна, отображающие скрываемые слои (см. ГЭ 'Объект').

Команды меню СПРАВКА

Меню **Справка** содержит следующие команды:

- **Содержание** – вызов контекстной справки TRACE MODE 6.
- **О программе** – вызов справки о МРВ.
- **Справка по проекту** – вызов из папки узла справки по проекту (разрабатывается пользователем). Файл справки ищется в следующем порядке: **<имя dbb-файла>.chm**, **<имя dbb-файла>_help.html**, **<имя dbb-файла>_help.txt**. Если ни одного из указанных файлов в папке узла нет, команда **Справка по проекту** недоступна.

Масштабирование графического экрана

Графический экран можно масштабировать в реальном времени с сохранением центра видимой области. При нажатии **CTRL+Shift+Z** экран переходит в режим масштаба Fit to Screen, т.е. пропорционально масштабируется до исчезновения полос прокрутки. При изменении границ окна экрана установленный масштаб не меняется. При повторном нажатии **CTRL+Shift+Z** происходит возврат в масштаб 1:1 с прокруткой до центрирования положения курсора на момент нажатия.

Дополнительные ключи команды запуска

Вид отображения графических экранов может быть задан с помощью следующих ключей команды запуска МРВ (см. также **Задание параметров работы мониторов**):

```
<МРВ> <узел> [/fullscreen [/hidemenu]] [/run]
[/windows] [/windowsxp]
```

- **/fullscreen** – в полноэкранном режиме; если ключ не задан – в окне;
- **/fullscreen /hidemenu** – в полноэкранном режиме без меню;
- **/run** – запуск узла. Если в узле существуют каналы класса **ПОЛЬЗОВАТЕЛЬ**, МРВ пытается запустить узел от лица пользователя с логином **default** и паролем **default**;
- **/windows** и **/windowsxp** – переопределяют стиль по умолчанию. Windows Vista и Windows 7: ключ **/windows** устраняет уменьшение ширины полосы прокрутки (ошибка Qt).

Ключи работают, если узел содержит хотя бы один экран.



Окно событий





В окне событий отображаются данные из файла ***.avb** (см. **Регистратор аварийных событий**), а также из SIAD.

В верхней части окна отображаются следующие параметры и элементы управления:

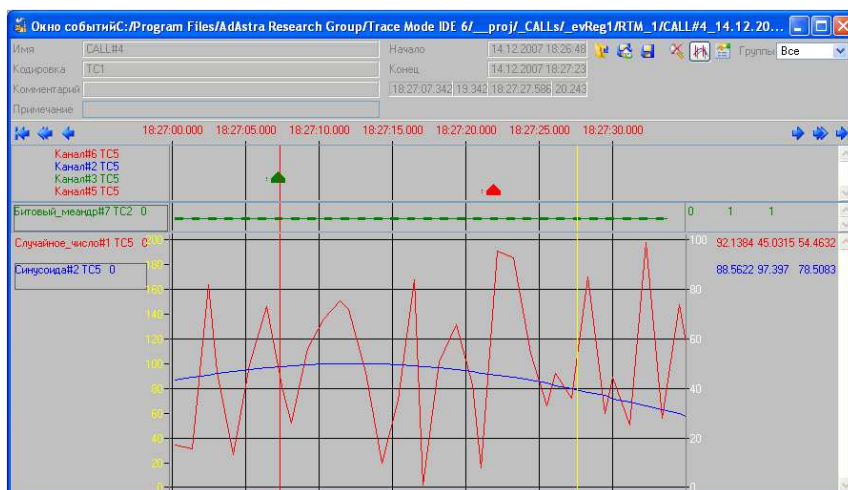
- **Имя** – имя регистратора (канала CALL.57);
- **Кодировка** – кодировка регистратора;
- **Комментарий** – комментарий регистратора;
- **Примечание** – в это поле можно ввести произвольный текст;
- **Начало** – T_{START} регистратора;
- **Конец** – T_{END} регистратора;
- Поля **Абсолютное время левого курсора**, **Относительное время левого курсора** (относительно T_{START}), **Абсолютное время правого курсора** и **Интервал** (промежуток времени между курсорами). Эти поля снабжены всплывающими подсказками;
- **Группы** – в этом списке выбирается сконфигурированная группа кривых.


В верхней части окна находится также панель инструментов, содержащая следующие команды, снабженные всплывающими подсказками:

-  **Загрузить файл** – открыть диалог выбора файла ***.avb**;
-  **Печать** – вывести на принтер по умолчанию;

-  **Сохранить** – сохранить конфигурацию окна событий в файл `crush_view.xml` (создается в папке MPB);
-  **Сбросить шкалы** – возврат к исходному масштабу;
-  **Два курсора** – переключатель количества курсоров тренда (один/два);
-  **Редактировать свойства** – открыть диалог конфигурирования окна событий.

В нижней части окна расположен тренд.



Верхняя часть тренда содержит ось времени, слева и справа от которой расположены кнопки  перехода по времени (соответственно **в начало**, **экран назад**, **деление назад**, **деление вперед**, **экран вперед** и **в конец**).

Под осью времени располагается **панель событий**, на которой с помощью меток отображается времена возникновения событий ($arg0_i = 1$), ниже располагаются **дискретная** и **аналоговая** панели для отображения значений сопутствующих каналов.

Слева от панелей отображается информация о каналах (настраиваемая), справа – значения кривых в точке установки курсоров.

Диалог конфигурирования содержит вкладки **Основные** (типовые свойства тренда), **События** (типовые свойства панели событий) и **Кривые и группы**.

Для создания/удаления новых групп/кривых используются команды контекстного меню, добавление кривой в группу выполняется методом drag-and-drop.

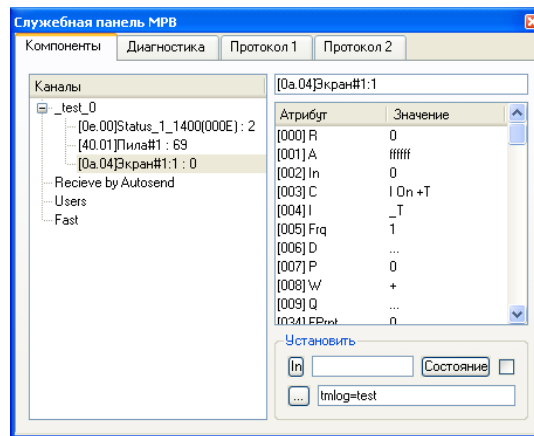
Для просмотра данных SIAD нужно выполнить следующие операции:

- создать группы;
- в разделе **Дополнительные кривые** создать кривые и распределить их по группам;
- в списке **Группы** выбрать одну из созданных групп для отображения соответствующих архивных данных. Диапазон прокрутки по времени – 2 года, команды **в конец** и **в начало** запрещены.

Панель MPB

Панель MPB открывается по одноименной команде меню **Вид** (см. **Профайлер с поддержкой графических экранов**). Панель содержит несколько вкладок и предназначена для задания/отображения различных параметров работы MPB.

Вкладка 'Компоненты'



В левой части вкладки отображаются объекты узла (см. **Вкладка 'Интегрированная среда разработки' / Раздел 'Сохранить для MPB'**), а правая содержит инструменты задания атрибутов канала, выделенного в левой части. Слева от имени канала отображаются его подтип и дополнение к подтипу (в формате HEX).

Для задания входного значения канала (атрибута 0, **In**) нужно ввести требуемое значение в поле справа от кнопки **In** и нажать ЛК на этой кнопке.

Для задания значения произвольного атрибута нужно выделить атрибут в списке, ввести требуемое значение в поле справа от кнопки, на которую выводится короткое имя выбранного атрибута, и нажать ЛК на этой кнопке.

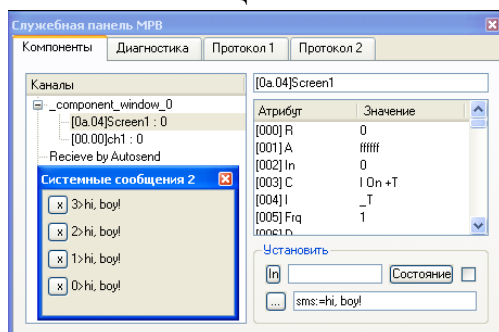
Для присвоения атрибуту текущего времени ОС можно ввести знак равенства «=». Для присвоения атрибуту текущего

времени ОС, смещенного на N секунд, можно ввести выражение «= N» или «= - N».

Для выключения канала нужно установить флаг в поле справа от кнопки **Состояние** и нажать ЛК на этой кнопке.

В разделе **Установить** допускаются следующие команды:

- **tmlog=clear** – удаляет все сообщения (аналог посылки 1 в бит 23 переменной 14.14 **@Debug** с **Параметр=2** – см. **Группа СИСТЕМНЫЕ**);
- **tmlog=test** – выдача тестовых сообщений по всем направлениям (см. 14.14 **@Debug** с **Параметр=2**);
- **ping** <IP-адрес>.
- **SMS:=<текст>** – отладка SMS-сообщений (ответ – в окне «СИСТЕМНЫЕ СООБЩЕНИЯ 2»):



- **SMS-><текст>** – для отсылки ОТ по SMS на телефоны, заданные ключом **SMS_ALARME_PHONE_LIST** (см. **Задание параметров работы мониторов**);
- **ADDPHONE:<телефон>** – добавление телефона в список, заданный ключом **SMS_CHECK_PHONE_LIST** (см. **Редактор параметров COM-порта**);
- **DELPHONE:<телефон>** – удаление телефона из списка **SMS_CHECK_PHONE_LIST**.

На вкладке могут быть заданы те же ключи, что и в файле CNF (см. **Задание параметров работы мониторов**), при этом в любом регистре.

Ручное конфигурирование канала

С помощью следующих команд на вкладке **Компоненты** возможно ручное конфигурирование каналов (для отладки проекта):

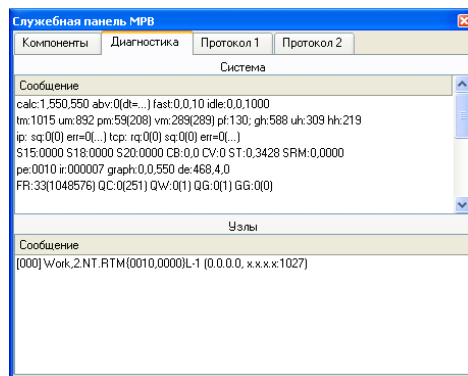
- для числового аргумента канала CALL.TVC, не привязанного к каналу:
 - **@fa=1/0** – установка/снятие признака недостоверности зна-

- чения аргумента;
- **@wc=1/0** – установка/снятие признака «отключен» для аргумента;
- **@sc=1/0** – установка/снятие признака «выключен» для аргумента;
- **@p=<n>** – задание номера интервала для значения аргумента;
- **@=<m>** – одновременное задание предыдущих четырех характеристик;
- **@=255** – значение аргумента не выводится в таблицу;
- для временного аргумента канала CALL.TVC, не привязанного к каналу:
 - **@=<k>** – значение миллисекунд;
- для аргумента канала класса CALL (кроме CALL.TVC):
 - **@=<k>** – номер атрибута, привязанного к аргументу;
- для атрибута 9, **Q** канала CALL.TVC:
 - ввод числа **N**:
 - если **arg0<arg1**, заданный этими аргументами диапазон выборки разбивается на **N** равных интервалов;
 - если **arg0=arg1**, **T_TO** определяется текущим временем, выровненным на минуту, **T_FROM = T_TO – (N+1)**;
 - **@size=<n>** – число точек;
 - **@curve= <m>** – число кривых.

Вкладка 'Диагностика'

Раздел СИСТЕМА

Раздел **Система** этой вкладки отображает следующие системные параметры TRACE MODE 6:



- информация о временах циклов CALC, IDLE и FAST:
 - **clc** – время пересчета;
 - **loop** – получившийся цикл;
 - **want** – желаемый цикл;
 - **count** – счетчик превышений времени цикла;
 - **dt** – время последнего превышения;
- информация о памяти:
 - **tm** – всего памяти;
 - **um** – занято памяти;
 - **pm** – занято памяти монитором, в скобках – максимальное значение;
 - **vm** – занято виртуальной памяти монитором, в скобках – максимальное значение;
 - **pf** – число страниц, с которыми произошел свопинг;
 - **gh** – счетчик объектов GDI;
 - **uh** – счетчик объектов USER;
 - **hh** – счетчик дескрипторов;
- информация об очередях отправки по сети:
 - **ip:sq** – средняя (максимальная) очередь на отправку по UDP (см. **Группа ДИАГНОСТИКА** / **@q_IP_Send_Q**);
 - **ip:err** – код ошибки (время ошибки);
 - **tcp:rq** – средняя (максимальная) очередь на отправку по TCP при внешних запросах к данному узлу;
 - **tcp:sq** – средняя (максимальная) очередь на отправку по TCP при собственных запросах данного узла;
 - **tcp:err** – код ошибки (время ошибки);
- информация о синхронизации:
 - **S15**, **S18** и **S20** – значения **@Net_DDE**, **@Logging** и **@Input_Output** (см. **Группа СИСТЕМНЫЕ**);
 - **CB** – число блоков, реально затрачиваемых на синхронизацию;
 - **SB** – баланс (при сбоях синхронизации растет, значения 0,1 и 2 считаются нормой);
 - **ST** – приблизительное время цикла синхронизации (мс) и заданный период синхронизации по биту 12 **@Input_Output** (мс);
 - **SM** – маска синхронизации (см. описание **@RTM_Parameter** с **Параметр=138**, **SyncMask** в разделе **Группа СИСТЕМНЫЕ**);
- информация о синхронизации (см. **Синхронизация резервов**):

- **CB:%d,%d** – число блоков, реально затрачиваемых на синхронизацию, и число потерянных блоков;
- **SB:%d** – баланс (при сбоях синхронизации растет, значения 0,1 и 2 считаются нормой);
- **ST:%d,%d** – приблизительное время цикла синхронизации (мс) и заданный период синхронизации;
- **SM:%x** – last request data for time;
- информация о проблемах:
 - **pe** – биты проблем;
 - **ir** – биты проблем с ресурсами;
- информация об отчете тревог (OT):
 - **FR** – номер последней записи (максимальное число строк);
 - **QC** – число строк в кэше;
 - **QW** – текущее (максимальное) число строк для записи;
 - **QG** – текущее (максимальное) число строк для отсылки в графику;
 - **GG** – текущее (максимальное) число строк для отсылки в СМС и звук;
 - **NEM** – число отосланных/принятых считываний и число отосланных/принятых сообщений (сброс при подсоединении консоли);
- информация об архивных выборках:
 - **cR** – число выборок из архива;
 - **cnR** – число сетевых выборок из архива;
 - **tR** – среднее время выборки из архива;
 - **sR** – среднее время выборки срезов из архива;
 - **wb** – число мегабайт на запись (операции с диском);
 - **rb** – число мегабайт на чтение (операции с диском);
- три строки для архивов, каждая строка содержит следующие параметры:
 - **E** – ошибка архива (см. **@e_SIAD**);
 - **QS** – очередь на выборку (см. **@q_SIAD_Q**);
 - **IC** – процентная очередь на запись;
 - **LN** – процент заполнения;
 - **KU** – коэффициент использования;
 - **LP** – потери;
 - **DT** – интервал удаления записей архива для записи новых (при зацикливании);
- три строки для первых копий архивов, параметры – ошибка, очередь на выборку, время начала и время конца;

- три строки для вторых подключенных копий архивов, параметры те же.

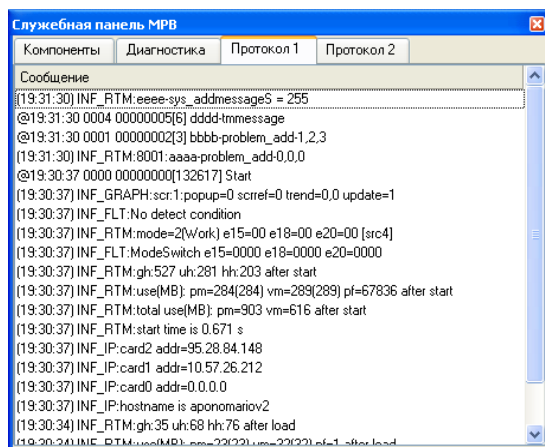
Раздел УЗЛЫ

Раздел **Узлы** вкладки отображает параметры узлов.

Каждый узел описывается в отдельной строке, число строк равно **MAXNODE** – см. **Файл CNF** в разделе **Задание параметров работы мониторов**).

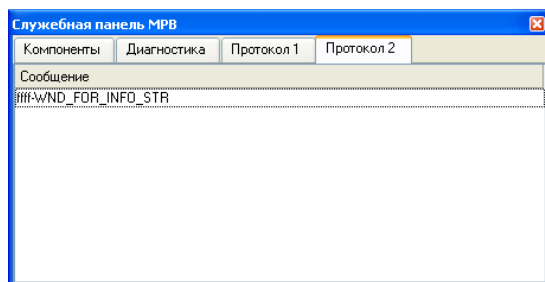
Формат строки (см. **Группа СИСТЕМНЫЕ / @Status**): статус (**STATUS & 0xFF**), ОС (**STATUS & 0xF00**), продукт TRACE MODE (**STATUS & 0xF000**), время, IP-адрес и имя компьютера.

Вкладка 'Протокол 1'



Эта вкладка отображает протокол MPB **<имя файла prj>_<порядковый номер узла>.txt**, а также сообщения протокола **tm6_log.txt** (см. **Файл tm6_log.txt**), начинающиеся со знака **@<**».

Вкладка 'Протокол 2'



Эта вкладка отображает сообщения протокола MPV **tm6_log.txt** (см. **Файл tm6_log.txt**), за исключением тех сообщений, которые отображаются на вкладке **Протокол 1**.

Вкладка 'Пользователи'

Эта вкладка содержит инструменты для работы с каналами класса **Пользователь**:

Доступ к управлению пользователями

Логин

Пароль

Войти

Действия

Применить Добавить Удалить

Имя

Логин

Пароль

Телефон

Email

MPV

Выход Изменение Доступ к экранам

Останов

Запуск Квитирование Доступ к действиям

Права (дополнительно)

Редактировать Создать Удалить

Компоненты

Редактировать Создать Удалить

Пользователи

Редактировать Добавить Удалить

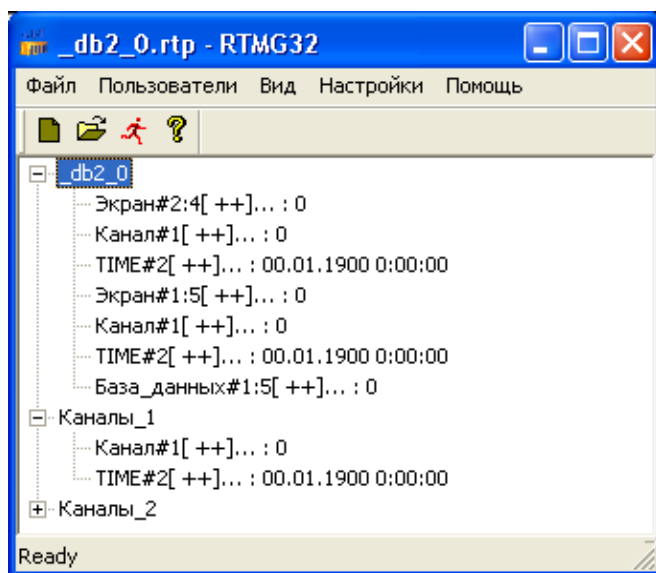
На вкладке можно редактировать права пользователей, созданных при разработке проекта, а также удалять пользователей и создавать новых. Верхний раздел диалога предназначен для авторизации, остальные разделы содержат те же флаги, что и редактор канала класса **Пользователь** (см. **Канал класса ПОЛЬЗОВАТЕЛЬ**).

Профайлер без поддержки графических экранов




Этот профайлер обеспечивает отображение графических панелей (см. **Графические панели**).

Монитор генерирует сообщения об ошибках загрузки узла (см. **Группа СИСТЕМНЫЕ / @Load**).

Графическая оболочка профайлера содержит меню, панель инструментов и рабочее поле, в котором отображаются объекты базы каналов узла (группы каналов, каналы, а также атрибуты каналов):



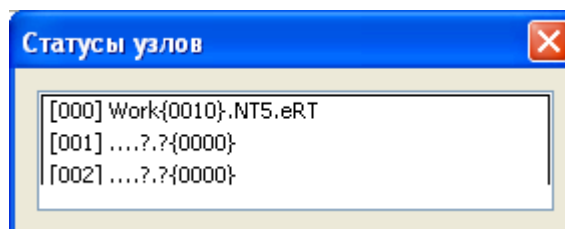
Для выделения канала жирным шрифтом (или снятия выделения) нужно дважды нажать ПК на канале (для снятия выделения всех каналов нужно выполнить команду **Снять маркировку** из меню **Настройки**).

Меню **Файл** и панель инструментов содержат команды открытия (, CTRL+O), перезагрузки () и запуска/останова (, CTRL+R) узла, а также команду выхода из программы (ALT+X).

Меню **Пользователи** содержит команды авторизации/окончания сеанса (**Вход**, CTRL+L / **Выход**, ALT+L), а также команду **Управление** (см. описание вкладки **Пользователи** в разделе **Панель MPB**).

Меню **Вид** содержит флаги управления видимостью строки статуса и па-

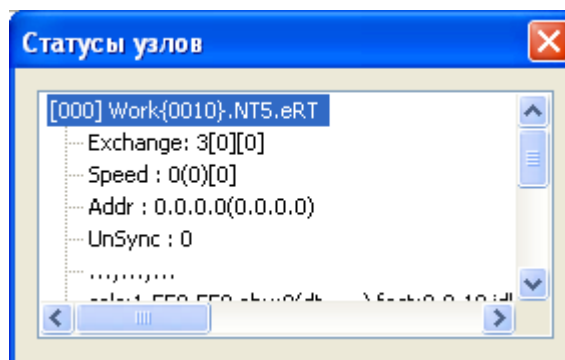
нели инструментов, а также команду открытия окна статусов узлов:



Это окно отображает текущее состояние узлов проекта (каждый узел описывается в отдельной строке, число строк равно **MAXNODE** – см. **Файл CNF** в разделе **Задание параметров работы мониторов**).

Формат строки (см. **Имена и идентификаторы объектов структуры**, а также **Группа СИСТЕМНЫЕ / @Status**): **ordinal**, статус (**STATUS & 0xFF**), ОС (**STATUS & 0xF00**) и продукт TRACE MODE (**STATUS & 0xF000**).

Чтобы показать/скрыть дополнительные параметры узла, нужно дважды нажать ЛК в строке этого узла:



Формат строки **Exchange** (см. **@Status**):

```
<Last_Exchange> [<CC2>] [<NodeState>] [<LogDiag>]
```

Для собственного узла второй параметр показывает количество отброшенных блоков (адресованных другому узлу).

Формат строки **Speed** (см. **@Status**):

```
<SYS_check> (<AVR_Time>) [<AVR_RS_Time>]
```

В строке **Addr** индицируются IP-адреса адаптеров приема/передачи.

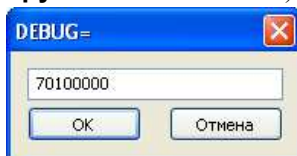
Формат строки **UnSync** (см. **@Status**):

```
<UnSync>
```

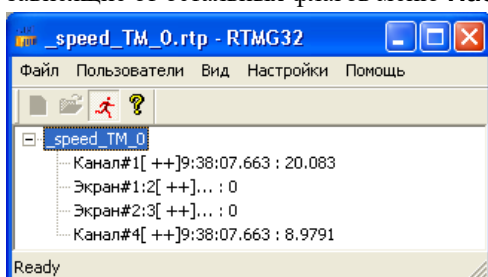
Остальные параметры узла аналогичны параметрам, которые отображаются в **rtc.exe** (см. **Панель MPB**).

Меню **Настройки** содержит следующие команды:

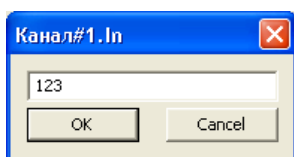
- **Отладка** – при выполнении этой команды на экране появляется диалог задания значения системной переменной **@Debug** (см. **Группа СИСТЕМНЫЕ**):



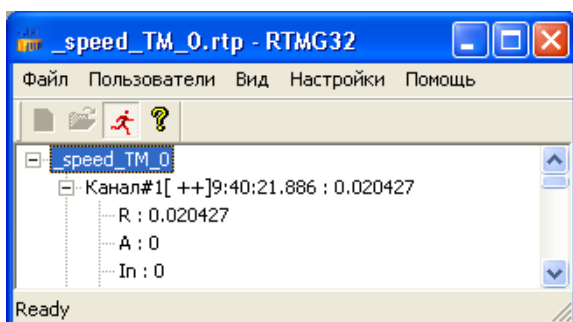
- **Сокращенно** – если этот флаг установлен (значение по умолчанию), профайлер не отображает атрибуты каналов (в строке канала отображается его реальное значение и некоторые другие атрибуты, зависящие от остальных флагов меню **Настройки**):



В этом режиме при двойном нажатии ЛК на канале на экране появляется окно задания его входного значения (атрибута 0, **In**):

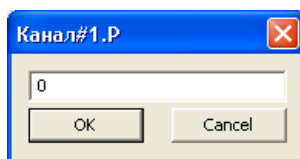


Если флаг не установлен, профайлер отображает атрибуты каналов (см. **Атрибуты каналов, отображаемые профайлером**):



В этом режиме при двойном нажатии на атрибуте на экране появляется окно задания его значения (для задания значения времени

могут быть использованы те же выражения, что и в **rts.exe** – см. **Панель MPB**):



- **Флаги** – если этот флаг установлен (значение по умолчанию), в строке канала отображается состояние флагов (3, **C**) **Состояние** и (8, **W**) **Подключение**:

Канал#1[++] : 123

При установке каналу признака аппаратной недостоверности к указанным флагам добавляется знак «?»:

Канал#1[?+] : 123

- **Время изменения** – если этот флаг установлен (значение по умолчанию), в строке канала отображается время изменения его значения:

ch1[++]17:00:46.000 : 5

- **Дополнительно** – при установке этого флага (по умолчанию он не установлен) в строке канала отображается: слева от имени – идентификатор канала, справа – подтип, дополнение к подтипу (см. **Подтипы каналов**) и тип (**i/o**):

[001]ch1[0.0,i] : 1

Если флаг установлен, слева от короткого имени атрибута отображается его номер:

...[045] T : 30.11.2006 16:43:39

- **Скорость обновления** – выбор времени цикла отображения профайлера (по умолчанию – 1с);
- **Экспорт СПАД** – экспорт данных архива.

В текстовый файл экспортируются только сообщения об изменении атрибута 0, **R** канала.

Для экспорта данных SIAD нужно выполнить следующие действия:

- открыть в профайлере файл узла ***.rtp**;
- с помощью команды **Экспорт СПАД** открыть диалог, показанный ниже.

В диалоге нужно задать временной интервал для выборки (T_FROM = **Начало**, T_TO = **Конец**) и в списке **Имя** выбрать файл архива.

Параметры выборки

Начало: 02.06.2006 8:48:00

Конец: 02.06.2006 9:11:05

Имя: arch

Результат:

Только инф. Все Не стирать Один файл

Разделитель: SpaceBar

В списке **Имя** отображаются имена, заданные для файлов архивов при конфигурировании узла в ИС. Чтобы открыть стандартный диалог ОС выбора файла в файловой системе, нужно нажать кнопку **Выбрать** справа от списка **Имя**.

- с помощью опций диалога задать параметры экспорта:
 - **Один файл** – если этот флаг установлен, в папке узла создается файл **<имя архива>.txt**, который содержит все извлеченные архивные данные. Если флаг не установлен, для каждого архивируемого канала в папке узла создается файл **<имя канала>.txt**, который содержит извлеченные архивные данные по этому каналу;
 - **Только информация** – если этот флаг установлен, при экспорте в папке узла создается текстовый файл ***.inf**, содержащий информацию о реально обнаруженном временном интервале записей на основании заданного диапазона (T_FROM, T_TO). Кроме того, этот флаг запрещает автоматическое закрытие диалога после экспорта; в этом случае в диалоге отображается реально обнаруженный интервал записей. Флаг имеет более высокий приоритет по отношению к флагу **Все**;
 - **Все** – если этот флаг установлен, в папке узла создается один или несколько файлов ***.txt** (в зависимости от флага **Один файл**) и файл ***.inf**, содержащий информацию о реально обнаруженном временном интервале записей, об общем числе записей в этом интервале, а также о количестве записей по каждому каналу (имени канала предшествует его ID);
 - **Не стирать** – если этот флаг установлен, экспортные данные добавляются в файлы ***.txt**, которые в противном случае перезаписываются;
 - **Разделитель** – выбор разделителя объектов в строке (в создаваемых файлах);
 - **Результат** – задание/выбор имени файла ***.txt** (при экспорте в один файл) и файла ***.inf**. Если данное поле оставить пустым,

файлы будут иметь имя архива;

- нажать кнопку **ОК** диалога;
- **Снять маркировку** – снять выделение каналов.
- **Конвертировать СПАД** – конвертировать архив для релиза 6.06 (см. также **Архивы SIAD**);
- **Конвертировать дампы** – конвертировать дампы для релиза 6.06.

В команде запуска монитора (см. **Задание параметров работы мониторов**) могут быть дополнительно заданы следующие ключи:

- **/RUN** – запуск узла;
- **/HIDE** – запуск без меню и панели инструментов.

Файл `tm6_log.txt`

Этот файл создается в папке узла; он содержит сообщения системы встроенного контроля МРВ.

Большинство видов сообщений записываются в файл вне зависимости от флага **Разрешение `tm6_log`** (см. **Задание параметров узла**). Если запись некоторого вида сообщения зависит от флага, это указано в описании этого вида сообщения (см. ниже).

Сообщение записывается в файл в виде отдельной строки. Общий формат сообщения:

T N1 N2 [N3] Text

где

- **T** – время сообщения (без даты);
- **N1** – номер потока, который сгенерировал сообщение;
- **N2** – 8-значное число;
- **N3** – число;
- **Text** – необязательный текст.

Виды сообщений в файле `tm6_log.txt`

T 0000 00000000 [0] <дата>

При изменении даты, а также при старте МРВ.

T 0000 00000000 [0] Start

Старт МРВ.

T 0000 00000000 [0] Stop

Останов МРВ.

T 0000 00000032 [0] <имя канала>

Изменилось значение канала OUTPUT с установленным флагом **Отладка** (49, **DBG**), но предыдущее значение канала не передано.

T 0000 00000032 [N3] <имя канала>

Неустраняемая ошибка в канале (в канале устанавливается флаг аппаратной недостоверности и он выключается). Расшифровка значений **N3**:

- 1 – нет аргументов;
- 2 – неправильное поле **связь** (нет привязки);
- 3 – число аргументов превышено;

4 – подсистема не инициализирована (например, не создана виртуальная машина);

5 – обращение к несуществующему архиву;

6 – фатальная ошибка MPB на канале, вызывающем функции DLL;

7 – фатальная ошибка MPB при пересчете канала.

Ошибка канала может быть не обнаружена, если для канала проведена процедура ремэппинга (в MPB с ограничением числа каналов).

T 0000 00000032[8] RTM dump write

Сбой при записи дампа.

T 0000 00000032[8] RTM dump read

Сбой при чтении дампа.

T 0000 00000001[N3] Calc loop is big

Превышено время цикла CALC. **N3** – счетчик числа превышений.

T 0000 00000001[N3] Calc loop is big (T)

Время цикла CALC превышает в течение **N3** с.

T 0016 00000032[9] <имя канала>

Фатальная ошибка MPB на канале вызова графического экрана.

T 0016 00000032[8] <имя канала>

Ошибка записи буфера тренда в файл.

T 0017 <номер архива>[<код ошибки>]

Ошибка архива.

T 0000 00000000[4..5]

Переход в состояние WORK (4) или TRACE (5). Этот вид сообщения блокируется флагом **Разрешение tm6_log**.

T 0029 N2[N3] Overstress: CPU

Перегрузка ЦП. **N2** – время цикла; **N3** – число превышений времени цикла.

T 0029 N2[N3] Insufficient: Memory

Не хватает памяти. **N2** – время цикла; **N3** – число превышений времени цикла.

T 0029 N2[N3] pf:N4 fm:N5 vm:N6

Не хватает памяти. **N2** – счетчик объектов GDI, **N3** – счетчик

объектов USER, **N4** – число страниц, с которыми произошел свопинг; **N5** – занятый объем ОЗУ, **N6** – занятый объем виртуальной памяти;

T 0029 N2[0] Insufficient: GDI objects

Не хватает объектов GDI. **N2** – счетчик объектов GDI.

T 0029 N2[0] Insufficient: USER objects

Не хватает объектов USER. **N2** – счетчик объектов USER.

T 0017 N2[0] Recover from incorrect stop

Структура и данные архива восстановлены полностью. **N2** – номер архива.

T 0017 N2[N3] Recover from incorrect stop Data lost

Структура архива восстановлена, однако данные могут быть потеряны. **N2** – номер архива, **N3** – код ошибки.

T 0017 N2[0] Wrong Recover

Архив не запускается. **N2** – номер архива.

T 2000 0000 Modem not ready

При попытке отправки SMS-сообщения не детектируется модем.

T 1829 <число мс> <имя канала>

MPV не может обработать канал (обычно – нарисовать панель/экран).

T 10049 <номер узла> NET

Указанный IP-адрес отсутствует в сети (соответствие «номер узла – IP-адрес» задано в **addr.ind**).

1640 – ошибки панелей.

T 1640 101

Ошибка загрузки bitmap (нет файла).

T 1640 102

Ошибка загрузки bitmap (неверный формат файла).

T 1640 116

Ошибка загрузки bitmap (недостаточно памяти).

T 1640 266

Ошибка загрузки bitmap (недостаточно памяти).

Коды диагностируемых ошибок

DEC-коды ошибок, диагностируемых MPB:

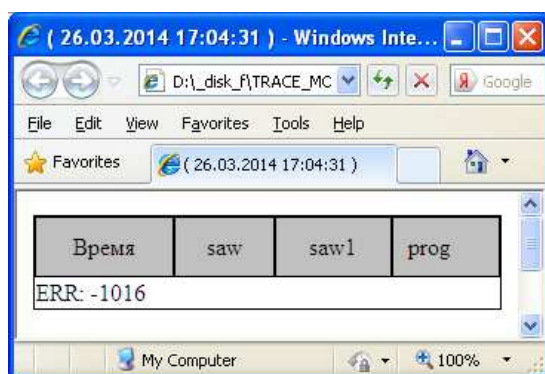
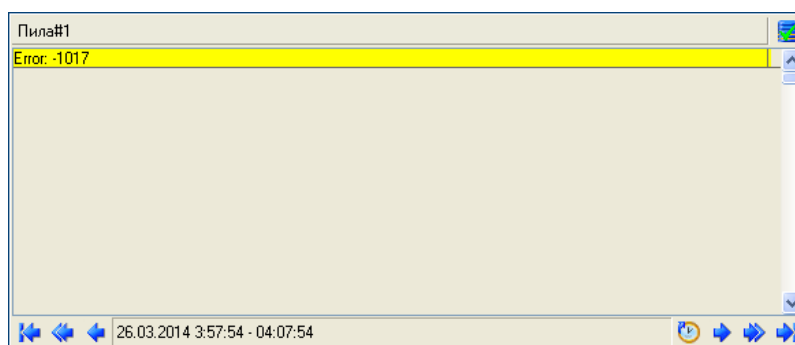
- 1, ERR_RT_FILE – файл не открыт;
 - В CALL.Document(Report) – ошибка открытия/создания документа;
- 2, ERR_RT_SEEK – ошибка позиционирования (в архиве, списке и т.п.);
- 3, ERR_RT_WRITE – ошибка записи;
 - В CALL.Document(Report) – ошибка записи документа;
- 4, ERR_RT_READ – ошибка чтения;
- 5, ERR_RT_MEM – ошибка работы с памятью;
 - В CALL.Document(Report) – не хватает памяти для генерации документа;
- 6, ERR_RT_LIST – ошибка списка;
- 7, ERR_RT_FORMAT – ошибка формата;
 - В CALL.Document(Report) – невозможно вставить картинку или документ в документ;
- 8, ERR_RT_COUNT – неверное число символов;
- 9, ERR_RT_TIMEOUT – превышен таймаут;
- 10, ERR_RT_RESP – ошибка ответа;
- 11, ERR_RT_FUNC – ошибка внутренней функции. Такая ошибка при обмене по GSM означает переход модема в состояние OFF; при переходе в режим READY ошибка сбрасывается;
- 12, ERR_RT_NOTFOUND – объект не найден;
- 13, ERR_RT_CODE –
- 14, ERR_RT_FSC – ошибка контрольной суммы;
- 15, ERR_RT_WAIT – неправильный аргумент;
- 16, ERR_RT_PLACE – ошибка размещения;
- 17, ERR_RT_CONNECT – ошибка соединения;
 - В CALL.Document(Report) – ошибка печати или публикации документа;
- 18, ERR_RT_HANGUP – ошибка модема;
- 20 – ошибка дополнительной контрольной суммы;
- 31, ERR_RT_REPEAT – ошибка повторения;
- 32, ERR_RT_EXECUTE – ошибка выполнения (если возникает при обмене по GSM, модем требует аппаратного перезапуска, при этом восстановление работы после перезапуска не гарантируется);

- В CALL.Document(Report) – общая ошибка генерации документа;
- 33, ERR_RT_DEADLOCK – объект заблокирован;
- 34, ERR_RT_FILELIMIT – превышен размер файла;
- 35, ERR_RT_OUTOFRANGE – значение вне допустимого диапазона;
- 36, ERR_RT_FILEERR – ошибка работы с файлом;
- 37, ERR_RT_LOADDLL – ошибка загрузки DLL;
- 39 – не хватает ресурсов;
- 40 – устройство занято.

Диагностируемые ошибки загрузки/перезагрузки узла указаны в описании переменной **@Load** в разделе **Группа СИСТЕМНЫЕ**.

Ошибки выборки данных по запросу экрана/документа

Ниже перечислены ошибки (-1001 .. -1031), которые МРВ может вернуть в ответ на запрос данных, полученный от документа/экрана. При получении ошибки от МРВ, документ/экран прекращает запрос данных, а соответствующие объекты документа/экрана отображают ошибки:



- -1001 – нет данных;
- -1002 – ошибочное время выборки в запросе;
- -1003 – запрос к удаленному узлу содержит некорректные параметры;
- -1004 – запрашиваемый канал удаленного узла не содержится в архиве;
- -1005 – удаленному узлу не хватает памяти для исполнения запроса;
- -1006 – удаленный узел вернул ошибку превышения таймаута при сборе данных;
- -1007 – попытка получить данные по запросу, который не активирован, отсутствует и т.п.;
- -1008 – ошибка отправки запроса удаленному узлу;
- -1009 – удаленный узел не найден;
- -1010 – не хватает памяти для выполнения запроса;
- -1011 – неправильные параметры на запрашивающем узле (например, число запрашиваемых данных – 0);
- -1012 – таймаут получения данных от удаленного узла;
- -1013 – данный канал не поддерживается для данного запроса;
- -1014 – разорвано соединение с удаленным узлом;
- -1016 – общая ошибка задания времени;
- -1017 – ошибка времени T_FROM;
- -1018 – некорректный ID канала (нет такого канала в МРВ);
- -1019 – некорректный индекс канала CALL;
- -1020 – некорректный индекс аргумента;
- -1021 – данные запрашиваются у константы (т.е. у аргумента нет привязки);
- -1022 – данные запрашиваются из архива, который не открыт;
- -1023 – запрашиваются данные из сбойного архива;
- -1024 – запрашиваемое время среза (или T_TO в случае запроса тренда) не содержится ни в одном архиве;
- -1028 – данный канал не поддерживает срез;
- -1029 – данный канал не поддерживает тренд (например, хранит срез);
- -1030 – не найдено ни одного свободного запроса данных (т.е. МРВ не может запросить данные);
- -1031 – операция не поддерживается в данной версии.

Отладочные сообщения

В данном разделе приведены сообщения, которые генерируются при некоторых значениях системных переменных TRACE MODE\$ 6.

Сообщения при DBG_INFO_NET=ON

Если в файле *.cnf задан ключ **DBG_INFO_NET=ON** (или установлен бит 2 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться сообщения, приведенные ниже.

Сообщения, отображаемые в окне системных сообщений MPB (см. **Про-файлер с поддержкой графических экранов**):

node_%d stopped(it%d) [byT]

узел N остановлен (SH_ERR_NODE_UNLOAD)

node_%d unload(it%d) [byT]

узел N выгружен (SH_ERR_NODE_UNLOAD)

node_%d not found(it%d)

узел N не найден (SH_ERR_NODE_UNLOAD)

node_%d work,%d

узел N работает (SH_ERR_NODE_PRESENT)

Reserve node not found

резервный узел не найден (SH_ERR_NODE_NO_RESERVE)

Reserve node OK

резервный узел найден (SH_ERR_NODE_FIND_RESERVE)

DONT SET CBUF SIZE

не задан размер буфера TCP (SH_ERR_TCP_NOT_SET_BUFSIZE)

DONT REC BROADCAST

данный узел не принимает широковещательные посылки (SH_ERR_IP_IM_NO_REC_BROADCAST)

SET DONT SEND BROADCAST

задание для данного узла невозможности отправки широковещательных посылок (SH_ERR_IP_IM_NO_SEND_BROADCAST)

USE DONT SEND BROADCAST

использование для данного узла невозможности отправки широковещательных посылок (SH_ERR_IP_IM_NO_SEND_BROADCAST)

node %d send: dont_rec_broadcast

узел N сообщает, что не принимает широковещательные послылки (SH_ERR_IP_NO_SEND_BROADCAST)

net udp %d: in error

ошибка сетевого адаптера N (SH_ERR_UDP_ERROR_CARD)

net udp %d: ok

сетевой адаптер N в норме (SH_ERR_UDP_OK_CARD)

UDP send by card%d %s

номер и IP-адрес сетевого адаптера для отправки по UDP (SH_ERR_IP_CHANGE_SEND_CARD)

Другие сообщения:

ide have %s:%d card=%d

IP-адрес и номер адаптера IDE

node %d have %s:%d card=%d for udp r%d

IP-адрес и номер адаптера узла N для UDP

```
"nib=%d Length of pocket=%d IPSIZE=%d,%d
queue=%d",rtm.ip_p.count_of_ind,rtm.ip_p.POCKET_LEN
,rtm.ip_p.lim.ip_inbuf_size,rtm.ip_p.lim.ip_outbuf_
size,ip_p.ip_send_list->SetMaxCount(-1);
```

nib, длина пакета, размер IP-блока, размер очереди, число буферов для индивидуальных рассылок, длина пакета, размер приемного буфера, размер буфера на передачу, размер списка отправки

bridge_0 is node = %d

узел N является первым мостом

bridge_1 is node = %d

узел N является вторым мостом

hostname is %s

имя хоста

default ip addr not found

IP-адрес по умолчанию для UDP-обмена не найден

default tcp addr not found

IP-адрес по умолчанию для TCP-обмена не найден

p%d bind addr %s not found

Для отмены генерации сообщений используется ключ **DBG_INFO_NET=OFF**.

Сообщения при **DBG_INFO_NODE=ON**

Если в файле *.cnf задан ключ **DBG_INFO_NODE=ON** (или установлен бит 9 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие сообщения:

MTA0: TCP=WANT node%d = ok/bad

вставка намерения узла N подключиться (от архива)

MT00: TCP=WANT node%d = ok/bad

вставка намерения узла N подключиться (от канала)

MTL0: TCP=WANT node%d = ok/bad

вставка намерения узла N подключиться к регистратору

MT00: TCP=OFF,init node%d

узел N не найден при старте, обмен по TCP выключается

MT00: TCP=CLOSE node%d

намерение разорвать соединение с узлом N вручную (или по времени при отсутствии обмена)

MT00: TCP=ON,OK node%d

соединение с узлом N доступно, обмен по TCP работает

MT00: TCP=OFF,BAD node%d

обмен с неопределенным узлом, установка FA=1, обмен по TCP выключается

MI00: IP=OK node%d

восстановление обмена по UDP

MI00: IP=BAD node%d

при попытке обмена по UDP в канале узла N устанавливается FA=1

MI00: IP=OFF node%d

обмен с узлом N отключен

MI00: IP=ON node%d

обмен с узлом N разрешен

MC00: node_%d=Close

вставка намерения узла N разорвать соединение

MCL0: node_%d=Close

вставка намерения узла N разорвать соединение с регистратором

MTL0: Fail / LFail

ни один регистратор не обнаружен

MTL0: Present / LPresent

обнаружен некоторый регистратор

MTL0: save=%d

сохранение списка отправки

MCL0: node_%d = ok/false

регистратор: соединение с узлом N в норме / невозможно установить соединение с узлом N (нет адреса, сокет не создан)

MC00: node_%d = ok/false

соединение с узлом N в норме / невозможно установить соединение с узлом N (нет адреса, сокет не создан)

MCL0:node_%d = E%d

регистратор: ошибка M подключения к узлу N

MC00:node_%d = E%d

ошибка M подключения к узлу N

MCL0:node_%d=err

регистратор: соединение с узлом в норме, ошибка отправки первого блока и отключение

MC00:node_%d=err

соединение с узлом в норме, ошибка отправки первого блока и отключение

MCL0:node_%d=wrong, t

регистратор: соединение с узлом в норме, первый блок отправлен, отключение по таймауту ответа

MC00:node_%d=wrong,t

соединение с узлом в норме, первый блок отправлен, отключение по таймауту ответа

MCL0:node_%d=wrong,err=%d

регистратор: соединение с узлом в норме, первый блок отправлен, отключение по ошибке чтения ответа

MC00:node_%d=wrong,err=%d

соединение с узлом в норме, первый блок отправлен, отключение по ошибке чтения ответа

MCL0:node%d=bad,unpresent

регистратор: соединение с узлом в норме, первый блок отправлен, ошибка данных в ответе узла

MC00:node%d=bad,unpresent

соединение с узлом в норме, первый блок отправлен, ошибка данных в ответе узла

MCL0:node%d=OK

регистратор: соединение с узлом в норме

MC00:node%d=OK

соединение с узлом в норме

MCL0:node_%d= check OK

регистратор: соединение с узлом в норме, проверка в норме, отключение обоих

MC00:node_%d= check OK

соединение с узлом в норме, проверка в норме, отключение обоих

MTL0: ERR=%d

ошибка отсылки в регистратор

MTL0: sleep=%d

небольшое ожидание отсылки в регистратор (норма)

MTL0: waite

отсылка в регистратор по доступности IP (норма)

MTL0: [TCP=CLOSE] node_%d, send=%d, {all, allT}

отсылка в регистратор, **allT** – остановка по времени, **all** – отсылка всех данных

MTL0: TCP=actC

отсылка проверочного блока b100 в регистратор по TCP

MTL0: TCP=CLOSE, sts=trace

разрыв соединения при переходе узла в статус TRACE или ручную

MTL0: TCP=CLOSE, manual

разрыв соединения вручную

MTL0: TCP=actP

намерение подключиться к регистратору по ошибке по времени

MTL0: TCP=actN

намерение подключиться к регистратору (норма)

RC100: TCP=CHECK, CLOSE node_%d

прием проверочного блока b100

Механизм предотвращения потерь при обмене по сети

(См. **Механизм предотвращения потерь при обмене по сети**):

WD11: sz=%d

HOST обработал данные

>D11: node_%d sz=%d

SLAVE послал данные

>M11: node_%d

HOST послал запрос

RM11: not supported by%d

SLAVE принял запрос и ответил, что такой запрос не поддерживает

RM11: node_%d total=%d

SLAVE принял запрос и подготовил TOTAL данных

RD11: node_%d sz=%d

SLAVE отослал **SZ** данных

RD11: node_%d total=%d

SLAVE отослал TOTAL данных

BN11: ON

SLAVE включил накопление

BN11: OFF,CLEAR

SLAVE выключил накопление и очистил буфер

BD11: ON

SLAVE включил накопление после старта

BD11: ON,CLEAR

SLAVE включил и очистил накопление после старта

Для отмены генерации сообщений используется ключ **DBG_INFO_NODE=OFF**.

Сообщения при **DBG_INFO_MARKER=ON**

Если в файле *.cnf задан ключ **DBG_INFO_MARKER=ON** (или установлен бит 7 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие сообщения:

ST199: node_%d [send=wrong]

узел N: запрос на разрыв соединения

SC199: node_%d [send=wrong]

узел N: подтверждение запроса на разрыв соединения

RT199: node_%d, TCP=close

узел N: прием b199, разрыв соединения

RC199: node_%d, TCP=close

узел N: прием b199, подтверждение разрыва соединения

MS100: node_%d

узел N: посылка b100

MR100: node_%d=ACK

узел N: прием ответа на b100

MR100: node_%d

узел N: прием b100

S100=c%d any:%d

широковещательная посылка b100 через адаптер c%d на указанный порт

S100=c%d node_%d %s:%d [I W N D R]

посылка b100 через адаптер c%d на IP-адрес %s:%d узла N

- **I** – индивидуальная посылка
- **W** – прямая посылка
- **D** – прямая посылка (таймаут записи)
- **R** – требование ответа
- **N** – удаленный узел не может принимать широковещательные посылки

S000:c%d any:%d

широковещательная посылка b000 (данные автопосылки) через адаптер c%d на указанный порт

S000=c%d node_%d %s:%d bB0

посылка b000 через адаптер c%d на IP-адрес %s:%d узла N (первый мост)

S000=c%d node_%d %s:%d bB1

посылка b000 через адаптер c%d на IP-адрес %s:%d узла N (второй мост)

R100,any=node_%d,sts=%d;[tcp=%s:%d;][W R]

прием широковещательного b100 от узла N со статусом sts и адресом tcp

R100,dir=node_%d,sts=%d;[tcp=%s:%d;][W R]

прием прямого b100 от узла N со статусом sts и адресом tcp

R000,any=node_%d,sts=%d;[tcp=%s:%d;][W R]

прием широковещательного b000 от узла N со статусом sts и адресом tcp

R000,dir=node_%d,sts=%d;[tcp=%s:%d;][W R]

прием прямого b000 от узла N со статусом sts и адресом tcp

RC199: TCP=CLOSE, same node_%d

регистратор принимает первый блок, разрывает соединение и со-

здает новое соединение

```
RC199: node_%d, TCP=CLOSE, rec=%d
```

регистратор принимает от узла N блок отключения и разрывает соединение (rec=%d – блок в сессии)

Для отмены генерации сообщений используется ключ **DBG_INFO_MARKER = OFF**.

Сообщения при SYNC_MARKER=ON

Если в файле *.cnf задан ключ **SYNC_MARKER=ON**, могут генерироваться следующие отладочные сообщения резервирования.

Узел **TRACE**:

```
?->107 / !->107
```

посылка запроса на получение начального блока синхронизации (? – по времени, ! – норма)

```
<-R%.3d count=%d
```

прием блока синхронизации типа N

```
<-108
```

прием последнего блока синхронизации

```
<-R112
```

прием блока ожидания

Узел **WORK**:

```
<-R107 [first]
```

прием запроса на получение начального блока синхронизации

```
->S%.3d
```

посылка начального блока синхронизации

```
->S%.3d count=%d
```

посылка блока синхронизации типа N

```
->S108
```

посылка последнего блока синхронизации

Типы блоков синхронизации:

- 107 – начальный;
- 108 – последний;
- 109/110 – данные;
- 112 – блок ожидания;

- 113 – данные SIAD;
- 115 – данные каналов Vector;
- 116 – данные каналов СОБЫТИЕ.

Для отмены генерации сообщений используется ключ **SYNC_MARKER = OFF**.

Сообщения при **DBG_PEVE_TCP=ON**

Если в файле *.cnf задан ключ **DBG_PEVE_TCP=ON** (или установлен бит 28 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие сообщения.

Протокол IEC104 SLAVE

iec104 err=%d

глобальная ошибка сокета прослушивания

iec104 listen %s

iec104 connect from %s

iec104 manual disconnect from %s

iec104 disconnect from %s by wrong frame %d

iec104 soc close by timeout

=> TESTFR(act) [,err=%d] [soc close]

Протокол MODBUS SLAVE

smdb%d bind %s,err=%d

smdb%d listen %s,err=%d

smdb%d listen %s

smbd%d accept %s

smbd%d disconnect from %s by wrong frame %d

smbd%d manual disconnect from %s

smbd%d soc close by timeout

Протокол HOST

p%dHOST bind %s:%u[err=%d]

HOST %s:%u connect to %s

%s:%u TO %s:%u

Протокол UDP

UDP HOST bind %s:%u[err=%d]

UDP Omron:th%x bind %s:%u[err=%d]

UDP Omron: %s to bad

UDP Omron: unit%d %s restore nTH%d

SNMP: unit%d %s restore nTH%d

Snmp: %s to bad

Для отмены генерации сообщений используется ключ **DBG_PEVE_TCP**
= **OFF**.

Сообщения при DBG_INFO_SLAVE=ON

Если в файле *.cnf задан ключ **DBG_INFO_SLAVE=ON** (или установлен бит 27 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие отладочные сообщения о SLAVE-протоколах.

=> TESTFR (act) [,err=%d] [soc close]

сообщение IEC104 в случае ошибки

=> TESTFR (act)

сообщение IEC104, если обмен в норме

iec104 waite send,not ack (%d)

ожидание отправки данных, подтверждения нет

iec104 send,restored (%d)

-> IDT=100,COT=%d,data=%d

-> IDT=101,COT=%d,data=%d

-> I-frame, size=%d, err=%d

посылка ошибки i_frame

-> I-frame, size=%d, IDT=%d COT=%d

посылка i_frame

**<- I-frame IDT=%d COT=%d host=(S)%d (R)%d
slave=(S)%d (R)%d -> rec i_frame**

<= S-frame host=(S)%d (R)%d slave=(S)%d (R)%d

<= STARTDT (act)

прием начальной команды клиента

=> STARTDT (con) [err=%d]

посылка уведомления об активации

```
<= STOPDT (act)
```

прием конечной команды клиента

```
=> STOPDT (con) [err=%d]
```

посылка уведомления об окончании сессии

```
<= TESTFR (act)
```

```
=> TESTFR (con) [err=%d]
```

```
<= STARTDT (con)
```

прием запроса клиента о посылке уведомления об активации

```
<= STOPDT (con)
```

прием запроса клиента о посылке уведомления об окончании сессии

```
<= TESTFR (con)
```

```
smdb%d<- cmd=%d a=%X c=%d
```

```
smdb%d send <tcp addr>,err=%d [soc close]
```

Для отмены генерации сообщений используется ключ **DBG_INFO_SLAVE = OFF**.

Сообщения при **DBG_INFO_TCP=ON**

Если в файле *.cnf задан ключ **DBG_INFO_TCP=ON** (или установлен бит 16 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие отладочные сообщения.

```
r%d, %d. %d<ch%d> [nTH%d] ->%s
```

число байт, протокол, адрес, ID канала, {число байт данных, HEX}
(посылка)

```
r%d,%d.%d<ch%d>[nTH%d]<-%s
```

число байт, протокол, адрес, ID канала, {число байт данных, HEX}
(прием)

```
MDB: unit_%d(%d) %s nTH%d TMOUT=%d
```

```
%s: CMD=%d CH=%x
```

```
channels : total=%d,output=%d,scoff=%d,wcoff=%d
```

```
MITSUBISHI: unit_%d(%d) %s nTH%d TMOUT=%d
```

```
IEC104: unit_%d_%d(lch%d) %s sz(%d,%d,%d,%d)
```

хост, общий адрес ASDU, ID канала, TCP-адрес

Для отмены генерации сообщений используется ключ **DBG_INFO_TCP = OFF**.

Сообщения при **DBG_INFO_IEC104=ON**

Если в файле *.cnf задан ключ **DBG_INFO_IEC104=ON** (или установлен бит 25 переменной 14.14 **@Debug** с **Параметр=1**), могут генерироваться следующие отладочные сообщения.

Примечания:

59 atr main channel for 104

p2 – низкий приоритет;

Q – для выхода.

```
{ ,Q,p2}ADDR=%d CASDU=%d COT=%d IOA=%d VAL=%d QV=%x  
IDT=%d
```

```
{ ,Q,p2}ADDR=%d CASDU=%d COT=%d IOA=%d VAL=%g QV=%x  
IDT=%d
```

```
{ ,Q,p2}ADDR=%d CASDU=%d COT=%d IOA=%d VAL=%x(%d)  
QV=%x IDT=%d
```

```
=====
```

```
-> STOPDT(act) %s
```

```
-> IDT100(20) %s[,err=%d]
```

```
-> IDT101(5) %s[,err=%d]
```

```
-> S-frame %s[,err=%d]
```

```
iecl04 disconnect from %s,by wrong frame=%d
```

```
iecl04 manual disconnect from %s
```

```
-> STARTDT(act) %s,err=%d,soc close
```

```
-> STARTDT(act) %s
```

```
%s,timeout,soc close -> afte testfr
```

```
-> TESTFR(act) %s[err=%d][soc close]
```

```
HOST waite send data to %s,not ack %d
```

```
HOST restore send data to %s
```

```
-> I-frame %s,err=%d
```

```
-> I-frame %s size=%d,IDT=%d COT=%d
```

```
<- rec err=%d %s
```

```
unknown first symbol
```

```
<- I-frame, size=%d %s, IDT=%d COT=%d
```

```
<- S-frame %s
```

```
<- STARTDT (con) %s
```

```
<- STOPDT (con) %s
```

```
<- TESTFR (con) %s
```

```
<- STARTDT (act) %s
```

```
<- TESTFR (act) %s
```

Для отмены генерации сообщений используется ключ **DBG_INFO_IEC104 = OFF**.

Сообщения при **DBG_INFO_LOGGER=ON**

Если в файле *.cnf задан ключ **DBG_INFO_LOGGER=ON**, могут генерироваться следующие отладочные сообщения.

HOST:

```
<id ch send> = <value> <time>.<ms> -> [ch name  
recieve]
```

SLAVE:

```
<ch name send><id ch send> = <value> <time>.<ms>
```

Для отмены генерации сообщений используется ключ **DBG_INFO_LOGGER = OFF**.

AdAstra Research Group, Ltd

107076 Россия, Москва, а/я 38

Тел. (495) 771-71-74

Факс (495) 518-98-46

E-mail: adastra@adastra.ru

<http://www.adastra.ru>