СЕРИЯ МОДУЛЕЙ К-3ХХХ

КОММУНИКАЦИОННО – ЛОГИЧЕСКИЙ КОНТРОЛЛЕР К-3102

РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ

СГВП2.390.016 РЭ

2006

СОДЕРЖАНИЕ

Стр.

Настоящее руководство по эксплуатации предназначено для ознакомления с устройством контроллера типа К-3102 (далее в тексте - контроллера), правилами эксплуатации, транспортирования и хранения с целью поддержания его в рабочем состоянии в течение срока эксплуатации.

1 НАЗНАЧЕНИЕ

1.1 Контроллер предназначен для сбора и обработки информации, конвертирования протоколов связи при решении задач автоматизации.

1.2 Контроллер применяется в составе КТС-2000 ТУ4371-006-12221545-01.

1.3 Контроллер позволяет решать следующие задачи в любом сочетании или одновременно:

- сбор и обработка первичной информации;
- сбор и обработка данных от средств автоматизации "третьих фирм", например от интеллектуальных датчиков, приборов и т.п.;
- сопряжение с аппаратурой разных уровней по интерфейсам Ethernet (протокол Modbus TCP), RS-485/232 в протоколе Modbus RTU.

2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1 Основные характеристики контроллера приведены в таблице 1:

Таблица 1

Характеристика	Значение
Количество последовательных портов RS-485	3
Количество последовательных портов RS-485/RS-232	1
Количество Ethernet портов	1
Встроенный коммутатор (switch)	да
Поддерживаемые протоколы: порт RS-485/232 порт Ethernet	 Modbus RTU Modbus TCP
Скорость обмена по RS-485 (бит/с)	1200,2400,4800,9600. 19200,28800,38400, 57600,76800,115200, 128000,153600,230400
Скорость обмена по Ethernet (Мбит/с)	10/100
Максимальное количество поддерживаемых TCP/IP соединений	до 32*
Напряжение питания	24 В
Потребляемый ток	не более 0,2 А
Степень защиты	IP20
Размер памяти программ	65535 байт
Размер памяти данных	4096 байт

* - зависит от версии программного обеспечения

- Функциональные кнопки и переключатели:

R (Reset) – перезапуск контроллера и восстановление последних сохраненных параметров;

DEF (Default) – установка настроек по умолчанию при перезапуске контроллера;

ON-485 – режим работы последовательного порта 4 (RS-485/ RS-232).

- Светодиоды для индикации:

  - LED – программируемые функции (красный/зеленый);

  - PWR – индикатор питания (зеленый);

  - активность портов 1 – 4 (красные).

2.2 Время готовности контроллера с момента подачи питания, с учетом времени на автоматический контроль исправности - не более 10 сек.

2.3 Контроллер должен сохранять работоспособность при следующих параметрах линий связи интерфейса RS-485:

1) длина, не более 1200 м;

2) емкость, не более 50 нФ;

3) сопротивление, не более 50 Ом;

4) сопротивления изоляции, не менее 50 кОм.

Тип линии – двухпроводная экранированная витая пара.

2.4 Контроллер обеспечивает хранение в энергонезависимом ПЗУ заданных настроек при исчезновении напряжения в питающей сети.

2.5 Контроллер обеспечивает связь с АВУ по интерфейсам Ethernet (протокол Modbus TCP), RS-485(RS-232) в протоколе Modbus RTU.

2.6 Диапазон напряжения питания постоянного тока – 18...36 В, номинальное напряжение питания – 24 В.

2.7 Потребляемая мощность - не более 5 Вт.

2.8 Режим работы – непрерывный, длительный.

2.9 Установка контроллера производится на DIN рельс.

2.10 Электрическая изоляция между соединенными группами входных и выходных проводников и контактом заземления в нормальных климатических

условиях выдерживает в течение 1 мин синусоидальное переменное напряжение 0,5 кВ частотой 45-65 Гц.

2.11 Электрическое сопротивление изоляции между соединенными группами входных и выходных проводников и контактом заземления в нормальных климатических условиях не менее 20 МОм.

2.12 Контроллер предназначен для эксплуатации в диапазоне температур от минус 40 до 70 ^{0}C (группа исполнения С2 по ГОСТ 12997–84) при относительной влажности воздуха до 93 % при температуре окружающей среды 40°С без конденсации влаги.

2.13 Контроллер устойчив к воздействию синусоидальной вибрации с частотой от 10 до 150 Гц и величиной ускорения 0,5 g.

2.14 Контроллер прочен к воздействию вибрации с частотой от 10 до 150 Гц и величиной ускорения 1 g.

2.15 Степень защиты контроллера от проникновения воды, пыли и посторонних твердых частиц – IP 20 по ГОСТ 14254.

2.16 Габаритные размеры (длина × высота × ширина),
не более 140 х 80 х 40 мм.

2.17 Масса, не более 0,4 кг.

2.18 Срок службы контроллера - не менее 10 лет.

Внешний вид контроллера должен соответствовать сборочно-габаритному чертежу, приведенному в Приложении Б.

3 КОМПЛЕКТНОСТЬ

3.1 Комплектность поставки контроллера должна соответствовать таблице 2

Таблица 2

Наименование	Кол-во	Примечание
Контроллер К-3102	1	–
Паспорт СГВП2.390.016 ПС	1	-
Руководство по эксплуатации СГВП2.390.016 РЭ	1	На партию изделий, направляемых в один адрес, но не более чем на 10
Розетка WAGO	5	Ответные части

4 УСТРОЙСТВО И РАБОТА ИЗДЕЛИЯ

4.1 Конструкция контроллера

4.1.1 Внешний вид контроллера показан на рисунке 1. Контроллер выполнен в алюминиевом корпусе. Внутри корпуса установлена печатная плата с размещенными на ней радиоэлементами. Корпус имеет крепления для установки модуля на стандартный 35 мм DIN-рельс.

Рисунок 1. Внешний вид контроллера К-3102

4.2 На передней панели контроллера расположены разъемы для подключения интерфейсов Ethernet, RS-485/232 (1–4), питания (24В), функциональные кнопки и переключатели, а также светодиодные индикаторы, показывающие активность портов 1 – 4.

4.3 Последовательные порты 1 – 3 реализуют интерфейсы RS-485, порт 4 интерфейс RS-485/232, со скоростью приема-передачи данных до 304 кБод/с.

4.4 Принцип действия контроллера основан на приеме и передаче данных по интерфейсам Ethernet, RS-485/232, их логической обработке, отображении и выводе.

4.5 На многослойной плате расположен микроконтроллер PIC. Микроконтроллер управляет чтением/записью данных в flash памяти программ и вводом-выводом данных по интерфейсам Ethernet, RS-485/232.

4.6 Перемычки портов 1–4 в положении ON подключают резистор-терминатор 120 Ом между линиями A и B интерфейса RS-485. Включение перемычки обязательно, если контроллер установлен в начале или конце линии интерфейса.

4.7 Настройка контроллера производится в соотвестсвии с инструкцией по настройке и программированию (Приложение А).

4.8 Гальваническая развязка питания и интерфейсов RS-485 обеспечивается конструкцией за счет применения преобразователей напряжения питания DC/DC.

5 УКАЗАНИЯ МЕР БЕЗОПАСНОСТИ

5.1 Контроллер по способу защиты человека от поражения электрическим током относится к III классу по ГОСТ 12.2.007.0.

5.2 К работе с контроллером допускаются лица, изучившие настоящие руководство по эксплуатации, а также прошедшие инструктаж по технике безопасности.

6 ПОДГОТОВКА К РАБОТЕ

6.1 При монтаже контроллера следует соблюдать:

1) "Правила устройства электроустановок "(ПУЭ)

2) "Правила техники безопасности при эксплуатации электроустановок потребителей" (ПТБ);

3) "Правила технической эксплуатации электроустановок потребителей" (ПЭЭП)

4) Требования настоящего руководства по эксплуатации;

5) Требования эксплуатационной документации на изделия, в составе которых применяется контроллер.

6.2 Перед установкой контроллер должен быть осмотрен. Особое внимание необходимо обратить на:

- отсутствие повреждений корпуса;

- отсутствие повреждений разъемов;

- наличие всех крепежных элементов;

6.3 Контроллер устанавливается вне взрывоопасных зон в местах, обеспечивающих защиту от воздействия прямого солнечного излучения, кислотных, щелочных и других агрессивных примесей, токопроводящей пыли и механических повреждений.

6.4 Монтаж контролера проводить в следующей последовательности:

- определить место установки;

- установить контроллер на DIN рельс в соответствии с приложением Б.

- к месту установки подвести проводники и кабели необходимой длины;

- подключить проводники к контактам рязъемов и контактам заземления в соответствии со схемами подключения.

Проводники должны подключаться без натяжения.

7 ПОРЯДОК РАБОТЫ

7.1 Подать на контроллер напряжение питания.

7.2 По включению питания происходит процесс диагностики электронных компонентов, микроконтроллера, а также чтение текущей конфигурации.

7.3 Подключить интерфейсные кабеля к портам, используемым программой пользователя.

7.4 Провести необходимые настройки в соответствии с инструкцией по настройке и программированию (Приложение А).

8 ТЕХНИЧЕКОЕ ОБСЛУЖИВАНИЕ

8.1 Прием контроллера в эксплуатацию после монтажа (установки) и выполнение мероприятий по технике безопасности должны производиться в соответствии с "Правилами технической эксплуатации электроустановок потребителей" (ПЭЭП) и настоящим руководством по эксплуатации.

8.2 При эксплуатации контроллера необходимо поддерживать его работоспособность в соответствии с требованиями настоящего руководства по

эксплуатации и проводить его техническое обслуживание в объеме проведения профилактических работ.

Рекомендуется один раз в три месяца проводить следующий объем профилактических работ:

- визуальный осмотр - проверить крепление контроллерая, кабелей и разъемов, состояние маркировки, отсутствие механических повреждений;
- удаление загрязнений, пыли и влаги: скопление пыли удаляйте продувкой сухим воздухом и мягкой тканью, влагу – сухой мягкой тканью;

Проверка крепления проводников к контактам разъемов и удаление загрязнений, пыли и влаги проводится при необходимости на отключенном контроллере.

8.3 Организацию и контроль за проведением работ по техническому обслуживанию контроллера осуществляет инженерно-технический персонал, обслуживающий технические средства эксплуатирующей организации.

8.4 При проведении технического обслуживания соблюдайте меры безопасности, указанные в разделе 5.

9 ВОЗМОЖНЫЕ НЕИСПРАВНОСТИ И МЕТОДЫ ИХ УСТРАНЕНИЯ

9.1 В случае неисправности контроллера в первую очередь отключите напряжение питания.

9.2 Краткий перечень возможных неисправностей и способы их устранения приведены в таблице 3.

Таблица 3

Наименование неисправности, внешние проявления и дополнительные признаки	Вероятная причина	Способ устранения
Контроллер не работает Отсутствует индикация	Отсутствие напряжения питания	Проверить (подать) напряжение питания
Нет передачи данных	Обрыв линии интерфейса связи RS-485/232	Проверить целостность и отсутствие разрывов линии интерфейса связи RS-485/232

9.3 При возникновении прочих более сложных неисправностей их устранение может проводиться только на предприятии-изготовителе подготовленными специалистами.

10 ТРАНСПОРТИРОВАНИЕ И ХРАНЕНИЕ

10.1 Контроллеры следует транспортировать любым видом транспорта в крытых транспортных средствах (в железнодорожных вагонах, закрытых автомашинах, герметизированных отапливаемых отсеках самолетов, трюмах и т.д.) на любые расстояния при температуре окружающего воздуха от минус 25 до 55 ^{0}С и относительной влажности до 98 % при температуре 35 ^{0}С.

10.2 Условия хранения должны соответствовать требованиям группы 1(Л) по ГОСТ 15150 в закрытых отапливаемых помещениях при температуре окружающего воздуха от 5 до 40 °С.

10.3 В помещении для хранения не должно быть токопроводящей пыли, паров кислот и щелочей, а также газов, вызывающих коррозию и разрушающих изоляцию.

11 ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

11.1 Предприятие-изготовитель гарантирует соответствие контроллера требованиям настоящего руководства в течение 18 месяцев с момента ввода в эксплуатацию при соблюдении потребителем правил эксплуатации, транспортирования, хранения и монтажа.

11.2 Гарантийный срок хранения – 6 месяцев с момента отгрузки потребителю.

11.3 Контроллеры, у которых во время гарантийного срока будет выявлено несоответствие требованиям настоящего руководства, безвозмездно заменяется или ремонтируется предприятием-изготовителем.

11.4 Адрес предприятия изготовителя:

ООО «СИНКРОСС», Россия, 410010, г. Саратов, ул. Жуковского, д. 9А, тел. (8452) 55-66-56, e-mail: office@sinkross.ru.

12 СВЕДЕНИЯ О РЕКЛАМАЦИЯХ

Рекламации потребителя предъявляются и удовлетворяются в следующем порядке:

При получении контроллера от транспортной организации получателю следует визуальным осмотром проверить целостность транспортной упаковки и комплектности.

В случае обнаружения повреждений транспортной тары или комплектности, составляется соответствующий акт в присутствии грузополучателя.

Контроллер, у которого в течение гарантийного срока, при условии соблюдения правил хранения, транспортирования, монтажа и эксплуатации, будут выявлены отказы в работе или неисправности, безвозмездно ремонтируется или заменяется на исправный предприятием-изготовителем.

При отказе контроллера в период гарантийного срока потребителем должен быть составлен технический акт, в котором указывается:

- заводской номер;
- дата начала эксплуатации;
- условия эксплуатации;
- количество часов работы до момента отказа;
- дата возникновения отказа;
- характер отказа;
- предполагаемая причина возникновения отказа;
- меры, принятые после возникновения отказа.

Акт высылается предприятию-изготовителю для устранения выявленных дефектов.

ПРИЛОЖЕНИЕ А . ИНСТРУКЦИЯ ПО НАСТРОЙКЕ И ПРОГРАММИРОВАНИЮ

КОНТРОЛЛЕР К-3102

2011

1 ВВЕДЕНИЕ

Настоящая инструкция предназначена для ознакомления с контроллером К-3102, принципом его настройки и программирования.

2 ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ

Внешний вид контроллера К-3102 изображен на рисунке 1.

Рисунок 1. Внешний вид контроллера К-3102

3 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Технические характеристики контроллера приведены в таблице 1.

Таблица 1

Характеристика	Значение
Количество последовательных портов RS-485	3
Количество последовательных портов RS-485/RS-232	1
Количество Ethernet портов	1
Встроенный коммутатор (switch)	да
Поддерживаемые протоколы: порт RS-485 порт Ethernet	 Modbus RTU Modbus TCP
Скорость обмена по RS-485 (бит/с)	1200,2400,4800,9600. 19200,28800,38400, 57600,76800,115200, 128000,153600,230400
Скорость обмена по Ethernet (Мбит/с)	10/100
Максимальное количество поддерживаемых TCP/IP соединений	до 32*
Напряжение питания	24 В
Потребляемый ток	не более 0,2 А
Степень защиты	IP20
Размер памяти программ	65535 байт
Размер памяти данных	4096 байт

* - зависит от версии программного обеспечения

4 ОПИСАНИЕ РАБОТЫ

Контроллер К-3102 представляет собой программируемое логическое устройство предназначенное для работы в системах сбора данных. В состав контроллера входят: 4 последовательных порта RS-485 (протокол Modbus RTU), 1 порт Ethernet, двухпортовый ethernet коммутатор (switch).

4.1 Режимы работы последовательные портов

Последовательные порты контроллера К-3102 работают по протоколу Modbus RTU в одном из следующих режимов:

Режим 'Master': В этом режиме контроллер является ведущим линии Modbus RTU. Выбор режима работы 'Master' осуществляется установкой параметров «Master адрес»=0 и «Максимальный Master адрес»=0.

Режим "Slave": В этом режиме прибор является ведомым сети Modbus RTU. Поддерживаемые функции Modbus RTU приведены в таблице 4.1.

Таблица 4.1

№	Функция	Описание	Примечание
1	0x00	Пустая команда	
2	0x03	Чтение регистров	
3	0x04	Чтение регистров	
4	0x10	Запись многих регистров	
5	0x06	Запись одного регистра	
7	0x7A	Чтение идентификатора устройства	
8	0x7D	Работа с терминалом последовательного порта	
9	0x6D	Транзит данных	
10	0x50-0x53	Расширенный транзит данных	

Адрес ведомого на соответствующем порту при работе в этом режиме определяется параметром «Slave адрес».

Выбор режима «Slave» осуществляется установкой значения параметра «Master адрес» большего чем значение параметра «Максимального Master адреса».

Режим 'Multi Master': Режим используется для организации мульти-мастерной сети на основе протокола Modbus RTU. Данная модификация протокола позволяет присутствовать в одной подсети нескольким Master-устройствам. При настройке Master-устройств в одной подсети создается очередность работы Master-портов. Центральный арбитр отсутствует, а последовательность работы Master-портов храниться непосредственно в каждом устройстве.

Мульти-мастерная подсеть работает циклами. За один проход каждое Master-устройство генерирует один запрос и передает управление следующему. Существует виртуальный курсор, указывающий на активное Master-устройство. Курсор определяется и вычисляется в каждом Master-устройстве самостоятельно. Для синхронизации этих виртуальных курсоров в одной подсети используется «пакет синхронизации», посредством которого Master-устройство и осуществляет передачу управления следующему. В случае выхода из строя одного из Master-устройств, следующее в очередности работы исправное устройство подхватит управление.

Для настройки режима в настройках последовательного порта устанавливается значение параметра «Максимальный Master адрес» равное количеству мульти-мастерных устройств в данной сети. В каждом мульти-мастерном устройстве устанавливается уникальный «Master адрес». Задаются временные параметры работы мульти-мастерной сети («Время захвата линии мульти-мастером» и «Время стартовой синхронизации»). Параметр «Время захвата линии мульти-мастером» определяет время ожидания получения «пакета синхронизации» от активного мастера сети, по истечении которого произойдет смена виртуального курсора. Параметр «Время

стартовой синхронизации» определяет время после запуска, в течение которого устройство ожидает получения «пакета синхронизации».

4.2 Транзит данных последовательными портами

Последовательные порты RS-485 прибора поддерживают функции транзита данных. Для передачи пакетов между последовательными портами прибора используются Modbus RTU функции 0x6d, 0x50-0x54.

При получении пакета с Modbus функцией 0x6D контроллер на данном порту формирует пакет подтверждения согласно таблице 4.2. К содержимому пакета добавляется контрольная сумма, полученный пакет направляется на порт, указанный в параметре «Порт функции 0x6D». При повторном запросе контроллер возвращает полученный ответ или пакет подтверждения согласно таблице 4.2.

Алгоритм работы контроллера при получении команд расширенного транзита (функции 0x50-0x53) аналогичен описанному выше, за исключением того, что номер порта на который осуществляется транзит данных определяется функцией запроса (0x50-транзит на порт 1, 0x51-транзит на порт 2 и т.д.).

Таблица 4.2

Код	Наименование	Примечания
		Коды ошибок
05h	Подтверждение, ожидание ответа	Slave принял и обрабатывает запрос. Для получения ответа Master-устройству следует повторить запрос полностью позднее.
06h	Занят, Отказ в ответе	Slave занят и не может выполнить запрос в данный момент. Master-устройству следует повторить запрос позднее.
82h	На ответном транзитном порту выключена Master-служба	Необходимо включить Master-службу на ответном транзитном порту данного транспорта или задать другой путь
83h	Нет ответа на запрос	Указанный в пути адрес не существует

4.3 Работа в качестве преобразователя Modbus TCP – Modbus RTU

Контроллер К-3102 может работать в режиме преобразователя протоколов Modbus TCP в Modbus RTU. Последовательный порт, на который осуществляется транзит данных, должен работать в режиме “Master” или “Multi Master”.

Поддерживается до 16 подключений клиентов. Запросы клиентов Modbus TCP транслируются на последовательные порты согласно таблице транзита. Таблица транзита определяет привязку Modbus адресов устройств и последовательных портов контроллера к которым они подключены.

Если запрашиваемый адрес совпадает с собственным адресом К3102, то запрос обрабатывается контроллером, далее проверяется наличие полученного адреса в таблице транзита.

Если устройство с таким адресом найдено в таблице транзит, то к содержимому пакета добавляется контрольная сумма и пакет отправляется на порт, указанный в таблице. Порт, на который осуществляется транзит данных, должен быть настроен на работу в режиме «Master» или «MultiMaster» в противном случае тразит данных на порт блокируется.

Настройка таблицы транзита осуществляется с помощью ПО конфигурации (см п.7.2.2).

5 ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение контроллера К3102 делится на две части – прикладное и системное ПО. Прикладное ПО служит для реализации различных арифметико-логических функций и алгоритмов управления согласно конкретным задачам применения. Оно представляет собой набор инструкций интерпретатора. Системное ПО представляет собой встроенную

операционную систему, включающую в себя: поддержку интерфейсов и протоколов связи, интерпретатор команд прикладного ПО и механизмы конфигурирования.

5.1 Разработка прикладного программного обеспечения

Для написания программы используется внешний текстовый редактор, не вносящий своих управляющих тегов в код программы (Notepad, WordPad и т.п.). Файл программы должен иметь расширение .asm.

Допускается подключение к основной программе программных модулей, оформленных в виде отдельных файлов с расширением .inc. Для этого используется ключевое слово #include.

Примечание: Обязательным условием является подключение к программе модуля k3102.inc. Файл k3102.inc должен быть скопирован в каталог разрабатываемой программы.

Для определения символьных и строковых констант используется директива #define.

Текст программы должен заканчиваться ключевым словом end.

Пример кода программы:

```
#include k3102.inc

;*****************************************************************
; системная задача (см. пункт 4.3 Работа с задачами)
;*****************************************************************

 start_task 1,task1              ; запуск задачи 1
end_main_task                    ; инструкция завершения работы системной задачи

;*****************************************************************
; тело задачи 1
;*****************************************************************
task1   ….
        jmp task1

end
```

Компиляция и загрузка программы выполняется при помощи программы конфигурации контроллера К-3102 (см. п.7.2.3 Компиляция и запись прикладной программы в контроллер).

5.2 Организация памяти прикладного программного обеспечения

Программисту предоставляется адресуемое множество команд в интервале [0x0000, 0x4FFF]. Команды не имеют фиксированной длины. Прямые операции со стеком не доступны программисту. Состояние стека меняется автоматически при использовании команд работы с подпрограммами. Размер стека для каждой задачи равен 32. Таким образом, программист не должен превысить порог в 32 вложения подпрограмм.

Общий объем памяти данных доступных программисту 4096 байт. Минимально адресуемая ячейка памяти команд – слово. Память 0x0000-0x0FFF доступна для чтения/записи (функции 0x03,0x04,0x06,0x10).

5.3 Работа с задачами

Программисту доступно до 5 задач, выполняемых попеременно за каждый проход основного цикла программы-интерпретатора.

Существующие ограничения: контроль доступа к общим ресурсам не реализован, программист сам контролирует, что ресурс (переменная, таймер или буфер приема-передачи) используемый в задаче 1, не используется задачей 2 и т.д..

При старте системы автоматически создается системная задача, запускающая задачи 1-5 программиста и завершающая свою работу. В последствии работают только те задачи, объявление которых было выполнено в теле основной задачи. Объявление и запуск задач осуществляется при помощи инструкции start_task. Тело задачи должно представлять собой бесконечный цикл (см. пример 1).

Пример 1: Запуск задачи 1,2,3

```
;*****************************************************************
; системная задача
;*****************************************************************

 start_task 1,task1             ; запуск задачи 1
 start_task 2,task2             ; запуск задачи 2
end_main_task                   ; инструкция завершения работы системной задачи


;*****************************************************************
; тело задачи 1
;*****************************************************************
task1    ….
         jmp task1

;*****************************************************************
; тело задачи 2
;*****************************************************************
task2    ….
         jmp task2
```

Завершение системной задачи осуществляется инструкцией end_main_task.

5.4 Работа с таймерами

Программисту доступно 16 программных 16-разрядных таймеров. Инкрементирование регистра таймера происходит с частотой 1кГц (период 1 мс).

Запуск таймера осуществляется командой SFTStrt n, time, где n- номер таймера, time – период работы таймера в миллисекундах. При достижении таймером значения равного time выставляется флаг «счет завершен», таймер останавливается.

Проверка флага выполняется командой SFTStat n, где n- номер таймера. Результатом проверки является установка/снятие флага Z. Повторный запуск таймера командой SFTStrt сбрасывает таймер и флаг и начинает новый отсчет.

Функция ожидания таймера SFWtTmr n приостанавливает выполнение задачи до тех пор, пока не будет выставлен флаг таймера n.

Пример 2. Инкрементирование регистра reg1 раз в секунду

```
;*****************************************************************
; тело задачи 1
;*****************************************************************
```

```
task1  incr reg1              ; инкремент регистра reg 1
       SFTStrt 1,d'1000'      ; запуск таймера 1 с периодом 1000 мс
       SFWtTmr 1              ; ожидание флага таймера 1
       jmp task1
```

Пример 3. Инкрементирование регистра reg1 раз в секунду без останова выполнения задачи

```
;*****************************************************************
; тело задачи 1
;*****************************************************************
task1     incr reg1              ; инкремент регистра reg 1
          SFTStrt 1,d'1000'      ; запуск таймера 1 с периодом 1000 мс
label1    SFTStat  1             ; проверка флага таймера 1
          jz task1
          …                      ; выполнение команд
          jmp label 1
```

5.5 Работа со светодиодами

Программисту доступно управление двумя светодиодами, расположенными на передней панели. Включение светодиода осуществляется командой SFledon n, n- номер светодиода. Для выключения используется команда SFledoff n. Инверсия состояния светодиода выполняется командой SFledinv n.

5.6 Работа с последовательными портами

Программисту доступно 4 последовательных порта RS-485, работающих по протоколу Modbus RTU. Работа с портом осуществляется через программный буфер размером 256 байт. Данные предназначенные для отправки в последовательный порт помещаются в связанный с ним буфер и хранятся там до тех пор, пока пакет не отправлен полностью.

Чтение регистров modbus устройств осуществляется функцией SFRdreg, запись функцией SFWrreg. Функции SFRdreg и SFWrreg используют стандартные функции Modbus RTU – 0x03 , 0x04, 0x10. Передача пакета Modbus RTU с функциями отличными от вышеперечисленных осуществляется инструкцией SFWrData. Статус приемо-передающего буфера определяется командой SFBufstat, в качестве параметра которой указывается адрес регистра сохранения статуса.

Возможные значения статуса приемо-передающего буфера указаны в таблице 2.

Таблица 2. Возможные значения статуса буфера последовательного порта

Обозначение	Значение	Описание
NOT_COMPLETE	0xFFFF	«Запрос выполняется». Значение устанавливается сразу после записи данных в последовательный порт, и не изменяется до тех пор, пока не наступит одно из событий: приход ответа от спрашиваемого устройства или наступление таймаута ожидания ответа. Статус буфера NOT_COMPLETE фактически означает, что данный момент буфер занят, программист должен дождаться изменения статуса, прежде чем выполнять какие-либо операции записи\чтения для этого буфера.
NO_ANSWER_TIMEOUT	0xAAAA	«Таймаут ответа». Значение устанавливается если в течении времени таймаута от устройства не получен корректный ответ.

ERROR_ANSWER	0xCCCC	«Исключительный ответ от устройства». Выставляется, когда от устройства получен исключительный ответ.
CONNECT	0x5555	«Готов ответ». Выставляется, когда устройства пришел корректный ответ.

После завершения передачи порт устройства переключается на прием и ожидает прихода ответного пакета или таймаута в зависимости, что наступит ранее.

При выполнении команды SFBufStor произойдет запись указанного количества регистров принятых данных из программного буфера по указанному адресу в память данных.

Важно отметить, что для получения ожидаемых результатов выполнения программы
1. Не допускается одновременное использование одних и тех же приемо-передающих буферов в разных задачах.
2. Не допускаются операции чтения/записи буфера при его статусе NOT_COMPLETE.

При наличии свободных приемо-передающих буферов возможно использование несколько буферов для передачи на один последовательный порт. В этом случае запрос буфера n будет поставлен в очередь ожидания последовательного порта и его выполнение начнется сразу после прихода ответа или наступления таймаута запроса n-1.

Пример 4 Чтение регистров 0x0000-0x000a из подчиненного с адресом 0x75

```
;*****************************************************************
; тело задачи 1
;*****************************************************************

task1    SFRdreg UART1,BUF1,1,"75",0x03,0x0000,0x0a    ; послать запрос в порт 1 через буфер BUF1
wait1    SFBufstat BUF1, BUF1_STAT                     ; считать статус приемного буфера
                                                       ; в регистр BUF1_STAT
         cmplneq BUF1_STAT,NOT_COMPLETE                ; проверить что запрос выполнен
         jmp wait1
         cmplneq BUF1_STAT,NO_ANSWER_TIMEOUT           ; проверить что не таймаут ответа
         jmp dev_fail
         cmpleq BUF1_STAT,CONNECT                      ; проверить ответ от устройства
         jmp dev_fail

         SFBufStor BUF1,dev_answer,0,all               ; считать все данные из буфера сохранить в
                                                       ; памяти данных начиная с адреса dev_answer;
         incr dev_goodcount                            ; инкрементировать счетчик ответов
         jmp task1

dev_fail incr dev_failcount                            ; инкрементировать счетчик ошибок
         jmp task1
```

Пример 5 Чтение входов функцией 0x6e и запись выходов функцией 0x7c ДВВ адресом 0x02

```
;*****************************************************************
; тело задачи 1
;*****************************************************************
task1    movlr dev1_buf,0x026e                         ; записать 0x026e по адресу dev1_buf
         movlr dev1_buf+1,0x027c                       ; записать 0x027c по адресу dev1_buf+1

task12   SFWrData UART1,BUF1,dev1_buf,2                ; записать 2 байта начиная с адреса dev1_buf
                                                       ; в порт uart1 через буфер buf1
```

```
wait1   SFBufstat BUF1, BUF1_STAT
        cmplneq BUF1_STAT,NOT_COMPLETE
        jmp wait1
        cmplneq BUF1_STAT,NO_ANSWER_TIMEOUT
        jmp dev1_fail
        cmpleq BUF1_STAT,CONNECT
        jmp dev1_fail
        SFBufStor BUF1,dev1_answer,0,all          ; сохранить ответ ДВВ начиная с адреса dev1_answer

        movlr dev1_buf+2,0x5555                   ; записать  0x5555  по адресу dev1_buf+2

  task1_3  SFWrData UART1,BUF1,dev1_buf+1,4       ; записать 4 байта начиная с адреса dev1_buf
                                                  ; в порт uart1 через буфер buf1

  wait2  SFBufstat BUF1, BUF1_STAT
         cmplneq BUF1_STAT,NOT_COMPLETE
         jmp wait2
         cmplneq BUF1_STAT,NO_ANSWER_TIMEOUT
         jmp dev1_fail
         cmpleq BUF1_STAT,CONNECT
         jmp dev1_fail
        SFBufStor BUF1,dev1_answer,0,all
        jmp task 1_2

dev1_fail  incr dev_failcount                     ;  инкрементировать счетчик ошибок
           jmp task1_2
```

5.7 Работа с клиентом Modbus TCP

Программисту доступны 16 клиентских сокетов Modbus TCP. Чтение и запись регистров с\в серверного(ное) устройство выполняется инструкциями SF_TCPRdreg и SF_TCPWrreg соответственно.

Как и в случае с последовательными портами одновременное использование сокетов разными задачами не допускается.

При выполнении инструкции SF_TCPRdreg или SF_TCPWrreg происходит соединение с серверным устройством, с последующим обменом данными и разрывом соединения.

В отличии от работы с последовательными портами результат выполнения инструкции SF_TCPRdreg будет записан не в приемо-передающий буфер, а непосредственно по указанному адресу в память данных.

Пример 6. Чтение регистров 0x0000-0x000a из серверного устройства с IP адресом 192.168.23.06

```
;*****************************************************************
; тело задачи 1
;*****************************************************************
```

; считать через 1 сокет с устройства 192.168.23.06 порт 502, ID 0x05 функцией 0x03 10 регистров начиная с 0x0000 статус запроса отображать в регистре socket1_stat, результат сохранить в памяти начиная с dev1_answer

```
task1  SF_TCPRdreg SOCKET1,"192.168.23.06",d'502',socket1_stat,0x05,0x03,0x0000,0x0a, dev1_answer
wait1  cmplneq socket1_stat,NOT_COMPLETE
       jmp wait1
       cmplneq socket1_stat,NO_ANSWER_TIMEOUT
       jmp dev_fail
```

```
        cmpleq socket1_stat,CONNECT
        jmp dev_fail
        ;  ……                                    получен корректный ответ
        jmp task1

dev_fail  incr dev_failcount                      ;  инкрементировать счетчик ошибок
          jmp task1
```

6. СПИСОК КОМАНД

№	Наименование	Описание	Флаги
		1. Системные команды	
1	start_task	Запуск задачи	
2	end_main_task	Завершение основной задачи	
3	SFTStrt	Запуск таймера	
4	SFTStat	Проверка состояния таймера	Z
5	SFWtTmr	Ожидание таймера	
6	SFledon	Включение светодиода	
7	SFledoff	Выключение светодиода	
8	SFledinv	Инвертировать состояние светодиода	
9	SFRdreg	Чтение регистров из подчиненного	
10	SFWrreg	Запись регистров подчиненного	
11	SFWrData	Запись данных в подчиненного	
12	SFBufstat	Чтение статуса буфера	
13	SFBufStor	Прочитать данные из буфера	
14	SF_TCPRdreg	Прочитать регистры с серверного устройства по протоколу Modbus TCP	
15	SF _TCPWrreg	Записать регистры с серверного устройства по протоколу Modbus TCP	
		2. Словно ориентированные команды	
		2.1 Логические команды	
16	andlr	Логическое «И» содержимого регистра и константы	Z,N
17	andrr	Логическое «И» содержимого регистров	Z,N
18	andrlr	Логическое «И» содержимого регистра и константы с сохранением результата	Z,N
19	andrrr	Логическое «И» содержимого регистров с сохранением результата	Z,N
20	orlr	Логическое «ИЛИ» содержимого регистра и константы	Z,N
21	orrr	Логическое «ИЛИ» содержимого регистров	Z,N
22	orrlr	Логическое «ИЛИ» содержимого регистра и константы с сохранением результата	Z,N
23	orrrr	Логическое «ИЛИ» содержимого регистров с сохранением результата	Z,N
24	xorlr	Логическое «исключающее ИЛИ» содержимого регистра и константы	Z,N
25	xorrr	Логическое «исключающее ИЛИ» содержимого регистров	Z,N
26	xorrlr	Логическое «исключающее ИЛИ» содержимого регистра и константы с сохранением результата	Z,N

27	xorrrr	Логическое «исключающее ИЛИ» содержимого регистров с сохранением результата	Z,N
		2.2 Арифметические команды	
28	addlrnc	Суммирование содержимого регистра и константы	Z,N,C,OV
29	addrrnc	Суммирование содержимого регистров	Z,N,C,OV
30	addlrc	Суммирование содержимого регистра и константы с учетом флага переноса C	Z,N,C,OV
31	addrrc	Суммирование содержимого регистров с учетом флага переноса C	Z,N,C,OV
32	sublrnc	Вычитание константы из содержимого регистра	Z,N,C,OV
33	subrrnc	Вычитание содержимого регистров	Z,N,C,OV
34	sublrc	Вычитание константы из содержимого регистра с учетом флага заёма	Z,N,C,OV
35	subrrc	Вычитание регистров с учетом флага заёма	Z,N,C,OV
36	mullr	Умножение содержимого регистра на константу	Z
37	mulrr	Умножение содержимого регистров	Z
38	divlr	Деление содержимого регистра на константу	Z
39	divrr	Деление регистров	Z
		2.3 Команды сдвига	
40	rlc	Циклический сдвиг влево содержимого регистра через флаг переноса	Z,N
41	rrc	Циклический сдвиг вправо содержимого регистра через флаг переноса	Z,N
42	rlnc	Циклический сдвиг влево содержимого регистра	Z,N
43	rrnc	Циклический сдвиг вправо содержимого регистра	Z,N
44	shl	Сдвиг влево содержимого регистра	Z,N
45	shr	Сдвиг вправо содержимого регистра	Z,N
		2.4 Специальные команды	
46	movlr	Запись константы в регистр	
47	movrr	Копирование регистра	
48	incr	Инкрементирование содержимого регистра	Z,N,C,OV
49	decr	Декрементирование содержимого регистра	Z,N,C,OV
50	not	Инверсия содержимого регистра	Z,N
51	neg	Инверсия содержимого регистра+1	Z,N,C,OV
52	swapr	Поменять местами байты регистра	
		3. Бит ориентированные команды	

53	bset	Установить бит	
54	bclr	Очистить бит	
55	btgl	Инверсия бита	
	bcopy	копирование бита m регистра reg2 в бит n регистра reg1	
		4. Команды работы с подпрограммами	
56	call	Вызов подпрограммы	
57	return	Возврат из подпрограммы	
		5. Команды переходов	
58	jmp	Безусловный переход на адрес	
59	jz	Переход на адрес, если установлен флаг Z	
60	jnz	Переход на адрес, если флаг Z не установлен	
61	jc	Переход на адрес, если установлен флаг C	
62	jnc	Переход на адрес, если флаг C не установлен	
63	jn	Переход на адрес, если установлен флаг N	
64	jnn	Переход на адрес, если флаг N не установлен	
65	jbs	Переход на адрес, если установлен бит регистра	
66	jbc	Переход на адрес, если не установлен бит регистра	
67	cmpleq	Пропустить, если значение регистра равно значению константы	
68	cmplneq	Пропустить, если значение регистра не равно значению константы	
69	cmplgt	Пропустить, если значение регистра больше чем значение константы	
70	cmpllt	Пропустить, если значение регистра меньше чем значение константы	
71	cmpreq	Пропустить, если значения регистров равны	
72	cmprneq	Пропустить, если значения регистров не равны	
73	cmprgt	Пропустить, если больше	
74	cmprlt	Пропустить, если меньше	
75	jeqrr	Переход на адрес, если значения регистров равны	
76	jneqrr	Переход на адрес, если значения регистров не равны	
77	jeqrl	Переход на адрес, если значение регистра равно значению константы	
78	jneqrl	Переход на адрес, если значение регистра не равно значению константы	
		6. Команды косвенной адресации	
75	movMR	Скопировать содержимое регистра 1 в регистр с адресом, лежащим в регистре 2	
76	movRM	Скопировать содержимое регистра с адресом, лежащим в регистре 1 в регистр 2	

7 НАСТРОЙКА

7.1 Установка параметров по умолчанию

Для запуска контроллера с параметрами по умолчанию, необходимо установить переключатель def (см. рисунок 1) в положение ON, подать питание на контроллер. Контроллер будет загружен с параметрами по умолчанию.

Значение IP адреса по умолчанию - 192.168.10.1. Последовательные порты контроллера по умолчанию устанавливаются в режим «Slave» с адресами 0x00.

При старте контроллера в режиме по умолчанию прикладная программа не выполняется.

7.2 Изменение настроек при помощи программы конфигурации

Настройка контроллера, а также запись прикладной программы, осуществляется при помощи программы конфигурации.

7.2.1 Установка связи

Обмен данными между контроллером и программой конфигурации производится по сети Ethernet. Для конфигурирования контроллера используется служебный сокет (порт) с номером 5000.

Для изменения настроек подключите контроллер к локальной сети, подайте питание. Запустите программу конфигурации. В появившемся окне (рисунок 2.1) выберете тип контроллера К-3102.

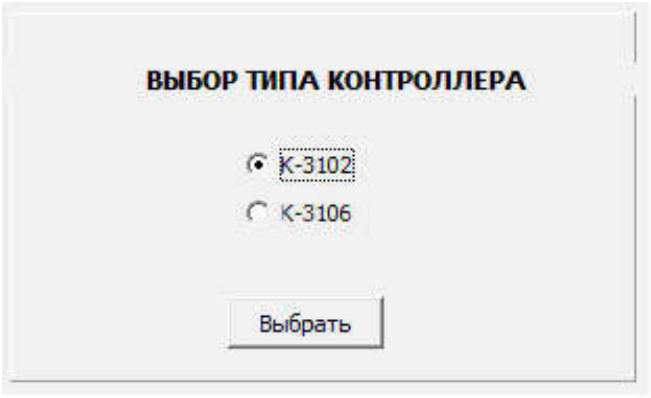


Рисунок 2.1

Внешний вид программы конфигурации после запуска представлен на рисунке 2.2.

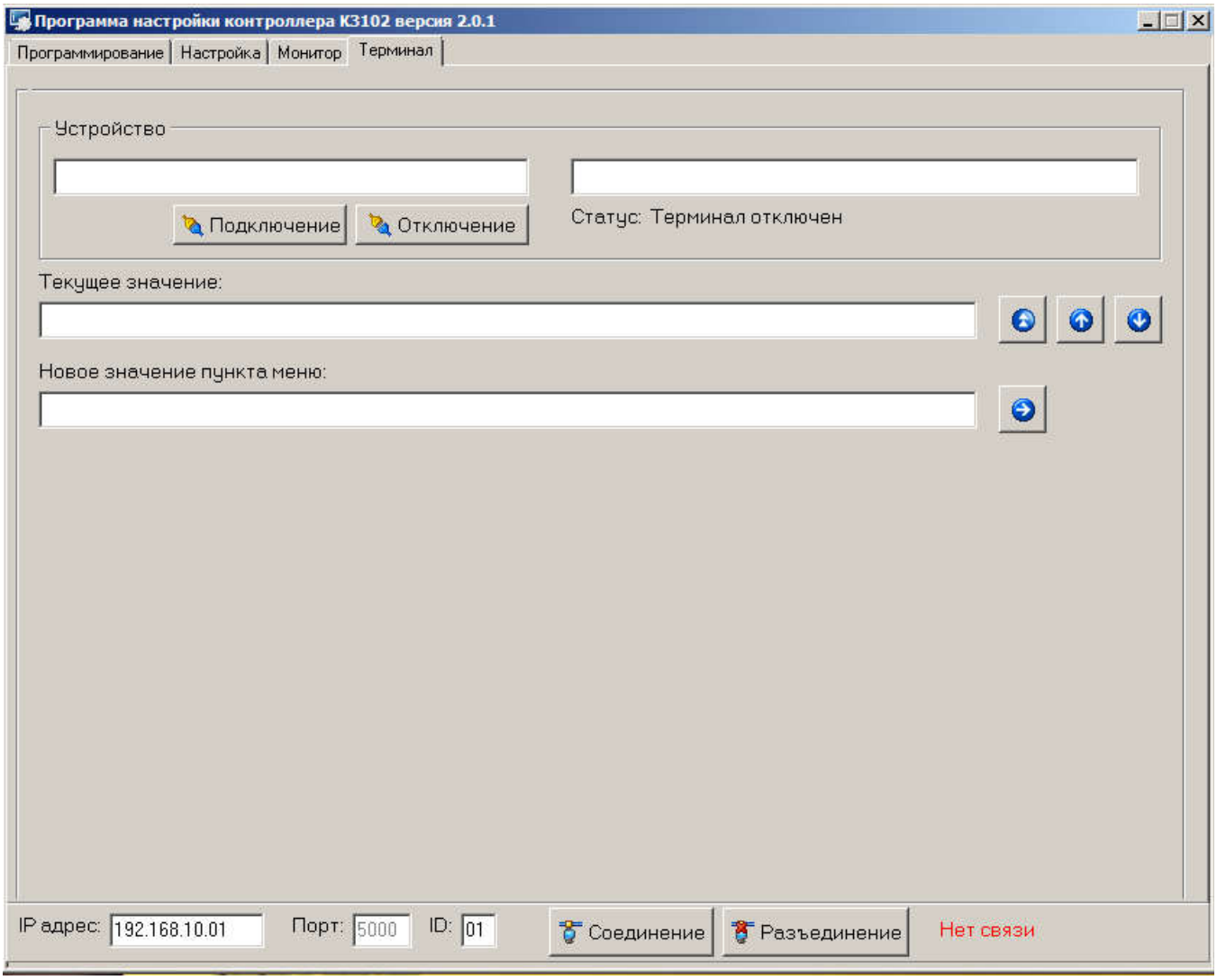


Рисунок 2.2 Внешний вид программы конфигурации при запуске.

Установите IP адрес контроллера и нажмите кнопку соединение. При правильном подключении и указании IP адреса будет выполнено подключение к контроллеру. В нижнем углу экрана отображается сообщение «Соединение установлено»

7.2.2 Настройка параметров

Для установки параметров порта выберите вкладку «Настройка». На экране отобразится окно рисунок 3.

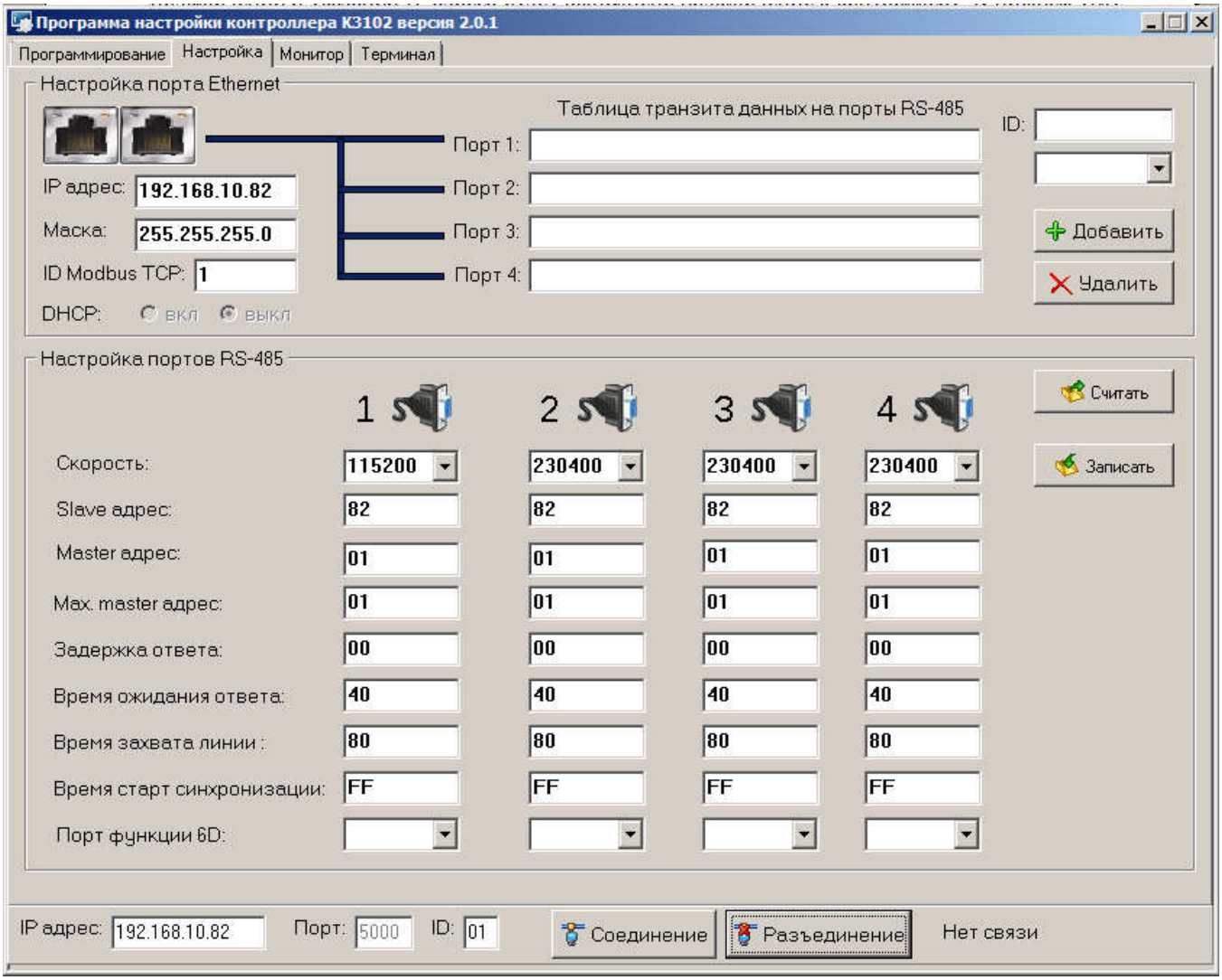


Рисунок 3 Окно настройки

Для чтения текущей конфигурации контроллера используется кнопка «Считать». Для сохранения параметров в энергонезависимой памяти – кнопка «Записать».

Установите требуемые параметры и сохраните их в контроллере, нажав кнопку «Записать». При успешном выполнении операции записи отображается сообщение с текстом «Изменения приняты».

Примечание: При изменении в IP адреса контроллера сообщение с «Изменения приняты» не появляется, связь с устройством разрывается, на экране отображается сообщение с текстом «Нет ответа от устройства». Для установки связи с контроллером нажмите кнопку сброса и выполните действия согласно п.7.2, используя новое значение IP адреса.

Внесение изменений в таблицу транзита преобразователя Modbus TCP - Modbus RTU выполняется следующим образом:
Для добавления устройства в таблицу: в поле ID укажите адрес устройства (HEX формат), выберите порт, к которому подключено устройство и нажмите кнопку «Добавить». При положительном результате операции адрес устройства отображается в таблице. Для удаления устройства из таблицы укажите его адрес в поле ID и нажмите кнопку удалить. В обоих случаях при положительном результате операции отображается сообщение с текстом «Изменения приняты».

7.2.3 Компиляция и запись прикладной программы в контроллер

Выберите вкладку «Программирование». На экране отобразится окно рисунок 4.

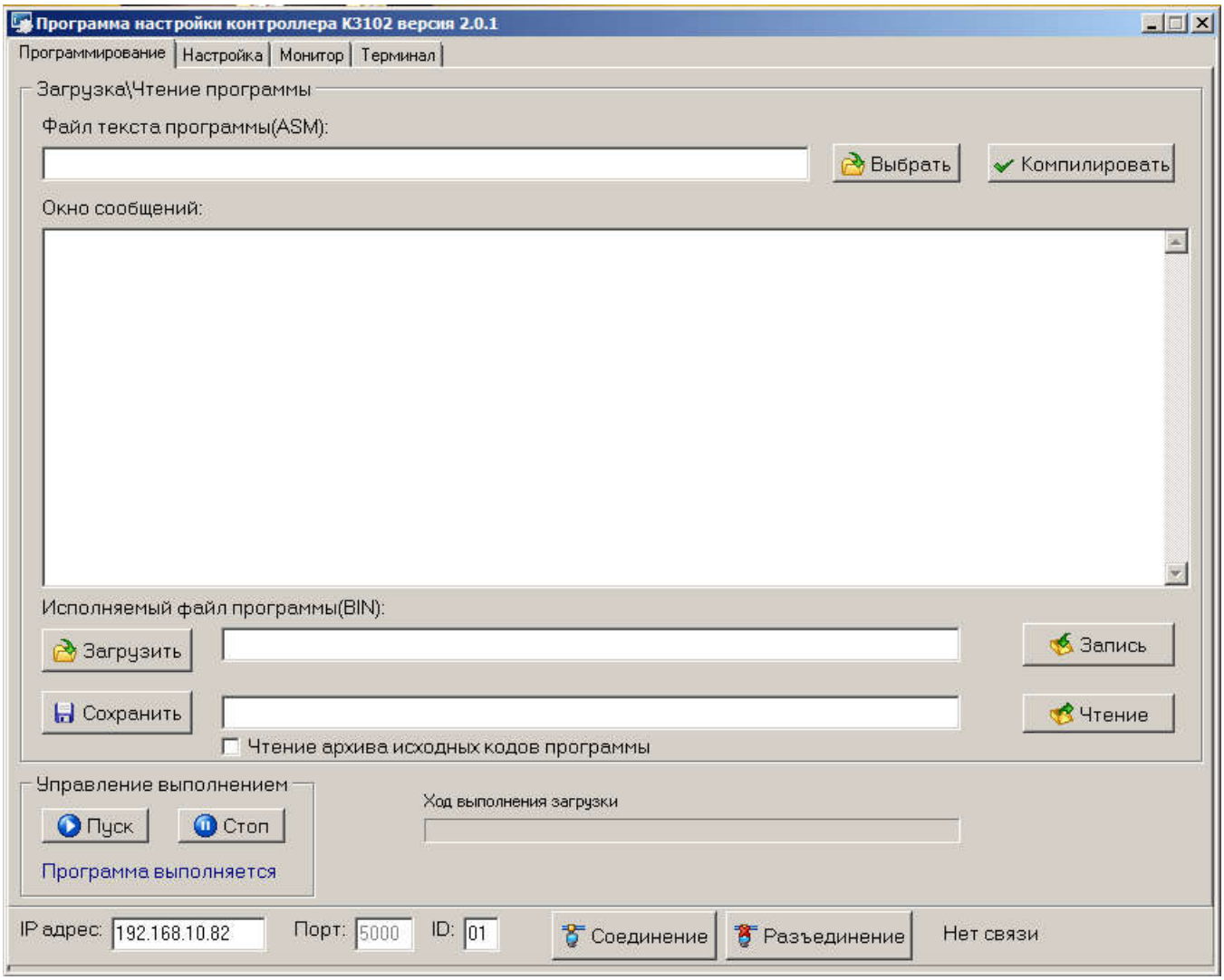


Рисунок 4 Внешний вид окна «Программирование»

7.2.3.1 Компиляция программы

Выберите файл с исходным кодом программы, нажав клавишу «Выбрать». Для компиляции программы используется кнопка «Компилировать». Ход, ошибки и предупреждения компилятора отображаются в окне сообщений. При успешном завершении компиляции отображается сообщение: «УСПЕШНО: Компиляция завершена». В папке с программой создается одноименный исполняемый файл с расширением .bin.

7.2.3.2 Загрузка и чтение программы

Для записи исполняемого кода в контроллер выберите файл, нажав клавишу «Загрузить». Загрузка программы осуществляется нажатием кнопки «Запись». Ход загрузки отображается в окне сообщений, а так же контролируется по индикатору загрузки. При положительном результате операции выводится сообщение «Загрузка программы завершена».

Для чтения исполняемого кода программы из контроллера необходимо выбрать имя файла, в котором будет сохранен исполняемый код контроллера. Для этого используется кнопка «Сохранить». Запуск операции чтения исполняемого кода производится нажатием кнопки «Чтение».

7.2.3.3 Управление выполнением программы

Для удобства отладки предусмотрена возможность остановки и возобновления выполнения программы. Выполнение прикладной программы будет остановлено/продолжено в зависимости от посланной в контроллер команды (кнопка «Стоп» - останавливает выполнение, кнопка «Пуск» возобновляет выполнение программы). Результат операции отображается в строке «Состояние:». При выключении питания или сбросе контроллера блокировка выполнения прикладной программы снимается автоматически.

7.2.4 Работа с терминалом

Для удаленной работы с терминалом последовательных портов используется вкладка «Терминал» см. рисунок 5. Терминал использует стандартный порт протокола Modbus TCP (порт 502). Перед началом работы с терминалом убедитесь, что адрес вызываемого устройства добавлен в таблицу транзита контроллера (см. п.7.2.2).

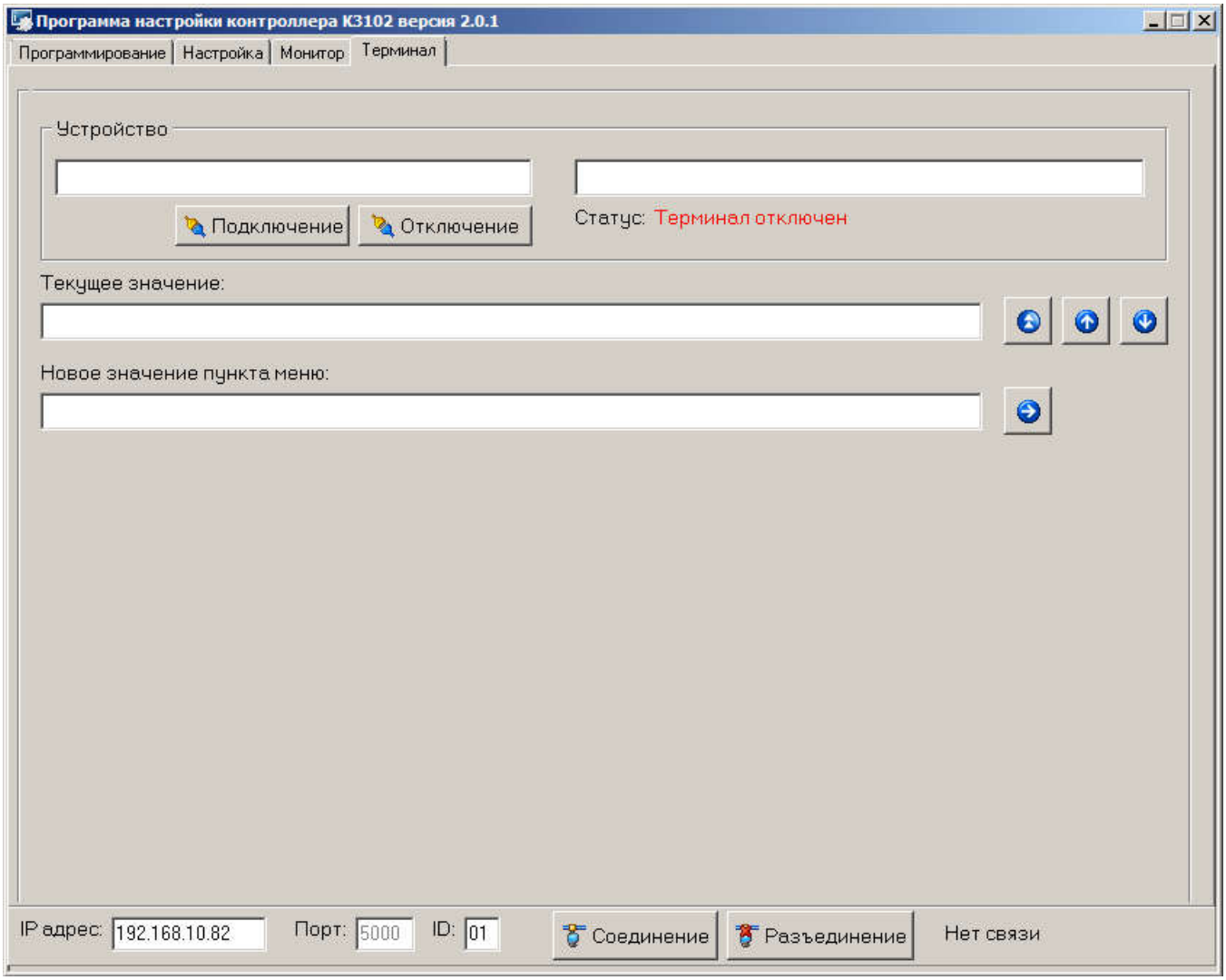


Рисунок 5 Внешний вид вкладки «Терминал»

Для подключения к удаленному устройству введите адрес в поле «Устройство». Адрес вводиться в hex формате.

Примечание: Функция 0x6D не указывается, для ввода функции 0x50-0x53 используется символ «#».
Пример: Строка «010315» в поле устройства в запросе. Строка «01#20251» соответствует последовательности 0x01, 0x52, 0x02, 0x6d, 0x51.

Нажмите кнопку «Подключение» при обнаружении устройства его идентификатор отображается в строке статуса. Текущий пункт меню выводится в поле «Текущее значение». Назначение кнопок управления терминалом:

 - указатель на начало меню

 - указатель на один пункт вверх

 - указатель на один пункт вниз

 - установка нового значения

7.3 Изменение настроек при помощи терминала последовательного порта

Настройка параметров производится по интерфейсу RS-485 с помощью персонального компьютера с использованием программы TestComm2 или MTest.

Запустите программу TestComm2 или MTest и настройте параметры работы указав порт связи, скорость интерфейса и Slave-адрес порта к которому осуществляется подключение.
После установления связи, открывается доступ к пунктам меню терминала. Каждый пункт меню представляет собой либо параметр, либо команду, действующую непосредственно, либо индикатор режима.

7.3.1 Описание пунктов меню терминала

Основное меню терминала имеет вид:

Пункт меню	Описание
Настройка порта 1	Используется для доступа к настройкам последовательного порта
Настройка порта 2	
Настройка порта3	
Настройка порта 4	
Настройка ETHERNET	Используется для доступа к настройкам порта LAN
Запись команды	Используется для доступа к сервисным функциям устройства

Подменю COM x

Пункт меню	Описание
Скорость передачи	параметр отображает текущее и позволяет задавать новое значение скорости обмена порта, доступные значения: 1,2 kBd, 2,4 kBd, 4,8 kBd, 7,2 kBd, 9,6 kBd, 14,4 kBd,19,2 kBd, 28,8 kBd 38,4 kBd, 57,6 kBd, 76,8 kBd, 4,8 kBd, 115,2 kBd, 153,6 kBd,230,4 kBd, 307,2 kBd;
‘Slave’ адрес	параметр отображает текущее и позволяет задавать новое значение Slave-адреса порта
‘Master’ адрес	параметр отображает текущее и позволяет задавать новое значение Master-адреса порта,
Максимальный ‘Master’ адрес	параметр отображает текущее и позволяет задавать новое значение максимального Master-адреса в мульти-мастерной сети
Задержка ответа	параметр отображает текущее и позволяет задавать новое значение времени задержки передачи ответа при работе порта в режиме Slave. Значение задается во временах передачи 1 байта на данной скорости обмена.
Время ожидания ответа	параметр отображает текущее и позволяет задавать новое значение времени ожидания ответа от Slave устройства. Значение задается во временах передачи 1 байта на данной скорости обмена.
Время работы мастера	параметр отображает текущее и позволяет задавать новое значение таймаута ожидания пакета синхронизации от активного мастера в мульти-мастерной сети. Значение задается во временах передачи 1 байта на данной скорости обмена.
Время стартовой синхронизации	параметр отображает текущее и позволяет задавать новое значение времени ожидания пакета синхронизации в мульти-мастерной сети при старте устройства. Значение задается во временах передачи 1 байта на данной скорости обмена.
Порт для функции 0x6D	параметр отображает текущее и позволяет задавать новое значение номера порта на который осуществляется перенаправление данных (функция 0x6d).
Применить и выйти	Сохранение настроек
Отмена	Выход в основное меню

Примечание: Доступные для ввода значения 00-FF (значения вводятся в шестнадцатеричном виде 0-9..A-F);

Подменю ETHERNET

Пункт меню	Описание
IP адрес	Значение IP адреса в формате xxx.xxx.xxx.xxx
Маска	Значение маски в формате xxx.xxx.xxx.xxx
ID	Адрес устройства (Modbus TCP)
Применить и выйти	Сохранение настроек
Отмена	Выход в основное меню

Подменю SYSTEM

Пункт меню	Описание
Запись команды	Используется для ввода сервисных команд
Отмена	Выход в основное меню

7.4 Изменение настроек при помощи WEB интерфейса

Настройка контроллера через WEB интерфейс осуществляется по протоколу HTTP. В строке браузера введите IP адрес устройства в окне браузера отобразится страница рисунок 6.

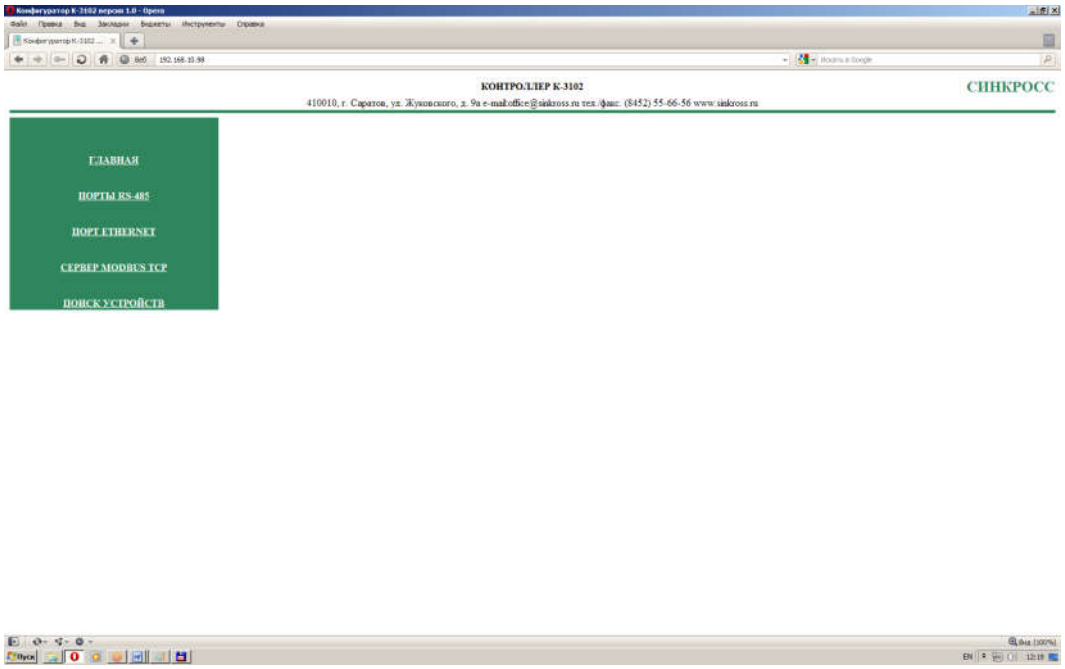


Рисунок 6 Главная страница настроек

Для перехода к странице настроек последовательных портов нажмите на вкладку «Порты RS-485». На экране отобразится страница настройки параметров последовательного порта рисунок 7.

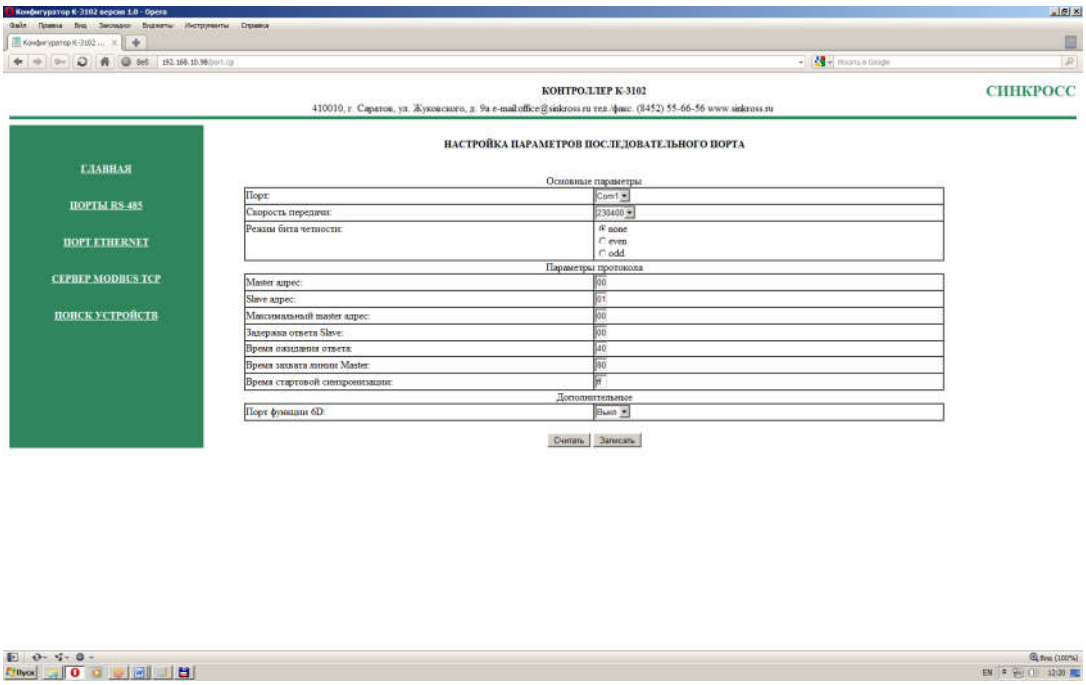


Рисунок 7 Страница настройки последовательных портов

Выберите нужный порт из списка и нажмите на кнопку «Считать». Произойдет обновление страницы, на экране отображаются параметры выбранного последовательного порта.

После изменения настроек выбранного порта нажмите на кнопку «Записать» для сохранения изменений.

Для перехода к странице сетевых настроек К-3102 нажмите на вкладку «Порт Ethernet». На экране отобразится страница рисунок 8. Измените IP адрес и маску прибора. Сохранение изменений осуществляется нажатием на кнопку «Записать».

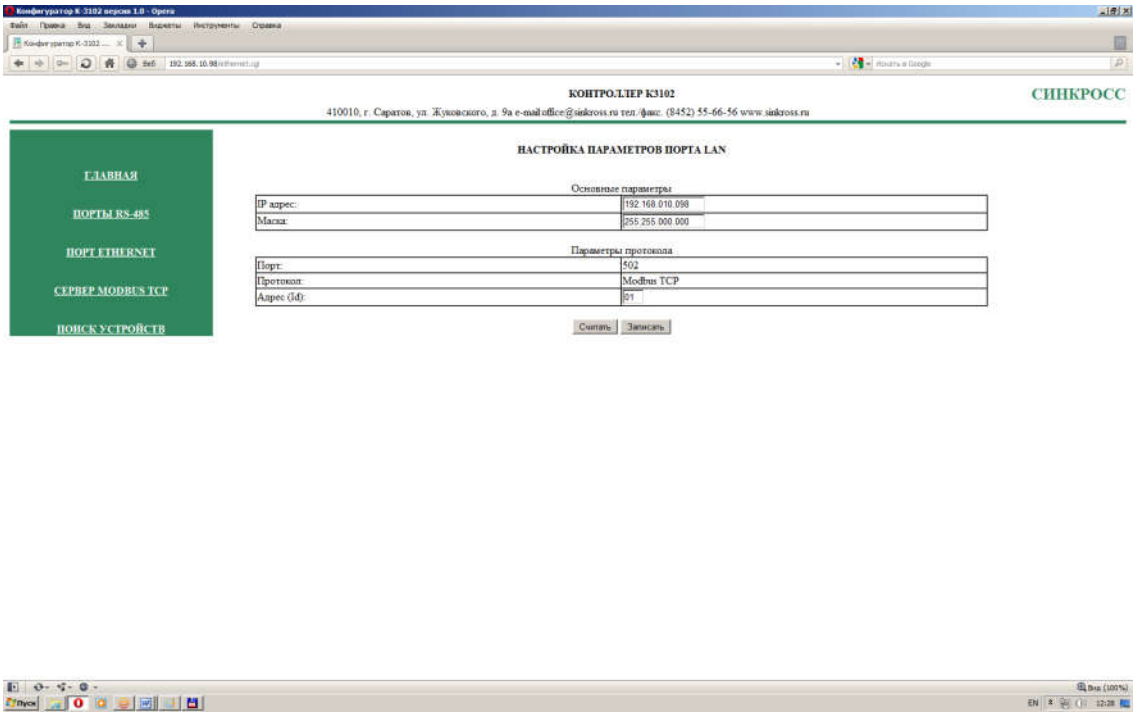


Рисунок 8 Страница настройки порта Ethernet

Для выполнения поиска устройств Modbus RTU подключенных к портам К-3102 выберите вкладку «Поиск устройств». В окне браузера отобразится страница рисунок 9. При нажатии кнопки «Поиск» будет выполнен поиск устройств подключенных к последовательным портам К-3102. В окне браузера отображаются результаты поиска. Поиск выполняется только на текущей скорости работы порта, при этом порт должен быть настроен в режиме Master или Multi-master. Поиск устройств на порту в режиме Slave не выполняется.

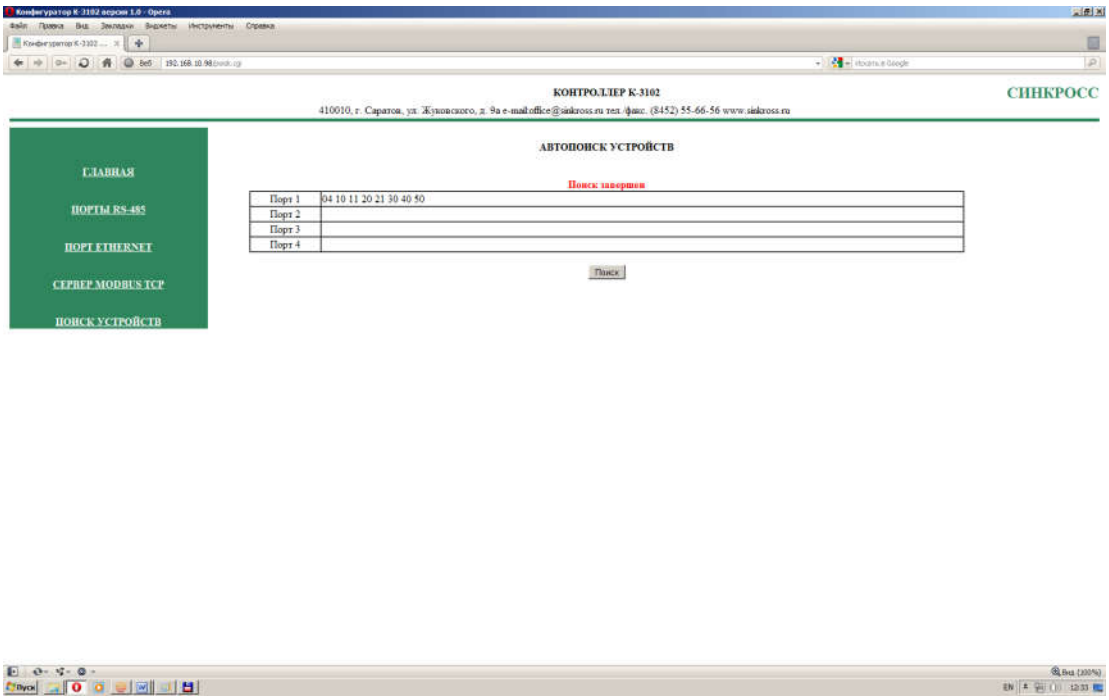


Рисунок 9 Страница поиска устройств

Для внесения изменений в таблицу транзита Modbus TCP сервера выберите вкладку «Сервер Modbus TCP». В окне браузера отобразится страница рисунок 10.

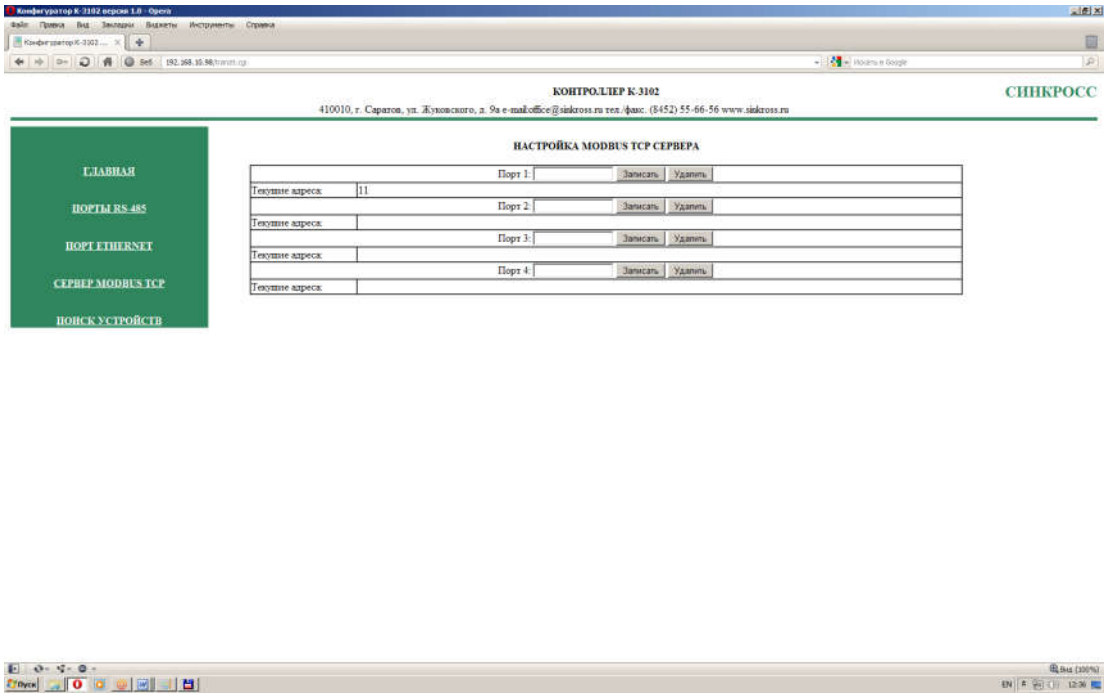


Рисунок 10 Страница настроек сервера Modbus TCP

В поле «Порт» введите через запятую Modbus-адреса устройств, запросы Modbus TCP к которым будут перенаправлены на соответствующий порт. Возможно указание диапазона адресов в формате: начальный адрес - конечный адрес. Адреса устройств указываются в шестнадцатеричном виде. Сохранение изменений выполняется нажатием на кнопку “Записать”. Удаление адресов из списка порта осуществляется кнопкой «Удалить».

Пример: К последовательному порту 1 подключены устройства с Modbus-адресами 5h,6h,7h,8h,18h. Для направления пакетов от Modbus TCP сервера к этим устройствам введите значение поля Порт1: 5h-8h,18h. Сохраните значения нажатием кнопки «Записать».

ОПИСАНИЕ ИНСТРУКЦИЙ

<table>
<tr><th colspan="2">1. Системные команды</th></tr>
<tr><td colspan="2"></td></tr>
<tr><td>1. start_task Запуск задачи
Синтаксис: start_task n, label
Операнды:0<n<6; 0x0000<offset<0x4FFF
Описание: Запускает задачу с номером n с начальным адресом offset</td><td>2. end_main_task Завершение основной задачи
Синтаксис: end_main_task
Описание: Останавливает выполнение системной задачи</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>3. SFTStrt Запуск таймера
Синтаксис: SFTStrt n, time
Операнды: 0<n<17;
0x0000<time<0xFFFF

Описание: Резервирует и запускает таймер n с периодом работы time</td><td>4. SFTStat Проверка состояния таймера
Синтаксис: SFTStrt n
Операнды: 0<n<17
Флаги: Z
Описание: Проверяет состояние таймера n выставляет флаг Z если таймер завершил отсчет</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>5. SFWtTmr Ожидание таймера
Синтаксис: SFWtTmr n
Операнды: 0<n<17
Описание: Останавливает выполнение задачи до тех пор, пока таймер n не завершит отсчет</td><td>6. SFledon Включение светодиода
Синтаксис: SFledon n
Операнды: 0<n<3
Описание: Включает светодиод n</td></tr>
<tr><td>7. SFledoff Выключение светодиода
Синтаксис: SFledoff n
Операнды: 0<n<3
Описание: Выключает светодиод n</td><td>8. SFledinv Инверсия состояния светодиода
Синтаксис: SFledinv n
Операнды: 0<n<3
Описание: Выключает светодиод n</td></tr>
<tr><td colspan="2">8. SFRdreg Чтение регистров из подчиненного
Синтаксис:SFRdreg port, buf, spath, path, func, offset, nreg
Операнды: 0<port<4; 1 <buf<6; 0 <spath; path=” ” (asckii строка); func=0x03 или 0x04;
0x0000≤ offset≤0x0FFF; 0x0000< nreg≤0x007E
Описание: Прочитать с порта port через буфер buf с устройства адресом path (spath- размер адреса в байтах) функцией func начиная с адреса offset количество регистров nreg

Пример: SFRdreg UART1,BUF1,5,”75#425#201”,0x03,0x0000,16</td></tr>
<tr><td colspan="2">9. SFWrreg Запись регистров подчиненного
Синтаксис: SFWrreg port, buf, spath, path, dstadr, srsadr, nreg
Операнды: 0<port<4; 1 <buf<6; spath>0; path=” ” (asckii строка); 0x0000≤ offset≤0x0FFF;
0x0000≤ dstadr ≤0x0FFF; 0x0000≤ srsadr ≤0x0FFF; 0x00< nreg≤0x7E
Описание: Записать в порт port через буфер buf в устройство адрес path (spath- размер пути) регистры начиная с Srs_adr в регистры начиная с Dst_adr число регистров nreg

Пример: SFRWrreg UART1,BUF1,5,”75#425#201”,0,0,16</td></tr>
</table>

10. SFWrData Запись данных в подчиненного
Синтаксис: SFWrData port, buf, srsadr, nbyte
Операнды: 0<port<4; 1 <buf<6; 0x0000≤ srsadr ≤0x0FFF; 0x00< nbyte≤0xFF
Описание: Записать в порт port через буфер buf данные, начиная с адреса srsadr количеством байт nbyte.

11. SFBufstat Чтение статуса буфера
Синтаксис: SFBufstat buf, adr
Операнды 1 <buf<6; 0x0000≤ adr ≤0x0FFF
Описание: Прочитать статус буфера buf в регистр с адресом adr. Возможные значения статуса приемо-передающего буфера приведены в таблице 2.

12. SFBufStor Прочитать данные из буфера
Синтаксис: SFBufStor buf, adr, offset, nreg
Операнды: 1 <buf<6; 0x0000≤ adr ≤0x2000;0x00≤ offset≤0x7E; 0x00< nreg≤0x7E
Описание: Считать данные из буфера buf со смещением offset в память начиная с адреса adr количеством nreg. Ключевое слово all используется в качестве указателя на то, что должны быть считаны все данные, находящиеся в буфере.
Пример: SFBufStor BUF1,0x0000,0,all

13. SF_TCPRdreg Прочитать регистры с серверного устройства по протоколу Modbus TCP
Синтаксис: SF_TCPRdreg socket, ip, port, adr_status, dev_adr, func, offset, nreg, answer_adr
Операнды: 0<socket<3; iP=”xxx.xxx.xxx.xxx”; 0x0000≤ port≤0xFFFF; 0x0000≤ adr_status ≤0x2000; 0x00< dev_adr≤0xFF; func=0x03 или 0x04; 0x0000≤ offset≤0xFFFF; 0x00< nreg≤0x7E; 0x0000≤ answer_adr ≤0x2000
Описание: Считать с устройства dev_adr функцией func кол-во регистров nreg начиная с адреса offset через сокет socket IP адрес ip порт port. adr_status-регистр статуса запроса answer_adr - указатель на адрес сохранения ответа.

14. SF_TCPWrreg Записать регистры с серверного устройства по протоколу Modbus TCP
Синтаксис: SF_TCPWrreg socket,ip,port,adr_status,dev_adr,func,offset,nreg,srsadr
Операнды: 0<socket<3; iP=”xxx.xxx.xxx.xxx”; 0x0000≤ port≤0xFFFF; 0x0000≤ adr_status ≤0x2000; 0x00< dev_adr≤0xFF; func=0x10; 0x0000≤ offset≤0xFFFF; 0x00< nreg≤0x7E; 0x0000≤ srsadr ≤0x2000
Описание: Записать в устройство dev_adr функцией func кол-во регистров nreg начиная с адреса offset через сокет socket IP адрес ip порт port. adr_status-регистр статуса запроса, srsadr - указатель на данные

2. Словно ориентированные инструкции

2.1 Логические команды

1. andlr Логическое «И» содержимого регистра и константы
Синтаксис: andlr reg,literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF
Операция: (reg) AND literal →(reg)
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистра и константы

2. andrr Логическое «И» содержимого регистров
Синтаксис: andrr reg1,reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: (reg1) AND (reg2) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистров

3. andrlr Логическое «И» содержимого регистра и константы с сохранением результата
Синтаксис: andrlr reg1,reg2,literal
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF; 0x0000≤literal≤0xFFFF
Операция: (reg2) AND literal →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистра и константы с сохранением результата

4. andrrr Логическое «И» содержимого регистров с сохранением результата
Синтаксис: andrrr reg1,reg2,reg3
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF; 0x0000≤reg3≤0x0FFF;
Операция: (reg2) AND (reg3) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистров с сохранением результата

5. orlr Логическое «ИЛИ» содержимого регистра и константы
Синтаксис: orlr reg,literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF
Операция: (reg) OR literal →(reg)
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистра и константы

6. orrr Логическое «ИЛИ» содержимого регистров
Синтаксис: orrr reg1,reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: (reg1) OR (reg2) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистров

7. orrlr Логическое «ИЛИ» содержимого регистра и константы с сохранением результата
Синтаксис: orrlr reg1,reg2,literal
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF; 0x0000≤literal≤0xFFFF
Операция: (reg2) OR literal →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистра и константы с сохранением результата

8. orrrr Логическое «ИЛИ» содержимого регистров с сохранением результата
Синтаксис: orrrr reg1,reg2,reg3
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF; 0x0000≤reg3≤0x0FFF;
Операция: (reg2) OR (reg3) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистров с сохранением результата

9. xorlr Логическое «исключающее ИЛИ» содержимого регистра и константы
Синтаксис: xorlr reg,literal
Операнды: 0x0000≤reg≤0x2000; 0x0000≤literal≤0xFFFF
Операция: (reg) XOR literal →(reg)
Флаги: Z, N
Описание: Логическое побитовое «исключающее ИЛИ» содержимого регистра и константы

10. xorrr Логическое «исключающее ИЛИ» содержимого регистров
Синтаксис: xorrr reg1,reg2
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000;
Операция: (reg1) XOR (reg2) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «исключающее ИЛИ» содержимого регистров

11. xorrlr Логическое «исключающее ИЛИ» содержимого регистра и константы с сохранением результата
Синтаксис: xorrlr reg1,reg2,literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF
Операция: (reg2) XOR literal →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «исключающее ИЛИ» содержимого регистра и константы с сохранением результата

12. xorrrr Логическое «исключающее ИЛИ» содержимого регистров с сохранением результата
Синтаксис: xorrrr reg1,reg2,reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) XOR (reg3) →(reg1)
Флаги: Z, N
Описание: Логическое побитовое «исключающее ИЛИ» содержимого регистров с сохранением результата

2.2 Арифметические команды

12. addlrnc Суммирование содержимого регистра и константы
Синтаксис: addlrnc reg1,reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) +literal →(reg1)
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистра reg2 и константы literal с сохранением результата в регистре reg1

13. addrrnc Суммирование содержимого регистров
Синтаксис: addrrnc reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) +(reg3) →(reg1)
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистров reg2 и reg3 с сохранением результата в регистре reg1

14. addlrc Суммирование содержимого регистра и константы с учетом флага переноса C
Синтаксис: addlrc reg1, reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) +literal+C →(reg1)
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистра reg2 и константы literal с учетом флага переноса C с сохранением результата в регистре reg1

15. addrrc Суммирование содержимого регистров с учетом флага переноса C
Синтаксис: addrrc reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) +(reg3) +C→(reg1)
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистров reg2 и reg3 с учетом флага переноса C с сохранением результата в регистре reg1

16. sublrnc Вычитание константы из содержимого регистра
Синтаксис: sublrnc reg1,reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) - literal →(reg1)
Флаги: Z, N, C, OV
Описание: Вычитание константы literal из содержимого регистра reg2 и с сохранением результата в регистре reg1

17. subrrnc Вычитание содержимого регистров
Синтаксис: subrrnc reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) - (reg3) →(reg1)
Флаги: Z, N, C, OV
Описание: Вычитание содержимого регистра reg3 из reg2 с сохранением результата в регистре reg1

18. sublrc Вычитание константы из содержимого регистра с учетом флага заёма
Синтаксис: sublrc reg1, reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) – literal - (~C) →(reg1)
Флаги: Z, N, C, OV
Описание: Вычитание константы literal из содержимого регистра reg2 с учетом флага заема с сохранением результата в регистре reg1

19. subrrc Вычитание регистров с учетом флага заёма
Синтаксис: subrrc reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) - (reg3) – (~C)→(reg1)
Флаги: Z, N, C, OV
Описание: Вычитание содержимого регистра reg3 из reg2 с сохранением результата в регистре reg1

20. mullr Умножение содержимого регистра на константу
Синтаксис: mullr reg1, reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) *literal→(reg1)(reg1+1)
Флаги: Z
Описание: Умножение содержимого регистра reg2 на константу с сохранением результата в регистры reg1 и reg1+1

20. mulrr Умножение содержимого регистров
Синтаксис: mulrr reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2) *(reg3)→(reg1)(reg1+1)
Флаги: Z
Описание: Умножение содержимого регистра reg2 на reg3 с сохранением результата в регистры reg1 и reg1+1

21. divlr Деление содержимого регистра на константу
Синтаксис: divlr reg1, reg2, literal
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg2) /literal→(reg1)(reg1+1)
Флаги: Z
Описание: Деление содержимого регистра reg2 на константу с сохранением результата в регистры reg1 и reg1+1

22. divrr Деление содержимого регистров
Синтаксис: devrr reg1, reg2, reg3
Операнды: 0x0000≤reg1≤0x2000; 0x0000≤reg2≤0x2000; 0x0000≤reg3≤0x2000;
Операция: (reg2)/(reg3)→(reg1)(reg1+1)
Флаги: Z
Описание: Умножение содержимого регистра reg2 на reg3 с сохранением результата в регистры reg1 и reg1+1

2.3 Команды сдвига

23. rlc Циклический сдвиг влево содержимого регистра через флаг переноса
Синтаксис: rlc reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Циклический сдвиг влево содержимого регистра reg1 через флаг переноса n раз

24. rrc Циклический сдвиг вправо содержимого регистра через флаг переноса
Синтаксис: rrc reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Циклический сдвиг вправо содержимого регистра reg1 через флаг переноса n раз

25. rlnc Циклический сдвиг влево содержимого регистра
Синтаксис: rlnc reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Циклический сдвиг влево содержимого регистра reg1 n раз

26. rrnc Циклический сдвиг вправо содержимого регистра
Синтаксис: rrnc reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Циклический сдвиг вправо содержимого регистра reg1 n раз

27. shl Сдвиг влево содержимого регистра
Синтаксис: shl reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Сдвиг влево содержимого регистра reg1 n раз

28. shr Сдвиг вправо содержимого регистра
Синтаксис: shr reg1, n
Операнды: 0x0000≤reg1≤0x2000; 0x00≤n≤0x0F;
Операция:
Флаги: Z, N
Описание: Сдвиг вправо содержимого регистра reg1 n раз

29. movlr Запись константы в регистр
Синтаксис: movlr reg,literal
Операнды: 0x0000≤reg≤0x2000; 0x0000≤literal≤0xFFFF;
Операция: (reg)=literal
Флаги:
Описание: Запись константы в регистр

30. movrr Копирование регистра
Синтаксис: movrr reg1,reg2
Операнды: 0x0000≤reg≤0x2000;
Операция: (reg1)=(reg2)
Флаги:
Описание: Запись содержимого регистра reg2 в регистр reg1

31. incr Инкрементирование содержимого регистра
Синтаксис: incr reg
Операнды: 0x0000≤reg≤0x2000;
Операция: (reg)=(reg) +1
Флаги: Z,C,N,OV
Описание: Инкрементирование содержимого регистра reg

32. decr Декрементирование содержимого регистра
Синтаксис: decr reg
Операнды: 0x0000≤reg≤0x2000;
Операция: (reg)=(reg) - 1
Флаги: Z,C,N,OV
Описание: Декрементирование содержимого регистра reg

2.3 Специальные команды

33. not Инверсия содержимого регистра
Синтаксис: not reg
Операнды: 0x0000≤reg≤0x2000;
Операция: (reg)=~(reg)
Флаги: Z, N
Описание: Инвертирование регистра reg

34. neg Отрицание +1
Синтаксис: neg reg
Операнды: 0x0000≤reg≤0x2000;
Операция: (reg)= ~(reg)+1
Флаги: Z, N
Описание: Перевод содержимого регистра в дополнительный код

35. swapr Поменять местами байты регистра
Синтаксис: swapr reg
Операнды: 0x0000≤reg≤0x2000;
Операция:
Флаги:
Описание: Поменять местами старший и младший байты регистра

3. Бит ориентированные команды

1. bset Установить бит
Синтаксис: bset reg, n
Операнды: 0x0000≤reg≤0x0FFF; 0x00≤n≤0x0F;
Операция: 1→([reg],n)
Флаги:
Описание: Установить бит n регистра reg

2. bclr Очистить бит
Синтаксис: bclr reg, n
Операнды: 0x0000≤reg≤0x0FFF; 0x00≤n≤0x0F;
Операция: 0→([reg],n)
Флаги:
Описание: Очистить бит n регистра reg

3. btgl Инверсия бита
Синтаксис: btgl reg, n
Операнды: 0x0000≤reg≤0x0FFF; 0x00≤n≤0x0F;
Операция: ~([reg],n)→([reg],n)
Флаги:
Описание: Инвертировать бит n регистра reg

4. Команды работы с подпрограммами

1. call Вызов подпрограммы
Синтаксис: call k
Операнды: 0x0000≤k≤0xFFFF;
Операция: PC+1→[SP]; k→ PC; SP+1→SP
Флаги:
Описание: Безусловный вызов подпрограммы

2. return Возврат из подпрограммы
Синтаксис: return
Операнды:
Операция: SP-1→SP; [SP] →PC
Флаги:
Описание: Безусловный возврат из подпрограммы

5. Команды переходов	
1. jmp Безусловный переход на адрес Синтаксис: jmp k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Безусловный переход на адрес	
2. jz Переход на адрес, если установлен флаг Z Синтаксис: jz k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Выполнить переход на адрес k если установлен флаг Z	
3. jnz Переход на адрес, если флаг Z не установлен Синтаксис: jnz k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Выполнить переход на адрес k если не установлен флаг Z	
4. jc Переход на адрес, если установлен флаг C Синтаксис: jc k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Выполнить переход на адрес k если установлен флаг C	
5. jnc Переход на адрес, если флаг C не установлен Синтаксис: jnc k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Выполнить переход на адрес k если не установлен флаг C	
6. jn Переход на адрес, если установлен флаг N Синтаксис: jn k Операнды: 0x0000≤k≤0xFFFF; Операция: k→ PC Флаги: Описание: Выполнить переход на адрес k если установлен флаг N	

7. jnn Переход на адрес, если флаг N не установлен
Синтаксис: jnn k
Операнды: 0x0000≤k≤0xFFFF;
Операция: k→ PC
Флаги:
Описание: Выполнить переход на адрес k если не установлен флаг N

8. jbs Переход на адрес, если установлен бит регистра
Синтаксис: jbs k,reg,n
Операнды: 0x0000≤k≤0xFFFF; 0x0000≤reg≤0x0FFF; 0x00≤n≤0x0F;
Операция: k→ PC
Флаги:
Описание: Выполнить переход на адрес k если установлен бит N регистра reg

9. jbc Переход на адрес, если не установлен бит регистра
Синтаксис: jbc k,reg,n
Операнды: 0x0000≤k≤0xFFFF; 0x0000≤reg≤0x0FFF; 0x00≤n≤0x0F;
Операция: k→ PC
Флаги:
Описание: Выполнить переход на адрес k если не установлен бит N регистра reg

10. cmpleq Пропустить, если значение регистра равно значению константы
Синтаксис: cmpleq reg, literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF;
Операция: если (reg)=literal,то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если содержимое регистра равно значению константы

11. cmplneq Пропустить, если значение регистра не равно значению константы
Синтаксис: cmplneq reg, literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF;
Операция: если (reg)!=literal,то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если содержимое регистра не равно значению константы

12. cmplgt Пропустить, если значение регистра больше чем значение константы
Синтаксис: cmplgt reg, literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF;
Операция: если (reg)>literal,то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если содержимое регистра больше значения константы

13. cmpllt Пропустить, если значение регистра меньше чем значение константы
Синтаксис: cmpllt reg, literal
Операнды: 0x0000≤reg≤0x0FFF; 0x0000≤literal≤0xFFFF;
Операция: если (reg)<literal,то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если содержимое регистра меньше значения константы

14. cmpreq Пропустить, если значения регистров равны
Синтаксис: cmpreq reg1, reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)=(reg2),то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если значения регистров равны

15. cmprneq Пропустить, если значения регистров не равны
Синтаксис: cmprneq reg1, reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)!=(reg2),то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если значения регистров не равны

16. cmprgt Пропустить, если больше
Синтаксис: cmprgt reg1, reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)>(reg2),то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если значение регистра reg1 больше значения регистра reg2

17. cmprlt Пропустить, если меньше
Синтаксис: cmprlt reg1, reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)<(reg2),то PC+2→ PC
Флаги:
Описание: Пропустить следующую команду, если значение регистра reg1 меньше значения регистра reg2

18. jeqrr Переход, если значения регистров равны
Синтаксис: jeqrr reg1, reg2, label
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)=(reg2),то label→ PC
Флаги:
Описание: Переход на метку label, если значение регистра reg1 равно значению регистра reg2

19. jneqrr Переход, если значения регистров не равны
Синтаксис: jneqrr reg1, reg2, label
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)!=(reg2),то label→ PC
Флаги:
Описание: Переход на метку label, если значение регистра reg1 не равно значению регистра reg2

20. jeqrl Переход, если значения регистра равно константе
Синтаксис: jeqrl reg1, litteral, label
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: если (reg1)=literal,то label→ PC
Флаги:
Описание: Переход на метку label, если значение регистра reg1 равно константе literal

21. jneqrl Переход, если значения регистра не равно константе
Синтаксис: jneqrl reg1, litteral, label
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;

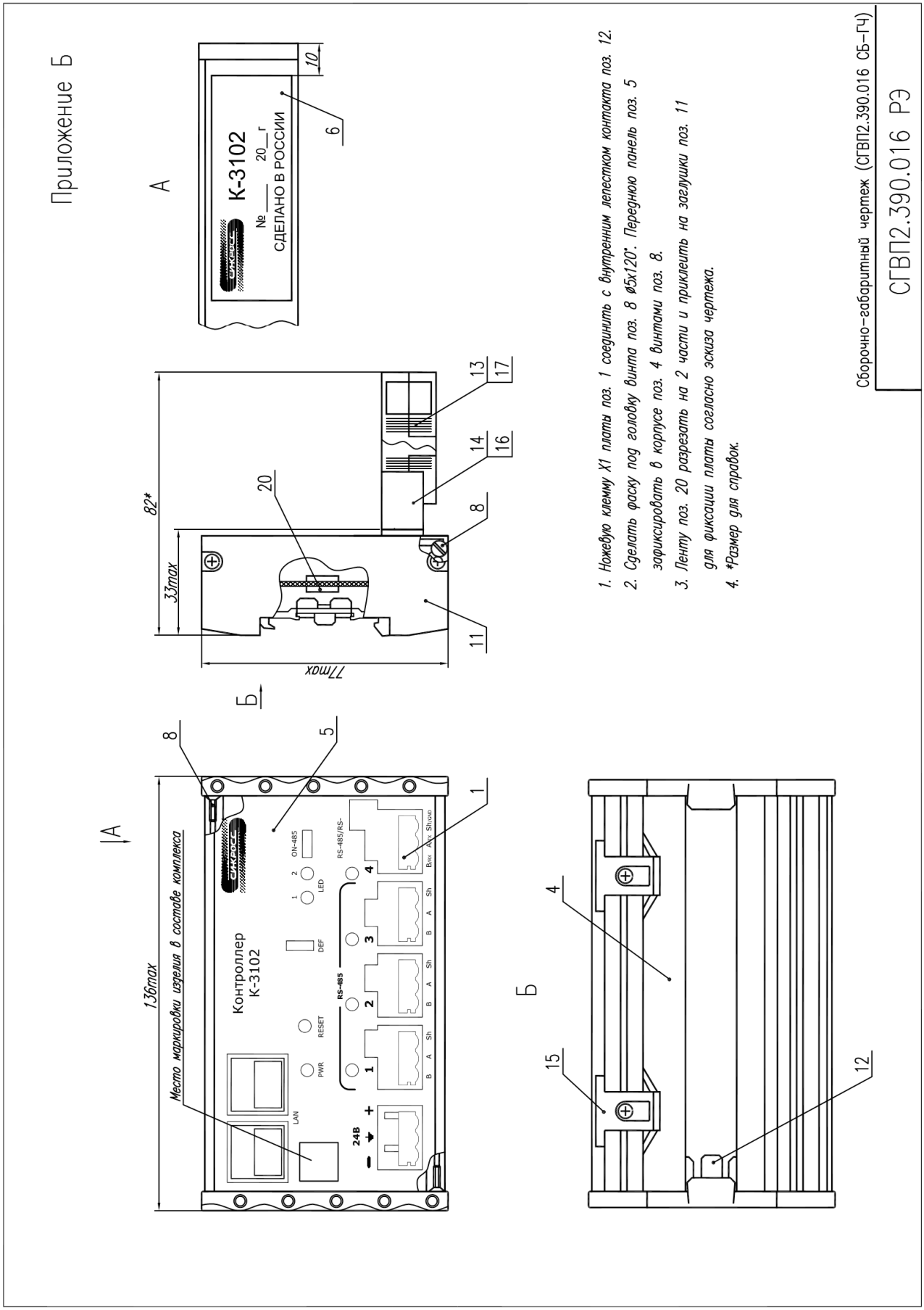
Операция: если (reg1)!=literal,то label→ PC
Флаги:
Описание: Переход на метку label, если значение регистра reg1 не равно константе literal

6. Команды косвенной адресации

1. movMR Скопировать содержимое регистра 2 в регистр с адресом лежащим в регистре 1
Синтаксис: movMR reg1,reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: [reg1]=reg2
Флаги:
Описание: Скопировать содержимое регистра reg1 в регистр с адресом лежащим в регистре reg2

2. movRM Скопировать содержимое регистра с адресом лежащим в регистре 1 в регистр 2
Синтаксис:movRM reg1,reg2
Операнды: 0x0000≤reg1≤0x0FFF; 0x0000≤reg2≤0x0FFF;
Операция: reg1=[reg2]
Флаги:
Описание: Скопировать содержимое регистра с адресом лежащим в регистре 1 в регистр 2

ПРИЛОЖЕНИЕ Б. СБОРОЧНО-ГАБАРИТНЫЙ ЧЕРТЕЖ.



ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводи-тельного докум. и дата	Подп.	Дата
	Измененных	Замененных	Новых	Аннулированных					