



НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ
СИСТЕМОТЕХНИКА-НН

Утвержден

НБКГ.00001 РП ЛУ



Система программирования и исполнения «Scorpio»

Руководство программиста

НБКГ.00001 РП

Листов 304

EAC

г. Нижний Новгород

2014 г.

АННОТАЦИЯ

Данное руководство предназначено для изучения правил и методов разработки и отладки алгоритма для контроллеров КСА-02 с помощью системы программирования и исполнения «Scorpio».

Система программирования и исполнения «Scorpio» (система «Scorpio») — это комплекс программ, предназначенный для разработки, отладки и исполнения алгоритмов рабочей программы контроллера КСА-02, а также для конфигурирования, диагностики и работы модулей, входящих в состав контроллера.

Инструментальное средство разработки алгоритма для контроллера КСА-02 — язык функциональных блоковых диаграмм (FBD), представляющий собой механизм объектного визуального программирования.

Для корректной работы системы «Scorpio» к ПЭВМ предъявляются следующие требования:

- установлена одна из следующих операционных систем - Astra Linux, Debian GNU/Linux, Ubuntu, MS Windows;
- свободное пространство на диске не менее 200 Мб;
- графический видеоадаптер с разрешением не менее 800x600;
- последовательный порт COM1 или COM2, также допускается использование эмулятора последовательного порта;
- клавиатура и манипулятор «мышь».

В руководстве представлена следующая информация:

- архитектура системы «Scorpio»;
- основные концепции построения и функционирования системы «Scorpio»;
- описание языка функциональных блоковых диаграмм (FBD), использующегося для разработки алгоритма рабочей программы для контроллера КСА-02;
- описание правил и наиболее эффективных способов создания и редактирования алгоритма рабочей программы для контроллера КСА-02;
- описание сервисной программы контроллера КСА-02, использующейся для управления работой контроллера КСА-02 непосредственно с ПЭВМ;
- описание протоколов обмена данными, используемых в системе «Scorpio».

Руководство рассчитано на инженерно-технических работников, занимающихся автоматизированными системами управления технологическими процессами.

Содержание

1. Контроллер КСА-02.....	11
1.1. Что такое контроллер КСА-02.....	11
1.2. Модули контроллера КСА-02.....	13
1.2.1. Модуль питания (СТ1CPS220, СТ2CPS220, СТ1CPS24).....	14
1.2.2. Процессорные модули.....	16
1.2.3. Модули ввода/вывода.....	18
1.2.3.1. Модули ввода аналоговых сигналов (на примере модуля СТ1АСИ08).....	18
1.2.3.3. Модули ввода дискретных сигналов постоянного тока.....	23
1.2.3.4. Модуль ввода дискретных сигналов переменного тока СТ1ДАИ16.....	25
1.2.3.6. Особенности модулей ввода/вывода дискретных сигналов.....	29
1.2.3.7. Модуль управления кранами СТ1ВСТ02 (СТ1ВСТ03).....	31
1.2.4.1. Модуль горячего резервирования СТ1НСВ10 (СТ2НСВ10).....	34
2. Основные концепции, понятия, определения.....	44
2.1. Структура системы «Scorpio».....	44
2.1.1. Базовая директория.....	44
2.1.2. Директория проекта.....	45
2.1.3. Типы данных.....	46
2.1.4. Протоколы передачи данных.....	48
2.1.5. Протоколы синхронизации времени.....	49
3. Язык функциональных блок-диаграмм (FBD).....	50
3.1. Основные элементы языка программирования FBD.....	50
3.1.1. Диаграмма.....	50
3.1.2. Функция.....	50
3.1.3. Составная функция.....	51
3.1.4. Блок.....	52
3.1.5. Рамки.....	54
3.1.6. Связь.....	54
3.1.7. Теги.....	56
3.1.8. Контакт.....	58
3.1.8.1. Типы контактов.....	59
3.1.8.2. Графическое изображение контакта.....	60
3.2. Правила языка FBD.....	62
4. Разработка алгоритма рабочей программы для контроллера КСА-02.....	65
4.1. Среда разработки алгоритма «FBD_Builder».....	65
4.1.1. Загрузка.....	65
4.2. Редактор параметров контроллера КСА-02.....	66
4.2.1. Назначение и основные принципы работы.....	66
4.2.2. Настройка параметров контроллера КСА-02.....	67
4.2.2.1. Порты. Общие положения.....	67
4.2.2.2. Порты. Окна настройки протоколов.....	69
4.2.2.3. Порты. Поддержка семейства протоколов МЭК870 (IEC60870).....	73
4.2.2.4. Порты. Поддержка протокола NTP.....	75
4.2.2.5. Буферы.....	76
4.2.2.6. Резервирование.....	78
4.2.2.7. Удаленный ввод/вывод.....	80
4.2.2.8. Модули СТ1СРN.....	82
4.2.2.9. Настройки COM1 и COM2 для модуля СТ 1СРM10.....	83
4.2.2.10. Настройка протокола Modbus-Slave.....	84
4.2.2.11. Настройка протокола RTM64-Slave.....	85

4.2.2.12. Настройка протоколов Modbus-Master, МКSA-Master.....	87
4.2.2.13. Протоколы для модуля СТ1СРЕ10.....	88
4.2.2.14. Общие настройки протоколов и настройка протокола Modbus-TCP (slave).....	88
4.2.2.15. Особенности настройки протокола Proton3-IP (slave).....	90
4.2.2.16. Настройка протокола МЭК870-5-104 (IEC60870-104).....	90
4.2.2.17. Настройка протокола Modbus-TCP (master).....	92
4.2.3. Сообщения редактора параметров контроллера КСА-02 об ошибках.....	93
4.3. Редактор языка «FBD_Builder».....	97
4.3.1. Общие сведения.....	97
4.3.2. Команды «FBD_Builder».....	97
4.3.2.1. Меню «FBD_Builder».....	97
4.3.2.2. Панель инструментов «FBD_Builder».....	99
4.3.2.3. Контекстное меню «FBD_Builder».....	100
4.3.3. Свойства проекта.....	101
4.3.3.1. Свойства проекта алгоритма.....	101
4.3.3.2. Свойства проекта составной функции.....	103
4.4. Порядок построения схемы алгоритма в программе «FBD_Builder».....	104
4.4.1. Вставка нового блока функции, перемещение и удаление блоков.....	104
4.4.2. Создание связи между выходами и входами блоков.....	105
4.4.3. Установка инверсии на вход функции.....	107
4.4.4. Вставка входного значения (привязка).....	108
4.4.5. Удаление входного значения (отвязка).....	110
4.4.6. Ввод комментариев.....	110
4.4.7. Окно «Теги».....	111
4.4.8. Окно «составные функции».....	113
4.5. Компиляция алгоритма.....	116
4.5.1. Ошибки и предупреждения.....	118
4.6. Справочная система.....	119
4.6.1. Справка по программе.....	119
4.7. Способы изменения масштаба и местоположения изображения.....	122
4.8. Способы копирования объектов диаграммы.....	123
4.9. Переименование тегов и изменение текста комментариев при копировании логически целостных частей алгоритма.....	124
4.10. Пример разработки алгоритма рабочей программы для контроллера КСА-02.....	126
5. Сервисная программа контроллера КСА-02.....	129
5.1. Условия для корректной работы сервисной программы на ПЭВМ.....	131
5.2. Соединение ПЭВМ с контроллером КСА-02.....	132
5.3. Оболочка сервисной программы.....	133
5.4. Пункты главного меню. Состав и назначение.....	136
5.4.1. Меню «Файл».....	139
5.4.1.1. Пункт «Загрузить конфигурацию».....	140
5.4.1.2. Пункт «Сохранить конфигурацию».....	141
5.4.1.3. Пункт «Открыть проект».....	142
5.4.1.4. Пункт «Открыть сессию».....	143
5.4.1.5. Пункт «Сохранить сессию».....	143
5.4.1.6. Открытие проекта выбором из списка.....	143
5.4.1.7. Пункт «Выход».....	143
5.4.2. Меню «Мониторинг».....	144
5.4.2.1. Пункт «Обновить данные».....	144
5.4.2.2. Пункт «Включить анимацию».....	144

5.4.2.3. Пункт «Выключить анимацию».....	145
5.4.2.4. Пункт «Добавить окно DM».....	145
5.4.2.5. Пункт «Добавить окно PM».....	145
5.4.2.6. Пункт «Добавить окно модуля».....	146
5.4.2.7. Пункт «Добавить окно модуля сети СУВД».....	150
5.4.2.8. Пункт «Буфер входных данных (Logic)».....	151
5.4.2.9. Пункт «Буфер входных данных (Integer)».....	153
5.4.2.10. Пункт «Буфер входных данных (Float)».....	154
5.4.2.11. Пункт «Буфер внутренних данных».....	154
5.4.3. Меню «Управление».....	156
5.4.3.1. Пункт «Запустить/Остановить ядро».....	156
5.4.3.2. Пункт «Запустить/Остановить алгоритм».....	156
5.4.3.3. Пункт «Перезапуск контроллера».....	156
5.4.3.4. Пункт «Загрузить программу».....	157
5.4.3.5. Пункт «Управление соединением».....	157
5.4.4. Меню «Настройка».....	157
5.4.4.1. Пункт «Заводские настройки».....	157
5.4.4.2. Пункт «Конфигурация контроллера».....	158
5.4.4.3. Пункт «Статистика работы сети СУВД».....	159
5.4.4.4. Пункт «Статистика связи по интерфейсам».....	160
5.4.4.5. Пункт «Статистика связи по шине».....	162
5.4.4.6. Пункт «Установка часов».....	162
5.4.5. Меню «Алгоритм».....	163
5.4.5.1. Пункт «Удалить алгоритм».....	163
5.4.5.2. Пункт «Загрузить алгоритм».....	163
5.4.5.3. Пункт «Информация об алгоритме».....	163
5.4.5.4. Пункт «Отладка».....	164
5.4.5.5. Пункт «Убрать все функции».....	165
5.4.6. Меню «Тесты».....	166
5.4.6.1. Пункт «Тесты модулей ввода/вывода».....	166
5.4.6.2. Пункт «Тест СТ2СРУЗЗ».....	166
5.4.6.3. Пункт «Самодиагностика».....	166
5.4.6.4. Пункт «Тест общей памяти».....	166
5.4.6.5. Пункт «Тест Ethernet».....	167
5.4.6.6. Пункт «Тест СОМ-портов».....	167
5.4.7. Меню «Окна».....	167
5.4.7.1. Пункт «Каскадом».....	167
5.4.7.2. Пункт «Упорядочить».....	168
5.4.7.3. Пункт «Свернуть все».....	169
5.4.7.4. Пункт «Восстановить все».....	169
5.4.7.5. Пункт «Закрыть все».....	169
5.4.7.6. Пункт «Управление».....	170
5.4.8. Меню «Помощь».....	170
5.4.8.1. Пункт «Справка по программе».....	171
5.4.8.2. Пункт «Справка по функциям».....	171
5.4.8.3. Пункт «Справка по контроллеру».....	171
5.4.8.4. Пункт «О программе».....	171
5.5. Старт сервисной программы контроллера КСА-02.....	172
5.5.1. Общие сведения о работе с сервисной программой контроллера КСА-02.....	172
5.5.2. Настройка подключения ПЭВМ к контроллеру КСА-02.....	173
5.5.3. Запись параметров подключения контроллера КСА-02 к ПЭВМ.....	174

5.5.4. Индикация результатов старта сервисной программы.....	174
5.6. Конфигурирование контроллера КСА-02.....	176
5.6.1. Вызов окна «Конфигурация контроллера».....	176
5.6.2. Информационные поля и кнопки окна «Конфигурация контроллера».....	176
5.7. Настройка модулей контроллера КСА-02.....	179
5.7.1. Настройка модулей СТ1АСI08.....	179
5.7.2. Настройка модулей СТ1АСO04.....	184
5.7.3. Настройка модуля СТ1АRI08.....	186
5.7.4. Настройка модуля СТ1АTI08.....	190
5.7.5. Настройка модуля СТ1ВСТ02, СТ1ВСТ03.....	193
5.7.6. Настройка модулей СТ1DIO29, СТ2DIO29, СТ3DIO29.....	195
5.7.7. Настройка модулей СТ3DDI30 – СТ10DDI30.....	197
5.7.7.1. Настройка модулей СТ3DDI30, СТ5DDI30, СТ7DDI30, СТ9DDI30.....	198
5.7.7.2. Особенности настройки модулей СТ4DDI30, СТ6DDI30, СТ8DDI30, СТ10DDI30.....	199
5.7.8. Настройка модулей СТ2DDO30.....	200
5.7.9. Настройка модулей СТ1DAI16.....	202
5.8. Настройка параметров контроллера КСА-02.....	204
5.8.1. Параметры контроллера КСА-02.....	204
5.8.1.1. Параметры главного процессора.....	204
5.8.1.2. Настройка параметров портов COM1 и COM2.....	206
5.8.1.3. Параметры буферов контроллера КСА-02.....	208
5.8.1.4. Настройка параметров резервирования.....	210
5.8.1.5. Настройка параметров удаленного ввода-вывода.....	213
5.8.1.6. Модули СТ1СРN.....	216
5.8.1.7. Архивы.....	222
5.9. Тестирование контроллера КСА-02.....	225
5.9.1. Работа в режиме «Тесты модулей ввода/вывода».....	226
5.9.2. Тест модуля СТ2СРU33.....	227
5.9.2.1. Общие сведения.....	227
5.9.2.2. Управление.....	229
5.9.2.3. Автоматический тест модуля СТ2СРU33.....	230
5.9.2.4. Тест процессора.....	231
5.9.2.5. Тест ОЗУ.....	232
5.9.2.6. Тест SRAM/NVSRAM.....	234
5.9.2.7. Тест Flash.....	237
5.9.2.8. Тест COM1 и COM2.....	239
5.9.3. Самодиагностика.....	241
5.9.4. Тест общей памяти.....	244
5.9.4.1. Общие сведения.....	244
5.9.4.2. Управление.....	245
5.9.4.3. Тест Shared Memory.....	246
5.9.4.4. Настройки.....	246
5.9.4.5. Статистика.....	247
5.9.4.6. Отображение информации о работе теста общей памяти.....	248
5.9.5. Тест Ethernet.....	248
5.9.5.1. Общие сведения.....	248
5.9.5.2. Управление.....	249
5.9.5.3. Тест Ethernet.....	250
5.9.5.4. Настройки.....	250
5.9.5.5. Статистика (модуль CPU).....	250

5.9.5.6. Статистика (модуль СРЕ).....	251
5.9.5.7. Формат отображения текстового сообщения из категории «ошибка чтения/ записи».....	251
5.9.6. Тест СОМ-портов.....	251
5.9.6.1. Общие сведения.....	251
5.9.6.2. Управление.....	252
5.9.6.3. Тест СОМ-портов.....	253
5.9.6.4. Настройки.....	253
5.9.6.5. Статистика.....	254
5.9.6.6. Статистика СОМ- портов.....	254
5.10. Предупредительные и аварийные сообщения сервисной программы.....	256
5.10.1. Классификация и способы отображения предупредительных и аварийных сообщений сервисной программы.....	256
5.10.2. Предупредительные и аварийные сообщения, отображаемые в нижней информационной строке основного окна сервисной программы.....	257
5.10.3. Предупредительные и аварийные сообщения, отображаемые в окне «Ошибка».....	259
5.11. Справочная система.....	261
6. Протоколы.....	262
6.1. Введение.....	262
6.1.1. Физические линии связи.....	262
6.1.2. Принципы организации обмена данными.....	262
6.1.3. Программные уровни.....	263
6.2. Протокол Modbus в режиме Master (Modbus-Master).....	265
6.2.1. Общие сведения.....	265
6.2.2. Типы данных.....	265
6.2.3. Адресация и функции доступа.....	266
6.2.4. Использование в алгоритме.....	267
6.3. Протокол Modbus в режиме Slave (Modbus-Slave).....	271
6.3.1. Общие сведения.....	271
6.3.2. Особенности адресации и использования функций доступа.....	271
6.3.3. Использование в алгоритме.....	272
6.4. Особенности протокола Modbus-TCP.....	273
6.4.1. Особенности Modbus-TCP (Master).....	273
6.4.2. Особенности Modbus-TCP (Slave).....	275
6.5. Протокол МКСА в режиме Master (МКСА-Master).....	275
6.5.1. Общие сведения.....	275
6.5.2. Основная характеристика контроллеров МКСА.....	275
6.5.3. Коммуникационное устройство связи со счетчиками "Меркурий".....	276
6.5.4. Использование в алгоритме.....	277
6.6. Протокол RTM64 в режиме Slave.....	278
6.7. Семейство протоколов Proton3.....	278
6.7.1. Общие сведения.....	278
6.7.2. Ресурсы и типы данных.....	279
6.7.3. Использование в алгоритме.....	280
6.8. Протокол МЭК870-5-104 (IEC60870-104).....	282
6.8.1. Общие сведения.....	282
6.8.2. Типы данных.....	283
6.8.3. Размещение данных МЭК870 в буферах контроллера.....	284
6.8.4. Обращение к данным МЭК870 из алгоритма.....	285
ПРИЛОЖЕНИЕ.....	287

1. Словарь:.....	287
2. Зарезервированные символы для формирования литеры:.....	293
3. Описание сообщений компилятора об ошибках.....	294
4. Архитектура системы «Scorpio».....	304

Общие сведения

Система программирования и исполнения «Scorpio» — комплекс программ, выполняемых и взаимодействующих под управлением пользователя (прикладного программиста). Она предназначена для разработки, отладки и исполнения алгоритмов рабочей программы контроллера КСА-02.

Прежде чем изучать эту систему, нужно понимать, для каких именно целей она служит. Для этого необходимо рассмотреть, что же такое контроллер КСА-02, как он устроен и каким образом он работает с алгоритмами, создаваемыми в системе «Scorpio». Кроме того, нужно представлять себе и сам технологический процесс, ради которого существует и контроллер, и его алгоритмы.

Контроллер КСА-02 — устройство, которое измеряет и обрабатывает сигналы, поступающие от датчиков и сигнализаторов, установленных на технологическом оборудовании, формирует команды и воздействия на объекты управления, а также осуществляет связь с системой верхнего уровня (например, с АРМ оператора) или подсистемами нижнего уровня (подчинёнными устройствами).

Все эти сигналы, команды, воздействия, вся информация, попадающая в контроллер, обрабатываемая в нём и отправляемая им после обработки на внешние устройства, кратко и суммарно называется данными, а процесс передвижения данных по системе управления технологическим процессом называется обменом данными.

Во всей системе управления технологическим процессом условно выделяют три уровня, участвующих в обмене данными.

Верхний уровень, или SCADA-система.

Основные функции SCADA:

- контролирование технологического процесса;
- сбор, обработка и передача данных;
- отображение данных в цифровом и графическом виде (вывод на экран монитора в виде, удобном для обслуживающего персонала);
- оперативное информирование персонала (тревоги и события);
- архивирование (базы данных, библиотека, паспортизация оборудования);
- формирование отчётов;

- разграничение доступа (авторизация пользователей).

Средний уровень — это контроллер и загруженный в него алгоритм.

На этот уровень возлагается функция автоматического выполнения наиболее ответственных частей алгоритма, таких как аварийные защиты, формирование управляющих воздействий на исполнительные механизмы, выполнение расчетов, зависящих от времени.

Средний уровень является связующим звеном между верхним и нижним уровнем.

На этом этапе информация, поступающая с обеих сторон, обрабатывается и распределяется по назначению.

Нижний уровень — это устройства, которые можно разделить на две группы:

- устройства ввода, информация от которых поступает в контроллер на обработку: датчики, сигнализаторы;
- устройства вывода (исполнительные устройства), которые принимают обработанную информацию от контроллера в виде команд: двигатели, насосы, силовые приводы управления устройствами, высоковольтные выключатели и т.д.

Все три уровня постоянно обмениваются информацией через контроллер в обоих направлениях.

В данной инструкции речь пойдет именно о среднем уровне, то есть о контроллере и том программном и техническом содержании, которое находится внутри него.

1. Контроллер КСА-02

1.1. Что такое контроллер КСА-02

Контроллер КСА-02 представляет собой набор модулей, которые объединены металлическим корпусом. Корпус, содержащий в себе набор модулей, подобранный для выполнения определённой задачи, называется корзиной. В составе контроллера должна быть как минимум одна корзина главного процессора (ГП), в которой присутствует процессорный модуль с исполняемым алгоритмом рабочей программы контроллера. В случае необходимости удаленного сбора данных или увеличения количества сигналов, подключаемых к контроллеру он может комплектоваться корзинами станций удаленного ввода-вывода (СУВД).

Средний срок службы контроллера 12 лет.

Масса контроллера не более 5 кг, высота его корпуса 265 мм, глубина 130 мм. Длина корзины зависит от типа применённого корпуса.

Контроллер КСА-02 может иметь в своем составе до 8 СОМ-портов, при помощи которых он имеет возможность связи с системой верхнего уровня или подсистемами нижнего уровня по интерфейсам RS-232 или RS-485.

Контроллер КСА-02 может иметь в своем составе до 3 Ethernet-портов, при помощи которых он имеет возможность связи с системой верхнего уровня или подсистемами нижнего уровня.

Контроллер способен принимать дискретные сигналы телесигнализации и выдавать дискретные сигналы телеуправления.

Контроллер способен принимать системные аналоговые сигналы телеизмерений — токовые сигналы от 4 до 20мА, и выдавать системные аналоговые сигналы телерегулирования — токовые сигналы от 4 до 20мА.

Кроме того, контроллер способен напрямую принимать сигналы от термометров сопротивления и термопар.

Для работы с контроллером имеется мощный инструментарий — система программирования и исполнения «Scorpio». С ее помощью осуществляется удобная, легкая разработка и отладка алгоритма, выполняемого контроллером КСА-02, конфигурирование контроллера и калибровка входящих в его состав модулей.

Контроллер выполняет заложенный в него алгоритм независимо от наличия связи с системой верхнего уровня.

Существует возможность горячего [резервирования](#) главных процессоров контроллера. [Станция удаленного ввода-вывода](#) может обрабатывать такое же количество сигналов, как и главный процессор, но не может иметь выполняемого алгоритма и сети ведомых устройств.

Часто в системе имеются одновременно два главных процессора, дублирующие друг друга для большей надёжности работы системы. При этом один из них является ведущим (основным), а второй — ведомым (резервным). Оба ГП выполняют одинаковый алгоритм обработки данных. Разница лишь в том, что ведущий выдаёт системе команды на управление процессом, а ведомый этих команд не выдаёт, но следит за исправностью работы ведущего. Если ведущий выйдет из строя, то ведомый тут же становится ведущим и берёт на себя все его функции. Соответственно, вся оперативная информация должна поступать ведущему ГП.

1.2. Модули контроллера КСА-02

Набор модулей в корзине не всегда одинаков и подбирается в зависимости от конкретной технологической задачи. Однако существует ограничение на максимальное число слов, занимаемых модулями ввода/вывода в памяти процессорного модуля (не более 128). Соответственно, ограничено и количество модулей ввода/вывода, т.к. эти модули занимают наибольшее число слов памяти — 10 слов для аналоговых и до 19 слов — для дискретных. Кроме этого есть ограничение на суммарный потребляемый модулями ток от блока питания (не более 10 А).

Таким образом, суммарное количество измеряемых сигналов зависит от конфигурации контроллера. В корзине контроллера КСА-02 обязательно должны присутствовать модуль питания и процессорный модуль. Таким образом, например, при применении в контроллере каркаса на максимальное число модулей (16), возможно установить только 14 модулей ввода/вывода (из них до трёх интерфейсных модулей).

Модули имеют функционально различное назначение, которое будет рассмотрено ниже.

В контроллере КСА-02 возможно применение следующих модулей:

- модуль питания: СТ1CPS220, СТ2CPS220, СТ1CPS24;
- процессорный модуль: СТ2CPU33, СТ1RPU33;
- модули ввода/вывода:
- модули ввода аналоговых сигналов: СТ1ACI08, СТ2ACI08⁽¹⁾, СТ1ARI08, СТ1ATI08;
- модуль вывода аналоговых сигналов: СТ1ACO08⁽¹⁾, СТ1ACO04;
- модули ввода дискретных сигналов: СТ1DDI30⁽¹⁾, СТ2DDI30⁽¹⁾, СТ1DAI16, СТ3DDI30, СТ4DDI30, СТ5DDI30, СТ6DDI30, СТ7DDI30, СТ8DDI30, СТ9DDI30, СТ10DDI30 ;
- модуль вывода дискретных сигналов: СТ1DDO30⁽¹⁾, СТ2DDO30;
- модуль ввода-вывода дискретных сигналов: СТ1DIO29, СТ2DIO29, СТ3DIO29;
- модули управления кранами: СТ1BCT02, СТ1BCT03;
- коммуникационные модули:
- специальные модули: СТ1HSB10, СТ2HSB10, СТ1RHA33;
- интерфейсные модули: СТ1CPM10, СТ1CPE10.

Примечание. ⁽¹⁾ Модуль снят с производства. Описание модуля в данном руководстве удалено или сокращено. Для получения более полных сведений о данном модуле смотрите предыдущую версию руководства.

1.2.1. Модуль питания (СТ1CPS220, СТ2CPS220, СТ1CPS24)

Модуль обеспечивает электропитание всех модулей в корзине. В корпусе контроллера модуль питания всегда устанавливается на позиции 1.

Модули питания CPS обеспечивают питание модулей, установленных в контроллере (по шине) номинальным напряжением 5В постоянного тока.

Модули питания СТ2CPS220 дополнительно обеспечивают питание внешних устройств номинальным напряжением 12В постоянного тока.

Выходной ток модулей питания СТ1CPS220 и СТ1CPS24 (по шине) не менее 10А.

Выходной ток модулей питания СТ2CPS220 (по шине) не менее 6А. Выходной ток модулей питания СТ2CPS220 для питания внешних устройств не менее 2А.

Питание модуля СТ1CPS24 осуществляется от сети постоянного тока напряжением 24В.

Питание модуля СТ1CPS220 осуществляется от сети переменного тока 220В частотой (50 ± 1) Гц.

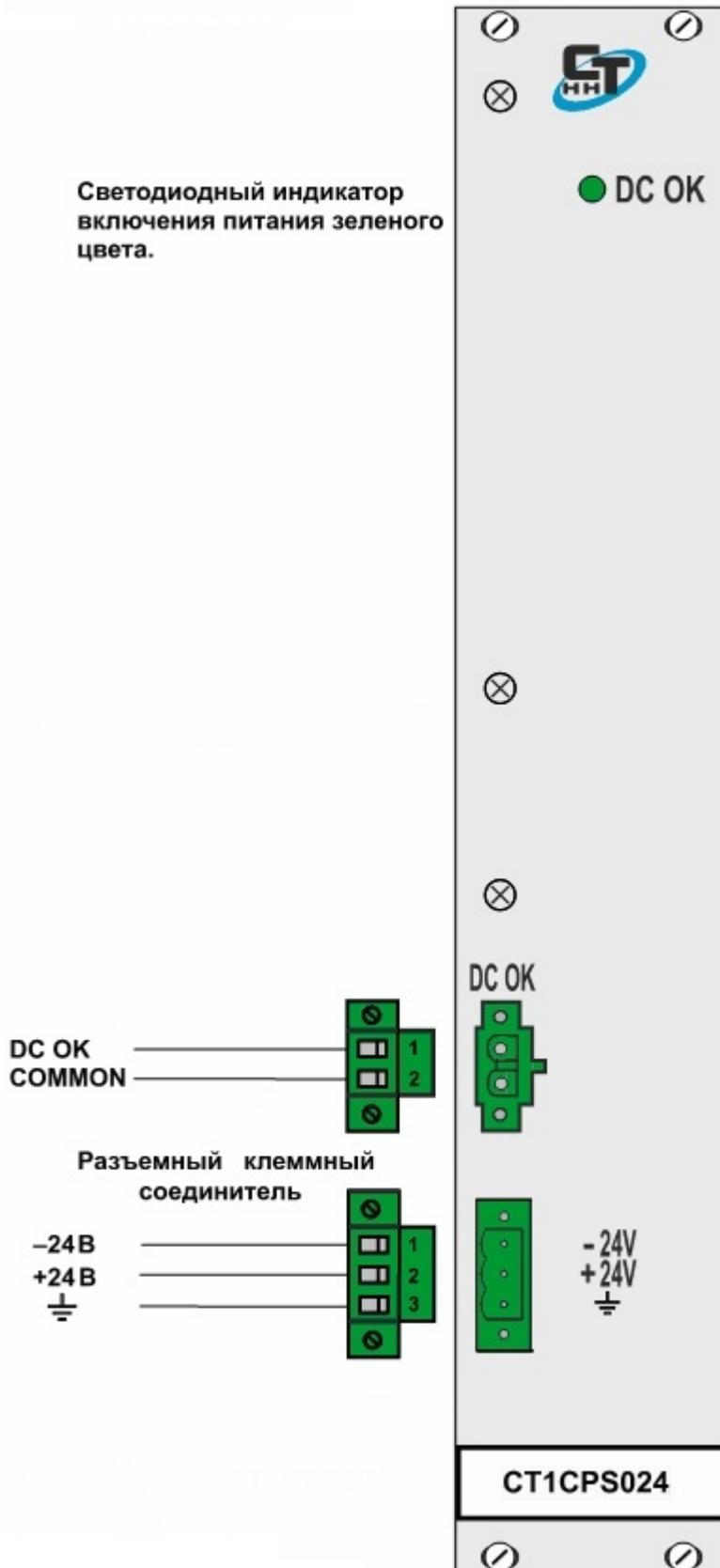
Питание модуля СТ2CPS220 осуществляется от сети переменного тока 220В частотой (50 ± 1) Гц и от внешнего аккумулятора номинальным напряжением 12В.

Мощность, потребляемая модулями питания: не более 60Вт.

На лицевой панели расположены индикатор зелёного цвета наличия выходного напряжения модуля питания (+5В), держатель предохранителя и разъемные клеммные соединители для подключения внешнего питания.

Модули питания CPS не имеют программных настроек и не занимают места в памяти процессорного модуля.

Светодиодный индикатор
включения питания зеленого
цвета.



1.2.2. Процессорные модули

Процессорный модуль ST2CPU33 — «мозг» контроллера. В нем выполняется ядро контроллера, обеспечивающее выполнение основных функций контроллера. Сюда же загружается после разработки и алгоритм рабочей программы. Процессорный модуль всегда устанавливается на позицию 2 в корзине контроллера КСА-02. Модуль ST2CPU33 устанавливается корзине главного процессора.

Процессорный модуль обеспечивает опрос модулей ввода/вывода, установленных в контроллере и обмен с внешними устройствами по трем последовательным портам: COM1, COM2, COM3.

Порт COM3 обеспечивает обмен с внешним устройством по интерфейсу RS232 и используется только для настройки контроллера и его технологических проверок.

Порт COM1 обеспечивают обмен с внешними устройствами по интерфейсу RS232. К данному порту возможно подключение модема с использованием управляющих линий CTS и RTS.

Порт COM2 обеспечивают обмен с внешними устройствами по интерфейсу RS485.

Для выполнения заложенных функций процессорный модуль имеет процессор, внешнюю оперативную память (SRAM) и внешнюю энергонезависимую память (NVRAM и Flash). Работа модуля осуществляется под управлением основного (управляющего) процессора.

Ток, потребляемый модулем по шине «+5В»: не более 0,6А.

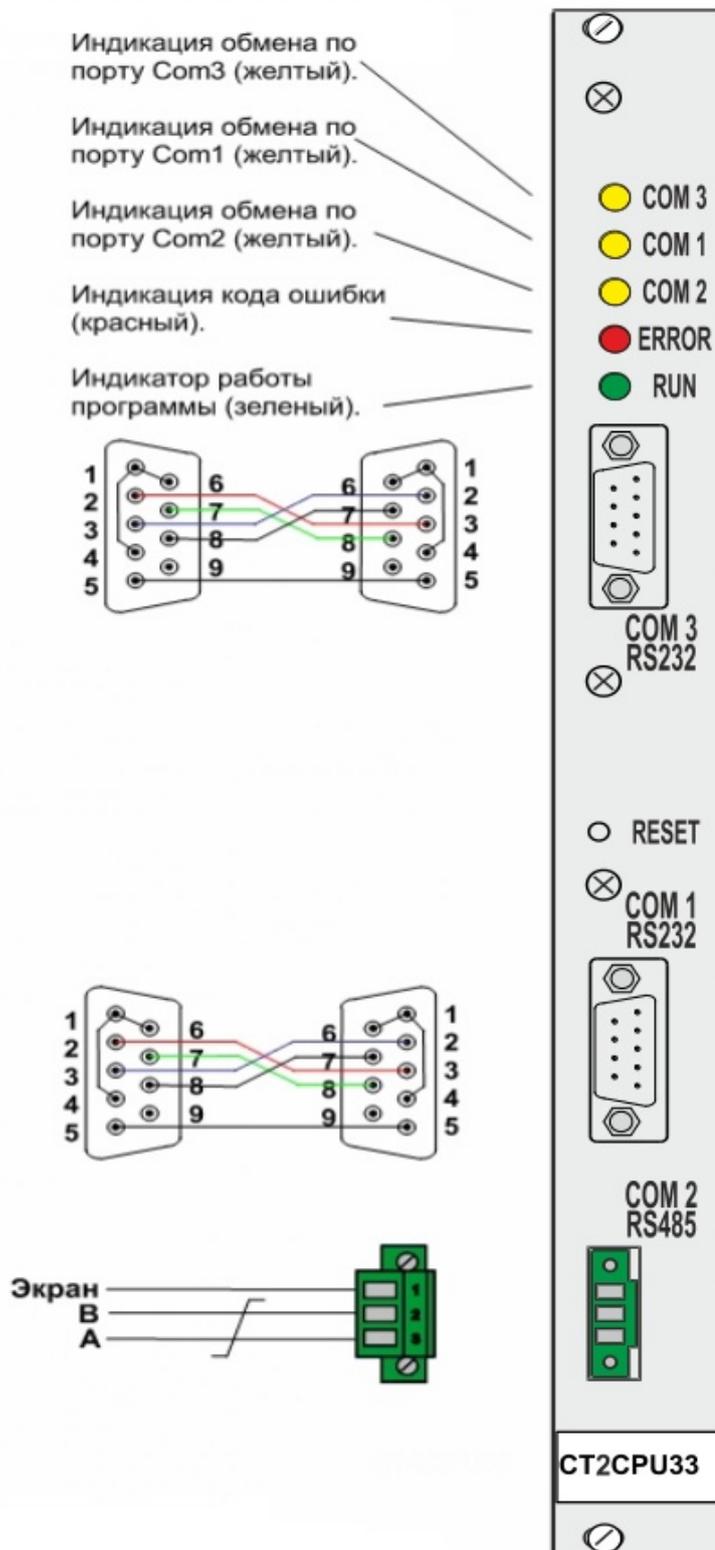
На лицевой панели модуля имеются два 9-контактных разъема для подключения последовательных портов COM1, COM3, 2-контактный разъем для подключения последовательного порта COM2, индикаторы жёлтого цвета работы последовательных портов, индикатор красного цвета состояния ошибки, индикатор зелёного цвета выполнения программы контроллера.

Также там имеется кнопка «Reset», которая останавливает/запускает контроллер (выполнение рабочей программы контроллера, включая все драйвера). Применение «Reset» аналогично перезагрузке контроллера.

Модуль не занимает слов в памяти буфера обмена.

Основные отличия модуля RPU от CPU:

- он может быть установлен только в станции удаленного ввода/вывода и его рабочим алгоритмом является программа обмена данными с главным процессором,
- последовательный порт COM1 обеспечивает обмен по интерфейсу RS485, соответственно для подключения порта используется 2-контактный разъем,
- отсутствует внешняя оперативная память SRAM,
- отсутствует внешняя энергонезависимая память NVRAM.



1.2.3. Модули ввода/вывода

Модули ввода/вывода собирают данные с нижнего уровня и передают их на обработку в процессорный модуль. Затем получают от него результаты обработки и отправляют их к исполнительным устройствам.

К модулям ввода/вывода относятся:

- модули ввода аналоговых сигналов: CT1ACI08, CT1ARI08, CT1ATI08;
- модуль вывода аналоговых сигналов: CT1ACO04;
- модули ввода дискретных сигналов: CT3DDI30, CT4DDI30, CT5DDI30, CT6DDI30, CT7DDI30, CT8DDI30, CT9DDI30, CT10DDI30, CT1DAI16;
- модуль вывода дискретных сигналов: CT2DDO30;
- модули ввода-вывода дискретных сигналов: CT1DIO29, CT2DIO29, CT3DIO29;
- модули управления кранами: CT1BCT02, CT 1BCT03.

1.2.3.1. Модули ввода аналоговых сигналов (на примере модуля CT1ACI08)

Модуль принимает и обрабатывает непрерывные аналоговые сигналы тока от 0 до 20мА или от 4 до 20мА.

Предельное значение входного сигнала: 25мА.

Входное сопротивление: 124Ом ± 0,1Ом.

Предел допустимой основной погрешности: ± 0,2%.

Датчики, подключаемые ко входам, обеспечиваются напряжением постоянного тока: 24 (+30%, -25%)В.

Количество гальванически развязанных от вторичных цепей электропитания и от корпуса входов: 8.

Ток, потребляемый по шине «+5В»: не более 1,2А.

Допустимое количество бит выходного кода: от 12 до 16.

Допустимое значение выходного кода: от 0x0000 до 0xFFFF ([подробнее](#))

Модуль имеет следующие настройки: тип входного сигнала, количество бит, ограничение, инструментальная ошибка, постоянные фильтров.

Настройки хранятся в EEPROM модуля, записываются в него при выпуске модуля из производства и могут быть изменены позже из сервисной программы контроллера КСА-02 (меню «Настройка», пункт «Конфигурация контроллера»).

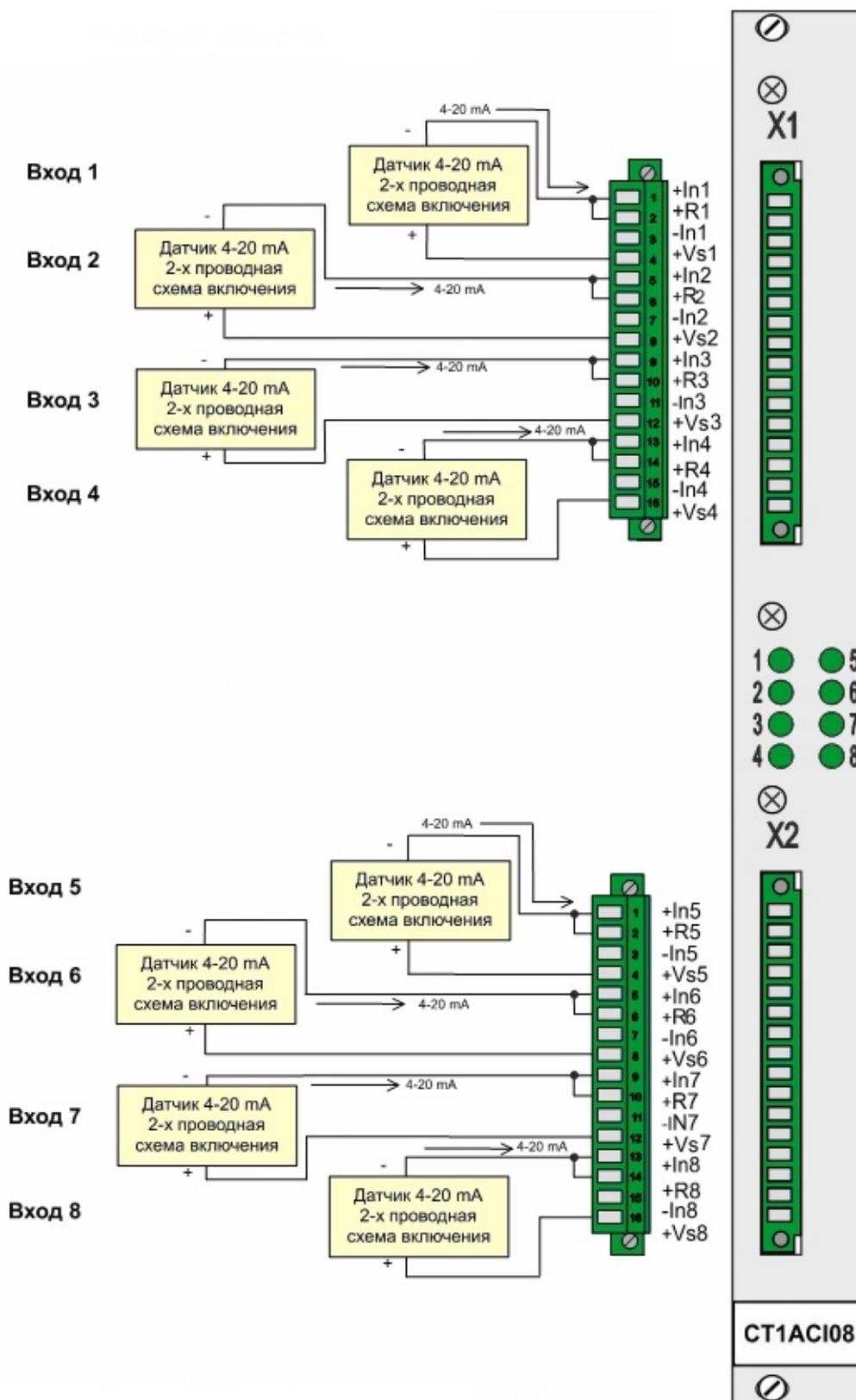
Модули АТІ и АRI, в отличие от модуля АСІ, производят измерение температуры, в то время как АСІ измеряет ток (давление, уровни жидкости, расходы и т.д.).

ARI принимает и обрабатывает сигналы с термометров сопротивления, подключенных к его входам; диапазон измерения температуры — от -50°С до +400°С.

АТІ принимает и обрабатывает сигналы с термопар, подключенных к его входам, а также температуру в месте своей установки (температура холодного спая). Диапазон измерения температуры составляет от -50°С до +1050°С.

Все модули аналогового ввода занимают по 10 слов в буфере обмена по шине и выполняют свою работу под управлением собственного процессора.

Примеры настроек модулей ввода аналоговых сигналов см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [СТ1АСІ08](#), [СТ1АRI08](#), [СТ1АТІ08](#).



1.2.3.2. Модуль вывода аналоговых сигналов (СТ1АС004)

Модуль преобразует входные коды в аналоговые сигналы постоянного тока со своих выходов.

Диапазон воспроизведения аналогового сигнала постоянного тока: от 4мА до 20мА или от 0 до 20 мА.

Предел допустимой погрешности воспроизведения аналогового сигнала постоянного тока: $\pm 0,1\%$.

Работа модуля выполняется под управлением собственного процессора.

Количество гальванически развязанных от вторичных цепей электропитания и от корпуса выходов: 4.

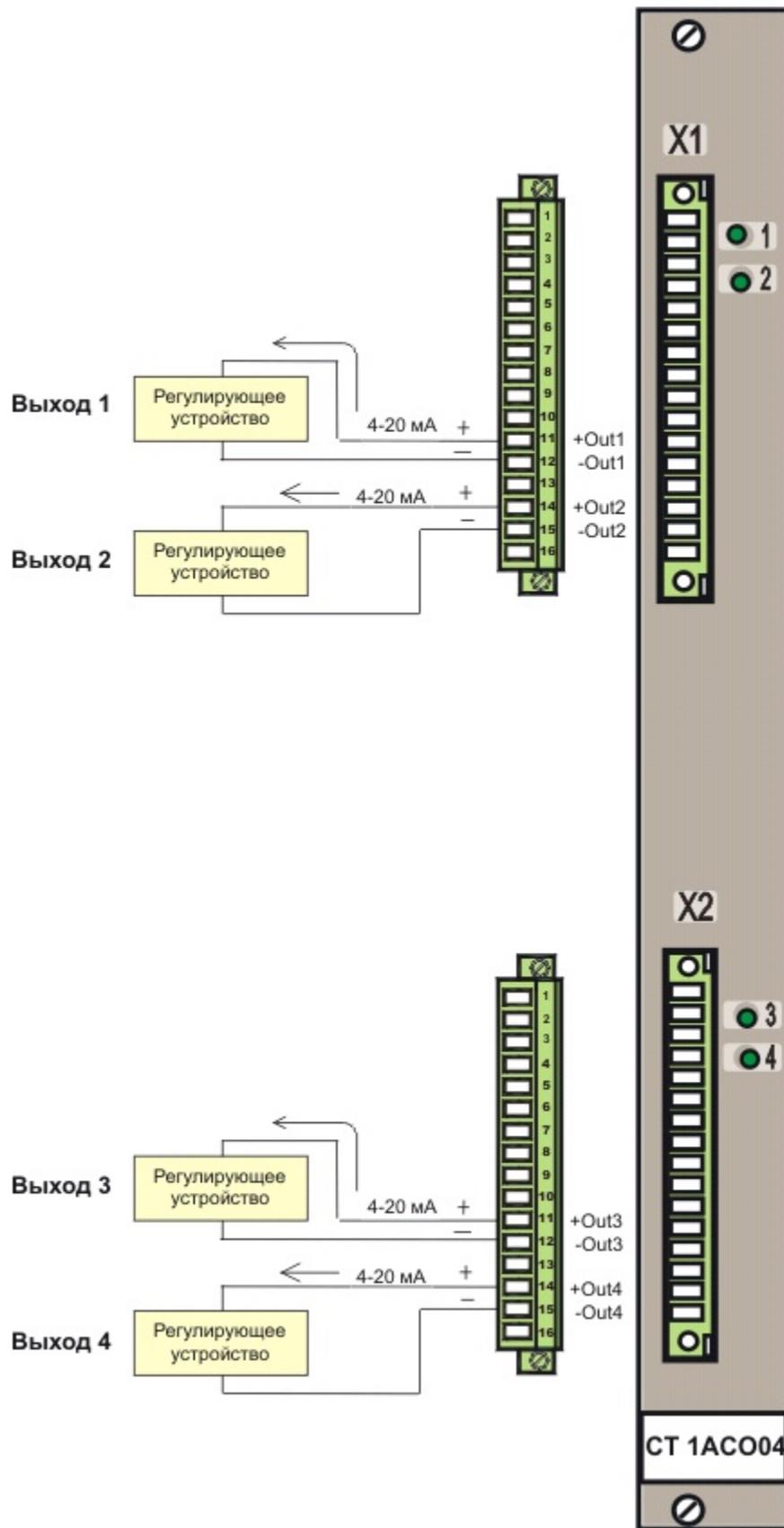
Ток потребления по шине «+5В»: не более 1,2А.

Модуль занимает 6 слов в буфере обмена с модулями ввода/вывода.

Допустимое количество бит выходного кода: от 12 до 16.

Допустимое значение выходного кода: от 0x0000 до 0xFFFF ([подробнее](#)).

Примеры настроек модулей вывода аналоговых сигналов см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [СТ1АС004](#).



1.2.3.3. Модули ввода дискретных сигналов постоянного тока.

Модули обеспечивает регистрацию, прием и обработку дискретных входных сигналов. Количество гальванически развязанных от вторичных цепей электропитания и от корпуса входов: 30. Входы объединены в две гальванически развязанные между собой группы по 15 входов в группе. Часть модулей поддерживает функцию организации подсчета импульсов на дискретном входе (до 8 каналов). Параметры сигналов приведены в таблице.

Параметр	Модификация модуля							
	СТ3DDI30	СТ4DDI30	СТ5DDI30	СТ6DDI30	СТ7DDI30	СТ8DDI30	СТ9DDI30	СТ10DDI30
Номинальный уровень сигнала, В	24		12		24		12	
Уровень логической "1", В	от 19,2 до 31,2		от 9,6 до 15,6		от 19,2 до 31,2		от 9,6 до 15,6	
Уровень логического "0", В	от 0 до 7,2		от 0 до 3,6		от 0 до 7,2		от 0 до 3,6	
Полярность общей цепи в группах входов	общий минус в группах				общий плюс в группах			
Поддержка функции счета импульсов (до 8 каналов)	есть	нет	есть	нет	есть	нет	есть	нет

Максимальный ток на каждом входе не более 10мА;

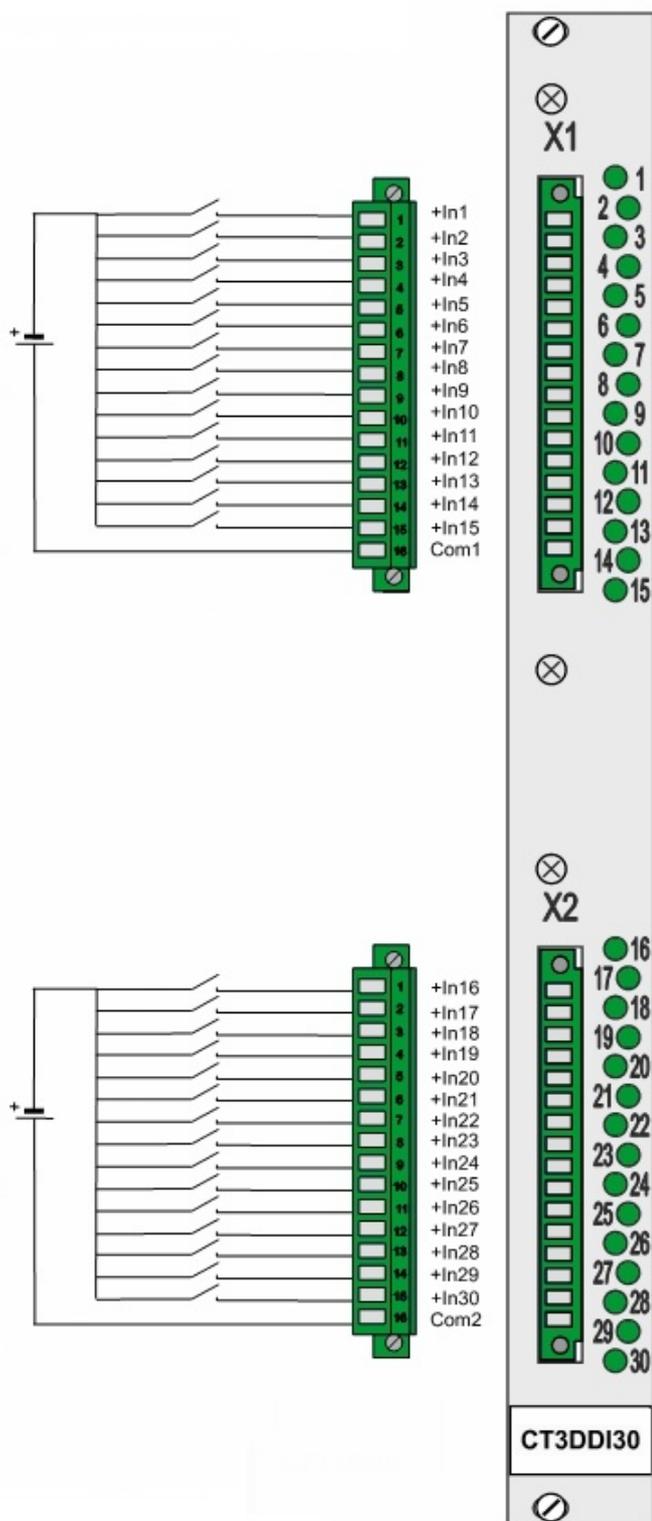
Работа модуля выполняется под управлением собственного процессора.

Каждому измерительному входу сопоставлен индикатор зелёного цвета. Зажжённое состояние индикатора говорит о наличии логической 1 на соответствующем входе модуля.

Ток потребления по шине «+5В»: не более 0,2А.

Модули с поддержкой функции счета импульсов занимают 16 слов в буфере обмена с модулями ввода/вывода. Модули без поддержки функции счета импульсов занимают 6 слов в буфере обмена с модулями ввода/вывода.

Примеры настроек модулей ввода дискретных сигналов постоянного тока см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [CT3DDI30 – CT10DDI30](#).



1.2.3.4. Модуль ввода дискретных сигналов переменного тока CT1DAI16.

Модуль CT1DAI16 не имеет индикаторов состояния входов и рассчитан на более высокие напряжения:

Уровень логической 1 — напряжение переменного тока от 187В до 242В;

Уровень логического 0 — напряжение переменного тока от 0 до 31,2В;

Максимальный ток на каждом входе: не более 20мА;

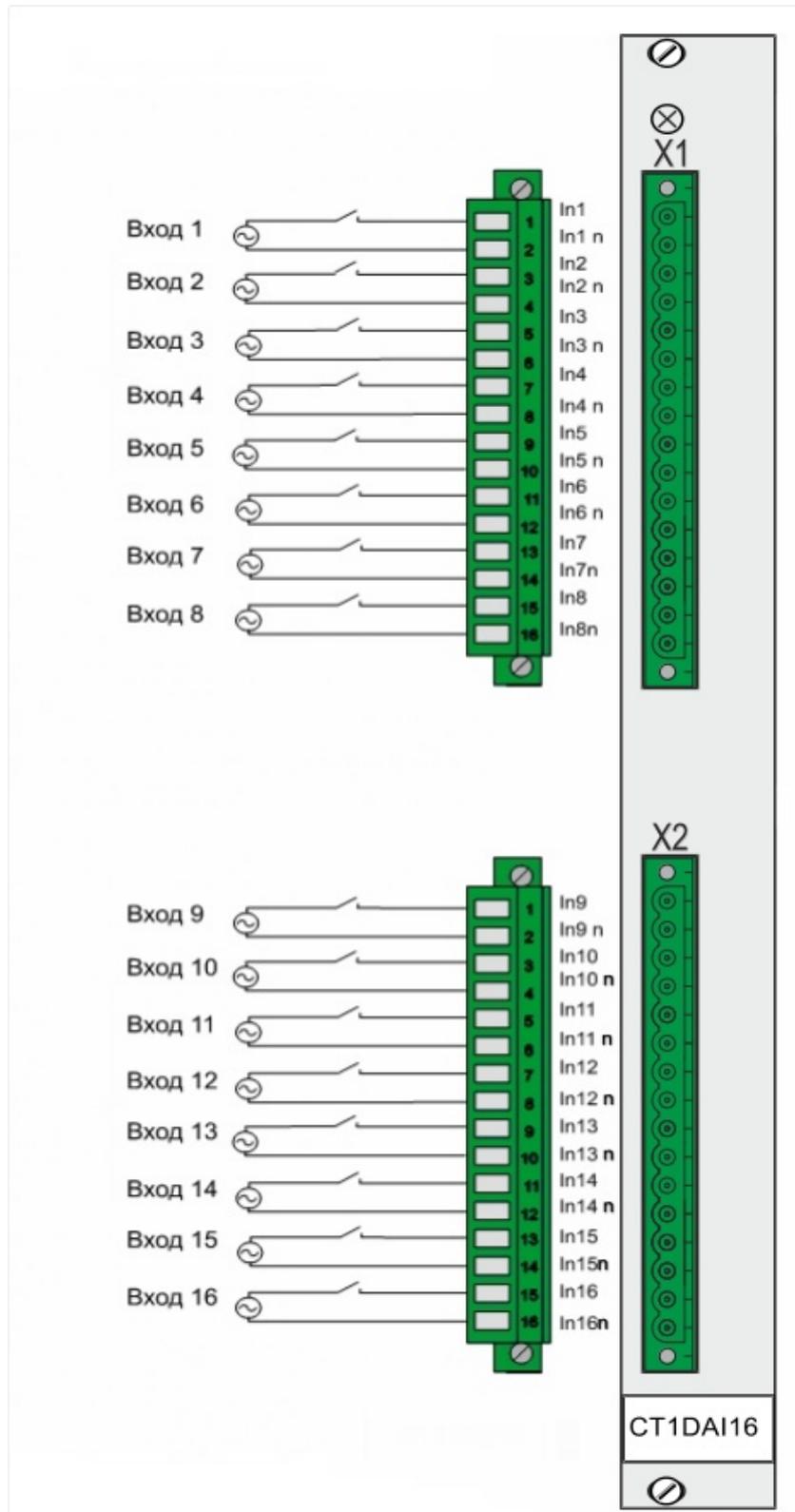
Ток потребления по шине «+5В»: не более 0,2А.

Работа модуля выполняется под управлением собственного процессора.

Количество входов у модуля CT1DAI16, гальванически развязанных между собой от вторичных цепей электропитания и от корпуса: 16.

Модуль CT1DAI16 занимает 6 слова в буфере обмена по шине.

Примеры настроек модулей ввода дискретных сигналов переменного тока см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [CT1DAI16](#).



1.2.3.5. Модуль вывода дискретных сигналов CT2DDO30

Модуль обеспечивает подключение внешней нагрузки к внешнему источнику питания постоянного тока.

Максимальное коммутируемое напряжение на каждом выходе: не менее 32В.

Максимальный коммутируемый ток на каждом выходе: не менее 100мА.

Количество гальванически развязанных от вторичных цепей электропитания и от корпуса выходов: 30. Выходы объединены в две гальванически развязанные между собой группы по 15 выходов в группе с общим минусом источника питания.

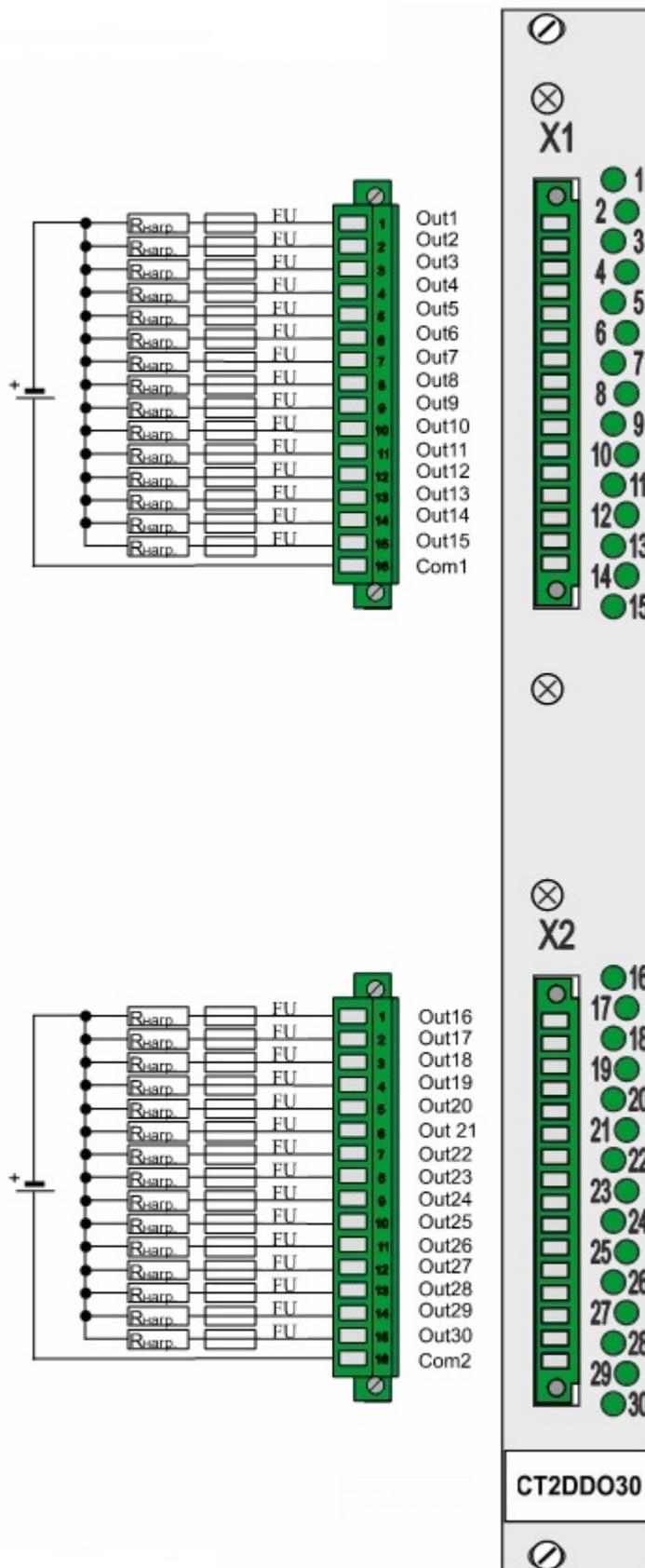
Каждому выходу поставлен в соответствие индикатор зеленого цвета. Зажженное состояние индикатора свидетельствует о состоянии «замкнуто» на соответствующем выходе модуля.

Ток потребления по шине «+5В»: не более 0,3А.

Работа модуля выполняется под управлением собственного процессора.

Модуль занимает 6 слов в буфере обмена по шине.

Примеры настроек модулей вывода дискретных сигналов постоянного тока см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [CT2DDO30](#).



1.2.3.6. Особенности модулей ввода/вывода дискретных сигналов

Модули обеспечивает регистрацию, прием и обработку дискретных входных сигналов. Количество гальванически развязанных от вторичных цепей электропитания и от корпуса входов: 24. Входы объединены в две гальванически развязанные между собой группы по 15 и 9 входов в группе. Модули поддерживает функцию организации подсчета импульсов на дискретном входе (до 8 каналов). Параметры входных сигналов приведены в таблице.

Параметр	Модификация модуля		
	СТ1DIO29	СТ2DIO29	СТ3DIO29
Номинальный уровень сигнала, В	12		24
Уровень логической "1", В	от 9,6 до 15,6		от 19,2 до 31,2
Уровень логического "0", В	от 0 до 3,6		от 0 до 7,2
Полярность общей цепи в группе входов	минус	плюс	минус
Поддержка функции счета импульсов (до 8 каналов)	есть		

Максимальный ток на каждом входе не более 10мА.

Каждому измерительному входу сопоставлен индикатор зелёного цвета. Зажжённое состояние индикатора говорит о наличии логической 1 на соответствующем входе модуля.

Модуль обеспечивает подключение внешней нагрузки к внешнему источнику питания постоянного тока. Параметры выходного сигнала полностью соответствуют параметрам для модуля СТ2DDO30.

Количество гальванически развязанных от вторичных цепей электропитания и от корпуса выходов: 5.

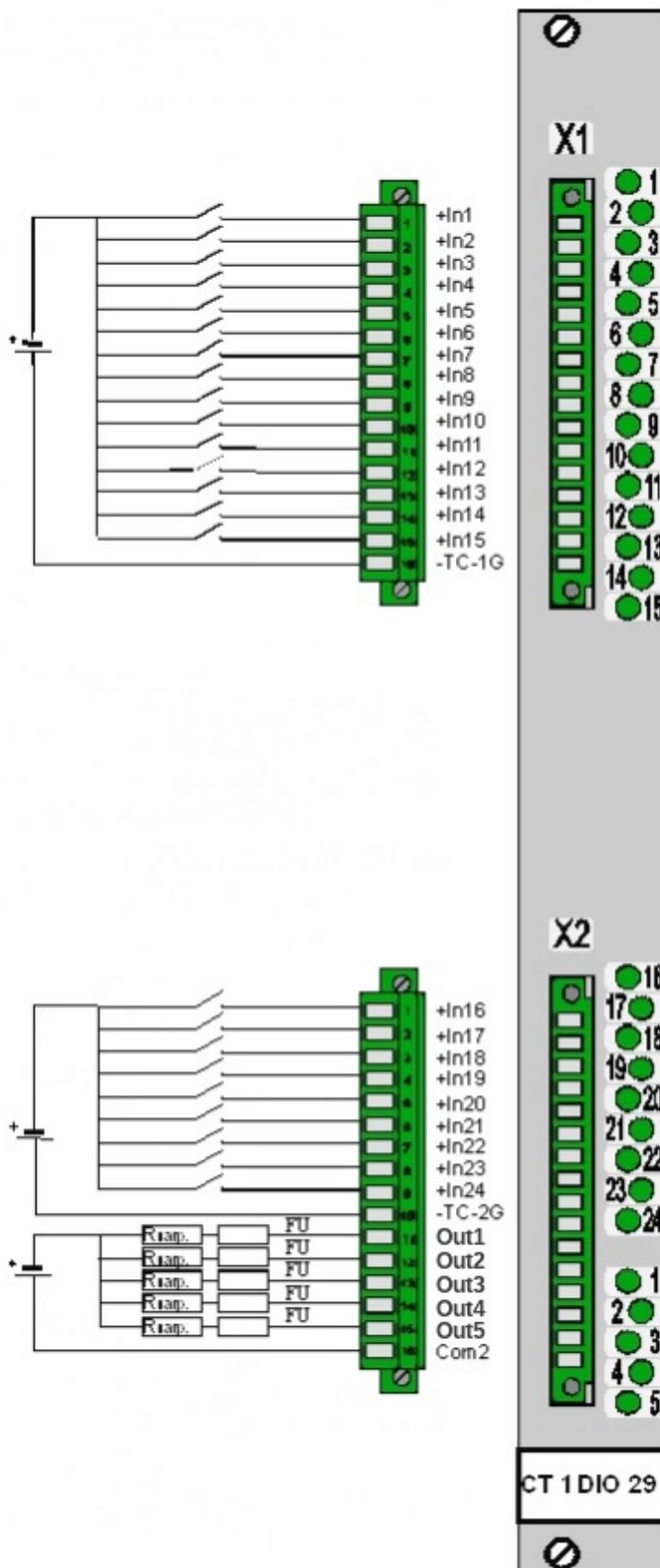
Каждому выходу поставлен в соответствие индикатор зеленого цвета. Зажженное состояние индикатора свидетельствует о состоянии «замкнуто» на соответствующем выходе модуля.

Ток потребления по шине «+5В»: не более 0,2А.

Работа модуля выполняется под управлением собственного процессора.

Модуль занимает 19 слов в буфере обмена по шине.

Примеры настроек модулей ввода/вывода дискретных сигналов постоянного тока см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [СТ1DIO29-СТ3DIO29](#).



1.2.3.7. Модуль управления кранами СТ1ВСТ02 (СТ1ВСТ03)

Модуль управления кранами имеет обозначение:

СТ1ВСТ02: 2 крана с 3х-соленоидной схемой;

СТ1ВСТ03: 3 крана с 2х-соленоидной схемой.

Модули управления кранами обеспечивают управление внешними устройствами.

Модули анализируют по встроенному алгоритму входные сигналы и на основании результата анализа передают на внешнее устройство управляющие сигналы. Модули также выполняют команды контроллера, с помощью выходных реле.

Модуль может быть установлен на любой слот в корзине контроллера, кроме 1 и 2.

Работа модуля управляется собственным процессором.

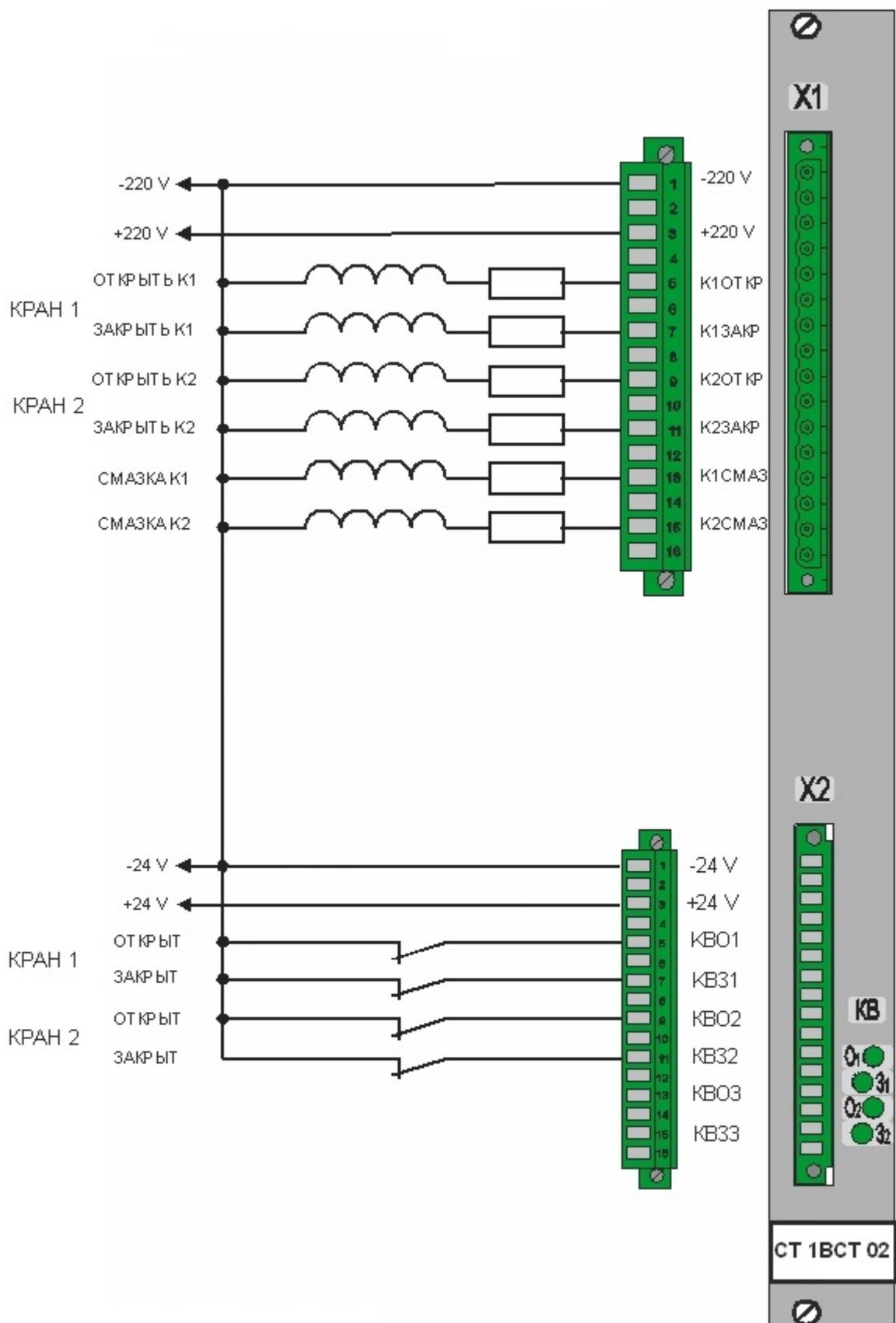
Ток потребления по шине «+5В»: не более 0,1А.

Максимально коммутируемое напряжение постоянного тока: 260В.

Максимальный постоянный ток: 0,5А.

Модули ВСТ занимают по 8 слов в буфере обмена по шине.

Пример настроек модуля управления кранами см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: [СТ1ВСТ02 \(СТ1ВСТ03\)](#).



1.2.4. Коммуникационные модули

Коммуникационные модули нужны для связи и обмена данными по интерфейсным линиям связи. Они устанавливаются только в главном процессоре и могут поддерживать связь с дублирующим контроллером (если в системе предусмотрено резервирование главных процессоров) или станциями удалённого ввода-вывода (устройства, также содержащие модули ввода/вывода, но не выполняющие алгоритм управления). Также возможна связь и с любыми другими устройствами, работающими по стандартным протоколам, поддерживаемым контроллером КСА-02.

Коммуникационные модули не занимают слов в области памяти ввода-вывода.

Коммуникационные модули делятся на специальные и интерфейсные.

Специальные модули служат для связи и обмена данными между ведущим и ведомым главными процессорами, между главным процессором и станциями удаленного ввода-вывода.

Интерфейсные модули служат для связи и обмена данными между устройствами сети и главным процессором. Модули СТ2CPU33 и СТ1RPU33 тоже осуществляют подобную связь, но к ним не относятся.

Специальные модули:

- СТ1HSB10 (СТ2HSB10) — модуль горячего резервирования, служит для поддержания функции горячего резервирования между собой ведущего и ведомого главных процессоров по сети Ethernet;
- СТ1RNA33 — служит для связи главного процессора со станциями удаленного ввода-вывода;

Интерфейсные модули:

- СТ1CPM10 — соединяет главный процессор с другими устройствами по сети RS-485 с использованием сетевых протоколов;
- СТ1CPE10 — соединяет главный процессор с другими устройствами по сети Ethernet.

1.2.4.1. Модуль горячего резервирования СТ1HSB10 (СТ2HSB10)

Модули СТ1HSB10 и СТ2HSB10 в составе главных процессоров контроллера КСА-02 обеспечивают обмен информацией двух главных процессоров в режиме горячего резервирования.

В одном из главных процессоров устанавливается модуль СТ1HSB10, в другом — СТ2HSB10.

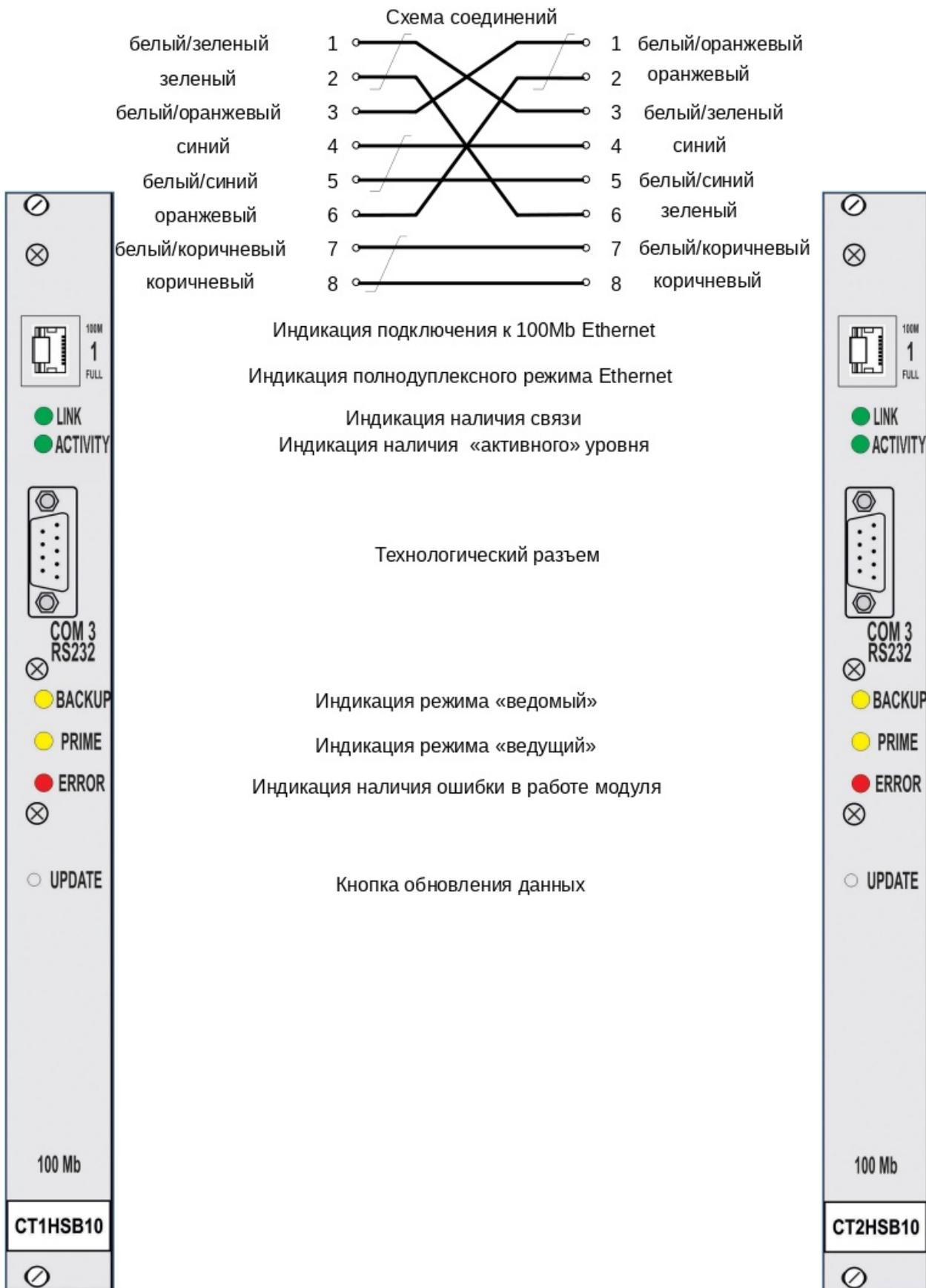
Связь модуля горячего резервирования с процессорным модулем СТ2CPU33 осуществляется по внутренней шине контроллера КСА-02.

Работа модуля осуществляется под управлением собственного процессора.

Ток, потребляемый модулем по шине «+5В»: не более 0,6А.

Модуль может устанавливаться на любое место в корзине контроллера, кроме 1 и 2.

Настройками модуля HSB является конфигурация системы резервирования, подробнее о настройках параметров резервирования можно прочитать в разделе «Редактор параметров контроллера КСА-02», гл. 4.2.2.6. [«Резервирование»](#).



1.2.4.2. Модуль удаленного ввода СТ1RNA33

Модуль связи со станцией удаленного ввода-вывода. Модуль устанавливается в главном процессоре контроллера КСА-02 и управляет сетью из станций удаленного ввода-вывода. Он обеспечивает опрос СУВД по одному или двум последовательным портам СОМ1, СОМ2 по интерфейсу RS-485 на скорости 512К. При использовании двух портов СОМ1 и СОМ2 осуществляется резервирование линии связи с СУВД.

Связь модуля RNA с процессорным модулем СТ2CPU33 осуществляется по внутренней шине контроллера.

Работа модуля осуществляется под управлением собственного процессора.

Ток, потребляемый модулем по шине «+5В»: не более 0,6А.

Модуль может устанавливаться на любой слот в корзине контроллера, кроме 1 и 2.

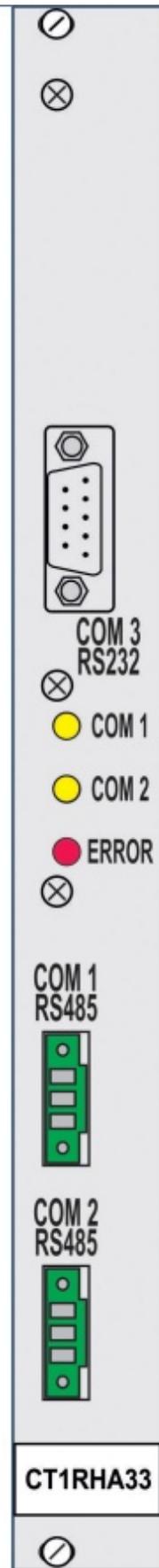
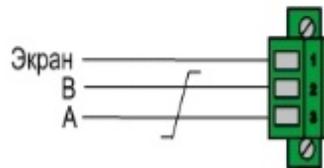
Настройками модуля СТ1RNA33 является конфигурация удалённого ввода-вывода, подробнее см. в разделе «Редактор параметров контроллера КСА-02», гл. 4.2.2.7. [«Удалённый ввод/вывод»](#).

Технологический разъем

Индикация обмена по порту COM1

Индикация обмена по порту COM2

Индикация кода ошибки



1.2.4.3. Интерфейсный модуль СТ1СРМ10

Модуль СТ1СРМ10 обеспечивает связь главного процессора контроллера КСА-02 с другими устройствами по одному или двум портам COM1, COM2 (интерфейс RS-485).

Связь модуля связи с процессорным модулем осуществляется по внутренней шине контроллера.

Модуль устанавливается только в главном процессоре КСА-02, в станциях удаленного ввода-вывода он отсутствует.

Работа модуля осуществляется под управлением собственного процессора.

Ток, потребляемый модулем по шине «+5В»: не более 0,6А.

Модуль может устанавливаться на любой слот в корзине контроллера, кроме 1 и 2.

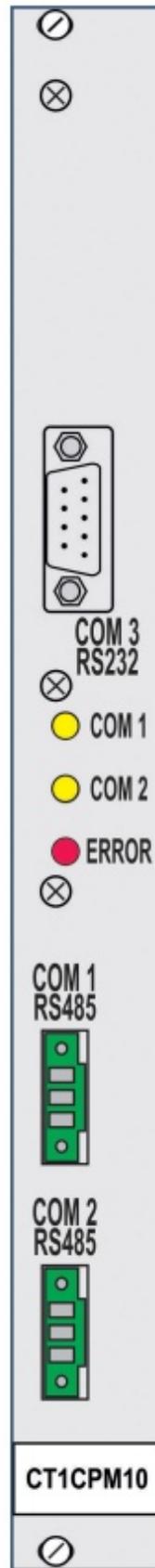
Настройками модуля СРМ являются настройки протоколов, по которым он работает. Подробности см. в соответствующей главе части «Редактор параметров контроллера КСА-02»: 4.2.2.8. [«Модули СТ1СРМ»](#).

Технологический разъем

Индикация обмена по порту COM1

Индикация обмена по порту COM2

Индикация кода ошибки



1.2.4.4. Интерфейсный модуль СТ1СРЕ10

Модуль СТ1СРЕ10 обеспечивает связь главного процессора контроллера КСА-02 с другими устройствами по сети Ethernet.

Обмен данными модуля СРЕ с процессорным модулем CPU осуществляется по внутренней шине контроллера.

Работа модуля осуществляется под управлением собственного процессора.

Ток, потребляемый модулем по шине «+5В»: не более 0,6А.

Модуль может устанавливаться на любой слот в главном процессоре контроллера, кроме 1 и 2. При установке модуля в слоте 3 модуль может поддерживать ограниченную функциональность даже при отсутствии алгоритма или остановке ядра контроллера (так называемая "системная" установка модуля). Кроме того, некоторые протоколы поддерживаются только при установке модуля в слоте 3.

Настройками модуля СРЕ являются настройки протоколов, по которым он работает. Подробности см. в соответствующей главе части «Сервисная программа контроллера КСА-02»: 4.2.2.8. [«Модули СТ1СРN»](#).

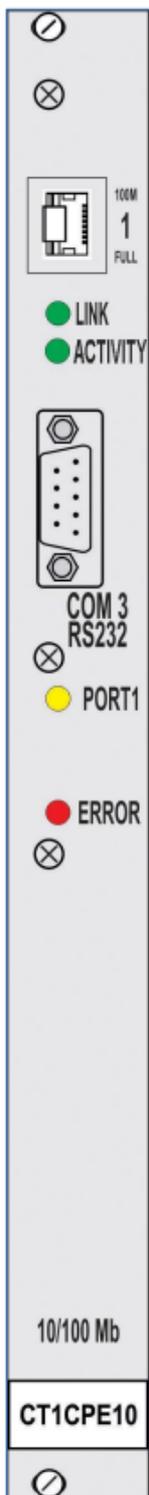
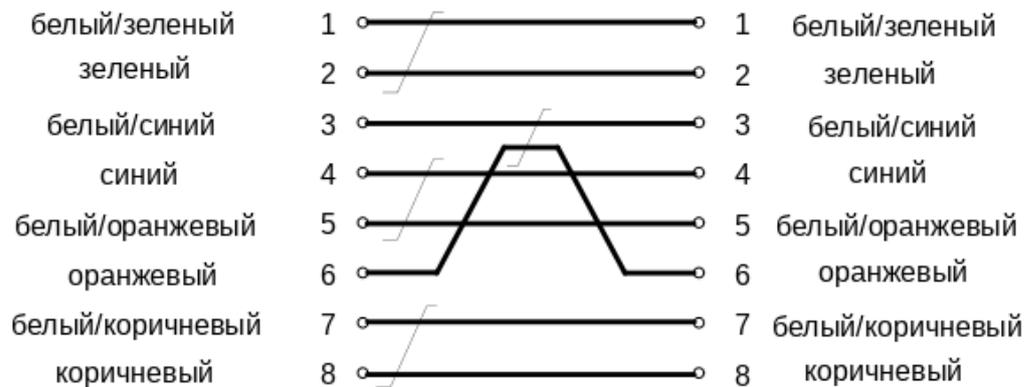


Схема соединений



Индикация подключения к 100Mb Ethernet

Индикация полнодуплексного режима Ethernet

Индикация наличия связи

Индикация наличия «активного» уровня

Технологический разъем

Индикация обмена

Индикация наличия ошибки в работе модуля

1.3. Структура программного обеспечения контроллера КСА-02

Программное обеспечение контроллера представляет собой среду исполнения системы «Scorpio» и состоит из трех основных частей — это загрузчик, ядро и алгоритм рабочей программы, а также рабочие программы коммуникационных модулей и модулей ввода/вывода.

Загрузчик - это программа, которая при включении или перезагрузке модуля центрального процессора осуществляет перегрузку ядра в оперативную программную память модуля. Во время работы загрузчик обеспечивает обмен через технологический порт COM3, а также (по команде ядра) осуществляет физическое управление светоиндикаторами на передней панели процессорного модуля.

Ядро — это программа, загружаемая из Flash памяти в оперативную программную память внутри модуля центрального процессора ST2CPU33. Ядро выполняет роль «операционной системы» контроллера. Все процессы, происходящие в контроллере КСА-02, управляются программой ядра. После включения контроллера сразу начинается выполнение этой программы. Она выполняет самодиагностику, осуществляет обмен данными по внутренним и внешним интерфейсам связи и т.д.

Разработка программы ядра ведется только на предприятии-изготовителе. Поставка программы ядра производится в комплекте с аппаратной частью контроллера.

В программе ядра организован цикл, который выполняется со строго заданным интервалом времени. Именно в этом цикле происходит выполнение [алгоритма](#).

Алгоритм рабочей программы — это «приложение», которое должен выполнять контроллер. Алгоритм представляет собой массив данных, записанный в энергонезависимую память процессорного модуля. Ядро циклически обращается к этому массиву и выполняет закодированные в нем действия.

Создание, редактирование, загрузка в контроллер и отладка алгоритма выполняются с помощью средств разработки системы «Scorpio».

Взаимодействие алгоритма с внешними устройствами (модули ввода/вывода контроллера КСА-02, сеть СУВД, SCADA -системы и т.п.) осуществляется через входные и выходные буферы данных, а также область данных модулей ввода/вывода, в т.ч. модулей СУВД.

Цикл выполнения алгоритма схематически выглядит следующим образом:

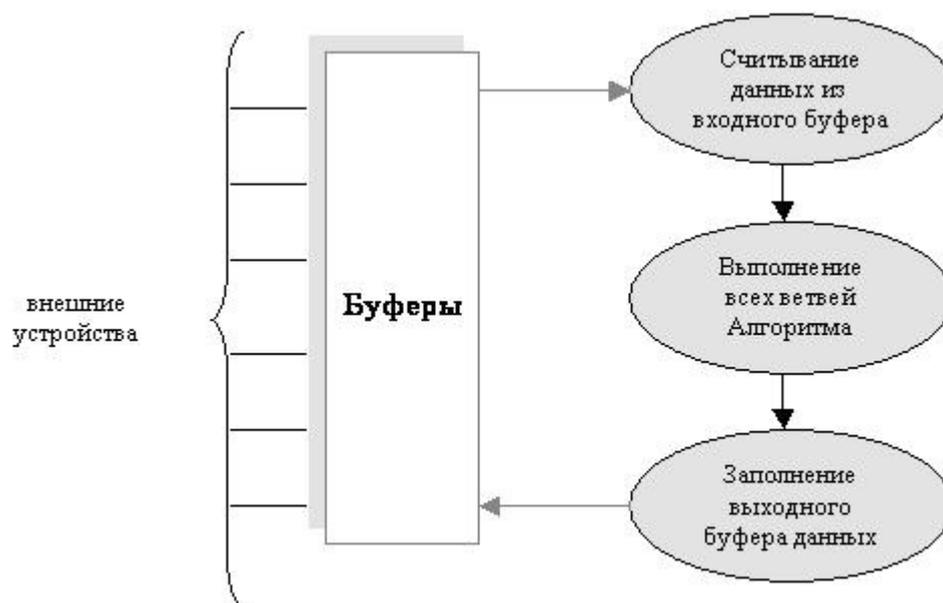


Рис. 1.3.1. Цикл выполнения алгоритма

Алгоритм управления технологическим процессом или объектом может быть как единым целым, так и состоять из множества отдельных взаимосвязанных между собой или самостоятельных ветвей, каждая из которых описывает логику работы и управления отдельного технологического устройства или части технологического процесса.

Выполнение алгоритма — повторяющийся цикл, имеющий следующие особенности:

- все функции, входящие в алгоритм, выполняются последовательно, одна за другой;
- время выполнения алгоритма (или всех его ветвей) — 50 мс (далее по тексту — цикл контроллера).

Если для выполнения всех функций, содержащихся в алгоритме, затрачивается время меньше 50 мс, то следующий цикл выполнения алгоритма все равно начнется через 50 мс.

При превышении времени выполнения алгоритма 50 мс контроллер фиксирует ошибку, останавливает ядро и переходит в состояние ошибки.

Время считывания данных из входного буфера и время заполнения выходного буфера может сильно изменяться в зависимости от того, включена система резервирования или нет, и от размера области для копирования системы резервирования. В этом случае у главного процессора в режиме ведомого время выполнения будет больше (приблизительно на 5 мс. при области копирования в 8000 слов) это необходимо учитывать при проектировании алгоритма.

2. Основные концепции, понятия, определения

2.1. Структура системы «Scorpio»

Все компоненты системы «Scorpio» по своему функциональному назначению можно условно разделить на три тесно взаимодействующие подсистемы:

Подсистема выполнения: эта подсистема вся целиком находится в контроллере: в процессорном модуле СТ2CPU33, специальных модулях и модулях ввода/вывода. Сюда относятся ядро и алгоритм рабочей программы контроллера КСА-02, выполняемые в процессорном модуле и управляющие программы процессоров остальных модулей контроллера.

Подсистема разработки алгоритма: эта подсистема функционирует непосредственно на ПЭВМ. Сюда относятся такие части системы «Scorpio» как программа «FBD_Builder» и [компилятор](#). Их посредством можно:

- настраивать параметры контроллера КСА-02, отвечающие за обмен данными;
- создавать, редактировать и компилировать алгоритм, созданный на языке FBD с использованием функций системы «Scorpio».

Подсистема отладки алгоритма: эта подсистема образуется путем взаимодействия ядра контроллера и сервисной программы. Она позволяет управлять работой контроллера, если он подключен к ПЭВМ. Подсистема отладки позволяет:

- просматривать значения, передаваемые модулями ввода/вывода по системной шине, задавать эти значения вручную, блокируя действительные значения;
- просматривать значения на входах функций в режиме реального времени (при выполняемом алгоритме), а также задавать эти значения вручную, блокируя действительные значения;
- останавливать или запускать выполнение алгоритма, загруженного в контроллер, например, для выполнения отладки.

2.1.1. Базовая директория

Все компоненты подсистем разработки и отладки алгоритма, хранятся в базовой директории, т.е. в отдельной папке на локальном ресурсе компьютера. Например:

C:\Program Files\Scorpio_5

Основные исполняемые файлы, расположенные в базовой директории:

- FBD_Builder — графический построитель;
- ServiceProgramm — оболочка сервисной программы контроллера KCA-02;
- TestProgramm — оболочка тестовой программы контроллера KCA-02A;
- KSA_Drv — драйвер обмена для оболочки и программных модулей сервисной и тестовой программ;
- compiler — компилятор алгоритма;
- scorp_cfg — редактор параметров алгоритма.

Также базовая директория содержит и другие исполняемые файлы, которые не обязательны при разработке или отладке алгоритма, но могут облегчить решение некоторых частных вопросов:

- Funcbase — редактор библиотеки функций;
- L1Edit — формирователь блока уставок;
- C1Reader, C2Reader — программы чтения файлов конфигурации контроллера.

В базовой директории содержатся несколько поддиректорий. Наиболее важные из них следующие:

- SP_modules - содержит исполняемые программные модули сервисной программы;
- SP_config - содержит конфигурационные файлы сервисной и тестовой программ.

2.1.2. Директория проекта

Совокупность файлов, необходимых для создания и отладки алгоритма рабочей программы, называется *проект*. Состав файлов проекта:

Имя файла	Пояснение	Примечание
*.fsc	Основной файл проекта.	Создаётся и изменяется пользователем с помощью графического построителя.
*.ssc	Файл составной функции.	Создаётся и изменяется пользователем с помощью графического построителя; может отсутствовать в проекте.

Имя файла	Пояснение	Примечание
*.sc	Файл алгоритма.	Создаётся компилятором «Scorpio», загружается в контроллер с помощью сервисной программы.
*.lum	Файл, содержащий информацию об очередности выполнения функций алгоритма.	Создаётся компилятором «Scorpio».
*.err	Файл, содержащий информацию о предупреждениях и ошибках в выявленных в процессе компиляции (или об их отсутствии).	Создаётся компилятором «Scorpio».

Помимо этого программа может создавать в папке проекта и некоторые другие файлы. Они используются программой для служебных целей и пользователю нет необходимости обращаться к ним.

Процесс разработки алгоритма начинается с создания файла *.fsc, и (при необходимости) файлов *.ssc. Остальные файлы появляются в проекте при выполнении компиляции и отладки алгоритма.

2.1.3. Типы данных

Для алгоритма определены три типа данных:

LOGIC (логические числа)	данные, значение которых может быть представлено двумя состояниями, (например, «есть сигнал — нет сигнала», или «0, либо 1»);
INTEGER (целые числа)	данные, представляющие собой числа от -32767 до +32767.
FLOAT (числа с плавающей запятой)	данные, представляющие собой числа с плавающей запятой одинарной точности формата IEEE 754, могут принимать числовые значения от $\pm 1.18E-38$ до

	$\pm 3.4E+38$, а также 0.
--	----------------------------

Буферы контроллера — это своего рода общая область памяти, доступная для алгоритма. Входящие в контроллер данные ядро записывает в так называемый **входной буфер**. Данные, предназначенные для отправки из контроллера, ядро помещает в **выходной буфер**. Существует также ещё и **внутренний буфер** данных, где хранятся служебные данные, нужные для работы алгоритма и для обмена данными по протоколам, где контроллер выступает как «Мастер» устройство. На базе входного и внутреннего буфера также основана синхронизация данных в системе резервирования.

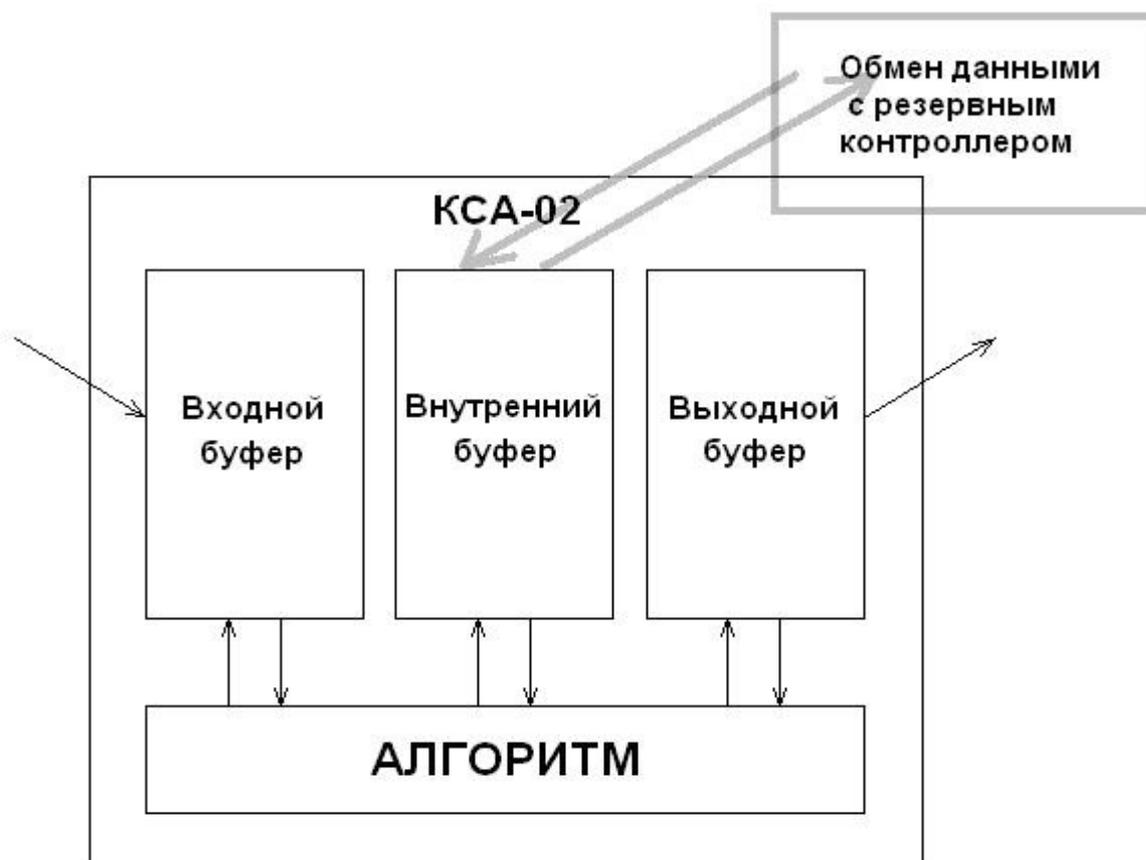


Рис. 2.1.3.1. Схема обмена данными между алгоритмом, буферами и резервным контроллером.

Входной и выходной буферы могут быть настроены для хранения трех различных типов данных. Поэтому внутри входного и выходного буферов существует ещё следующее деление:

- Буфер логических данных (LOGIC);

- Буфер целых чисел (INTEGER);
- Буфер чисел с плавающей запятой (FLOAT).

Внутренний буфер предназначен для работы только с целыми числами (INTEGER).

Размер каждого буфера задаётся на этапе создания алгоритма и в тех единицах, в которых он измеряется.

Каждый тип буфера имеет свою размерность. Соответствие типа буфера и его размерности приведены в таблице:

Тип буфера	Размерность
LOGIC	1 бит
INTEGER	16 бит
FLOAT	32 бит

Это означает, что при определении размера буфера определённого типа следует учитывать особенности его размерности. Так, одно 16-битное слово может вмещать 16 единиц LOGIC, 1 единицу INTEGER или 0,5 единицы FLOAT. Суммарный размер всех буферов не может превышать 8192 16-битных слов.

Размер логических буферов и буферов чисел с плавающей запятой учитывается в пересчете на целые (16-битные) слова с округлением в большую сторону.

Замечание. При использовании протокола ГОСТ Р МЭК 870-5 (IEC60870) (далее по тексту МЭК870) обеспечивается автоматическое конвертирование типов данных протокола МЭК870 к типам данных контроллера КСА-02 и обратно. Более подробно см. описание настройки протокола МЭК870.

2.1.4. Протоколы передачи данных

Поскольку контроллер КСА-02 — это сетевое устройство, работа которого имеет смысл лишь при взаимодействии с другими устройствами, то между ними существует информационная сеть. Настройки этой сети задаются протоколами передачи информации.

Протокол передачи информации — набор правил, позволяющий осуществлять соединение и обмен данными между двумя или более включёнными в сеть устройствами.

При сетевом обмене данными между контроллером и внешними устройствами могут использоваться следующие протоколы:

- Modbus (Master, Slave, ELAM Slave),
- Modbus TCP (Master, Slave),
- MKSA-Master,
- Proton3-Slave,
- Proton3-IP, Proton3-GSM-Client,
- RTM64,
- МЭК870-5 (IEC60870).

Характеристики и описание каждого из них см. в [разделе 6 «Протоколы»](#).

Протоколы, которые будут использоваться в работе информационной сети на производственном объекте, определяются и настраиваются на этапе разработки алгоритма.

2.1.5. Протоколы синхронизации времени

Синхронизация времени в контроллере осуществляется с сервера верхнего уровня при использовании протокола Proton3 любой модификации.

Начиная с версии 5.0.5 ядра контроллера и версии 6.0.0 программного обеспечения модуля ST1CRE10 обеспечивается поддержка синхронизации времени от сервера точного времени по протоколу NTP.

3. Язык функциональных блоковых диаграмм (FBD)

3.1. Основные элементы языка программирования FBD

3.1.1. Диаграмма

В системе «Scorpio» Алгоритм представляет собой [диаграмму](#), на которой условно показана логическая взаимосвязь входных, выходных и внутренних данных. Диаграмма состоит из прямоугольников — «чёрных ящиков», между которыми проведены «линии связи».

Такое представление алгоритма позволяет пользователю легко проследить логические цепочки (ветви) алгоритма, а также ввести в него изменения, если это необходимо. Процесс программирования превращается в простое «рисование» диаграммы на экране компьютера.

Все, что изображено на диаграмме алгоритма, контроллер КСА-02 должен выполнить за один цикл.

Каждый «чёрный ящик» на диаграмме — это специальная [функция](#) ядра (или совокупность функций). Линии связи между ними изображают передачу данных от одной функции к другой. Каждая функция выполняет какой-то определённый вид обработки данных. Говоря техническим языком, функция по своему устройству — подпрограмма, выполняющая различные арифметико-логические операции.

3.1.2. Функция

Функция — это основной объект алгоритма. Любой алгоритм должен содержать хотя бы одну функцию.

Большинство функций выполняют какую-либо арифметико-логическую операцию над входными данными, а результат выполнения этой операции является для них выходными данными. Также существуют функции работы с устройствами (чтение и запись данных от модулей ввода-вывода, обмен по интерфейсам, показания часов и пр.).

Соответственно каждая функция имеет **входы** («аргументы функции») и **выходы** («результаты функции»). В алгоритме выходы одних функций связаны с входами других, и таким образом осуществляется сложная обработка данных всего алгоритма.

Входными данными для функций могут быть: событие в системе, время, значения внутри контроллера (например, ячейки внутреннего буфера), а также [промежуточные данные функции](#).

Наряду с входными данными для функции могут быть определены **атрибуты**. С помощью атрибутов инициализируются некоторые константы, используемые функцией.

Функция может менять значение промежуточных данных в процессе работы, а может использовать их в качестве начальной константы. Имеется возможность задавать начальные значения некоторых промежуточных данных из графического построителя. Такие промежуточные данные называются [атрибутами](#).

После компиляции проекта каждой функции присваивается **порядковый номер**; он указывает, какой по счёту в алгоритме эта функция будет выполняться.

Присвоенный номер действителен только до следующей компиляции; после редактирования алгоритма и его перекомпилирования порядковые номера функций могут быть изменены на другие.

Каждая функция имеет **имя**. Имя функции начинается с латинской буквы F, далее следуют четыре цифры и суффикс (может отсутствовать), состоящий из двух цифр и обозначающий, что данная функция представлена в программе в нескольких вариациях, например, с разным количеством входов, выходов и т.д.

Например: **F0004** — без суффикса; **F0027-05** — с суффиксом

3.1.3. Составная функция

Повторяющиеся фрагменты алгоритма (либо не повторяющиеся, но представляющие собой логически целостные вычислительные блоки) могут оформляться в виде так называемых [составных функций](#). Некоторое количество функций объединяется в одну составную и заключается в единый блок.

Каждая составная функция в свою очередь может иметь в своём составе другие составные функции, сама входить в состав другой функции, и таким образом глубина вложения может достигать 32 уровней.

При редактировании и создании составных функций необходимо избегать ситуации, когда две функции одновременно попадают в состав друг друга (в таком случае компилятор сообщит об ошибке).

Достоинства применения составных функций:

- упрощение структуры диаграммы алгоритма;
- возможность вносить изменения сразу во все повторяющиеся фрагменты алгоритма, облегчение его отладки;
- сильное сокращение числа объектов на графическом поле алгоритма.

По отношению к проекту составные функции условно делятся на три типа:

Системные составные функции — хранятся в базовой директории системы «Scorpio» и записываются туда при установке программы. Могут использоваться несколькими алгоритмами.

Проектные составные функции — создаются пользователем специально для конкретного алгоритма и хранятся в той же папке, что и проект.

Общие составные функции — могут храниться в любом месте и также использоваться несколькими алгоритмами.

3.1.4. Блок

На диаграмме функции имеют вид прямоугольников с ответвлениями на правой и левой стороне («чёрный ящик», имеющий «входы» и «выходы»).

Такие прямоугольники на диаграмме называются блоками. Каждый блок изображает функцию (простую или составную).

Наличие и состав входов и выходов (а также атрибутов) у блоков соответствует представляемой им функции. У некоторых блоков могут отсутствовать входы, у некоторых — выходы. В этом случае входными (или выходными) данными для данной функции являются буферы контроллера.

Допустимое количество входов, выходов и атрибутов блока приведено в таблице.

Наименование	Количество
входы	от 0 до 32
выходы	от 0 до 32
атрибуты	от 0 до 128

Входы всегда расположены с левой стороны блока, выходы — с правой. Атрибуты на диаграмме не изображены, их можно просмотреть или задать с помощью специального диалогового окна (настройка свойств блока).

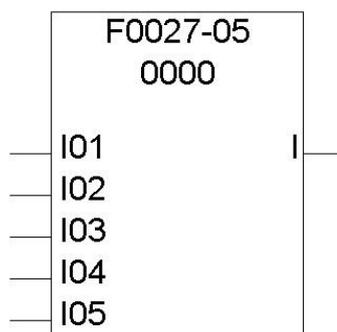


Рис. 3.1.4.1. Пример блока.

Блок содержит внутри себя имя выполняемой им функции (на рисунке — F0027-05) и её порядковый номер в алгоритме. У изображённой на рисунке функции этот номер 0000, такой номер присваивается самой первой по порядку выполнения функции. Следующая за ней функция получает номер 0001, потом 0002 и т. д. Входы и выходы функции имеют свои имена; на рисунке отображены входы I01, I02 и т.д., а единственный выход называется I.

Графически блок составной функции выглядит как и блок простой функции, лишь с одной разницей — у блока составной функции отчерчен верхний левый уголок.

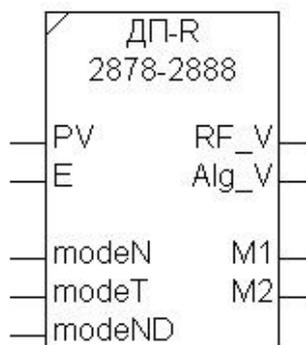


Рис. 3.1.4.2. Так выглядит составная функция.

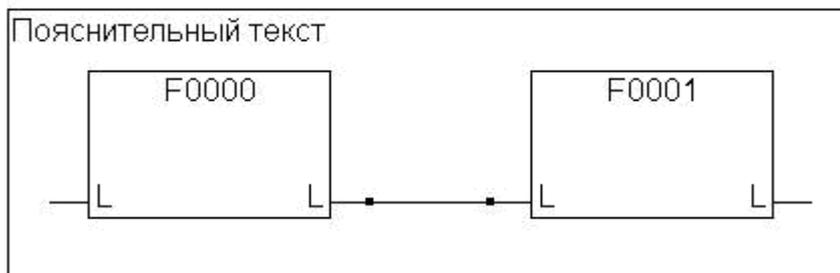
Номер составной функции — это номера первой и последней по порядку выполнения в алгоритме функций, которые выполняются внутри составной, и записывается в следующем формате: <номер первой выполняемой функции> — <номер последней выполняемой функции>.

У изображённого на рисунке блока этот номер 2878-2888, это означает, что блок, который будет выполняться первым внутри данной составной функции, имеет в алгоритме порядковый номер 2878, а блок, который будет выполняться последним в составе этой функции — имеет номер 2888.

3.1.5. Рамки

Рамка — это объект, в верхней части которого можно помещать комментарий — пояснительный текст к какой-либо части алгоритма или отдельной функции.

Рамки предназначены для облегчения понимания работы создаваемого алгоритма, а также для визуального разделения программы алгоритма на смысловые части, и не оказывают никакого влияния на его выполнение.

*Рис. 3.1.5.1. Пример рамки.*

3.1.6. Связь

Связь — графическое изображение пути данных, передаваемых из одной функции в другую. Связь соединяет выход одного блока и вход другого блока:

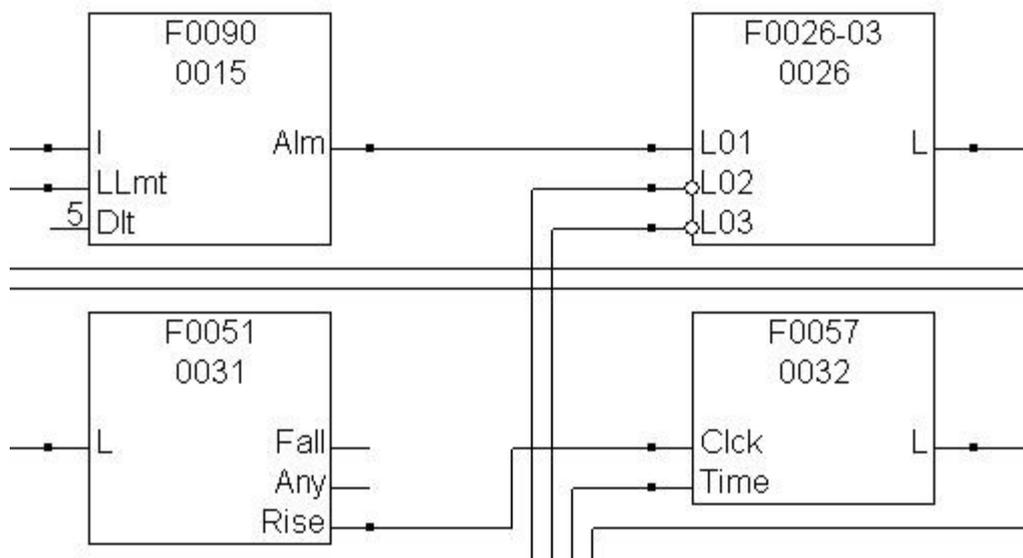


Рис. 3.1.6.1. Блоки со связями.

Одна связь может соединять один выход и несколько входов других функций. В этом случае говорится, что у связи несколько соединений. Кроме того, в месте примыкания связи к блокам и в месте разветвления соединений помещаются точки разветвления «узлы» (они отображают место разветвления, чтобы отличать его от мест графического пересечения не связанных между собой линий).

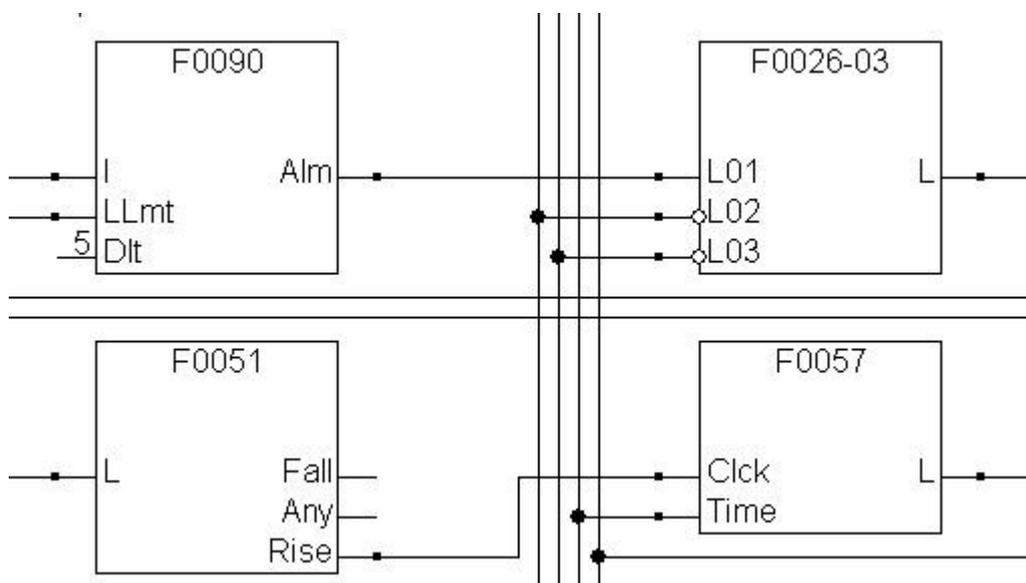


Рис. 3.1.6.2. Пример точек разветвления «узел».

Соединение можно нарисовать или стереть с помощью соответствующих команд редактора программы FBD_Bulider. Если нужно скопировать или перенести связь, чтобы привязать её ко входу другого блока, надо проследить за тем, чтобы начало и конец связи

начинались/заканчивались строго на входе или выходе блока, а не «повисали» в пространстве.

Если связь остаётся не присоединённой ни к какому объекту, то оборванное окончание линии графически помечается крестиком:

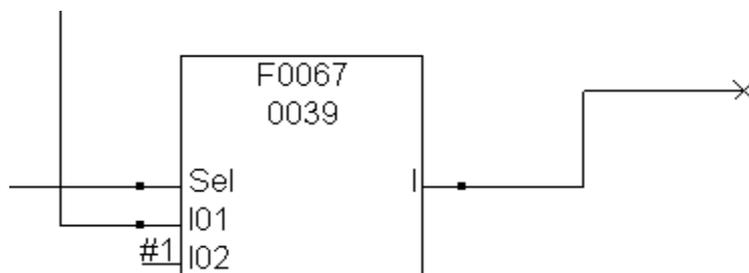


Рис. 3.1.6.3. Пример неприсоединённой связи.

В процессе разработки алгоритма иногда может потребоваться оставить связь неприсоединённой. Но к моменту компиляции проекта все связи должны иметь привязку к объектам, иначе компилятор выдаст предупреждение.

Входы и выходы функций подразделяются на те же типы, что и буферы памяти: логический (LOGIC), целый (INTEGER), с плавающей запятой (FLOAT).

Связь может соединять входы и выходы только одного типа.

3.1.7. Теги

Тег — это метка, используемая в алгоритме для указания на местоположение данных и устанавливаемая на вход или выход функции.

В системе «Scorpio» существуют следующие типы тегов:

Тип тега	Пояснение	Где используется
Именованная связь	Эквивалент линии связи между входом одного блока и выходом другого блока или нескольких блоков.	Алгоритм, составная функция

Тип тега	Пояснение	Где используется
Входная переменная	Указывает местоположение данных, находящихся во входном буфере.	Алгоритм
Выходная переменная	Указывает местоположение данных, находящихся в выходном буфере.	Алгоритм
Внутренняя переменная	Указывает местоположение данных, находящихся во внутреннем буфере.	Алгоритм
Вход составной функции	Именованная связь, обозначающая входы блоков составной функции.	Составная функция
Выход составной функции	Именованная связь, обозначающая выходы блоков составной функции.	Составная функция
Константа	Данные, значение которых определено и всегда постоянно.	Алгоритм, составная функция
Атрибут	Обозначает атрибут составной функции.	Составная функция

У каждого типа тега есть имя и комментарий. В зависимости от типа тег может иметь тип значения, адрес, позицию и значение.

Имя тега отображается на диаграмме (как правило, в виде условного буквенно-цифрового кода).

Комментарий — это краткое текстовое пояснение к назначению тега, его можно прочитать в таблице тегов рядом с именем.

Тип значения — один из типов данных, применяемых в алгоритме (LOGIC, INTEGER, FLOAT). У тегов «атрибут» предусмотрен дополнительный тип: FLASH, означающий, что данные этого тега хранятся в ячейке энергонезависимой памяти контроллера. При сбросе контроллера данные могут быть восстановлены из ячеек flash-памяти.

Адрес — целое число, которое обозначает номер ячейки в буфере или позицию входа, выхода или атрибута.

Позиция существует только у тегов, используемых в составных функциях (в т.ч. у атрибутов) и указывает номер входа или выхода составной функции, к которому привязан тег.

Есть три способа привязки тегов к входам функций: «**тег**», «**адрес**» и «**литера**».

«**Тег**» — это метка, своего рода ссылка, указывающая функции, в каком буфере ей брать данные для обработки или в какой буфер отправлять их после обработки.

«**Адрес**» — так же, как и константа, инициализирует данные начальным значением, только в качестве константы здесь используется адрес ячейки памяти, в которой находятся данные. Адрес — неизменяемое число, указывающее на местоположение определённой ячейки памяти в буфере контроллера. Адрес записывается в формате &X, где X — имя тега, и применим только к тегам, имеющим адрес (входные, выходные, внутренние).

[«Литера»](#) — это устойчивое неизменное значение, назначенное входу или выходу блока и присущее только ему. Литера записывается в поле «Имя»; формат имени литеры см. в разделе приложения [«Зарезервированные символы для формирования литеры»](#). В список тегов литеры не попадают.

3.1.8. Контакт

Контакты применяются как в алгоритме, так и в составных функциях.

Контакты не применяются при использовании данных протокола МЭК870 до их конвертации. В остальных случаях применение контактов возможно.

Данные передаются посредством тегов. Можно установить тег непосредственно на вход или выход, а можно установить его внутрь контакта и протянуть связь от блока к контакту.

[Контакт](#) — это объект диаграммы, который используется для графического изображения передачи данных.

3.1.8.1. Типы контактов

Контакты — это аналог функций «Приёмник» и «Передатчик», из которых «Приёмник» принимает данные из буфера памяти для обработки в алгоритме, а «Передатчик» — передаёт в буфер уже обработанные в алгоритме данные.

Соответственно, контакты называются похожим образом: входные и выходные (при этом они имеют либо вход, либо выход):

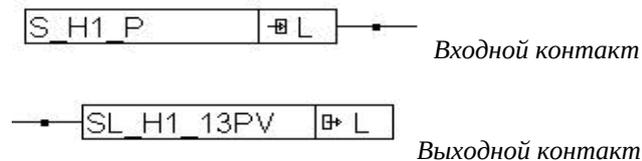


Рис. 3.1.8.1.1. Примеры входного и выходного контактов.

Входной контакт поставляет данные в функцию из буфера на обработку, а выходной — принимает из функции уже обработанные данные и сохраняет их в буфере.

Удобство применения контактов состоит, во-первых, в том, что таким образом гораздо удобнее копировать и перемещать теги по полю программы, а во-вторых, в более удобном визуальном восприятии имён тегов, что немаловажно при контроле правильности построения диаграммы. Это наглядно отображено на следующем рисунке:

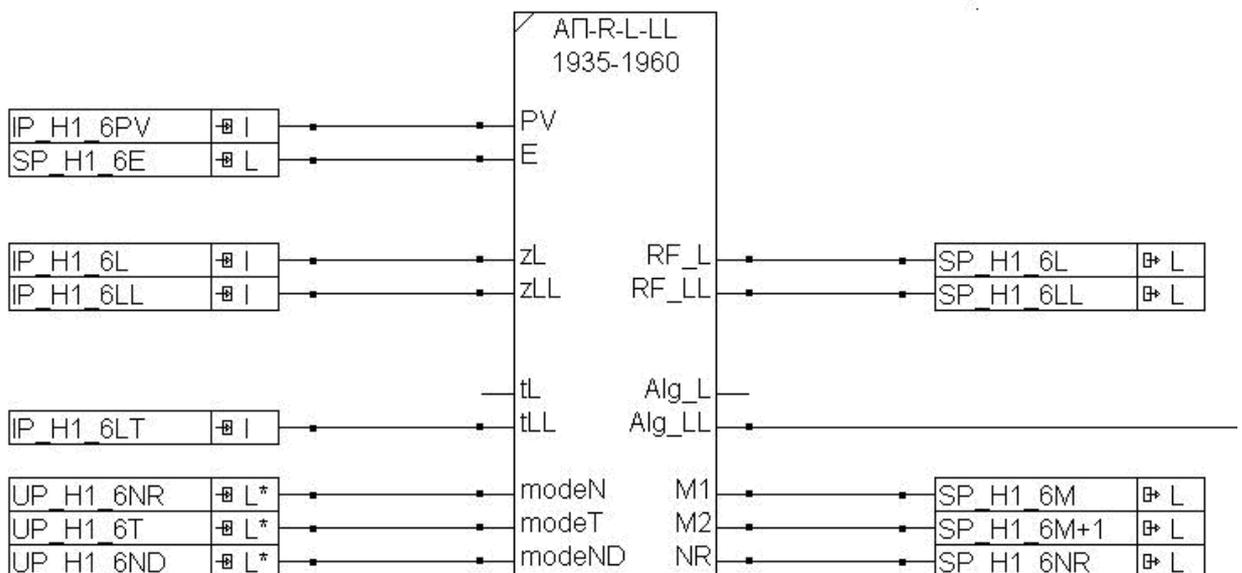


Рис. 3.1.8.1.2. Блок с привязанными к нему входными и выходными контактами.

Контакты позволяют выполнять такую важнейшую операцию по обработке данных как сброс после приёма. То есть, функция, считав данные из тега, содержащегося в контакте,

очищает ячейку памяти буфера. Это, например, позволяет наладить передачу в алгоритм логических команд (телеуправления) с верхнего уровня, ячейка после приема сбрасывается и готова к приёму новой команды.

3.1.8.2. Графическое изображение контакта

Рассмотрим на примере рис. 3.1.8.2.1, какую информацию несёт в себе графическое изображение контакта.



Рис. 3.1.8.2.1. Графическое изображение контакта.

Поскольку у изображённого контакта нет входа, но есть выход, то это входной контакт, и его задача — передавать данные из буфера для дальнейшей обработки в алгоритме.

Левая (большая) часть контакта содержит имя тега, в данном случае это UT_H1_3BNR. Имя тега назначается программистом и содержит зашифрованную информацию о тех данных, с которыми этот тег связан.

Правая (меньшая) часть контакта содержит следующую информацию:

- значок — отображает тип тега, возможные значения: входная переменная, выходная переменная и внутренняя переменная. В данном случае это входная переменная (прямоугольник со стрелкой внутрь). Значок обозначает выходную переменную, отсутствие значка обозначает внутреннюю переменную;
- символ **L** — обозначает тип данных: Logic (**L**), Integer (**I**) или Float (**F**);
- значок ***** — признак наличия у контакта свойства «сброс после приёма». Контакт, имеющий значок *****, работает следующим образом: приняв данные из ячейки памяти буфера, он обнуляет её значение.

Надо иметь ввиду, что поскольку в алгоритме могут присутствовать несколько контактов, берущих данные из этой же ячейки, то когда до них дойдёт по циклу очередь выполнения, они прочитают из ячейки уже значение «0», записанное туда первым выполнившимся контактом.

Такая функция контакта может быть удобна во многих случаях, например, если требуется подсчитать количество логических «1», поступивших в ячейку буфера за промежуток времени.

3.2. Правила языка FBD

При разработке алгоритма необходимо придерживаться следующих основных правил для его корректной компиляции и работы:

- блоки могут располагаться произвольно на графическом поле программы «FBD_Builder»;
- блоки геометрически могут накладываться друг на друга, но не должны полностью перекрываться;
- выходы блоков не должны соединяться связями со своими входами ни напрямую, ни через другие функции; исключение составляет связь через функцию «Обратная связь»;
- нельзя соединять выходы с выходами;
- связь должна соединять выходы блоков с входами такого же типа; т.е. выход блока типа Integer может быть соединен только с входом типа Integer, и т.д.

При выполнении блока происходит следующее: данные считываются со входов блока, обрабатываются функцией, а результат обработки данных затем передаётся на выходы.

Входы функции могут быть инициализированы несколькими способами:

- с помощью литеры: на входе устанавливается постоянное неизменяемое значение;
- с помощью тега: значение на входе может меняться в зависимости от значения тега;
- с помощью входящей связи: на вход функции поступает значение, являющееся результатом вычислений предыдущих функций;
- значение на входе может отсутствовать: в этом случае со входа функции по умолчанию считывается 0. Если речь идет о составной функции, то значение по умолчанию для её входов определяется индивидуально.

Начальные значения при разработке алгоритма устанавливаются на вход блока с помощью инструмента «Привязка», выбираемого кнопкой  на верхней панели инструментов.

Тип устанавливаемых начальных значений должен соответствовать типу входа блока (логические, целые и с плавающей запятой).

Если вход блока одновременно связан с выходом другого блока и имеет установленное значение, то при выполнении алгоритма данное значение будет проигнорировано и заменено на значение с выхода предыдущей функции.

Если вход блока, привязанный к литере, соединяется связью с блоком «Обратная связь», то значение литеры будет использовано только один раз — при первом цикле выполнения

алгоритма. При каждом последующем цикле функция будет выполняться с использованием значения, передаваемого на её вход блоком «Обратная связь».

Рассмотрим для наглядности два случая взаимодействия блоков: в первом случае данные приходят на вход блока из функции «Обратная связь», во втором — из другого блока.

- Взаимодействие изначальных входных данных и данных из блока «Обратная связь».

В этом примере данные циркулируют по кругу: в блоке «Сложение» (F0040-02) над ними производятся вычисления, результат поступает в блок «Обратная связь» (F0004) и возвращается оттуда в начало цикла, где снова подвергается вычислениям.

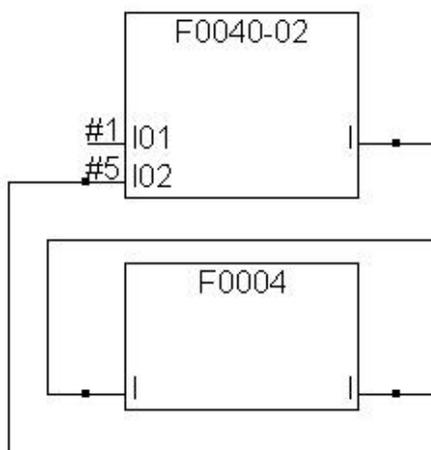


Рис. 3.2.1. Пример 1.

— 1-й цикл контроллера:

у блока F0040-02 на вход I01 устанавливается значение 1, а на вход I02 — значение 5; эти значения суммируются, и на выходе данного блока получается значение 6

— 2-й и все последующие циклы контроллера:

значение входа I01 у блока F0040-02 по-прежнему меняться не будет, т.е. равно значению 1, а на вход I02 будет передаваться значение с выхода блока F0040-02, т.е. сумма значений входов I01 и I02 предыдущего цикла контроллера. Эти значения будут суммироваться, и на выходе блока F0040-02 будет следующий результат:

2-й цикл $1+6=7$

3-й цикл $1+7=8$

4-й цикл $1+8=9$ и т.д.

Если на вход I01 у блока F0040-02 не устанавливать значение, то по умолчанию оно будет считаться равным 0, и при взаимодействии функций, представленном на рисунке, на выходе блока F0040-02 всегда будет значение 5.

- Взаимодействие изначальных входных данных и данных из блока, не являющегося блоком «Обратная связь».

Здесь данные циркулируют тем же образом, что и в предыдущем примере, лишь с той разницей, что результат каждого вычисления передаётся не только в блок «Обратная связь», но и уходит в блок «Повторитель» (F0001).

При этом значение, стоящее на входе «Повторителя», игнорируется и замещается входящим по связи. Установленное на входе блока значение (в данном случае «8»), не имеет смысла, если его вход соединён связью с блоком, не являющимся блоком «Обратная связь», как это и наблюдается на входе блока «Повторитель».

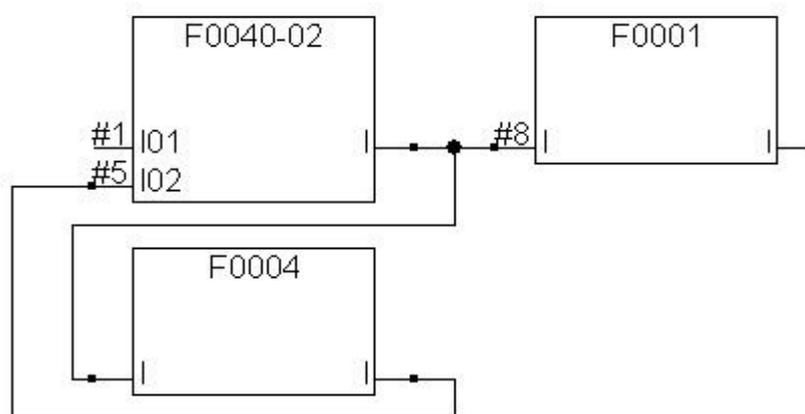


Рис. 3.2.2. Пример 2.

С данными можно осуществлять следующие операции:

- устанавливать на входе блока, используя инструмент «Привязка»;
- перемещать, копировать как вместе с соответствующим блоком с целью изменения расположения на поле алгоритма или размещения ещё одного такого же блока, так и отдельно от блока с целью установки такого же значения на вход другого блока;
- удалять как вместе с соответствующим блоком, так и отдельно от него.

При перемещении и копировании данных с целью установки значения на входе блока, необходимо учитывать, что тип данных должен соответствовать типу входа.

4. Разработка алгоритма рабочей программы для контроллера КСА-02

4.1. Среда разработки алгоритма «FBD_Builder»

Среда разработки алгоритма представлена в рамках системы «Scorpio». Среда разработки представляет собой набор выполняемых и конфигурационных файлов и объединяет инструменты, необходимые для создания, редактирования и компиляции алгоритма рабочей программы для контроллера КСА -02.

4.1.1. Загрузка

Чтобы загрузить среду разработки «FBD_Builder», нужно запустить из базовой директории файл FBD_Builder и создать в нем новый проект (кнопка  или команда меню «Файл» — «Создать»).

Появится окно, в котором нужно отметить, что именно вы хотите создать: новый алгоритм или же составную функцию.

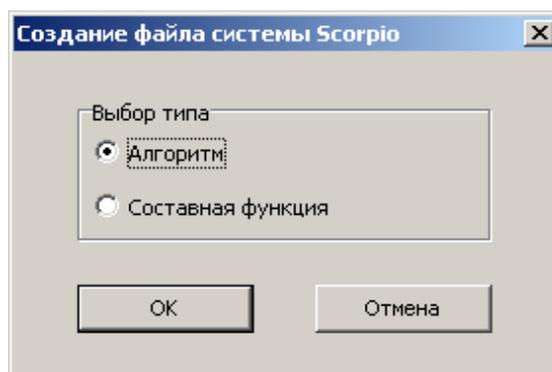


Рис. 4.1.1.1. Окно «Создание файла системы «Scorpio»».

Если выбран пункт «Составная функция», то после нажатия кнопки «ОК» совершается переход непосредственно к началу создания составной функции. Готовую составную функцию можно впоследствии использовать при создании любых проектов.

Если выбран пункт «Алгоритм», то произойдет переход к созданию нового проекта. Но сначала появится окно «Редактор параметров», где нужно задать первоначальные настройки параметров контроллера КСА-02, связанные с алгоритмом.

На редакторе параметров контроллера КСА-02 следует остановиться поподробнее.

4.2. Редактор параметров контроллера КСА-02

4.2.1. Назначение и основные принципы работы

Редактор параметров контроллера КСА-02 предоставляет доступ к редактированию настроек контроллера КСА-02, являющихся частью алгоритма рабочей программы контроллера и необходимых для корректности последующего обмена данными.

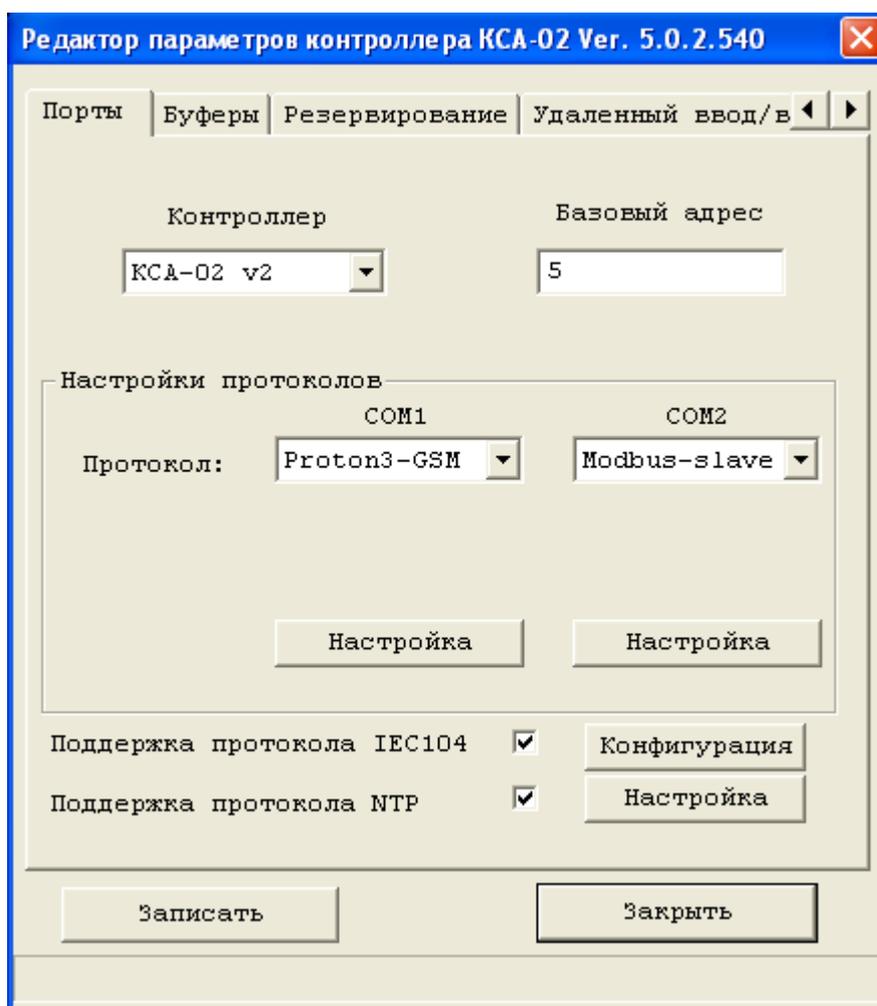


Рис. 4.2.1.1. Окно Редактора параметров контроллера КСА-02.

Окно редактора параметров контроллера КСА-02 содержит пять разделов, названия которых приводятся на закладках:

- Порты ([перейти к закладке](#));
- Буферы ([перейти к закладке](#));
- Резервирование ([перейти к закладке](#));
- Удалённый ввод/вывод ([перейти к закладке](#));
- Модули СТ1СРN ([перейти к закладке](#)).

Каждый из разделов предназначен для настройки определенной группы параметров.

Кнопка «**Записать**» принимает установленные пользователем настройки, изменённые в редакторе параметров контроллера, и сохраняет их в текущем проекте.

Кнопка «**Закреть**» закрывает окно редактора. При наличии изменений, которые не были сохранены, перед закрытием будет предложено сохранить внесённые изменения. Впоследствии редактор параметров может быть вызван кнопкой  «Редактор параметров контроллера».

4.2.2. Настройка параметров контроллера КСА-02

4.2.2.1. Порты. Общие положения

В разделе «Порты» задаются основные параметры для обмена по COM1, COM2. Из этой же формы задаются опции поддержки протоколов МЭК870 (IEC60870) и NTP.

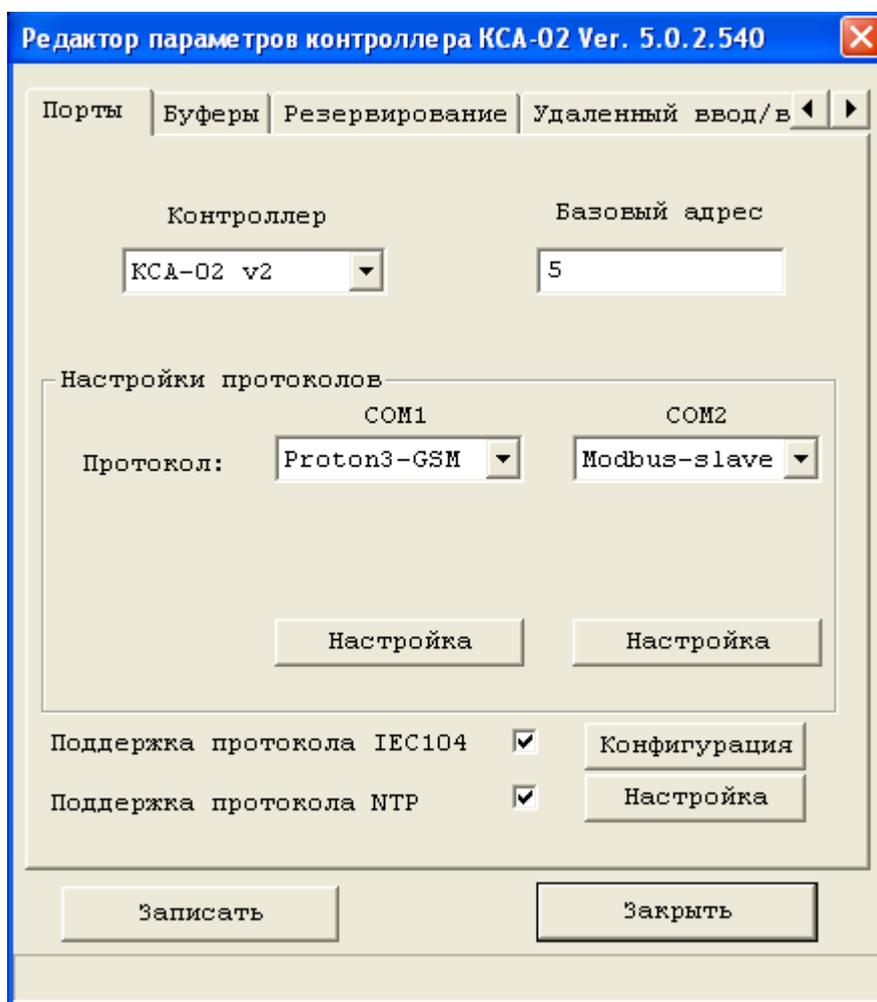


Рис. 4.2.2.1.1. Раздел «Порты».

Прежде всего выбирается из списка используемая платформа, в зависимости от года выпуска процессорного модуля: КСА-01 v2 (до 2006 года), КСА-02 (до 2014 года), КСА-02

v2 (после 2014 года). Значение базового адреса порта COM1, COM2 задаётся в целых числах десятичной системы исчисления, находящихся в диапазоне, зависящем от выбранных протоколов обмена. Базовый адрес контроллера обычно равен сетевому. Если предусмотрена система резервирования и контроллер во время работы изменяет свой режим с ведущего на ведомый, значение сетевого адреса по сравнению с базовым увеличивается на 32.

Далее настраиваются протоколы: в строке «Протокол» отображаются наименования протоколов, по которым будет производиться обмен через порты COM1 и COM2.

Нужные протоколы можно выбрать из списков в столбцах COM1 и COM2. У платформы КСА-02 v2 для порта COM1 возможны следующие варианты: Modbus-slave, Proton3-slave, Proton3-GSM, Modbus-master, MKSA-master. Для порта COM2 возможны варианты: Modbus-slave, Modbus-master, MKSA-master.

Редактирование настроек протокола осуществляется следующим образом:

- Нажать кнопку «Настройка» под названием перенастраиваемого порта («COM1» или «COM2»);
- Появится окно с набором соответствующих настроек протокола, изменение которых допускается в системе «Scorpio». Набор настроек индивидуален, и зависит от используемой платформы и протокола.

Тем не менее, можно выделить настройки, которые применяются почти во всех протоколах:

«Скорость обмена». Скорость обмена данными по выбранному протоколу (бод). На COM1 поддерживаются скорости: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. На COM2 поддерживаются скорости: 1200, 2400, 4800, 9600, 19200.

«Контроль чётности». Настройка наличия и типа бита чётности; возможны следующие варианты None (нет бита четности), Even (дополнение до четности), Odd (дополнение до нечетности).

«Время ответа». Только для Slave-протоколов. Время начала ответа после получения последнего байта от Master-устройства.

«Сетевой адрес». Только для Slave-протоколов. Индивидуальный адрес, по которому порт принимает пакеты. Данные, поступающие по другим адресам (за исключением группового) не обрабатываются.

«Изменение адреса». Только для Slave-протоколов. Установка флажка означает, что если главной процессор переходит в режим ведомого, то его сетевой адрес порта увеличивается на 32.

Кнопка "Записать" осуществляет сохранение измененных настроек. Кнопка "Закреть" осуществляет закрытие окна настройки протокола без сохранения проведенных изменений.

4.2.2.2. Порты. Окна настройки протоколов

Окно настройки протокола Modbus-slave.

При работе по протоколу Modbus-slave контроллер (slave) в ответ на пакет мастера возвращает запрашиваемые данные.

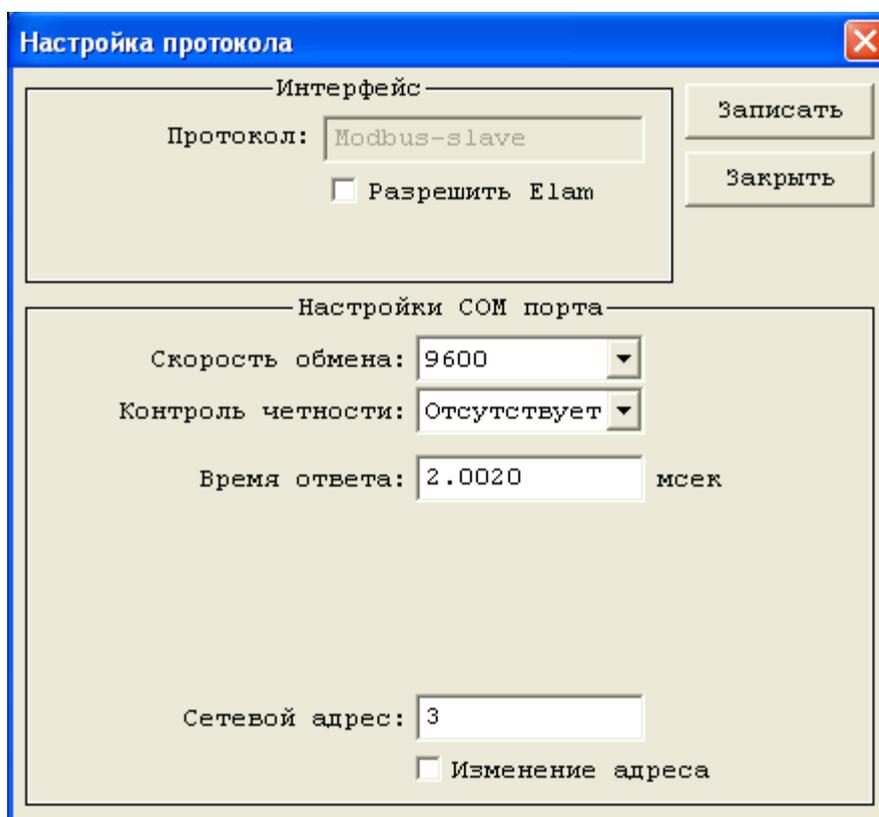


Рис. 4.2.2.2.1. Окно настройки протокола Modbus в режиме Slave (Modbus-Slave).

Флаг "Разрешить ELAM" доступен только на порту COM1. При установке флага на порту поддерживается протокол Modbus-slave-ELAM. Остальные настройки стандартные.

Окно настройки протокола Proton3 slave.

При работе по протоколу Proton3-slave контроллер (slave) в ответ на пакет мастера возвращает новые данные или квитанцию об отсутствии новых данных.

Настройка протокола

Интерфейс

Протокол: Proton3-slave

Записать

Закрыть

Настройки COM порта

Скорость обмена: 9600

Контроль четности: Отсутствует

Время ответа: 29.9520 мсек

Диапазон мастеров: Начальный: 1, Конечный: 2

Время разрыва соединения: 180.00 сек

Сетевой адрес: 13

Изменение адреса

4.2.2.2.2 Окно настройки протокола Proton3-Slave.

«Диапазон мастеров». Диапазон адресов Master-устройств, с которыми ведется обмен.

«Время разрыва соединения». Если на протяжении указанного времени не было успешных обменов с Master-устройством, Slave-устройство считает соединение разорванным. Должно быть равно аналогичному на мастере. Для систем телемеханики, работающих по радиоканалу, обычно принимается равным 180 сек, в Ethernet-системах – 32 сек.

Остальные настройки стандартные.

Окно настройки протокола Proton3-GSM-client.

При работе по протоколу Proton3-GSM контроллер (client) и внешнее устройство (server) отправляют пакеты с новыми данными по мере их появления без запроса. Полученные данные должны быть подтверждены. При долгом отсутствии новых данных посылаются служебные пакеты проверки связи.

Настройка протокола

Интерфейс

Протокол: Proton3-GSM

Записать

Закрыть

Настройки COM порта

Скорость обмена: 115200

Контроль четности: Отсутствует

Интервал после приема: 1.0004 мсек

Интервал передач, сек: 5 120

Интервал синхр. времени, час: 1.0

Ожидание подтвр., сек: 40

Сетевой адрес: 5

4.2.2.2.3. Окно настройки протокола Proton3-GSM-client.

«Интервал после приема». В течение указанного интервала после приема пакета контроллер игнорирует все получаемые пакеты. Может использоваться для защиты от эхо-сигнала.

«Интервал передач, минимальный». Интервал между двумя последовательно отправляемыми пакетами не может быть меньше этого интервала. Если новые данные поступили до окончания этого интервала, их передача задерживается до окончания интервала.

«Интервал передач, максимальный». Интервал между двумя последовательно отправляемыми пакетами не может быть больше этого интервала. Если до окончания этого интервала новых данных не поступило, отправляется служебный пакет проверки связи.

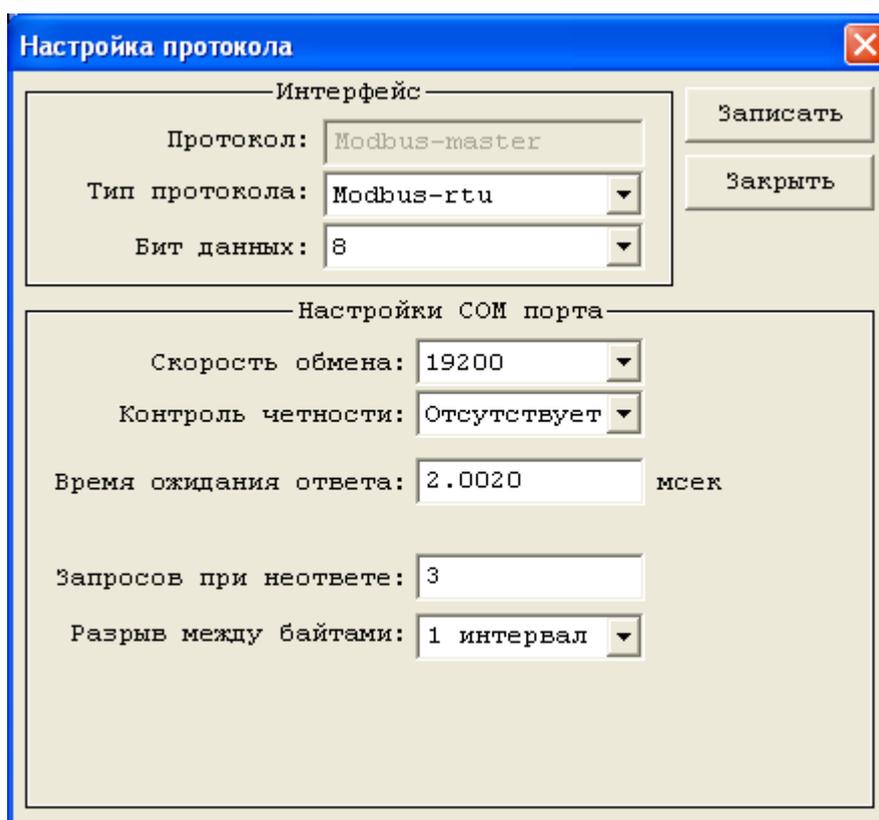
«**Интервал синхронизации времени**». Интервал между двумя последовательно отправляемыми служебными пакетами синхронизации времени.

«**Ожидание подтверждения**». Интервал, в течение которого контроллер ожидает пакета подтверждения получения переданных данных. Если до окончания этого интервала пакет подтверждения не получен, неподтвержденные данные передаются повторно. После нескольких неудачных повторных передач, пакет с такими данными удаляется из очереди передачи.

Остальные настройки стандартные.

Окно настройки протокола Modbus-Master.

При работе по протоколу Modbus-master контроллер (master) запрашивает у подключенных устройств (slave) данные и ожидает в ответ их получения. Формирование запросов осуществляется функциями алгоритма.



4.2.2.2.4. Окно настройки протокола Modbus в режиме Master (Modbus-Master).

«**Тип протокола**». Из двух возможных типов протокола — «Modbus RTU» и «Modbus ASCII» — нужно выбрать необходимый для использования. Более подробно о различиях в разновидностях протокола Modbus см. в описании протокола [«Протокол Modbus в режиме Master»](#).

«**Бит данных**». Только для протокола MKSA-Master. Количество бит данных в байте. Для Modbus RTU количество бит обычно всегда 8, для Modbus ASCII обычно всегда 7.

«**Запросов при неответе**». Число повторов запроса, которые Master-устройство пошлёт при неответе Slave-устройству, прежде чем перейдёт в состояние ошибки (максимальное значение 15):

- «0» — 1 запрос, без повтора;
- «15» — 1 запрос и 15 повторов.

«**Разрыв между байтами**». Задаётся в количестве интервалов, равных 4 битам. Если разрыв между поступающими данными превышает указанное количество интервалов, фиксируется ошибка.

Остальные настройки стандартные.

4.2.2.3. Порты. Поддержка семейства протоколов МЭК870 (IEC60870)

Установка флага "**Поддержка протокола IEC104**" на вкладке "Порты" (см.рис.4.2.2.1.1) обеспечивает конвертацию типов данных, используемых в семействе протоколов МЭК870, к типам данных, используемых в контроллере. После установки флага становится доступна кнопка "**Конфигурация**", нажатие на которую осуществляет переход к форме настройки параметров конвертации.

Конвертация типов данных поддерживается, начиная с версии 5.0.4 программного обеспечения процессорного модуля. Установка данного флага не приводит к автоматическому выбору протокола из семейства МЭК870 на порту контроллера. Для такого выбора тип нужного протокола должен быть явно указан при настройке порта контроллера.

Поддержка протокола МЭК870-5-104 (IEC60870-104) осуществляется через модуль ST1SPE10, начиная с версии программного обеспечения 6.0 данного модуля.

Поддержка протокола МЭК870-5-101 (IEC60870-101) пока не реализована.

Общие настройки протоколов IEC

Смещения блоков IEC в буферах

Буфер входной логический	16
Буфер входной целый	8
Буфер входной с плавающей точкой	4
Буфер выходной логический	16
Буфер выходной целый	8
Буфер выходной с плавающей точкой	4

Условия на инициативную передачу

	Смещение в выходном буфере	Начальный адрес
Нормализованные величины	48	0
Масштабированные величины	64	0
Величины с плавающей точкой	8	0

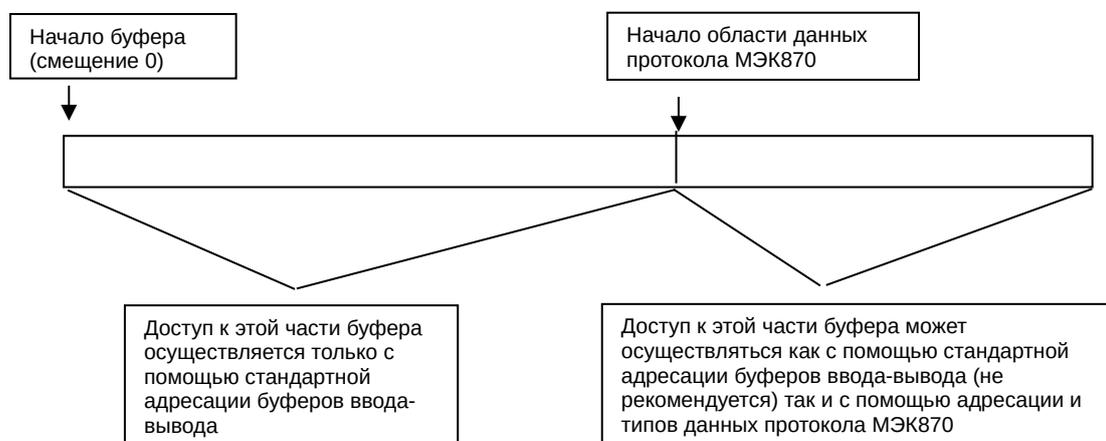
Количества и начальный адреса информационных элементов

	Входные элементы (команды)		Выходные элементы (данные)	
	Количество	Начальный адрес	Количество	Начальный адрес
Одиночный логический	32	2100	32	100
Двойной логический	16	2200	16	200
Шаговый регулятор	8	2300	8	300
Битовый стринг	4	2400	4	400
Нормализованная величина	16	2500	16	500
Масштабированная величина	16	2600	16	600
Величина с плавающей точкой	4	2700	4	700
Интегральный счетчик			4	800

Принять Отмена

4.2.2.3.1. Окно настройки конфигурации поддержки протоколов МЭК870 (IEC60870).

Настройки в группе "Смещения блоков IEC в буферах" выделяют в стандартных буферах ввода-вывода контроллера области, доступ к которым может осуществляться с использованием адресации и типов данных протокола IEC60870.



4.2.2.3.2. Выделение в буфере ввода-вывода области данных протоколов МЭК870 (IEC60870).

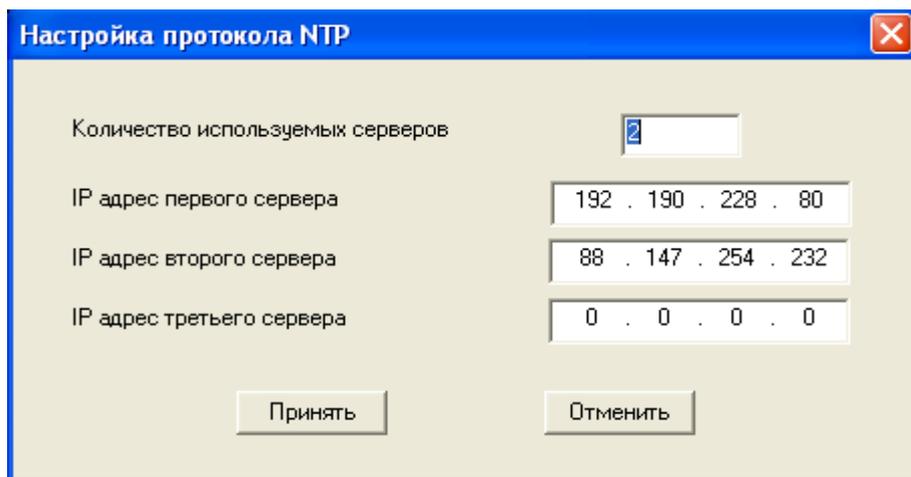
Если выделенная область буфера недостаточна для хранения заявленного объема данных протокола IEC60870, при компиляции алгоритма выдается предупреждение.

В группе **"Количества и начальные адреса информационных элементов"** для поддерживаемых типов данных протокола МЭК870 задается количество элементов для типа данных и адрес первого элемента данного типа. Предполагается, что адресация по типу данных последовательная. В примере на рис. 4.3.3.2.1 для типа данных "выходной двойной логический" определено 16 элементов данного типа; причем к данному типу относятся информационные элементы с адресами от 200 до 215 включительно.

Настройки в группе **"Условия на инициативную передачу"** выделяют в стандартных буферах ввода-вывода контроллера области, в которых находятся значения нечувствительности для передачи данных протокола МЭК870 аналогового типа. Настройка **"Смещение в выходном буфере"** дается по стандартной адресации буферов ввода-вывода; для типов данных "Нормализованная величина" и "Масштабированная величина" - в буфере выходном целом, для типа данных "Величина с плавающей точкой" - в буфере выходном с плавающей точкой. Значение в поле **"Начальный адрес"** дается по системе адресации протоколу МЭК870. Ненулевое значение данного поля дает возможность менять значение нечувствительности по протоколу МЭК870. Если значение в поле "Начальный адрес" равно нулю, изменение нечувствительности по протоколу МЭК870 не поддерживается.

4.2.2.4. Порты. Поддержка протокола NTP

Установка флага **"Поддержка протокола NTP"** на вкладке "Порты" (см.рис.4.2.2.1.1) обеспечивает возможность синхронизации времени контроллера с помощью внешнего сервера точного времени по протоколу NTP (SNTP). После установки флага доступна кнопка "Настройка".



4.2.2.4. Настройка протокола NTP.

«Количество используемых серверов». Количество серверов точного времени, по которым может проводиться синхронизация точного времени. Минимальное число серверов - один, максимальное число серверов - три.

«IP адрес ... сервера». IP адреса первого, второго и третьего серверов точного времени. При использовании протокола NTP для синхронизации используется лучший из доступных серверов; при использовании протокола SNTP для синхронизации используется первый сервер, а остальные сервера являются резервными на случай недоступности первого сервера. Если используется только один сервер, работа протоколов NTP и SNTP одинакова.

Для использования синхронизации времени по протоколу NTP (SNTP), необходимо, чтобы данный протокол поддерживался как ядром процессорного модуля, так и системным интерфейсным модулем.

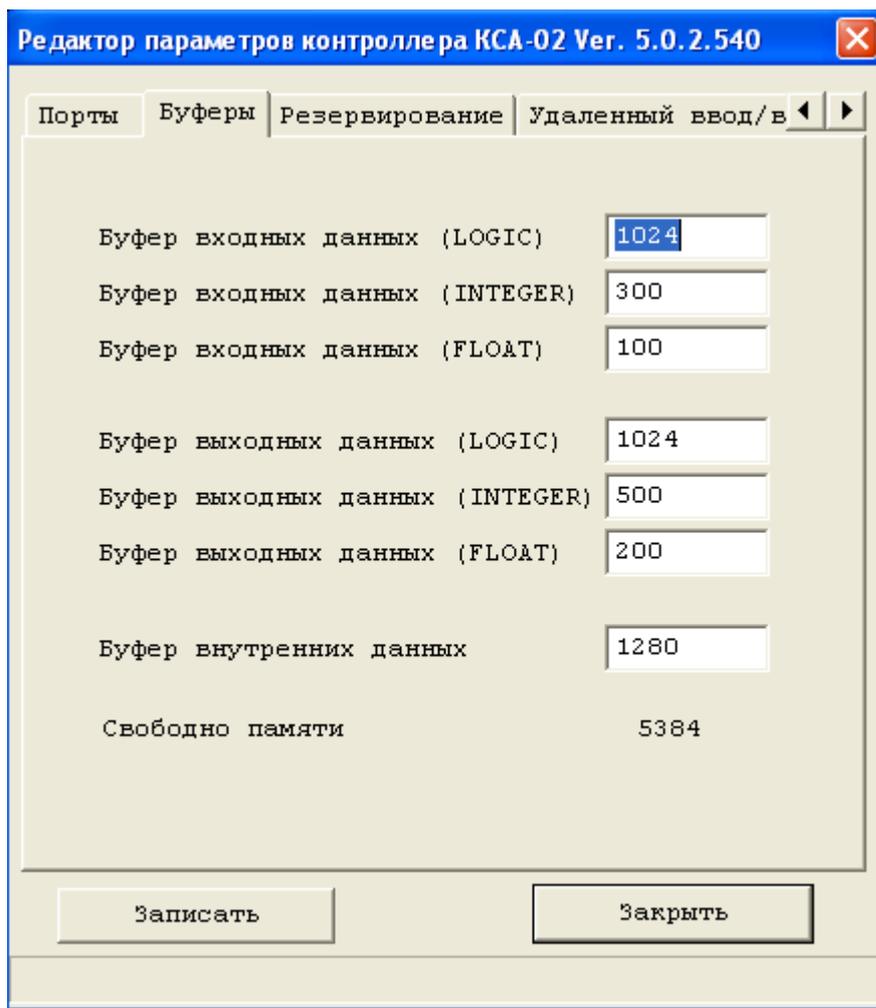
Протокол NTP поддерживается, начиная с версии 5.0.4 программного обеспечения процессорного модуля.

Протокол SNTP поддерживается, начиная с версии 6.0 программного обеспечения интерфейсного модуля СТ1СРЕ10.

Период синхронизации точного времени по протоколу NTP (SNTP) задается из алгоритма.

4.2.2.5. Буферы

Раздел «Буферы» предназначен для определения размера буферов обмена в соответствии с объёмом и типом передаваемых данных. Размер буфера указывается в единицах его *размерности* (соответствие размерности и типа буфера приведено в разделе [«Типы данных»](#)).



4.2.2.5. Вкладка «Буферы».

Вкладка «Буферы» содержит следующие поля:

Буфер входных данных (LOGIC): текущее значение буфера входных данных типа LOGIC, размерность — бит. Значение данного буфера всегда кратно 16 (1 слово).

Буфер входных данных (INTEGER): текущее значение буфера входных данных типа INTEGER, размерность — слово. Значение данного буфера всегда кратно 1 (1 слово).

Буфер входных данных (FLOAT): текущее значение буфера входных данных типа FLOAT, размерность — 2 слова. Значение данного буфера всегда кратно 1 (2 слова).

Буфер выходных данных (LOGIC): текущее значение буфера выходных данных типа LOGIC, размерность — бит. Значение данного буфера всегда кратно 16 (1 слово).

Буфер выходных данных (INTEGER): текущее значение буфера выходных данных типа INTEGER, размерность — слово. Значение данного буфера всегда кратно 1 (1 слово).

Буфер выходных данных (FLOAT): текущее значение буфера выходных данных типа FLOAT, размерность — 2 слова. Значение данного буфера всегда кратно 1 (2 слова).

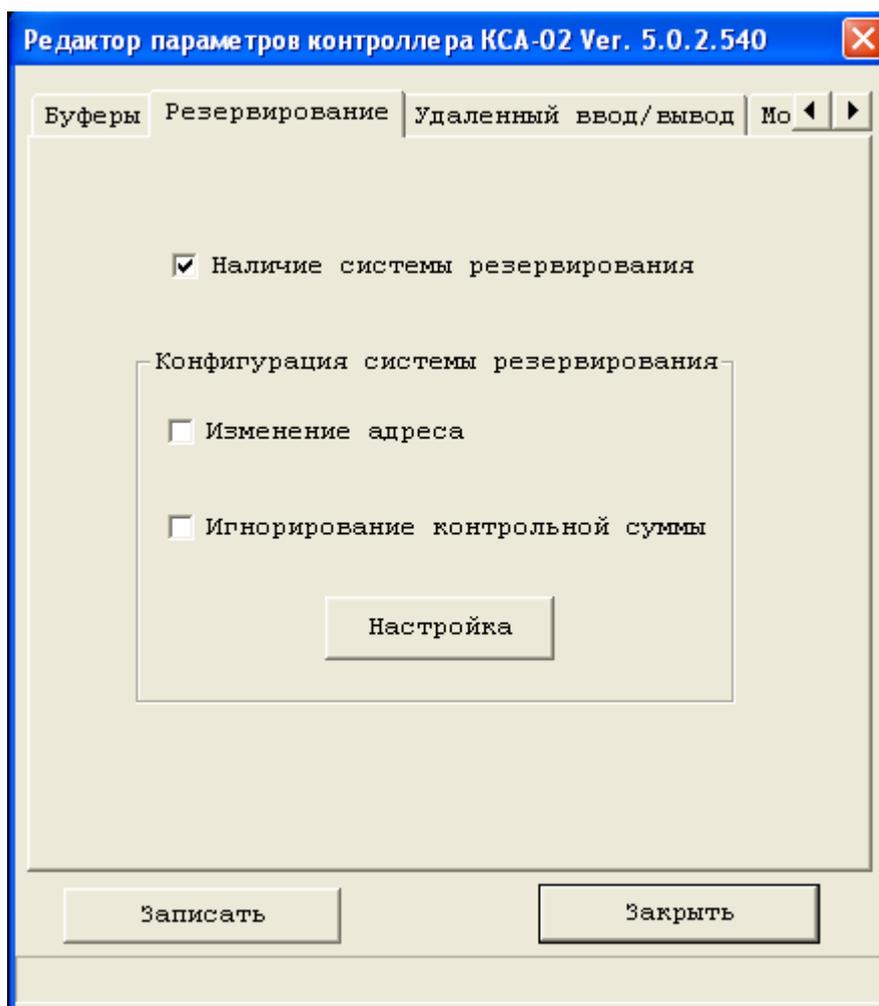
Буфер внутренних данных: текущее значение буфера внутренних данных, размерность — 1 слово. Значение данного буфера всегда кратно 1 (1 слово).

Свободно памяти: текущее значение свободной памяти, размерность — 1 слово. Значение данного буфера всегда кратно 1 (1 слово). Отображаемое количество свободных слов памяти корректируется каждый раз при установке размера какого-либо типа буфера.

Общий участок памяти выделяемый для использования под буферы данных, составляет 8192 слова и располагается в памяти контроллера. Для других целей данный участок памяти не используется.

4.2.2.6. Резервирование

Если проектом предусмотрено резервирование главного процессора контроллера КСА-02, то при разработке алгоритма необходимо сконфигурировать параметры работы и настроить буферы обмена данной системы резервирования. Для этого на вкладке «Резервирование» расположены соответствующие поля и кнопки:



4.2.2.6.1. Окно настройки системы резервирования контроллера КСА-02.

Флажок «Наличие системы резервирования»: отображает наличие системы резервирования в контроллере КСА-02. Чтобы сделать доступными конфигурирование параметров работы системы резервирования и настроить для неё буферы обмена, надо установить флажок «Наличие системы резервирования».

Флажок «Изменение адреса»: установка флажка означает, что если контроллер переходит в режим ведомого, то его сетевой адрес увеличивается на 32, и таким же образом увеличатся и сетевые адреса коммуникационных модулей контроллера.

Флажок «Игнорирование контрольной суммы»: установка флажка необходима при резервировании контроллеров с различающимися алгоритмами. Она означает, что если алгоритм участвует в системе резервирования, то контрольная сумма его алгоритма не будет сравниваться с контрольной суммой алгоритма дублирующего контроллера.

Кнопка «Настройка»: открывает окно, где следует указать область резервирования для каждого типа буферов.

	начальный адрес	размер
Буфер входных данных (LOGIC)	0	0
Буфер входных данных (INTEGER)	0	32
Буфер входных данных (FLOAT)	0	16
Буфер внутренних данных	0	32

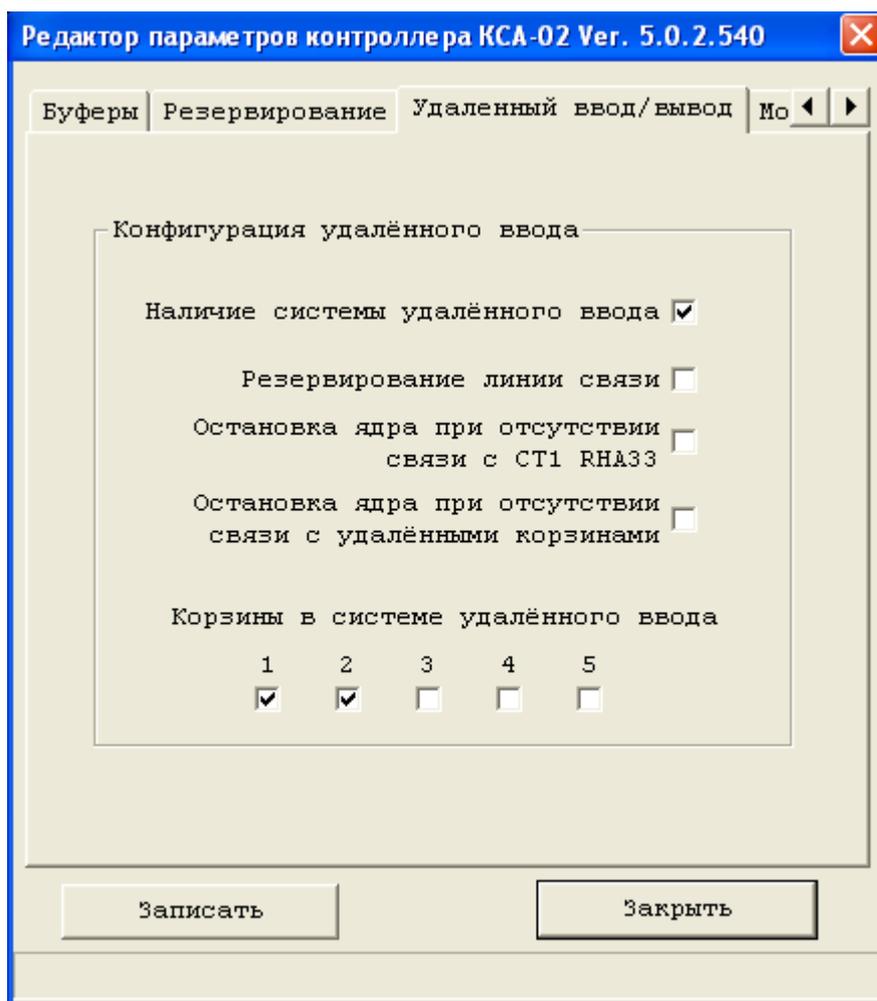
Закрыть

4.2.2.6.2 Окно настройки буферов для системы резервирования.

Если нужно зарезервировать весь буфер какого-либо типа целиком, то в качестве начального адреса задаётся 0, а в поле «размер» вписывается полный размер этого буфера (указанный на вкладке «Буферы»). Если нужно зарезервировать лишь часть буфера — указывается число, которое станет начальным адресом резервируемой части, а в поле «размер» указывается размер части, подлежащей резервированию.

4.2.2.7. Удаленный ввод/вывод

Если проектом предусмотрено использование станций удаленного ввода-вывода контроллера КСА-02, то при разработке алгоритма необходимо сконфигурировать и настроить параметры их работы. Для этого на вкладке «Удалённый ввод/вывод» имеются соответствующие флажки:



4.2.2.7.1. Окно настройки системы удалённого ввода/вывода.

Наличие системы удалённого ввода: признак наличия в конфигурации контроллера системы удалённого ввода-вывода. Установка флажка открывает доступ к дальнейшей конфигурации параметров работы системы.

Резервирование линии связи: установка флажка означает, что линия связи с удалёнными корзинами станций ввода-вывода резервируется (дублируется).

Остановка ядра при отсутствии связи с СТ1RHA33: установка флажка означает, что при отсутствии связи с модулем обмена с СУВД алгоритм будет автоматически остановлен;

Остановка ядра при отсутствии связи с удалёнными корзинами: установка флажка означает, что ядро будет автоматически остановлено при одновременном выполнении следующих условий:

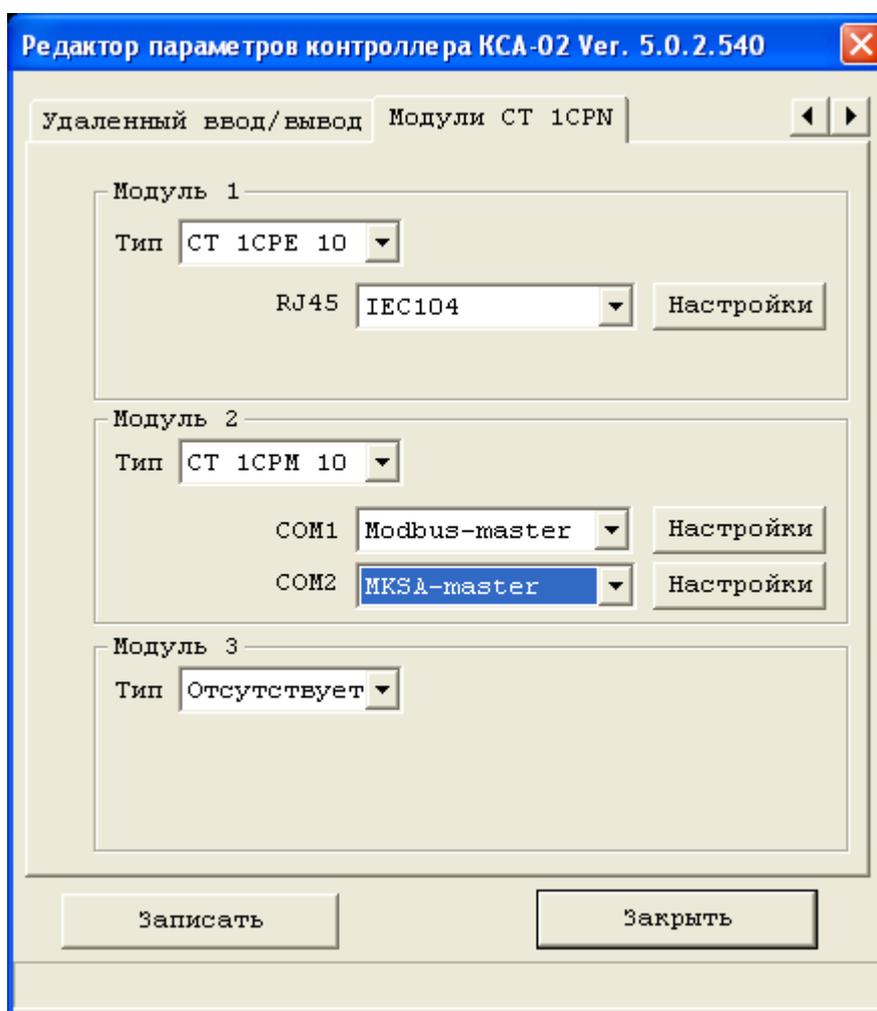
- отсутствие связи со всеми СУВД (их количество может быть различным);

- отсутствие связи с модулем СТ1RHA33, расположенным в дублирующем главном процессоре (если включена система резервирования).

Корзины в системе удалённого ввода: необходимо указать, какие из станций удаленного ввода-вывода присутствуют в системе (флажки «1» — «5»).

4.2.2.8. Модули СТ1CPN

Если в состав контроллера КСА-02 входят интерфейсные модули СТ1CPM10 или СТ1CPE10, то при разработке алгоритма необходимо указать, какие именно модули присутствуют в корзине и сконфигурировать настройки их СОМ-портов. Для этого на вкладке «Модули СТ1CPN» имеются соответствующие области и элементы:



4.2.2.8. Окно настройки «Модули СТ1CPN».

Модуль 1, (наименование): настройки первого интерфейсного модуля, включенного в корзину главного процессора контроллера КСА-02; счёт ведется слева направо от процессорного модуля, т.е. первым считается самый левый интерфейсный модуль.

Модуль 2, (наименование): настройки второго интерфейсного модуля (следующий интерфейсный модуль, расположенный правее первого).

Модуль 3, (наименование): настройки, относящиеся к третьему интерфейсному модулю (следующий интерфейсный модуль, расположенный правее второго).

В одной корзине главного процессора контроллера КСА-02 может находиться максимум 3 интерфейсных модуля (СТ1СРМ10, СТ1СРЕ10), причём их расположение в корзине контроллера может быть произвольно. Счёт модулей производится по порядку; таким образом ситуация, когда присутствуют модули 1 и 3, а модуль 2 отсутствует, невозможна.

СОМ1, СОМ2: наименование протокола, по которому будет производиться обмен данными.

Кнопка «Настройки»: открывает окно дополнительных настроек протокола.

Модуль СТ1СРЕ10, установленный в 3 слоте корзины контроллера называется системным. Его частичная работоспособность обеспечивается даже при остановленном ядре контроллера. Модуль, который предполагается использовать в качестве системного, при конфигурировании всегда должен находиться в группе "Модуль 1".

4.2.2.9. Настройки СОМ1 и СОМ2 для модуля СТ 1СРМ10

Модуль СТ1СРМ10 на своих последовательных портах поддерживает следующие протоколы: Modbus-slave, RTM64, Modbus-master, MKSA-master.

На поле «Модуль №» следует определить следующие настройки:

«Протокол». Наименование протокола, по которому будет производиться обмен данными через СОМ-порт. (не все настройки содержатся в ЭТОМ окне)

«Настройка». Для просмотра дополнительных настроек, зависящих от выбранного протокола, предназначено окно «Настройка протокола», которое открывается на экране монитора при нажатии на кнопку «Настройка».

4.2.2.10. Настройка протокола Modbus-Slave

Настройка протокола

Интерфейс

Протокол: Modbus-slave

Настройки COM порта

Скорость обмена: 38400

Контроль четности: Отсутствует

Время ответа: 9.9840 мсек

Сетевой адрес: 33

Изменение адреса

Записать

Закрыть

4.2.2.10. Настройка Modbus-Slave для модуля CT1CPM10.

«**Скорость обмена**». Скорость обмена данными по выбранному протоколу; возможны следующие варианты 1200, 2400, 4800, 9600, 19200, 38400.

«**Контроль чётности**». Настройка чётности; возможны следующие варианты None (нет), Even (четный), Odd (нечетный)

«**Время ответа**». Только для Slave-протоколов. Время начала ответа после получения последнего байта от Master-устройства.

«**Разрыв между байтами**». Задаётся в количестве интервалов, равных 4 битам. Если разрыв между поступающими данными превышает указанное количество интервалов, контроллер фиксирует ошибку.

«**Сетевой адрес**». Адрес коммуникационного модуля в сети, работающей по протоколу Modbus в режиме Slave.

«**Изменение адреса**». Установка флажка означает, что если главный процессор переходит в режим ведомого, то его сетевой адрес увеличивается на 32, и таким же образом будут увеличены и сетевые адреса коммуникационных модулей входящих в его состав.

4.2.2.11. Настройка протокола RTM64-Slave

Настройка протокола

Интерфейс

Протокол: RTM64-slave

Настройки COM порта

Скорость обмена: 38400

Контроль четности: Отсутствует

Время ответа: 9.9840 мсек

Сетевая

Сетевой адрес: 1

Изменение адреса

Настройки RTM64 (число команд 0)

Совместимость с "РОТОР" Проверка интервала

Список команд

Параметры команды

Команда: []

Ответная команда: []

Тип: []

Тип буфера: []

Начальный адрес: []

Размер: []

Добавить

Изменить

Удалить

4.2.2.11. Настройка протокола RTM64-Slave для модуля CT1CPM10.

«Совместимость с системой РОТОР». Для совместимости с модифицированным протоколом RTM64 используемым в системе «Ротор» необходимо установить флажок «Совместимость с системой «РОТОР»». Это означает, что контрольная сумма будет дополняться до математической суммы (более подробно см. описание протокола RTM64).

«Проверка интервала». Данные передаются с определёнными интервалами. При установке флажка «Проверка интервала» этот промежуток проверяется на предмет отсутствия в нём данных. Если они обнаруживаются, фиксируется ошибка.

«Список команд». Команды нужны для работы с входными и выходными буферами. Они содержат информацию о том, как распределять по буферам данные каждого типа (Logic, Integer или Float) при записи в контроллер и при чтении из контроллера.

Для создания команды необходимо в поле «Команда» ввести её номер — шестнадцатеричное число от 0 до FFFF — и нажать кнопку «Добавить».

«Команда», «Ответная команда». С помощью этих параметров команде присваивается номер (имя).

«Тип». Тип действия с данными: чтение из контроллера или запись в контроллер.

«Тип буфера». Тип буфера, куда будут записываться данные.

- LOGIC_IN — логический входной
- INT_IN — целых чисел входной

- FLOAT_IN — чисел с плавающей запятой входной
- INSIDE — внутренний
- LOGIC_OUT — логический выходной
- INT_OUT — целых чисел выходной
- FLOAT_OUT — чисел с плавающей запятой выходной

«Начальный адрес». Начальный адрес части буфера, куда будут записываться данные указанного типа.

«Размер». Размер части буфера, куда будут записываться данные указанного типа. Таким образом, если в качестве начального адреса будет указано число 5, а размер части буфера — 7, то данные будут записаны в ячейки с адресами 5, 6, 7, 8, 9, 10, 11.

Для удаления существующей команды необходимо выделить требуемую команду в «Списке команд» и нажать кнопку «Удалить».

«Редактирование команды». Для редактирования команды необходимо выбрать требуемую команду из «Списка команд», в секции «Редактирование команды» задать требуемые настройки. Все настройки команд сохраняются автоматически, сразу после ввода.

В этих полях указываются номера команд, которые получает контроллер от внешнего устройства через радиомодем, в зависимости от номера полученной команды контроллер производит те или иные действия. В ответ ему также необходимо передать номер команды, чтобы устройство с которым общается контроллер поняло, что его команда принята корректно.

«Изменение адреса». Если флажок «Изменение адреса» установлен — адрес главного процессора контроллера увеличивается на 32, используется для установки режима ведомого.

4.2.2.12. Настройка протоколов Modbus-Master, MKSA-Master

Настройка протокола

Интерфейс

Протокол: Modbus-master

Тип протокола: Modbus-ascii

Бит данных: 8

Настройки COM порта

Скорость обмена: 9600

Контроль четности: Отсутствует

Время ожидания ответа: 9.9840 msec

Запросов при неответе: 1

Разрыв между байтами: 1 интервал

Записать

Закрыть

4.2.2.12 Настройка Modbus-Master и MKSA-Master для модуля CT1CPM10.

«**Тип протокола**». Из двух возможных типов протокола — «Modbus RTU» и «Modbus ASCII» — нужно выбрать необходимый для использования. Более подробно о различиях в разновидностях протокола Modbus см. в описании протокола Modbus.

«**Бит данных**». Только для протокола Modbus-Master. Количество бит данных в байте. Для Modbus RTU количество бит обычно всегда 8, для Modbus ASCII обычно всегда 7.

«**Скорость обмена**». Скорость обмена данными по выбранному протоколу.

«**Контроль чётности**». Настройка контроля чётности.

«**Время ожидания ответа**». Только для Master-протоколов. Время начала передачи первого байта ответа от Slave-устройства после окончания передачи последнего бита последнего байта в запросе Master-устройства. Максимальное значение 106,47 миллисекунд, значение должно быть кратно 0,026 миллисекундам.

«**Запросов при неответе**». Число повторов запроса, которые Master-устройство пошлёт при неответе Slave-устройству, прежде чем перейдёт в состояние ошибки (максимальное значение 15):

- «0» — 1 запрос, без повтора;

- «15» — 1 запрос и 15 повторов.

«Разрыв между байтами». Задаётся в количестве интервалов, равных 4 битам. Если разрыв между поступающими данными превышает указанное количество интервалов, контроллер фиксирует ошибку.

4.2.2.13. Протоколы для модуля ST1CPE10

Модуль ST1CPE10 работает с протоколами обмена данными: Modbus-TCP (slave), Proton3-IP (slave), Modbus-TCP (master), МЭК870-104(client). Тип протокола обмена данными выбирается на вкладке "Модули ST1 CPN10" (рис. 4.2.2.8). Для каждого из протоколов задаются настройки.

Протокол Modbus-TCP (slave) поддерживается всегда. В качестве КСА-адреса и IP-адреса используются соответствующие настройки выбранного протокола. Для получения данных по этому протоколу следует использовать порт 502.

Протокол Proton3-IP поддерживается только в случае, если модуль ST1CPE10 установлен в 3 слоте контроллера. Более того, начиная с версии 5.0 программного обеспечения модуля ST1CPE10, поддержка протокола для модуля в 3 слоте осуществляется независимо от выбранного протокола обмена данными.

Поддержка протокола МЭК870-5-104 (IEC60870-104) осуществляется, начиная с версии 6.0 программного обеспечения модуля ST1CPE10.

Кроме протоколов обмена данными, модуль ST1CPE10 также поддерживает служебные протоколы ARP, ICMP; а начиная с версии 6.0 программного обеспечения - протокол синхронизации времени NTP. Протокол NTP поддерживается только модулем, установленном в 3 слоте контроллера.

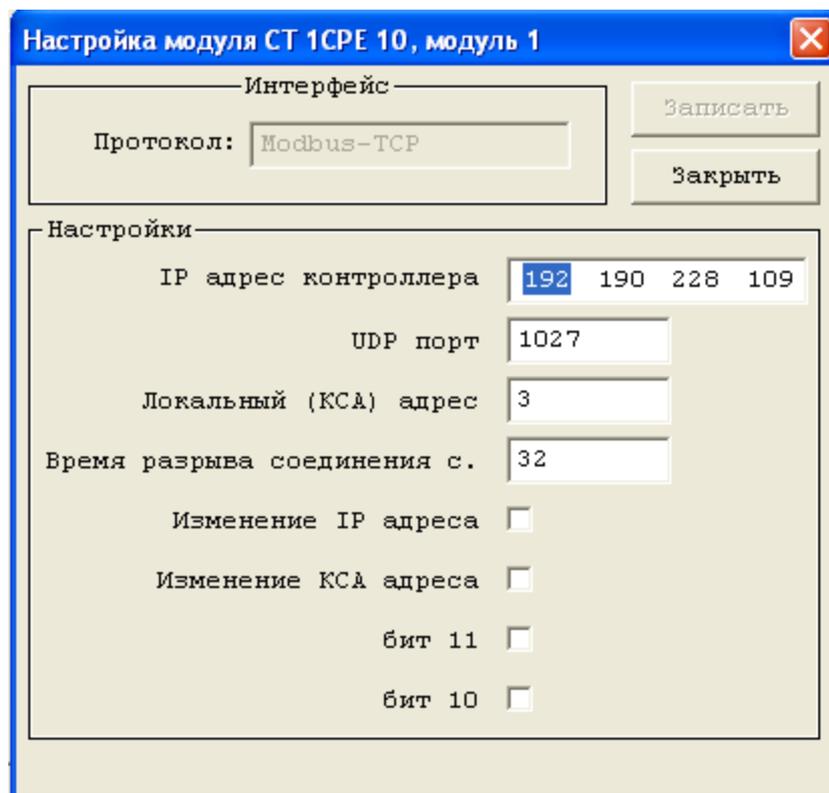
4.2.2.14. Общие настройки протоколов и настройка протокола Modbus-TCP (slave)

При применении любого протокола обмена данными (в том числе Modbus-TCP (slave)) необходимо настроить следующие параметры:

«IP адрес контроллера». IP адрес интерфейсного модуля ST1CPE10 контроллера КСА-02 в сети Ethernet, к которой он подключен. Адрес представляет собой 32-битное число из четырёх октетов, разделённых точками; каждый октет может содержать в себе до 3х цифр. При увеличении адреса в резервированной системе изменяться могут 2 последних октета адреса. То есть если адрес был равен 2.3.254.250, то получится 2.3.255.26

«**UDP порт**». Для некоторых протоколов - «**TCP порт**». Порт UDP (TCP), открытый для обмена данными. Как уже отмечалось ранее, порт TCP 502 открыт для обмена независимо от настроек.

«**Локальный (КСА) адрес**». Локальный адрес контроллера при работе в режимах slave, client.



4.2.2.14.. Настройка Modbus-TCP для модуля СТ1СРЕ10.

«**Время разрыва соединения**». Используется для регистрации разрыва соединения.

«**Изменение IP адреса**», «**Изменение КСА адреса**». Флажки «Изменение адреса» устанавливаются в случае, если включена система резервирования и один из главных процессоров имеет режим ведущего, а второй — ведомого.

Флажок «Изменение адреса» не устанавливается, если система резервирования включена, но оба главных процессора должны иметь одинаковый адрес для обмена с системой верхнего уровня, либо если система резервирования не включена. Кроме этого IP и КСА адреса можно задать неизменными в конфигурации корзины главного процессора с помощью сервисной программы, в этом случае устанавливать флажок «Изменение адреса» не имеет смысла.

Если флажок «Изменение адреса» установлен — при изменении режима сетевой адрес главного процессора, ставшего ведомым, увеличивается на 32.

«бит 11», «бит 10». Флажки необходимы для обеспечения совместимости с предыдущими версиями ядра.

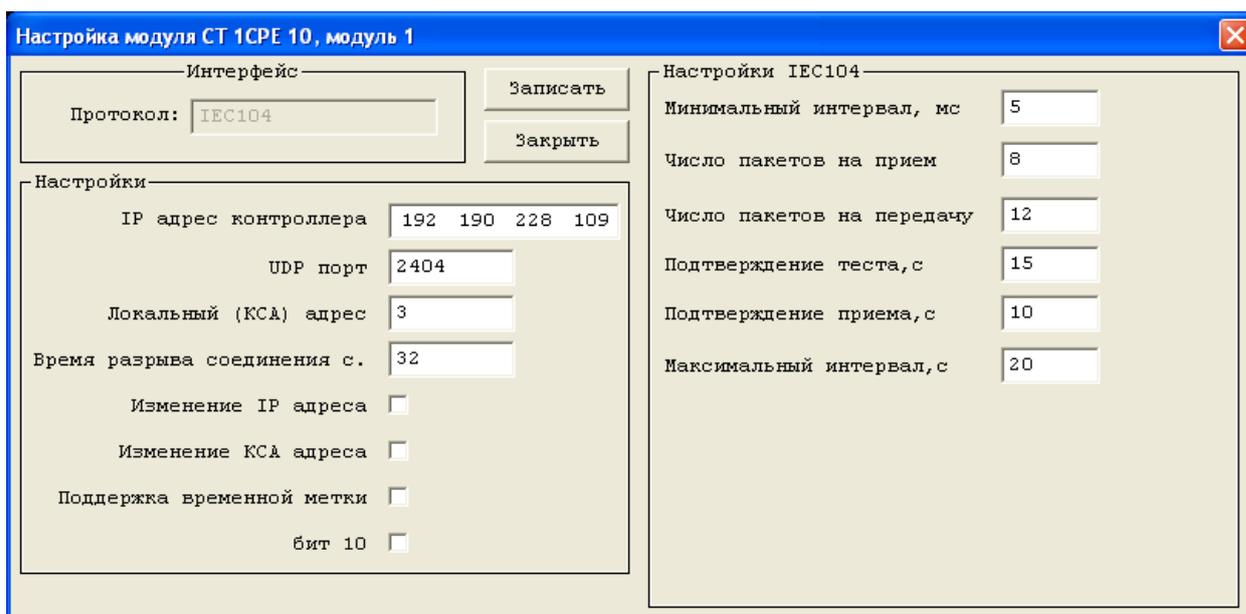
Описанные настройки являются общими для всех протоколов.

4.2.2.15. Особенности настройки протокола Proton3-IP (slave)

Настройки протокола Proton3 полностью совпадают с настройками для протокола Modbus-TCP (slave). На более низком уровне, протокол может использоваться в двух вариантах: Proton3-IP - протокол уровня IP, и Proton3-UDP - протокол уровня UDP. При использовании протокола в варианте Proton3-UDP настройка **«UDP порт»** обозначает номер UDP порта, открытого для обмена данными по протоколу.

Протокол Proton3 поддерживается только модулем, установленным в 3 слоте контроллера.

4.2.2.16. Настройка протокола МЭК870-5-104 (IEC60870-104)



Интерфейс	
Протокол:	IEC104
[Записать] [Закрыть]	

Настройки	
Настройки контроллера	
IP адрес контроллера	192 190 228 109
UDP порт	2404
Локальный (КСА) адрес	3
Время разрыва соединения с.	32
Изменение IP адреса	<input type="checkbox"/>
Изменение КСА адреса	<input type="checkbox"/>
Поддержка временной метки	<input type="checkbox"/>
бит 10	<input type="checkbox"/>
Настройки IEC104	
Минимальный интервал, мс	5
Число пакетов на прием	8
Число пакетов на передачу	12
Подтверждение теста, с	15
Подтверждение приема, с	10
Максимальный интервал, с	20

4.2.2.16. Настройка протокола МЭК870-5-104 для модуля СТ1CPE10.

Особенности общих настроек:

«UDP порт». Порт, открытый для обмена данными. В приложениях принято использовать порт 2404. Однако, это требование не является строго обязательным.

«Поддержка временной метки». Флаг в базе данных занимает место стандартной настройки "бит 11". При установке флага, передача одиночных данных сопровождается

временной меткой. При сброшенном флаге данные передаются без временной метки. При передаче групповых данных, временная метка никогда не передается.

Индивидуальные настройки протокола имеют симметричный характер, т.е. относятся как к контроллеру (client), так и к его корреспонденту (server). Соответственно, они должны быть одинаковы у клиента и сервера.

«Минимальный интервал». Минимальный интервал между пакетами. При большом количестве данных, интервал между передаваемыми пакетами не может быть меньше указанного интервала. При работе на прием, в случае, если интервал между двумя последовательно принятыми пакетами меньше указанного значения, второй пакет не обрабатывается.

«Число пакетов на прием». Максимальное число пакетов, которое устройство может принять без подтверждения. В случае, когда число принятых пакетов (без подтверждения) достигло назначенной величины, устройство обязано немедленно подтвердить полученные данные.

«Число пакетов на передачу». Максимальное число пакетов, которое устройство может передать без подтверждения (для сервера - передать в один адрес). В случае, когда число переданных пакетов (без подтверждения) достигло назначенной величины, устройство обязано прекратить передачу данных до получения подтверждения.

«Подтверждение теста». Интервал, в течение которого устройство обязано ответить на тестовую посылку. Если устройство не получило ответа на тестовый запрос в течение указанного интервала, фиксируется тайм-аут по пропаданию связи.

«Подтверждение приема». Интервал, в течение которого устройство обязано подтвердить получение данных (независимо от количества неподтвержденных пакетов). Если устройство не получило подтверждения переданных данных в течение указанного интервала, неподтвержденные пакеты считаются не принятыми и должны быть переданы снова.

«Максимальный интервал». Максимальный интервал между пакетами. При отсутствии передач в течение указанного интервала, должен быть отправлен тестовый пакет.

4.2.2.17. Настройка протокола Modbus-TCP (master)

Настройка модуля СТ 1СРЕ 10, модуль 1

Интерфейс

Протокол: Modbus-TCP-master

Настройки

		IP				КСА
Основной адрес		192	190	228	109	3
Первый доп. адрес	<input checked="" type="checkbox"/>	13	10	113	15	20
Второй доп. адрес	<input type="checkbox"/>	0	0	0	0	0
Третий доп. адрес	<input type="checkbox"/>	0	0	0	0	0
Время разрыва соединения (сек)						32
Номер порта						3300
Число повторов при ретрансляции						8
Тайм-аут по ретрансляции данных (сек)						5

При выборе протокола "Modbus-master" контроллер запрашивает данные у ведомых устройств по протоколу "Modbus-TCP". Ведомые устройства могут находиться как в основной сети контроллера, так и в других (дополнительных) сетях, при наличии физической достижимости этих сетей. При работе в дополнительной сети контроллер использует адресацию, принятую в этой сети. Маска адреса дополнительной сети задается тремя первыми октетами IP адреса. Поддерживается до трех дополнительных сетей.

«Основной адрес IP». Собственный IP-адрес контроллера в основной сети.

«Основной адрес КСА». Локальный собственный адрес контроллера в основной сети. Используется при запросе данных у контроллера другим устройством сети (протокол Modbus-TCP-slave).

Флаги **«Первый (второй, третий) доп. адрес».** Установка флагов приводит к поддержке контроллером дополнительных сетей.

«Первый (второй, третий) доп. адрес IP». Собственный IP-адрес контроллера в дополнительной сети.

«Первый (второй, третий) доп. адрес КСА». Локальный собственный адрес контроллера в дополнительной сети. Используется при запросе данных у контроллера другим устройством дополнительной сети (протокол Modbus-TCP-slave).

«Номер порта». Номер порта контроллера для получения данных. Не может быть равен 502, а также другим номерам, зарезервированным в стандартах для конкретных применений.

«Число повторов при ретрансляции». Максимальное число повторов передачи пакетов при отсутствии их подтверждения.

4.2.3. Сообщения редактора параметров контроллера КСА-02 об ошибках

«Область резервирования LOGIC больше буфера LOGIC»

Возникает при определении области резервирования для передачи данных в буфере входных данных типа LOGIC в разделе «Резервирование» в случае, если область, определенная значениями начального адреса и размера буфера, выходит за пределы входного буфера данных (LOGIC), определенного в разделе «Буферы».

« Область резервирования INTEGER больше буфера INTEGER»

Возникает при определении области резервирования для передачи данных в буфере входных данных типа INTEGER в разделе «Резервирование» в случае, если область, определенная значениями начального адреса и размера буфера, выходит за пределы входного буфера данных (INTEGER), определенного в разделе «Буферы».

«Область резервирования FLOAT больше буфера FLOAT»

Возникает при определении области резервирования для передачи данных в буфере входных данных типа FLOAT в разделе «Резервирование» в случае, если область, определенная значениями начального адреса и размера буфера, выходит за пределы входного буфера данных (FLOAT), определенного в разделе «Буферы».

«Область внутренних данных больше внутр. буфера»

Возникает при определении области резервирования для передачи данных во внутреннем буфере данных «Резервирование» в случае если область, определенная значениями начального адреса и размера буфера выходит за пределы внутреннего буфера данных, определенного в разделе «Буферы».

«Недостаточно памяти для буферов данных»

Возникает при определении размера входных и выходных буферов в разделе «Буферы» в случае если суммарный размер всех типов буферов превышает 8192 слов.

«Неверное значение входного буфера типа данных FLOAT»

Возникает при определении размера буфера входных данных типа FLOAT в разделе «Буферы» в случае если значение размера буфера превышает 4096.

«Неверное значение входного буфера типа данных INTEGER»

Возникает при определении размера буфера входных данных типа INTEGER в разделе «Буферы» в случае если значение размера буфера превышает 8192.

«Неверное значение входного буфера типа данных LOGIC»

Возникает при определении размера буфера входных данных типа LOGIC в разделе «Буферы» в случае если значение размера буфера превышает 131072.

«Неверное значение выходного буфера типа данных FLOAT»

Возникает при определении размера буфера выходных данных типа FLOAT в разделе «Буферы» в случае если значение размера буфера превышает 4096.

«Неверное значение выходного буфера типа данных INTEGER»

Возникает при определении размера буфера выходных данных типа INTEGER в разделе «Буферы» в случае если значение размера буфера превышает 8192.

«Неверное значение выходного буфера типа данных LOGIC»

Возникает при определении размера буфера выходных данных типа LOGIC в разделе «Буферы» в случае если значение размера буфера превышает 131072.

«Неверные параметры буфера внутренних данных»

Возникает при определении размера буфера внутренних данных в разделе «Буферы», в случае если значение размера буфера превышает 8192.

«Неверные параметры адреса портов»

Возникает при определении базового адреса портов, в случае если значение базового адреса портов выходит за пределы диапазона 1..255.

«Настройки драйвера MKSA-Slave для COM1 превышают размер внутреннего буфера»

Возникает при определении области для передачи или приёма данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave) при конфигурировании порта COM1, в

случае если область, определенная значениями начала и размера буфера выходит за пределы внутреннего буфера данных, определенного в разделе «Буферы».

«Настройки драйвера MKSA-Slave для COM2 превышают размер внутреннего буфера»

Возникает при определении области для передачи или приёма данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave) при конфигурировании порта COM2, в случае если область, определенная значениями начала и размера буфера выходит за пределы внутреннего буфера данных, определенного в разделе «Буферы».

«Неверное значение количества повторов»

Возникает при определении количества повторов при неответе при настройке параметров работы драйвера по COM-порту в разделе «Драйверы» в случае если устанавливаемое значение количества повторов выходит за пределы диапазона 0...15.

«Неверное значение времени ожидания ответа»

Возникает при определении времени ожидания ответа при настройке параметров работы драйвера по COM-порту в разделе «Драйверы», в случае если устанавливаемое значение времени ожидания ответа выходит за пределы диапазона 0...106,5 мсек.

«Неверное значение начала буфера для передачи»

Возникает при определении значения начала буфера для передачи данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave), в случае если устанавливаемое значение начала буфера для передачи данных выходит за пределы 0...255.

«Неверное значение размера буфера для передачи»

Возникает при определении значения размера буфера для передачи данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave), в случае если устанавливаемое значение размера буфера для передачи данных выходит за пределы 0...255.

«Неверное значение начала буфера для приёма»

Возникает при определении значения начала буфера для приёма данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave), в случае если устанавливаемое значение начала буфера для приёма данных выходит за пределы 0...255.

«Неверное значение размера буфера для приёма»

Возникает при определении значения размера буфера для приёма данных в окне настройки протокола MKSA в режиме Slave (MKSA-Slave), в случае если устанавливаемое значение размера буфера для приёма данных выходит за пределы 0...255.

«Недостаточен размер буфера (тип буфера). Требуется (размер буфера)»

Сообщение подпрограммы конвертации типов данных протокола МЭК870 к типам данных контроллера. Возникает, если объявленный размер буфера указанного типа недостаточен для конвертации заявленного количества данных МЭК870. Сообщается минимальный размер буфера, достаточный для успешной конвертации данных.

4.3. Редактор языка «FBD_Builder»

4.3.1. Общие сведения

Редактор языка «FBD_Builder» предназначен для создания в графическом виде и редактирования алгоритма рабочей программы для контроллера КСА-02 и позволяет добавлять, перемещать, копировать и удалять объекты FBD (Function Block Diagram) — блоки и связи, точки разветвлений связи, начальные данные входов функций и комментарии. FBD является языком программирования стандарта IEC61131-3.

4.3.2. Команды «FBD_Builder»

Программа «FBD_Builder» располагает набором разнообразных команд, необходимых для работы по созданию и редактированию алгоритмов.

Кнопки и строки вызова этих команд расположены в программе наиболее удобным для пользователя образом.

Все команды, необходимые для разработки схемы алгоритма, а также сервисные команды, находятся в **меню программы**.

Команды, которые наиболее часто используются при построении схемы, вынесены на **панель инструментов** окна программы в виде кнопок. Туда же помещены кнопки включения/выключения существующих режимов работы программы.

Для быстрого вызова наиболее часто используемых команд существует также **контекстное меню**, вызываемое щелчком правой кнопкой мыши по полю программы или объекту.

4.3.2.1. Меню «FBD_Builder»

В списках пунктов меню содержатся все необходимые команды программы «FBD_Builder», используемые при построении и редактировании алгоритмов.

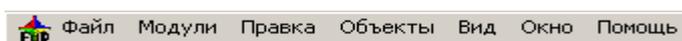


Рис. 4.3.2.1.1. Так выглядит панель меню программы «FBD_Builder».

Краткое описание состава и назначения пунктов меню приведено в таблице:

	Подпункт меню (клавиши вызова)	Описание команды
Файл		
	Создать... (Ctrl+N)	Создать новый проект

	Открыть... (Ctrl+O)	Открыть существующий проект
	Сохранить (Ctrl+S)	Сохранить открытый проект
	Сохранить как...	Сохранить открытый проект с выбором имени файла
	Закрыть (Ctrl+Q)	Закрыть текущий открытый проект
	Печать... (Ctrl+P)	Распечатать диаграмму открытого проекта
	Свойства проекта	Открыть окно для определения свойств проекта
	...	Быстрый доступ к ранее открытым проектам
	Выход (Alt+F4)	Закончить работу с редактором «FBD_Builder»
Модули (может отсутствовать)		
	Этот пункт меню содержит список названий установленных к программе дополнительных вспомогательных небольших программ. Если ни одна такая программа не установлена, пункт меню «Модули» также отсутствует.	
Правка		
	Вернуть (Ctrl+Z)	Отменить последнее действие
	Восстановить (Ctrl+Y)	Восстановить последнее отмененное действие
	Вырезать (Ctrl+X)	Удалить выделенное со схемы и разместить в буфере обмена
	Копировать (Ctrl+C)	Скопировать выделенное на схеме в буфер обмена
	Вставить (Ctrl+V)	Вставить в схему данные из буфера обмена
	Удалить (Del)	Удалить со схемы выделенные объекты
	Замена (F7)	Вызывает окно поиска и замены текста
Объекты		
	Поиск (Ctrl+F)	Поиск объектов (только для объектов с заданным полем «Текст»)
	Редактор параметров контроллера (Ctrl+J)	Вызвать окно настройки параметров
	Компилировать (F9)	Компилировать открытый проект
	Список сообщений (F8)	Вызов окна с сообщениями компилятора
	Теги (F4)	Вызов окна «Теги»
	Составные функции (Ctrl+D)	Вызов списка используемых в проекте составных функций
	Добавить блок... (Ctrl+B)	Вызвать окно для добавления в схему нового блока

	Добавить контакт (Ctrl+K)	Добавить новый контакт
	Добавить рамку (Ctrl+R)	Добавить новую рамку
	Свойства	Вызов окна свойств для текущего выбранного объекта (объектов)
Вид		
	Сдвиг (H)	Включить режим перемещения по схеме — сдвиг
	Увеличить	Увеличить отображаемый масштаб схемы
	Уменьшить	Уменьшить отображаемый масштаб схемы
	Перейти... (Ctrl+G)	Перемещение по схеме посредством прямого ввода координат X, Y
	Запомнить место (F2)	Запомнить масштаб и координаты для быстрого перемещения (F3)
	На место (F3)	Отобразить диаграмму в том виде, в каком она была запомнена в последний раз (F2)
	Полосы прокрутки	Включить на схеме полосы прокрутки (альтернатива механизма «Сдвиг»)
	Настройки	Глобальные настройки редактора (шрифт, наличие сетки и т.п.)
Окно		
В качестве подпунктов этого меню выступают заголовки открытых диаграмм, что позволяет быстро переключаться между ними, когда открыто несколько разных диаграмм.		
Примечание: наименование файла диаграммы и название самой диаграммы могут не совпадать.		
Помощь		
	Помощь (F1)	Отобразить файл помощи
	О программе...	Отобразить окно с информацией о программе

Действие любой выполненной команды редактирования до момента сохранения файла алгоритма можно отменить, используя команду «Вернуть».

4.3.2.2. Панель инструментов «FBD_Builder»

На панели инструментов содержится набор кнопок с отображением наиболее часто употребляемых команд, а также кнопки включения/выключения режимов работы программы.



Рис. 4.3.2.2.1. Панель инструментов с кнопками быстрого доступа к командам.

Кнопки включения/выключения режимов работы:

	Команда (клавиши вызова)	Описание команды
	Привязка (1)	Включить режим привязки к тегам
	Установка инверсии (2)	Включить режим установки инверсии для логических входов
	Рисование связей (3)	Включить режим рисования новых связей
	Показывать сетку (4)	Включает/отключает сетку
	Показывать пояснения (Ctrl+F1)	Включить режим отображения подсказок к объектам диаграммы

4.3.2.3. Контекстное меню «FBD_Builder»

Контекстное меню вызывается щелчком правой кнопкой мыши на поле или объекте и содержит команды, чаще всего вызываемые в процессе создания и редактирования схемы алгоритма. Кроме того, здесь находятся команды, относящиеся к редактированию связей и расположения объектов на поле алгоритма.

Ниже рассмотрены команды, которые можно вызвать только из контекстного меню:

- **Дублировать (F5)** — позволяет копировать и поместить на поле выделенный объект или несколько объектов, не помещая их для этого в буфер обмена (clipboard) операционной системы.
- **Отсоединить (Backspace)** — отсоединяет выделенный объект от привязанных к нему связей. Если выделен участок связи — он отсоединится от того соседнего участка, ближе к которому поставлен курсор.

	Добавить блок	Ctrl+B
	Добавить контакт	Ctrl+K
	Добавить рамку	Ctrl+R
	Вырезать	Ctrl+X
	Копировать	Ctrl+C
	Вставить	Ctrl+V
	Удалить	Del
	Дублировать	F5
	Отсоединить	BkSp
	Соединить	=
	Выделить все	Ctrl+A
	Выделить объекты под рамкой	
	Стоп	Esc
	Выделить связь	
	Добавить ответвление	Ctrl+L
	Создать разрыв	Ctrl+T
	Выравнивание по объекту	F10
	Режим	▶
	Поиск	Ctrl+F
	Свойства	

Рис.4.3.2.3.1. Контекстное меню

- **Соединить (=)** — используется для соединения пристыкованных друг к другу разорванных концов связи.
- **Выделить все (Ctrl+A)** — выделяет все объекты на диаграмме.
- **Выделить объекты под рамкой** — с помощью этой команды можно выделить все объекты, находящиеся внутри выделенной рамки.
- **Стоп (Esc)** — команда, отменяющая выделение объекта или группы объектов — как выделение объектов с помощью мыши, так и выделение тегов и ошибочных конструкций, помеченных программой мигающим красно-синим цветом.
- **Выделить связь** — в применении к выделенному участку связи эта команда выделяет все остальные участки и ответвления, относящиеся к этой связи.
- **Добавить ответвление (Ctrl+L)** — добавляет к связи ответвление из точки, на которую установлен курсор.
- **Создать разрыв (Ctrl+T)** — разрывает связь в том месте, где стоит курсор.
- **Выравнивание по объекту (F10)** — отображает на поле линии координат, точкой отсчёта которых избирается верхний левый угол выделенного объекта. По этим линиям вручную выравниваются остальные объекты. Клавишей F10 можно устанавливать этот режим для выделенных объектов и отключать для невыделенных.
- **Режим** — повторение в контекстном меню кнопок панели инструментов, осуществляющих включение/отключение режимов работы программы «FBD_Builder».

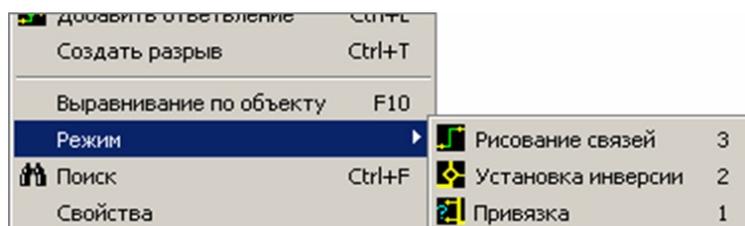


Рис. 4.3.2.3.2. Пункт «Режим»

контекстного меню

4.3.3. Свойства проекта

При создании проекта до его компиляции следует определить его свойства.

Нужно открыть пункт меню «Файл» и выбрать подпункт «Свойства проекта».

В рамках программы «FBD_Builder» следует различать проект алгоритма и проект составной функции, поскольку и алгоритм, и составная функция — самостоятельные файлы с собственными особенностями и свойствами.

4.3.3.1. Свойства проекта алгоритма

При выборе в меню «Файл» подпункта «Свойства проекта» откроется следующее окно:

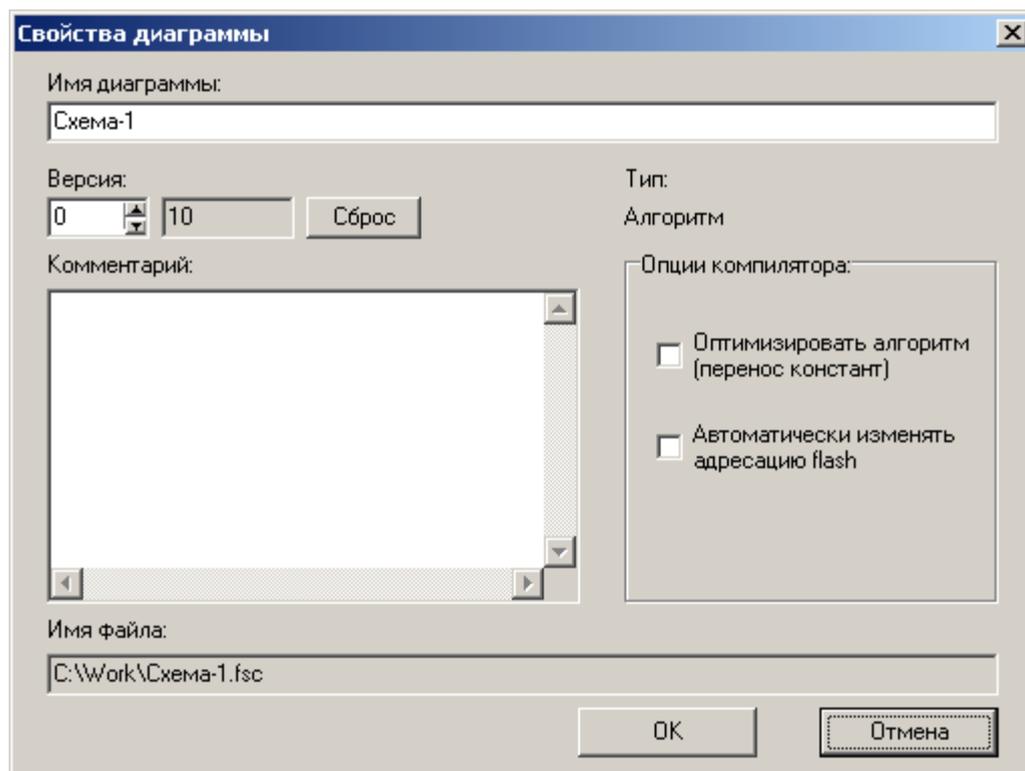


Рис. 4.3.3.1.1. Окно «Свойства проекта» для проекта алгоритма.

Поле «Имя диаграммы»: сюда следует вписать название проекта, которое будет отображаться на заголовке окна программы при открытии проекта. Это имя может отличаться от имени файла проекта *.fsc.

Комментарий — это текстовое пояснение к проекту. Комментарий несёт только информационную нагрузку и не влияет на выполнение алгоритма.

«Оптимизировать алгоритм (перенос констант)»: по умолчанию флажок снят. Это значит, что при компиляции блок каждой функции будет рассматриваться только сам по себе, а данные, вынесенные за его пределы, будут проигнорированы (например, если тег значения входа вынесен с помощью контакта за пределы блока). Если флажок установить, эти данные будут обнаруживаться компилятором и учитываться при компиляции.

«Автоматически изменять адресацию flash»: каждый тег в проекте должен иметь свой уникальный адрес. При создании тега этот адрес нулевой, и может быть либо изменён вручную, либо проставляется компилятором. Если флажок «Автоматически изменять адресацию flash» снят (по умолчанию), адреса следует задать вручную. Если флажок установлен, компилятор сам назначит эти адреса, в т.ч. перепишет те, которые уже были заданы вручную.

Имя файла: если проект после создания уже был сохранён на жёсткий диск, то в этом поле отображается имя файла, содержащего проект. Оно может отличаться от имени самого проекта.

4.3.3.2. Свойства проекта составной функции

При выборе в меню «Файл» подпункта «Свойства проекта» откроется следующее окно:

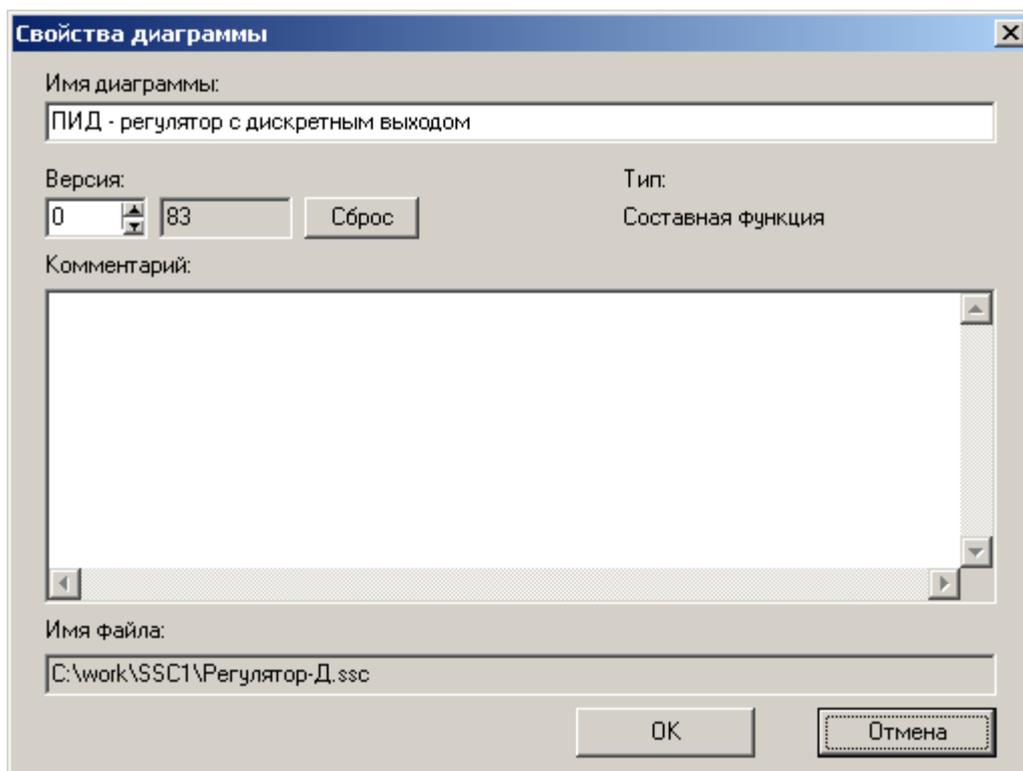


Рис. 4.3.3.2.1. Окно «Свойства проекта» для проекта составной функции

Поле «Имя диаграммы»: сюда следует вписать название проекта, которое будет отображаться на заголовке окна программы при открытии проекта. Это имя может отличаться от имени файла *.fsc, содержащего проект.

Комментарий — это текстовое пояснение к проекту. Комментарий несёт только информационную нагрузку и не влияет на выполнение алгоритма.

Имя файла: если проект после создания уже был сохранён на жёсткий диск, то в этом поле отображается имя файла, содержащего проект. Оно может отличаться от имени самого проекта.

Флажки «Оптимизация алгоритма (перенос констант)» и «Автоматически изменять адресацию flash» в окне свойств проекта составной функции отсутствуют, поскольку файл составной функции не компилируется, а лишь проверяется компилятором на наличие ошибок.

4.4. Порядок построения схемы алгоритма в программе «FBD_Builder»

4.4.1. Вставка нового блока функции, перемещение и удаление блоков

Команда «Добавить блок»  позволяет выбирать блок из библиотеки и вставлять его на поле алгоритма. При вызове этой команды на экране монитора появится окно «Блоки». Вкладка «Функции» отображает список всех имеющихся в библиотеке функций. На вкладке «Составные функции» отображается список всех задействованных в алгоритме составных функций (если таковые есть).

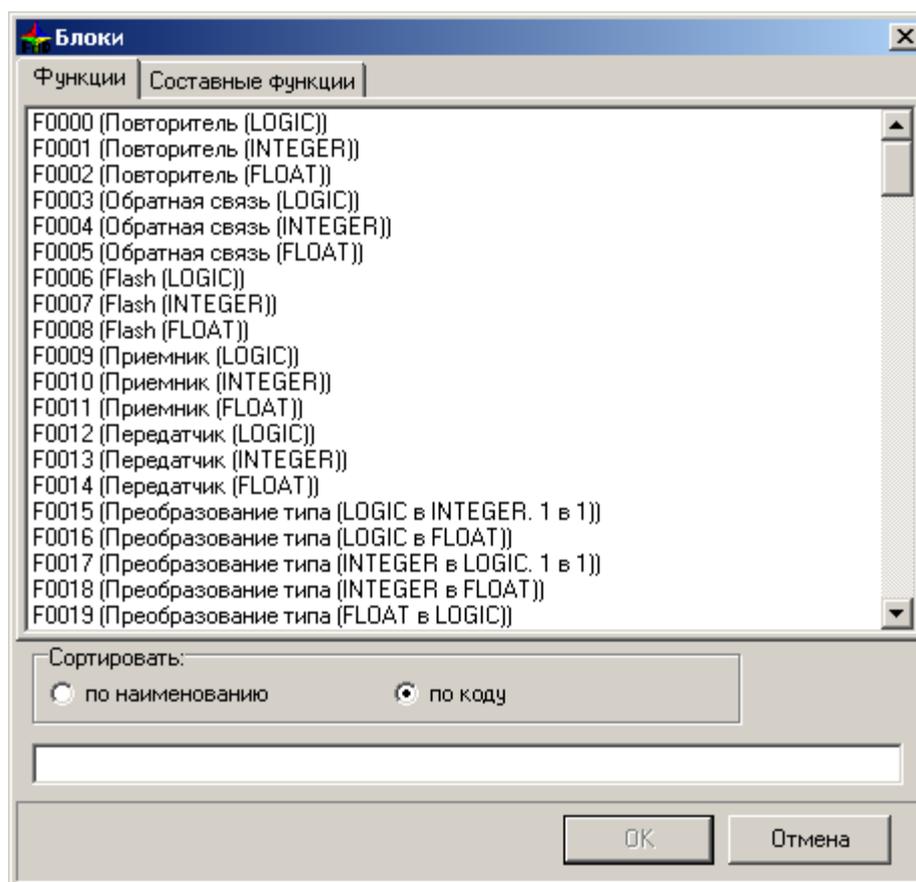


Рис. 4.4.1.1. Окно «Блоки»

В помощь пользователю в нижней части окна есть два флажка, позволяющие отсортировать отображённый список по наименованию функции или по её коду.

Текстовое поле под флажками дублирует название функции из списка, на которую установлен курсор.

Вкладка и список составных функций устроены аналогично.

В списке выбирается необходимая функция, и при нажатии кнопки ОК изображение блока, содержащего эту функцию, появится на поле программы и будет следовать за курсором

мышь. Его положение на поле фиксируется щелчком мыши. Отменить выбор блока до его фиксации можно клавишей Esc.

Чтобы изменить положение блока на поле, нужно щёлкнуть его мышью и, удерживая кнопку, переместить его на желаемую позицию.

Чтобы удалить блок, нужно щёлкнуть его мышью для выделения и выбрать команду «Удалить» (кнопка ).

Чтобы удалить группу элементов, нужно выделить её мышью (при этом захватывать предназначенные к удалению элементы следует целиком). Затем также выбрать команду «Удалить».

4.4.2. Создание связи между выходами и входами блоков

Для создания связей между блоками и добавления ответвлений к уже существующим связям существует режим «Рисование связей».

Точкой начала связи может быть вход блока, выход блока или другая связь.

Выберите на панели инструментов кнопку  и нажмите её. Эта команда включает возможность присоединения связей ко входам и выходам блоков.

Чтобы присоединить связь к выходу блока, следует установить курсор строго на изображение этого выхода и щёлкнуть его мышью. После этого за курсором потянется связь в виде зелёной линии, которую надо привязать ко входу другой функции или к другой связи щелчком мыши по месту привязи.

Чтобы начать рисовать связь от другой связи, нужно выделить имеющуюся связь, установить на ней курсор в точку начала предполагаемой связи и выбрать в контекстном меню команду «Добавить ответвление» (клавиши Ctrl+L).

При рисовании связи на поле алгоритма можно изменять её направление в соответствии с расположением объектов на поле, щелчками мыши фиксируя точки изменения направления.

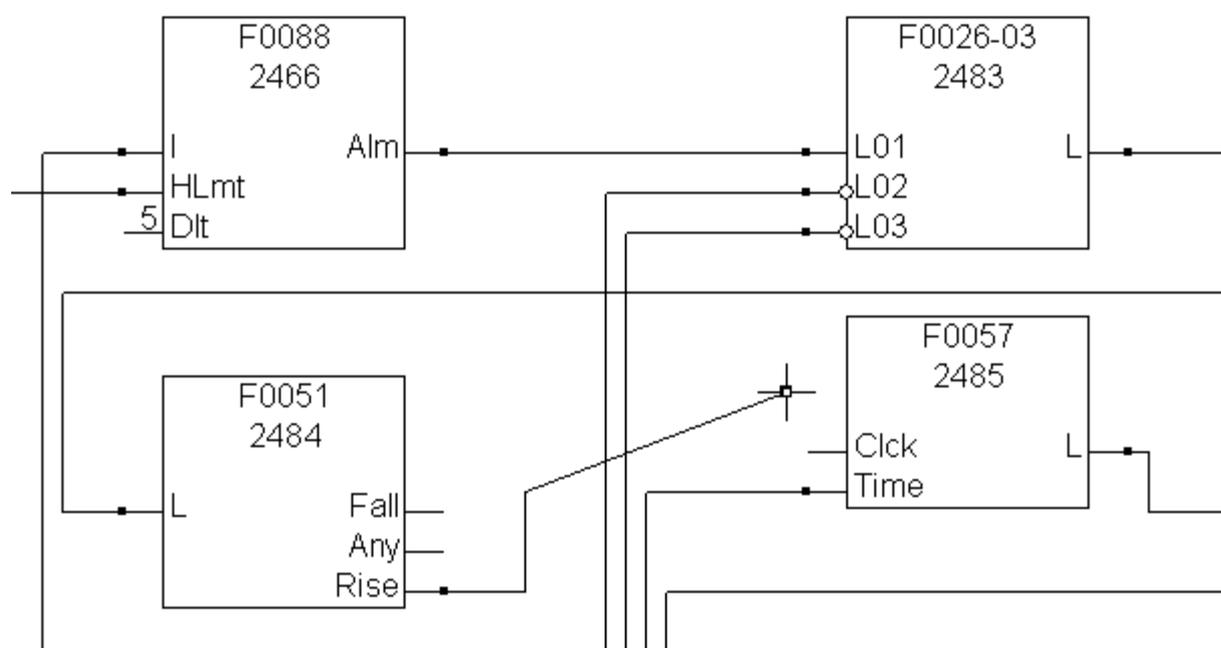


Рис. 4.4.2.1. Пример рисования связи.

Отменить предыдущий шаг рисования связи можно двумя способами: нажатием на клавиатуре клавиши «Backspace» или комбинацией клавиш «Ctrl+Z». Различие между этими способами состоит в том, что при нажатии клавиши «Backspace» происходит последовательное удаление нарисованных участков связи, а при нажатии комбинации клавиш «Ctrl+Z» происходит последовательная отмена действий пользователя.

Иногда в процессе работы бывает нужно оборвать связь, оставив её не привязанной ни к какому месту. Во время её рисования это можно сделать двойным щелчком мыши или нажатием клавиши «Escape» на клавиатуре. Разорвать уже имеющуюся связь можно выделив её, установив курсор на предполагаемую точку разрыва и выбрав команду «Создать разрыв».

Чтобы удалить не всю связь, а её часть, нужно выделить удаляемую часть и нажать клавишу «Delete» или кнопку . Также можно выделить один участок и нажатием клавиши «Backspace» отсоединять ненужные участки один за другим, по завершении чего дважды щёлкнуть мышью по схеме или нажать клавишу «Escape».

При несоблюдении правил рисования связей могут возникнуть различные ошибки, о которых редактор предупреждает следующими сообщениями:

«Кольцо!» — связь не должна присоединяться к участку связи, который уже привязан ко входу того блока, от которого в данный момент рисуется связь;

«**Типы несовместимы!**» — связь должна соединять выходы блоков с входами одинакового типа;

«**Выходы соединять нельзя!**» — связь может соединять выход только с входами.

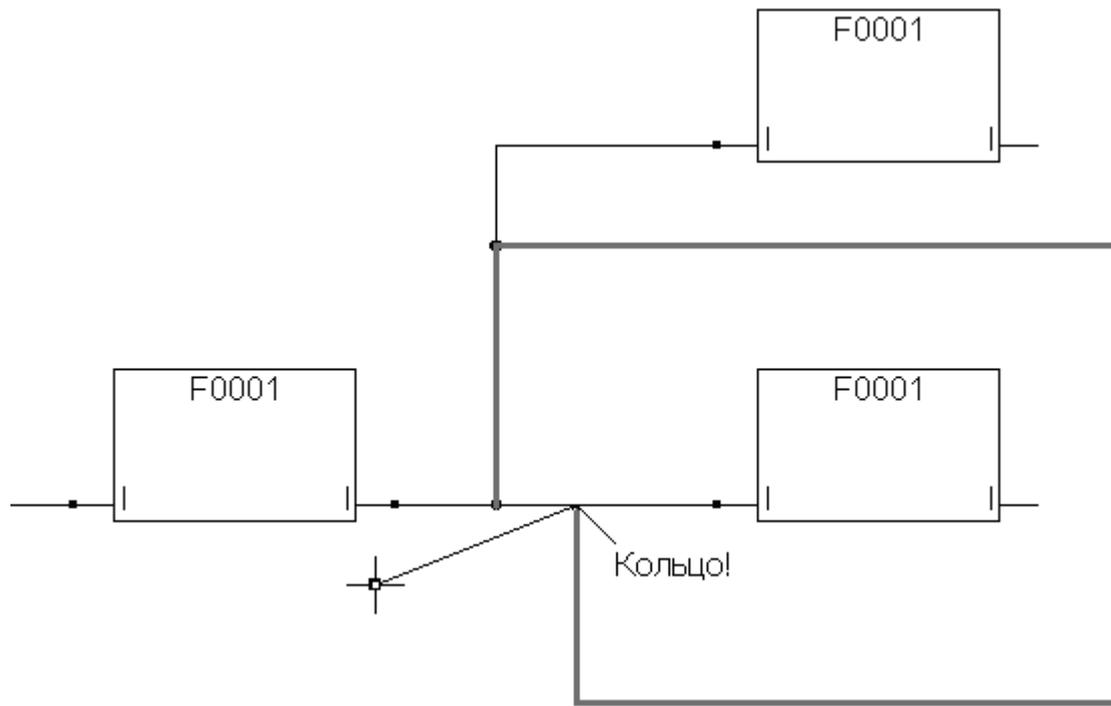


Рис. 4.4.2.2. Пример отображения ошибки «Кольцо!», возникшей при попытке привязать ответвление связи к тому же участку, от которого оно отведено (ошибочная ветка связи выделена толстой линией).

4.4.3. Установка инверсии на вход функции

Если вход функции помечен признаком *инверсии*, то значения, приходящие на этот вход, при попадании в функцию будут меняться на противоположные. Инверсия применима только ко входам логического типа.

Если включить режим «Установка инверсии» (кнопка ) , станет можно инвертировать логические типы входов функции. Если в этом режиме щёлкнуть мышью вход функции, то на входе отобразится признак установившейся инверсии — кружочек в месте графического соединения входа с блоком. Если инверсия на указанном входе уже была установлена, то в этом случае она будет снята.

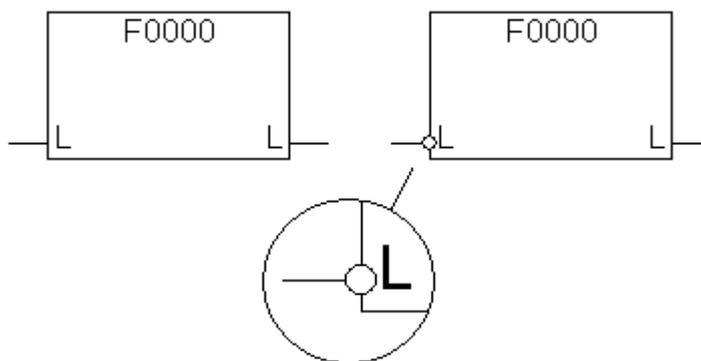


Рис. 4.4.3.1. Пример блока с инвертированным входом.

Режим «Установка инверсии» отключается при включении режима «Привязка к переменным» или «Рисование связей».

4.4.4. Вставка входного значения (привязка)

Если включить режим «Привязка», становится возможным устанавливать начальные значения на входах блоков.

Если щёлкнуть мышью тот вход блока, на который требуется записать значение, то появится окно привязки ко входу функции:

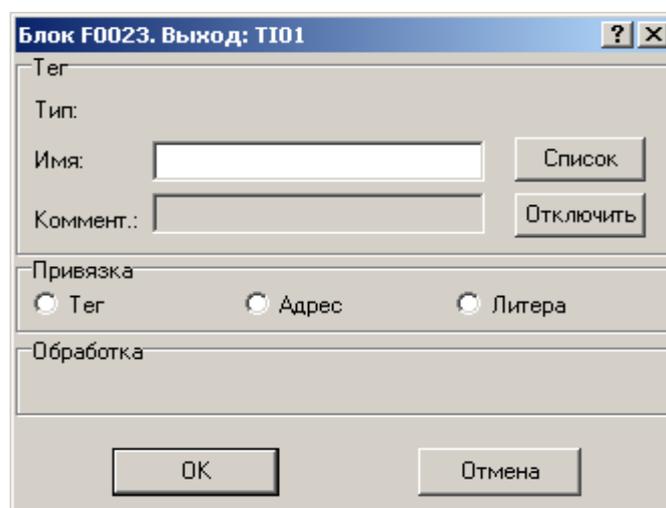


Рис. 4.4.4.1. Окно привязки входа к тегу

Чтобы привязать ко входу тег, нужно вписать его имя в поле «Имя», либо выбрать существующий из таблицы «Теги» (доступно через кнопку «Список»).

Если вписать в поле «Имя» название уже имеющегося в списке тега, то введённое название изменит цвет шрифта с чёрного на синий, а также в окне отобразятся все установленные для этого тега свойства.

Также в этом окне можно установить инверсию на данный вход или выход, если проставить флажок «Инверсия».

Если программа не обнаруживает введённое имя в списке уже имеющихся тегов, она предлагает создать новый тег с этим именем (за исключением типа «литера»).

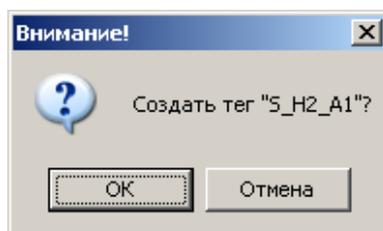


Рис. 4.4.4.2. Программа предлагает создать тег с введённым новым именем.

После создания нового тега нужно определить его параметры. Для этого следует открыть созданный тег щелчком мыши на месте его привязки. Откроется уже знакомое окно, но на этот раз заполненное информацией о созданном теге:

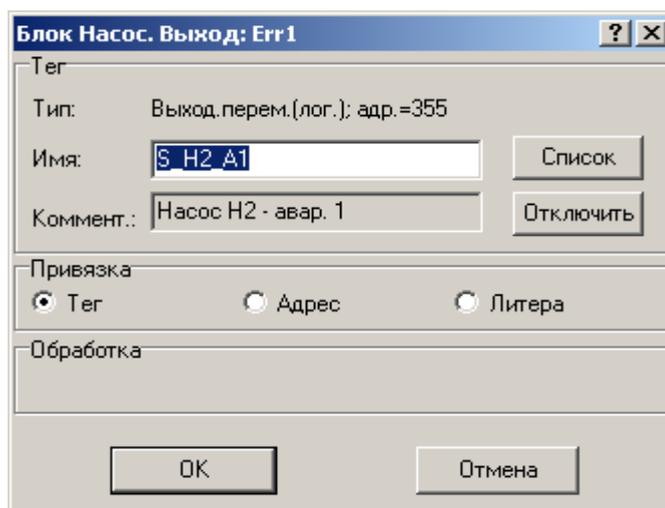


Рис. 4.4.4.3. Окно параметров тега

Чтобы назначить остальные параметры, следует открыть окно «Свойства тега». Это можно сделать двумя способами: нажать кнопку «Список» и в открывшемся списке тегов внизу нажать кнопку «Свойства», либо не открывая никаких других окон, просто щёлкнуть надпись, содержащую тип тега, которая размещена над полем с именем тега. Откроется следующее окно:

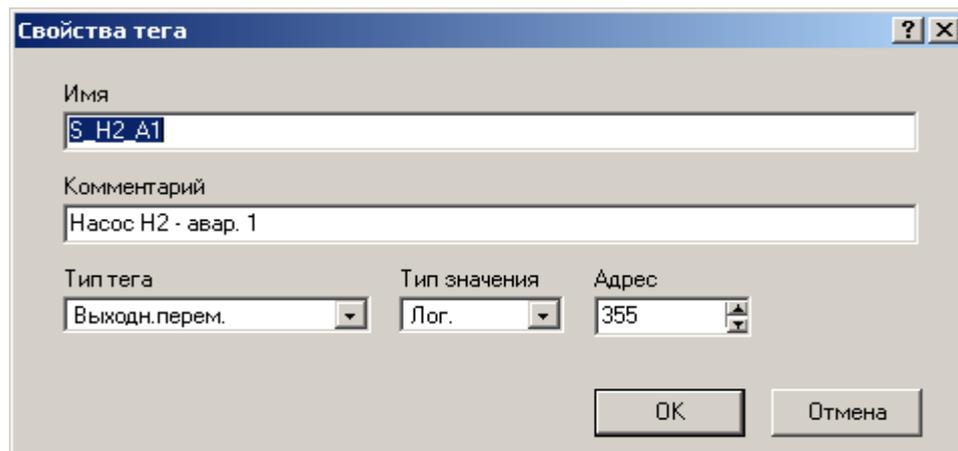


Рис. 4.4.4.4. Окно свойств тега.

В этом окне нужно выбрать тип тега, затем, если предусмотрено типом тега — тип значения, адрес (в алгоритме) или позицию (в составной функции) и значение.

При вводе значения входа блока с типом Logic необходимо устанавливать значения 0 или 1 (единственно допустимые). Если ввести другое значение, то при компиляции алгоритма компилятор выдаст ошибку.

Можно также заполнить поле «Комментарий» и добавить туда текстовое описание данного тега. Наличие комментария, его отсутствие, а также его содержание никак не влияют на работу алгоритма.

Нажатие кнопки «ОК» сохраняют свойства тега и закрывают окно.

4.4.5. Удаление входного значения (отвязка)

Чтобы отменить существующую привязку тега ко входу, в окне параметров тега существует кнопка «Отключить».

Нажатие на эту кнопку снимает с выбранного входа привязку к тому или иному тегу, но не удаляет сам тег из списка «Теги». Если удалить тег из списка, то все существующие к нему привязки также будут удалены автоматически.

4.4.6. Ввод комментариев

Комментарий — это пояснительный текст к части алгоритма, облегчающий пользователю понимание его работы и не влияющий на выполнение алгоритма. Ввести комментарий позволяет объект «Рамка». Следует выбрать команду «Добавить рамку» (кнопка ) , разместить появившийся объект на схеме щелчком мыши, при необходимости изменить его размер и местоположение. Двойной щелчок мыши по границе рамки, а также команда

«Свойства» контекстного меню открывают окно, куда вводится собственно текст комментария:

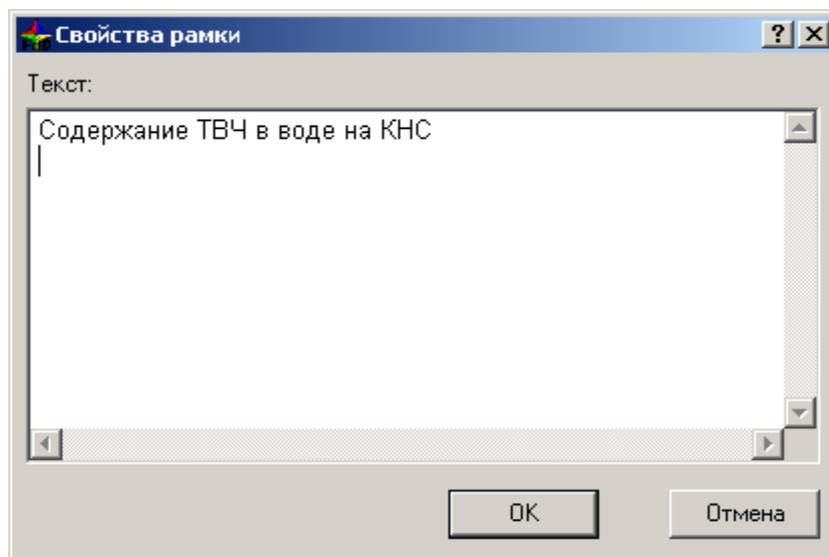
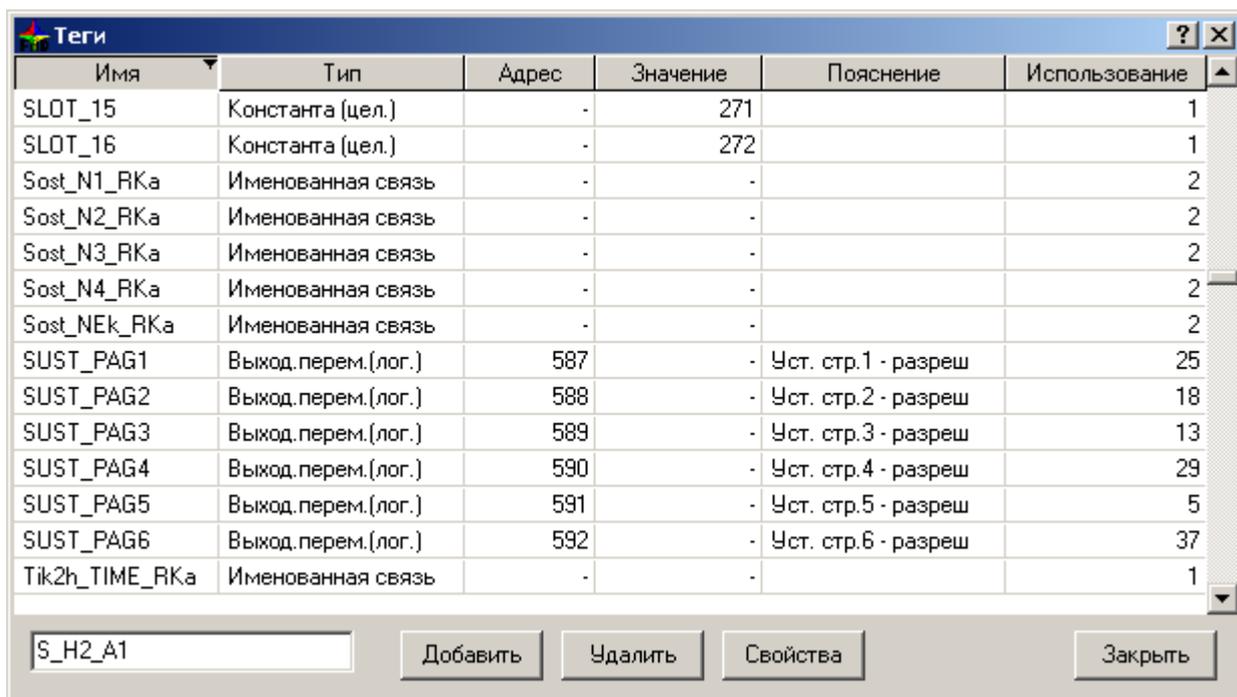


Рис. 4.4.6.1. Пример окна с введённым текстом комментария.

Существует также возможность создать текст «без рамки»: если необходимо сделать так, чтобы на экране отображался только текст, то можно с помощью мыши изменить размеры рамки до минимума, сведя её в точку. При этом рамка перестанет быть видна, и на экране останется только текст комментария.

4.4.7. Окно «Теги»

Все созданные во время разработки алгоритма теги заносятся в список, который можно вызвать, открыв окно «Теги» (пункт меню «Объекты», команда «Теги» или нажатие клавиши F4).



Имя	Тип	Адрес	Значение	Пояснение	Использование
SLOT_15	Константа (цел.)	-	271		1
SLOT_16	Константа (цел.)	-	272		1
Sost_N1_RKa	Именованная связь	-	-		2
Sost_N2_RKa	Именованная связь	-	-		2
Sost_N3_RKa	Именованная связь	-	-		2
Sost_N4_RKa	Именованная связь	-	-		2
Sost_NEk_RKa	Именованная связь	-	-		2
SUST_PAG1	Выход.перем.(лог.)	587	-	Уст. стр.1 - разреш	25
SUST_PAG2	Выход.перем.(лог.)	588	-	Уст. стр.2 - разреш	18
SUST_PAG3	Выход.перем.(лог.)	589	-	Уст. стр.3 - разреш	13
SUST_PAG4	Выход.перем.(лог.)	590	-	Уст. стр.4 - разреш	29
SUST_PAG5	Выход.перем.(лог.)	591	-	Уст. стр.5 - разреш	5
SUST_PAG6	Выход.перем.(лог.)	592	-	Уст. стр.6 - разреш	37
Tik2h_TIME_RKa	Именованная связь	-	-		1

Рис. 4.4.7.1. Окно «Теги».

Окно имеет 6 столбцов:

- **Имя:** содержит имена тегов алгоритма;
- **Тип:** указывает, к какому типу принадлежит каждый тег;
- **Адрес:** отображает адрес тега (если есть);
- **Значение:** отображает значение тега (если есть);
- **Пояснение:** отображает текстовое пояснение к тегу (если есть);
- **Использование:** указывает, сколько раз тег использован в диаграмме алгоритма.

Если нужно отсортировать строки, следует щёлкнуть мышью заголовок столбца, и строки отсортируются по алфавиту по значениям этого столбца.

В левой нижней части окна расположено поле поиска по имени тега. Оно доступно только в том случае, когда список тегов в таблице отсортирован по столбцу «Имя». При работе со списком оно отображает имя тега, на котором стоит курсор мыши.

Оно также может быть использовано для поиска тега по имени. Если вводимое имя уже есть в списке тегов, то по мере введения символов этого имени в поле поиска курсор сам будет устанавливаться в списке на подходящее имя тега.

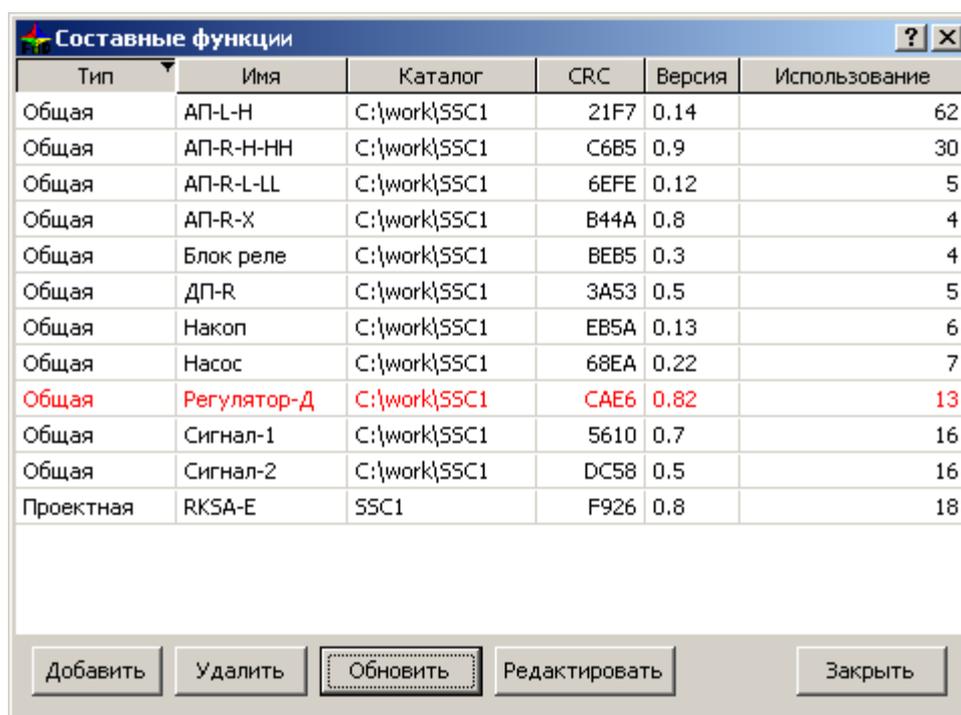
Окно «Теги» содержит следующие кнопки:

- **Добавить:** открывает окно добавления нового тега;
- **Удалить:** удаляет выделенный в списке тег;

- **Свойства:** открывает окно свойств выделенного в списке тега;
- **Заккрыть:** закрывает окно «Теги».

4.4.8. Окно «составные функции»

Чтобы иметь возможность использовать при построении диаграммы какую-либо составную функцию, её следует внести в список используемых. Для этого существует окно списка составных функций, которое открывается из пункта меню «Объекты» командой «Составные функции» (Ctrl+D). Для добавления в имеющийся список новой составной функции в нижней части окна расположена кнопка «Добавить».



Тип	Имя	Каталог	CRC	Версия	Использование
Общая	АП-L-H	C:\work\SSC1	21F7	0.14	62
Общая	АП-R-H-HH	C:\work\SSC1	C6B5	0.9	30
Общая	АП-R-L-LL	C:\work\SSC1	6EFE	0.12	5
Общая	АП-R-X	C:\work\SSC1	B44A	0.8	4
Общая	Блок реле	C:\work\SSC1	BEB5	0.3	4
Общая	ДП-R	C:\work\SSC1	3A53	0.5	5
Общая	Накоп	C:\work\SSC1	EB5A	0.13	6
Общая	Насос	C:\work\SSC1	68EA	0.22	7
Общая	Регулятор-Д	C:\work\SSC1	CAE6	0.82	13
Общая	Сигнал-1	C:\work\SSC1	5610	0.7	16
Общая	Сигнал-2	C:\work\SSC1	DC58	0.5	16
Проектная	RKSA-E	SSC1	F926	0.8	18

Рис. 4.4.8.1. Окно списка составных функций

Окно имеет 6 столбцов:

- **Тип:** отображает тип составной функции (системный, проектный или общий);
- **Имя:** отображает имя составной функции;
- **Каталог:** указывает адрес на жёстком диске компьютера, где хранится составная функция;
- **CRC:** это контрольная сумма, которая пересчитывается программой каждый раз при внесении изменений в составную функцию. Это нужно для отслеживания обновлений версии составной функции;
- **Версия:** отображает номер версии составной функции, который увеличивается после каждого редактирования;

- **Использование:** указывает, сколько раз составная функция использована в диаграмме алгоритма.

Если нужно отсортировать строки, достаточно щёлкнуть мышью заголовок столбца, и строки отсортируются по алфавиту по значениям этого столбца.

В нижней части окна расположены кнопки:

- **Добавить:** открывает окно для добавления к диаграмме составной функции. Чтобы добавить функцию к списку доступных к использованию, нужно выбрать в списке нужную составную функцию либо указать её местонахождение на компьютере:

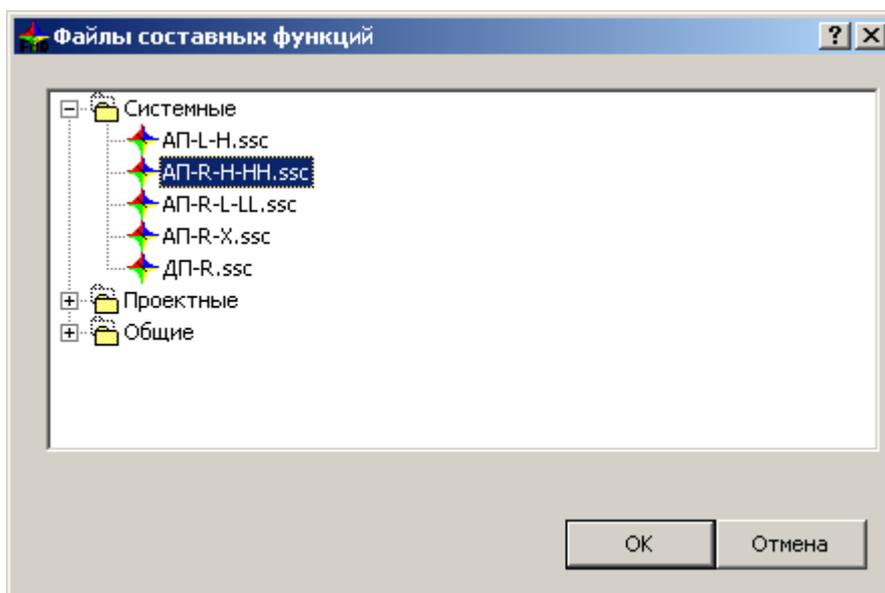


Рис. 4.4.8.3. Окно выбора файла составных функций

- **Удалить:** удаляет из диаграммы выделенную в списке составную функцию;
- **Обновить:** если в диаграмму составной функции были внесены изменения, её строка выделяется красным шрифтом, что сигнализирует о необходимости обновить список. Нажатие этой кнопки произведёт обновление списка;
- **Редактировать:** открывает окно, где можно внести некоторые изменения в свойства составной функции (тип, каталог её хранения):

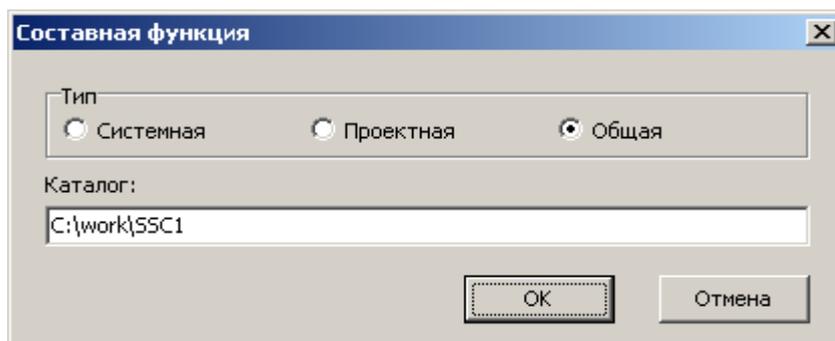


Рис. 4.4.8.4. окно «Составная функция».

- **Закреть:** закрывает окно «Составные функции».

4.5. Компиляция алгоритма

Программа-компилятор (исполняемый файл Compiler) вызывается из программы «FBD_Builder» нажатием кнопки  или клавиши F9.

Компилятор осуществляет перевод всей написанной программы в эквивалентную ей программу-результат, которую можно будет записать в контроллер и запустить в эксплуатацию. Кроме того, компилятор обнаруживает и позволяет просматривать возможные ошибки построения алгоритма.

Перед компиляцией диаграммы необходимо определить её свойства. Для правильного процесса компиляции важны два параметра: оптимизация алгоритма (перенос констант) и автоматическая адресация flash.

Если диаграмма компилируется впервые, компилятор попросит сохранить файл алгоритма и присвоить ему название. Необходимо учитывать, что имя может содержать не более 31 символа. Если файл уже компилировался прежде, и диаграмма была затем изменена, будет предложено заменить старые файлы проекта новыми:

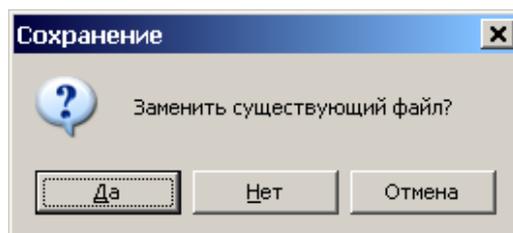


Рис. 4.5.1. Компилятор предлагает заменить старый файл проекта на изменённый

Во время компиляции между всеми функциями алгоритма распределяется очерёдность их выполнения; каждой функции присваивается порядковый номер.

Чтобы функция могла выполняться, значения на её входах должны быть определены, т.е. известны программе. Это возможно при следующих условиях:

- вход функции не связан связью ни с одной другой функцией (при этом значение может быть присвоено входу изначально, а может и отсутствовать и тогда считаться равным нулю);
- вход функции связан с выходом функции «Обратная связь»;
- вход функции связан с выходом функции, которая уже выполнена;

Функция считается выполненной, если результат арифметико-логических операций, произведенных с её входными данными, сосчитан и записан на выход функции.

Порядок выполнения функций алгоритма определяется при компиляции следующим образом:

— Среди функций алгоритма (все они до компиляции считаются невыполненными) ищется функция, значения на входах которой определены. Такой функции присваивается порядковый номер выполнения 0000, она помечается как выполненная и исключается из списка для дальнейшего поиска;

— Среди оставшихся функций снова ищется такая, значения на входах которой определены; ей присваивается порядковый номер выполнения 0001, она помечается как выполненная и тоже исключается из поиска;

— таким образом последовательно исключаются все функции алгоритма.

Если за проход компилятора по всем функциям алгоритма не удалось найти ни одной функции, которую можно пометить как выполненную, либо оказалось невозможно выполнить лишь некоторые функции, то после компиляции на экране возникает сообщение о найденных ошибках и их количестве.

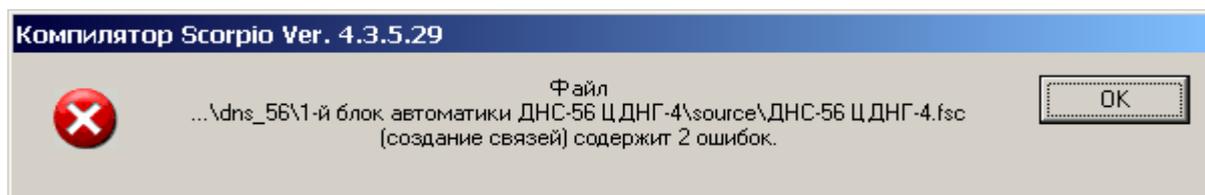


Рис. 4.5.2. Сообщение о найденных компилятором ошибках.

После нажатия кнопки «ОК» в окне программы всплывёт перечень найденных компилятором ошибок и предупреждений, которые программисту надлежит исправить. Предупреждения компилятор выдаёт в тех случаях, когда не может определить, является ли данный случай ошибкой, либо не является. Например, если галочка «Автоматически изменять адресацию flash» проставлена не была (меню «Файл» — «Свойства проекта»), а пользователь мог забыть задать уникальные адреса нескольким объектам.

Если же ошибок не найдено и файл успешно скомпилировался, будет выдано соответствующее сообщение, указывающее имя созданного проекта и его размер:

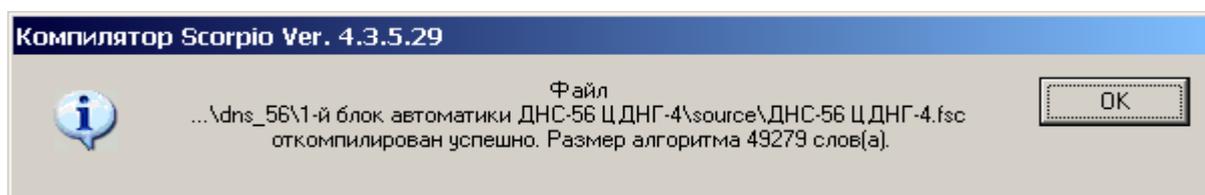


Рис. 4.5.3. Сообщение об успешной компиляции проекта.

При этом, однако, компилятор может выдать предупреждения, если они найдены. Окончательно отлаженный алгоритм не должен содержать ни ошибок, ни предупреждений.

После компиляции создаются файлы проекта:

- выполняемый файл «<Имя файла алгоритма>.sc»;
- файл ошибок и предупреждений «<Имя файла алгоритма>.err».

Если компиляция прошла без ошибок, то файл ошибок создастся, но будет пустым.

Полный список сообщений, предупреждений и ошибок см. в [разделе 3 Приложения](#).

4.5.1. Ошибки и предупреждения

Окно «Сообщения» (F8) предназначено для отображения списка текущих ошибок и предупреждений:

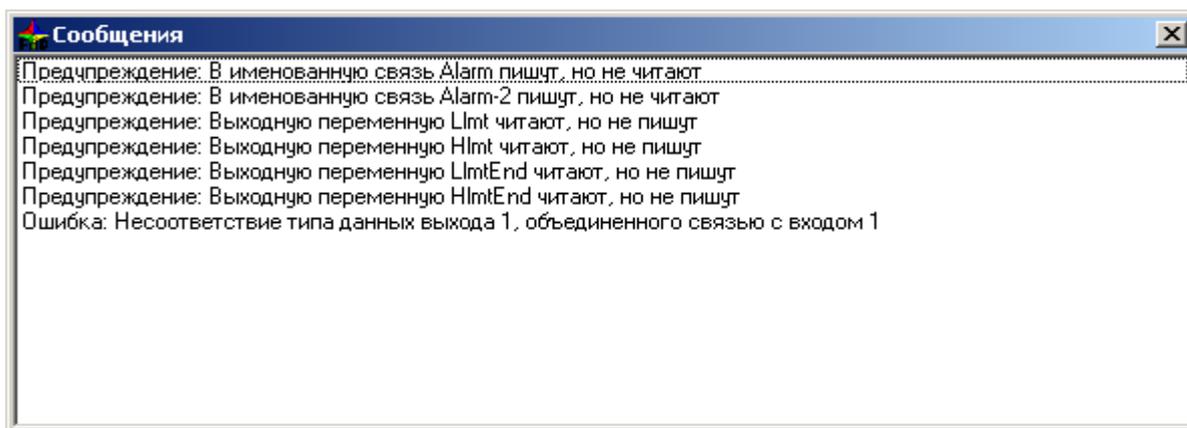


Рис. 4.5.1.1. Список ошибок и предупреждений.

Это окно также используется для их поиска на диаграмме алгоритма. Из списка с помощью мыши выбирается ошибка, после чего изображение на экране перемещается на тот участок диаграммы, где была допущена ошибка. Масштаб изображения увеличивается, а ошибочная конструкция на диаграмме начинает мигать красно-синим цветом. Исправленные ошибки удаляются из списка после перекомпиляции проекта.

4.6. Справочная система

Справочная система среды разработки алгоритма в системе «Scorpio» предназначена для предоставления разработчику алгоритма оперативной справки о любой функции, используемой в алгоритме, а также полного списка поддерживаемых системой «Scorpio» функций с подробным описанием их назначения и принципа действия.

Справка вызывается нажатием кнопки  или клавиши F1, при этом в справке отображается информация о том объекте, который в данный момент выделен на поле диаграммы. Если выделено несколько объектов, в т.ч. как блоков, так и прочих элементов, то предпочтение отдаётся блоку, а среди блоков — самому верхнему и левому. Если не выделен ни один объект, откроется общая справка по программе графического построителя системы «Scorpio».

4.6.1. Справка по программе

Справка по графическому построителю системы «Scorpio» отражает общую информацию о языке FBD и программе «FBD_Builder».

Справка по функциям: используется, если необходимо получить оперативную справку о функции (блоке), которая уже нарисована в поле алгоритма. Для этого после выбора интересующего блока необходимо нажать клавишу F1, после чего появится окно оперативной справки о данной функции.

Быстрая справка (пояснения): при включении опции «Показывать пояснения» (Ctrl+F1) возможен просмотр краткой информации об объектах без вызова каких-либо окон — в виде всплывающих подсказок.

Дополнительные возможности программы, предназначенные для облегчения и упрощения разработки алгоритма.

Настройка опций программы: некоторые возможности программы пользователь может настраивать так, как ему удобно. Это можно сделать, выбрав в меню «Вид» пункт «Настройки» и открыв окно «Опции»:

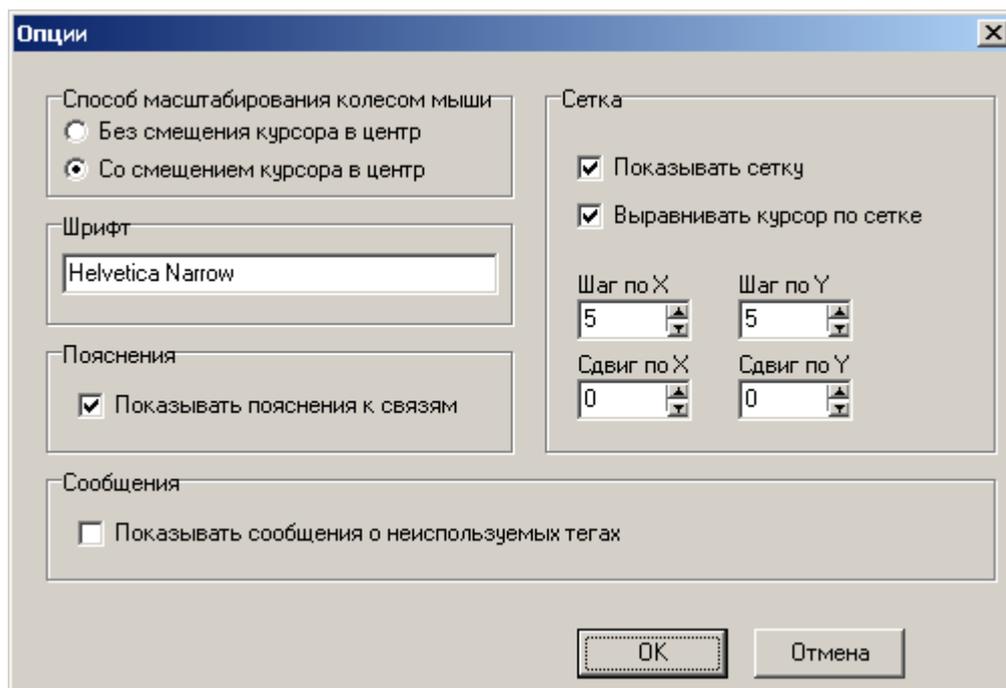


Рис. 4.6.1.1. Окно «Опции».

Здесь можно произвести следующие настройки программы «FBD_Builder»:

- **Способ масштабирования колесом мыши.** При вращении колеса мыши вперёд или назад масштаб изображения соответственно увеличивается или уменьшается. Если выбрать вариант «Без смещения курсора в центр», то при изменении масштаба изображения в центре экрана останется та же точка изображения, что и до масштабирования. Если же выбрать вариант «Со смещением курсора в центр», то при масштабировании в центр экрана будет перемещаться та точка изображения, на которой установлен курсор.
- **Шрифт.** По умолчанию в программе установлен шрифт «Helvetica Narrow», обеспечивающий оптимальное отображение на экране всех текстовых элементов диаграммы; при желании его можно заменить другим, если в поле «Шрифт» вручную вписать наименование желаемого шрифта.
- **Пояснения.** Если установить флажок «Показывать пояснения к связям», то при включенном режиме отображения пояснений (кнопка ) всплывающие подсказки будут появляться при наведении курсора не только на объекты, но и на связи.
- **Сообщения.** Если флажок «Показывать сообщения о неиспользуемых тегах» установлен, то в появляющемся после компиляции окне ошибок будут отображаться предупреждения о тегах, которые есть в списке, но не используются в алгоритме (если

таковые имеются). Если флажок снят, то сообщения о неиспользуемых тегах в окне ошибок отображаться не будут.

- **Сетка.** Отображается при установленном флажке «Показывать сетку». Если установлен флажок «Выравнивать курсор по сетке», то курсор при перемещении будет «прилипать» к узлам сетки. Поля «Шаг» и «Сдвиг» по осям X и Y позволяют настраивать плотность сетки и её положение на экране относительно диаграммы.

4.7. Способы изменения масштаба и местоположения изображения

Для изменения масштаба изображения в программе есть ряд способов:

- Команды панели инструментов «Увеличить» () и «Уменьшить» ();
- Вращение колёсика мыши: при вращении вперёд масштаб увеличивается, при вращении назад — уменьшается. Если установить флажок «Со смещением курсора в центр» (меню «Вид» пункт «Настройки»), то при масштабировании в центр экрана будет перемещаться та точка изображения, на которой находится курсор мыши.
- Клавиатурный способ: клавиша «+» увеличивает масштаб, а клавиша «-» — уменьшает. Если при этом удерживать нажатой клавишу «Ctrl», то при наличии на диаграмме выделенных объектов они будут при масштабировании помещаться в центр экрана.

Чтобы переместить диаграмму выше, ниже, влево или вправо, также существует ряд способов.

- Команда панели инструментов «Сдвиг», которая включается нажатием кнопки  или клавишей «Н»;
- Использование полос прокрутки на экране;
- Нажатие колеса мыши и удерживание его при перемещении мыши;
- Использование клавиш «←», «↑», «→», «↓» в сочетании с нажатием и удерживанием клавиши «Ctrl».

4.8. Способы копирования объектов диаграммы

Чтобы скопировать один или несколько выделенных объектов диаграммы, следует воспользоваться одним из предложенных ниже способов:

- Использование команд панели инструментов или контекстного меню «Копировать» () и «Вставить» ();
- Использование функциональной клавиши F5 (при этом копируемое изображение сразу дублируется на экране, минуя буфер обмена).

При вставке копируемое изображение отображается на поле программы, следуя за курсором до момента фиксации. Если копируется сразу несколько объектов, курсор помещается в центре между ними.

4.9. Переименование тегов и изменение текста комментариев при копировании логически целостных частей алгоритма

При копировании и вставке части диаграммы дубликат будет полностью идентичен скопированной части, поэтому имена тегов и текст комментариев будут повторяться. Их можно исправить вручную, но в программе существует возможность автоматизировать этот процесс, что особенно удобно при использовании большого числа тегов. К тому же это снижает возможность допущения ошибки при переименовании тегов.

Соответствующая команда называется «Переподключить теги» и вызывается либо нажатием клавиши F7, либо из меню программы, пункт «Правка».

Открывается следующее окно:

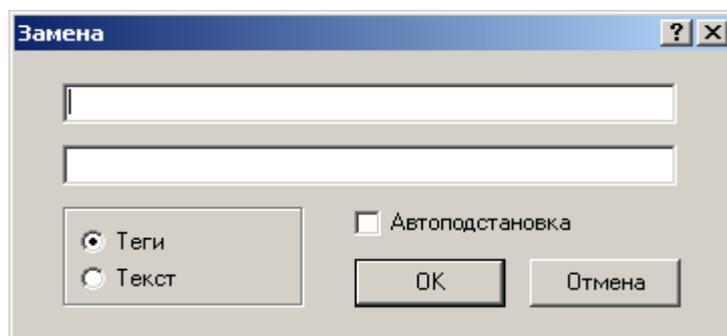


Рис. 4.9.1. Окно «Замена».

Сначала следует отметить флажком, в какие элементы предполагается внести изменения: в имена тегов или в текст комментариев. Если установить флажок «Автоподстановка», то программа сама проанализирует выделенный фрагмент диаграммы и найдёт текстовую часть, общую для всех содержащихся во фрагменте комментариев или имён тегов (по выбору пользователя), и предложит её в поле «Найти».

Если это не требуется, то следует вручную ввести в поле «Найти» подлежащий замене текст, а в поле «Заменить на» — текст для замены. При этом можно использовать подстановочные символы:

* — любое количество любых символов;

? — один символ (знак, цифра или буква).

После нажатия кнопки «ОК» программа сообщит о количестве сделанных замен:

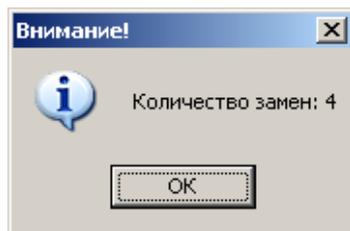


Рис. 4.9.2. Окно сообщения о количестве выполненных замен.

4.10. Пример разработки алгоритма рабочей программы для контроллера КСА-02

Пусть требованиями проекта определено следующее:

1. 4-й, 5-й и 6-й каналы модуля СТ1DAI16, находящегося в контроллере КСА-02 в слоте 3, содержат следующую информацию о состоянии устройства ГГК (Главный Газовый Клапан):

Значение 1 в канале 4 означает, что ГГК открыт;

Значение 1 в канале 5 означает, что ГГК закрыт;

Значение 1 в канале 6 означает, что ГГК работает в автоматическом режиме.

2. По 6-му и 7-му каналам модуля СТ1DDO16, находящегося в контроллере КСА-02 в слоте 11, на устройство ГГК передаются следующие управляющие сигналы с удерживанием этого управляющего воздействия в течение некоторого времени:

- Значение 1 в канале 6 означает открыть ГГК;
- Значение 1 в канале 7 означает закрыть ГГК.

3. Устройство ГГК может находиться в состояниях «открыт», «закрыт», «неисправность» и работать в режимах «кнопочный», «не автоматический». Эту информацию необходимо отображать с помощью SCADA-системы.

4. Контроллер КСА-02 обменивается информацией со SCADA-системой — по порту COM2. Протокол обмена — Modbus.

После анализа исходных данных проекта и разработки алгоритмической логики работы устройства ГГК необходимо провести следующую работу:

- для разработки алгоритма с учетом логики работы устройства ГГК из библиотеки функций системы «Scorpio» выбрать функции:

F0000 Повторитель (LOGIC);

F0012 Передатчик (LOGIC);

F0022 Запись в модуль СТ1DDO16;

F0026-2 И (LOGIC);

F0058 Задержка на выключение (импульс);

F0094 Приёмник со сбросом (LOGIC);

F0105 Чтение модуля СТ1DAI16;

- в среде разработки алгоритма с помощью редактора параметров контроллера КСА-02 выбрать протоколы обмена данными по портам:

COM1 — Modbus-Master;

COM2 — Modbus-Slave;

- выбрать необходимый базовый адрес контроллера;
- в среде разработки алгоритма с помощью редактора параметров контроллера КСА-02 сконфигурировать буферы для передачи данных:
- входной логический с размером 160 (**примечание:** размер буфера определяется в соответствии с объёмом передаваемых данных, в примере он указан произвольно — без учета объёма) для передачи на устройство ГГК управляющего воздействия «открыть» через 77-ю и «закрыть» через 78-ю ячейки этого буфера;
- выходной логический с размером 256 (**примечание:** размер буфера определяется в соответствии с объёмом передаваемых данных, в примере он указан произвольно — без учета объёма) для передачи в SCADA-систему следующей информации об устройстве ГГК:

через ячейку 192 — режим работы ГГК не автоматический;

через ячейку 193 — режим работы ГГК кнопочный;

через ячейку 190 — ГГК открыт;

через ячейку 191 — ГГК закрыт;

через ячейку 189 — ГГК неисправен.

- в среде разработки алгоритма с помощью команд редактора языка FBD нарисовать алгоритм рабочей программы для контроллера КСА-02.

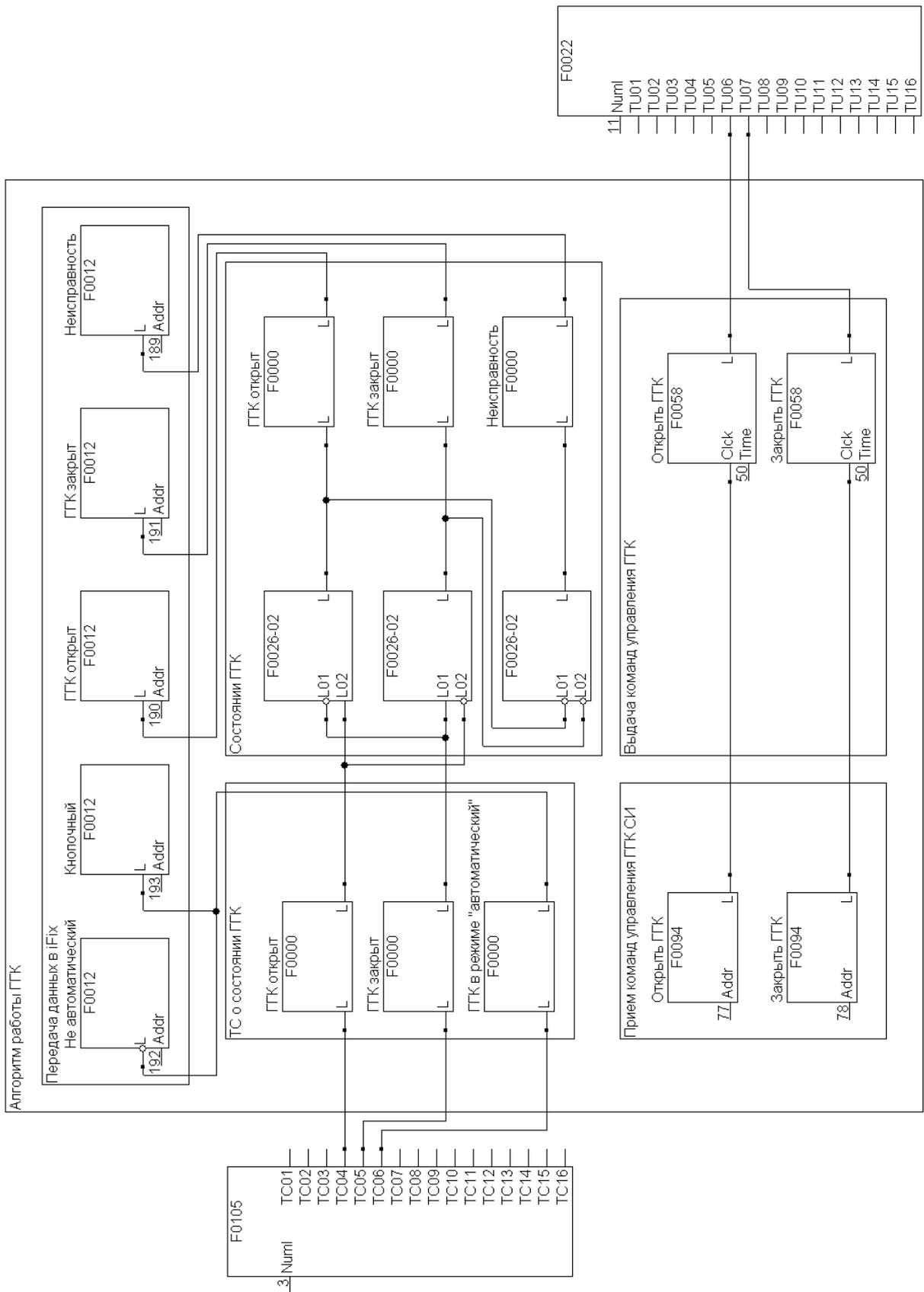


Рис. 4.10.1. Пример части алгоритма для рабочей программы контроллера КСА-02 (повернуто на 90°).

5. Сервисная программа контроллера КСА-02

Если создавать и редактировать алгоритм можно средствами программы «FBD_Builder», то запускать его выполнение, отслеживать возможные ошибки и управлять работой алгоритма при эксплуатации контроллера позволяет сервисная программа контроллера КСА-02.

Сервисная программа контроллера КСА-02 является составной частью системы «Scorpio» и предназначена для управления контроллером КСА-02.

Сервисная программа контроллера КСА-02 обеспечивает выполнение следующих функций:

- Мониторинг модулей ввода/вывода контроллера КСА-02;
- Мониторинг памяти основного процессора модуля СТ2CPU33 – Data Memory (DM), Program Memory (PM);
- Управление работой процессорного модуля;
- Загрузка алгоритма, созданного и откомпилированного в графическом редакторе системы «Scorpio»;
- Конфигурирование контроллера;
- Мониторинг состояния контроллера;
- Мониторинг состояния обмена главных процессоров в режиме горячего резервирования (при наличии);
- Мониторинг состояния обмена главного процессора со станциями удаленного ввода/вывода (при наличии);
- Управление блоком постоянных данных алгоритма, отладка алгоритма, загруженного в контроллер;
- Проверка технических характеристик и своевременное выявление отказов контроллера КСА-02.

Сервисная программа контроллера включает в себя оболочку, исполняемые модули и конфигурационные файлы.

Оболочка предназначена для удобной навигации по доступным функциям и вызова исполняемых модулей.

Исполняемый модуль обеспечивает выполнение какой-либо конкретной функции сервисной программы (нескольких функций). Запуск исполняемых модулей возможен из

оболочки сервисной программы и автономно. При автономном запуске требуется указать ключи запуска. При запуске из оболочки нужные ключи устанавливаются оболочкой. Автономный запуск программных модулей в данном руководстве не рассматривается.

Конфигурационные файлы содержат сведения об отдельных элементах сервисной программы и могут редактироваться текстовым редактором. Описание отдельных конфигурационных файлов можно посмотреть в разделе 5.5 настоящего руководства.

5.1. Условия для корректной работы сервисной программы на ПЭВМ

Для запуска сервисной программы контроллера КСА-02 от ПЭВМ требуется наличие последовательного порта (допускается использование эмулятора последовательного порта).

Схема включения контроллера КСА-02 и используемого оборудования для проверок с помощью сервисной программы приведены в соответствующей документации на контроллер КСА-02.

5.2. Соединение ПЭВМ с контроллером КСА-02

Компьютер с установленной на нём сервисной программой соединяется с контроллером с помощью нуль-модемного кабеля. Для этого на процессорном модуле СТ2CPU33 контроллера КСА-02 имеется специальный разъём COM3, предназначенный именно для соединения с ПЭВМ. Один разъём кабеля подключается к порту COM3 процессорного модуля, а другой разъём — к порту COM1 или COM2 компьютера.

При запуске сервисной программы на экране появляется окно:

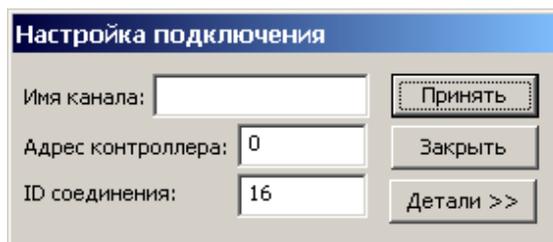


Рис. 5.2.1. Окно «Настройка подключения».

В этом окне следует указать имя канала, адрес контроллера и ID соединения. Как правило, необходимо указать общий сетевой адрес, равный «0» (по умолчанию). Также можно ввести базовый адрес контроллера, однако в этом случае следует знать базовый адрес контроллера заранее. Для отображения окна настроек подключения в расширенном виде надо нажать на кнопку «Детали >>».

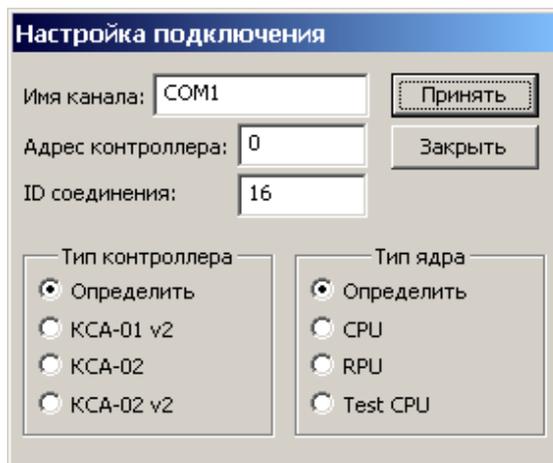


Рис. 5.2.2. Окно «Настройка подключения. Расширенный вид».

В расширенном виде окна можно указать тип ядра и тип контроллера. Если переключатель находится в положении «Определить», данный параметр будет определен сервисной программой автоматически. Явное указание типа контроллера и типа ядра целесообразно, если не удастся автоматическое определение.

5.3. Оболочка сервисной программы

Оболочка сервисной программы контроллера КСА-02 имеет заголовок, главное меню и информационную строку.

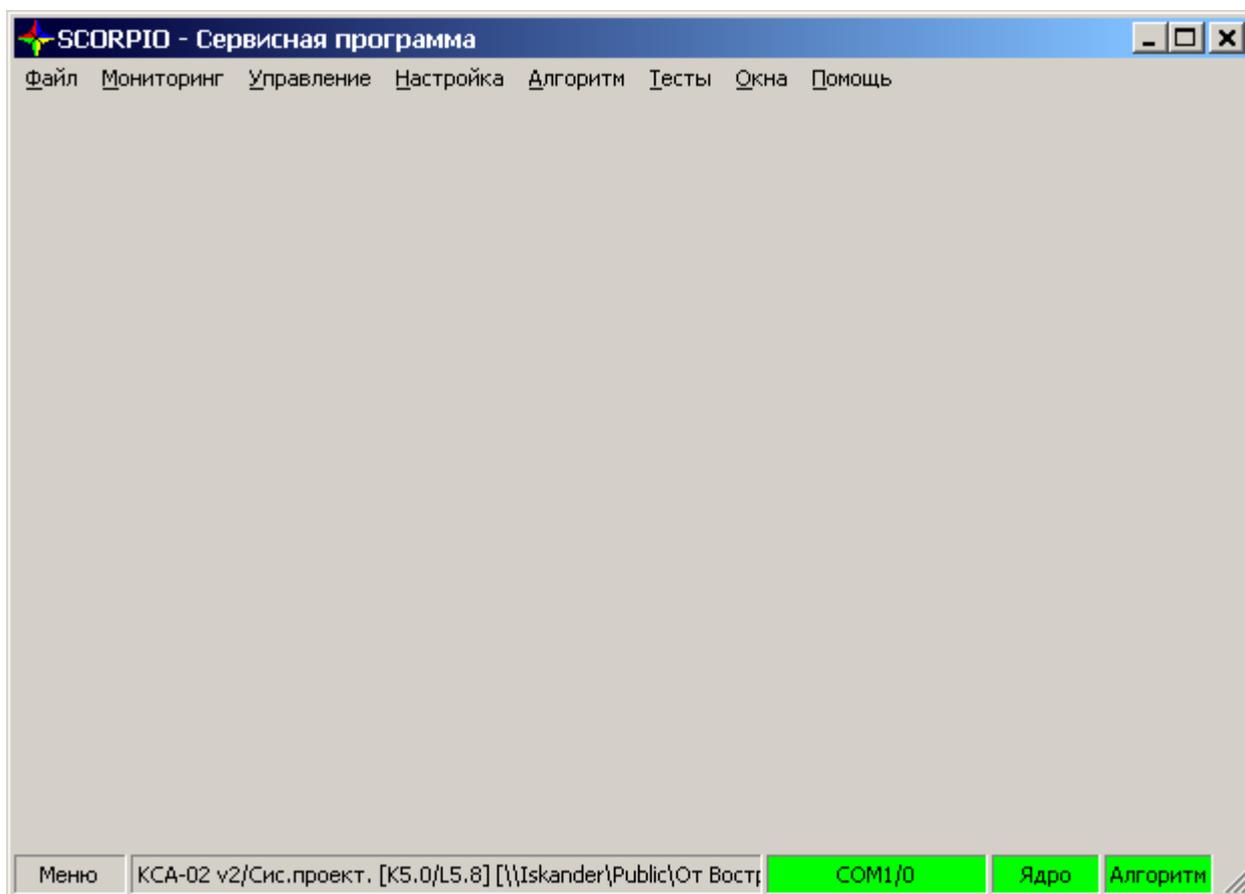


Рис. 5.3.1. Оболочка сервисной программы.

Заголовок предназначен для отображения названия программы: «Scorpio – Сервисная программа».

Главное меню организовано по принципу ниспадающего меню и служит для выбора различных инструментов для работы с контроллером и выдачи управляющих команд в контроллер. Пункты главного меню выбираются с помощью мыши.

Так же пункты главного меню отображаются при нажатии правой клавишей мыши на поле «Меню» информационной строки (рис. 5.3.2).

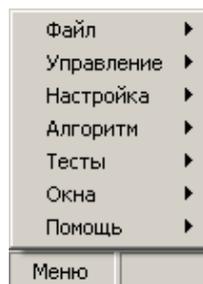


Рис. 5.3.2 Пункты меню информационной строки.

Информационная строка расположена в нижней части основного окна и предназначена для отображения текстовых сообщений сервисной программы контроллера КСА-02 и индикации **выполнения команд** главного меню и **состояния системы**.

Для индикации **выполнения команд** главного меню левая часть информационной строки содержит сообщение. Текстовое сообщение может быть **информационным**, **предупредительным** или **аварийным**. При применении команд из главного меню с целью корректировки своих дальнейших действий пользователю необходимо следить за сообщениями сервисной программы, которые сопровождают выполнение команд главного меню, отображаясь в левой части информационной строки.

Для отображения состояния контроллера достаточно подвести курсор мыши к текстовому сообщению, находящемуся в левой части информационной строки. После подведения курсора появится подсказка (рис. 5.3.3).

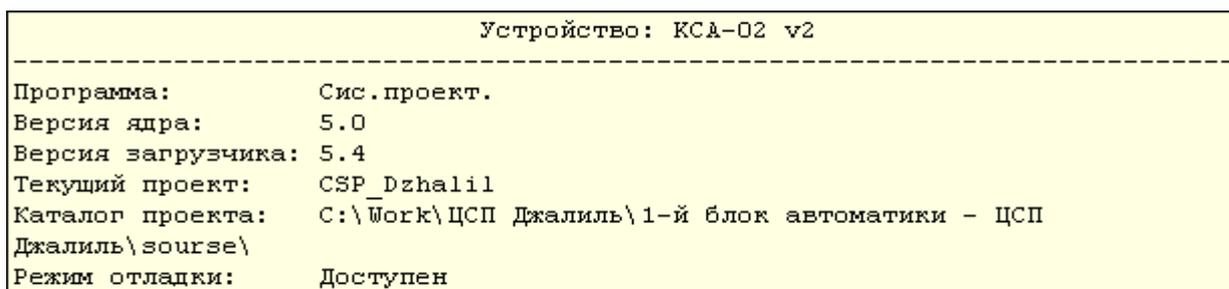


Рис. 5.3.3 Подсказка о состоянии контроллера.

Для индикации **состояния системы** предназначена правая часть информационной строки (Рис. 5.3.4). Она содержит три поля. Наименование полей, цвет и соответствующее им состояние системы определяются следующим образом:

«СОМ1» (порт ПЭВМ, к которому подключен интерфейсный жгут)

- **красный** – отсутствует связь ПЭВМ с контроллером КСА-02;
- **зеленый** – установлена связь ПЭВМ с контроллером КСА-02.

«Ядро»

- **красный** – ядро среды исполнения системы «Scorpio» во Flash памяти контроллера КСА-02 отсутствует;
- **желтый** – ядро среды исполнения присутствует во Flash памяти контроллера КСА-02, но не активно;
- **зеленый** – ядро среды исполнения присутствует во Flash памяти контроллера КСА-02 и запущено;
- **красный/зеленый** – ядро среды исполнения присутствует во Flash памяти контроллера КСА-02, но контроллер в состоянии ошибки и дальнейшее управление невозможно без перезагрузки контроллера.

«Алгоритм»

- **красный** – алгоритм в NVRAM контроллера КСА-02 отсутствует или во Flash памяти контроллера КСА-02 отсутствует ядро среды исполнения системы «Scorpio»;
- **желтый** – алгоритм загружен в NVRAM контроллера КСА-02, но не активен;
- **зеленый** – алгоритм загружен в NVRAM контроллера КСА-02 и запущен.



Рис. 5.3.4 Правая часть информационной строки, отображающая состояния системы.

Для отображения управляющих команд полей информационной строки необходимо правой клавишей мыши щёлкнуть на интересующее поле. После щелчка правой кнопки мыши появляется ниспадающее меню, пункты которого также выбираются мышью (Рис.5.3.5).

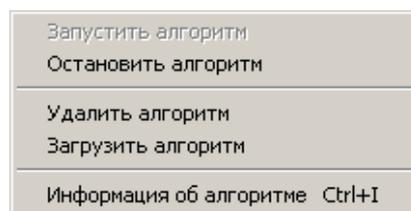


Рис.5.3.5. Управляющие команды поля «COM1» информационной строки.

Для отображения только информационной строки без основного окна необходимо дважды щёлкнуть правой клавишей мыши по информационной строке. При этом основное окно свернётся, а информационная строка переместится в верхнюю часть экрана.

5.4. Пункты главного меню. Состав и назначение.

Пункты главного меню в своем составе имеют различные инструменты для работы с контроллером и команды управления контроллером. Краткое описание состава и назначения пунктов главного меню приведено в таблице 5.4.1.

Таблица 5.4.1

Команда (клавиши вызова)	Описание команды
Файл — работа с файлами алгоритма	
Загрузить конфигурацию	Загрузка файла конфигурации *.c2
Сохранить конфигурацию	Сохранение файла конфигурации *.c2
Открыть проект	Выбор и открытие файла проекта *.sc
Открыть сессию	Выбор и открытие файла сессии *.ses
Сохранить сессию	Сохранение открытой сессии*.ses
1...4	Пути к 4 последним файлам проектов
Выход	Завершение работы сервисной программы
Мониторинг — работа с данными модулями ввода/вывода	
Обновить данные	Однократное обновление отображаемой информации открытых окон модулей ввода/вывода
Включить/Выключить анимацию	Включить/выключить обновление отображаемой информации в открытых окнах модулей ввода/вывода в реальном времени
Добавить окно DM	Открытие окна, отображающего данные Data Memory
Добавить окно PM	Открытие окна, отображающего данные Program Memory
Добавить окно модуля	Открытие окна, отображающего данные выбранного модуля
Добавить окно модуля сети СУВД	Открытие окна, отображающего данные модуля сети станций удаленного ввода-вывода
Буфер входных данных (Logic)	Открытие окна, отображающего данные входного буфера типа Logic
Буфер входных данных (Integer)	Открытие окна, отображающего данные входного буфера типа Integer
Буфер входных данных (Float)	Открытие окна, отображающего данные входного буфера типа Float

Команда (клавиши вызова)	Описание команды
Буфер выходных данных (Logic)	Открытие на экране монитора окна, отображающего данные выходного буфера типа Logic
Буфер выходных данных (Integer)	Открытие на экране монитора окна, отображающего данные выходного буфера типа Integer
Буфер выходных данных (Float)	Открытие на экране монитора окна, отображающего данные выходного буфера типа Float
Буфер внутренних данных	Открытие на экране монитора окна, отображающего данные внутреннего буфера данных
Управление — управление работой контроллера	
Запустить/Остановить ядро	Запуск/Остановка опроса модулей ввода/вывода и алгоритма контроллера. При останове работает таймер и другие прерывания, основной цикл продолжается по 50 мс, останавливается система горячего резерва, останавливается обмен с СУВД, останавливаются драйверы ввода/вывода
Запустить/Остановить алгоритм	Запуск/Остановка работы алгоритма контроллера. При останове работы алгоритма стандартный опрос модулей ввода/вывода продолжается. Работает таймер и другие прерывания, основной цикл продолжается по 50 мс. При включенной системе резервирования и режиме ведущего, даётся команда на переключение в режим ведомого. Обмен по сети резервирования продолжается
Перезапуск контроллера	Выдача контроллеру команды перезапуска, инициализация конфигурации и параметров контроллера
Загрузить программу	Загрузка ядра или тестовой программы в контроллер
Управление соединением	Содержит набор команд управления параметрами соединения
Настройка — настройка параметров контроллера	
Заводские настройки	Просмотр/ изменение заводского номера контроллера
Конфигурация контроллера	Выбор типа корзины, блока питания, настройка параметров
Статистика работы сети СУВД	Вызывает окно просмотра статистики по связи с удаленными корзинами сети СУВД
Статистика связи по интерфейсам	Просмотр статистики работы коммуникационных модулей CPN
Статистика связи по шине	Просмотр статистики обмена (количество ошибок) по шине с модулями HSB, RHA, CPM, CPE

Команда (клавиши вызова)	Описание команды
Установка часов	Просмотр/изменение показаний часов контроллера в соответствии с текущим временем ПЭВМ
Алгоритм — отладка алгоритма и управление его работой	
Удалить алгоритм	Удаление алгоритма из памяти контроллера
Загрузить алгоритм	Загрузка алгоритма в контроллер
Информация об алгоритме	Открытие на экране монитора окна с отображением информации об алгоритме, загруженном в контроллер
Отладка	Вызывает окно постановки функций на отладку и снятия функций с отладки
Убрать все функции	Закрытие на экране монитора всех окон функции открытых для отладки
Тесты — работа с тестовым обеспечением контроллера	
Тесты модулей ввода/вывода	Тестирование модулей ввода/вывода
Тест модуля СТ2CPU33	Тестирование модуля СТ2CPU33
Самодиагностика	Диагностика состояния контроллера КСА-02
Тесты общей памяти	Тестирование модулей обмена данными
Тесты Ethernet	Тестирование сети Ethernet
Тесты СОМ-портов	Тестирование СОМ-портов
Окна — изменение порядка расположения открытых окон на экране монитора	
Каскадом	Каждое следующее открытое окно перекрывает часть предыдущего открытого окна
Упорядочить	Расположение открытых окон друг за другом без перекрытия предыдущего окна
Свернуть все	Свернуть все открытые окна
Восстановить все	Восстановить на экране окна, которые были свернуты
Закрыть все	Закрыть все открытые окна
Управление	Открыть окно управления положением окон
Помощь — работа со справочной информацией	
Справка по программе (F1)	Справочная информация о работе сервисной программы контроллера КСА-02
Справка по функциям	Справочная информация о библиотеке функций

Команда (клавиши вызова)	Описание команды
Справка по контроллеру	Справочная информация о модулях ввода вывода
О программе...	Номер версии сервисной программы контроллера КСА-02

5.4.1. Меню «Файл»

Меню «Файл» содержит инструменты, позволяющие:

- Открыть проект из директории проекта, сформированный при разработке и компиляции алгоритма или модифицированный ранее средствами сервисной программы;
- Сохранить во внешнем файле информацию из контроллера, а именно: конфигурацию контроллера, хранящуюся в энергонезависимой памяти процессорного модуля; типы и версии ПО модулей, определяемых по автоконфигурации; тип и версию ПО ядра процессорного модуля; алгоритм контроллера; значения назначенных в алгоритме уставок; базы данных модулей, работающих под управлением собственного процессора;
- Загрузить из внешнего файла в контроллер следующую информацию: конфигурацию контроллера; алгоритм; значения назначенных в алгоритме уставок; необходимые для работы по назначению части баз данных модулей, работающих под управлением собственного процессора;
- Выбрать проект из списка четырёх предыдущих проектов, которые модифицировались средствами сервисной программы;
- Завершить работу сервисной программы контроллера.

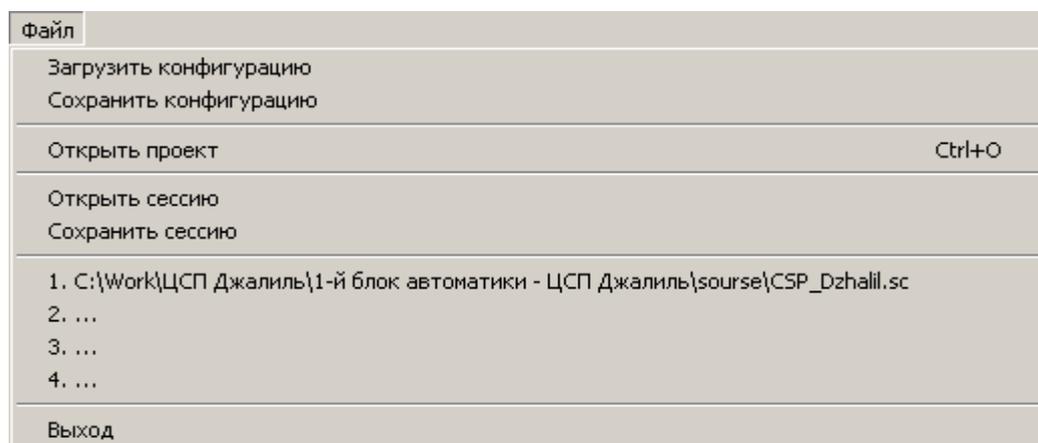


Рис. 5.4.1.1. Меню «Файл».

5.4.1.1. Пункт «Загрузить конфигурацию»

Пункт предназначен для загрузки в память контроллера файла конфигурации. Команда открывает окно, где следует указать путь к файлу конфигурации, имеющего расширение *.c2.

После открытия требуемого файла *.c2 откроется окно выбора секций файла для загрузки в контроллер:

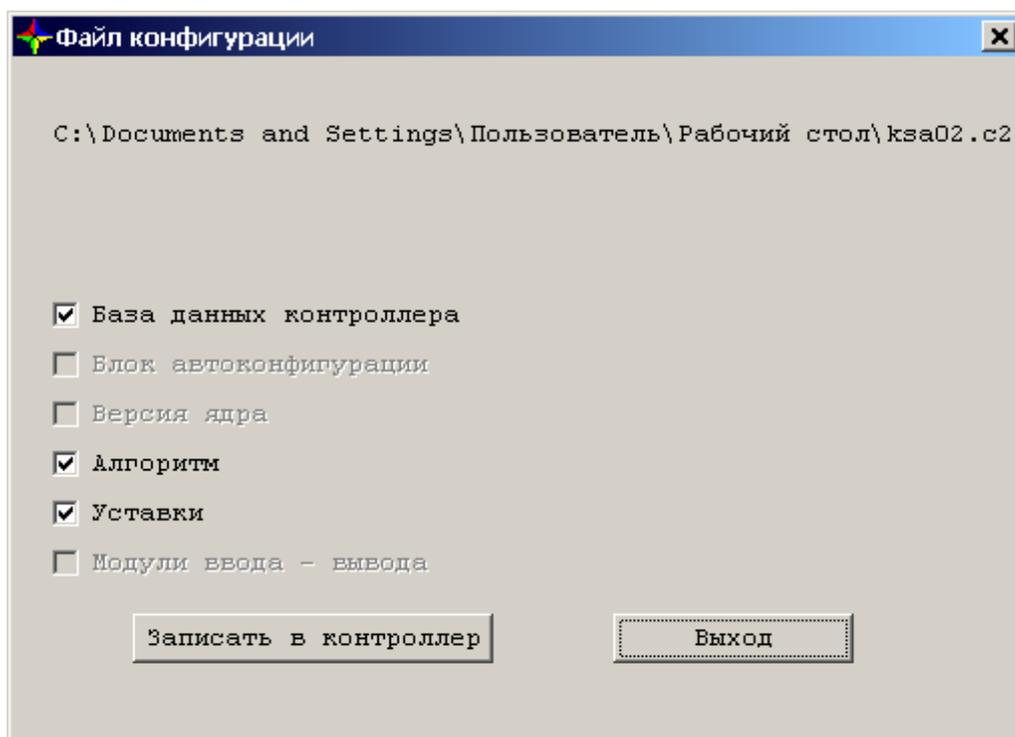


Рис. 5.4.1.1.1. Окно выбора секций для загрузки в контроллер.

Нужные секции следует отметить флажками. При этом секции «Блок автоконфигурации» и «Версия ядра» отключены: эти параметры сами автоматически определяются контроллером в процессе работы.

При загрузке сектора «Модули ввода - вывода» не загружаются первые 5 слов базы данных каждого модуля. Эти слова содержат индивидуальные характеристики модулей: тип, заводской номер, дату выпуска, версию ПО модуля.

Для процессорных модулей типа RPU, независимо от выбора пользователя, также не проводится загрузка секторов «Алгоритм» и «Уставки» (модули этого типа их не содержат).

По умолчанию к загрузке предлагаются все доступные секции.

Загрузка предлагаемых секций по отдельности рассчитана на опытных пользователей, так как надо хорошо представлять её последствия.

Загруженные база данных, алгоритм и уставки вступают в силу только после перезагрузки контроллера, которая будет предложена программой сразу после завершения операции загрузки конфигурации:

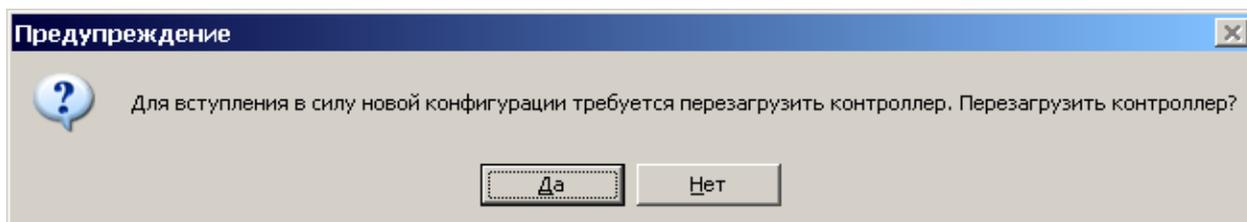


Рис. 5.4.1.1.2. Система предлагает перезагрузить контроллер для внесения изменений в память.

5.4.1.2. Пункт «Сохранить конфигурацию»

При выборе пункта «Сохранить конфигурацию» пользователю предлагается сохранить информацию из контроллера в файле *.c2. При этом работа ядра останавливается, а алгоритм приостанавливает своё выполнение.

Появляется окно, где следует указать путь к существующему файлу, если он есть. Если такого файла ещё не существует, он создаётся заново. При этом в файл записываются все секции.

Если файл с выбранным названием уже существует, предпринимается попытка его изменить; при этом предлагается выбрать секции, в которые будут внесены изменения.

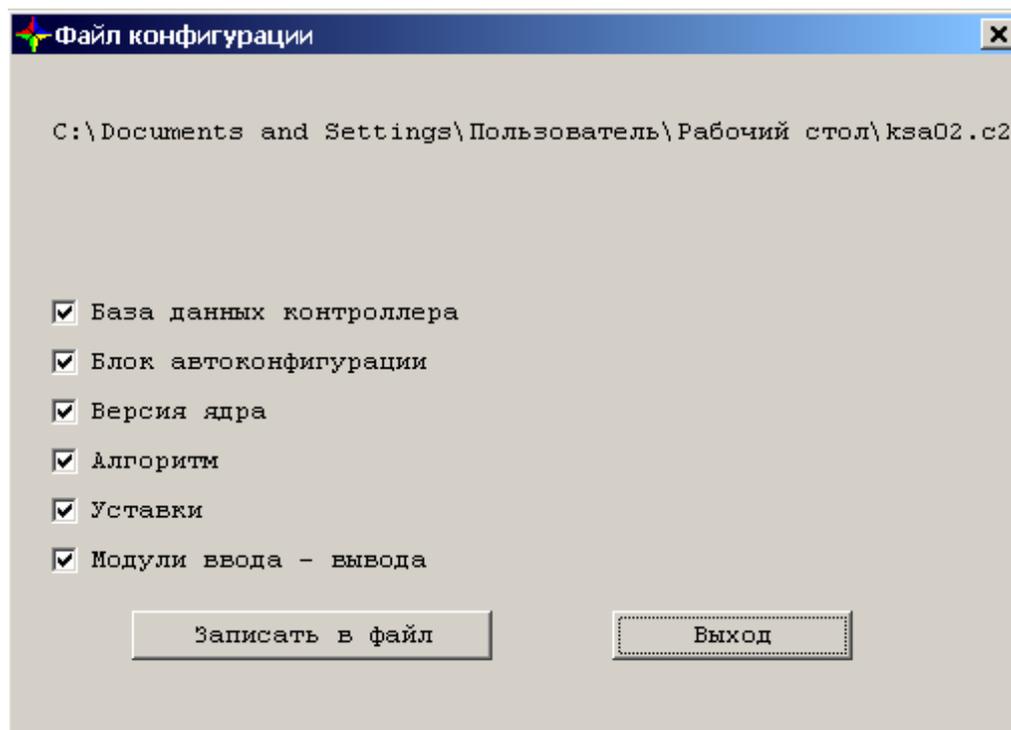


Рис. 5.4.1.2.1. Окно выбора секций для изменения файла.

Изменение существующего файла является не всегда корректной операцией и рассчитано на опытных пользователей. Поэтому желательно сначала создать на компьютере резервную копию этого файла на случай возникновения ошибки, поскольку иначе отменить сделанные изменения будет невозможно.

Возобновление работы ядра будет предложено программой сразу после завершения операции сохранения конфигурации:

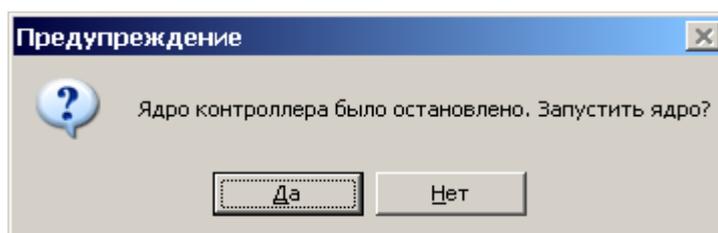


Рис. 5.4.1.2.2. Система предлагает запустить ядро.

5.4.1.3. Пункт «Открыть проект»

При выборе пункта «Открыть проект» на экране монитора открывается окно, где следует указать путь к файлу скомпилированного проекта *.sc, с которым будет работать сервисная программа.

После выбора проекта проверяется соответствие версий открываемого проекта (алгоритма, конфигурации контроллера) и проекта, уже записанного в контроллере КСА-

02 (алгоритма, конфигурации контроллера). Если эти версии совпадают, то устанавливается соединение контроллера с выбранным проектом, что отражается сообщением в информационной строке "Режим отладки доступен". Если версии не совпадают, то в информационной строке появляется сообщение "Режим отладки не доступен".

Также проверяется соответствие поколения процессорного модуля для которого откомпилирован проект и фактического, установленного в корзине контроллера.

При открытом проекте алгоритм всегда можно загрузить в память контроллера. Это позволяет, не прекращая работу сервисной программы, отредактировать алгоритм средствами подсистемы разработки алгоритма и загрузить отредактированную версию алгоритма для отладки в память контроллера (пункт «Загрузить алгоритм» в меню «Алгоритм»), не открывая повторно проект.

5.4.1.4. Пункт «Открыть сессию»

При выборе пункта «Открыть сессию» на экране монитора открывается окно, где следует указать путь к сохраненному ранее файлу сессии *.ses, с которым будет работать сервисная программа.

5.4.1.5. Пункт «Сохранить сессию»

При выборе пункта «Сохранить сессию» в файле сессии сохраняется информация обо всех открытых пользователем окнах (окнах модулей, окнах добавленных на отладку функций), — если таковые были, т.е. делается «слепок» основного окна сервисной программы на момент сохранения сессии.

5.4.1.6. Открытие проекта выбором из списка

Если необходимо открыть проект, который модифицировался средствами сервисной программы в числе четырёх последних проектов, то выбрать его можно из списка, отображаемого в меню «Файл».

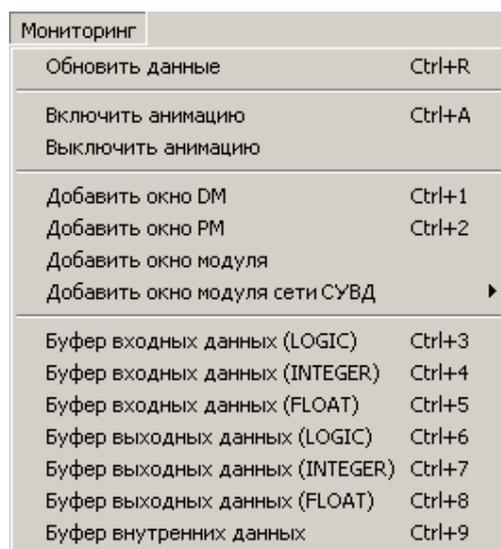
Если при этом в программе уже имеется открытый проект, то сначала автоматически происходит его закрытие.

5.4.1.7. Пункт «Выход»

Пункт «Выход» предназначен для завершения работы сервисной программы. После выдачи команды всплывает окно подтверждения выхода из сервисной программы контроллера.

5.4.2. Меню «Мониторинг»

Меню «Мониторинг» содержит инструменты и команды, позволяющие отображать на экране монитора в режиме реального времени информацию, находящуюся в контроллере КСА-02. Отображение информации осуществляется в окнах, вид которых зависит от типа отображаемой информации — ячейки памяти DM и PM, модуля ввода/вывода, или буферы данных.



Мониторинг	
Обновить данные	Ctrl+R
Включить анимацию	Ctrl+A
Выключить анимацию	
Добавить окно DM	Ctrl+1
Добавить окно PM	Ctrl+2
Добавить окно модуля	
Добавить окно модуля сети СУВД	▶
Буфер входных данных (LOGIC)	Ctrl+3
Буфер входных данных (INTEGER)	Ctrl+4
Буфер входных данных (FLOAT)	Ctrl+5
Буфер выходных данных (LOGIC)	Ctrl+6
Буфер выходных данных (INTEGER)	Ctrl+7
Буфер выходных данных (FLOAT)	Ctrl+8
Буфер внутренних данных	Ctrl+9

Рис. 5.4.2.1. Меню «Мониторинг».

5.4.2.1. Пункт «Обновить данные»

Пункт «Обновить данные» предназначен для выдачи в контроллер команды однократного обновления отображаемой информации открытых окон сети СУВД, модулей ввода/вывода, буферов данных, входов функций находящихся на отладке.

Команда «Обновить данные» доступна при наличии открытых (одного или несколько) окон — сети СУВД, модулей ввода/вывода или буферов данных или функций на отладке.

5.4.2.2. Пункт «Включить анимацию»

Пункт «Включить анимацию» предназначен для выдачи в контроллер команды циклического обновления отображаемой информации открытых в среде сервисной программы окон сети СУВД, модулей ввода/вывода, буферов данных, входов функций находящихся на отладке. Применение команды «Включить анимацию» блокирует доступ к некоторым пунктам главного меню. Исключение составляют пункты «Файл», «Мониторинг», «Окна» (если открыто более одного окна), «Помощь».

Команда «Включить анимацию» доступна при наличии открытых (одного или несколько) окон — сети СУВД, модулей ввода/вывода или буферов данных, или функций на отладке.

5.4.2.3. Пункт «Выключить анимацию»

Пункт «Выключить анимацию» предназначен для выдачи в контроллер команды отключения циклического обновления отображаемой информации открытых в среде сервисной программы окон сети СУВД, модулей ввода/вывода, буферов данных, входов функций находящихся на отладке.

5.4.2.4. Пункт «Добавить окно DM»

Пункт «Добавить окно DM» предназначен для мониторинга ячеек памяти DataMemory – памяти данных в ОЗУ центрального процессора процессорного модуля контроллера КСА-02.

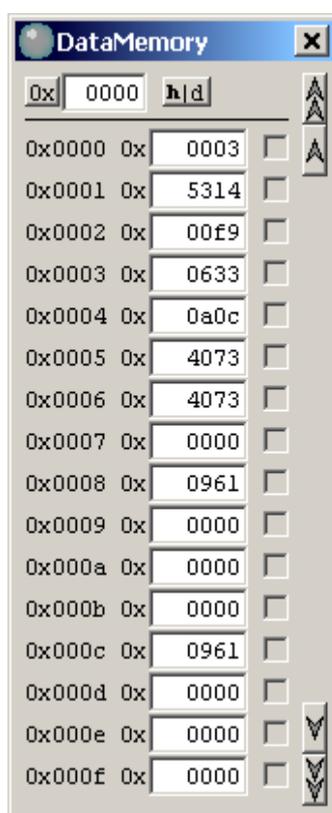


Рис. 5.4.2.4.1 Окно «DataMemory».

5.4.2.5. Пункт «Добавить окно PM»

Пункт «Добавить окно PM» предназначен для мониторинга ячеек памяти ProgramMemory - памяти программ в ОЗУ центрального процессора процессорного модуля контроллера КСА-02.

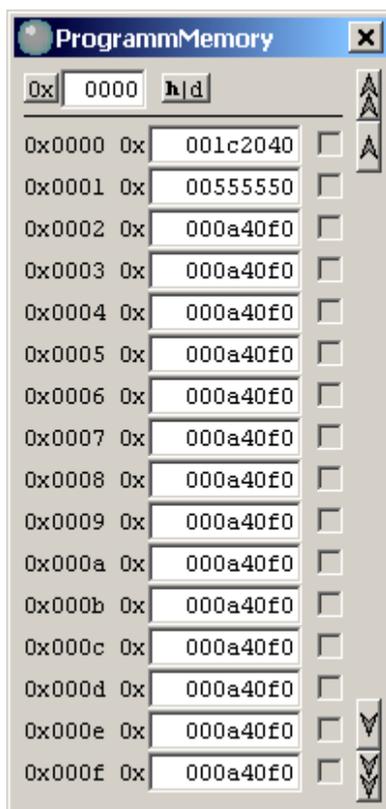


Рис. 5.4.2.5.1 Окно «ProgramMemory».

5.4.2.6. Пункт «Добавить окно модуля»

Пункт меню «Добавить окно модуля» предназначен для мониторинга состояния входов входных модулей (либо выходов выходных модулей) ввода/вывода контроллера КСА-02. Также сервисная программа КСА-02 позволяет подменять реальное значение входного значения модуля ввода/вывода на любое другое значение.

Пункт «Добавить окно модуля» доступен только в том случае, если ядро среды исполнения системы «Scorpio» присутствует во Flash памяти контроллера КСА-02 и запущено. При выборе пункта «Добавить окно модуля» на экране монитора открывается окно «Выбор модуля».

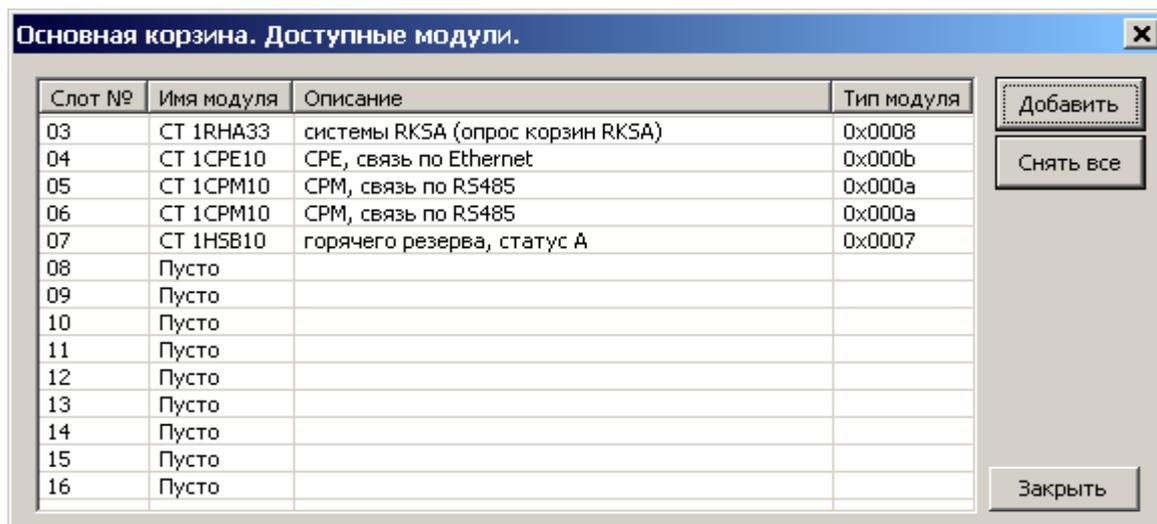


Рис. 5.4.2.6.1. Окно «Выбор модуля».

В этом окне в виде списка названий модулей ввода/вывода по слотам отображается конфигурация контроллера, записанная в его памяти. Если эта конфигурация не будет соответствовать реальной — набору модулей, физически находящихся в корзине контроллера — то операции по обращению к модулям ввода/вывода средствами сервисной программы будут заканчиваться с ошибкой. Поэтому необходимо записывать в память контроллера реально существующую конфигурацию, для чего предназначен пункт «Конфигурация контроллера» в меню «Настройка».

Информацию по модулям ввода/вывода отображают окна, вид которых индивидуален для каждого модуля и зависит от количества каналов модуля и его типа (дискретный или аналоговый).

Вид этих окон представлен ниже на примере модуля СТ1АСI08 (для аналоговых модулей) и модуля СТ1DDI30 (для дискретных):

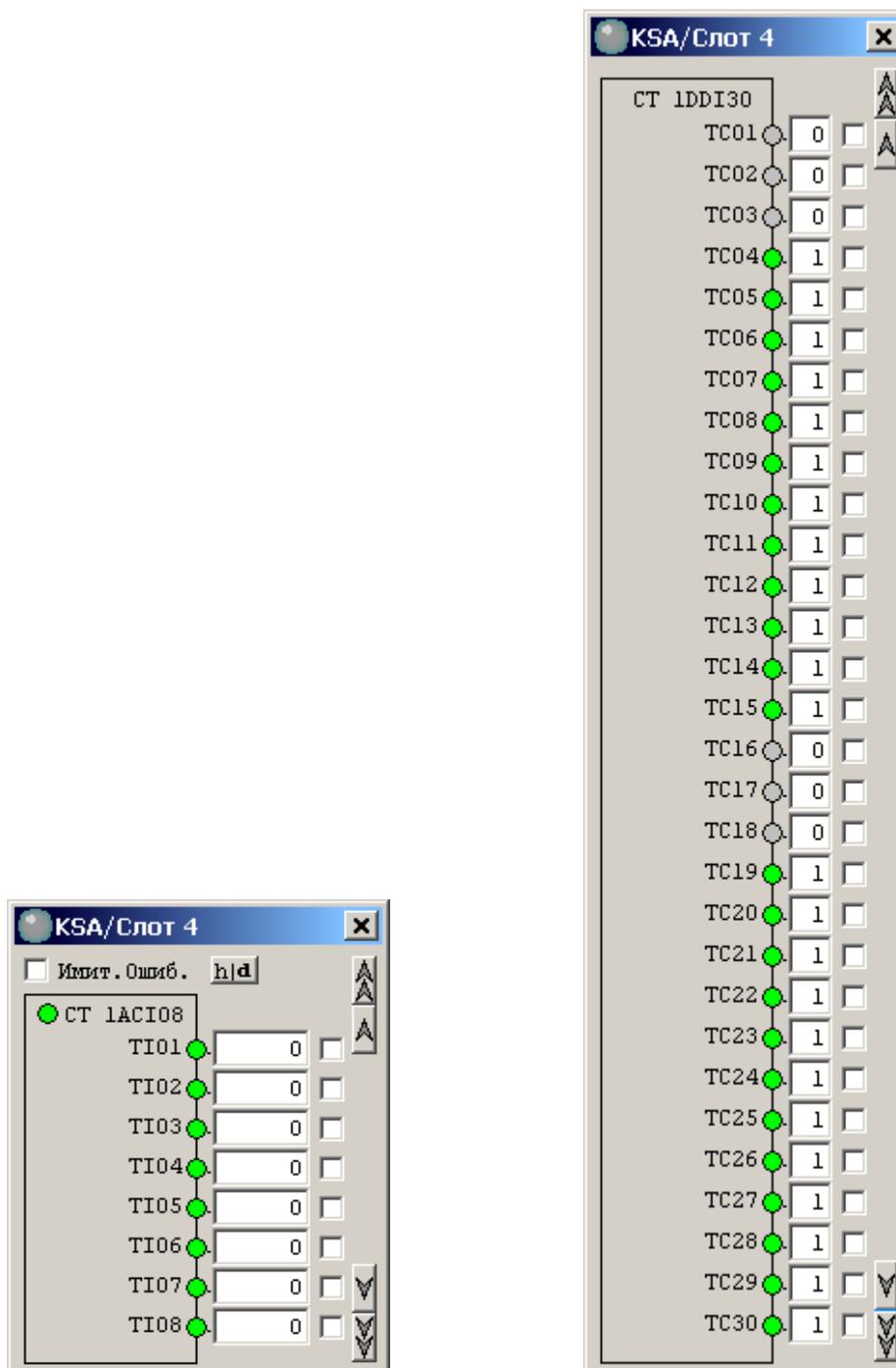


Рис.5.4.2.6.2. Окна для отображения информации по модулям ввода/вывода.

Окно отображения информации по аналоговым модулям.

Внешний вид этого окна меняется лишь в плане количества полей, обозначающих каналы модуля (их количество зависит от типа модуля — 4, 8, 16 или 30).

В верхней части окна расположены следующие элементы:

- информация о названии модуля (например, «СТ1АС108»);

- информация о порядковом расположении модуля в корзине (например, «Слот 3»);
- индикатор анимации (зелёный — происходит постоянное обновление информации по модулю, серый — не происходит постоянного обновления).

При открытии окна отображения информации по модулю ввода/вывода происходит однократное считывание состояния выбранного модуля. Для постоянного обновления необходимо включить режим анимации.

Кроме этого, окно информации по модулям ввода/вывода содержит:

- область «Каналы» — отображает информацию о состоянии и значении входов каждого из каналов модуля:
- поле, содержащее значение, измеренное на входе (число в шестнадцатиричной форме исчисления);
- флажок «Ввод»: при его установке значение, поданное на модуль, заменяется значением, введенным из сервисной программы. Используется при отладке алгоритма. При этом будет перезаписано значение, будут считаны новые значения всех каналов модуля и всех блокировок модуля, а в окне отобразится новая информация;
- индикаторы состояния каналов: зелёный цвет — канал исправен, серый — неисправен.
- флажок «Имитацию ошибок вкл.» — установка этого флажка включает режим, позволяющий вручную определять исправность состояния каналов и модуля;
- переключатель отображения значения данных **h** – шестнадцатеричный, **b** – двоичный.

Окно отображения информации по дискретным модулям.

Внешний вид этого окна меняется лишь в плане количества полей, обозначающих каналы модуля (их количество зависит от типа модуля — 4, 8, 16 или 30).

В верхней части окна расположены следующие элементы:

- информация о названии модуля (например, «СТ1DDI30»);
- информация о порядковом расположении модуля в корзине (например, «Слот 4»);
- флажок «Имитацию ошибок вкл.» — установка этого флажка позволяет вручную определять исправность состояния каналов и модуля;
- область для отображения информации о неисправности модуля: если модуль неисправен, то в верхнем правом углу окна отображается надпись: «Неисправность модуля»;
- индикатор анимации (зелёный — происходит постоянное обновление информации по модулю, серый — не происходит постоянного обновления).

При открытии окна отображения информации по модулю ввода/вывода происходит однократное считывание состояния выбранного модуля. Для постоянного обновления необходимо включить режим анимации.

Кроме этого, окно информации по модулям ввода/вывода содержит:

- область «Каналы» — отображает информацию о состоянии и значении входов каждого из каналов модуля;
- индикаторы отображения значений каналов: зелёный цвет — логическая 1, серый — логическая 0;
- поле, содержащее значение, измеренное на входе (число в двоичной форме исчисления);
- флажок «Ввод»: при его установке значение, поданное на модуль, заменяется значением, введенным из сервисной программы. Используется при отладке алгоритма. При этом будет перезаписано значение, будут считаны новые значения всех каналов модуля и всех блокировок модуля, а в окне отобразится новая информация;

5.4.2.7. Пункт «Добавить окно модуля сети СУВД»

Пункт «Добавить окно модуля сети СУВД» предназначен для мониторинга состояний входов модулей ввода/вывода корзины станции удаленного ввода-вывода ([определение](#)). Также сервисная программа КСА-02 позволяет заменять реальное значение входного значения модуля ввода/вывода на любое другое значение.

Пункт «Добавить окно модуля сети СУВД» доступен только в том случае, если ядро среды исполнения системы "Scorpio" присутствует во Flash памяти контроллера КСА-02 и запущено. При выборе пункта «Добавить окно модуля сети СУВД» открывается подменю выбора корзины СУВД.

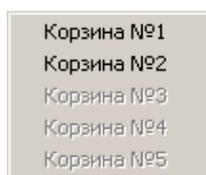


Рис. 5.4.2.7.1. Подменю для выбора удаленной корзины СУВД.

В списке отображаются корзины СУВД. Корзины, присутствующие в системе отображаются черным цветом, отсутствующие – серым, неактивны. Информация о наличии корзин приводится согласно конфигурации системы ввода-вывода. Для выбора

корзины необходимо выбрать соответствующий пункт подменю. После данной операции откроется окно выбора модуля.

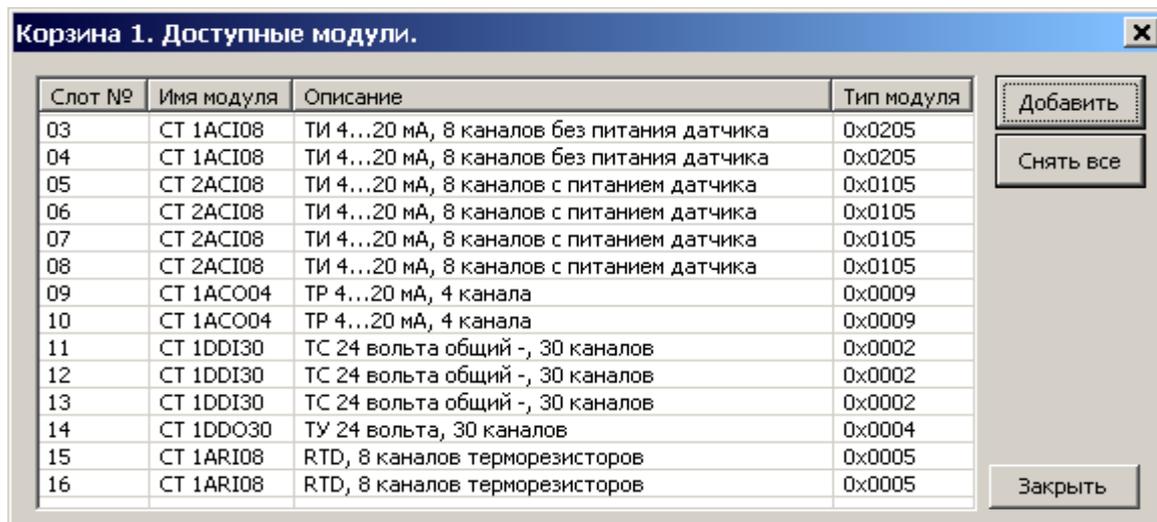


Рис. 5.4.2.7.2. Окно для выбора модуля удаленной корзины СУВД.

Дальнейшие операции аналогичны [5.4.2.6. Пункт «Добавить окно модуля»](#)

В окне «Выбор модуля» в виде списка названия модулей ввода/вывода по слотам отображается конфигурация удаленной корзины СУВД, записанная в его памяти. Если эта конфигурация не будет соответствовать реальной — набору модулей ввода/вывода, физически находящихся в корзине, то операции по обращению к модулям ввода/вывода средствами сервисной программы будут заканчиваться с ошибкой или отображаемая информация не будет соответствовать реальной. Поэтому необходимо записывать в память контроллера реально существующую конфигурацию.

5.4.2.8. Пункт «Буфер входных данных (Logic)»

С помощью пункта «Буфер входных данных (Logic)» открывается окно, отображающее значения, хранящиеся во входном буфере типа Logic.

Окно «Буфер входных данных (Logic)» используется при отладке алгоритма и позволяет просматривать и изменять содержимое буфера логических входных данных.

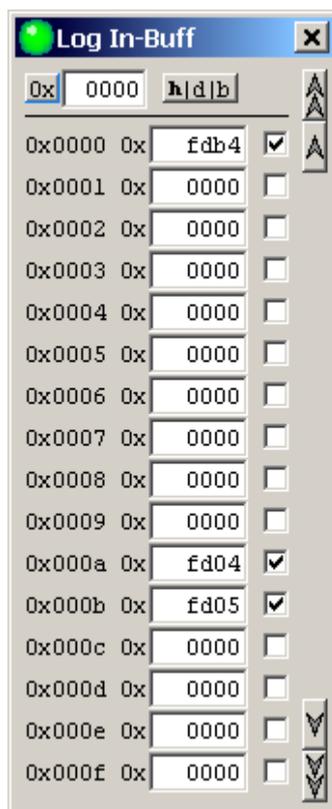


Рис. 5.4.2.8.1. Окно «Буфер входных данных».

Формат отображения ячейки буфера в окне:

{адрес} {данные} {признак записи в ячейку}

Адрес ячеек в окне отображения начинается с нулевого значения (действительный адрес не отображается).

Отображение значения данных возможно в трех видах — **h** (шестнадцатеричный), **d** (десятичный) и **b** (двоичный), для чего предназначена соответствующая кнопка, расположенная в области окна.

h — отображается целыми 16-битными словами данных логического буфера с соответствующей адресацией (0x0000, 0x0010, 0x0020 и т.д. до размера буфера, определенного при создании алгоритма). При изменении (перезаписи) значения любого бита 16-битного слова средствами сервисной программы перезаписываются все 16 бит, поэтому при установке значения в поле ввода следует указывать шестнадцатеричное значение, оставляющее неизменными остальные биты 16-битного слова.

b — отображается один бит данных логического буфера с соответствующей адресацией (0x0000, 0x0001, 0x0002 и т.д. до размера буфера, определенного при создании алгоритма).

Чтобы изменить (перезаписать) значение в ячейке буфера средствами сервисной программы, нужно ввести нужное значение в поле «Данные» в соответствии с требуемым форматом (h или b) и ввести его, нажав клавишу «Enter».

Если перезапись значения осуществляется в режиме циклической перезаписи (в «Режиме анимации»), то при вводе нового значения напротив соответствующего поля установится признак циклической записи в ячейку буфера .

Если перезапись значения осуществляется в режиме однократной перезаписи (команда меню «Обновить данные»), то при вводе нового значения происходит цикл однократной записи/чтения и после получения ответа значение в данном поле ввода измениться на значение, прочитанное из контроллера.

5.4.2.9. Пункт «Буфер входных данных (Integer)»

С помощью пункта «Буфер входных данных (Integer)» открывается окно, отображающее значения, хранящиеся во входном буфере типа Integer.

Окно «Буфер входных данных (Integer)» используется при отладке алгоритма и позволяет просматривать и изменять содержимое соответствующего буфера входных данных.

Формат отображения ячейки буфера в окне:

{адрес} {данные} {признак записи в ячейку}

Адрес ячеек в окне отображения начинается с нулевого значения (действительный адрес не отображается).

Отображение значения данных возможно в трех видах — **h** (шестнадцатеричный) и **d** (десятичный), для чего предназначена соответствующая кнопка, расположенная в области окна.

Чтобы изменить (перезаписать) значение в ячейке буфера средствами сервисной программы, нужно ввести нужное значение в поле «Данные» в выбранном формате (или d) и зафиксировать изменения, нажав на клавиатуре клавишу «Enter».

Если перезапись значения осуществляется в режиме циклической перезаписи (см. «Режим анимации»), то при вводе нового значения напротив соответствующего поля установится признак циклической записи в ячейку буфера .

Если перезапись значения осуществляется в режиме однократной перезаписи (см. «Обновить данные»), то при вводе нового значения происходит цикл однократной

записи/чтения и после получения ответа значение в данном поле ввода измениться на значение, прочитанное из контроллера.

5.4.2.10. Пункт «Буфер входных данных (Float)»

С помощью пункта «Буфер входных данных (Float)» открывается окно, отображающее значения, хранящиеся во входном буфере типа Float.

Окно «Буфер входных данных (Float)» используется при отладке алгоритма и позволяет просматривать содержимое соответствующего буфера входных данных.

Пункт предназначен для открытия на экране монитора окна, отображающего значения, хранящиеся во входном буфере типа Float.

Формат отображения ячейки буфера в окне:

{адрес} {данные}

Адрес ячеек в окне отображения начинается с нулевого значения (действительный адрес не отображается).

Отображение значения данных возможно в трех видах — **df** (десятичный с точкой), **de** (десятичный [экспоненциальный](#)), **h** (шестнадцатиричный).

5.4.2.11. Пункт «Буфер внутренних данных»

Пункт «Буфер внутренних данных» предназначен для открытия на экране монитора окна, отображающего значения, хранящиеся в буфере внутренних данных.

Отображение окна «Буфер внутренних данных» в основном используется при отладке алгоритма и позволяет как просматривать, так и изменять содержимое буфера внутренних данных.

Формат отображения ячейки буфера в окне:

{адрес} {данные} {признак записи в ячейку}

Адрес ячеек в окне отображения начинается с нулевого значения (действительный адрес не отображается).

Отображение осуществляется 16-битными словами с соответствующей адресацией. Отображение значения данных возможно в двух видах — **h** (шестнадцатиричный) и **d** (десятичный), для чего в окне имеется соответствующая кнопка.

Чтобы изменить (перезаписать) значение в ячейке буфера средствами сервисной программы, нужно ввести нужное значение в поле «Данные» в выбранном формате (**h** или **d**) и зафиксировать изменения, нажав на клавиатуре клавишу «Enter».

Если перезапись значения осуществляется в режиме циклической перезаписи (в «Режиме анимации»), то при вводе нового значения напротив соответствующего поля установится признак циклической записи в ячейку буфера .

Если перезапись значения осуществляется в режиме однократной перезаписи (см. «Обновить данные»), то при вводе нового значения происходит цикл однократной записи/чтения, и после получения ответа значение в данном поле ввода меняется на значение, прочитанное из контроллера.

5.4.3. Меню «Управление»

Меню «Управление» содержит инструменты и команды, предназначенные для управления работой контроллера.

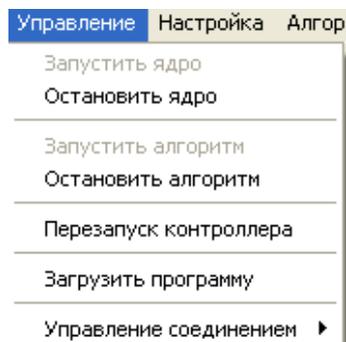


Рис. 5.4.3.1. Меню «Управление».

Команды управления работой контроллера применяются при отладке алгоритма или при сбоях в его работе.

5.4.3.1. Пункт «Запустить/Остановить ядро»

Запуск/остановка опроса модулей ввода/вывода и алгоритма контроллера. При остановке работает таймер и другие прерывания, основной цикл продолжается по 50 мс. Останавливаются система горячего резерва, обмен данными по сети СУВД и работа драйверов ввода-вывода.

5.4.3.2. Пункт «Запустить/Остановить алгоритм»

Запуск/остановка работы алгоритма контроллера. При остановке работы алгоритма стандартный опрос модулей ввода/вывода продолжается. Работает таймер и другие прерывания, основной цикл продолжается по 50 мс. Если включена система резервирования, а главный процессор имеет режим ведущего, даётся команда на переключение его в режим ведомого. Обмен данными по сети резервирования продолжается.

5.4.3.3. Пункт «Перезапуск контроллера»

Данная команда выдаёт контроллеру команду на перезапуск, инициализацию конфигурации и параметров контроллера.

5.4.3.4. Пункт «Загрузить программу»

Команда на загрузку программ в контроллер (например, тестовых). Здесь также можно производить загрузку новой версии ядра среды исполнения системы «Scorpio», для этого в качестве программы необходимо указать соответствующий файл ядра с расширением **.dxe**.

5.4.3.5. Пункт «Управление соединением»

Пункт «Управление соединением» содержит набор команд управления параметрами соединения:

- Переключатель «Имя канала/Интерфейс»;
- «Изменить». Открывает окно «Настройка подключения»;
- «Включить мониторинг/Отключить мониторинг» Включает/Отключает мониторинг связи с контроллером;
- Обновить. Обновляет индикаторы состояния связи с контроллером.

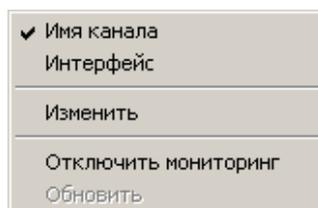


Рис. 5.4.3.5. Подменю «Управление соединением».

5.4.4. Меню «Настройка»

Меню «Настройка» содержит инструменты и команды, предназначенные для настройки различных параметров работы контроллера КСА-02.

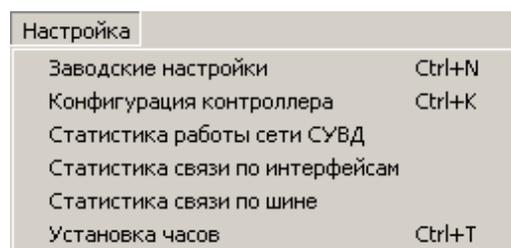


Рис. 5.4.4. Меню «Настройка»

5.4.4.1. Пункт «Заводские настройки»

Пункт «Заводские настройки» предназначен для просмотра заводских настроек контроллера: номера модуля, системного IP, типа Com1, версии загрузчика, системного адреса, локальной часовой зоны.

При выборе пункта «Заводские настройки» на экране монитора открывается окно, вид которого приведен на рисунке.

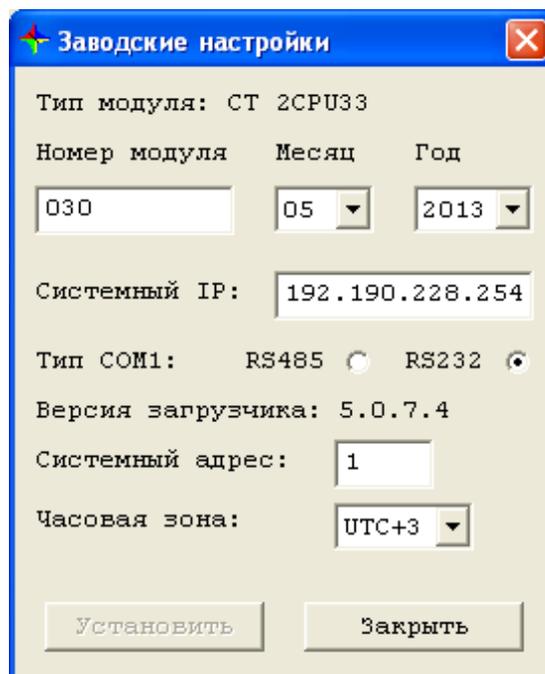


Рис. 5.4.4.1. Окно «Заводские настройки».

Заводской номер контроллера представляет собой число в десятичной системе исчисления и присваивается при выпуске контроллера.

Параметр "**Часовая зона**" показывает разницу между местным временем и всемирным координированным временем (UTC). Внутренние часы контроллера обычно устанавливаются по местному времени. Синхронизация времени, в зависимости от принятого протокола, может осуществляться как по местному времени, так и по UTC. Для случаев, когда синхронизация времени осуществляется по UTC (протокол Proton3-GSM) или производным от UTC (протокол NTP), настройка обеспечивает установку внутренних часов контроллера по местному времени. По-умолчанию принята часовая зона " UTC+3 " - московское время.

5.4.4.2. Пункт «Конфигурация контроллера»

Для отображения конфигурации и осуществления операций конфигурирования контроллера КСА-02 предназначено окно «Конфигурация контроллера».

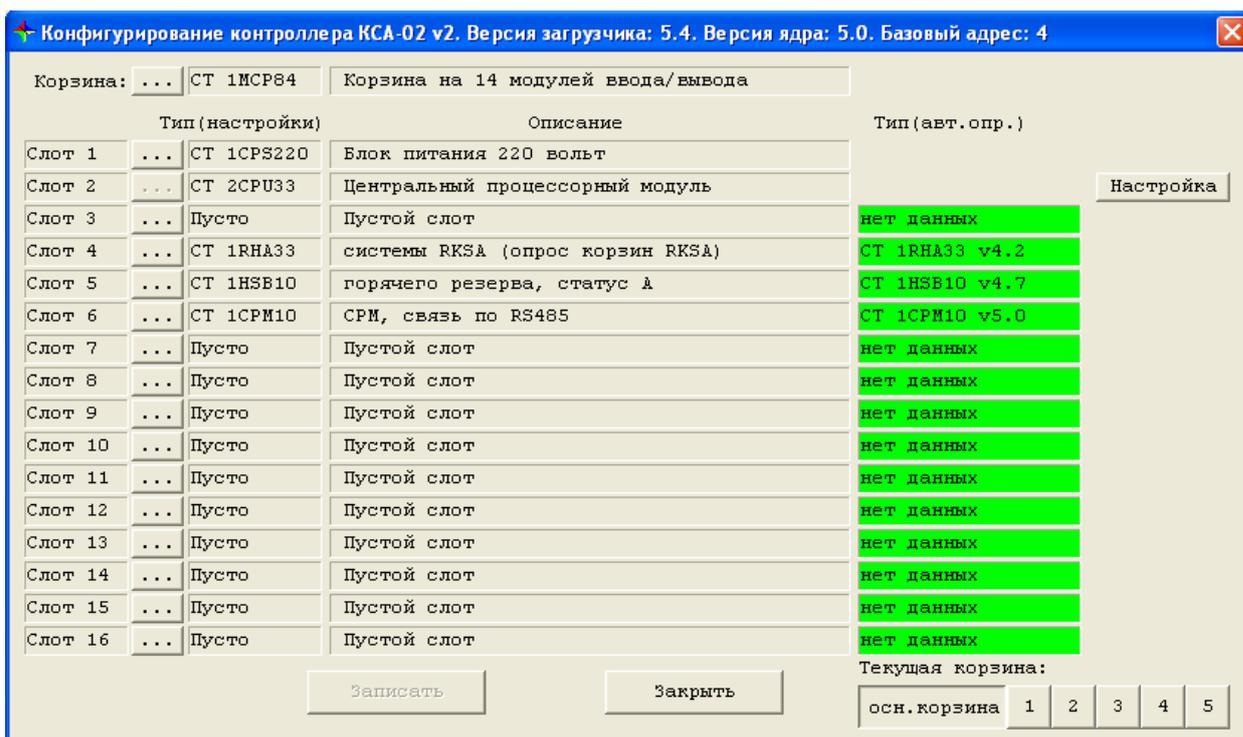


Рис. 5.4.4.2. Окно «Конфигурация контроллера».

Конфигурирование контроллера включает в себя выбор каркаса (корзины), содержащего монтажную кросс-плату, типа и настроек модулей ввода/вывода.

В одной корзине контроллера KCA-02 имеется до 16 слотов. Поддерживаются три типа корзин на 5, 7 и 16 слотов. Слоты с 3-го по 16-й предназначены для модулей ввода/вывода. Подробнее о конфигурировании контроллера см. раздел [5.6. «Конфигурирование контроллера KCA-02»](#).

5.4.4.3. Пункт «Статистика работы сети СУВД»

Окно предназначено для просмотра статистики работы сети СУВД.

Позволяет просмотреть статистику обмена контроллера KCA-02 с корзинами сети СУВД.



Рис. 5.4.4.3. Окно «Статистика работы сети СУВД».

В поле «Всего» отображается общее количество обменов пакетами с главным процессором, включая ошибочные.

В поле «Ошибок» отображается количество ошибочных обменов. Данные непрерывно обновляются.

Цветовыми индикаторами обозначаются признаки наличия связи главного процессора контроллера КСА-02 с корзинами СУВД. Они могут принимать следующие значения:

- **Серый** — корзина отсутствует в параметрах системы удаленного ввода;
- **Красный** — связь с корзиной не установлена;
- **Зелёный** — связь с корзиной установлена.

Если в параметрах системы удаленного ввода не установлено резервирование линии связи, то для порта COM2 все корзины будут указаны как отсутствующие.

5.4.4.4. Пункт «Статистика связи по интерфейсам»

Окно предназначено для просмотра статистики работы интерфейсных модулей СТ1СРМ10 и СТ1СРЕ10, а также процессорного модуля:

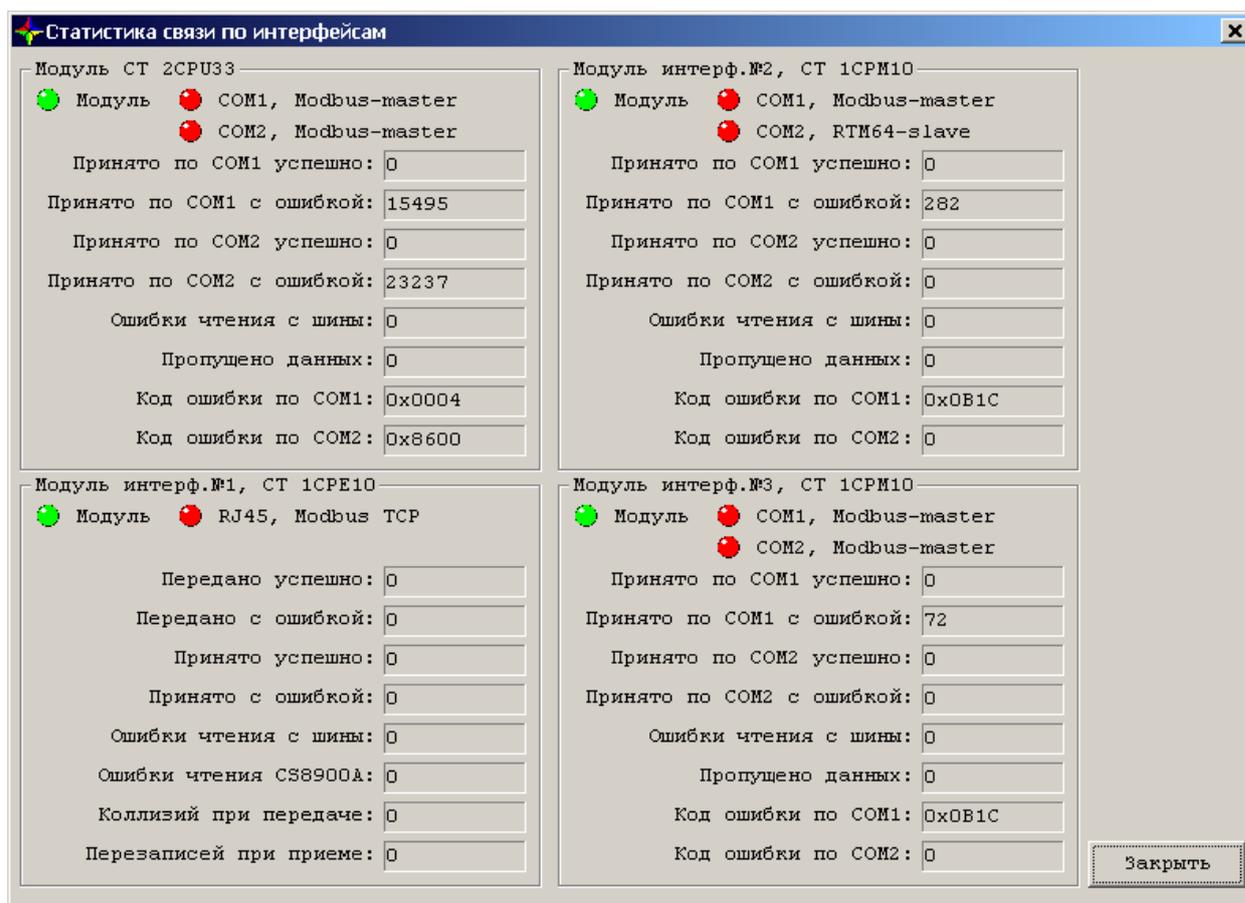


Рис. 5.4.4.4. Окно «Статистика связи по интерфейсам».

Внешний вид окна «Статистика работы модулей CPN» меняется в зависимости от количества подключенных интерфейсных модулей CPN и порядка их нахождения в корзине контроллера КСА-02. Максимальное количество интерфейсных модулей в одной корзине КСА-02 — три.

В поле «Принято по COM1 (2) успешно» отображается количество пакетов обмена с главным процессором, успешно принятых через порт COM1 (2).

В поле «Принято по COM1 (2) с ошибкой» отображается количество пакетов обмена с главным процессором, принятых через порт COM1 (2) с ошибкой.

Ошибки чтения с шины: количество ошибочных обменов по шине.

Пропущено данных: возможны ситуации, когда буфер обмена модуля полностью занят и не может принять все поступающие данные, которые передаются ему по шине. Часть данных оказывается пропущенными; их количество отображается в поле «Пропущено данных».

Поле «Код ошибки по COM1 (2)» отображает код ошибки.

Признаки наличия связи может принимать следующие значения:

- **Красный** — связь не установлена;
- **Зелёный** — связь установлена.

5.4.4.5. Пункт «Статистика связи по шине»

Окно предназначено для просмотра статистики обмена пакетами по внутренней шине контроллера КСА-02 между процессорным модулем и модулями СТ1HSB10 (СТ2HSB10), СТ1RNA33, СТ1CPM10, СТ1CPE10.

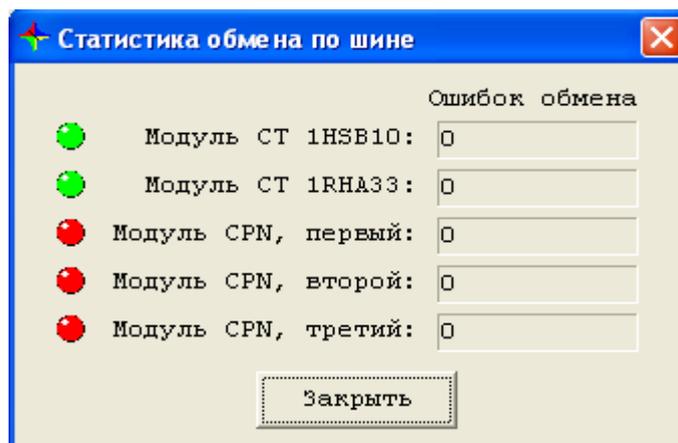


Рис. 5.4.4.5. Окно «Статистика обмена по шине».

Поле «Ошибок обмена» отображает количество ошибочных обменов. Данные непрерывно обновляются.

Признаки наличия связи могут принимать следующие значения:

- **Красный** — связь не установлена, модуль неисправен или физически отсутствует в корзине головного процессора контроллера КСА-02;
- **Зелёный** — связь с модулем установлена.

5.4.4.6. Пункт «Установка часов»

Окно предназначено для просмотра текущего значения внутренних часов контроллера и его синхронизации с текущим значением системных часов подключенной ПЭВМ.

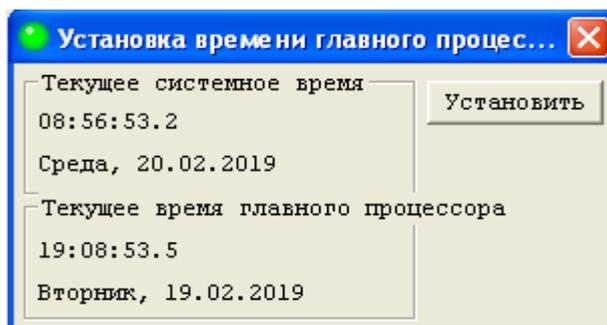


Рис. 5.4.4.6. Окно «Установка часов».

Для изменения текущего значения внутренних часов контроллера необходимо нажать кнопку «Установить». Для закрытия окна служит кнопка «Закрыть».

5.4.5. Меню «Алгоритм»

Меню «Алгоритм» предназначено для работы с алгоритмами. Оно позволяет загружать и удалять алгоритмы в главном процессоре, а также отлаживать выполнение алгоритма с помощью окон, позволяющих просматривать и изменять в режиме реального времени значение входов функций алгоритма.

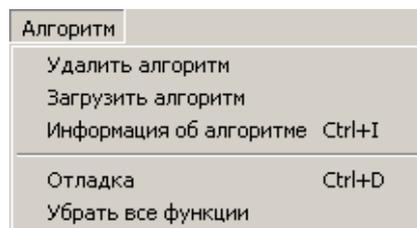


Рис. 5.4.5.1. Меню «Алгоритм».

Пункт меню «Загрузить алгоритм» становится доступными только при открытом проекте.

5.4.5.1. Пункт «Удалить алгоритм»

Служит для удаления алгоритма из контроллера; перед выполнением команды программа запросит подтверждение действия.

5.4.5.2. Пункт «Загрузить алгоритм»

Служит для загрузки алгоритма в контроллер. Алгоритм загружается из открытого в программе проекта. Процесс загрузки алгоритма отображается на индикаторе выполнения, также указывается процент выполненной загрузки.

Пункт меню доступен только при открытом проекте.

5.4.5.3. Пункт «Информация об алгоритме»

Предназначен для отображения информации об установленном процессорном модуле главного процессора и версии его программного обеспечения, об алгоритме, открытом в программе, и об алгоритме, который загружен в контроллер.

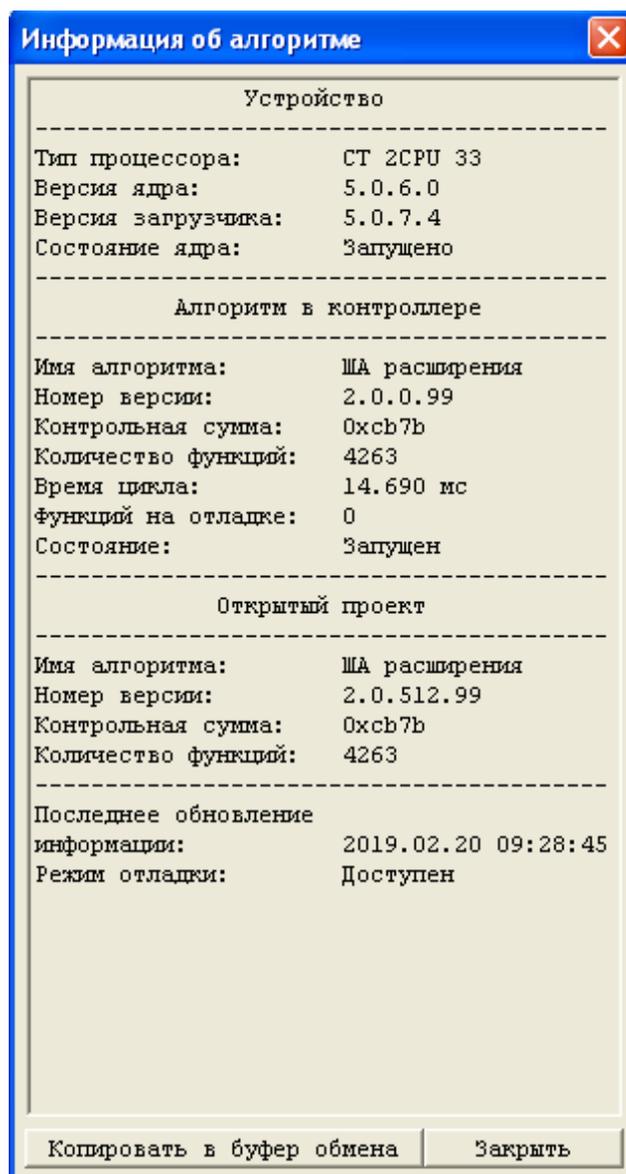


Рис. 5.4.5.3. Окно «Информация об алгоритме».

5.4.5.4. Пункт «Отладка»

Служит для добавления на отладку функций алгоритма, и снятия функций с отладки.

При выборе пункта меню «Отладка» на экране появляется окно «Постановка функции на отладку», содержащее список функций, доступных для добавления на отладку. Кроме того, в правой части окна располагается:

- поле, в котором отображается количество функций на отладке (количество функций, стоящих на отладке, может быть не более 32);
- кнопка «Добавить», служащая для добавления функции на отладку;
- кнопка «Перейти», служащая для перехода к отладке функции;
- кнопка «Снять», служащая для снятия функции с отладки;

- кнопка «Снять все», служащая для снятия с отладки всех функций;
- поле ввода номера функции и кнопка «Добавить по номеру» для добавления функции на отладку по ее номеру;
- кнопка «Закрыть», служащая для закрытия окна «Постановка функции на отладку».

Возможен также альтернативный способ добавления на отладку функции открытого проекта. Для этого необходимо открыть диаграмму проекта в программе «FBD_Builder», выделить нужные функции на диаграмме и нажать клавишу F6. Выделенные функции отобразятся в окне сервисной программы.

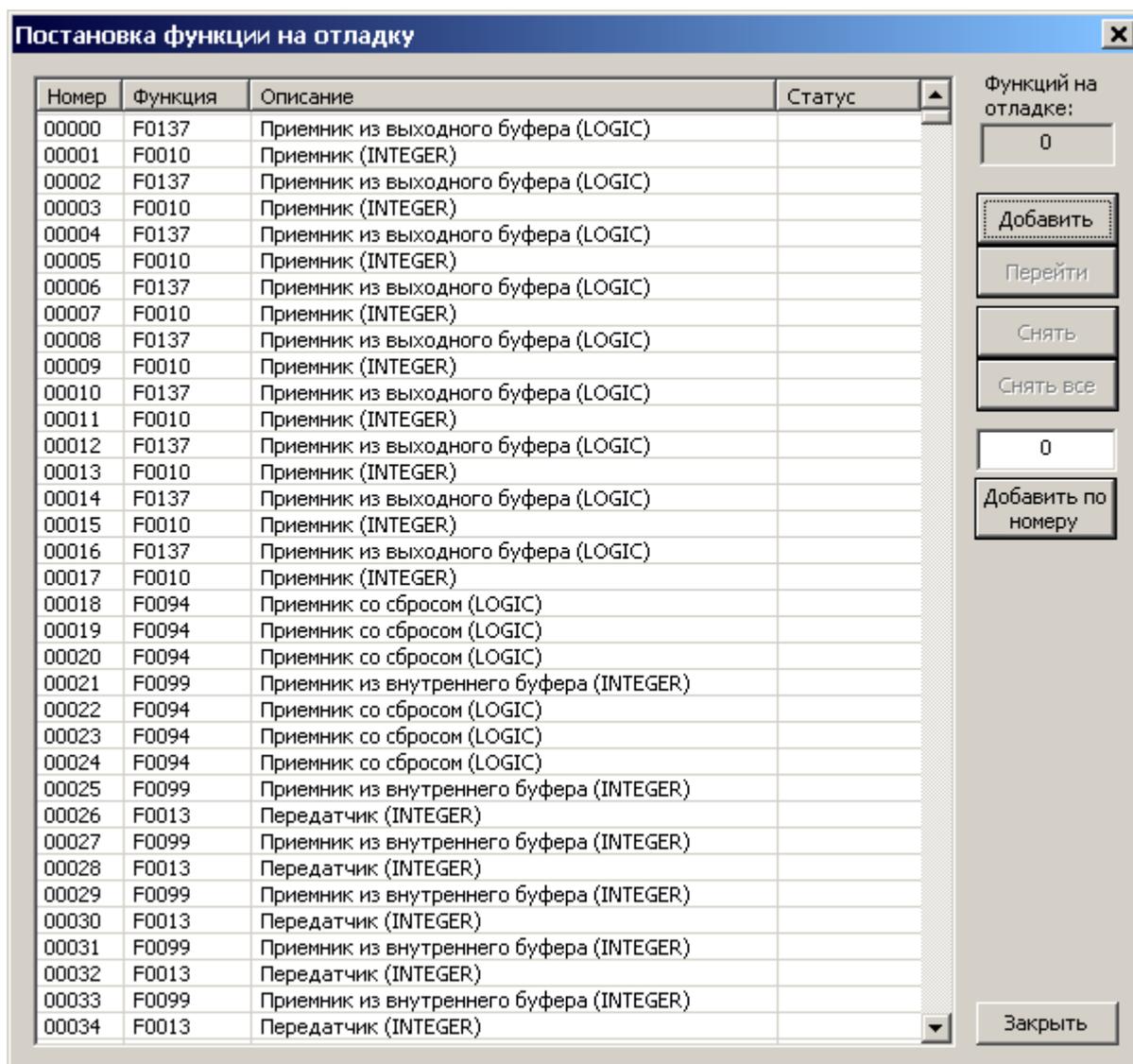


Рис. 5.4.5.4. Окно «Постановка функций на отладку»

5.4.5.5. Пункт «Убрать все функции»

Служит для удаления с отладки всех функций алгоритма одновременно. Окна отладки функций будут закрыты.

5.4.6. Меню «Тесты»

Меню «Тесты» содержит инструменты и команды, предназначенные для работы с тестовым обеспечением контроллера КСА-02.

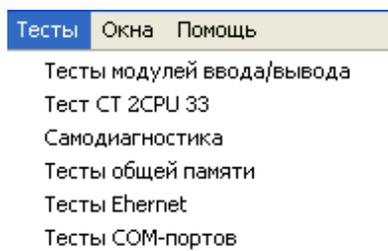


Рис. 5.4.6.1. Меню «Тесты».

5.4.6.1. Пункт «Тесты модулей ввода/вывода»

Тесты модулей ввода/вывода предназначены для тестирования модулей ввода/вывода контроллера в соответствии с конфигурацией контроллера. Подробнее в п. [5.9.1. Работа в режиме «Тесты модулей ввода/вывода»](#).

5.4.6.2. Пункт «Тест СТ2CPU33»

В рамках теста модуля СТ2CPU33 проводятся следующие виды тестирования:

- **Тест процессора:** тестирование процессора модуля СТ2CPU33.
- **Тест ОЗУ:** тестирование памяти процессора модуля СТ2CPU33 (DM, PM).
- **Тест SRAM/NVRAM:** тестирование памяти алгоритма и энергонезависимой памяти (SRAM и NVRAM).
- **Тест Flash:** тестирование памяти Flash.
- **Тест COM1:** тестирование обмена по интерфейсу COM1.
- **Тест COM2:** тестирование обмена по интерфейсу COM2.

Подробнее в п. [5.9.2. Тест модуля СТ2CPU33](#).

5.4.6.3. Пункт «Самодиагностика»

Самодиагностика контроллера КСА-02 предназначена для диагностирования состояния контроллера КСА-02. Подробнее в п. [5.9.3. Самодиагностика](#).

5.4.6.4. Пункт «Тест общей памяти»

Предназначен для тестирования общей памяти модулей CPN, HSB или RHA контроллера КСА-02. Подробнее в п. [5.9.4. Тест общей памяти](#).

5.4.6.5. Пункт «Тест Ethernet»

Тест проверяет состояние связи контроллера КСА-02 по сети Ethernet. Подробнее в п. [5.9.5. Тест Ethernet](#).

5.4.6.6. Пункт «Тест СОМ-портов»

Тест проверяет состояние связи между портами СОМ1 и СОМ2 путём передачи тестовых пакетов данных. Подробнее в п. [5.9.6. Тест СОМ-портов](#).

5.4.7. Меню «Окна»

Меню «Окна» предназначено для изменения порядка расположения открытых окон на экране монитора.

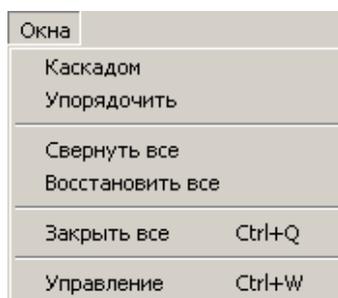


Рис. 5.4.7.1. Меню «Окна».

5.4.7.1. Пункт «Каскадом»

Каждое следующее открытое окно перекрывает часть предыдущего открытого окна.

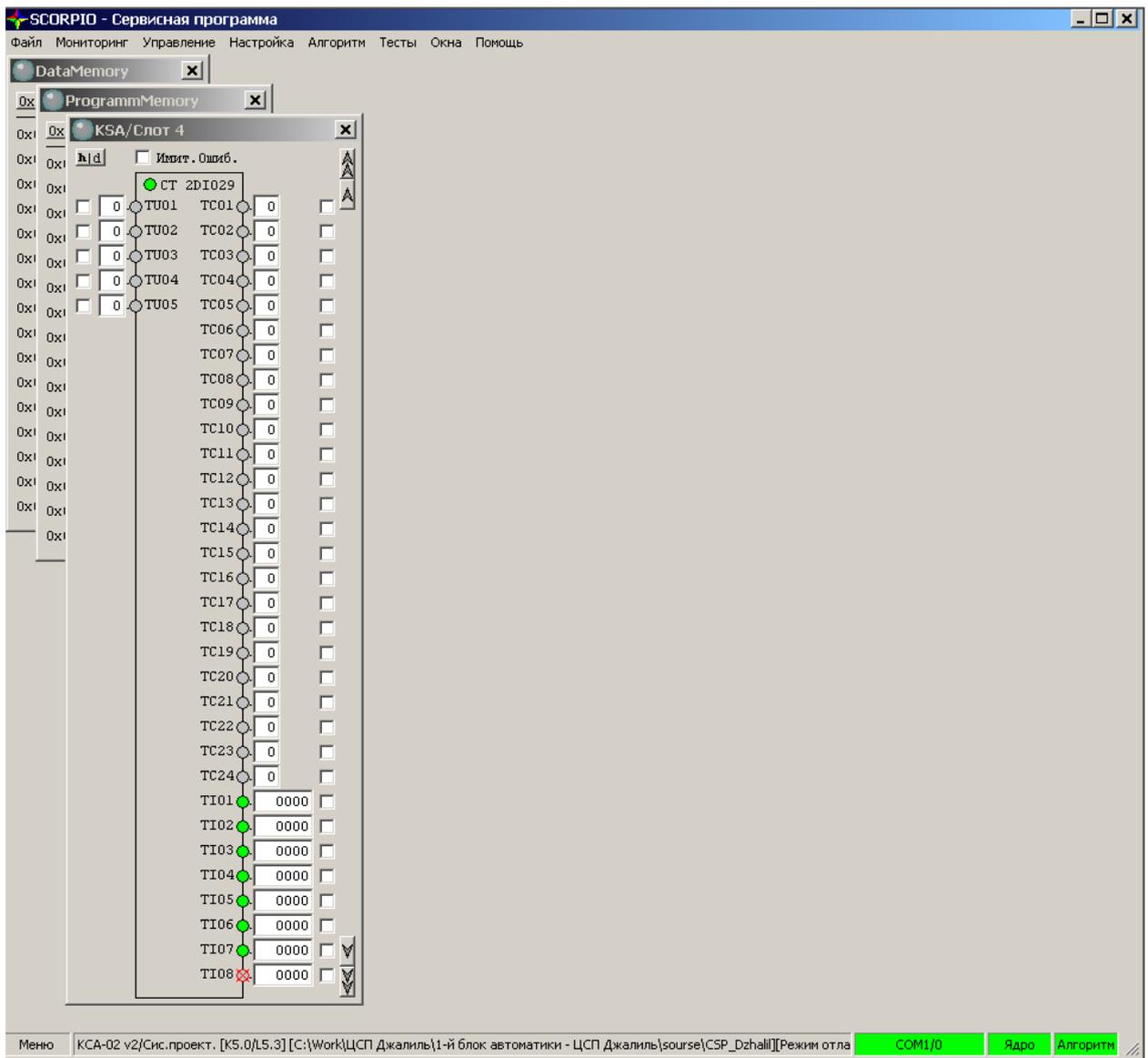


Рис. 5.4.7.1.1. Расположение окон сервисной программы в результате применения команды «Каскадом».

5.4.7.2. Пункт «Упорядочить»

Расположение открытых окон рядом друг с другом без перекрытия предыдущего окна.

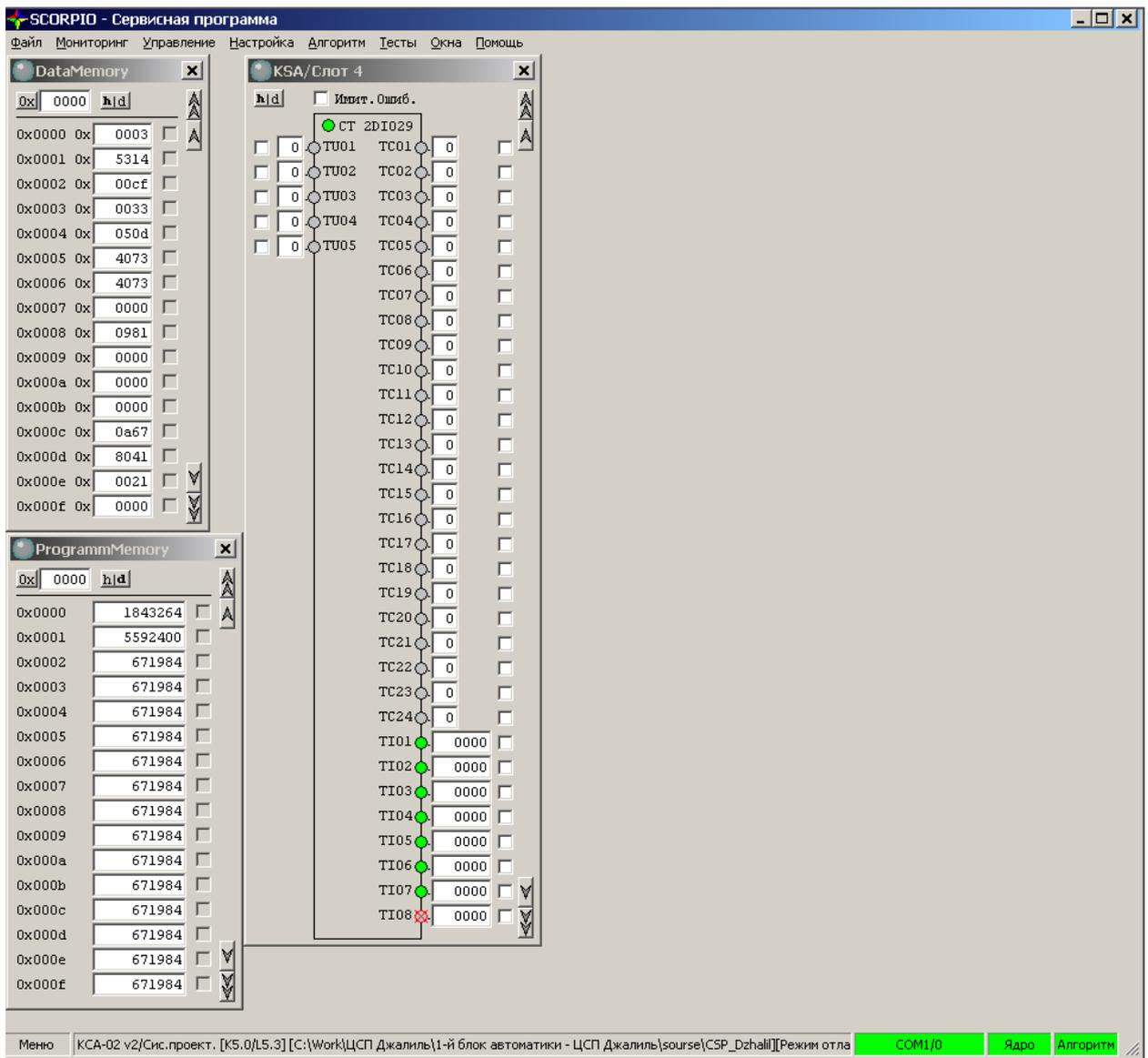


Рис. 5.4.7.2.1. Расположение окон сервисной программы в результате применения команды «Упорядочить».

5.4.7.3. Пункт «Свернуть все»

Все открытые окна будут свернуты, и будут расположены в нижней части экрана.



Рис. 5.4.7.3.1. Свернутое окно.

5.4.7.4. Пункт «Восстановить все»

Все свернутые окна будут восстановлены на экране.

5.4.7.5. Пункт «Закрыть все»

Все открытые окна будут закрыты.

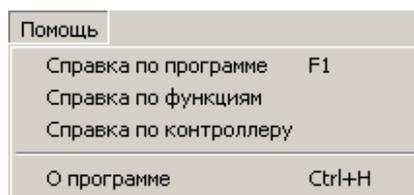


Рис. 5.4.8.1. Меню «Помощь».

Для выбора необходимой информации из справочной системы сервисной программы предназначено меню, отображаемое на экране монитора при выборе пункта основного меню «Помощь».

5.4.8.1. Пункт «Справка по программе»

Справочная информация о работе сервисной программы контроллера КСА-02.

Справка по всем меню сервисной программы:

- справка о назначении полей и кнопок использующихся окон;
- правила работы с сервисной программой (использование «горячих» клавиш и т.п.).

5.4.8.2. Пункт «Справка по функциям»

Справочная информация о библиотеке функций.

- Полный список поддерживаемых системой «Scorpio» функций с подробным описанием их назначения, принципа действия и внешнего вида соответствующих блоков.

5.4.8.3. Пункт «Справка по контроллеру»

Справочная информация о модулях ввода/вывода.

- Полный список поддерживаемых системой «Scorpio» модулей ввода/вывода с описанием их назначения.

5.4.8.4. Пункт «О программе»

Краткая справочная информация о программе, её версии и дате создания.

5.5. Старт сервисной программы контроллера КСА-02

5.5.1. Общие сведения о работе с сервисной программой контроллера КСА-02

Перед работой с контроллером КСА-02 средствами сервисной программы необходимо подключить к последовательному порту ПЭВМ с одной стороны и к COM3 на лицевой панели модуля с другой, затем подать питание на контроллер.

После включения питания осуществляется процесс загрузки рабочей программы в оперативную программную память процессорного модуля, а затем - процесс самодиагностики. При выполнении процессов загрузки и самодиагностики, на лицевой панели процессорного модуля загораются индикаторы «COM1», «COM2», «RUN» и «ERROR».

Если процесс самодиагностики завершился без ошибок, то индикатор «RUN» будет мигать, а индикаторы «ERROR», «COM1», «COM2» погаснут. Если в процессе самодиагностики контроллера КСА-02 возникла ошибка, то индикатор «ERROR» («Ошибка») начнет мигать, а индикатор «RUN» погаснет, причём количество миганий индикатора «ERROR» будет равно коду возникшей ошибки:

- 1– ошибка контрольной суммы Program memory (неправильная загрузка программы из Flash);
- 2– ошибка контрольной суммы постоянных данных и конфигурационной таблицы (неправильная загрузка из ПЗУ процессора);
- 3– ошибка контрольной суммы энергонезависимого ОЗУ (алгоритма);
- 4– превышение количества допустимых команд в буфере команд обмена по шине ввода-вывода;
- 5– ошибка конфигурации или часов реального времени;
- 6– превышение времени опроса периферийных устройств (не уложились в один цикл);
- 7– отсутствует в конфигурации, либо неисправен модуль СТ1HSB10 (СТ2HSB10), при включенной системе резервирования;
- 8– в процессе работы с включенной системой резервирования модуль СТ1HSB10 (СТ2HSB10) не отвечает или неисправен;
- 9– неисправность или отсутствует в конфигурации модуль СТ1RNA33 при включенных системах резервирования и удаленного ввода/вывода;

10– отсутствует связь со всеми СУВД при включенных системах резервирования и удаленного ввода/вывода;

11– установлены разные протоколы обмена в ядре и конфигурации контроллера;

12– не соответствуют число или типы модулей CPN в алгоритме и в конфигурации контроллера;

13– ошибка инициализации архивов;

14– протоколы CPE модулей, установленных в контроллере, несовместимы с заданными в алгоритме;

15– (в текущей версии ПО данная ошибка отсутствует);

16– ошибка теста Data Memory;

17– ошибка заводских настроек.

Среднее время выполнения процессов загрузки и самодиагностики составляет 5 - 10 секунд. Если процессы загрузки и самодиагностики не закончены за значительно больший интервал времени (например, постоянно горит индикатор «ERROR»), то это свидетельствует о серьезной аппаратной неисправности процессорного модуля. Такой модуль должен быть отправлен в ремонт.

Затем следует запустить на ПЭВМ исполняемый файл сервисной программы ServiceProgram. На экране откроется основное окно сервисной программы контроллера КСА-02 с отображением окна «Настройка подключения» для выбора последовательного асинхронного порта ПЭВМ и адреса контроллера.

5.5.2. Настройка подключения ПЭВМ к контроллеру КСА-02

Выбор имени канала в окне «Настройка подключения» осуществляется в зависимости от того, к какому из портов ПЭВМ подключен интерфейсный жгут.

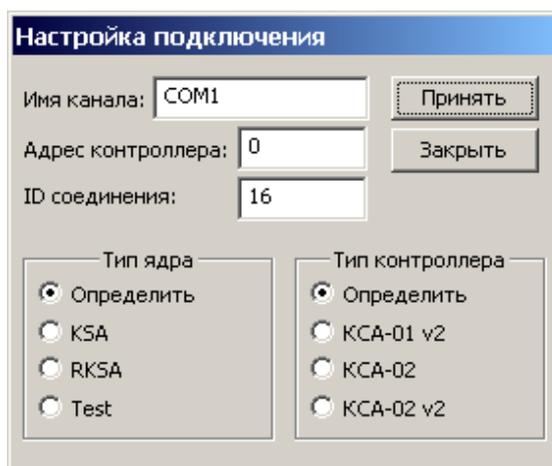


Рис. 5.5.2.1. Окно «Настройка подключения».

5.5.3. Запись параметров подключения контроллера KCA-02 к ПЭВМ

Параметры подключения контроллера KCA-02 к ПЭВМ записываются в файл config.ini, который создаётся автоматически сервисной программой при нажатии на кнопку «ОК» в окне «Параметры подключения». При этом окно «Параметры подключения» закрывается, а пункты меню оболочки сервисной программы контроллера KCA-02 становятся доступными для пользователя.

5.5.4. Индикация результатов старта сервисной программы

Индикация установления связи ПЭВМ с контроллером KCA-02 осуществляется цветом индикатора «COM1», расположенной в правой части информационной строки основного окна, следующим образом:

- **Зелёный** — связь есть;
- **Красный** — ошибка связи.



Рис. 5.5.4.1. Индикация установления связи ПЭВМ с контроллером KCA-02.

При наведении на информационную строку всплывает сообщение, содержащее справочную информацию об алгоритме контроллера и загруженном проекте.

Устройство: KCA-02 v2	
Программа:	Сис.проект.
Версия ядра:	5.0
Версия загрузчика:	5.4
Текущий проект:	ША расширения
Каталог проекта:	C:\Public\от Мордовина\BA4\
Режим отладки:	Доступен

Рис. 5.5.4.2. Всплывающее сообщение.

5.6. Конфигурирование контроллера КСА-02

Для отображения конфигурации и осуществления операций конфигурирования контроллера КСА-02 предназначено окно «Конфигурация контроллера».

Конфигурирование контроллера включает в себя операции по выбору каркаса (корзины), содержащего монтажную кросс-плату, а также выбора типа и настроек модулей ввода/вывода.

В одной корзине контроллера КСА-02 имеется до 16 слотов. Поддерживаются три типа корзин на 5, 7 и 16 слотов. Слоты с 3-го по 16-й предназначены для модулей ввода/вывода.

5.6.1. Вызов окна «Конфигурация контроллера»

Чтобы вызвать окно «Конфигурация контроллера» на экран монитора, необходимо в меню «Настройка» выбрать пункт «Конфигурация контроллера». Откроется окно:



Рис. 5.6.1. Окно «Конфигурация контроллера» при подключении к основной корзине контроллера.

5.6.2. Информационные поля и кнопки окна «Конфигурация контроллера»

Окно «Конфигурация контроллера» построено по типу таблицы. В этой таблице отображаются строки и столбцы, предназначенные для конфигурации модулей, а также кнопки:

- столбец «Слот XX», где XX — десятичное число от 1 до 16, обозначающее номер слота контроллера КСА-02;
- кнопка  вверху слева (в строке «Корзина») — кнопка выбора типа модуля и типа корзины. Тип корзины далее выбирается из списка (кнопка ).

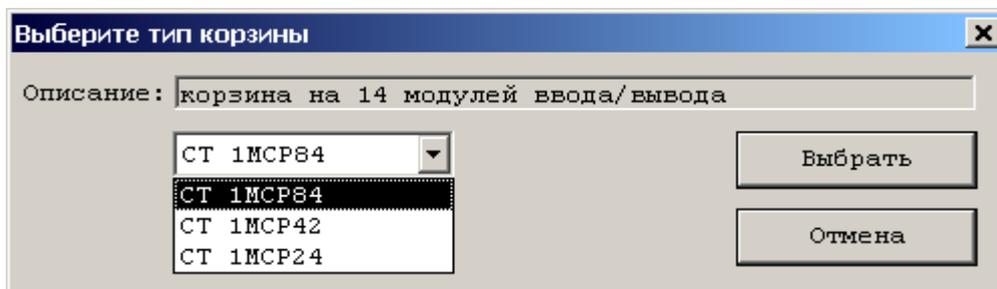


Рис. 5.6.2.1. Окно выбора типа корзины.

- Столбец «**Тип (настройки)**» — отображает наименования модулей, находящихся в слотах контроллера КСА-02, номера которых указаны в столбце «Слот XX». Если слот пуст, отображается строка «Пусто».
- Столбец «**Описание**» — предназначен для отображения краткого описания модуля, находящегося в слоте контроллера КСА-02, номер которого указан в поле «Слот XX». Если в слоте нет модуля, отображается строка «Пустой слот».
- Столбец «**Типы (авт.опр.)**» — предназначен для отображения соответствия реально существующей конфигурации корзины модуля и конфигурации, указанной пользователем. Возможные значения: «**Нет данных**» — конфигурация не определена (слот пуст, либо данные не обновлены); «**Ошибка**» — если конфигурации не совпадают; «**Нет ядра**» — если отсутствует соединение программы с контроллером. Если конфигурации совпадают, соединение установлено и ошибок нет, то в ячейках столбца будут отображаться названия модулей, имеющих в слотах корзины контроллера, а также версия их программного обеспечения.
- Кнопка «**Настройка**» — отображается не для всех модулей ввода/вывода и предназначена для задания их настроек.
- Группа «**Текущая корзина**» — при подключении сервисной программы к основной корзине позволяет провести выбор между настройкой конфигурации основной корзины и просмотром настройки конфигурации корзин СУВД №1 - №5, которая записана в память процессорного модуля корзины главного процессора. При подключении сервисной программы к корзине СУВД название и вид этой группы

меняется, возможна реальная настройка конфигурации модулей подключенной корзины и её адреса.

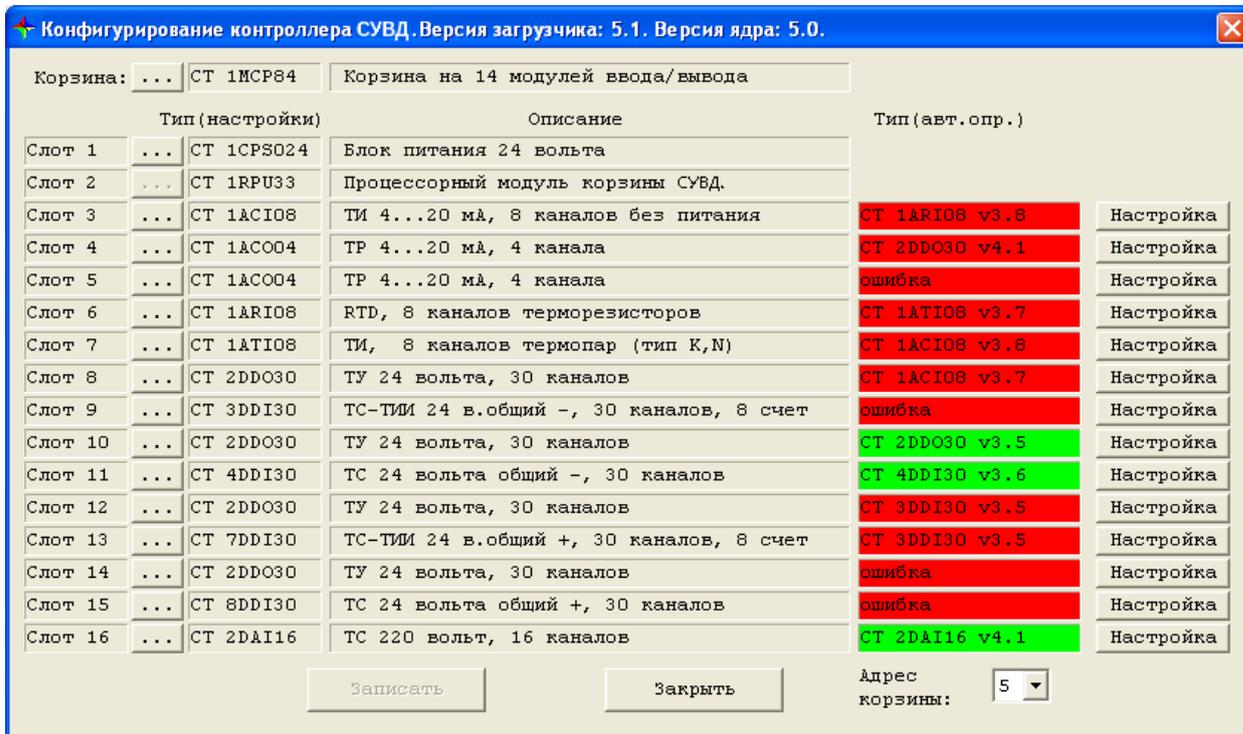


Рис. 5.6.2.2. Окно «Конфигурация контроллера» при подключении к корзине СУВД.

В нижней части окна расположены кнопки «**Записать**» и «**Закрыть**».

Кнопка «**Записать**» предназначена для записи в память конфигурации корзины, изменённой пользователем, и доступна в том случае, если конфигурация, отображаемая в окне, отличается от конфигурации, записанной в памяти процессорного модуля (это определяется сервисной программой контроллера КСА-02). При нажатии на кнопку «**Записать**» осуществляется передача блока конфигурации в процессорный модуль, а затем пользователю предлагается его перезагрузить. Для вступления в силу проведенных изменений следует нажать кнопку "Да".

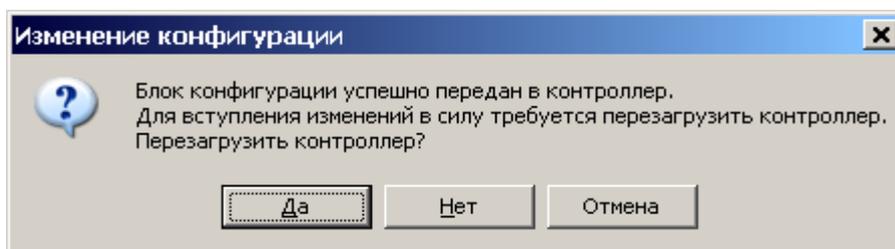


Рис. 5.6.2.3. Окно «Изменение параметров».

Кнопка «**Закрыть**» предназначена для завершения работы в режиме «Конфигурация контроллера».

5.7. Настройка модулей контроллера КСА-02

Настройка модуля типа СТ1АСИ08, СТ1АСО04, СТ1АРИ08, СТ1АТИ08, СТ1ВСТ02, СТ1ВСТ03, СТ1ДИО29, СТ3ДДИ30, СТ2ДДО30, СТ1ДАИ16 доступна только при наличии в контроллере ядра системы «Scorpio». Индикация наличия ядра системы проектирования осуществляется следующим образом:

- на лицевой панели корпуса контроллера КСА-02 мигает светодиод «RUN»;
- на экране монитора в нижней информационной строке основного окна сервисной программы контроллера КСА-02 индикатор «Ядро» имеет зеленый или желтый цвет.

Все действия по настройке модулей осуществляются в соответствующих окнах:

- окно настройки модуля СТ1АСИ08;
- окно настройки модуля СТ1АСО04;
- окно настройки модуля СТ1АРИ08;
- окно настройки модуля СТ1АТИ08;
- окно настройки модуля СТ1ВСТ02, СТ1ВСТ03;
- окно настройки модуля СТ1ДИО29, СТ2ДИО29, СТ3ДИО29;
- окно настройки модуля СТ3ДДИ30 – СТ10ДДИ30;
- окно настройки модуля СТ2ДДО30;
- окно настройки модуля СТ1ДАИ16.

Настройка модулей ввода/вывода предполагает установку различных параметров модулей. Значения параметров таких, как тип входного сигнала, количество бит выходного кода, ограничение шкалы (для модулей СТ1АСИ08, СТ1АРИ08, СТ1АТИ08), выбираются, исходя из удобства обработки цифрового кода в системе сбора и обработки данных.

Калибровка измерительных каналов модулей аналогового ввода/вывода осуществляется с помощью калибратора и в соответствии с правилами его использования при калибровке измерительных каналов модулей аналогового ввода/вывода, изложенными в руководстве по эксплуатации контроллера КСА-02 (НБКГ.466543.003 РЭ).

5.7.1. Настройка модулей СТ1АСИ08

Модуль СТ1АСИ08 преобразует входной аналоговый сигнал в выходной цифровой код по восьми каналам.

Преобразование входного сигнала осуществляется на основании комбинации таких параметров модуля, как тип входного сигнала, количество бит выходного кода,

ограничение шкалы. Также на основании настройки модуля формируется значение инструментальной ошибки модуля (один из факторов его неисправности).

Заголовок окна настройки модуля СТ1АС108 называется «Модуль СТ1АС108, слот XX», где XX — номер слота, в котором находится модуль СТ1АС108.

Модуль СТ 1АС108, слот 9

Общие
Заводской №: 1023 02 2019
Дата выпуска: 26 02 2019
Записать
Закреть

Диапазон измерения
 4 - 20 мА
 0 - 20 мА
 0 - 5 В
 1 - 5 В

Калибровка
Выбор канала: 1
Состояние: не откалиброван
Калибровка

Ограничение шкалы
 Ограничение снизу
 Ограничение сверху

Инструментальная ошибка
снизу: 3.900 мА
сверху: 20.100 мА

Количество бит выходного кода: 12

Время усреднения, мс
Канал 1: 1000
Канал 2: 500
Канал 3: 200
Канал 4: 200
Канал 5: 20
Канал 6: 500
Канал 7: 500
Канал 8: 500

Метрологически значимая часть
Версия метрологически значимой части: 1.0
Контрольная сумма метрологически значимой части: 0x6441

Рис. 5.7.1.1. Окно настройки модуля 1СТАС108.

Чтобы вызвать это окно, необходимо в окне «Конфигурация контроллера» нажать кнопку «Настройка», расположенную в строке с номером слота, содержащего данный модуль.

В окне настройки модуля СТ1АС108 расположены кнопки «Записать» и «Закреть». Кроме того, окно настройки модуля СТ1АС108 по параметрам настройки разделено на следующие группы: Общие; Диапазон измерения; Калибровка; Ограничение шкалы; Инструментальная ошибка; Количество бит выходного кода; Время усреднения.

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Диапазон измерения: модуль может работать в двух режимах:

- в режиме измерения тока по каналам в диапазонах 0 — 20мА и 4 — 20мА;
- в режиме измерения напряжения по каналам в диапазонах 0 — 5 В и 1 — 5В.

Режим работы модуля устанавливается на модуле внешними переключками (см. руководство по эксплуатации контроллера КСА-02 (НБКГ.466543.003 РЭ)), в соответствии с этим выбирается диапазон измерения при настройке модуля средствами сервисной программы.

Калибровка: каждый канал на модуле СТ1АСI08 имеет индивидуальную калибровку. Калибровка канала проводится всегда по току. На основании калибровки определяются калибровочные коэффициенты для каждого канала.

Область окна настройки модуля СТ1АСI08 «Калибровка» предназначена для калибровки каждого из восьми каналов модуля СТ1АСI08 и содержит следующие поля и кнопки:

- **«Выбор канала»:** при нажатии на кнопку  отображается список номеров каналов модуля СТ1АСI08 от 1 до 8. Из этого списка выбирается номер канала.
- **«Состояние»:** отображается состояние выбранного канала модуля СТ1АСI08; возможные значения: «откалиброван» и «не откалиброван».
- **«Калибровка»:** калибровка канала осуществляется по двум калибровочным значениям тока (для этого необходимо подать соответствующий ток с [поверенного прибора](#) на вход нужного модуля), которые устанавливаются в соответствующих полях окна «Калибровка канала», (открывается при нажатии на кнопку «Калибровка»).

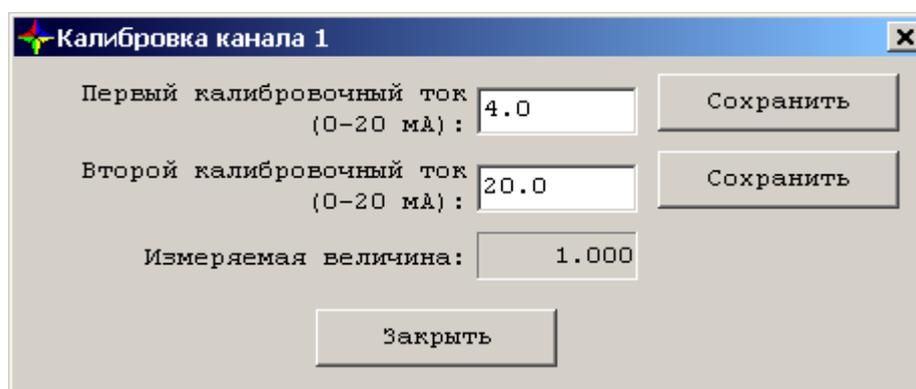


Рис. 5.7.1.2. Окно настройки калибровки канала.

Значение первого калибровочного тока должно быть меньше значения второго калибровочного тока.

Рекомендуется устанавливать значение первого калибровочного тока близко к нижней границе диапазона калибровочного тока, а значение второго калибровочного тока — близко к верхней границе диапазона.

Чтобы установить калибровочное значение, нужно в окне «Калибровка канала» ввести необходимое значение, а затем нажать на кнопку «Сохранить».

При установке неверного значения (несоответствие диапазону калибровочного тока) на экране монитора появляется окно с предупреждающим сообщением.

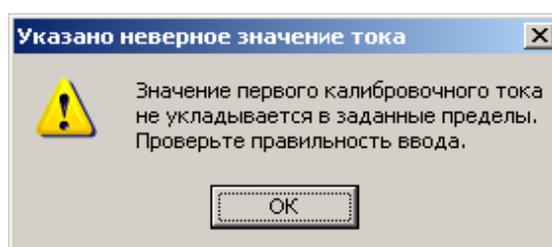


Рис. 5.7.1.3. Окно сообщения об ошибке.

После закрытия окна «Калибровка канала» в строке «Состояние» окна настройки модуля СТ1АСI08 появится сообщение о результате калибровки канала: «откалиброван» или «не откалиброван».

Если все действия пользователя были правильными, а в строке «Состояние» отображается сообщение «не откалиброван», то это означает, что модуль неисправен.

Ограничение шкалы: флажок «Ограничение шкалы» определяет ограничение диапазона выходного цифрового кода его номинальным диапазоном. Возможны следующие варианты ограничений: «Ограничение снизу» и «Ограничение сверху».

Модуль СТ1АСI08 преобразует следующие номинальные входные сигналы:

- Ток в диапазонах от 0 до 20мА, от 4 до 20мА;
- Напряжение в диапазонах от 0 до 5 В, от 1 до 5В.

Установка флажка «Ограничение снизу» означает, что входной сигнал менее 4мА (1В) преобразуется в выходной код, соответствующий входному сигналу 4мА (1В). То есть если входной сигнал будет равен 0, то выходной код всё равно будет соответствовать 4мА, и сигналом о неполадке здесь станет инструментальная ошибка (канал будет определён как неисправный).

Установка флажка «Ограничение сверху» означает, что выходной код не может превысить верхней границы, даже если входной сигнал превышает объявленный диапазон (технически возможно измерение тока до 20,16мА).

Если флажок не установлен, а входной ток превышает объявленный номинальный диапазон, то выходной код превысит верхнюю границу номинального выходного кода (кроме случая 16-битного выходного кода).

Инструментальная ошибка: инструментальная ошибка определяется снизу и сверху. В случае, если величина входного сигнала меньше, чем значение инструментальной ошибки снизу или больше, чем значение инструментальной ошибки сверху, соответствующий канал модуля СТ1АСI08 считается неисправным.

Значения инструментальной ошибки можно изменять вручную, для этого надо ввести нужные значения в поле «Инструментальная ошибка» и зафиксировать изменения нажатием клавиши «Enter».

Количество бит выходного кода: настройка устанавливает соответствие между входным измеренным значением аналогового сигнала и цифровым кодом, полученным в результате его преобразования. При этом 0 соответствует нижнему значению шкалы измерения, а для верхнего числа шкалы измерения имеется следующее соответствие:

Количество бит	Верхнее значение номинального цифрового кода
12	0x0FFF
13	0x1FFF
14	0x3FFF
15	0x7FFF
16	0xFFFF

Однако надо иметь в виду, что данная настройка не ограничивает допустимого диапазона цифрового кода, поэтому возможно превышение верхнего значения номинального цифрового кода.

Требуемое количество бит выходного кода выбирается в строке «Количество бит выходного кода» из списка при нажатии на кнопку .

Время усреднения: интервал времени в миллисекундах, за который осуществляется усреднение сигнала по каналу. Усреднение сигнала — это специальная функция фильтра, которая позволяет «смягчить» колебания сигнала, замедляя на определённый промежуток времени изменение его показаний. Таким образом получается, что при скачках сигнала вверх-вниз показания сигнала будут выглядеть гораздо более ровными (пока показания медленно дорастают до высокого показателя, сигнал успеет упасть, и показания, так и не дойдя до верхней точки, снова начнут снижаться). Допустимые значения времени усреднения из ряда: 20, 200, 500, 1000, 2000, 5000. Если пользователь вводит другое значение, оно будет изменено на ближайшее допустимое значение.

Замечание 1. Обновление выходных данных за время усреднения осуществляется четыре раза; например, если выбрано время усреднения 200 мс, то обновление данных осуществляется через 50 мс.

Замечание 2. Если желателен режим "совсем без усреднения" следует выбрать значение 20 мс. Обновление выходных данных будет осуществляться через 5 мс. Необходимо отметить, что этот интервал значительно меньше цикла опроса модуля ядром (50 мс).

В нижней части окна располагается идентификационная информация о метрологически значимой части ПО: номер версии и контрольная сумма.

Кнопка «Записать»: запись в память контроллера настроек модуля СТ1АС108, изменённых пользователем. Кнопка доступна только когда настройки, отображаемые в окне, отличаются от настроек, записанных в памяти контроллера (это определяется сервисной программой контроллера КСА-02). При нажатии на кнопку «Записать» осуществляется запись в память контроллера изменённых настроек модуля СТ1АС108, после чего кнопка «Записать» отключается.

Кнопка «Закреть»: завершает работу окна настройки модуля. При этом изменённые настройки модуля не сохраняются в памяти контроллера, если предварительно не была произведена их запись.

5.7.2. Настройка модулей СТ1АС004

Окно настройки модуля СТ1АС004 имеет заголовок «Модуль СТ1АС004, слот ХХ», где ХХ — номер слота, в котором находится модуль СТ1АС004. Чтобы вызвать это окно на экран монитора, необходимо в окне «Конфигурация контроллера» нажать кнопку «Настройка» в строке с номером слота, содержащего данный модуль.

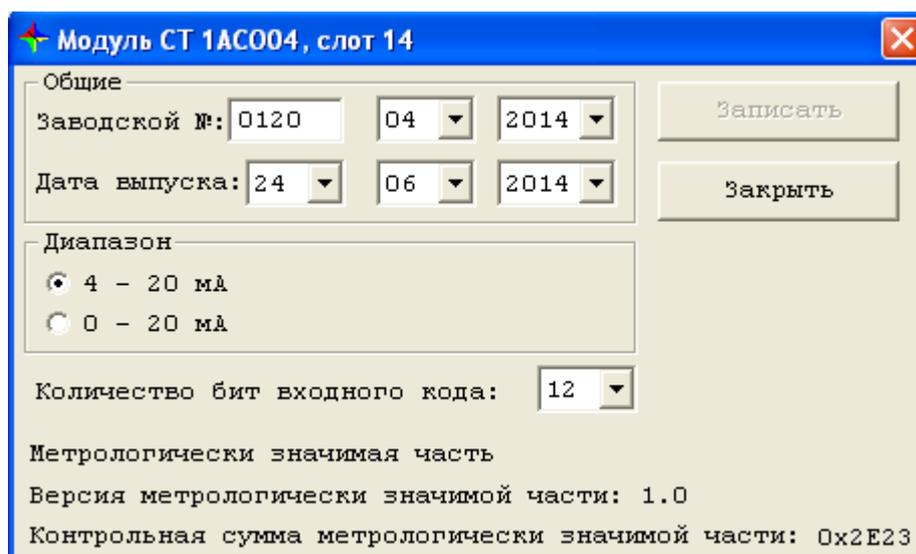


Рис. 5.7.2.1. Окно настройки модуля СТ1АС004.

В окне настройки модуля СТ1АС004 расположены кнопки «Записать» и «Закреть».

Кроме того, окно «Модуль СТ1АС004, слот XX» по параметрам настройки разделено на следующие группы:

- **Общие:** эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве;
- **Диапазон воспроизведения:** это переключатель, позволяющий определить, в каком диапазоне будет реализовываться выходной сигнал модуля. Есть два варианта: 4 — 20мА и 0 — 20мА.

Количество бит выходного кода: устанавливает линейное соответствие между входным измеренным значением аналогового сигнала и цифровым кодом, полученным в результате его преобразования. При этом 0 соответствует нижнему значению шкалы измерения, а для верхнего числа шкалы измерения имеется следующее соответствие:

Количество бит	Верхнее значение номинального цифрового кода
12	0x0FFF
13	0x1FFF
14	0x3FFF
15	0x7FFF
16	0xFFFF

Однако надо иметь в виду, что данная настройка не ограничивает допустимого диапазона цифрового кода; поэтому, возможно превышение верхнего значения номинального цифрового кода.

В нижней части окна располагается идентификационная информация о метрологически значимой части ПО: номер версии и контрольная сумма.

5.7.3. Настройка модуля CT1ARI08

Модуль CT1ARI08 преобразует входной аналоговый сигнал от датчика термосопротивления в цифровой код. Преобразование входного сигнала осуществляется на основании комбинации таких параметров модуля, как тип входного сигнала, количество бит выходного кода, ограничение шкалы. Также на основании настройки модуля формируется значение инструментальной ошибки модуля (один из факторов его неисправности). Входной сигнал в соответствии с калибровочными коэффициентами канала преобразуется в температуру. На основании рассчитанной температуры в соответствии с функцией преобразования определяется выходной код модуля CT1ARI08.

Для настройки параметров модуля CT1ARI08, необходимых для её правильного функционирования, предназначено окно настройки «Модуль CT1ARI08, слот XX», где XX — номер слота, в котором находится модуль CT1ARI08. Чтобы вызвать это окно на экран монитора, необходимо в окне «Конфигурация контроллера» нажать кнопку «Настройка», расположенную в строке с номером слота, содержащего данный модуль.

Модуль СТ 1ARI08, слот 3

Общие

Заводской №: 0367 11 2014

Дата выпуска: 19 04 2018

Записать

Заккрыть

Термосопротивление

100П, K = 1.3910

100П, K = 1.3850

50П, K = 1.3910

50П, K = 1.3850

53М, K = 1.4280

53М, K = 1.4260

50М, K = 1.4280

50М K = 1.4260

Калибровка

Выбор канала: 1

Состояние: откалиброван

Калибровка

Ограничение шкалы

Ограничение снизу

Ограничение сверху

Инструментальная ошибка

снизу: -55.000 гр.С сверху: 155.000 гр.С

Количество бит выходного кода: 12

Время усреднения

канал №1: 1.2 с

канал №2: 1.2 с

канал №3: 1.2 с

канал №4: 1.2 с

канал №5: 1.2 с

канал №6: без усреднения

канал №7: 100 мс

канал №8: 300 мс

канал №8: 1.2 с

канал №8: 5 с

Метрологически значимая часть

Версия метрологически значимой части: 1.0

Контрольная сумма метрологически значимой части: 0x5282

Рис. 5.7.3.1. Окно настройки модуля СТARI08.

В окне настройки модуля СТ1ARI08 расположены кнопки «Записать» и «Заккрыть» (аналогично всем окнам настройки модулей).

Кроме того, окно настройки модуля СТ1ARI08 по параметрам настройки разделено на следующие группы:

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Термосопротивление: предназначена для выбора типа термосопротивления. Выбор типа относится ко всем каналам модуля одновременно и выбирается путём установки нужного флажка.

Калибровка: предназначена для калибровки каждого из восьми каналов модуля СТ1ARI08 и содержит следующие поля и кнопки:

- **«Выбор канала»:** при нажатии на кнопку  отображается список номеров каналов модуля СТ1ARI08 от 1 до 8. Из этого списка выбирается номер канала.
- **«Состояние»:** отображается состояние выбранного канала модуля СТ1ARI08, у него могут быть значения «откалиброван» и «не откалиброван».
- **«Калибровка»:** калибровка канала осуществляется по двум калибровочным значениям тока (для этого необходимо подать соответствующий ток с проверенного прибора на вход нужного модуля), которые устанавливаются в соответствующих полях окна «Калибровка канала», (открывается при нажатии на кнопку «Калибровка»).

Значение первого калибровочного тока должно быть меньше значения второго калибровочного тока.

Рекомендуется устанавливать значение первого калибровочного тока близко к нижней границе диапазона калибровочного тока, а значение второго калибровочного тока — близко к верхней границе диапазона.

Чтобы установить калибровочное значение, нужно в окне «Калибровка канала» ввести необходимое значение, а затем нажать на кнопку «Сохранить».

При установке неверного значения (несоответствие диапазону калибровочного тока) на экране появляется окно с предупреждающим сообщением.

После закрытия окна «Калибровка канала» в строке «Состояние» окна настройки модуля СТ1ARI08 появится сообщение о результате калибровки канала: «откалиброван» или «не откалиброван».

Если все действия пользователя были правильными, а в строке «Состояние» отображается сообщение «не откалиброван», то это означает, что модуль неисправен.

Количество бит выходного кода: настройка устанавливает соответствие между входным измеренным значением аналогового сигнала и цифровым кодом, полученным в результате его преобразования. При этом 0 соответствует нижнему значению предела измерения (-

55°C), а для верхнего числа шкалы измерения (+150°C или +400°C) имеется следующее соответствие:

Количество бит	Верхнее значение номинального цифрового кода
12	0x0FFF
13	0x1FFF
14	0x3FFF
15	0x7FFF
16	0xFFFF

Однако данная настройка не ограничивает допустимого диапазона цифрового кода, поэтому возможно превышение верхнего значения номинального цифрового кода.

Требуемое количество бит выходного кода выбирается в строке «Количество бит выходного кода» из списка при нажатии на кнопку .

Ограничение шкалы: флажок «Ограничение шкалы» определяет ограничение диапазона выходного цифрового кода его номинальным диапазоном. Возможны два варианта ограничений: «Ограничение снизу» и «Ограничение сверху».

Модуль СТ1ARI08 преобразует входные сигналы:

от термосопротивлений типа 50П (K = 1,3910 или K = 1,3850) или 100П (K = 1,3910 или K = 1,3850) в диапазоне от -50°C до +400°C;

от термосопротивлений типа 50М (K = 1,4280 или K=1.4260) или 100М (K = 1,4280 или K=1.4260) в диапазоне от -50°C до +150°C.

Если флажок «Ограничение снизу» установлен, то даже если входной сигнал будет соответствовать температуре более низкой, чем -50, то выходной код всё равно будет соответствовать -50°C.

Установка флажка «Ограничение сверху» означает, что выходной код не может превысить верхней границы, даже если входной сигнал превышает объявленный диапазон. Если флажок не установлен, а входной ток превышает объявленный диапазон, то входной код превысит верхнюю границу выходного кода (кроме случая 16-битного выходного кода).

Инструментальная ошибка: инструментальная ошибка определяется снизу и сверху. В случае, если величина входного сигнала меньше, чем значение инструментальной ошибки снизу или больше, чем значение инструментальной ошибки сверху, соответствующий канал модуля СТ1ARI08 считается неисправным.

Пользователь имеет возможность изменять значения инструментальной ошибки вручную.

Время усреднения: время усреднения выходного сигнала. Выбор времени усреднения возможен из ряда: без усреднения, 100 мс, 300 мс, 1.2 с, 5 с. За интервал усреднения выходные данные обновляются четыре раза. Если выбран режим "без усреднения" выходные данные обновляются один раз в 20 мс.

В нижней части окна располагается идентификационная информация о метрологически значимой части ПО: номер версии и контрольная сумма.

Кнопка «Записать»: предназначена для записи в память контроллера настроек модуля СТ1АТІ08, изменённых оператором, и доступна в том случае, если настройки, отображаемые в окне, отличаются от настроек, записанных в памяти контроллера (это определяется сервисной программой контроллера КСА-02). При нажатии на кнопку «Записать» осуществляется запись в память контроллера изменённых настроек модуля СТ1АТІ08, после чего кнопка «Записать» отключается.

Кнопка «Закреть»: завершает работу окна настройки модуля. При этом изменённые настройки модуля не сохраняются в памяти контроллера, если предварительно не была произведена их запись.

5.7.4. Настройка модуля СТ1АТІ08

Модуль СТ1АТІ08 предназначен для преобразования входного аналогового сигнала от термоэлектрических преобразователей (далее по тексту — «термопар») в выходной цифровой код по восьми каналам, а также для измерения температуры в месте своей установки (далее по тексту — «температура холодного спая»).

Настраиваемые параметры модуля (тип термопары, количество бит выходного кода, ограничение шкалы) и фактическая исправность канала измерения температуры холодного спая задают условия, на основании которых входной сигнал преобразуется в цифровой код. На основании настройки модуля также формируется значение инструментальной ошибки модуля (один из факторов его неисправности).

Входной сигнал в соответствии с нелинейной аппроксимационной характеристикой канала преобразуется в температуру, к которой затем добавляется температура холодного спая. На основании рассчитанной температуры в соответствии с функцией преобразования определяется выходной код модуля СТ1АТІ08. Для настройки параметров модуля СТ1АТІ08 существует окно настройки «Модуль СТ1АТІ08, слот XX», где XX — номер

слота, в котором находится модуль СТ1АТ108. Чтобы вызвать это окно, необходимо в окне «Конфигурация контроллера» нажать кнопку «Настройка», расположенную в строке с номером слота, содержащего данный модуль.

Модуль СТ 1АТ108, слот 6

Общие
Заводской №: 1023 07 2010
Дата выпуска: 11 07 2011
Записать
Закрыть

Термопара
 Хромель-алюмель (К)
 Железо-константан (J)
 Хромель-копель (L)
 Нихросил-нисил (N)

Калибровка
Выбор канала: 1
Состояние: не откалиброван
Калибровка

Ограничение шкалы
 Ограничение снизу
 Ограничение сверху

Диапазон измерения
от: -50 гр.С
до: 1050 гр.С

Инструментальная ошибка
снизу: -50 гр.С сверху: 800 гр.С

Количество бит выходного кода: 14

Метрологически значимая часть
Версия метрологически значимой части: 1.0
Контрольная сумма метрологически значимой части: 0x3018

Рис. 5.7.4.1. Окно настройки модуля СТ1АТ108.

Помимо общих, модуль имеет следующие группы настроек: термопара, калибровка, ограничение шкалы, диапазон измерения, инструментальная ошибка, количество бит.

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Термопара: предназначена для выбора типа используемых термопар. Выбор типа относится ко всем каналам модуля одновременно.

При записи измененного типа термопары автоматически устанавливаются новые коэффициенты аппроксимационной характеристики преобразования входного сигнала термопары в температуру. Также устанавливаются границы диапазона измерения и инструментальной ошибки.

Калибровка: предназначена для калибровки каждого канала модуля СТ1АТІ08 и содержит следующие поля и кнопки:

- **Выбор канала:** при нажатии на кнопку  отображается список номеров каналов модуля СТ1АТІ08 от 1 до 8 (соответствуют каналам измерения термопар), либо надпись «ХС» (соответствует каналу измерения температуры холодного спая). Из этого списка выбирается номер канала.
- **Состояние:** отображается состояние выбранного канала модуля СТ1АТІ08 и его значение: «откалиброван» и «не откалиброван».
- **Кнопка «Калибровка»:** осуществляется калибровка текущего выбранного канала; при этом на калибруемый канал должно быть подано нулевое напряжение (или его входы должны быть закорочены). Для канала измерения холодного спая при нажатии на кнопку открывается окно ввода калибровочных сопротивлений. Калибровка канала измерения холодного спая осуществляется аналогично калибровке канала в модуле СТ1АТІ08. Если калибровка канала unsuccessful, появится сообщение об ошибке; в случае успеха поле «Состояние» будет иметь значение «откалиброван».

Ограничение шкалы: флажок «Ограничение шкалы» определяет ограничение диапазона выходного цифрового кода его номинальным диапазоном. Возможны следующие типы ограничений: «Нет», «Только снизу», «Только сверху», «Снизу и сверху».

Установка флажка «Ограничение снизу» означает, что для входного сигнала, выходящего за границу диапазона снизу, выходной код будет иметь значение, соответствующее нижней границе диапазона измерения.

Начиная с версии 3.6 программного обеспечения модуля флаг "ограничение снизу" не поддерживается и не влияет на выполнение рабочей программы модуля.

Установка флажка «Ограничение сверху» означает, что для входного сигнала, выходящего за границу диапазона сверху, выходной код будет иметь значение, соответствующее верхней границе диапазона измерения. Флаг "ограничение сверху" не влияет на выполнение рабочей программы модуля в случае, если параметр "число бит выходного кода" равен 16.

Диапазон измерения: правильнее этот параметр назвать "диапазон показаний". Назначает значения выходного сигнала, соответствующие нижней и верхней границам выходного кода. Диапазон может быть изменён вручную. Для этого следует ввести необходимые значения в поля «от» и «до», обозначающие нижний и верхний пороги диапазона

измерения модуля. При изменении типа термопары диапазон измерения изменяется автоматически.

Важно! Не допускается устанавливать значения данного параметра уже, чем метрологический диапазон измерения, приведенный в технических условиях для данного типа термопары. Стандартная практика: нижняя граница назначается на 5° С меньше нижней границы метрологического диапазона измерения; верхняя граница назначается на 5° С больше верхней границы метрологического диапазона измерения.

Инструментальная ошибка: значения могут быть изменены вручную. Инструментальная ошибка определяется снизу и сверху. В случае, если величина входного сигнала меньше, чем значение инструментальной ошибки снизу или больше, чем значение инструментальной ошибки сверху, формируется признак неисправности соответствующего канала модуля СТ1АТІ08.

Количество бит выходного кода: настройка устанавливает соответствие между входным измеренным значением аналогового сигнала и цифровым кодом, полученным в результате его преобразования. При этом 0 соответствует нижнему значению шкалы измерения, а для верхнего числа шкалы измерения имеется следующее соответствие:

Количество бит	Верхнее значение номинального цифрового кода
12	0x0FFF
13	0x1FFF
14	0x3FFF
15	0x7FFF
16	0xFFFF

Однако надо иметь в виду, что данная настройка не ограничивает допустимого диапазона цифрового кода; поэтому, возможно превышение верхнего значения номинального цифрового кода.

В нижней части окна располагается идентификационная информация о метрологически значимой части ПО: номер версии и контрольная сумма.

5.7.5. Настройка модуля СТ1ВСТ02, СТ1ВСТ03

Модуль обрабатывает входные сигналы, передаёт их значения на внешнее устройство, а также выполняет команды внешних устройств с помощью выходных реле.

Заголовок окна настройки называется «Модуль СТ1ВСТ02, слот XX», где XX — номер слота, в котором находится модуль СТ1ВСТ02. Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Модуль СТ 1ВСТ02, слот 4

Общие

Заводской №: 0235 05 2008

Дата выпуска: 01 05 2008

Записать

Закрыть

Кран 1

Отслеживать таймеры Наличие возврата

Наличие режимов

Таймеры: Схода 1.0 сек Дожима 2.0 сек

Перестановки 60.0 сек Смазки 2.0 сек

Антидребезга 497.28 мсек

Кран 2

Отслеживать таймеры Наличие возврата

Наличие режимов

Таймеры: Схода 1.0 сек Дожима 2.0 сек

Перестановки 60.0 сек Смазки 2.0 сек

Антидребезга 497.28 мсек

Рис. 5.7.5.1. Окно настройки модуля СТ1ВСТ02.

В окне настройки модуля СТ1ВСТ02 расположены кнопки «Записать» и «Закрыть».

Кроме того, окно настройки модуля СТ1ВСТ02 по параметрам настройки разделено на следующие группы: «Общие», «Кран 1» и «Кран 2». Для окна настройки модуля СТ1ВСТ03 существует также группа «Кран 3».

Содержание и принципы работы с элементами в группах «Кран 1», «Кран 2» и «Кран 3» идентичны друг другу.

Группы:

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Кран 1:

- **«Отслеживать таймер»:** если флажок установлен, происходит отслеживание таймеров «Схода» и «Перестановки», и в случае превышения установленного времени включается аварийный режим. Если флажок снят, таймеры «Схода» и «Перестановки» не отслеживаются, авария не фиксируется.
- **«Наличие режимов»:** при установленном флажке модуль позволяет работать в двух режимах: «автоматический» и «дистанционный», появляется возможность разделения команд сменного инженера и алгоритма. При снятом флажке модуль работает в режиме «дистанционный».
- **«Наличие возврата»** (имеет смысл лишь при наличии режима): если флажок установлен, то при автоматическом режиме, если состояние крана самопроизвольно изменилось (например, появилась двойная сигнализация), необходимо вернуть кран в исходное положение. При снятом флажке модуль работает в режиме «дистанционный».
- Таймеры:
- **«Схода»** — (максимум 30 мин) — в течение времени, отведенного на сход с выключателя, кран должен перейти в промежуточное состояние, иначе алгоритм переходит на стандартную обработку аварии;
- **«Перестановки»** — (максимум 30 мин) — в течение времени, отведенного на перестановку между крайними состояниями («открыто» и «закрыто»), кран должен изменить своё состояние на противоположное, иначе включится аварийный режим;
- **«Дожима»** — (максимум 2 мин) — чтобы подавить явление отскока в состоянии «дожим» (когда в результате открытия или закрытия кран достигнет конечного положения) блокируется изменение состояния концевых выключателей на время «Дожима» и удерживается команда управления на соленоид;
- **«Смазки»** — (максимум 2 мин) — время удержания соленоида смазки;
- **«Антидребезга»** — (максимум 1,133 сек) — модуль постоянно контролирует дребезг концевых выключателей, анализирует изменение сигнала и передаёт отфильтрованное значение. Сигнал считается изменившимся, если изменение сохраняется дольше, чем на время, установленное таймером защиты от дребезга.

5.7.6. Настройка модулей СТ1DIO29, СТ2DIO29, СТ3DIO29

Модуль обеспечивает регистрацию, прием и обработку входных дискретных сигналов постоянного тока по 24 каналам. Модуль поддерживает функцию счета импульсов входного дискретного сигнала по 8 каналам.

Модуль обеспечивает выдачу дискретного сигнала постоянного тока по 5 каналам.

Заголовок окна настройки называется «Модуль СТ1DIO29, слот XX», где XX — номер слота, в котором находится модуль СТ1DIO29. Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Общие									
Заводской №:	0235	05 2008							
Дата выпуска:	01 05 2008								
Записать									
Закреть									
Таймеры антидребезга									
Канал 1	96.8 мс	Канал 9 96.8 мс							
Канал 2	96.8 мс	Канал 10 96.8 мс							
Канал 3	96.8 мс	Канал 11 96.8 мс							
Канал 4	96.8 мс	Канал 12 96.8 мс							
Канал 5	96.8 мс	Канал 13 96.8 мс							
Канал 6	96.8 мс	Канал 14 96.8 мс							
Канал 7	96.8 мс	Канал 15 96.8 мс							
Канал 8	96.8 мс	Канал 16 96.8 мс							
Канал 17	96.8 мс	Канал 18 96.8 мс							
Канал 19	96.8 мс	Канал 20 96.8 мс							
Канал 21	96.8 мс	Канал 22 96.8 мс							
Канал 23	96.8 мс	Канал 24 96.8 мс							
Каналы счетных входов									
№ канала счет. входа	1	2	3	4	5	6	7	8	Размер счетчика
№ канала ТС	17	18	19	20	21	22	23	24	1000
Продолжительность импульса, мс	40.1	40.1	40.1	40.1	40.1	40.1	40.1	40.1	
Полярность	отр.	пол.	пол.	пол.	пол.	пол.	пол.	пол.	
Каналы ТУ									
№ канала ТУ	1	2	3	4	5				
Удержание уровня, мс	25	25	25	25	25				
Состояние при включении питания	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ				

Рис. 5.7.6.1. Окно настройки модуля СТ1DIO29.

В окне настройки модуля СТ1DIO29 расположены кнопки «Записать» и «Закреть».

Кроме того, окно настройки модуля СТ1DIO29 по параметрам настройки разделено на следующие группы: «Общие», «Таймеры антидребезга», «Каналы счетных входов» и «Каналы ТУ».

Группы:

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Таймеры антидребезга: эта группа содержит значение таймера антидребезга для двадцати четырех входных дискретных каналов. Модуль контролирует дребезг, анализирует изменение сигнала и передает отфильтрованное значение. Сигнал считается изменившимся, если изменение сохраняется дольше, чем на время, установленное таймером защиты от дребезга.

Каналы счетных входов: эта область содержит настройки каналов счетных входов модуля.

№ канала ТС: Каждому из каналов счетных входов может быть поставлен в соответствие один из 24 каналов ТС: значение от 1 до 24. При значении меньше 1 или больше 24, считается, что каналу счетного входа не соответствует ни один из каналов ТС.

Продолжительность импульса: Модуль анализирует изменение сигнала и передает отфильтрованное значение. Импульс будет засчитан, если его длительность превышает величину «Продолжительность импульса».

Полярность: Настройка назначает полярность регистрируемого импульса: положительную или отрицательную. Импульсу положительной полярности соответствует перепад уровня входного дискретного сигнала "низкий - высокий - низкий". Импульсу отрицательной полярности соответствует перепад уровня входного дискретного сигнала "высокий - низкий - высокий".

Размер счетчика: Настройка «Размер счетчика» задает количество импульсов, при достижении которого счетчик обнуляется, и отсчет импульсов начинается заново.

Каналы ТУ: эта группа содержит настройки каналов ТУ.

Удержание уровня: Минимальное время удержания заданного уровня выходного сигнала. Если две последовательные команды на один выходной дискретный сигнал поступили с интервалом меньшим, чем указанное время, выполнение второй команды задерживается до истечения указанного интервала.

Состояние при включении питания: При включении питания контроллера обычно ТУ остаются в выключенном состоянии. При необходимости можно настроить автоматическое включение ТУ при включении питания.

5.7.7. Настройка модулей СТ3DDI30 – СТ10DDI30

Модуль обеспечивает регистрацию, прием и обработку входных дискретных сигналов постоянного тока по 30 каналам.

Модули типов СТ3DDI30, СТ5DDI30, СТ7DDI30, СТ9DDI30 также поддерживают функцию счета импульсов входного дискретного сигнала постоянного тока по 8 каналам.

5.7.7.1. Настройка модулей СТ3DDI30, СТ5DDI30, СТ7DDI30, СТ9DDI30

Заголовок окна настройки называется «Модуль СТ3DDI30 (СТ5DDI30, СТ7DDI30, СТ9DDI30), слот XX», где XX — номер слота, в котором находится модуль СТ3DDI30 (СТ5DDI30, СТ7DDI30, СТ9DDI30). Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Модуль СТ3DDI30, слот 12

Общие
 Заводской №: 0006 12 2007 Дата выпуска: 11 07 2010

Таймеры антидребезга

Канал 1	101.2	мс	Канал11	101.2	мс	Канал21	101.2	мс
Канал 2	101.2	мс	Канал12	101.2	мс	Канал22	101.2	мс
Канал 3	101.2	мс	Канал13	101.2	мс	Канал23	101.2	мс
Канал 4	101.2	мс	Канал14	101.2	мс	Канал24	101.2	мс
Канал 5	101.2	мс	Канал15	101.2	мс	Канал25	101.2	мс
Канал 6	101.2	мс	Канал16	101.2	мс	Канал26	101.2	мс
Канал 7	101.2	мс	Канал17	101.2	мс	Канал27	101.2	мс
Канал 8	101.2	мс	Канал18	101.2	мс	Канал28	101.2	мс
Канал 9	101.2	мс	Канал19	101.2	мс	Канал29	101.2	мс
Канал10	101.2	мс	Канал20	101.2	мс	Канал30	101.2	мс

Каналы счетных входов

№ канала счет. входа	1	2	3	4	5	6	7	8	Размер счетчика
№ канала ТС	1	2	3	7	0	0	0	0	65000
Продолжительность импульса, мс	39.9	39.9	39.9	39.9	39.9	39.9	39.9	39.9	
Полярность	пол.								

Метрологически значимая часть
 Версия метрологически значимой части: 1.0
 Контрольная сумма метрологически значимой части: 0x69F8

Записать
 Закреть

Рис. 5.7.7.1. Окно настройки модуля СТ3DDI30

В окне настройки модуля СТ3DDI30 расположены кнопки «Записать» и «Закреть».

Кроме того, окно настройки модуля СТЗDDI30 по параметрам настройки разделено на следующие группы: «Общие», «Таймеры антидребезга», «Каналы счетных входов».

Области:

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Таймеры антидребезга: эта группа содержит значение таймера антидребезга для каждого из тридцати каналов. Модуль контролирует дребезг, анализирует изменение сигнала и передает отфильтрованное значение. Сигнал считается изменившимся, если изменение сохраняется дольше, чем на время, установленное таймером защиты от дребезга.

Каналы счетных входов: эта группа содержит настройки каналов счетных входов модуля.

№ канала ТС: Каждому из каналов счетных входов может быть поставлен в соответствие один из тридцати каналов ТС: значение от 1 до 30. При значении меньше 1 или больше 30, считается, что каналу счетного входа не соответствует ни один из каналов ТС.

Продолжительность импульса: Модуль анализирует изменение сигнала и передает отфильтрованное значение. Импульс будет засчитан, если его длительность превышает величину «Продолжительность импульса».

Полярность: Настройка назначает полярность регистрируемого импульса: положительную или отрицательную. Импульсу положительной полярности соответствует перепад уровня входного дискретного сигнала "низкий - высокий - низкий". Импульсу отрицательной полярности соответствует перепад уровня входного дискретного сигнала "высокий - низкий - высокий".

Размер счетчика: Настройка «Размер счетчика» задает количество импульсов, при достижении которого счетчик обнуляется, и отсчет импульсов начинается заново.

В нижней части окна располагается идентификационная информация о метрологически значимой части ПО: номер версии и контрольная сумма.

5.7.7.2. Особенности настройки модулей CT4DDI30, CT6DDI30, CT8DDI30, CT10DDI30

Заголовок окна настройки называется «Модуль CT4DDI30 (CT6DDI30, CT8DDI30, CT10DDI30), слот XX», где XX — номер слота, в котором находится модуль CT4DDI30 (CT6DDI30, CT8DDI30, CT10DDI30). Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Кроме того, окно настройки модуля CT4DDI30 по параметрам настройки разделено на следующие группы: «Общие», «Таймеры антидребезга». Настройка этих групп осуществляется аналогично соответствующим настройкам для модуля Модуль CT3DDI30.

Таймеры антидребезга								
Канал 1	101.2	мс	Канал11	101.2	мс	Канал21	101.2	мс
Канал 2	101.2	мс	Канал12	101.2	мс	Канал22	101.2	мс
Канал 3	101.2	мс	Канал13	101.2	мс	Канал23	101.2	мс
Канал 4	88.0	мс	Канал14	101.2	мс	Канал24	101.2	мс
Канал 5	162.8	мс	Канал15	101.2	мс	Канал25	101.2	мс
Канал 6	101.2	мс	Канал16	101.2	мс	Канал26	101.2	мс
Канал 7	101.2	мс	Канал17	101.2	мс	Канал27	101.2	мс
Канал 8	101.2	мс	Канал18	101.2	мс	Канал28	101.2	мс
Канал 9	101.2	мс	Канал19	101.2	мс	Канал29	101.2	мс
Канал10	101.2	мс	Канал20	101.2	мс	Канал30	101.2	мс

Рис. 5.7.7.2. Окно настройки модуля CT4DDI30.

5.7.8. Настройка модулей CT2DDO30

Модуль обеспечивает выдачу дискретных сигналов постоянного тока по 30 каналам.

Заголовок окна настройки называется «Модуль CT2DDO30, слот XX», где XX — номер слота, в котором находится модуль CT2DDO30. Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Модуль СТ 2DDO30, слот 4

Общие

Заводской №: 1023 11 2013 Дата выпуска: 15 11 2013

Записать

Закреть

Удержание уровня

Канал 1	10	мс	Канал11	0	мс	Канал21	0	мс
Канал 2	0	мс	Канал12	0	мс	Канал22	0	мс
Канал 3	0	мс	Канал13	0	мс	Канал23	0	мс
Канал 4	0	мс	Канал14	0	мс	Канал24	0	мс
Канал 5	0	мс	Канал15	0	мс	Канал25	0	мс
Канал 6	0	мс	Канал16	0	мс	Канал26	0	мс
Канал 7	0	мс	Канал17	30	мс	Канал27	0	мс
Канал 8	0	мс	Канал18	30	мс	Канал28	0	мс
Канал 9	0	мс	Канал19	0	мс	Канал29	0	мс
Канал10	0	мс	Канал20	0	мс	Канал30	0	мс

Защита от неисправности ЦП

Состояние ТУ при неисправности:

Последнее полученное значение не изменяется

Выставляются заранее определенные значения

Интервал задержки: 1500 мс

Установленное состояние ТУ при включении или неисправности

<input type="checkbox"/> ТУ 1	<input type="checkbox"/> ТУ 6	<input type="checkbox"/> ТУ 11	<input type="checkbox"/> ТУ 16	<input type="checkbox"/> ТУ 21	<input type="checkbox"/> ТУ 26
<input type="checkbox"/> ТУ 2	<input type="checkbox"/> ТУ 7	<input type="checkbox"/> ТУ 12	<input checked="" type="checkbox"/> ТУ 17	<input type="checkbox"/> ТУ 22	<input type="checkbox"/> ТУ 27
<input type="checkbox"/> ТУ 3	<input type="checkbox"/> ТУ 8	<input type="checkbox"/> ТУ 13	<input checked="" type="checkbox"/> ТУ 18	<input type="checkbox"/> ТУ 23	<input type="checkbox"/> ТУ 28
<input type="checkbox"/> ТУ 4	<input type="checkbox"/> ТУ 9	<input type="checkbox"/> ТУ 14	<input type="checkbox"/> ТУ 19	<input type="checkbox"/> ТУ 24	<input type="checkbox"/> ТУ 29
<input type="checkbox"/> ТУ 5	<input type="checkbox"/> ТУ 10	<input type="checkbox"/> ТУ 15	<input type="checkbox"/> ТУ 20	<input type="checkbox"/> ТУ 25	<input type="checkbox"/> ТУ 30

Рис. 5.7.8.1. Окно настройки модуля СТ2DDO30.

В окне настройки модуля СТ2DDO30 расположены кнопки «Записать» и «Закреть».

Кроме того, окно настройки модуля СТ2DDO30 по параметрам настройки разделено на следующие группы: «Общие», «Удержание уровня», «Защита от неисправности ЦП», «Установленное состояние ТУ при включении или неисправности».

Группы:

Общие: эта область содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Удержание уровня: Минимальное время удержания заданного уровня выходного сигнала. Если две последовательные команды на один выходной дискретный сигнал поступили с интервалом меньшим, чем указанное время, выполнение второй команды задерживается до истечения указанного интервала.

Защита от неисправности ЦП. Под "неисправностью ЦП" понимается прекращение связи модуля ST2DDO30 с процессорным модулем на интервал больший, чем это указано в настройке "Интервал задержки". При наступлении такого события возможны следующие варианты действий модуля: ранее выставленные значения выходных дискретных сигналов не изменяются или выставляются значения выходных сигналов, определенные в группе "Установленные значения ТУ при включении или неисправности".

Группа настроек поддерживается начиная с версии 4.0 программного обеспечения модулей. Для модулей с программным обеспечением более ранних версий, уровень выходных дискретных сигналов в случае пропадания связи с процессорным модулем не изменяется.

Установленное состояние ТУ при включении или неисправности: Эта область содержит флажки, обозначающие уровень ТУ при включении питания контроллера или при неисправности процессорного модуля.

Начиная с версии 4.1 программного обеспечения модуля данная настройка влияет только на уровни выходного сигнала при неисправности процессорного модуля, а значение ТУ при включении питания всегда равно нулю (низкое).

5.7.9. Настройка модулей ST1DAI16

Модуль обеспечивает регистрацию, прием и обработку входных дискретных сигналов переменного тока.

Заголовок окна настройки называется «Модуль ST1DAI16, слот XX», где XX — номер слота, в котором находится модуль ST1DAI16. Окно открывается нажатием кнопки «Настройка», расположенной в окне «Конфигурация контроллера» в строке с номером слота, содержащего данный модуль.

Модуль СТ 1DAI16, слот 16					
Общие					
Заводской №:	0509	06	2013		
Дата выпуска:	04	07	2012		
Антидребезг					
Канал 1	8.8	мс	Канал 9	101.2	мс
Канал 2	13.2	мс	Канал10	101.2	мс
Канал 3	17.6	мс	Канал11	101.2	мс
Канал 4	22.0	мс	Канал12	101.2	мс
Канал 5	101.2	мс	Канал13	101.2	мс
Канал 6	101.2	мс	Канал14	101.2	мс
Канал 7	101.2	мс	Канал15	101.2	мс
Канал 8	101.2	мс	Канал16	101.2	мс

Рис. 5.7.9.1. Окно настройки модуля СТ1DAI16.

В окне настройки модуля СТ1DAI16 расположены кнопки «Записать» и «Закрыть».

Кроме того, окно настройки модуля СТ1DAI16 по параметрам настройки разделено на следующие группы: «Общие», «Антидребезг».

Группы:

Общие: эта группа содержит заводской номер модуля и отображает дату его выпуска. Эти параметры изменить нельзя, т.к. они записываются в модуль при его изготовлении на производстве.

Антидребезг: эта группа содержит значение таймера антидребезга для каждого из шестнадцати каналов. Модуль контролирует дребезг, анализирует изменение сигнала и передаёт отфильтрованное значение. Сигнал считается изменившимся, если изменение сохраняется дольше, чем на время, установленное таймером защиты от дребезга.

5.8. Настройка параметров контроллера КСА-02

5.8.1. Параметры контроллера КСА-02

5.8.1.1. Параметры главного процессора

Окно «Параметры главного процессора» содержит инструменты и команды, которые позволяют:

- просматривать и/или изменять параметры обмена данными для контроллера КСА-02, которые конфигурируются при создании/редактировании алгоритма;
- просматривать параметры настройки системы резервирования (адрес и размер области резервирования в буферах данных), просматривать текущее состояние системы резервирования и управлять системой резервирования, переключая режим главного процессора в значение «ручное» и устанавливая режим ведущего или ведомого.
- просматривать и изменять параметры системы удаленного ввода-вывода, которые конфигурируются при создании/редактировании алгоритма.
- просматривать текущие настройки коммуникационных модулей, которые конфигурируются при создании/редактировании алгоритма.

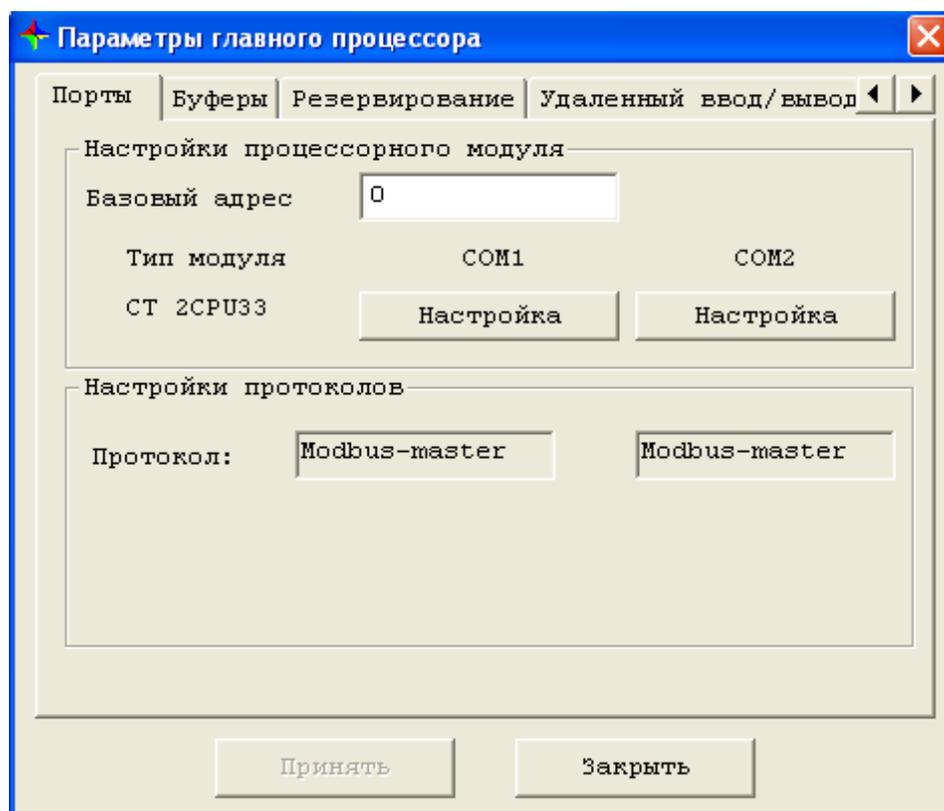


Рис. 5.8.1.1.1. Окно «Параметры главного процессора».

Изменение и просмотр параметров контроллера осуществляется по разделам, названия которых приведены на соответствующих вкладках окна «Параметры главного процессора».

Существует возможность изменения параметров контроллера из сервисной программы. Изменение возможно только для параметров, записанных в энергонезависимую память контроллера; параметры, заданные в алгоритме, необходимо задавать на этапе разработки алгоритма.

Существует возможность указать контроллеру использовать настройки из алгоритма или настройки, сохранённые в энергонезависимой памяти контроллера; там, где это возможно, установлены переключатели применения «Алгоритм» - «Контроллер».

В зависимости от положения переключателя применения «Алгоритм» или «Контроллер», контроллер будет использовать разные настройки: сохранённые в энергонезависимой памяти контроллера либо заданные в алгоритме.

Значения параметров, изменённые на вкладках окна «Параметры главного процессора» и принятые к изменению при нажатии на кнопку «Принять» (при наличии таковой), будут сохраняться в памяти контроллера только при нажатии на кнопку «Сохранить», расположенную в нижней части окна «Параметры главного процессора». Сохранять отдельно каждое изменение не обязательно. Это можно сделать один раз после установки всех необходимых изменений. Изменения вступают в силу только после перезапуска контроллера.

На вкладке "Порты" выделяются группы: "Настройки процессорного модуля" и "Настройки протоколов".

В группе "**Настройки процессорного модуля**" находятся поля:

Базовый адрес - настройка в текущей версии программного обеспечения не используется.

Тип модуля - информационное поле, в котором указан тип процессорного модуля.

Настройка - кнопка для вызова формы настройки параметров COM1 или COM2.

В группе "**Настройки протоколов**" находятся информационные поля с указанием типов установленных из алгоритма протоколов по последовательным портам COM1 и COM2.

5.8.1.2. Настройка параметров портов COM1 и COM2

Форма настройки параметров портов вызывается нажатием на кнопку **Настройка** на вкладке «Порты».

Форма содержит две группы: "Настройка протокола" и "Настройка COM-порта". В столбце «Алгоритм» отображаются параметры, заданные в редакторе параметров контроллера при разработке алгоритма; в столбце «Контроллер» отображаются и доступны для изменения параметры, сохранённые в энергонезависимой памяти контроллера.

Модуль ST 2CPU33, слот 2

Настройка протокола

	Алгоритм	Контроллер
Тип протокола	Modbus-master	Modbus-slave
Сетевой адрес	не установлен	12

Дополнительные настройки для сетевого адреса

Изменение адреса в режиме Ваккуп

Настройка COM-порта

	Алгоритм	Контроллер
Скорость	1200	9600
Четность	None (нет)	None (нет)
Время ответа	0.000 мс	2.496 мс
Число повторов при неответе	0	

Принять Отменить

Рис. 5.8.1.2.1. Настройка параметров портов

Настройка протокола

В группе содержатся тип протокола, сетевой адрес, а также дополнительные настройки сетевого адреса "из алгоритма" и "из контроллера".

В строке «Тип протокола» отображаются наименования протоколов (с указанием режима работы: Slave, Master), по которым будет производиться обмен через соответствующий порт.

В строке «Сетевой адрес» отображается адрес коммуникационного модуля в сети, в режиме Slave.

Для параметров "тип протокола", "сетевой адрес" имеется переключатель применения. Переключатель работает следующим образом: если переключатель установлен в положение "алгоритм" и алгоритм в контроллер загружен, то в рабочей программе используются параметры "из алгоритма". В других случаях (т.е. если переключатель установлен в положение "контроллер" или алгоритм не загружен) используются параметры "из контроллера".

Выбор применения дополнительных настроек для сетевого адреса осуществляется совместно с выбором сетевого адреса, т.е. если применяется сетевой адрес "из алгоритма", то дополнительные настройки для сетевого адреса также применяются "из алгоритма".

Замечание. Конфигурацию с разными типами протоколов "из алгоритма" и из "контроллера" при переключателе применения, установленном в положение "из контроллера" следует использовать с осторожностью, т.к. при такой конфигурации не все возможные пары протоколов совместимы.

Настройка COM порта

В группе содержатся настройки для скорости порта, типа бита четности, времени ответа, количества повторов пакета при отсутствии ответа. Параметры "из алгоритма" задаются в редакторе параметров при создании алгоритма и приводятся только для чтения; параметры "из контроллера" - настраиваемые.

Каждый параметр снабжен переключателем применения.

В строке «Скорость» отображается скорость обмена. Возможны следующие варианты 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

Отображение и изменение настройки «Контроль чётности» осуществляется с помощью ниспадающих списков под заголовком «Чётность». Возможны следующие варианты None (нет), Even (четный), Odd (нечетный).

В строке «Время ответа» отображается время начала ответа после получения последнего байта от Master-устройства. Только для Slave-протоколов.

Как уже было сказано ранее, для настройки доступны только параметры "из контроллера", а также состояние переключателей применения. Отметим, что настройка переключателей применения независима, т.е. совершенно корректно установить некоторые переключатели в положение "из алгоритма", а другие - в положение "из контроллера".

Чтобы внести произведённые изменения в память контроллера, сначала следует нажать кнопку «Принять» в нижней части окна настройки протокола, после чего это окно закроется. Затем нажать на кнопку «Принять», расположенную в основной части окна «Параметры контроллера».

При нажатии на кнопку «Отменить» в окне настройки протокола окно закрывается без сохранения произведённых изменений.

5.8.1.3. Параметры буферов контроллера КСА-02

Вкладка «Буферы» предназначена для просмотра размеров буферов данных, установленных в редакторе параметров контроллера при разработке алгоритма. Изменение настроек буферов данных средствами сервисной программы невозможно.

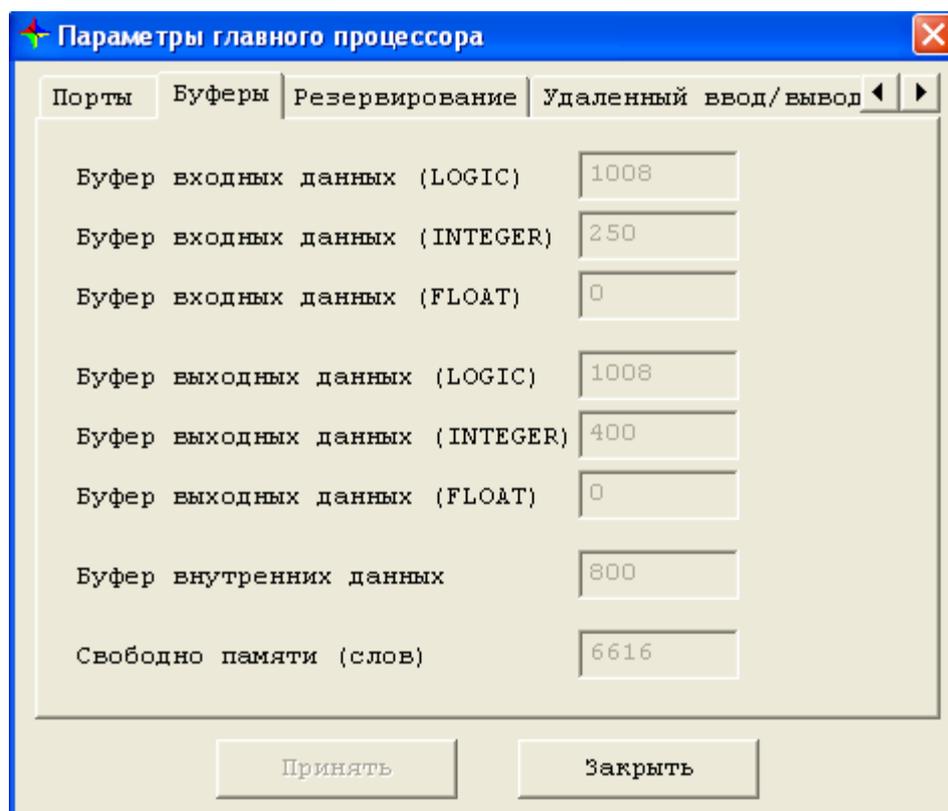


Рис. 5.8.1.3.1. Вкладка «Буферы».

Буфер входных данных (LOGIC)

Данное поле отображает текущее значение буфера входных данных типа LOGIC, размерность – бит. Значение данного буфера всегда кратно 16 (1 слово). Минимальное значение 0.

Буфер входных данных (INTEGER)

Данное поле отображает текущее значение буфера входных данных типа INTEGER, размерность – слово. Значение данного буфера всегда кратно 1 (1 слово). Минимальное значение 1.

Буфер входных данных (FLOAT)

Данное поле отображает текущее значение буфера входных данных типа FLOAT, размерность – 2 слова. Значение данного буфера всегда кратно 1 (2 слова). Минимальное значение 0.

Буфер выходных данных (LOGIC)

Данное поле отображает текущее значение буфера выходных данных типа LOGIC, размерность – бит. Значение данного буфера всегда кратно 16 (1 слово). Минимальное значение 0.

Буфер выходных данных (INTEGER)

Данное поле отображает текущее значение буфера выходных данных типа INTEGER, размерность – слово. Значение данного буфера всегда кратно 1 (1 слово). Минимальное значение 1.

Буфер выходных данных (FLOAT)

Данное поле отображает текущее значение буфера выходных данных типа FLOAT, размерность – 2 слова. Значение данного буфера всегда кратно 1 (2 слова). Минимальное значение 0.

Буфер внутренних данных

Данное поле отображает текущее значение буфера внутренних данных, размерность – 1 слово. Значение данного буфера всегда кратно 1 (1 слово). Минимальное значение 0.

Свободно памяти

Данное поле отображает текущее значение свободной памяти, размерность – 1 слово. Значение данного буфера всегда кратно 1 (1 слово).

Примечание. Общий участок памяти, выделяемый для использования под буферы данных, составляет 8192 слова, располагается в Data Memory контроллера. Для других целей данный участок памяти не используется.

5.8.1.4. Настройка параметров резервирования

Вкладка «Резервирование» позволяет просматривать параметры резервирования главного процессора контроллера КСА-02, сконфигурированные при разработке алгоритма. Значения всех полей устанавливаются только в редакторе параметров контроллера при разработке алгоритма.

Вкладка просмотра параметров резервирования выглядит следующим образом:

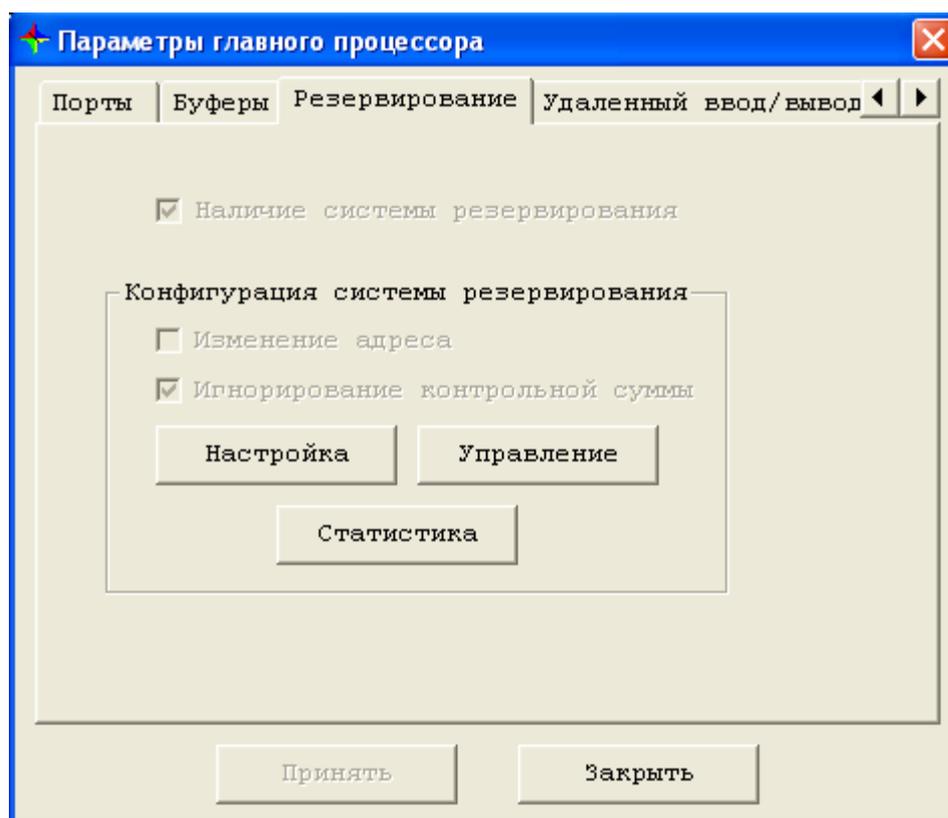


Рис. 5.8.1.4.1. Вкладка «Резервирование».

Наличие системы резервирования

Данное поле отображает наличие системы резервирования в контроллере КСА-02.

– нет системы резервирования;

– есть система резервирования.

Значение данного поля устанавливается в редакторе параметров контроллера при разработке алгоритма и средствами сервисной программы могут только просматриваться.

Конфигурация системы резервирования

Данная область содержит элементы, отображающие текущую конфигурацию системы резервирования.

Изменение адреса

– не изменяется базовый адрес;

– изменяется базовый адрес (см. пункт [5.8.1.2. Настройка параметров портов COM1 и COM2](#)) модуля, который находится в режиме ведомого – автоматически к адресу прибавляется 32.

Примечание. При изменении базового адреса необходимо следить, чтобы адрес не стал больше 255. При достижении этой величины, в системе резервирования могут наблюдаться ошибки.

Игнорирование контрольной суммы

– контрольная сумма не игнорируется;

– контрольная сумма игнорируется; система резервирования контроллера остается активной, даже если отличаются алгоритмы в главных процессорах контроллера.

При включении основной и резервный главный процессор осуществляют проверку алгоритмов на идентичность путём сравнения контрольной суммы. При идентичных алгоритмах главные процессоры образуют резервированную пару, в противном случае – нет. При выставлении галочки «игнорирование контрольной суммы» главные процессоры образуют резервированную пару и при неравенстве контрольных сумм.

Также вкладка содержит кнопки «Управление» и «Статистика», которых нет на соответствующей вкладке «Редактора параметров».

Кнопка «Настройка» открывает окно, позволяющее просматривать текущие настройки буферов для системы резервирования.

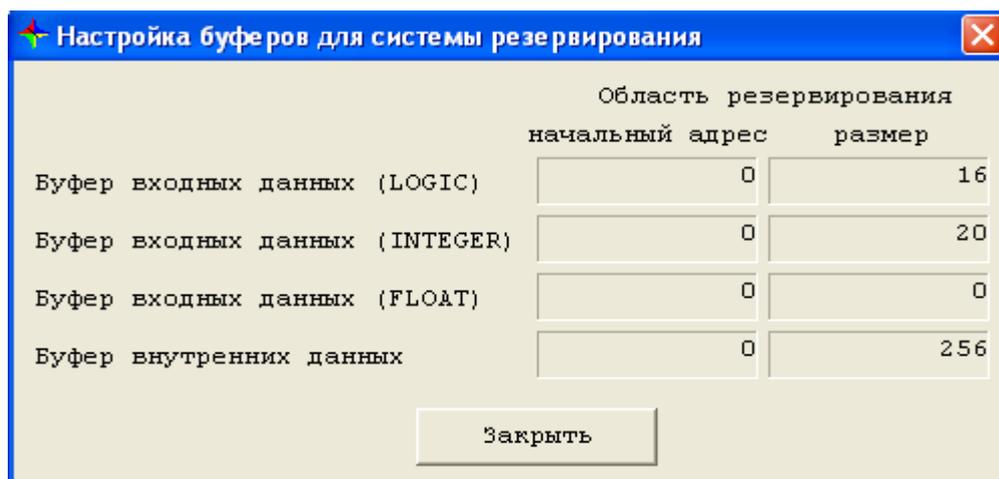


Рис. 5.8.1.4.2. Окно «Настройка буферов для системы резервирования».

Кнопка «Управление» открывает окно, позволяющее просматривать текущее состояние системы резервирования и управлять системой резервирования средствами сервисной программы.

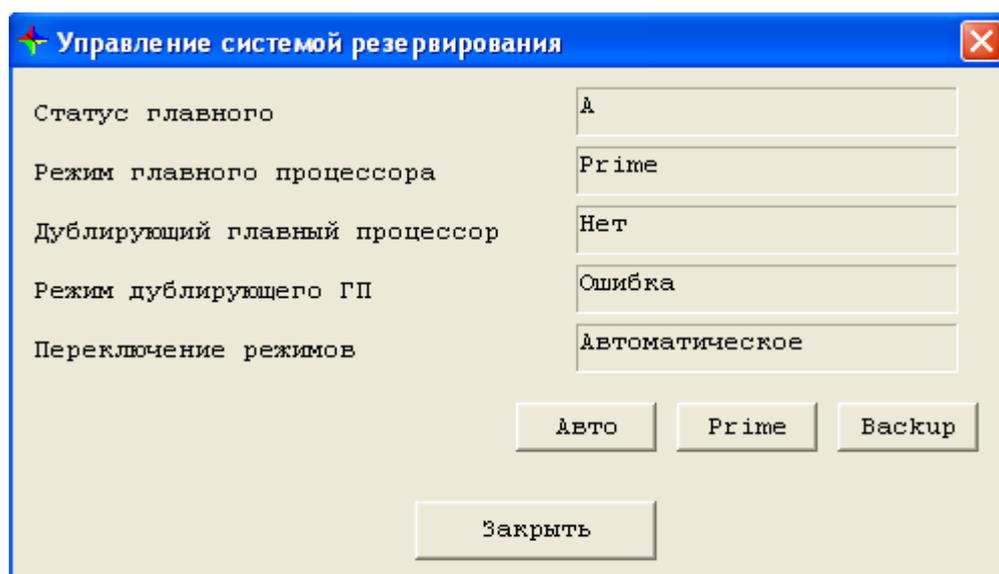


Рис. 5.8.1.4.3. Окно «Управление системой резервирования».

В окне «Управление системой резервирования» отображается текущее состояние следующих параметров:

- текущий статус главного процессора (А или В);
- текущий режим главного процессора (ведущий или ведомый);
- наличие/отсутствие дублирующего главного процессора;
- текущий режим дублирующего главного процессора (ведущий или ведомый);
- текущий способ переключения режимов («Автоматическое» или «Ручное»).

Кроме этого, там содержатся три кнопки: «Авто», «Prime» и «Backup».

Кнопка **«Авто»** изменяет значение способа переключения режимов главного процессора на «Автоматическое» (если он имел значение «Ручное»).

Кнопка **«Prime»** устанавливает ручной способ переключения режимов главного процессора и переводит контроллер в режим ведущего.

Кнопка **«Backup»** устанавливает ручной способ переключения режимов главного процессора и переводит контроллер в режим ведомого.

Примечание. Изменение режима главного процессора возможно только при ручном переключении режимов.

Кнопка **«Статистика»** открывает окно «Статистика по связи системы HotStandby», позволяющее просматривать статистику системы резервирования.

5.8.1.5. Настройка параметров удаленного ввода-вывода

Вкладка «Удалённый ввод/вывод» предназначена для просмотра и изменения параметров системы удалённого ввода-вывода. В процессе работы контроллер может использовать либо настройки, заданные в алгоритме (средствами редактора параметров), либо настройки, хранящиеся в энергонезависимой памяти самого контроллера. Выбор используемых настроек осуществляется переключателем применения "Алгоритм - Контроллер".

Для редактирования доступны только параметры "из контроллера", а также состояние переключателя применения.

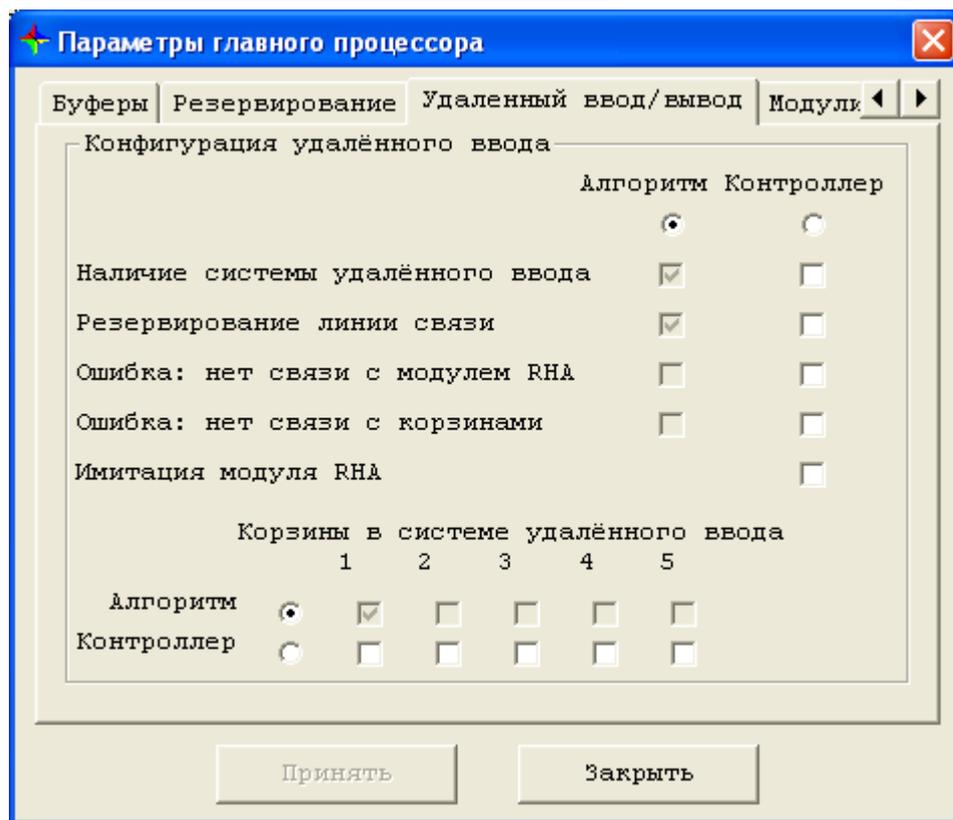


Рис. 5.8.1.5. Вкладка «Удаленный ввод-вывод».

В зависимости от положения переключателя применения контроллер будет использовать разные настройки:

Алгоритм <input type="checkbox"/>	Используются настройки, сохраненные в энергонезависимой памяти контроллера
Контроллер <input checked="" type="checkbox"/>	
Алгоритм <input checked="" type="checkbox"/>	Используется настройки, заданные в алгоритме
Контроллер <input type="checkbox"/>	

Выбор индикации осуществляется однократным нажатием с помощью мыши на соответствующую кнопку-указатель.

Пользователь может отключить систему удаленного ввода вывода целиком (изменив параметр «Наличие системы удаленного ввода/вывода») или отключить/подключить обмен с любой из СУВД (изменив параметр «1» – «5»). Аналогично пользователь может изменить условия остановки ядра при ошибках системы удаленного ввода/вывода, и наличия/отсутствия резервированной линии связи (изменив соответствующие параметры). Подробнее см. «Руководство на систему удаленного ввода/вывода».

Наличие системы удаленного ввода

Конфигурирование параметров работы системы удаленного ввода/вывода становится доступным при установке признака «Наличие системы удаленного ввода/вывода» щелчком мыши в соответствующем окне.

Вид и значение индикатора наличия системы удаленного ввода:

– есть система удаленного ввода;

– нет системы удаленного ввода.

Установка признака наличия осуществляется однократным нажатием с помощью мыши на индикатор, повторное нажатие на индикатор меняет его значение на обратное.

Конфигурация удаленного ввода

Данная область содержит элементы, отображающие текущую конфигурацию системы удаленного ввода.

Резервирование линии связи

Опция, отвечающая за наличие резервирования линии связи с корзинами сети СУВД.

Вид и значение индикатора наличия линии резервирования:

– есть резервирование линии связи с корзинами сети СУВД;

– нет резервирования линии связи с корзинами сети СУВД.

Установка признака наличия осуществляется однократным нажатием с помощью мыши на индикатор, повторное нажатие на индикатор меняет его значение на обратное.

Остановка ядра при отсутствии связи с СТ1RNA 33

Остановка ядра при отсутствии связи с удаленными корзинами

При неисправности по связи в сети СУВД программные средства системы «Scorpio» позволяют применять механизм остановки ядра. Возможность остановки ядра в главном процессоре контроллера КСА-02 при неисправностях по связи целесообразно использовать в системах с «горячим» резервированием главных процессоров. В сети СУВД с системой «горячего» резервирования главного процессора контроллера КСА-02 остановка ядра позволяет сохранить передачу информации со станций ввода-вывода. В сети СУВД без «горячего» резервирования ГП контроллера КСА-02 остановка ядра

является лишь альтернативным средством мониторинга неисправности по связи в сети СУВД.

Вид и значение индикатора остановки ядра при отсутствии связи:

- при отсутствии связи ядро будет остановлено;
- при отсутствии связи ядро будет продолжать работу.

Установка признака остановки ядра при отсутствии связи осуществляется однократным нажатием с помощью мыши на индикатор, повторное нажатие на индикатор меняет его значение на обратное.

Корзины в системе удаленного ввода

Для указания наличия подключенных корзин сети СУВД используются пять индикаторов под номерами 1, 2, 3, 4, 5.

Вид и значение индикатора подключения корзины сети СУВД:

- подключена;
- не подключена.

Установка признака подключения осуществляется однократным нажатием с помощью мыши на индикатор, повторное нажатие на индикатор меняет его значение на обратное.

Имитация модуля RHA

Флаг поддерживается только для настройки "из контроллера" и обычно применяется в целях отладки. При установленном флаге "Имитация модуля RHA" и положении переключателя применения в состоянии "Контроллер" работа контроллера осуществляется без физического наличия модуля ST1RHA33. Процесс обмена по шине для этого модуля имитируется процессорным модулем.

Замечание. На форме (рис.5.8.1.5) изображены два переключателя применения. Однако, эти переключатели зависимы: если один из переключателей находится в положении "алгоритм", то другой переключатель также будет находиться в положении "алгоритм". Соответствие положений переключателей поддерживается автоматически.

5.8.1.6. Модули ST1CPN

Вкладка «Модули ST1CPN» окна «Параметры главного процессора» предназначена для просмотра настроек интерфейсных модулей ST1CPM10, ST1CPE10.

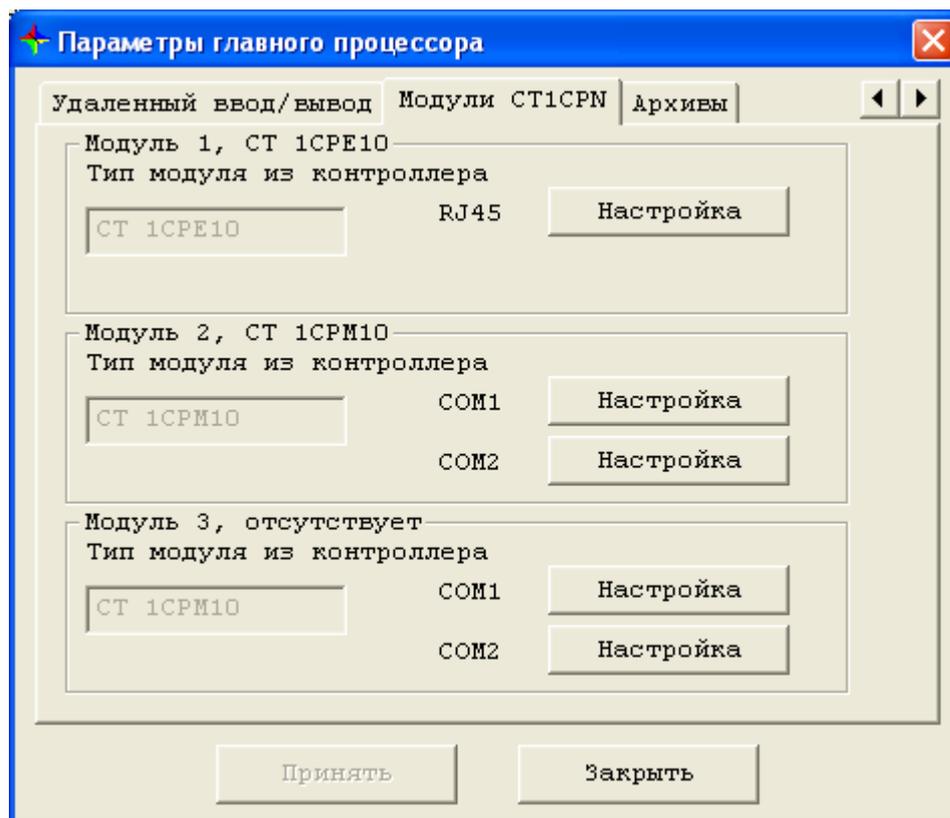


Рис. 5.8.1.6.1. Вкладка «Модули ST1CPN».

В форме выделяются группы "Модуль1", "Модуль 2", "Модуль 3".

Модуль 1,...

В заголовке группы указывается тип первого интерфейсного модуля, установленного в редакторе параметров контроллера при разработке алгоритма.

В информационном поле "Тип модуля из контроллера" указывается тип первого интерфейсного модуля, установленного в конфигурации главного процессора. Кнопка "Настройка" открывает форму настройки протокола порта интерфейсного модуля.

Модуль 2,...

В заголовке группы указывается тип второго интерфейсного модуля, установленного в редакторе параметров контроллера при разработке алгоритма.

В информационном поле "Тип модуля из контроллера" указывается тип второго интерфейсного модуля, установленного в конфигурации главного процессора. Кнопка "Настройка" открывает форму настройки протокола порта интерфейсного модуля.

Модуль 3...

В заголовке группы указывается тип третьего интерфейсного модуля, установленного в редакторе параметров контроллера при разработке алгоритма.

В информационном поле "Тип модуля из контроллера" указывается тип третьего интерфейсного модуля, установленного в конфигурации главного процессора. Кнопка "Настройка" открывает форму настройки протокола порта интерфейсного модуля.

Допустимы следующие ситуации:

- типы интерфейсных модулей из алгоритма и из контроллера на одинаковых позициях совпадают;
- типы интерфейсных модулей из контроллера избыточны по сравнению с типами интерфейсных модулей из алгоритма, например, на рис.5.8.1.6.1 на третьей позиции модуль из алгоритма не установлен, а в конфигурации на данную позицию назначен модуль СТ1СРМ10 .

В остальных случаях контроллер перейдет в ошибку.

Настройка

При нажатии на кнопку «Настройки», расположенную рядом с наименованием выбранного порта, открывается окно дополнительных настроек протокола.

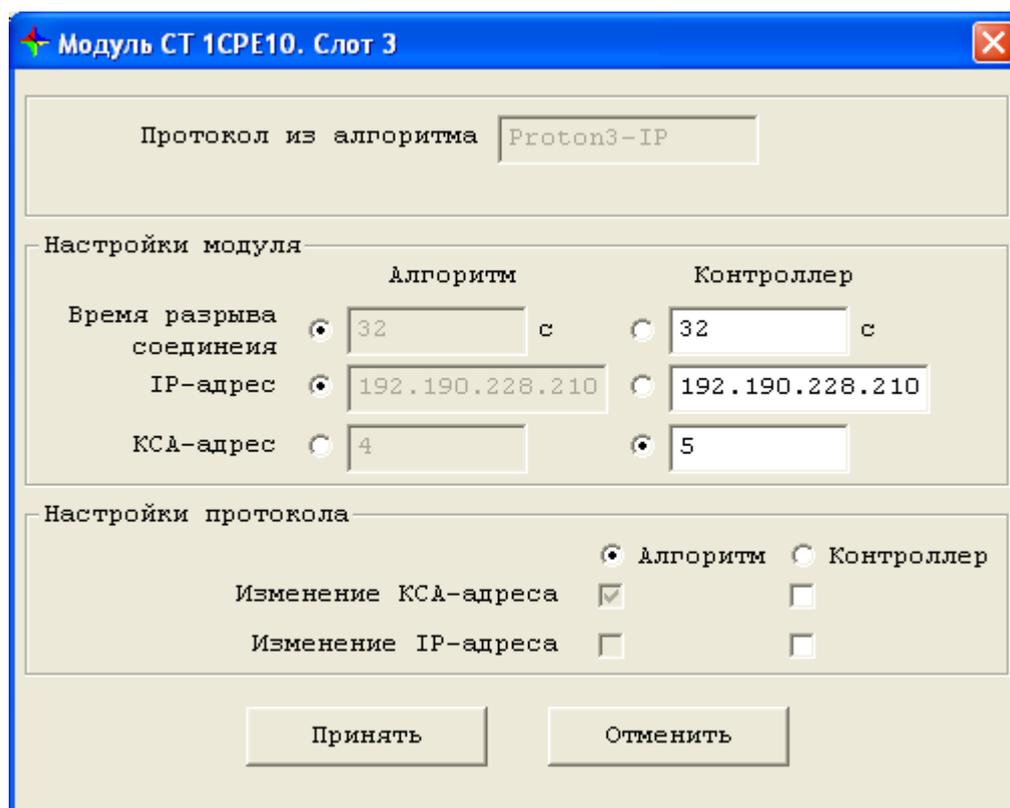


Рис. 5.8.1.6.2. Пример окна настройки модуля СТ1СРЕ10.

Окно настройки модуля СТ1СРЕ10 содержит три группы: "протокол из алгоритма", "настройки модуля" и "настройки протокола".

Протокол из алгоритма

Наименование протокола, заданного в алгоритме, по которому производится обмен данными.

Настройки модуля

Время разрыва соединения

До начала передачи данных на другое устройство, модуль должен узнать совокупность служебных параметров обмена, соответствующих устройству-получателю (адреса, порты, маркеры и проч.). Эти параметры образуют записи в специальных таблицах: таблица ARP, таблица TCP-соединений, и некоторых других. Записи имеют внутренний таймер. Обновление таймера записи происходит при приеме данных со служебными параметрами обмена, соответствующими данной записи. Если в течение интервала, равного времени разрыва соединения, корректного приёма таких данных не произошло, запись удаляется.

При наличии актуальной записи, служебные параметры обмена для передачи пакета берутся из этой записи. При отсутствии записи, а также в случае, когда запись была удалена, предварительно осуществляется передача пакетов для определения служебных параметров обмена.

Для Ethernet-систем время разрыва соединения обычно принимается равным 32 сек.

IP-адрес

Идентификатор устройства в сети, организованной с протоколом IP. Формат IP-адреса представляет собой 32-битовый числовой адрес, записанный с помощью четырех чисел от 0 до 255 каждое, разделенных точками (например: 192.168.17.189).

KCA-адрес

Сетевой адрес контроллера в соответствии с выбранным протоколом: для Modbus-TCP – Modbus адрес.

Каждый параметр из группы снабжен переключателем применения "алгоритм" - "контроллер".

Настройки протокола

Изменение IP-адреса

При установленном флаге осуществляется увеличение на 32 последнего октета IP-адреса для резервированного главного процессора в режиме ведомого. При использовании данного флага последний октет основного IP-адреса должен быть меньше или равен 222.

Изменение KCA-адреса

При установленном флаге осуществляется увеличение на 32 KCA-адреса для резервированного главного процессора в режиме ведомого. При использовании данного флага основной KCA-адрес должен быть меньше или равен 222.

Каждый параметр из группы снабжен переключателем применения "алгоритм" - "контроллер".

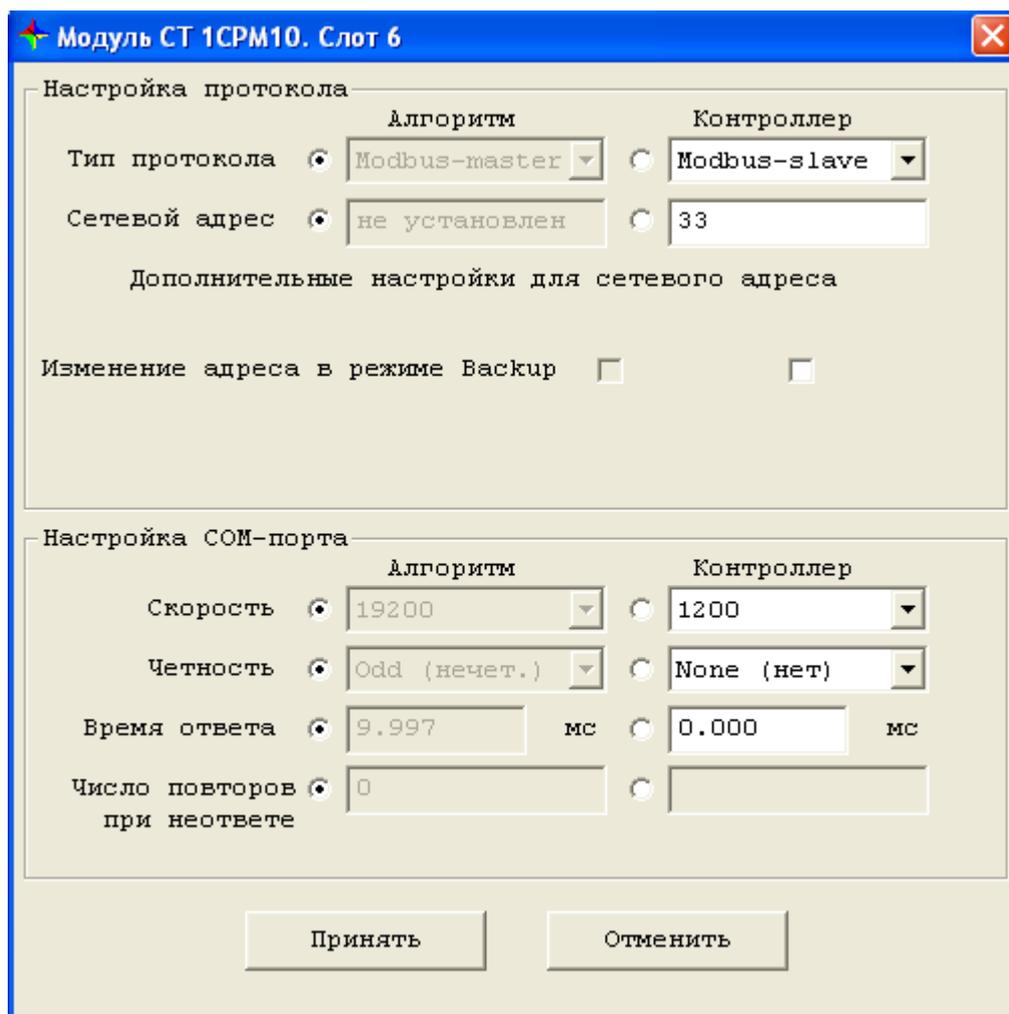


Рис. 5.8.1.6.3. Пример окна настройки модуля СТ1СРМ10.

Окно настройки модуля СТ1СРМ10 содержит группы: "настройка протокола", "настройка СОМ-порта".

Настройка протокола

Тип протокола

Наименование протокола, по которому производится обмен данными.

Сетевой адрес

Сетевой адрес контроллера в соответствии с выбранным протоколом (только для протоколов Slave).

Дополнительные настройки для сетевого адреса.

Параметры "тип протокола" и "сетевой адрес" снабжены каждый собственным переключателем применения. Дополнительные настройки сетевого адреса используют переключатель применения для сетевого адреса.

Замечание. Конфигурацию, в которой типы протоколов "из алгоритма" и "из контроллера" не совпадают, а переключатель применения установлен в положение "из контроллера" следует использовать с осторожностью. Существуют несовместимые пары протоколов, выбор которых может привести к непредсказуемым последствиям.

Настройка СОМ-порта

Скорость

Скорость обмена, возможны следующие варианты 1200, 2400, 4800, 9600, 19200, 38400.

Четность

Настройка четности, возможны следующие варианты None (нет), Even (четный), Odd (нечетный).

Время ответа

Только для Slave-протоколов. Время начала ответа после получения последнего байта от Master-устройства.

Число повторов при неответе.

Только для Master-протоколов. Число повторных запросов, в случае неполучения ответа. Если после установленного числа повторов ответ так и не получен, возвращается "результат обмена - ошибка".

5.8.1.7. Архивы

Система архивирования данных предназначена:

- для хранения исторических данных в энергонезависимой памяти контроллера;
- для формирования передачи этих данных на верхний уровень.

В конечном итоге, система архивирования данных позволяет расшифровать, какие события и когда происходили на контролируемом объекте. От момента наступления событий до их расшифровки может пройти достаточное время.

Данные не будут потеряны в следующих нештатных ситуациях:

- неисправность линии связи между контроллером и верхним уровнем в момент возникновения события;
- снятие напряжения с управляющего контроллера после возникновения события.

Вкладка «Архивы» окна «Параметры главного процессора» предназначена для просмотра состояния архивов.

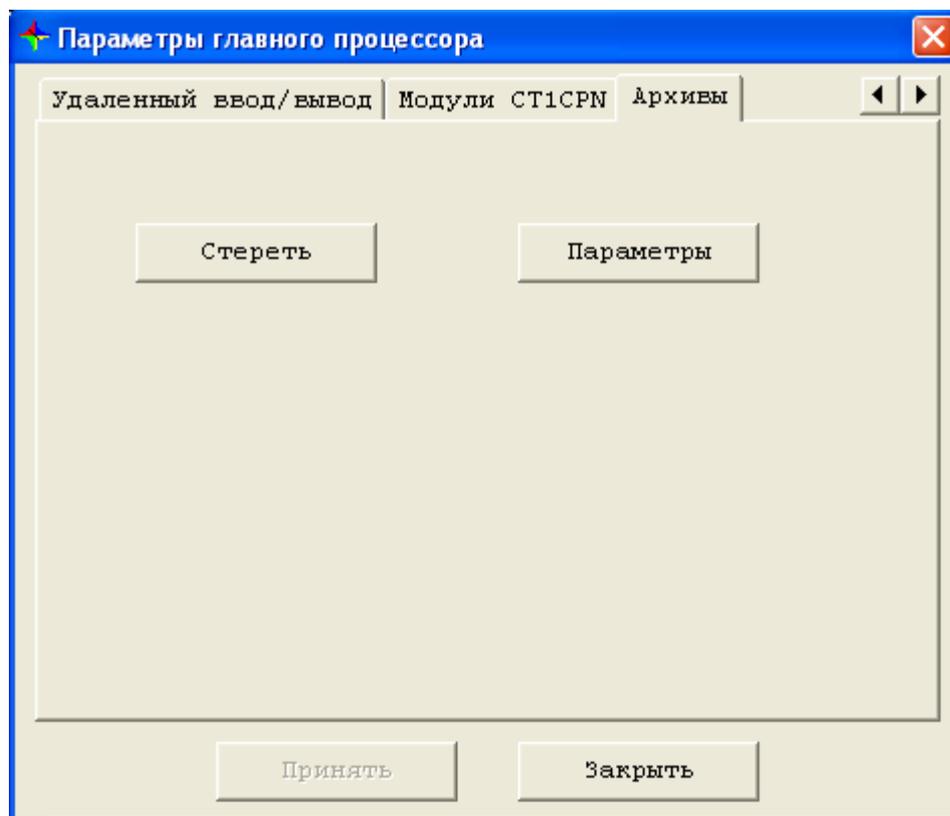


Рис. 5.8.1.7.1. Вкладка «Архивы».

На вкладке «Архивы» отображается:

- Кнопка «Стереть». При нажатии на кнопку архивы будут стерты;
- Кнопка «Параметры». При нажатии на кнопку открывается окно «Отображение состояния архивов», в котором отображается текущее состояние архивов.

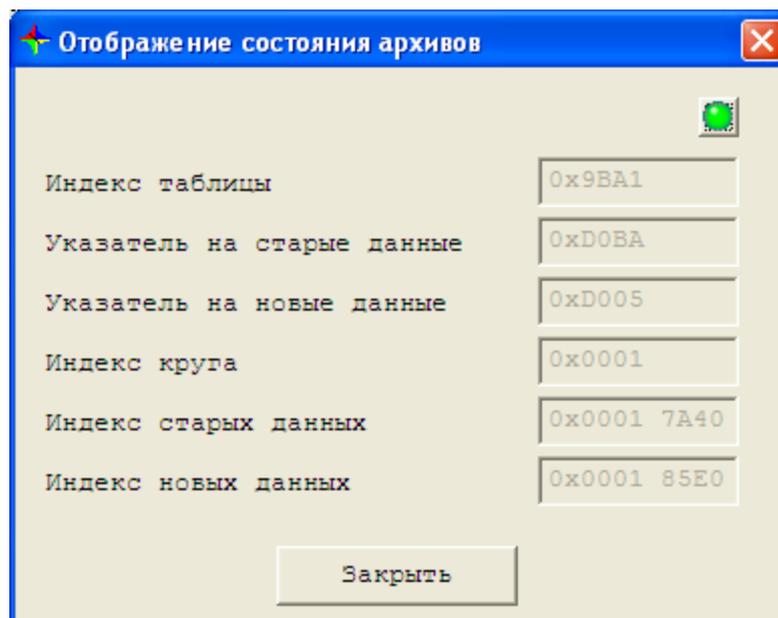


Рис. 5.8.1.7.2. Окно «Отображение состояния архивов».

Индекс таблицы

Индекс таблицы, а также контрольная сумма имеют чисто служебные функции. Индекс таблицы увеличивается на единицу каждый цикл алгоритма, в котором была запись данных в архив.

Указатель на старые данные

Указатель на начало архива (самые старые данные).

Указатель на новые данные

Указатель на точку записи новых данных. Указатель на конец архива (точка окончания записи последнего записанного блока) обычно равна (указатель на новые данные минус 1).

Индекс круга

Запись в архив осуществляется блоками данных. Каждый блок имеет свой порядковый номер, иначе называемый индексом. Уникальный индекс блока данных является важнейшей характеристикой блока данных и системы архивирования данных в целом. Индекс блока данных увеличивается на единицу при каждой записи в архив нового блока данных.

Использование индекса блока данных обеспечивает:

- упорядоченное расположение блоков данных в архиве;
- возможность произвольного доступа к блокам данных в архиве;

- ведение контроля прочитанной и непрочитанной архивной информации со стороны системы верхнего уровня.

При реальной записи в архив индекс записывается числом типа uint16. Однако этого типа данных недостаточно для уникальной идентификации блоков архивных данных, и реально индексы являются числами типа int32. Индекс круга используется при преобразовании записанного индекса типа uint16 к типу int32. Смотри ниже описание получения значения для индексов данных.

Индекс старых данных

Младшая часть индекса старых данных определяется чтением индекса блока данных, расположенного по указателю на старые данные. Старшая часть индекса равна индексу круга.

Индекс новых данных

Индекс новых данных – индекс, который будет присвоен новому блоку данных при помещении его в архив. После выполнения операции индекс нового блока данных увеличивается на единицу.

Индекс новых данных может быть рассчитан следующим образом. Считывается индекс последнего записанного блока данных (uint16) и увеличивается на единицу. Это будет младшая часть индекса новых данных. Если полученное число больше младшей части индекса старых данных (сравнение незнакомое), то старшая часть индекса новых данных равна индексу круга. В противном случае, она должна быть увеличена на единицу.

5.9. Тестирование контроллера КСА-02

Для контроллеров КСА-02 средствами сервисной программы можно провести различные виды проверок:

- **Тесты модулей ввода/вывода:** тестирование модулей ввода/вывода контроллера КСА-02 в соответствии с его конфигурацией;
- **Тест модуля СТ2CPU33** (п. [5.9.7. Тест модуля СТ2CPU33](#)):
 - **Тест процессора:** тестирование процессора контроллера.
 - **Тест ОЗУ:** тестирование памяти процессора модуля СТ2CPU33 (DM, PM).
 - **Тест SRAM/NVRAM:** тестирование памяти алгоритма и энергонезависимой памяти (SRAM и NVRAM).
 - **Тест Flash:** тестирование памяти Flash.

- **Тест COM1:** тестирование обмена по интерфейсу COM1.
- **Тест COM2:** тестирование обмена по интерфейсу COM2.
- **Самодиагностика:** диагностика состояния контроллера
- **Тест общей памяти:** тестирование модулей горячего резервирования контроллера (HSB) и модулей связи с контроллером удаленного ввода-вывода (RNA).
- **Тест Ethernet:** тестирование состояния связи по сети Ethernet путём пересылки ping-пакетов.
- **Тест COM-портов:** тестирование состояния связи с модулями контроллера по COM-портам.

Для каждого из тестов предусмотрено свое окно, работа с каждым из которых описана ниже (п. [5.9.1. Работа в режиме «Тесты модулей ввода/вывода»](#) – [5.9.6. Тест COM-портов](#)).

Работа некоторых тестов требует загрузки в оперативную память процессорного модуля специальной тестовой программы. Поэтому по завершении таких тестов следует перезапустить контроллер для загрузки рабочей программы ядра в оперативную память процессорного модуля (пункт меню «Управление», подпункт «Перезапуск контроллера»).

5.9.1. Работа в режиме «Тесты модулей ввода/вывода»

Режим «Тесты модулей ввода/вывода» доступен оператору только при наличии в контроллере ядра системы программирования и записанной в контроллер конфигурации. Индикация наличия ядра системы программирования осуществляется следующим образом:

- На лицевой панели процессорного модуля мигает индикатор «RUN», либо горит не мигая (если имеющееся ядро остановлено);
- На экране монитора в нижней информационной строке оболочки сервисной программы контроллера КСА-02 поле «Ядро» имеет зелёный или жёлтый цвет.

Для запуска тестов модулей ввода/вывода контроллера, следует в меню «Тесты» выбрать пункт «Тесты модулей ввода/вывода». При этом на экране монитора появится окно «Выбор модуля» (Рис. 5.9.1.1), которое отображает конфигурацию контроллера КСА-02 – номер слота и соответствующий ему тип модуля. Для выбора модуля для тестирования, необходимо в окне «Выбор модуля» нажать кнопку «Слот XX», где XX — номер слота. В зависимости от типа модуля, находящегося в выбранном для тестирования слоте, откроется окно для визуализации, контроля и управления тестированием соответствующего модуля.

Тип	
Слот 1	
Слот 2	СТ 1RPU33
Слот 3	СТ 1ARI08
Слот 4	СТ 1ACO04
Слот 5	СТ 1ACO04
Слот 6	СТ 1ARI08
Слот 7	СТ 1ATI08
Слот 8	СТ 2DDO30
Слот 9	СТ 1ACI08
Слот 10	СТ 2DDO30
Слот 11	СТ 4DDI30
Слот 12	СТ 2DDO30
Слот 13	СТ 7DDI30
Слот 14	СТ 1ACO04
Слот 15	СТ 8DDI30
Слот 16	СТ 1DAI16

Рис. 5.9.1.1. Окно «Выбор модуля»

5.9.2. Тест модуля СТ2CPU33

5.9.7.1. Общие сведения

Для запуска теста модуля СТ2CPU33 следует в меню «Тесты» выбрать пункт «Тест СТ2CPU33».

В рамках теста модуля СТ2CPU33 проводятся следующие виды тестирования:

- **Тест процессора:** тестирование процессора модуля СТ2CPU33 контроллера.
- **Тест ОЗУ:** тестирование памяти процессора модуля СТ2CPU33 (DM, PM).
- **Тест SRAM/NVRAM:** тестирование памяти алгоритма и энергонезависимой памяти (SRAM и NVRAM).
- **Тест Flash:** тестирование памяти Flash.
- **Тест COM1:** тестирование обмена по интерфейсу COM1.
- **Тест COM2:** тестирование обмена по интерфейсу COM2.

Замечание. Проведение теста SRAM / NVRAM приводит к удалению алгоритма, конфигурации и архивов контроллера. Проведение теста Flash приводит к удалению рабочей программы и заводских настроек контроллера.

Тест модуля ST2CPU33 осуществляются в окне «Тест модуля ST2CPU33» (Рис. 5.9.7.1.1, 5.9.7.1.2).

При запуске теста модуля ST2CPU33, а так же при старте автоматического теста модуля ST2CPU33, окно «Тест модуля ST2CPU33» отображается в компактном режиме.

В компактном режиме окно разделено на группы «Управление» и «Автоматический тест».

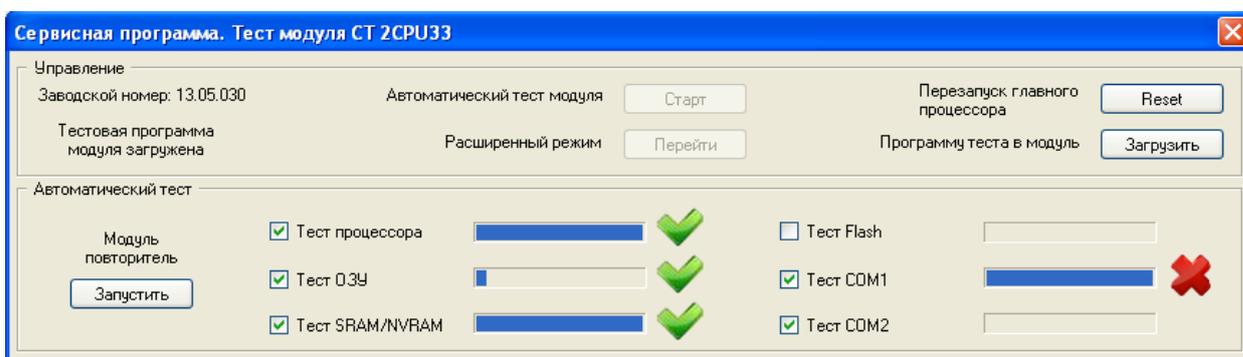


Рис. 5.9.2.1.1. Окно «Тест модуля ST2CPU33».

Для перехода в расширенный режим надо нажать на кнопку «Перейти» в группе «Управление».

В расширенном режиме окно разделено на следующие группы: «Управление», «Тест процессора», «Тест ОЗУ», «Тест SRAM/NVSRAM», «Тест Flash», «Тест COM1», «Тест COM2». В нижней части окна располагается поле, предназначенное для отображения текстовой информации об ошибках, возникающих в процессе проверок. Максимальное количество сообщений, отображаемых в этом поле, равно 50. Затем более ранние сообщения постепенно удаляются по мере поступления новой информации.

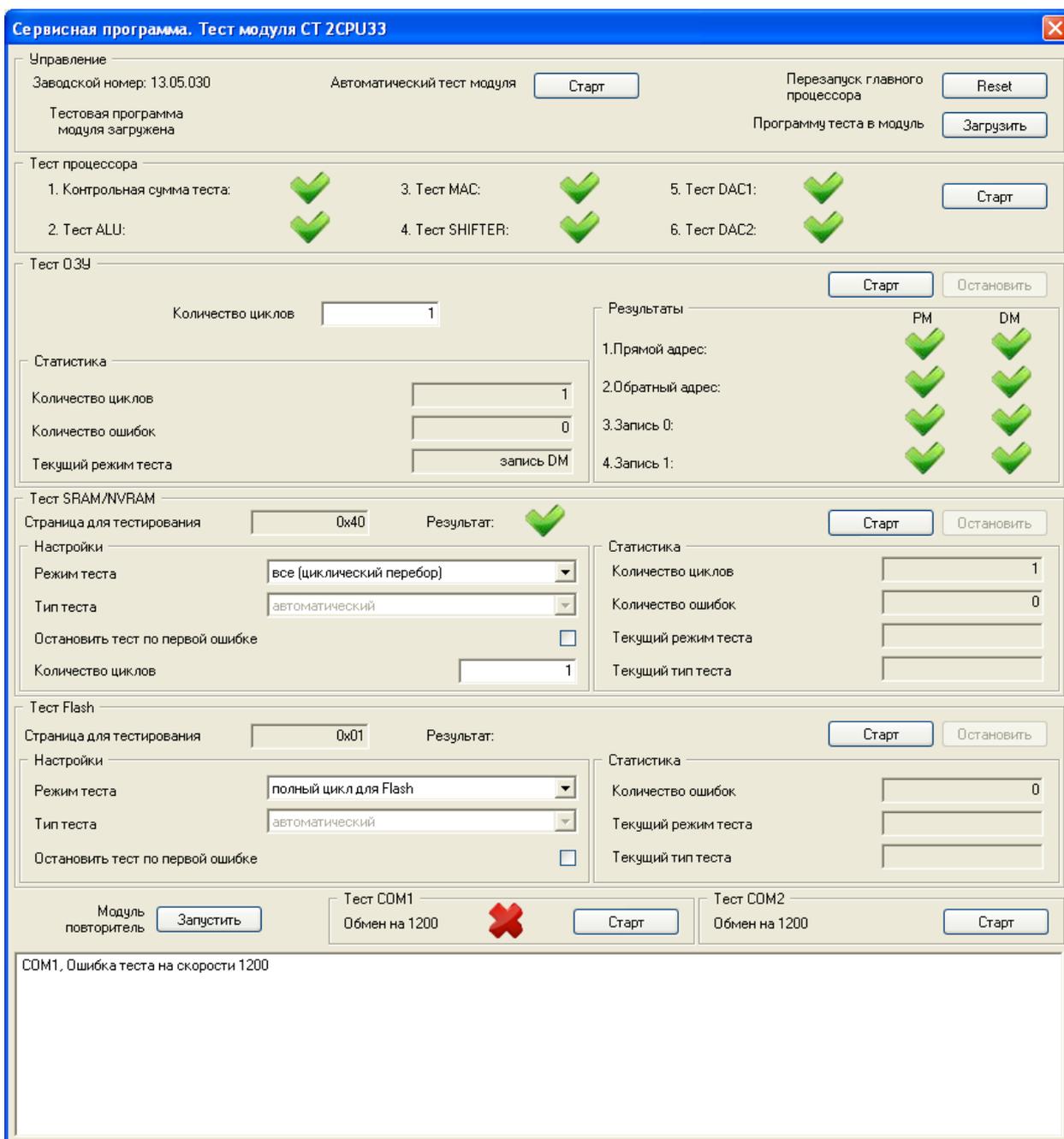


Рис. 5.9.2.1.2. Окно «Тест модуля СТ2CPU33». Расширенный режим.

5.9.2.2. Управление

В области «Управление» расположены:

- заводской номер контроллера;
- наличие тестовой программы в контроллере. При наличии программы в памяти контроллера отображается: «Тестовая программа модуля загружена»;
- кнопка «Старт». Служит для запуска автоматического тестирования СТ2CPU33;
- кнопка «Перейти». Доступна только в компактном режиме отображения окна «Тест модуля СТ2CPU33». Служит для перехода в расширенный режим;

- кнопка «Reset». Служит для перезапуска главного процессора;
- кнопка «Загрузить». Служит для ручного выбора файла программы теста для загрузки в модуль.

5.9.2.3. Автоматический тест модуля ST2CPU33

Общие сведения

При выполнении автоматического теста модуля ST2CPU33 поочередно выполняются отмеченные символом тесты. Выполняется один цикл каждого теста. Доступны следующие виды тестирования:

- тест процессора;
- тест ОЗУ;
- тест SRAM/NVRAM (Режим теста «Все (циклический перебор)», флаг «Остановить тест по первой ошибке» снят);
- тест Flash (Режим теста «Полный цикл для Flash», флаг «Остановить тест по первой ошибке» снят);
- тест COM1
- тест COM2.

Управление работой автоматического теста модуля ST2CPU33

Кнопка «Старт» в области «Управление» предназначена для запуска автоматического теста модуля ST2CPU33.

Существует возможность выбора видов тестирования, включенных в цикл автоматического теста модуля ST2CPU33.

Флаг напротив наименования каждого из видов тестирования обозначает:

– вид тестирования включен в цикл автоматического теста модуля ST2CPU33;

– вид тестирования не включен в цикл автоматического теста модуля ST2CPU33.

Для исключения вида тестирования из цикла автоматического теста необходимо снять флаг напротив его наименования.

По-умолчанию в автоматический тест включены все виды тестирования, кроме теста Flash.

Отображение информации о работе автоматического теста модуля ST2CPU33

Во время проведения каждого вида тестирования, напротив его наименования графически отображается процесс выполнения данного вида тестирования:

 – тестирование не выполнено;

 – тестирование выполняется;

 – тестирование выполнено.

Результат проведения тестирования графически отображается напротив наименования соответствующего тестирования:

 – тестирование пройдено успешно;

 – во время тестирования зафиксирована ошибка.

5.9.2.4. Тест процессора

Общие сведения

Тест процессора модуля СТ2CPU33 включает в себя следующие проверки:

- проверка контрольной суммы модуля;
- проверка работы ALU;
- проверка работы MAC;
- проверка SHIFTER;
- проверка DAC1 и DAC2.

Управление работой теста процессора

Кнопка «Старт» предназначена для запуска теста процессора модуля СТ2CPU33.

Отображение информации о работе теста процессора

Результат проведения проверки графически отображается напротив наименования соответствующей проверки:

 – проверка пройдена успешно;

 – во время проверки зафиксирована ошибка.

Поле в нижней части окна «Тест модуля СТ2CPU33» предназначено для отображения текстовой информации об ошибках, возникающих в процессе проверки.

Формат отображения текстового сообщения об ошибке

Формат сообщения об ошибке для теста процессора следующий:

«Процессор, <MESSAGE>»,

где <MESSAGE> – одно из следующих сообщений:

- ошибка контрольной суммы теста;
- ошибка теста ALU;
- ошибка теста MAC;
- ошибка теста SHIFTER;
- ошибка теста DAC1;
- ошибка теста DAC2.

5.9.2.5. Тест ОЗУ

Общие сведения

Тест ОЗУ модуля ST2CPU33 проводится с целью тестирования памяти управляющего процессора модуля. Тест представляет собой непрерывный повторяющийся цикл из четырёх различных видов проверки памяти, последовательно идущих друг за другом:

- **прямой адрес:** в каждую ячейку памяти осуществляется запись значения, равного младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **обратный адрес:** в каждую ячейку памяти осуществляется запись значения, равного инвертированным младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **запись 0:** в каждую ячейку памяти осуществляется запись значения 0x0000 с последующим чтением и проверкой;
- **запись 1:** в каждую ячейку памяти осуществляется запись значения 0xFFFF с последующим чтением и проверкой.

Управление работой теста ОЗУ

Для теста ОЗУ предусмотрены следующие элементы управления:

- **Поле «Количество циклов»** предназначено для задания количества циклов проверки ОЗУ.
- **Кнопка «Старт»** предназначена для запуска теста ОЗУ модуля ST2CPU33.

- **Кнопка «Остановить»** предназначена для остановки работы теста ОЗУ модуля ST2CPU33 во время его выполнения. После остановки тестирования кнопка «Остановить» отключается.

Отображение информации о работе теста ОЗУ

В области «Статистика» отображается статистическая информация о работе теста:

- в поле «Количество циклов» отображается количество выполненных циклов теста;
- в поле «Количество ошибок» отображается общее количество байт по всем выполненным циклам теста, в результате приёма которых была зафиксирована ошибка (переданное значение не совпало с принятым);
- в поле «Текущий режим теста» отображается текущий режим теста. Возможные значения: «чтение PM», «чтение DM», «запись PM», «запись DM».

В области «Результаты» графически отображаются результаты проведения проверок:



– проверка пройдена успешно;



– во время проверки зафиксирована ошибка.

Поле в нижней части окна «Тест модуля ST2CPU33» предназначено для отображения текстовой информации об ошибках, возникающих в процессе проверки.

Формат отображения текстового сообщения об ошибке

Формат сообщения об ошибке для теста ОЗУ следующий:

1) При проверке Program Memory управляющего процессора:

«ОЗУ, PM, Страница 0, Адрес <ADDRESS>, Записано <W_VALUE>, Прочитано <R_VALUE>», где

- <ADDRESS> — 16-битный адрес Program Memory управляющего процессора в 16-ричном виде (например, 0x0001);
- <W_VALUE>, <R_VALUE> — 32-битное значение ячейки Program Memory управляющего процессора в шестнадцатиричном виде (например, 0x000000FF);

2) При проверке Data Memory управляющего процессора:

«ОЗУ, DM, Страница 0, Адрес <ADDRESS>, Записано <W_VALUE>, Прочитано <R_VALUE>», где

- **<ADDRESS>** — 17-битный адрес Program Memory управляющего процессора в 16-ричном виде (например, 0x0001);
- **<W_VALUE>**, **<R_VALUE>** — 17-битное значение ячейки Program Memory управляющего процессора в 16-ричном виде (например, 0x00FF).

5.9.2.6. Тест SRAM/NVSRAM

Общие сведения

Тест SRAM/NVSRAM модуля CT2CPU33 проводится с целью тестирования внешней памяти (SRAM) и внешней энергонезависимой памяти (NVSRAM).

Тест представляет собой непрерывный повторяющийся цикл из четырёх различных видов проверки памяти, последовательно идущих друг за другом:

- **прямой адрес:** в каждую ячейку памяти осуществляется запись значения, равного младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **обратный адрес:** в каждую ячейку памяти осуществляется запись значения, равного инвертированным младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **запись 0:** в каждую ячейку памяти осуществляется запись значения 0x0000 с последующим чтением и проверкой;
- **запись 1:** в каждую ячейку памяти осуществляется запись значения 0xFFFF с последующим чтением и проверкой.

Управление работой теста SRAM/NVSRAM

Для теста SRAM/NVSRAM предусмотрены следующие элементы управления:

- **Кнопка «Старт»** предназначена для запуска теста SRAM/NVSRAM модуля CT2CPU33.
- **Кнопка «Остановить»** предназначена для остановки работы теста SRAM/NVSRAM модуля CT2CPU33 во время его выполнения. После остановки тестирования кнопка «Остановить» отключается.
- **Поле «Страница для тестирования»** доступна для каждого из режимов, кроме режима «Все (циклический перебор)». Предназначено для ввода страницы для тестирования. Для режима «Запись/чтение SRAM/NVSRAM (однократно)» диапазон доступных значений страницы от 0x40 до 0x4F, для режимов «Запись NVSRAM (однократно)» и «Чтение NVSRAM (однократно)» – от 0x80 до 0x8F.

В области «Настройки» расположены следующие инструменты управления тестом SRAM/NVSRAM:

- **Режим теста:**
 - все (циклический перебор);
 - запись/чтение SRAM/NVSRAM (однократно);
 - запись NVRAM (однократно);
 - чтение NVRAM (однократно).
- **Тип теста** (выбор типа теста доступен для каждого из режимов, кроме режима «Все (циклический перебор)»):
 - **тест прямых адресов:** в каждую ячейку памяти осуществляется запись значения, равного младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
 - **тест обратных адресов:** в каждую ячейку памяти осуществляется запись значения, равного инвертированным младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
 - **тест нулей (запись 0x0000):** в каждую ячейку памяти осуществляется запись значения 0x0000 с последующим чтением и проверкой;
 - **тест единиц (запись 0xFFFF):** в каждую ячейку памяти осуществляется запись значения 0xFFFF с последующим чтением и проверкой.
- **Флаг «Останавливать тест по первой ошибке».** Установка флага приводит к тому, что выполнение теста прекращается при возникновении первой ошибки, иначе выполнение теста прекращается нажатием кнопки «Остановить», либо при потере связи с тестируемым модулем.
- **Поле «Количество циклов»** предназначено для задания количества циклов проверки ОЗУ.

Отображение информации о работе теста SRAM/NVSRAM

В области «Статистика» отображается статистическая информация о работе теста:

- в поле «Количество циклов» отображается количество выполненных циклов теста;
- в поле «Количество ошибок» отображается общее количество байт по всем выполненным циклам теста, в результате приёма которых была зафиксирована ошибка (переданное значение не совпало с принятым);

- в поле «Текущий режим теста» отображается текущий режим теста. Возможные значения: «чтение DM», «запись DM»;
- в поле «Текущий тип теста» отображается текущий тип теста. Возможные значения: «прям. адресов», «обр. адресов», «нулей», «единиц».

В строке «Результат» графически отображается результат проведения проверки:



– проверка пройдена успешно;



– во время проверки зафиксирована ошибка.

Поле в нижней части окна «Тест модуля ST2CPU33» предназначено для отображения текстовой информации об ошибках, возникающих в процессе проверки.

Формат отображения текстового сообщения об ошибке

Формат сообщения об ошибке для теста SRAM/NVSRAM следующий:

1) При проверке памяти алгоритма:

«SRAM, Страница <STR>, Адрес <ADDRESS>, Записано <W_VALUE>, Прочитано <R_VALUE>», где

- <STR> — проверяемая страница;
- <ADDRESS> — 32-битный адрес памяти алгоритма в 16-ричном виде (например, 0x00000001);
- <W_VALUE>, <R_VALUE> — 17-битное значение ячейки памяти алгоритма в 16-ричном виде (например, 0x00FF);

2) При проверке энергонезависимой памяти:

«NVRAM, Страница <STR>, Адрес <ADDRESS>, Записано <W_VALUE>, Прочитано <R_VALUE>», где

- <STR> — проверяемая страница;
- <ADDRESS> — 32-битный адрес энергонезависимой памяти в 16-ричном виде (например, 0x00000001);
- <W_VALUE>, <R_VALUE> — 17-битное значение ячейки энергонезависимой памяти в 16-ричном виде (например, 0x00FF).

Замечание. Проведение теста NVSRAM приводит к уничтожению записанных в контроллер алгоритма, конфигурации, архивов.

5.9.2.7. Тест Flash

Общие сведения

Тест Flash модуля СТ2СРU33 проводится с целью тестирования Flash-памяти.

Управление работой теста Flash

Для теста Flash предусмотрены следующие элементы управления:

- **Кнопка «Старт»** предназначена для запуска теста Flash модуля СТ2СРU33.
- **Кнопка «Остановить»** предназначена для остановки работы теста Flash модуля СТ2СРU33 во время его выполнения. После остановки тестирования кнопка «Остановить» отключается.
- **Поле «Страница для тестирования»** доступна для каждого из режимов, кроме режима «Полный цикл для Flash». Предназначено для ввода страницы для тестирования. Диапазон доступных значений страницы от 0x01 до 0x20.

В области «Настройки» расположены следующие инструменты управления тестом Flash:

Режим теста:

- полный цикл для Flash;
- стирание Flash;
- запись Flash;
- чтение Flash.
- **Тип теста** (выбор типа теста доступен для каждого из режимов, кроме режима «Полный цикл для Flash»):
 - тест прямых адресов:** в каждую ячейку памяти осуществляется запись значения, равного младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
 - тест обратных адресов:** в каждую ячейку памяти осуществляется запись значения, равного инвертированным младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
 - тест нулей (запись 0x0000):** в каждую ячейку памяти осуществляется запись значения 0x0000 с последующим чтением и проверкой;
 - тест единиц (запись 0xFFFF):** в каждую ячейку памяти осуществляется запись значения 0xFFFF с последующим чтением и проверкой.

- **Флаг «Останавливать тест по первой ошибке».** Установка флага приводит к тому, что выполнение теста прекращается при возникновении первой ошибки, иначе выполнение теста прекращается нажатием кнопки «Остановить», либо при потере связи с тестируемым модулем.

Отображение информации о работе теста Flash

В области «Статистика» отображается статистическая информация о работе теста:

- в поле «Количество ошибок» отображается общее количество байт по всем выполненным циклам теста, в результате приёма которых была зафиксирована ошибка (переданное значение не совпало с принятым);
- в поле «Текущий режим теста» отображается текущий режим теста;
- в поле «Текущий тип теста» отображается текущий тип теста. Возможные значения: «прям. адресов», «обр. адресов», «нулей», «единиц».

В строке «Результат» графически отображается результат проведения проверки:



– проверка пройдена успешно;



– во время проверки зафиксирована ошибка.

Поле в нижней части окна «Тест модуля СТ2CPU33» предназначено для отображения текстовой информации об ошибках, возникающих в процессе проверки.

Формат отображения текстового сообщения об ошибке

Формат сообщения об ошибке для теста Flash следующий:

«Flash, Страница <STR>, Адрес <ADDRESS>, Записано <W_VALUE>, Прочитано <R_VALUE>», где

- <STR> — проверяемая страница;
- <ADDRESS> — 32-битный адрес ОЗУ в 16-ричном виде (например, 0x00000001);
- <W_VALUE>, <R_VALUE> — 17-битное значение ячейки Flash в 16-ричном виде (например, 0x00FF).

Замечание 1. Проведение теста Flash приводит к уничтожению записанных в контроллер рабочей программы и заводских настроек.

Замечание 2. Восстановление работоспособности модуля с уничтоженной рабочей программой может быть проведено следующим образом: запустить оболочку сервисной программы; перейти в расширенный режим настройки подключения (п.5.2); выбрать тип

контроллера "КСА-02 v2" и тип ядра "CPU"; в меню "Управление" выбрать пункт "Загрузить программу" (п. 5.4.3.4) и осуществить загрузку рабочей программы процессорного модуля. Рабочая программа модуля СТ2CPU33 обычно называется kernel.dxe.

5.9.2.8. Тест COM1 и COM2

Общие сведения

Тестирование COM1 (COM2) является результатом взаимодействия программы модуля контроллера, и дополнительной программы (программы повторителя).

Программа повторитель запускается на ПЭВМ, COM-порт которого соединен кабелем с тестируемым портом контроллера. Это может быть тот же компьютер, на котором запущена сервисная программа, если в его составе содержится более одного COM порта.

Кнопка "Запустить" Модуль повторитель предназначена запустить отдельно программу повторитель SPTestCOMRepeat на данном компьютере.

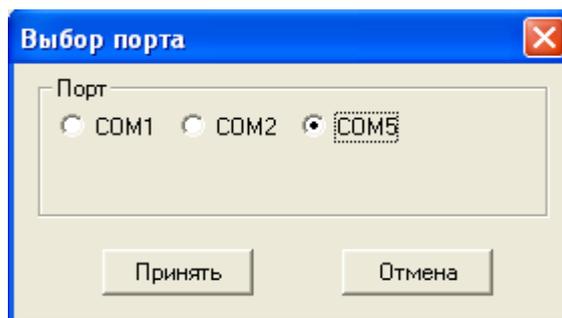


Рис. 5.9.2.8.1. Окно «Выбор порта»

При запуске программы повторителя открывается окно «Выбор порта» (Рис. 5.9.2.8.1), в котором отображается список доступных COM-портов компьютера. В окне необходимо выбрать порт компьютера, который соединен с тестируемым портом контроллера, и нажать на кнопку «Принять». Будет открыто окно «Программа повторитель» (Рис. 5.9.2.8.2).

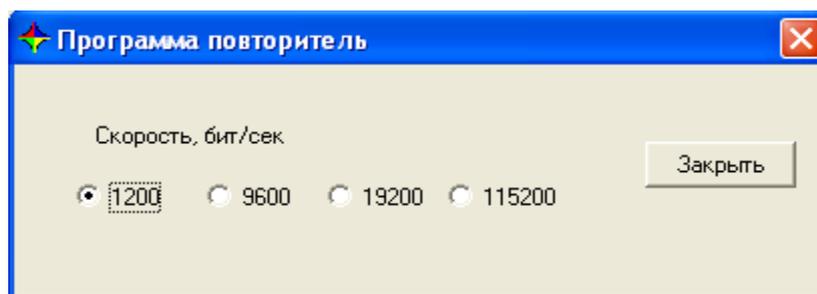


Рис. 5.9.2.8.2. Окно «Программа повторитель».

В окне «Программа повторитель» отображается список скоростей передачи данных: 1200, 9600, 19200, 115200 бит/сек.

Во время работы теста необходимо периодически устанавливать определенную скорость на повторителе. Для порта COM1 последовательно перебираются скорости 1200, 9600, 19200, 115200 бит/сек, для порта COM2 – скорости 1200, 9600, 19200 бит/сек.

Управление работой теста COM1 (COM2)

Кнопка «Старт» предназначена для запуска теста COM1 (COM2) модуля ST2CPU33.

Во время работы теста появляются сервисные сообщения с указанием установить определенную скорость на повторителе (Рис. 5.9.7.8.3). Для продолжения теста необходимо установить указанную скорость на повторителе и нажать на кнопку «Далее», для завершения теста – нажать на кнопку «Отмена».

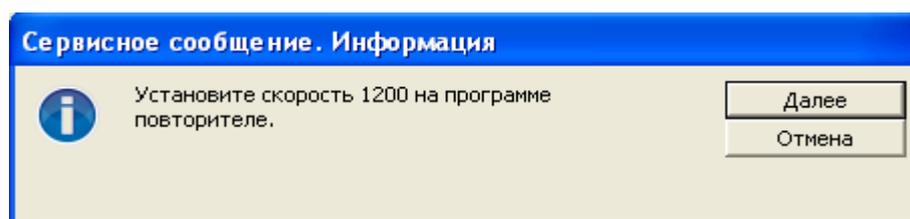


Рис. 5.9.2.8.3. Сервисное сообщение.

Отображение информации о работе теста COM1 (COM2)

Результат проведения проверки графически отображается напротив наименования соответствующей проверки:



– проверка пройдена успешно;



– во время проверки зафиксирована ошибка.

Поле в нижней части окна «Тест модуля ST2CPU33» предназначено для отображения текстовой информации об ошибках, возникающих в процессе проверки.

Формат отображения текстового сообщения об ошибке

Формат сообщения об ошибке для теста COM1 следующий:

«COM1, Ошибка теста на скорости <SPEED>»,

где <SPEED> – скорость, на которой возникла ошибка теста – может принимать значения: 1200, 9600, 19200, 115200.

Формат сообщения об ошибке для теста COM2 следующий:

«COM2, Ошибка теста на скорости <SPEED>»,

где <SPEED> – скорость, на которой возникла ошибка теста – может принимать значения: 1200, 9600, 19200.

5.9.3. Самодиагностика

Для запуска самодиагностики контроллера КСА-02 следует в меню «Тесты» выбрать пункт «Самодиагностика».

Визуализация самодиагностики контроллера осуществляются в окне «Самодиагностика» (Рис. 5.9.3.1).

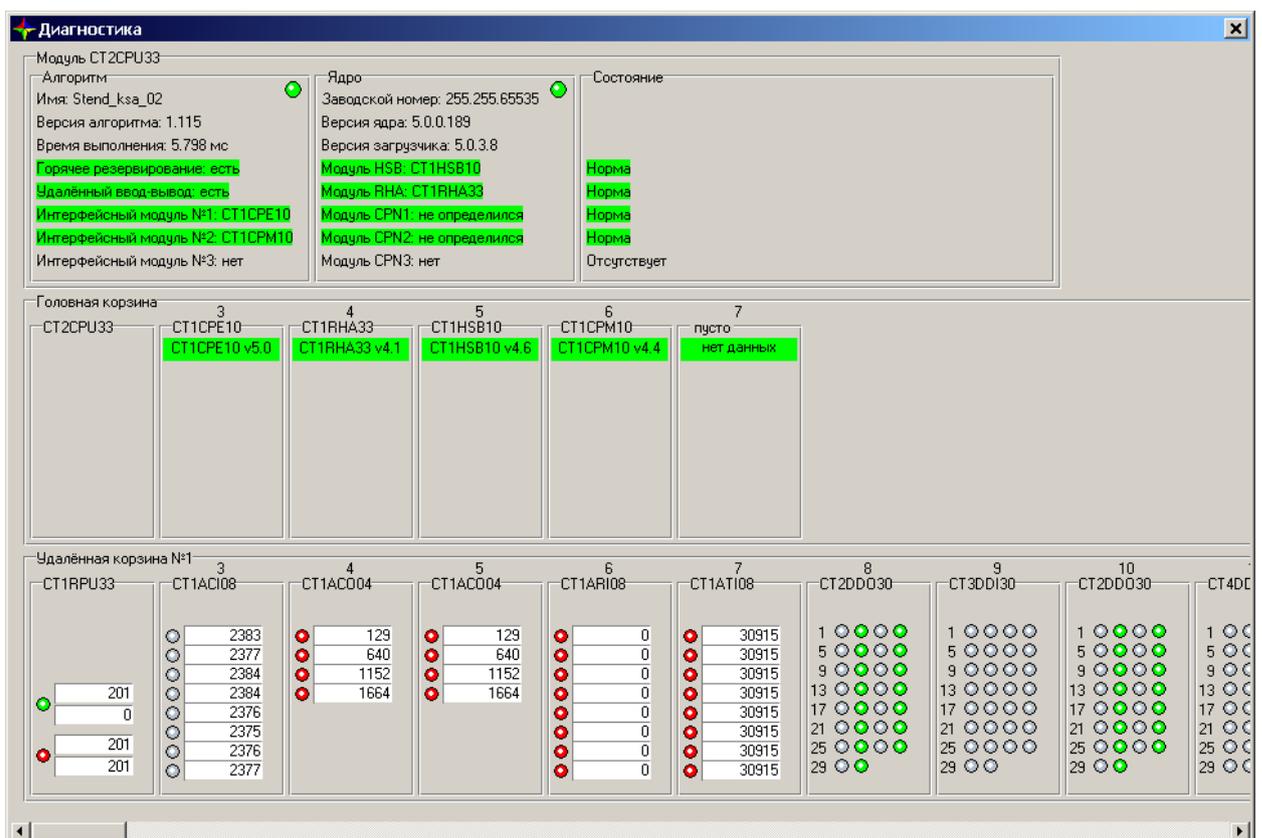


Рис.5.9.3.1. Окно «Диагностика».

В окне «Диагностика» отображается:

- общая информация о контроллере КСА-02;
- информация о корзине главного процессора;

- информация о корзинах СУВД.

Общая информация о контроллере отображается в группе «Модуль ST2CPU33». Группа «Модуль ST2CPU33» в свою очередь разделена на 3 подгруппы: «Алгоритм», «Ядро», «Состояние».

В подгруппе «**Алгоритм**» отображается информация об алгоритме:

- индикатор наличия алгоритма. Состояние индикатора обозначает наличие алгоритма:
 - (зеленый) – алгоритм есть;
 - (серый) – алгоритм отсутствует.

При отсутствии алгоритма другой информации об алгоритме не отображается. Для модуля RPU информации об алгоритме отсутствует.

- имя алгоритма;
- версия алгоритма;
- время выполнения алгоритма;
- наличие в алгоритме подсистемы горячего резервирования (есть или нет);
- наличие в алгоритме системы удаленного ввода-вывода (есть или нет);
- интерфейсный модуль №1 (№2, №3) (при наличии – тип модуля, при отсутствии – нет).

В подгруппе «**Ядро**» отображается информация о ядре:

- индикатор наличия ядра. Состояние индикатора обозначает наличие ядра:
 - (зеленый) – ядро есть;
 - (серый) – ядро отсутствует (не определено);
 - (красный) – ошибка ядра.

При отсутствии ядра другой информации о ядре не отображается.

- заводской номер;
- версия ядра;
- версия загрузчика;
- модуль HSB (при наличии – тип модуля, при отсутствии – нет);
- модуль RHA (при наличии – тип модуля, при отсутствии – нет);

- модуль CPN1 (CPN2, CPN3) (при наличии – тип модуля, при отсутствии – нет, если модуль не определен – не определился).

В подгруппе «Состояние» отображается диагностическая информация:

- описание аварии ядра при ее наличии
- напротив модулей HSB, RHA, отображается состояние модуля: «отсутствует» / «норма» / ошибка (описание ошибки соответствующей системы). Напротив каждого из модулей CPN отображается состояние модуля: «отсутствует» / «норма» / «ошибка». При состоянии «норма» вся строка окрашена в зеленый цвет, при состоянии «ошибка» – в красный.

Информация о корзине главного процессора отображается в группе «Головная корзина».

В группе «Головная корзина» схематично отображаются модули корзины главного процессора контроллера КСА-02.

В заголовке рамки каждого модуля отображается тип модуля по конфигурации. Ниже отображается тип модуля по автоконфигурации. Если в конфигурации присутствует недопустимый тип модуля, модуль отображается на красном фоне. Если тип модуля в конфигурации и в автоконфигурации соответствуют друг другу, то тип модуля по автоконфигурации отображается на зеленом фоне, иначе – на красном. Если в конфигурации стоит неизвестный тип модуля, или при невозможности расшифровать тип модуля, на месте типа модуля отображается слово «ошибка».

Модули ввода-вывода отображаются следующим образом:

- при отсутствии связи с модулем, модуль отображается красным;
- на дискретных модулях отображаются индикаторы для каждого из каналов модуля. Цвет индикатора обозначает значение соответствующего канала: ◐ (серый) – 0, ◑ (зеленый) – 1. Состояние индикатора совпадает с состоянием соответствующего индикатора, расположенного на лицевой панели модуля.
- На аналоговых модулях отображаются индикаторы для каждого из каналов модуля. Цвет индикатора обозначает наличие ошибки канала: ◑ (красный) – ошибка, ◐ (серый) – нет ошибки.

Информация о корзине СУВД №1 отображается в группе «Удаленная корзина №1».

В группе «Удаленная корзина №1» схематично отображаются модули корзины СУВД №1 контроллера КСА-02.

В заголовке рамки каждого модуля отображается тип модуля по конфигурации.

На изображении модуля СТ1RPU33 отображается статистика связи с корзиной СУВД: количество обменов и количество ошибок по СОМ1, количество обменов и количество ошибок по СОМ2. Рядом с данными полями располагаются индикаторы наличия связи с корзиной СУВД. Цвет индикатора свидетельствует о состоянии связи с удаленной корзиной:

- (серый) – нет связи с удаленной корзиной;
- (красный) – нет связи по соответствующему интерфейсу;
- (зеленый) – есть связь по соответствующему интерфейсу.

Отображение модулей ввода-вывода в корзине СУВД осуществляется аналогично отображению в корзине главного процессора.

Информация о корзинах СУВД №2 – №5 отображается аналогично информации о корзине СУВД №1.

Примечание. Если сервисная программа подключена к процессорному модулю корзины СУВД, окно «Диагностика» имеет следующие отличия:

- общая информация о контроллере отображается в группе «Модуль СТ1RPU33».
- группа «Модуль СТ1RPU33» разделена на 2 подгруппы: «Ядро», «Состояние». Подгруппа «Алгоритм» отсутствует;
- информация о системе горячего резервирования, удаленного ввода-вывода, интерфейсных модулях отсутствует;
- информация о корзине главного процессора отсутствует. Информация о подключенной корзине выводится в области «Удаленная корзина №Х», где Х – адрес корзины СУВД.

5.9.4. Тест общей памяти

5.9.4.1. Общие сведения

Коммуникационные модули типов СРЕ, СРМ, НSB, RHA содержат в себе DPM-память (dual port memory). Аппаратно обращения к этой памяти возможны как со стороны коммуникационного модуля, так и со стороны процессорного модуля. Тест общей памяти производится с целью тестирования DPM-памяти модулей. При этом, тестирование обращения к памяти со стороны процессорного модуля осуществляется с помощью

сервисной программы, а тестирование работы с памятью со стороны коммуникационного модуля осуществляется управляющей программой коммуникационного модуля во время его запуска и является частью программы самодиагностики модуля. Далее рассмотрим процесс тестирования памяти со стороны процессорного модуля.

Для запуска теста общей памяти, следует в меню «Тесты» выбрать пункт «Тест общей памяти». При этом сервисная программа контроллера автоматически загружает в процессорный модуль из базовой директории программу теста — TestSharedMem.dxe. Если программа теста общей памяти в базовой директории отсутствует, на экране монитора откроется окно, в котором необходимо выбрать файл TestSharedMem.dxe.

После загрузки программы теста общей памяти (автоматически или вручную) на экране монитора откроется новое окно «Сервисная программа. Тест общей памяти».

Окно содержит группы "Управление", "Тест Shared Memory", "Настройки", "Статистика" и поле для отображения текстовой информации о результатах работы теста.

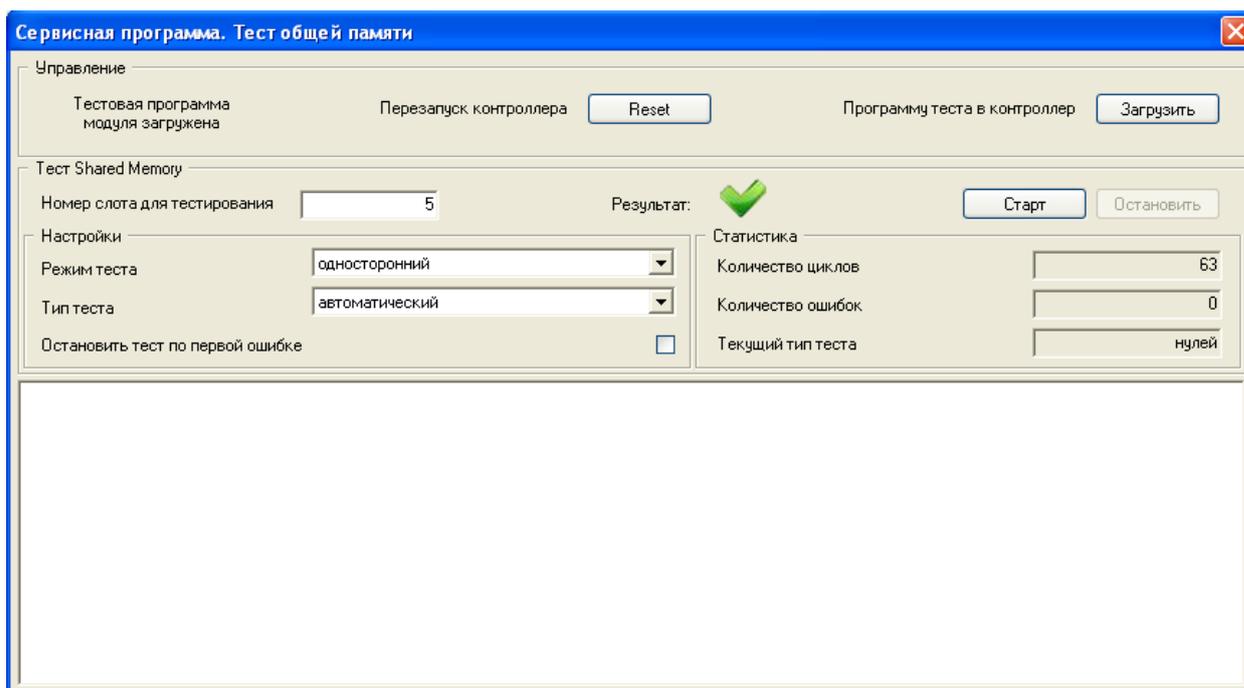


Рис. 5.9.4.1.1. Окно «Тест общей памяти».

До начала выполнения теста следует выбрать номер слота для тестирования (п.5.9.4.3), установить настройки (п.5.9.4.4). Для начала выполнения теста нажать на кнопку **Старт**.

5.9.4.2. Управление

Группа содержит индикатор загрузки программы, кнопки "Reset", "Загрузить".

Индикатор загрузки программы во время операции загрузки программы осуществляет графическое отображение процесса. Когда операция загрузки не осуществляется, индикатор представляет собой текстовое сообщение: "Тестовая программа модуля отсутствует", "Тестовая программа модуля загружена", "Связь с контроллером отсутствует".

Reset. Нажатие на кнопку осуществляет перезагрузку контроллера.

Загрузить. После нажатия на кнопку открывается окно для выбора тестовой программы, а затем осуществляется загрузка выбранной программы в процессорный модуль.

Замечание. При запуске теста из оболочки сервисной программы (п.5.9.4.1) загрузка тестовой программы в модуль осуществляется автоматически.

5.9.4.3. *Тест Shared Memory*

Группа содержит окно "Номер слота для тестирования", поле "Результат", кнопки "Старт" и "Остановить".

Номер слота для тестирования — задаётся номер слота в корзине, в который вставлен тестируемый модуль (от 3 до 16). По умолчанию значение ячейки 3.

Для изменения номера слота необходимо в поле «Номер слота для тестирования:» ввести с клавиатуры нужное значение и зафиксировать изменения нажатием клавиши «Enter».

Результат графически отображает итоги последнего выполненного теста:



– модуль соответствует требованиям, проверяемым данным тестом;



– модуль не соответствует требованиям, проверяемым данным тестом.

Результат отображается после остановки выполнения теста.

Старт. Кнопка активна при наличии загруженного, но в данный момент не выполняемого теста. Нажатие на кнопку начинает выполнение теста.

Остановить. Кнопка активна при наличии выполняемого теста. Нажатие на кнопку останавливает выполнение теста.

5.9.4.4. *Настройки*

Группа содержит переключатели "Режим теста", "Тип теста" и флаг "Останавливать тест при первой ошибке".

Режим теста. Для проведения тестирования необходимо выбрать «Односторонний тест памяти»:

- **односторонний тест памяти:** модуль ST2CPU33 записывает данные в DPM-память тестируемого модуля, а затем считывает их, сравнивает записанные данные со считанными и выводит результат.

Тип теста. Тип теста определяет значение каждого элемента в массиве записываемых данных. Поддерживается пять типов теста:

- **автоматический:** поочередное прохождение всех четырех тестов, описанных ниже;
- **тест прямых адресов:** в каждую ячейку памяти осуществляется запись значения, равного младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **тест обратных адресов:** в каждую ячейку памяти осуществляется запись значения, равного инвертированным младшим 16-ти битам адреса ячейки, с последующим чтением и проверкой;
- **тест нулей (запись 0x0000):** в каждую ячейку памяти осуществляется запись значения 0x0000 с последующим чтением и проверкой;
- **тест единиц (запись 0xFFFF):** в каждую ячейку памяти осуществляется запись значения 0xFFFF с последующим чтением и проверкой.

Для изменения режима или типа теста нужно при остановленном тесте нажать кнопку , расположенную около изменяемого поля, и выбрать из списка нужное значение. При этом на экране монитора отобразится список из возможных для данного поля значений.

Останавливать тест по первой ошибке. Установка флага приводит к тому, что выполнение теста прекращается при возникновении первой ошибки, иначе выполнение теста прекращается нажатием клавиши «Остановить», либо при потере связи с тестируемым модулем.

5.9.4.5. Статистика

Группа «Статистика» содержит поля "Количество циклов", "Количество ошибок", "Текущий тип теста".

- **количество циклов:** отображает количество выполненных циклов теста;
- **количество ошибок:** отображает общее количество ошибок по всем выполненным циклам теста;
- **текущий тест:** отображает тип выполняемого в данный момент теста: прямых адресов, обратных адресов, нулей, единиц.

5.9.4.6. *Отображение информации о работе теста общей памяти*

Поле, расположенное в нижней части окна теста, отображает текстовую информацию об ошибках, найденных в процессе проверки (если таковые есть).

Формат сообщения об ошибках из категории «ошибка чтения/записи» для теста общей памяти следующий:

«Ошибка. Адрес: <ADDRESS> Должно быть: <W_VALUE> Принято: <R_VALUE>», где

- <ADDRESS> — 16-битный адрес ячейки памяти тестируемого модуля в 16-ричном виде (например, 0x0001);
- <W_VALUE>, <R_VALUE> — 16-битное значение ячейки памяти тестируемого модуля в 16-ричном виде (например, 0xFFFF);

5.9.5. Тест Ethernet

5.9.5.1. *Общие сведения*

Тест Ethernet проводится для модулей, которые имеют порт обмена по Ethernet (HSB или SPE). Перед началом теста в тестируемый модуль должна быть записана рабочая программа модуля SPE.

Тест проверяет состояние связи по сети Ethernet путём пересылки ping-пакетов. С компьютера пользователя через Ethernet-порт ПЭВМ отправляются запросы указанному узлу сети (т.е. модулю HSB или SPE контроллера КСА-02) и фиксируются поступающие ответы. Время между отправкой запроса и получением ответа позволяет определять наличие связи по сети Ethernet и её исправность.

Для запуска теста Ethernet, следует в меню «Тесты» выбрать пункт «Тест Ethernet». При этом сервисная программа контроллера автоматически загружает из базовой директории в процессорный модуль программу теста — TestEthernet.dxe. Если программа теста общей памяти в базовой директории отсутствует, на экране монитора откроется окно, в котором необходимо выбрать файл TestEthernet.dxe.

После загрузки программы теста Ethernet (автоматически или вручную) на экране монитора откроется новое окно «Сервисная программа. Тест Ethernet».

Окно содержит группы "Управление", "Тест Ethernet", "Настройки", "Статистика (модуль CPU)", "Статистика (модуль CPE)" и поле для отображения текстовой информации о результатах работы теста.

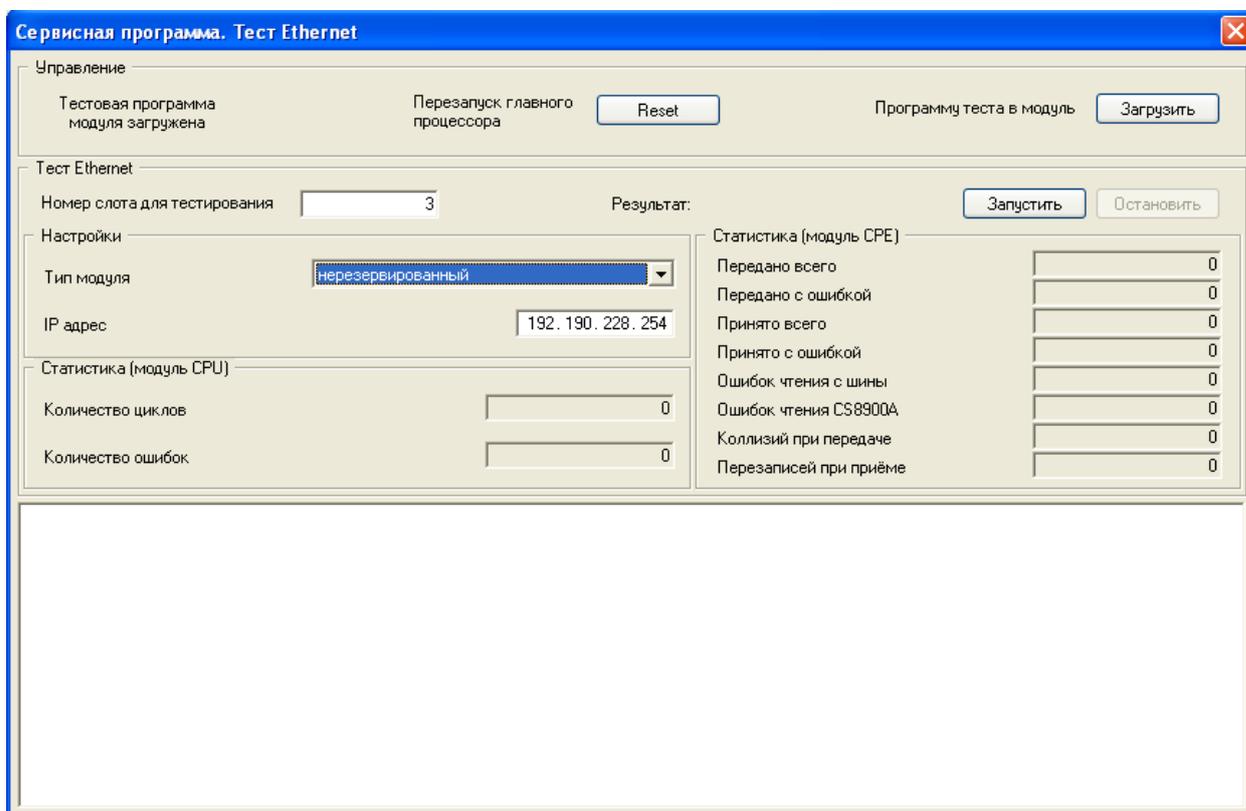


Рис. 5.9.5.1.1. Окно «Тест Ethernet».

5.9.5.2. Управление

Группа содержит индикатор загрузки программы, кнопки "Reset", "Загрузить".

Индикатор загрузки программы во время операции загрузки программы осуществляет графическое отображение процесса. Когда операция загрузки не осуществляется, индикатор представляет собой текстовое сообщение: "Тестовая программа модуля отсутствует", "Тестовая программа модуля загружена", "Связь с контроллером отсутствует".

Reset. Нажатие на кнопку осуществляет перезагрузку главного процессора.

Загрузить. После нажатия на кнопку открывается окно для выбора тестовой программы, а затем осуществляется загрузка выбранной программы в процессорный модуль.

Замечание. При запуске теста из оболочки сервисной программы (п.5.9.5.1) загрузка тестовой программы в модуль осуществляется автоматически.

5.9.5.3. Тест Ethernet

Группа содержит окно "Номер слота для тестирования", поле "Результат", кнопки "Запустить" и "Остановить".

Номер слота для тестирования — задаётся номер слота в корзине, в который вставлен тестируемый модуль (от 3 до 15). По умолчанию значение ячейки 3. Для изменения номера слота необходимо в поле «Номер слота для тестирования:» ввести с клавиатуры нужное значение и зафиксировать изменения нажатием клавиши «Enter».

Результат графически отображает итоги последнего выполненного теста:



– модуль соответствует требованиям, проверяемым данным тестом;



– модуль не соответствует требованиям, проверяемым данным тестом.

Результат отображается после остановки выполнения теста.

Запустить. Кнопка активна при наличии загруженного, но в данный момент не выполняемого теста. Нажатие на кнопку начинает выполнение теста.

Остановить. Кнопка активна при наличии выполняемого теста. Нажатие на кнопку останавливает выполнение теста.

5.9.5.4. Настройки

Группа содержит переключатель "Тип модуля" и поле " IP адрес ".

Тип модуля. Для тестирования предлагается выбрать тип модуля из списка: "нерезервированный", "резервированный, порт 1", "резервированный, порт 2". Для тестирования модулей с одним портом Ethernet следует выбрать пункт "нерезервированный". Два последних пункта предназначены для тестирования модулей, имеющих два порта Ethernet. Выпуск таких модулей в настоящее время не осуществляется.

IP адрес - адрес, с которым будет связываться команда ping. По умолчанию значение поля: 192.190.228.254. Для изменения IP адреса для тестирования, необходимо в поле «IP адрес» ввести с клавиатуры нужное значение и зафиксировать изменения нажатием клавиши «Enter».

5.9.5.5. Статистика (модуль CPU)

Группа «Статистика (модуль CPU)» содержит поля "Количество циклов", "Количество ошибок" и отображает статистику теста на стороне процессорного модуля.

- **количество циклов:** отображает количество выполненных циклов теста;
- **количество ошибок:** отображает общее количество ошибок по всем выполненным циклам теста.

5.9.5.6. Статистика (модуль СРЕ)

Группа «Статистика (модуль СРЕ)» содержит восемь полей и отображает статистику теста на стороне коммуникационного модуля:

- **передано всего:** отображает общее количество пакетов, переданных коммуникационным модулем по сети Ethernet за время тестирования;
- **передано с ошибкой:** отображает количество пакетов, переданных коммуникационным модулем по сети Ethernet с ошибкой;
- **принято всего:** отображает общее количество пакетов, принятых коммуникационным модулем по сети Ethernet;
- **принято с ошибкой:** отображает количество пакетов, принятых коммуникационным модулем по сети Ethernet с ошибкой;
- **ошибки чтения с шины:** отображает число ошибок чтения шины при проведении тестирования на стороне коммуникационного модуля.

Остальные окна оставлены для совместимости со снятыми с производства модулями.

5.9.5.7. Формат отображения текстового сообщения из категории «ошибка чтения/записи»

Оценка состояния связи по Ethernet производится по вышеописанным полям окна теста. Поле, расположенное в нижней части окна теста, зарезервировано для дальнейшего развития сервисной программы.

5.9.6. Тест СОМ-портов

5.9.6.1. Общие сведения

Тест проверяет состояние связи между портами СОМ1 и СОМ2 путём передачи тестовых пакетов данных; данные могут передаваться как с СОМ1 на СОМ2, так и с СОМ2 на СОМ1. Скорость передачи данных также имеет две вариации: 9600 и 115200 бит/с. Тест может проводиться для имеющих СОМ-порты интерфейсных модулей (СРМ, РНА).

Перед началом теста в тестируемый модуль должна быть записана рабочая программа модуля СРМ, а порты СОМ1 и СОМ2 модулей соединены между собой.

Для запуска теста Соm-портов, следует в меню «Тесты» выбрать пункт «Тест Соm-портов». При этом сервисная программа контроллера автоматически загружает из базовой

директории в процессорный модуль программу теста — TestCOM.dxe. Если программа теста Com-портов в базовой директории отсутствует, на экране монитора откроется окно, в котором необходимо выбрать файл TestCOM.dxe.

После загрузки программы теста Com-портов (автоматически или вручную) на экране монитора откроется новое окно «Сервисная программа. Тест Com-портов».

Окно содержит группы "Управление", "Тест СОМ-портов", "Настройки", "Статистика", "Статистика СОМ-портов" и поле для отображения текстовой информации о результатах работы теста.

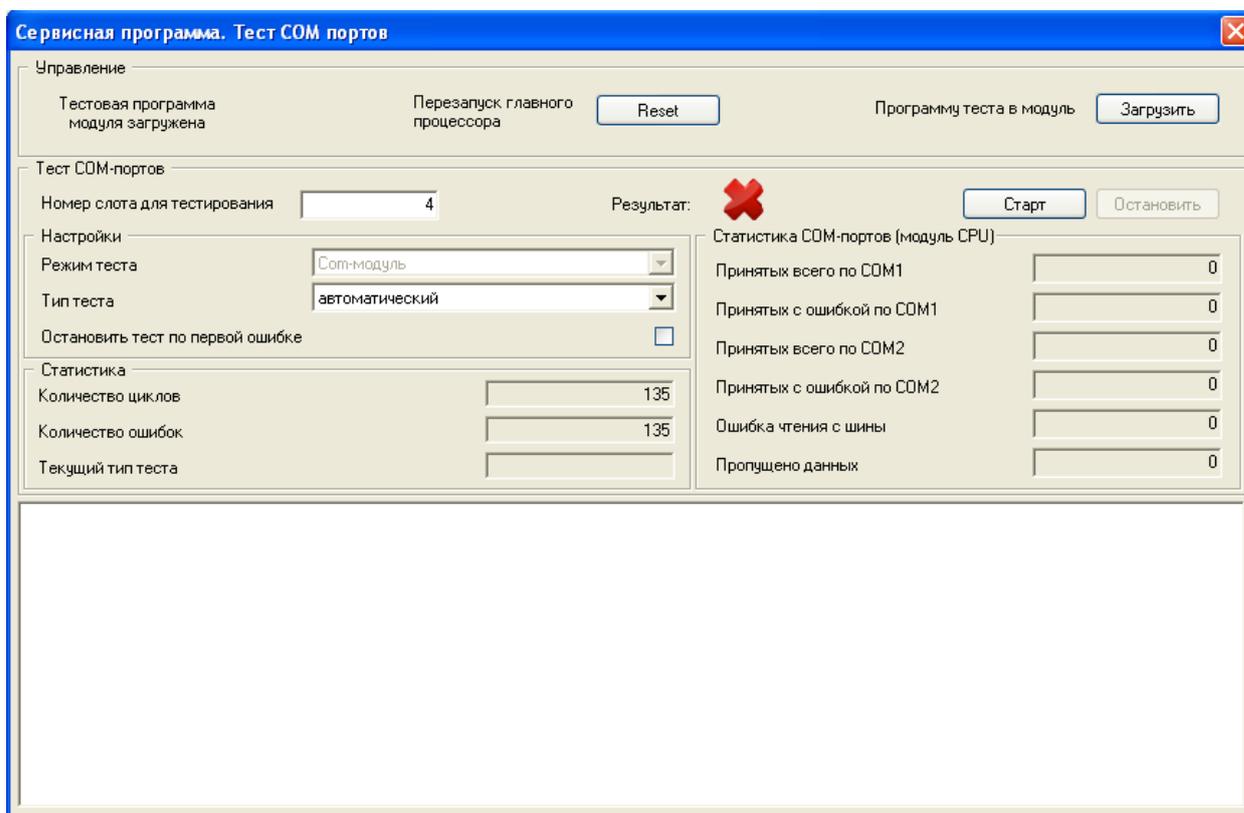


Рис. 5.9.6.1.1. Окно теста СОМ-портов.

5.9.6.2. Управление

Группа содержит индикатор загрузки программы, кнопки "Reset", "Загрузить".

Индикатор загрузки программы во время операции загрузки программы осуществляет графическое отображение процесса. Когда операция загрузки не осуществляется, индикатор представляет собой текстовое сообщение: "Тестовая программа модуля отсутствует", "Тестовая программа модуля загружена", "Связь с контроллером отсутствует".

Reset. Нажатие на кнопку осуществляет перезагрузку главного процессора.

Загрузить. После нажатия на кнопку открывается окно для выбора тестовой программы, а затем осуществляется загрузка выбранной программы в процессорный модуль.

Замечание. При запуске теста из оболочки сервисной программы (п.5.9.6.1) загрузка тестовой программы в модуль осуществляется автоматически.

5.9.6.3. Тест СОМ-портов

Группа содержит окно "Номер слота для тестирования", поле "Результат", кнопки "Старт" и "Остановить".

Номер слота для тестирования — задаётся номер слота в корзине, в который вставлен тестируемый модуль (от 3 до 16). По умолчанию значение ячейки 3.

Для изменения номера слота необходимо в поле «Номер слота для тестирования:» ввести с клавиатуры нужное значение и зафиксировать изменения нажатием клавиши «Enter».

Результат графически отображает итоги последнего выполненного теста:



– модуль соответствует требованиям, проверяемым данным тестом;



– модуль не соответствует требованиям, проверяемым данным тестом.

Результат отображается после остановки выполнения теста.

Старт. Кнопка активна при наличии загруженного, но в данный момент не выполняемого теста. Нажатие на кнопку начинает выполнение теста.

Остановить. Кнопка активна при наличии выполняемого теста. Нажатие на кнопку останавливает выполнение теста.

5.9.6.4. Настройки

Группа содержит переключатели "Режим теста", "Тип теста" и флаг "Останавливать тест при первой ошибке".

Режим теста. В текущей версии программного обеспечения для тестирования доступен только один режим - «Сом-модуль» - тестирование интерфейсного модуля.

Тип теста. Тип теста определяет скорость и направление передачи. Допустимые скорости передачи: 9600, 115200. Допустимые направления передачи: СОМ1 передает и СОМ2 принимает; СОМ2 передает и СОМ1 принимает. Автоматический тип теста проводится

последовательным проведением тестов с различными допустимыми скоростями и направлениями передачи. Всего поддерживается пять типов теста:

- автоматический;
- тест COM1 -> COM2 на скорости 9600;
- тест COM2 -> COM1 на скорости 9600;
- тест COM1 -> COM2 на скорости 115200;
- тест COM2 -> COM1 на скорости 115200.

5.9.6.5. Статистика

Группа «Статистика» содержит поля "Количество циклов", "Количество ошибок", "Текущий тип теста" и отображает статистику теста на стороне процессорного модуля:

- **количество циклов:** отображает количество выполненных циклов теста;
- **количество ошибок:** отображает общее количество ошибок по всем выполненным циклам теста;
- **текущий тест:** отображает тип выполняемого в данный момент теста: COM1 - COM2 на скорости 9600, COM2 - COM1 на скорости 9600, COM1 - COM2 на скорости 115200, прямых адресов, COM2 - COM1 на скорости 115200.

5.9.6.6. Статистика COM- портов

Группа «Статистика COM - портов» содержит шесть полей и отображает статистику теста на стороне интерфейсного модуля:

- **принято всего по COM1:** отображает общее количество пакетов, принятых интерфейсным модулем по порту COM1;
- **принято с ошибкой по COM1:** отображает количество пакетов, принятых интерфейсным модулем с ошибкой по порту COM1;
- **принято всего по COM2:** отображает общее количество пакетов, принятых интерфейсным модулем по порту COM2;
- **принято с ошибкой по COM2:** отображает количество пакетов, принятых интерфейсным модулем с ошибкой по порту COM2;
- **ошибки чтения с шины:** отображает число ошибок чтения шины при проведении тестирования на стороне интерфейсного модуля.
- **пропущено данных:** во время тестирования интерфейсных модулей возможны ситуации, когда буфер обмена тестируемого модуля полностью занят и не может принять все поступающие данные, которые ему по шине передаёт процессорный

модуль. Часть данных оказывается пропущенными; их количество отображается в поле «Пропущено данных».

5.10. Предупредительные и аварийные сообщения сервисной программы

5.10.1. Классификация и способы отображения предупредительных и аварийных сообщений сервисной программы

Предупредительные и аварийные сообщения сервисной программы контроллера КСА-02 по причине своего возникновения разделяются на следующие категории:

- ошибка связи ПЭВМ с контроллером КСА-02 (далее по тексту «ошибка связи»);
- прочие.

Возможные причины ошибки связи и действия по их устранению приведены в таблице 5.10.1:

Таблица 5.10.1

Категория ошибки	Возможная причина ошибки	Действия оператора по устранению ошибки
Ошибка связи	Не подключен интерфейсный жгут	Подключить интерфейсный жгут
	Ошибки в разводке интерфейсного жгута	Заменить интерфейсный жгут
	Неверно установлены параметры обмена по последовательному порту	Завершить работу сервисной программы контроллера КСА-02 и запустить её снова, правильно установив номер СОМ-порта и адрес контроллера
	Контроллер не включен или неисправен	Проверить правильность включения контроллера

Предупредительные и аварийные сообщения сервисной программы контроллера КСА-02 отображаются двумя способами:

- в нижней информационной строке основного окна;
- в окне «Ошибка».

5.10.2. Предупредительные и аварийные сообщения, отображаемые в нижней информационной строке основного окна сервисной программы

Таблица 5.10.2

Режим (пункты главного меню)	Выполняемая операция	Текст сообщения об ошибке	Категория ошибки	Возможная причина ошибки	Действия оператора по устранению ошибки
Настройка	Конфигурация контроллера	Не удалось произвести запись конфигурации.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Не удалось произвести чтение конфигурации.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Обнаружен признак первой загрузки.	Прочие	Не производилась запись конфигурации модуля ввода/вывода в память контроллера КСА-02.	Произвести конфигурирование контроллера КСА-02.
Управление	Запустить процессор	Контроллер не отвечает на команду старта процессора.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Остановить процессор	Контроллер не отвечает на команду остановки процессора.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Перезапустить процессор	Контроллер не отвечает на команду перезапуска процессора.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Остановить/ Запустить ядро	Контроллер не отвечает на команду остановки/ запуска ядра.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Загрузить программу	Программу не удалось загрузить в контроллер.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.

Режим (пункты главного меню)	Выполняемая операция	Текст сообщения об ошибке	Категория ошибки	Возможная причина ошибки	Действия оператора по устранению ошибки
		Неправильный формат файла.	Прочие	Формат файла является не допустимым для выполняемых файлов системы «Scorpio».	Загрузить файл, формат которого является допустимым для выполняемых файлов системы «Scorpio».
		Невозможно прочитать файл.	Прочие	ОС, под управлением которой работает сервисная программа, не может прочитать файл.	Проверить правильность файла и разрешение ОС на доступ к этому файлу.
Алгоритм	Запустить алгоритм	Контроллер не отвечает на команду запуска алгоритма.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Удалить алгоритм	Контроллер не отвечает на команду удаления алгоритма.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Загрузить алгоритм	Алгоритм не удалось загрузить в контроллер.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
	Выбор модуля	Не удалось определить параметры слота.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Не удалось произвести чтение конфигурации.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Конфигурация контроллера не определена.	Прочие	Не производилась запись конфигурации модуля ввода/вывода в память контроллера КСА-02.	Произвести конфигурирование контроллера КСА-02.

5.10.3. Предупредительные и аварийные сообщения, отображаемые в окне «Ошибка»

Таблица 5.10.3

Режим (пункты главного меню)	Выполняемая операция	Текст сообщения об ошибке	Категория ошибки	Возможная причина ошибки	Действия оператора по устранению ошибки	
Настройка	Конфигурация контроллера	Не удалось произвести чтение конфигурации.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
Тест	Тест модулей дискретного ввода/вывода	Не удалось вернуть значения блокировок.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
		Не удалось остановить ядро.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
		Контроллер не отвечает на команду запуска ядра. Тест будет завершен.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
		Не удалось провести инициализацию параметров. Тест будет завершен.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
		Устройство не отвечает. Код ошибки.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
	Тест модулей аналогового ввода/вывода	Не удалось запустить ядро.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
		Обмен с модулем завершен с ошибкой. Продолжить?	Прочие	Неисправность или отсутствие тестируемого модуля.	Вставить в слот исправный модуль.	
		Не удалось остановить ядро.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.	
	Тест ОЗУ	Тест ОЗУ	Не удалось определить настройки модуля. Тест будет завершен.	Прочие	Неисправность или отсутствие тестируемого модуля.	Вставить в слот исправный модуль.
			Не удалось найти исполняемый файл testmem.	Прочие	Файл testmem отсутствует в базовой директории.	

Режим (пункты главного меню)	Выполняемая операция	Текст сообщения об ошибке	Категория ошибки	Возможная причина ошибки	Действия оператора по устранению ошибки
		Невозможно прочитать файл.	Прочие	ОС, под управлением которой работает сервисная программа, не может прочитать файл.	Проверить правильность файла и разрешение ОС на доступ к этому файлу.
		Неправильный формат файла.	Прочие	Формат файла является не допустимым для выполняемых файлов системы «Scorpio».	Загрузить файл, формат которого является допустимым для выполняемых файлов системы «Scorpio».
		Не обнаружен загруженный тест памяти. Загрузить?	Прочие	В контроллере отсутствует признак наличия теста ОЗУ.	Загрузить исполняемый файл testmem
		Не удалось загрузить программу	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Устройство не отвечает.<Расшифровка причины>. Продолжить?	Ошибка связи или ошибка чтения/записи		
	Тест модуля ST2CPU33	Контроллер не отвечает на команду остановки процессора.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.
		Глобальная ошибка.<расшифровка ошибки>. Тест будет остановлен.	Ошибка связи	См. таблицу 5.10.1.	См. таблицу 5.10.1.

5.11. Справочная система

Для работы со справочной системой в сервисной программе предназначено меню «Помощь»:

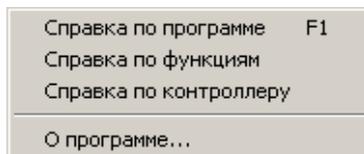


Рис. 5.11.1. Меню «Помощь».

Меню «Помощь» содержит следующую информацию:

Справка по программе (F1): информация о работе сервисной программы контроллера КСА-02:

- справка по всем меню сервисной программы;
- справка о назначении полей и кнопок использующихся окон;
- правила работы с сервисной программой (использование «горячих» клавиш и т.п.);

Справка по функциям: полный список поддерживаемых системой «Scorpio» функций с подробным описанием их назначения, принципа действия и внешнего вида соответствующих блоков.

Справка по контроллеру: полный список поддерживаемых системой «Scorpio» модулей ввода/вывода с описанием их назначения.

О программе...: информация о названии программы, её версии и годе релиза.

6. Протоколы

6.1. Введение

Список поддерживаемых протоколов обмена данными ранее уже был приведен в п.2.1.4. Процедура настройки различных протоколов приведена в п.4.2.2.2, пп.4.2.2.9 - 4.2.2.17.

6.1.1. Физические линии связи

Вообще говоря, использование того или иного протокола обмена данными требует применения линии связи вполне определенного типа.

Контроллер может работать по следующим линиям связи:

- через последовательный порт по проводной линии связи, интерфейс RS485 или RS232;
- через последовательный порт с подключенным модемом по радиосети;
- через порт Ethernet по сети Ethernet.

При работе по проводной линии связи через последовательный порт поддерживаются протоколы Modbus, MKSA, RTM64, Proton3-slave. Планируется поддержка протокола МЭК870-5-101 (IEC60870-101).

Все вышеперечисленные протоколы могут использоваться и при передаче данных по радиоканалу через последовательный порт с модемом; при этом протоколы RTM64 и Proton3-slave специально разработаны для этой задачи. Протокол Proton3-GSM может использоваться только для передачи данных по сети GSM через последовательный порт с подключенным GSM-модемом.

При работе по сети Ethernet поддерживаются протоколы Modbus-TCP, Proton3-IP, МЭК870-5-104 (IEC60870-104).

6.1.2. Принципы организации обмена данными

По используемым принципам организации обмена данными используемые протоколы можно разделить на следующие классы:

- запросные,
- маркерные (полуинициативные),
- инициативные.

В запросных протоколах ведомое устройство передает данные по запросу мастера, в объеме, который указан в запросе.

В маркерных протоколах ведомое устройство само определяет, какие данные необходимо передать. Передача данных осуществляется после получения маркера от мастера.

В инициативных протоколах ведомое устройство, при наличии технической возможности, передает данные сразу после возникновения необходимости их передачи без запроса со стороны мастера.

К запросным протоколам относятся протоколы Modbus, Modbus-TCP, MKSA, RTM64.

К маркерным протоколам относятся протоколы Proton3-slave, Proton3-IP.

К инициативным протоколам относятся протоколы Proton3-GSM и МЭК870-5-104.

Для обеспечения устойчивого обмена данными и поддержания баз данных в актуальном состоянии маркерные и инициативные протоколы обычно поддерживают команды принудительной передачи всех или части данных.

Инициативные протоколы поддерживают служебные команды проверки связи, которые передаются в случае долгого отсутствия данных для передачи.

При прочих равных условиях, трафик обмена для маркерных и инициативных протоколов обычно значительно ниже, чем для запросных.

6.1.3. Программные уровни

Организация обмена данными с внешними устройствами через порты является комплексом взаимодействующих задач подсистемы выполнения контроллера КСА-02.

Можно выделить следующие программные уровни этих задач:

- низкий уровень,
- системный уровень,
- внутренний транспортный уровень,
- уровень алгоритма (пользовательский).

В данном разделе основное внимание уделено задачам, решаемым на уровне алгоритма.

На низком уровне решается задача работы приемо-передатчиков портов. Запуск задачи осуществляется или по прерыванию обслуживания порта, или циклически с малым

периодом (13 мкс). Задачи низкого уровня имеют более высокий приоритет, чем остальные задачи. Также на низком уровне решается задача формирования буфера ответных данных для slave-протоколов запросного класса.

На системном уровне решается задача формирования буфера данных для передачи на низкий уровень и обработки буфера данных принятых на низком уровне. Для портов, находящихся на процессорном модуле, эта задача решается ядром процессорного модуля. Задача запускается периодически один раз в 50 мс, приоритет задачи обычный. Для портов, находящихся на интерфейсном модуле, эти задачи решаются ядром интерфейсного модуля. Для интерфейсных модулей с версией программного обеспечения меньше 5.0 задача имеет обычный приоритет и может быть запущена сразу после возникновения (если модуль в этот момент занят решением других задач равного приоритета, то после их окончания). Для интерфейсных модулей с версией программного обеспечения выше 5.0 задача запускается циклически через 500 мкс с повышенным приоритетом.

Внутренний транспортный уровень возникает при обмене данными через порты, находящиеся на интерфейсных модулях. На процессорном модуле к данному уровню относятся задачи обмена данными по протоколу Proton3 и задача обмена данными с интерфейсным модулем. Задача обмена данными по протоколу Proton3 запускается циклически (через 13 мкс) и имеет приоритет, равный приоритету задач низкого уровня. Задача обмена данными с интерфейсным модулем запускается один раз в 50 мс, приоритет задачи обычный. На интерфейсном модуле ситуация несколько более сложная, так как процессорный модуль записывает данные в несколько разных областей памяти, а чтение каждой области памяти представляет собой отдельную задачу. Для интерфейсных модулей с версией программного обеспечения меньше 5.0 все эти задачи имеют обычный приоритет и, при отсутствии других выполняемых задач, запускаются сразу после возникновения. Для интерфейсных модулей с версией программного обеспечения выше 5.0 эти задачи запускаются циклически один раз в 500 мкс, причем чтение некоторых областей памяти имеет повышенный приоритет, а чтение других областей памяти - обычный приоритет.

Задача алгоритма запускается один раз в 50 мс, приоритет задачи обычный. На этом уровне (в целях осуществления обмена данными с внешними устройствами) осуществляется:

- формирование запросов данных на ведомые устройства для Master - протоколов запросного типа;
- дальнейшая обработка данных, поступивших от ведомых устройств;
- формирование требований на передачу собственных данных для протоколов маркерного типа;
- обработка команд, поступивших от ведущих устройств для протоколов Slave.

6.2. Протокол Modbus в режиме Master (Modbus-Master)

6.2.1. Общие сведения

Протокол Modbus-Master является протоколом запросного типа и независимо поддерживается на всех портах модулей СТ2CPU33 и СТ1CPM10. Может быть установлена следующая скорость передачи данных:

- для порта COM1 модуля СТ2CPU33 — скорость 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с;
- для порта COM2 модуля СТ2CPU33 — скорость 1200, 2400, 4800, 9600, 19200 бит/с;
- для COM-портов модуля СТ1CPM10 — 1200, 2400, 4800, 9600, 19200, 38400 бит/с.

Настройка протокола описана в п.4.2.2.2 и п.4.2.2.12.

Протокол имеет две модификации: Modbus-RTU и Modbus-ASCII. С точки зрения программиста, обращение из алгоритма к этим модификациям совершенно одинаково. Тип поддерживаемой модификации протокола для ведомого устройства обычно указывается в документации на это устройство. Следует отметить, что протокол Modbus-ASCII является устаревшим, и применяется в настоящий момент очень редко.

6.2.2. Типы данных

Для протокола Modbus в стандартной конфигурации определены четыре типа данных, которые в англоязычной литературе обычно называются coil status (output status), input status, holding register (output register), input register. Прямой перевод на русский язык данных терминов иногда вызывает путаницу, так как направление данных в них указано со стороны мастера, а сами данные формируются ведомым устройством. Последовательная адресация элементов данных каждого типа осуществляется при помощи адресных ссылок. Ссылка является числом из четырех или пяти цифр. Первая цифра ссылки однозначно связана с типом данных. Остальные цифры являются порядковым номером элемента в

типе данных. Порядковый номер элемента в ссылке начинается с 1. Например, первый элемент в типе данных "Целый только для чтения" имеет адресную ссылку 30001 (или 3001). Таблица конкретизирует типы данных Modbus.

Тип данных	Описание	Размер (бит)	Вид ссылки
Coil status	Логический для чтения и записи	1	0XXX, 0XXXX
Input status	Логический только для чтения	1	1XXX, 1XXXX
Holding register	Целый для чтения и записи	16	4XXX, 4XXXX
Input register	Целый только для чтения	16	3XXX, 3XXXX

При использовании протокола Modbus-master контроллер может запрашивать у ведомого устройства данные всех типов, и осуществлять запись данных типа Coil status и Holding register.

6.2.3. Адресация и функции доступа

В базах данных, приводимых в документации на устройства, адресация приводится в виде ссылок. В функциях доступа к данным адресация задается в виде смещения от начального элемента типа данных. Смещение начального элемента типа данных равно нулю. Смещение элемента можно рассчитать по формуле (порядковый номер элемента минус 1). Например, смещение элемента с адресной ссылкой 10017 равно 16. В таблице приведен список функций доступа к данным, поддерживаемых контроллером КСА-02.

Функция	Описание	Вид ссылки	Макс.число элементов за один запрос
01	Чтение нескольких последовательных элементов типа Coil status	0XXX, 0XXXX	2000
02	Чтение нескольких последовательных элементов типа Input status	1XXX, 1XXXX	2000
03	Чтение нескольких последовательных элементов типа Holding register	4XXX, 4XXXX	125
04	Чтение нескольких последовательных элементов типа Input register	3XXX, 3XXXX	125
05	Запись одного элемента типа Coil status	0XXX, 0XXXX	1

06	Запись одного элемента типа Holding register	4XXX, 4XXXX	1
15	Запись нескольких последовательных элементов типа Coil status	0XXX, 0XXXX	2000 ⁽¹⁾
16	Запись нескольких последовательных элементов типа Holding register	4XXX, 4XXXX	60

⁽¹⁾ Максимальное число записываемых элементов для данной функции не нормировано. Указанное ограничение относится к контроллеру КСА-02. Чтобы узнать ограничение применения данной функции, накладываемое ведомым устройством, следует обратиться к документации на ведомое устройство.

6.2.4. Использование в алгоритме.

Выдача запросов по протоколу Modbus-master осуществляется с помощью функции алгоритма F0102. Функция имеет 6 входов и 2 выхода.

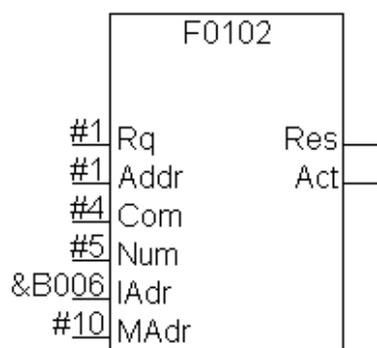


Рис. 6.1.3.1. Внешний вид функции F0102.

- Rq — требование передачи: если на входе стоит 0, то функция F0102 не работает и отключена, следовательно, обмен данными по протоколу не производится. Если на входе стоит 1, функция включена, обмен производится (тип LOGIC);
- Addr — адрес в сети ведомого устройства (тип INTEGER);
- Com: функция протокола Modbus (см. п.6.2.3) (тип INTEGER);
- Num: количество данных для передачи или приёма (тип INTEGER);
- IAdr: адрес ячейки во внутреннем буфере Master-устройства, с которой начинается область памяти, предназначенная для данных, полученных от Slave-устройства либо предназначенных для передачи в него (тип INTEGER);
- MAdr: смещение первого запрашиваемого элемента ведомого устройства (см. п.6.2.3) (тип INTEGER);

- Res: результат последнего обработанного обмена: если обработка произведена успешно и без ошибок, то на выходе появляется 1. Если обмен не состоялся или был произведён с ошибками, то на выход выставляется 0. До окончания обработки на выходе находится результат последнего обработанного обмена (тип LOGIC);
- Act: признак занятости функции: 1 на выходе означает, что функция приняла данные и находится в процессе их обработки. 0 на выходе означает, что функция обработала принятые данные и свободна для обработки следующих (тип LOGIC).

Вход «**Addr**» может инициализироваться литерой вида #aXX.YY (см. раздел приложения [«Зарезервированные символы для формирования литеры»](#)), где XX — номер COM-порта Master-устройства, а YY — адрес Slave-устройства.

Доступ к принятым данным осуществляется функцией чтения внутреннего буфера F0099 по адресу IAdr и последующим адресам; запись данных для передачи осуществляется функцией записи во внутренний буфер по адресу IAdr или эквивалентными этим функциям контактами.

Команда 5 обеспечивает доступ для записи на установку или сброс одиночного статуса со ссылками 0x вида 0XXX. Для установки статуса данные, расположенные по адресу «IAdr», должны иметь значение 0x0001, а для сброса — 0x0000.

Для команд 1, 2 при числе запрашиваемых данных не кратному 16, принятые биты добавляются нулями до целого слова.

Пример:

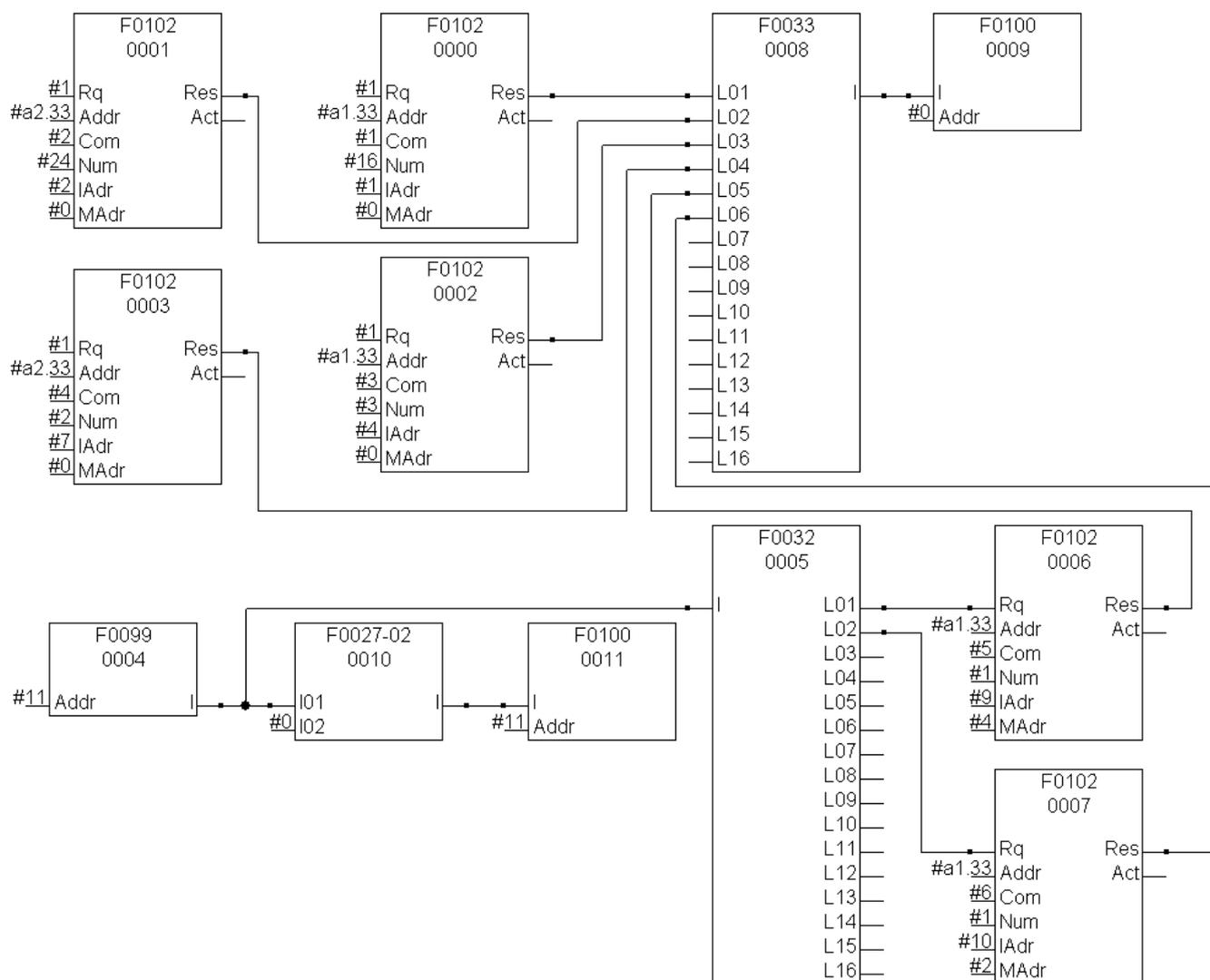


Рис. 6.2.4. Пример работы функции F0102 –передача по протоколу Modbus.

Функции F0102 (0000), (0002), (0006), (0007) ведут обмен по порту COM1 с устройством, имеющим сетевой адрес 33. Функции F0102 (0001) и (0003) ведут обмен по порту COM2 с устройством, имеющим сетевой адрес 33. Для определения COM2 к адресу устройства прибавляется число 256.

Наличие постоянной 1 на входе «Rq» функций (0000), (0001), (0002) и (0003) означает, что требование передачи для этих функций выдаётся постоянно.

Для функций с порядком выполнения (0006), (0007) однократное требование передачи возникает в случае записи 1 в два младших бита ячейки 11 внутреннего буфера.

Таким образом, в алгоритме установлена максимальная частота обмена по функциям с порядком выполнения (0000), (0001), (0002) и (0003); а обмен по функциям (0006) и (0007) осуществляется по требованию.

Одновременно драйвер выполняет задание только для одной функции, точнее — одной функции для каждого Com-порта (в данном примере — одной функции для COM1 и одной функции для COM2), другие функции, имеющие запрос на передачу, находятся в режиме ожидания освобождения драйвера.

Функция с порядковым номером выполнения (0000) с командой Modbus 1 осуществляет запрос 16 последовательных статусов, начиная со статуса 0001. В случае успешного обмена, обновленные данные будут расположены в ячейке с адресом 1 внутреннего буфера.

Функция с порядковым номером (0001) и командой Modbus 2 осуществляет запрос 24 последовательных статусов, начиная со статуса 1001. В случае успешного обмена, данные будут находиться во внутреннем буфере в ячейках с адресами 2 и 3, причём 8 старших бит в ячейке 3 будут автоматически дополнены нулями.

Функция с порядковым номером выполнения (0002) командой Modbus 3 осуществляет запрос 3 последовательных регистров, начиная с регистра 4001. Принятые данные при удачном обмене будут расположены во внутреннем буфере в ячейках с адресами с 4 по 6.

Функция с порядковым номером (0003) и командой Modbus 4 осуществляет запрос 2 последовательных регистров, начиная с регистра 3001. Принятые данные при удачном обмене будут находиться во внутреннем буфере в ячейках с адресами 7 и 8.

При выставлении требования обмена функция с порядковым номером выполнения (0006) командой Modbus 5 осуществляет установку или сброс статуса 0005, если в ячейке с адресом 9 внутреннего буфера будет записано 0xFF00 или 0x0000 соответственно. При других значениях в ячейке внутреннего буфера с адресом 9 команда будет признана ошибочной.

При выставлении требования обмена функция с порядковым номером (0007) командой Modbus 6 осуществляет запись в регистр 4003 числа, находящегося в ячейке внутреннего буфера с адресом 10.

Результат обмена (тип данных «логический») поступает на функцию F0033 порядковый номер выполнения в алгоритме (0008), преобразуются в тип данных «целый» и функцией F0100 порядковый номер (0009) записываются во внутренний буфер в ячейку по смещению 0.

6.3. Протокол Modbus в режиме Slave (Modbus-Slave)

6.3.1. Общие сведения

Протокол Modbus-slave является протоколом запросного типа и независимо поддерживается на всех портах модулей СТ2CPU33 и СТ1СРМ10. Для каждого порта может быть установлен индивидуальный сетевой адрес. Контроллер поддерживает следующие скорости передачи данных:

- для порта СОМ1 модуля СТ2CPU33 — скорость 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с;
- для порта СОМ2 модуля СТ2CPU33 — скорость 1200, 2400, 4800, 9600, 19200 бит/с;
- для СОМ-портов модуля СТ1СРМ10 — 1200, 2400, 4800, 9600, 19200, 38400 бит/с.

Настройка протокола описана в п.4.2.2.2 и п.4.2.2.10.

Протокол используется в конфигурации Modbus-RTU.

6.3.2. Особенности адресации и использования функций доступа

Общее введение в типы данных, адресацию и функции доступа для протокола Modbus приведено в пп. 6.2.2, 6.2.3. При использовании протокола Modbus-slave контроллер КСА-02 является ведомым устройством и возвращает мастеру запрашиваемые им данные. В таблице обозначено соответствие между поддерживаемыми контроллером функциями доступа и данными контроллера.

Функция	Описание	Вид ссылки	Макс.число элементов за один запрос
01	Чтение нескольких последовательных элементов из буфера контроллера логический входной	0XXX, 0XXXXX	2000
02	Чтение нескольких последовательных элементов из буфера контроллера логический выходной	1XXX, 1XXXXX	2000
03	Чтение нескольких последовательных элементов из буфера контроллера целый входной	4XXX, 4XXXXX	125
04	Чтение нескольких последовательных элементов из буфера контроллера целый выходной	3XXX, 3XXXXX	125
05	Запись одного элемента в буфер контроллера логический входной	0XXX, 0XXXXX	1
06	Запись одного элемента в буфер контроллера	4XXX, 4XXXXX	1

	целый входной		
15	Запись нескольких последовательных элементов в буфер контроллера логический входной	0XXX, 0XXXX	2000
16	Запись нескольких последовательных элементов в буфер контроллера целый входной	4XXX, 4XXXX	60

6.3.3. Использование в алгоритме.

Запросы данных (функции 01, 02, 03, 04) обрабатываются сразу на низком уровне (п.6.1.3); данными для передачи являются текущие значения соответствующих буферов контроллера (п.6.3.2). На уровне алгоритма требуется постоянно поддерживать актуальные значения буферов. Это осуществляется функциями алгоритма F0012 (передатчик в логический выходной буфер), F0013 (передатчик в целый выходной буфер) или соответствующими контактами. Прямая запись данных из алгоритма во входные буфера возможна, но используется редко: по смыслу, эти буфера предназначены для хранения входных, а не обработанных данных. Если такая необходимость все же возникает, можно использовать функции алгоритма F0145 (запись в логический входной буфер), F0146 (запись в логический выходной буфер).

Поступающие данные и команды (05, 06, 15, 16) после обработки их на низком, транспортном и системном уровнях оказываются записанными в соответствующие буфера контроллера. Обработка команд в алгоритме заключается, во-первых, в установлении самого факта приема команды, и, во-вторых, в проведении действий, определенных этой командой. Установление факта получения команды обычно осуществляется просмотром текущего состояния входных буферов и сравнения их с предыдущим значением. Для этого применяются функции алгоритма F0009 (приёмник из логического входного буфера), F0010 (приёмник из целого входного буфера). На практике также часто используется (особенно для логических команд) чтение с одновременным обнулением значения в буфере. Для этого используются функции алгоритма F0094 (приёмник со сбросом из логического входного буфера), F0095 (приёмник со сбросом из целого входного буфера). Дальнейшие действия после получения команды зависят от самой команды, стандартные обработчики команд оформляются в виде составных функций.

6.4. Особенности протокола Modbus-TCP

6.4.1. Особенности Modbus-TCP (Master)

Протокол Modbus-TCP (Master) является протоколом запросного типа и поддерживается на порту RJ45 модуля СТ1СРЕ10. Поддержка и параметры настройки протокола должны быть объявлены в алгоритме (4.2.2.17).

При использовании протокола Modbus-TCP (master) контроллер запрашивает данные и осуществляет передачу команд на ведомые устройства. Ведомые устройства могут находиться в основной или в дополнительных сетях контроллера.

Типы данных, адресация и поддерживаемые функции доступа соответствуют протоколу Modbus-master (п.6.2.2, 6.2.3).

Для формирования запросов по протоколу Modbus-TCP (master) в алгоритме используется функция F0174. Функция имеет 7 входов и 3 выхода. Описание входов и работа функции близки к F0102, с той разницей, что данные передаются на порт RJ45 модуля СТ1СРЕ10.

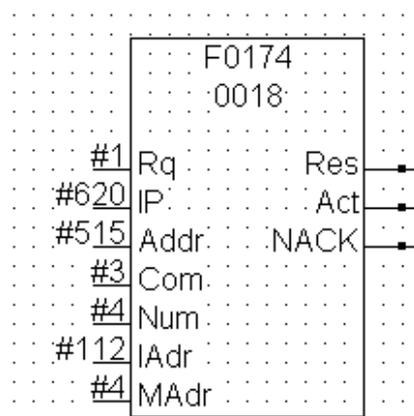


Рис. 6.4.1. Внешний вид функции F0174.

- Rq — требование передачи: если на входе стоит 0, то функция F0174 не работает и отключена, следовательно, обмен данными по протоколу не производится. Если на входе стоит 1, функция включена, обмен производится (тип LOGIC);
- IP — IP адрес в сети ведомого устройства (тип INTEGER);
- Addr — адрес в сети ведомого устройства (тип INTEGER);
- Com: функция протокола Modbus (см. п.6.2.3) (тип INTEGER);
- Num: количество данных для передачи или приёма (тип INTEGER);

- IAdr: адрес ячейки во внутреннем буфере Master-устройства, с которой начинается область памяти, предназначенная для данных, полученных от Slave-устройства либо предназначенных для передачи в него (тип INTEGER);
- MAdr: смещение первого запрашиваемого элемента ведомого устройства (см. п.6.2.3) (тип INTEGER);
- Res: результат последнего обработанного обмена: если обработка произведена успешно и без ошибок, то на выходе появляется 1. Если обмен не состоялся или был произведён с ошибками, то на выход выставляется 0. До окончания обработки на выходе находится результат последнего обработанного обмена (тип LOGIC);
- Act: признак занятости функции: 1 на выходе означает, что функция приняла данные и находится в процессе их обработки. 0 на выходе означает, что функция обработала принятые данные и свободна для обработки следующих (тип LOGIC).
- NACK: признак внезапного разрыва соединения: 1 на выходе означает, что соединение с ведомым устройством внезапно разорвано (тип LOGIC).

По сравнению с функцией F0102, в функции F0174 появились дополнительные входы и выходы.

Вход «IP». Старший байт входа указывает номер сети (0 - основная сеть, 1,2,3 - первая, вторая, третья дополнительные сети соответственно). Младший байт входа является младшим октетом IP адреса ведомого устройства. Три старших октета IP адреса являются общими для всей сети и берутся из настроек протокола. Вход может инициализироваться литерой вида #aXX.YY (см. раздел приложения [«Зарезервированные символы для формирования литеры»](#)), где XX — номер сети, в которой находится ведомое устройство, а YY — младший октет IP-адреса ведомого устройства.

Выход «NACK». Выход имеет смысл пропадания связи с ведомым устройством в процессе передачи запроса. В таблице описаны основные ошибки связи и соответствующие им значения флагов Res и NACK.

Описание ошибки	Res	NACK
Соединение с ведомым устройством установить не удалось	0	0
Соединение с ведомым устройством установлено. Запрос передан. Подтверждение запроса получено. Ответные данные не приняты.	0	0
Соединение с ведомым устройством установлено. Запрос передан. Подтверждение запроса не получено.	0	1

6.4.2. Особенности Modbus-TCP (Slave)

Протокол Modbus-TCP (Slave) является протоколом запросного типа и поддерживается на порту RJ45 модуля СТ1СРЕ10. Поддержка протокола осуществляется независимо от протокола, выбранного для модуля СТ1СРЕ10. Данные доступны по порту 502. О настройках протокола см.п. 4.2.2.13, 4.2.2.14.

При использовании протокола Modbus-TCP (Slave) контроллер КСА-02 является ведомым устройством и возвращает мастеру запрашиваемые им данные.

В остальном работа по протоколу Modbus-TCP (Slave) соответствует работе по протоколу Modbus-Slave (п.6.3.2, 6.3.3).

6.5. Протокол MKSA в режиме Master (MKSA-Master)

6.5.1. Общие сведения

Протокол MKSA-Master является протоколом запросного типа и независимо поддерживается на всех портах модулей СТ1СРМ10. Может быть установлена следующая скорость передачи данных:

- для COM-портов модуля СТ1СРМ10 — 1200, 2400, 4800, 9600, 19200, 38400 бит/с.

Настройка протокола описана в п.4.2.2.12.

При работе по протоколу MKSA-мастер контроллер запрашивает данные у ведомых устройств.

Протокол MKSA поддерживают контроллеры семейства МКСА, а также коммуникационное устройство связи со счетчиками электроэнергии "Меркурий".

6.5.2. Основная характеристика контроллеров МКСА

Понятие "Тип данных" для протокола MKSA не вводится. Вместо него существует понятие "Основная характеристика контроллера". Основная характеристика является структурой, которая включает в себя входные и выходные данные контроллера МКСА в соответствии с его назначением. Запрос основной характеристики позволяет передать или получить все данные от контроллера за один запрос. Расшифровка полученных данных по типам (если требуется) затем осуществляется средствами алгоритма. Основная характеристика каждого контроллера из семейства МКСА индивидуальна, и приводится в описании на данный тип контроллера. В таблице обобщены команды для запроса основной характеристики контроллеров семейства МКСА.

Тип контроллера	Команда	Описание основной характеристики	Слов на передачу	Слов на прием
МКСА-01М	0	16 ТС (16 бит), Переполнение (16 бит), Счетчики импульсов (16 слов)	0	18
МКСА-02М	0	8 ТС (8 бит)	0	1
	1	8 ТУ (8 бит)	1	0
МКСА-03М	0	8 ТИ (8 слов), признак ошибки (8 бит)	0	9
МКСА-05М	1	16ТУ (16 бит)	1	0
МКСА-06М	0	температура по 8 каналам термопар (8 слов), температура холодного спая (1 слово), признак ошибки (9 бит)	0	10
МКСА-07М	0	8RTD (8 слов), признак ошибки (8 бит)	0	9

6.5.3. Коммуникационное устройство связи со счетчиками "Меркурий"

Коммуникационное устройство связи со счетчиками электроэнергии "Меркурий" преобразует входной запрос по протоколу МКСА в запрос по протоколу счетчика "Меркурий". Ответные данные, полученные от счетчика, преобразуются в данные по протоколу МКСА и передаются на контроллер.

При работе с коммуникационным устройством должны быть установлены следующие параметры последовательного порта: скорость - 9600, четность - дополнение до нечетности. Время ответа рекомендуется установить в диапазоне 50 - 60 мс.

В таблице обобщены поддерживаемые коммуникационным устройством команды для запроса данных счетчиков "Меркурий" .

Описание запроса	Команда	Слов на передачу	Слов на прием
Чтение массивов накопленной энергии	5	1	4
Чтение вспомогательных параметров	8	1	2
Групповое чтение вспомогательных параметров (мощность)	8	1	5
Групповое чтение вспомогательных параметров (напряжение, ток)	8	1	4

Более подробно о работе с коммуникационным устройством и расшифровке слов данных смотри документы "Описание коммуникационного устройства" и "Протокол Меркурий 230".

6.5.4. Использование в алгоритме.

Выдача запросов по протоколу MKSA-master осуществляется с помощью функции алгоритма F0103. Функция имеет 7 входов и 2 выхода.

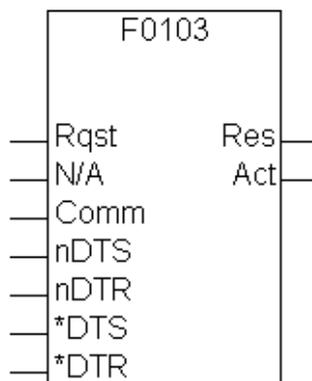


Рис. 6.5.4. Внешний вид функции F0103.

Функция F0103 имеет следующие входы и выходы.

- Rqst: требование передачи: если на входе стоит 0, то функция F0103 не работает и отключена, следовательно, обмен данными по протоколу не производится. Если на входе стоит 1, функция включена, обмен производится (тип входа LOGIC);
- N/A: сетевой адрес MKSA-устройства и номер COM-порта (INTEGER);
- Com команда протокола MKSA (тип INTEGER);
- nDTS: число слов данных, которое контроллер передаст удалённому устройству (тип INTEGER);
- nDTR: число слов данных, которое контроллер примет от удалённого устройства (тип INTEGER);
- *DTS: адрес ячейки памяти во внутреннем буфере Master-устройства, с которой начинается область, содержащая данные для передачи Slave-устройству (тип INTEGER);
- *DTR: адрес ячейки памяти во внутреннем буфере Slave-устройства, с которой начинается область, содержащая данные, предназначенные на передачу Master-устройству (тип INTEGER);

- Res: результат последнего обработанного обмена: если обработка произведена успешно и без ошибок, то на выходе появляется 1. Если обмен не состоялся или был произведён с ошибками, то на выход выставляется 0. До окончания обработки на выходе находится результат последнего обработанного обмена (тип LOGIC);
- Act: признак занятости функции: 1 на выходе означает, что функция приняла данные и находится в процессе их обработки. 0 на выходе означает, что функция обработала принятые данные и свободна для обработки следующих (тип выхода LOGIC).

6.6. Протокол RTM64 в режиме Slave

Протокол RTM64-Slave является протоколом запросного типа и независимо поддерживается на всех портах модулей СТ1СРМ10. Может быть установлена следующая скорость передачи данных:

- для СОМ-портов модуля СТ1СРМ10 — 1200, 2400, 4800, 9600, 19200, 38400 бит/с.

Настройка протокола описана в п.4.2.2.11.

При работе по протоколу RTM64-slave контроллер получает от мастера определенную команду и возвращает данные, соответствующие этой команде. Соответствие команд и данных устанавливается при разработке алгоритма в редакторе параметров.

Так же, как и для протокола Modbus-slave, команды запроса данных обрабатываются на низком уровне. Команды записи данных обрабатываются на низком, транспортном и системном уровне; после чего полученные данные оказываются записанными в буфера контроллера.

При применении данного протокола на уровне алгоритма следует постоянно поддерживать актуальное состояние буферов контроллера, устанавливать факт получения новых команд и осуществлять их обработку (см. п.6.3.3).

6.7. Семейство протоколов Proton3

6.7.1. Общие сведения

Семейство протоколов Proton3 включает в себя протоколы Proton3-slave, Proton3-IP, Proton3-GSM. Адресация, система команд и построение алгоритма одинаковы для всех протоколов семейства.

Протокол Proton3-slave является протоколом маркерного типа и поддерживается на порту СОМ1 модуля СТ2СРУ33. На практике к порту обычно подключается радиомодем. Настройка протокола описана в п.4.2.2.2.

Протокол Proton3-IP является протоколом маркерного типа и поддерживается на порту RJ45 модуля СТ1СРЕ10, установленном в 3 слоте контроллера КСА-02. Настройка протокола описана в п.4.2.2.15.

Протокол Proton3-GSM является протоколом инициативного типа и поддерживается на порту COM1 модуля СТ2СРU33. К порту должен быть подключен GSM-модем. Настройка протокола описана в п.4.2.2.2.

Замечание. Обмен данными по ресурсам 0 и 2 (см. далее п.6.7.2) поддерживается не более чем двумя потоками.

Ниже дается краткое описание протокола, необходимое прикладному программисту для создания рабочей программы (алгоритма) контроллера. Более подробное описание можно найти в документах "Описание протокола Proton3" и "Протокол Proton3-GSM".

6.7.2. Ресурсы и типы данных

Для семейства протоколов Proton3 определены следующие ресурсы:

- 0 - текущие данные,
- 1 - служебные данные,
- 2 - архивные данные.

В обычной работе обмен данными с ведущим устройством осуществляется по ресурсам 0 и 2. Данные для передачи формируются с уровня алгоритма. Единственное исключение - команда синхронизации времени. Команда синхронизации времени поступает по ресурсу 1, обрабатывается на низком и системном программном уровне. На уровень алгоритма эта команда не поступает.

Типы данных, поддерживаемые для протокола Proton3 по ресурсам 0 и 2, соответствуют буферам контроллера (п.2.1.3). Таким образом, поддерживается семь типов данных:

- входной логический,
- входной целый,
- входной с плавающей точкой,
- выходной логический,
- выходной целый,
- выходной с плавающей точкой,
- внутренний.

Адресация по типу данных совпадает с адресацией соответствующего буфера.

Замечание. Адрес 0 во входном логическом буфере для протокола Proton3 зарезервирован под команду принудительной передачи всех данных.

6.7.3. Использование в алгоритме.

При применении протокола Proton3 в любой модификации на уровне алгоритма должны решаться следующие задачи:

- поддержание буферов контроллера в актуальном состоянии, установка факта поступления команд и их обработка (п.6.3.3);
- выдача требований на передачу текущих данных;
- выдача требований на запись текущих данных в архив.

Следующие задачи решаются без обращения на уровень алгоритма:

- обработка требований на передачу текущих данных, передача текущих данных - решаются на системном и низком уровнях;
- обработка требований на запись данных в архив, запись данных в архив - решается на системном уровне;
- передача управляющей структуры состояния архива, передача архивных данных, управление состоянием архива - решается на системном и низком уровнях.

Выдача требований на передачу текущих данных и выдача требований на запись текущих данных в архив осуществляется с помощью функции алгоритма F0152. Функция имеет 7 входов и 1 выход.

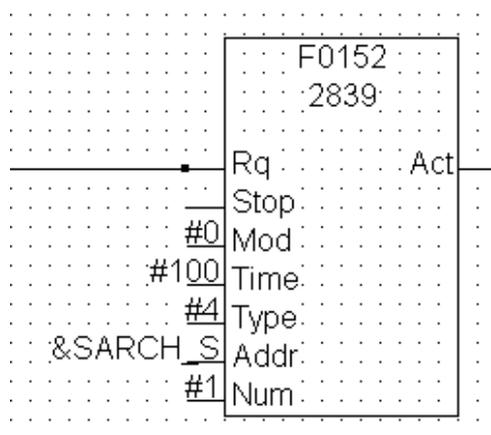


Рис. 6.7.3. Внешний вид функции F0152.

- Rq: флаг выдачи требования, требование выдается когда значение входа равно 1 (тип входа LOGIC);
- Stop: флаг запрета выдачи требования, когда значение входа равно 1, требование не выдается, даже если значение входа Rq = 1 (тип входа LOGIC);
- Mod: номер ресурса Proton3, допускаются значения 0 (требование по ресурсу 0), 2 (требование по ресурсу 2) (тип входа INTEGER);
- Time: минимальное время между двумя последовательными запросами; в ситуации когда Rq = 1, а интервал Time с момента выдачи предыдущего требования еще не истек, выдача нового требования будет отложена до окончания интервала Time (тип INTEGER);
- Type: условный номер для типа буфера, по которому выдается требование. Для ресурсов 0 и 2 поддерживаются разные типы буферов. В таблице приведено соответствие между условным номером, типом буфера и номером ресурса (тип INTEGER);

Условный номер	Описание буфера	Поддержка буфера	
		Res 0	Res 2
0	Входной логический	-	+
1	Входной целый	-	+
2	Входной с плавающей точкой	-	+
3	Внутренний	-	+
4	Выходной логический	+	+
5	Выходной целый	+	+
6	Выходной с плавающей точкой	+	+

"+" - выдача требования для данного буфера поддерживается;

"-" - выдача требования для данного буфера не поддерживается.

- Addr: адрес ячейки соответствующего буфера, для которой выставляется требование (когда Num = 1), адрес первой из ячеек соответствующего буфера, для которых выставляется требование (когда Num > 1) (тип INTEGER);
- Num: число ячеек соответствующего буфера, для которых выставляется требование. Если Num = 1, требование выставляется для ячейки, находящейся по адресу Addr. Если Num > 1, требование выставляется для ячейки, находящейся по адресу Addr и (Num - 1) последующих ячеек.

Не все комбинации Addr и Num допустимы. Функция осуществляет следующие проверки:

- фильтр превышения размера буфера - проверяется, чтобы число (Addr+Num) не превышало размера соответствующего буфера; если такое происходит, установка требования не осуществляется;
- фильтр максимального количества выставляемых требований. Для ресурса 2 максимальное число выставляемых требований ограничено в соответствии с таблицей. Для ресурса 0 максимальное число выставляемых требований ограничено размером соответствующего буфера. При превышении максимального количества выставляемых требований, установка требования не осуществляется.

Условный номер	Описание буфера	Максимальное число требований	
		Res 0	Res 2
0	Входной логический	-	512
1	Входной целый	-	32
2	Входной с плавающей точкой	-	16
3	Внутренний	-	32
4	Выходной логический	размер буфера	512
5	Выходной целый	размер буфера	32
6	Выходной с плавающей точкой	размер буфера	16

Замечание. Для выходного логического буфера (тип 4) Num = 1 устанавливает требование на одну ячейку, как и описано выше. В других случаях для логических буферов (т.е. для выходного логического буфера при Num > 1 и для входного логического буфера при любом Num) количество выставляемых требований округляется в большую сторону до числа кратному 16.

Выход функции:

Аст: - 0 - работа функции завершена, 1 - работа функции не завершена 1 (тип выхода LOGIC).

6.8. Протокол МЭК870-5-104 (IEC60870-104)

6.8.1. Общие сведения

Протокол МЭК870-5-104 является протоколом инициативного типа и поддерживается на порту RJ45 модуля ST1CPE10. Настройка протокола описана в п.4.2.2.16. Кроме этого, должна быть осуществлена настройка конвертации типов данных протокола МЭК870 к типам данных контроллера, как это описано в п.4.2.2.3.

При работе по протоколу МЭК870-5-104 (IEC60870-104) контроллер, после получения разрешения на передачу, передает данные ведущей станции в инициативном порядке. Ведущая станция может потребовать принудительной передачи всех или части данных.

Все данные (или, как их еще называют "информационные объекты") снабжаются описателем качества, который уточняет статус этих данных (актуальное/неактуальное, действительное/недействительное, блокирование изменения, замещение, переполнение).

6.8.2. Типы данных

При осуществлении инициативной передачи информации в направлении контроля контроллер поддерживает типы данных в соответствии с таблицей

Обозначение	Идентификатор	Описание	Основной размер (бит)
M_SP_NA_1	1	Одноэлементная информация	1
M_DP_NA_1	3	Двухэлементная информация	2
M_ST_NA_1	5	Информация шагового регулятора	7
M_BO_NA_1	7	Битовый строинг	32
M_ME_NA_1	9	Нормализованная величина	16
M_ME_NB_1	11	Масштабированная величина	16
M_ME_NC_1	13	Величина с плавающей точкой	32
M_IT_NA_1	15	Интегральная сумма	32
M_SP_TB_1	30	Одноэлементная информация с меткой времени	1
M_DP_TB_1	31	Двухэлементная информация с меткой времени	2
M_ST_TB_1	32	Информация шагового регулятора с меткой времени	7
M_BO_TB_1	33	Битовый строинг с меткой времени	32
M_ME_TD_1	34	Нормализованная величина с меткой времени	16
M_ME_TE_1	35	Масштабированная величина с меткой времени	16
M_ME_TF_1	36	Величина с плавающей точкой с меткой времени	32
M_IT_TB_1	37	Интегральная сумма с меткой времени	32

В зависимости от состояния флага "Поддержка метки времени" (см.п.4.2.2.16) при одиночной передаче данных будут поддерживаться типы данных с идентификаторами (1...15) или (30...37). При множественной передаче данных поддерживаются только типы данных с идентификаторами (1...15).

При приеме команд в направлении управления контроллер поддерживает следующие типы данных.

Обозначение	Идентификатор	Описание	Основной размер (бит)
C_SC_NA_1	45	Однопозиционная команда	1
C_DC_NA_1	46	Двухпозиционная команда	2
C_RC_NA_1	47	Команда пошагового регулирования	2
C_SE_NA_1	48	Команда уставки нормализованной величины	16
C_SE_NB_1	49	Команда уставки масштабированной величины	16
C_SE_NC_1	50	Команда уставки величины с плавающей точкой	32
C_BO_NA_1	51	Битовый стринг	32

Контроллер подтверждает получение команд. Формирование и отправка подтверждения команд осуществляется на низком уровне.

Контроллер также поддерживает следующие команды в направлении управления:

Обозначение	Идентификатор	Описание
C_IC_NA_1	100	Команда опроса
C_CI_NA_1	101	Команда опроса счетчика
C_CS_NA_1	103	Команда синхронизации времени

В ответ на получение данных команд контроллер возвращает запрашиваемые данные. Формирование и отправка ответа осуществляется на низком уровне.

6.8.3. Размещение данных МЭК870 в буферах контроллера

При инициализации контроллер резервирует в своих буферах место для команд и данных протокола МЭК870. Резервирование осуществляется в соответствии с настройками протокола МЭК870 (п.4.2.2.3). В таблице показывается расположение данных МЭК870 в

буферах контроллера, а также количество элементов буфера, которое занимает каждый тип данных МЭК870.

Идентификатор	Описание	Буфер контроллера	Размер
45	Однопозиционная команда	Входной логический	1
46	Двухпозиционная команда	Входной логический	2
47	Команда пошагового регулирования	Входной логический	2
51	Битовый стринг	Входной логический	32
48	Команда уставки нормализованной величины	Входной целый	1
49	Команда уставки масштабированной величины	Входной целый	1
50	Команда уставки величины с плавающей точкой	Входной с плавающей точкой	1
1,30	Одноэлементная информация	Выходной логический	1
3,31	Двухэлементная информация	Выходной логический	2
5,32	Информация шагового регулятора	Выходной логический	8
7,33	Битовый стринг	Выходной логический	32
9,34	Нормализованная величина	Выходной целый	1
11,35	Масштабированная величина	Выходной целый	1
13,36	Величина с плавающей точкой	Выходной с плавающей точкой	1
15,37	Интегральная сумма	Выходной целый	2

6.8.4. Обращение к данным МЭК870 из алгоритма

Обращение к данным МЭК870 из алгоритма возможно с использованием функций алгоритма типа "Приемник" или "Передатчик". Однако, во первых, это неудобно, и, во вторых, требует расчета адреса данных МЭК870 в буфере.

Для более удобной работы с информационными объектами МЭК870 существуют специальные функции, использующие адресацию протокола МЭК870, а также одновременно осуществляющие запись описателя качества информационного объекта:

F0176 - передатчик одноэлементной информации МЭК870;

F0177 - передатчик двухэлементной информации МЭК870;

F0178 - передатчик информации шагового регулятора МЭК870;

F0179 - передатчик битового строга МЭК870;

F0180 - передатчик нормализованной величины МЭК870;

F0181 - передатчик масштабированной величины МЭК870;

F0182 - передатчик величины с плавающей точкой МЭК870;

F0183 - передатчик интегральной суммы МЭК870;

F0184 - приемник однопозиционной команды без сброса МЭК870;

F0185 - приемник однопозиционной команды со сбросом МЭК870;

F0186 - приемник двухпозиционной команды со сбросом МЭК870;

F0187 - приемник команды пошагового регулирования МЭК870;

F0188 - приемник битового строга без сброса МЭК870;

F0189 - приемник битового строга со сбросом МЭК870;

F0190 - приемник уставки нормализованной величины МЭК870;

F0191 - приемник уставки масштабированной величины МЭК870;

F0192 - приемник уставки величины с плавающей точкой МЭК870;

F0193 - синхронизация времени МЭК870;

F0194 - преобразование типа float IEEE754 к нормализованной величине МЭК870;

F0195 - преобразование типа нормализованной величины МЭК870 к float IEEE754.

.

ПРИЛОЖЕНИЕ

1. Словарь:

Система программирования — программная система, предназначенная для разработки программ на конкретном языке программирования. Система программирования предоставляет пользователю специальные средства разработки программ: компилятор, редактор текстов программ, библиотеки стандартных подпрограмм, программную документацию, отладчик и др.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления периферией. В общем случае контроллер КСА-02 состоит из главного процессора и станции удаленного ввода-вывода.

АРМ — автоматизированное рабочее место — в АСУ ТП это программно-технический комплекс, зачастую на основе персонального компьютера, позволяющий персоналу получать доступ в удобной форме к текущим и архивным данным технологического процесса, производить управление технологическим оборудованием, настройку и диагностику автоматизированной системы управления.

Модуль — унифицированный легко заменяемый функциональный узел (блок, устройство, схема, элемент и т.п.) ЭВМ. Модуль контроллера КСА-02 — это функционально завершенная составная часть контроллера, выполняющая заданные функции.

Корпус — это металлический ящик с разъёмами для модулей. Существуют корпуса на 16, 7 и 4 разъёма (соответственно, на 14, 5 и 2 модуля ввода/вывода).

Корзина — это корпус, содержащий в себе набор модулей, подобранный для выполнения определённой задачи. Если в корзине присутствует процессорный модуль СТ2CPU33, то

она является главным процессором контроллера; если процессорный модуль типа СТ1RPU33 — то корзина является станцией удалённого ввода-вывода (корзиной СУВД).

SCADA-система (англ. Supervisory Control And Data Acquisition — диспетчерское управление и сбор данных) — это компьютерная система для сбора и анализа данных в режиме реального времени, управляемая человеком.

Ядро — это программа, загружаемая из Flash памяти в оперативную память внутри модуля центрального процессора СТ2CPU33 и управляющая всеми процессами в контроллере КСА-02. В тексте может употребляться как «ядро среды исполнения» или «ядро системы «Scorpio». Кроме того, системное управляющее ПО любого модуля контроллера так же может именоваться ядром, но в этом случае должна быть указана принадлежность ядра конкретному модулю контроллера.

Алгоритм — это последовательность действий (операций) и правил их выполнения или команд, предназначенных для решения определенной задачи или группы задач. Алгоритм рабочей программы контроллера КСА-02 — это «приложение», которое должен выполнять контроллер. Алгоритм представляет собой массив данных, записанный в энергонезависимую память контроллера.

Буфер — область памяти или отдельное ЗУ, предназначенные для временного хранения данных в процессе выполнения операций с целью обеспечения оптимальных условий обмена между устройствами ЭВМ, имеющими различное быстродействие (например оперативной памятью и устройствами ввода-вывода).

Протокол передачи информации — совокупность правил, регламентирующих формат и процедуры обмена информацией между двумя или несколькими независимыми устройствами или процессами.

Проект алгоритма — совокупность файлов, необходимых для создания и отладки алгоритма рабочей программы.

Диаграмма алгоритма — это графическое условное представление логической взаимосвязи входных, выходных и внутренних данных. Диаграмма состоит из прямоугольников — «чёрных ящиков», между которыми проведены линии связи.

Блок — графическое изображение функции, программа, выполняющая определенную часть решаемой задачи. Блок описывается алгоритмом, который определяет процессы обработки, хранения и передачи данных.

Функция — это подпрограмма, написанная на языке Assembler, выполняющая различные арифметико-логические операции. Каждая функция выполняет какую-либо арифметико-логическую операцию над входными данными, а результат выполнения этой операции является для данной функции выходными данными.

Промежуточные данные функции — это данные, которые наследуются ею из предыдущего цикла алгоритма, например, если функция накапливает и подсчитывает эти данные за определённый промежуток времени, в течение которого цикл неоднократно успевает повториться (или если она должна из цикла в цикл удерживать какое-то значение до тех пор, пока в алгоритм не поступит команда его изменить).

Атрибуты — это промежуточные данные функции, начальные значения которых можно задавать из графического строителя.

Составная функция — это единое графическое оформление повторяющегося фрагмента алгоритма (либо не повторяющегося, но представляющего собой логически целостный вычислительный блок). Некоторое количество функций объединяется в одну составную и заключается в единый блок.

Тег — это метка, используемая в алгоритме для указания на местоположение данных и устанавливаемая на вход или выход функции.

Литера — это устойчивое неизменное значение, назначенное входу или выходу блока и присущее только ему.

Контакт — это объект диаграммы, который используется для графического изображения передачи данных. Входной контакт поставляет данные в функцию из буфера на обработку, а выходной — принимает из функции уже обработанные данные и сохраняет их в буфере.

Резервирование — метод повышения надежности объекта введением дополнительных элементов и функциональных возможностей сверх минимально необходимых для нормального выполнения объектом заданных функций. В этом случае отказ наступает только после отказа основного элемента и всех резервных элементов.

Устройства удаленного ввода/вывода — устройства, предназначенные для сбора данных, регулирования параметров технологического процесса и диспетчерского управления территориально разнесенных объектов. Применяются при автоматизации технологических процессов в различных областях промышленности.

Инверсия (лат. *inversio* — переворачивание, перестановка) — процесс и результат нарушения нормального порядка и последовательности элементов, их перестановка или замена вплоть до противоположных.

Станция удаленного ввода-вывода (СУВД) — удаленная корзина контроллера КСА-02, с установленным в своем составе процессорным модулем СТ1RPU33, не имеющая своего

алгоритма управления технологическим процессом и лишь выполняющая сбор и передачу сигналов от датчиков и сигнализаторов главному процессору контроллера.

Компилятор — осуществляет перевод всей написанной программы в эквивалентную ей программу-результат, которую можно будет записать в контроллер и запустить в эксплуатацию. Кроме того, компилятор обнаруживает и позволяет просматривать возможные ошибки построения алгоритма.

Калибровка — процесс установления зависимости между показаниями прибора и размером измеряемой (входной) величины. Под калибровкой часто понимают процесс подстройки показаний выходной величины или индикации измерительного инструмента до достижения согласования между эталонной величиной на входе и результатом на выходе (с учётом оговоренной точности).

Драйвер — это часть ядра, отвечающая за передачу данных между устройствами: он открывает порт (RS232 или RS485), формирует данные в телеграммы и отправляет их другому устройству, в случае ошибки повторяет передачу, а в случае успешного обмена закрывает порт; также драйвер принимает данные от других устройств.

Экспоненциальная запись — представление действительных чисел в виде мантиссы и порядка.

- $N = M \times n^p$, где
- N — записываемое число
- M — мантисса (дробная часть десятичного числа)
- n — основание степени
- p (целое) — порядок

MAC-адрес — уникальный физический адрес, идентифицирующий сетевой узел. MAC-адрес присваивается каждому Ethernet порту интерфейсных модулей СТ1СРЕ33 на

главном процессоре и жестко прописывается производителем. При этом первые три байта MAC-адреса определяют самого производителя.

Поверенный прибор — это прибор, показания которого считаются достоверными, эталонными. С ним сравнивают показания других приборов.

2. Зарезервированные символы для формирования литеры:

Так обрабатываются компилятором зарезервированные обозначения при применении литерной привязки (приведены примеры и расшифровка обозначений).

Время: <u>#t20s</u>	
t	Символ, указывающий, что данная литера является временем;
20	Количество единиц времени;
s	Единицы времени в секундах (возможные значения: s, m, h, sec, min, hour).
Примечание: при компиляции время пересчитывается в тики контроллера (получится 400).	
Адрес: <u>#a12.90</u>	
a	Символ, указывающий, что данная литера является адресом;
12	Старшая часть адреса (возможные значения: от 0 до 255);
.	Символ разделения старшей и младшей части адреса;
90	Младшая часть адреса (возможные значения: от 0 до 255).
Примечание: при компиляции время 16-ти битное число, где старшая часть умножается на 256 и прибавляется младшая часть (получится 3162).	
Ввод двоичных чисел: <u>#b010101</u>	
b	Символ, указывающий, что данная литера является двоичным числом (допустимые комбинации: b, B, 0b, 0B);
10101	Двоичное число (число по основанию 2), допускаются цифры «0» и «1».
Ввод шестнадцатеричных чисел: <u>#h23a4</u>	
h	Символ, указывающий, что данная литера является шестнадцатеричным числом (допустимые комбинации: h, H, 0x, 0X);
23a4	Шестнадцатеричное число (число по основанию 16), допускаются все цифры и буквы: «a»...«f», «A»....«F».

3. Описание сообщений компилятора об ошибках

1. Сообщения компилятора	
Ошибка	Содержание сообщения
—	Адрес N <число> тега вход составной функции превышает количество входов N <количество>
—	Название проекта превышает 31 символ
—	Название проекта составной функции превышает 12 символов
—	Ошибка заголовка описания составной функции, доступ атрибута N <номер>.
—	Ошибка заголовка описания составной функции, значение атрибута N <номер>.
—	Ошибка заголовка описания составной функции, количество атрибутов (входов; выходов)
—	Ошибка заголовка описания составной функции, описание атрибута (входа, выхода) N <номер>.
—	Ошибка заголовка описания составной функции, тип атрибута (входа, выхода) N <номер>.
—	По одному входу 1 составной функции привязано две или более тегов
—	Путь директории проекта превышает 1024 символа
—	Размер программы превышает 130816 слов.
—	Тип входа 1 составной функции не соответствует типу привязанного тега
—	Файл <имя файла>.fsc (вставка составных функций) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (модуль) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (неинициализированные входы) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (построение) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (расстановка адресов Flash) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (расстановка флагов) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc (служебные функции) содержит N <количество> ошибок.

—	Файл <имя файла>.fsc (создание связей) содержит N <количество> ошибок.
—	Файл <имя файла>.fsc откомпилирован успешно. Размер выходного файла 999 слов(а). — Примечание: является сообщением об успешной компиляции и не относится к сообщениям об ошибках
2. Ошибки и предупреждения компилятора.	
Ошибка	Содержание сообщения
Системная ошибка (не связана с алгоритмом)	Не могу найти название проекта
	Не могу открыть файл <имя файла>.fsc
	Не могу открыть файл «Scorpio».fbd
	Не могу прочитать файл «Scorpio».shd
	Не могу создать файл <имя файла>.err (.rec,.sc)
	Ошибка строения «Scorpio».fbd
	Ошибка строения «Scorpio».fbd, функция F0000, атрибут (вход, выход) номер N
	Ошибка строения «Scorpio».fbd, функция F0000, количество атрибутов (входов, выходов) функции.
	Ошибка строения «Scorpio».fbd, функция F0000, комментарий функции.
	Ошибка строения «Scorpio».fbd, функция F0000, наличие авт. размножаемого входа у функции.
	Ошибка строения секции [def] файла <имя файла>.fsc
	Ошибка строения файла <пример>. Некорректная версия.ssc файла
	Ошибка строения файла <имя файла>.fsc Не могу найти секцию [brk]
	Ошибка строения файла <имя файла>.fsc Не могу найти секцию [connects]
	Ошибка строения файла <имя файла>.fsc Не могу найти секцию [def]
	Ошибка строения файла <имя файла>.fsc Не могу найти секцию [fun]
Ошибка строения файла <имя файла>.fsc Не могу найти секцию [lin]	
Ошибка строения файла <имя файла>.fsc Не могу найти	

	секцию [project]
	Ошибка строения файла <имя файла>.fsc Не могу найти секцию [var]
	Ошибка строения файла <имя файла>.fsc Некорректная версия.fsc файла
	Ошибка строения файла <имя файла>.fsc Некорректная секция [project]
	Секция [brk] файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [connects] файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [def] файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [fun] файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [lin] файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [sub](подпрограммы) файла <имя файла>.fsc содержит N <количество> ошибок.
	Секция [var] файла <имя файла>.fsc содержит N <количество> ошибок.
	В файле «Scorpio».fbd отсутствует функция <имя функции> повторитель <тип буфера>
	В файле «Scorpio».fbd отсутствует функция <имя функции> приёмник <тип буфера>
	В файле «Scorpio».fbd отсутствует функция <имя функции> передатчик <тип буфера>
	В файле «Scorpio».fbd отсутствует функция <имя функции> приёмник со сбросом <тип буфера>
	В файле «Scorpio».fbd отсутствует функция <имя функции> приёмник из выходного буфера <тип буфера>
	Количество соединений меньше двух у линии секции [lin]
	Не могу найти название функции из секции [fun] в базе данных функций
	Не могу найти подпрограмму из секции [fun] в списке подпрограмм
	Не могу найти функцию, на которую указывает связь
	Не могу привязать тег <имя тега> к контакту секции [connects]

	— неизвестный тип контакта
	Не могу расшифровать адрес тега <имя тега> в секции [var]
	Не могу расшифровать значение по умолчанию тега <имя тега> в секции [var]
	Не могу расшифровать идентификатор контакта в секции [connects]
	Не могу расшифровать идентификатор линии секции [lin]
	Не могу расшифровать идентификатор разрыва секции [brk]
	Не могу расшифровать идентификатор составной функции в секции [sub]
	Не могу расшифровать идентификатор тега в секции [var]
	Не могу расшифровать идентификатор функции в секции [fun]
	Не могу расшифровать инверсную маску функции в секции [fun]
	Не могу расшифровать координату X (или Y) контакта в секции [connects]
	Не могу расшифровать координату X (или Y) функции в секции [fun]
	Не могу расшифровать координату линии секции [lin]
	Не могу расшифровать название составной функции в секции [sub]
	Не могу расшифровать название тега в секции [var]
	Не могу расшифровать название функции в секции [fun]
	Не могу расшифровать направление линии секции [lin]
	Не могу расшифровать тип инверсии контакта в секции [connects]
	Не могу расшифровать тип контакта в секции [connects]
	Не могу расшифровать тип составной функции в секции [sub]
	Не могу расшифровать тип тега в секции [var]
	Не нахожу ответной ссылки при анализе связи
	Не смогли найти объект с заданным идентификатором, на который указывает связь
	Невозможное число у контакта в секции [connects]
	Недопустимое значение инверсии у контакта в секции [connects]
	Неправильный тип тега в секции [var]

	Номер входа (<i>или выхода</i>) функции (1), на который указывает связь, превышает максимально возможный
	Ошибка в строении связи (возможно, замыкается сама на себя)
	Ошибка при расшифровке CRC составной функции <имя> в секции [sub]
	Ошибка при расшифровке количества атрибутов (<i>входов, выходов</i>) составной функции <имя> в секции [sub]
	Ошибка при расшифровке описания составной функции <имя> в секции [sub]
Ошибки редактора параметров	Интерфейсный модуль 1, интерфейс 1, протокол не установлен
	Недопустимый аргумент у N-го <номер> интерфейса на N-ом <номер> модуле
	Неправильная последовательность определения интерфейсных модулей
	Неправильное количество байт для команды 5E4E протокола RTM64 slave прием (<i>передача</i>)
	Неправильное количество блоков для протокола RTM64 slave
	Неправильное суммарное число аргументов для N-го <порядковый номер> интерфейсного модуля
	Неправильное число аргументов для интерфейсного модуля 1, интерфейс 1, протокол Modbus slave (<i>возможная расшифровка: Modbus TCP, Proton 2 Slave, RTM64 Slave, Modbus Master, MKSA Master</i>)
	Неправильное число аргументов для интерфейсного модуля 1, интерфейс 1, протокол отсутствует
	Неправильное число аргументов для протокола RTM64 slave
	Неправильный тип платы по N-му <порядковый номер> интерфейвному модулю
	Превышено количество аргументов для N-го <порядковый номер> интерфейсного модуля
	Превышено суммарное число аргументов для интерфейсных модулей
	Присутствуют аргументы для невозможного номера интерфейсного модуля
	Размер области дублирования превышает размер буфера внутренний
	Размер области дублирования превышает размер буфера входной <тип буфера>

	Суммарный размер буферов превышает 8192 слов
Ошибки рисования диаграммы	Адрес у функции превышает размер внутреннего буфера
	Адрес у функции превышает размер входного буфера <тип буфера>
	Адрес у функции превышает размер выходного буфера <тип буфера>
	Адресный вход функции-передатчика F0013 не инициализирован никаким значением. (Возможные разновидности: функции F0126 (внутренний буфер), функции F0127 (внутренний буфер), функции F0127 (входной целый буфер), функции 0097 (внутренний буфер), функции F0098 (выходной целый буфер), функции F0150 (внутренний буфер))
	Атрибут <имя атрибута> Flash address может быть соединен только с Flash функцией, инициализируя в ней атрибут Flash address.
	Атрибут <имя атрибута> Flash address может быть соединен только с одной функцией, инициализируя в ней атрибут Flash address.
	В один тег привязать осуществляют запись несколько объектов на выходе N-й <номер> функции в секции [fun]
	В один тег <имя тега> осуществляют запись несколько объектов
	В связь осуществляют запись два объекта (три объекта, четыре объекта, пять объектов, несколько объектов)
	В составной функции не инициализирован адрес у функции Flash
	В файле проекта количество функций F0101 статус резервирования больше одной
	В файле проекта количество функций F0102, F0103 больше 64
	В файле проекта отсутствуют функции
	Для тега <имя тега> для контакта невозможна операция взятия адреса
	Для тега <имя тега> на атрибуте N-ой <номер> функции невозможна операция взятия адреса
Для тега <имя тега> на входе N-ой <номер> функции невозможна операция взятия адреса	
Значение по умолчанию для тега <имя тега> в секции [var] не соответствует его типу	

Именованную связь <имя> читают, но не пишут
Инверсия контакта применена к недопустимому типу в секции [connects]
К входу 1 присоединен тег <имя тега> выход составной функции, в который не осуществляется запись
К входу 1 присоединена именованная связь <имя>, в которую не осуществляется запись
К входу контакта присоединена именованная связь <имя>, в которую не осуществляется запись
Контакт читает выход подпрограммы <имя>, в который не осуществляется запись
Контакт читает именованную связь <имя>, в которую не осуществляется запись
Контакты образуют кольцо — замыкаются сами на себя
На вход N-й <номер> функции осуществляется запись с нескольких объектов
Невозможно применить инверсную маску для составной функции для выбранного метода компиляции
Невозможно произвести запись во вход N <номер> типа резерв
Не могу найти тег ID=00000 для объекта на атрибуте (входе, выходе) N-й <номер> функции в секции [fun]
Не могу найти тег для контакта [connects]
Не могу привязать данные к контакту со сбросом секции [connects]
Не могу привязать значение 123 к резервному входу N-й <номер> функции в секции [fun]
Не могу привязать тег <имя тега> (взятие адреса) к атрибуту (или входу) пл. точка N-й <номер> функции в секции [fun]
Не могу привязать тег <имя тега> (взятие адреса) к логическому атрибуту (или логическому входу) N-й <номер> функции в секции [fun]
Не могу привязать тег <имя тега> к атрибуту пл. точка N-й <номер> функции в секции [fun]
Не могу привязать тег <имя тега> к входу пл. точка N-й <номер> функции в секции [fun]
Не могу привязать тег <имя тега> к контакту секции [connects]
Не могу привязать тег <имя тега> к логическому атрибуту (или логическому входу) N-й <номер> функции в секции [fun]

	Не могу привязать тег <i><имя тега></i> к резервному входу N-й <i><номер></i> функции в секции [fun]
	Не могу привязать тег <i><имя тега></i> к целому атрибуту (или целому входу) N-й <i><номер></i> функции в секции [fun]
	Не могу привязать тег <i><имя тега></i> на выходе <i><тип выхода></i> N-ой <i><номер></i> функции в секции [fun]
	Не могу разорвать кольцо из функций — возможно необходима функция об. связи
	Не привязан тег для контакта [connects]
	Невозможно привязать значение 123 к выходу N-й <i><номер></i> функции в секции [fun]
	Невозможно привязать тег <i><имя тега></i> (взятие адреса) к выходу N-й <i><номер></i> функции в секции [fun]
	Недопустимая команда у функции F0102 — передача по Modbus
	Недопустимая литера на атрибуте (или входе) N-й <i><номер></i> функции
	Недопустимая литера у контакта секции [connects]
	Недопустимая операция — запись атрибута <i><имя атрибута></i> в выход составной функции
	Недопустимая операция запись в литеру в контакте в секции [connects]
	Недопустимая операция запись в тег (взятие адреса) в контакте в секции [connects]
	Недопустимое значение 123 для атрибута (или входа) пл. точка N-й <i><номер></i> функции в секции [fun]
	Недопустимое значение 123 для логического атрибута (или логического входа) N-й <i><номер></i> функции в секции [fun]
	Недопустимое значение 123.00 для целого атрибута N-й <i><номер></i> функции в секции [fun]
	Недопустимое значение <i><пример></i> для целого входа N-й <i><номер></i> функции в секции [fun]
	Недопустимое объединение атрибута <i><имя атрибута></i> составной функции с выходом <i><пример></i>
	Некорректный адрес у функции Flash
	Несоответствие CRC составной функции <i><имя></i> исходному коду в секции [sub]
	Несоответствие версии составной функции <i><имя></i> исходному

	коду в секции [sub]
	Несоответствие количества атрибутов (<i>входов, выходов</i>) составной функции <i><имя></i> исходному коду в секции [sub]
	Несоответствие типа данных входа составной функции <i><имя></i> , объединенного связью с выходом составной функции
	Несоответствие типа данных выхода 1, объединенного связью с входом 1 (<i>или с выходом составной функции</i>)
	Нет возможности расставить адреса у функций Flash
	Ошибка при компиляции составной функции <i><имя></i> в секции [sub]
	По одному адресу в выходной буфер <i><тип буфера></i> осуществляют запись две или более функций
	По одному адресу во внутренний буфер осуществляют запись две или более функций
	По одному адресу во входной буфер <i><тип буфера></i> осуществляют запись две или более функций
	По одному адресу инициализированы две или более функций Flash
	Составная функция <i><имя></i> имеет вхождение сама в себя
	Суммарное количество функций превышает 32752
Ошибки базы данных тегов	Адрес 1 тега атрибут составной функции превышает количество атрибутов (1)
	Адрес 1 тега вход составной функции превышает количество входов (1)
	Недопустимый тип тега <i><имя тега></i> для использования в основной программе
	Недопустимый тип тега <i><имя тега></i> для использования внутри составной функции
	Ошибка в названии имени тега <i><имя тега></i>
	К одному атрибуту (<i>или входу</i>) 1 составной функции привязано два или более тегов
	Тип входа 1 составной функции не соответствует типу привязанного тега
В связи отсутствуют объекты, осуществляющие запись	По одному адресу инициализированы две или более функций Flash — изменение адреса

Именованная связь подключена к выходу одной из функций, но не подключена ко входу другой.	В именованную связь <имя> пишут, но не читают
Выходную переменную читают, но не пишут	Выходную переменную <имя> читают, но не пишут
Предупреждение о неиспользуемых тегах	Тег <имя тега> не используется
Предупреждение о неиспользуемых контактах	Контакт не используется — игнорирую
Связь не присоединена к объекту	Связь не присоединена к объекту
Неэффективное значение на входе составной функции, связанное с выходом; возможна ошибка	Неэффективное значение на входе 1 составной функции, связанное с выходом; возможна ошибка
Нулевое значение на адресном входе функции-передатчика, возможно, не инициализирована	Нулевое значение на адресном входе функции-передатчика (или функции-приёмника), возможно, не инициализирована

4. Архитектура системы «Scorpio»

