



**СЕРИЯ МОДУЛЕЙ К-3XXX**

**КОНТРОЛЛЕР К-3106**

**РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ**

**СГВП2.390.018 РЭ**

2012 г.

## СОДЕРЖАНИЕ

	Стр.
1 НАЗНАЧЕНИЕ .....	4
2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....	5
3 КОМПЛЕКТНОСТЬ .....	10
4 УСТРОЙСТВО И РАБОТА ИЗДЕЛИЯ .....	10
5 ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ .....	15
6 УКАЗАНИЕ МЕР БЕЗОПАСНОСТИ .....	17
7 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ .....	17
8 ТРАНСПОРТИРОВАНИЕ И ХРАНЕНИЕ .....	18
9 ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА .....	18
10 СВЕДЕНИЯ О РЕКЛАМАЦИЯХ.....	19
Приложение А	Сборочно-габаритный чертеж;
Приложение Б	Схема электрическая подключения контроллера К-3106;
Приложение В	Инструкция по настройке и программированию.

Настоящее руководство по эксплуатации, предназначено для ознакомления с устройством контроллера К-3106 (далее в тексте – контроллер), принципом действия, правилами эксплуатации, монтажа, транспортирования и хранения с целью поддержания его в рабочем состоянии в течение срока эксплуатации.

Документ содержит технические характеристики контроллера, описание принципа действия, информацию необходимую для подключения, настройки, эксплуатации контроллера в составе проектно-компонованных систем пожарной, охранно-пожарной сигнализации и управления пожаротушением.

Список принятых сокращений:

АЦП	– аналогово-цифровой преобразователь;
БПЗ	– блоки защиты от перегрузки;
ВК	– выходные ключи;
ИП	– источник питания;
ИУ	– инструментальный усилитель;
КЗ	– короткое замыкание;
КТС	– комплекс технических средств охранно-пожарной сигнализации;
М	– мультиплексор;
МК	– микроконтроллер;
ОЗУ	– оперативно-запоминающее устройство;
ПЗУ	– постоянное запоминающее устройство;
ПИ	– пожарный извещатель;
ПЛК	– программируемый логический контроллер;
ПО	– программное обеспечение;
ППУ	– прибор управления пожарный;
ПЭВМ	– персональная электронно-вычислительная машина;
ФИ	– формирователи интерфейса;
ЦП	– центральный процессор;
ШС	– шлейф сигнализации.

## 1 НАЗНАЧЕНИЕ

1.1 Контроллер, входящий в состав комплекса технических средств охранно-пожарной сигнализации и управления пожаротушением КТС-2000 (ТУ 4371-006-12221545-01), предназначен для контроля шлейфов сигнализации защищаемых объектов с установленными в них неадресными охранными, пожарными и охранно-пожарными извещателями как при непосредственном подключении ШС к контроллеру, так и через удаленные модули по интерфейсу RS-485; выдачи тревожных извещений при срабатывании извещателей или нарушении целостности линий связи ШС (обрыв, короткое замыкание) на светозвуковые оповещатели через релейные выходы и в аппаратуру среднего/верхнего уровня (КТС-2000, ПЭВМ) по интерфейсам RS-485/Ethernet; управления дискретными сигналами с помощью твердотельного реле; локального и централизованного управления защищаемыми зонами, установками пожаротушения, дымоудаления, речевого оповещения, инженерным оборудованием, свето-звуковыми оповещателями.

1.2 Контроллер предназначен для построения систем малой и средней информативной емкости.

1.3 Контроллер обеспечивает выполнение следующих функций:

- прием электрических сигналов от ручных и автоматических ПИ;
- автоматический контроль целостности линий связи с ПИ с выдачей информации о нарушении в аппаратуру среднего/верхнего уровня;
- преимущественная регистрация и передача во внешние цепи извещения о пожаре по отношению к другим сигналам, формируемым контроллером;
- автоматический контроль работоспособности и состояния контроллера с возможностью выдачи извещения о неисправности во внешние цепи;
- автоматическую передачу отдельных извещений о пожаре, неисправности контроллера;
- формирование стартового импульса запуска ППУ;
- взаимодействие с активными (энергопотребляющими) ПИ и пассивными ПИ;

- выдачу сигналов на автоматическое включение исполнительных устройств систем противопожарной защиты (пожаротушения, дымоудаления, оповещения, инженерного оборудования и т.д.);
- выдачу сигналов на включение исполнительных устройств систем противопожарной защиты при помощи средств дистанционного пуска;
- автоматическое (в том числе дистанционное) отключение и восстановление режима автоматического управления исполнительными устройствами систем противопожарной защиты по направлениям защиты;
- возможность программирования тактики формирования извещения о пожаре и корректировки алгоритма оповещения.

1.4 Контроллер обеспечивает индикацию посредством встроенных светодиодов:

- состояние контроллера;
- наличие напряжения питания;
- состояние обмена по интерфейсам Ethernet/RS-485.

1.5 Контроллер предназначен для работы в составе проектно-компонруемых распределенных систем охранно-пожарной сигнализации с монтажом в стационарном конструктиве – шкафу, приборной стойке КТС и рассчитан на круглосуточный непрерывный режим работы.

1.6 Конструкция контроллера не предусматривает его использование в условиях воздействия агрессивных сред, пыли, а также во взрывопожароопасных помещениях.

## 2 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1 Электропитание контроллера осуществляется от источника питания постоянного тока напряжением 18-36 В. Номинальное напряжение питания - 24 В. Количество вводов питания – 1.

2.2 Информационная емкость контроллера (количество подключаемых шлейфов сигнализации) – 4 (аналоговых).

Информационная емкость контроллера при подключении по сети RS-485 модулей контроля неадресных шлейфов К-3206, может быть расширена до 512.

2.3 Контроллер обеспечивает на входах ШС в дежурном режиме работы постоянное напряжение равное напряжению питания.

2.4 Ток потребляемый контроллером в дежурном режиме, без учета тока потребляемого ШС, не более 250 мА. Максимальный ток потребления каждого шлейфа, не более 400 мА.

2.5 Контроллер обеспечивает отключение питания ШС при токе нагрузки более 400 мА.

2.6 При коротком замыкании одного из ШС прибор обеспечивает на входах остальных шлейфов постоянное напряжение согласно п. 2.3.

2.7 Максимальное сопротивление пожарного ШС, без учета сопротивления выносного элемента, при котором модуль сохраняет работоспособность, не более 1кОм.

2.8 Сопротивление утечки между проводами пожарного ШС, или каждым проводом и землей, не менее 50 кОм.

2.9 Максимальное сопротивление охранного ШС, без учета сопротивления выносного элемента, при котором модуль сохраняет работоспособность, не более 470 Ом.

2.10 Сопротивление утечки между проводами охранного ШС, или каждым проводом и землей, не менее 20 кОм.

2.11 Передача информации центральному контроллеру КТС-2000 осуществляется по интерфейсу RS-485/Ethernet.

2.12 Параметры передачи данных по интерфейсу RS-485:

- скорость 2400-230400;
- протокол Modbus RTU;
- процедура доступа к шине Master/Slave;
- тип передачи полудуплексный;
- тип линии двухпроводная экранированная витая пара.

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

- длина, не более 1200 м;
- емкость, не более 50 нФ;

- сопротивление, не более 50 Ом;
- сопротивления изоляции, не менее 50 кОм.

#### 2.14 Параметры передачи данных по интерфейсу Ethernet:

- скорость 10/100 Мбит/с;
- протокол Modbus TCP;
- макс. количество поддерживаемых TCP/IP соединений:
  - порт 1 5
  - порт 2 8
- тип передачи дуплексный;
- тип линии витая пара UTP кат.3/5;
- тип разъема 8P8C (RJ-45);
- длина линии, не более 100 м.

#### 2.15 Режим работы – непрерывный, длительный.

#### 2.16 Разветвлённость контроллера (количество коммутируемых цепей) – 8.

При подключении по интерфейсу RS-485 модулей вывода дискретных сигналов К-3202, количество коммутируемых цепей может быть расширено до 500.

Выходные дискретные сигналы формируются твердотельными реле:

- максимальный выходной ток при активной нагрузке не более 0,7 А;
- максимальное коммутируемое напряжение постоянного тока при активной нагрузке не более 36 В;
- минимальная длительность выходных сигналов 0,2 сек.

#### 2.17 Параметры входных дискретных сигналов:

- количество входов дискретных сигналов – 8 (при подключении по интерфейсу RS-485 модулей ввода дискретных сигналов К-3201, может быть расширено до 520);
- номинальное напряжение входных сигналов =24 В;
- диапазон тока входных сигналов 10-20 мА;

#### 2.18 Размер памяти программ - 16 Кбайт, размер памяти данных 4 Кбайта.

2.19 Информативность контроллера (количество видов выдаваемых извещений по интерфейсам RS-485/Ethernet, дискретным выходам), более 5:

В режиме «Дежурный»

- «Норма» – нет нарушения или неисправности зоны;
- «Тревога» – произошло детектирование нарушения зоны;
- «Пожар» – произошло детектирование возгорания в зоне;

В режиме «Диагностика»

- «Неисправность» – внутренняя неисправность контроллера;
- «Обрыв» – произошел обрыв линии связи ШС зоны;
- «Короткое замыкание» – произошло короткое замыкание ШС зоны.

2.20 Электрическая прочность изоляции между гальванически несвязанными группами:

- клеммы питания (=24 В) и клеммы ШС;
- клеммы порта №1 интерфейса RS-485;
- клеммы порта №2 интерфейса RS-485;
- клеммы дискретных портов ввода;
- клеммы дискретных портов вывода;
- клемма защитного заземления,

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

2.21 Электрическое сопротивление изоляции между гальванически несвязанными группами:

- клеммы питания (=24 В) и клеммы ШС;
- клеммы порта №1 интерфейса RS-485;
- клеммы порта №2 интерфейса RS-485;
- клеммы дискретных портов ввода;
- клеммы дискретных портов вывода;
- клемма защитного заземления,

в нормальных климатических условиях не менее 20 МОм.

2.22 Время готовности контроллера с момента подачи питания, с учетом времени на автоматический контроль исправности - не более 5 сек.

2.23 Контроллер обеспечивает хранение в энергонезависимом ПЗУ задан-



ных настроек при исчезновении напряжения в питающей сети.

2.24 Контроллер предназначен для эксплуатации в следующих климатических условиях:

- температура окружающего воздуха, °С от минус 40 до 70;
- относительная влажность воздуха без конденсации влаги, не более, % 93 при 40 °С;
- атмосферное давление, кПа 84-106,7.

2.25 Контроллер устойчив к воздействию синусоидальной вибрации с частотой 10 – 150 Гц, амплитудой 0,075 мм и постоянным ускорением 0,5g.

2.26 Контроллер прочен к воздействию синусоидальной вибрации в диапазоне частот 10 – 150 Гц с амплитудой 0,075 мм и постоянным ускорением 1g.

По помехоэмиссии и устойчивости к промышленным радиопомехам контроллер в составе КТС-2000 соответствует требованиям ГОСТ Р 53325, ГОСТ 50009:

- к наносекундным импульсным помехам – 3 степени жесткости по ГОСТ Р 51317.4.4;

- к контактными и воздушными электростатическим разрядам – 2 степени жесткости по ГОСТ Р 51317.4.2;

- к радиочастотному электромагнитному полю в диапазоне от 30 до 1000 МГц – 2 степени жесткости по ГОСТ Р 51317.4.3.

2.27 Радиопомехи, создаваемые контроллером, не превышают значений класса Б, по ГОСТ Р 51318.22 (СИСПР 22—2006) и ГОСТ Р 50009.

**ВНИМАНИЕ: Качество функционирования контроллера не гарантируется, если электромагнитная обстановка не соответствует требованиям условий эксплуатации.**

2.28 Степень защиты оболочки – IP20 по ГОСТ 14254.

2.29 Средняя наработка на отказ контроллера в дежурном режиме работы, не менее 35000 часов.

2.30 Контроллер является восстанавливаемым и обслуживаемым изделием. Среднее время восстановления не более 6 часов.

2.31 Средний срок службы контроллера не менее 10 лет.

2.32 Габаритные размеры (длина × высота × ширина), не более 180×80×40 мм.

2.33 Масса, не более, 0,4 кг.

2.34 Внешний вид контроллера должен соответствовать сборочно-габаритному чертежу, приведенному в приложении А.

2.35 Типовые схемы подключения контроллера приведены в приложении Б.

### 3 КОМПЛЕКТНОСТЬ

3.1 Комплект поставки контроллера должен соответствовать таблице 1.

Таблица 1

Наименование	Кол-во	Примечание
Контроллер К3106	1	-
Паспорт СГВП2.390.018 ПС	1	-
Розетка Wago	9	-
Руководство по эксплуатации СГВП2.390.018 РЭ	1	На партию контроллеров, направляемых в один адрес, но не более чем на 10

Примечание:

Программа для просмотра и изменения конфигурации модулей серии К-3XXX К310х\_Config, поставляется комплектно с программным обеспечением КТС-2000 и в комплект поставки контроллера не входит.

### 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-3106

4.1.2 На передней панели контроллера расположены разъемы для подключения интерфейсов Ethernet (Lan1, Lan2) и RS-485 (X7, X8), питания (X9), разъемы ввода (XP5) и вывода (XP6) дискретных сигналов, разъемы для подключения ШС (XP1-XP4), DIP переключатели, а также светодиодные индикаторы наличия напряжения питания, обмена данными по интерфейсным портам и программируемые индикаторы (Led1, Led2).

#### 4.2 Принцип работы

4.2.1 Контроллер в составе сети Modbus RTU/TCP, в зависимости от алгоритма прикладного программного обеспечения К-3106, может работать как в качестве ведомого – «Slave», исполняя команды ведущего – «Master» устройства (например, контроллеры К-3101, К-3102, К-2000, ПЛК платформ автоматизации), так и в качестве ведущего – «Master», передавая команды управления ведомым – «Slave» устройствам (например, модуль К-3201, К-3202, К-3206 и др.), принимая данных о их состоянии и проводя логическую обработку сигналов.

4.2.2 Контроллер имеет два независимых порта RS-485 и два независимых порта Ethernet для подключения к локальной сети.

Скорость обмена данными, сетевой адрес интерфейсных портов контроллера устанавливается индивидуально для каждого из портов при помощи прикладного программного обеспечения K310x\_Config.

4.3 Структурная схема контроллера (рисунок 2) содержит следующие основные функциональные узлы:

- источник питания, ИП;
- мультиплексор, М;
- инструментальный усилитель, ИУ;
- выходные ключи, ВК1-ВК4;
- микроконтроллер, МК;
- аналогово-цифровой преобразователь, АЦП;
- оперативно-запоминающее устройство, ОЗУ;
- формирователи интерфейса RS-485, ФИ1-ФИ2;
- блоки защиты от перегрузки ШС, БЗП1-БЗП4;
- центральный процессор, ЦП;
- флэш-память, Flash;
- стек TCP/IP, Stack Lan1-Stack Lan2;
- группа входных каскадов порогового устройства, DI1-DI8;
- группа выходных твердотельных реле, DO1-DO8.

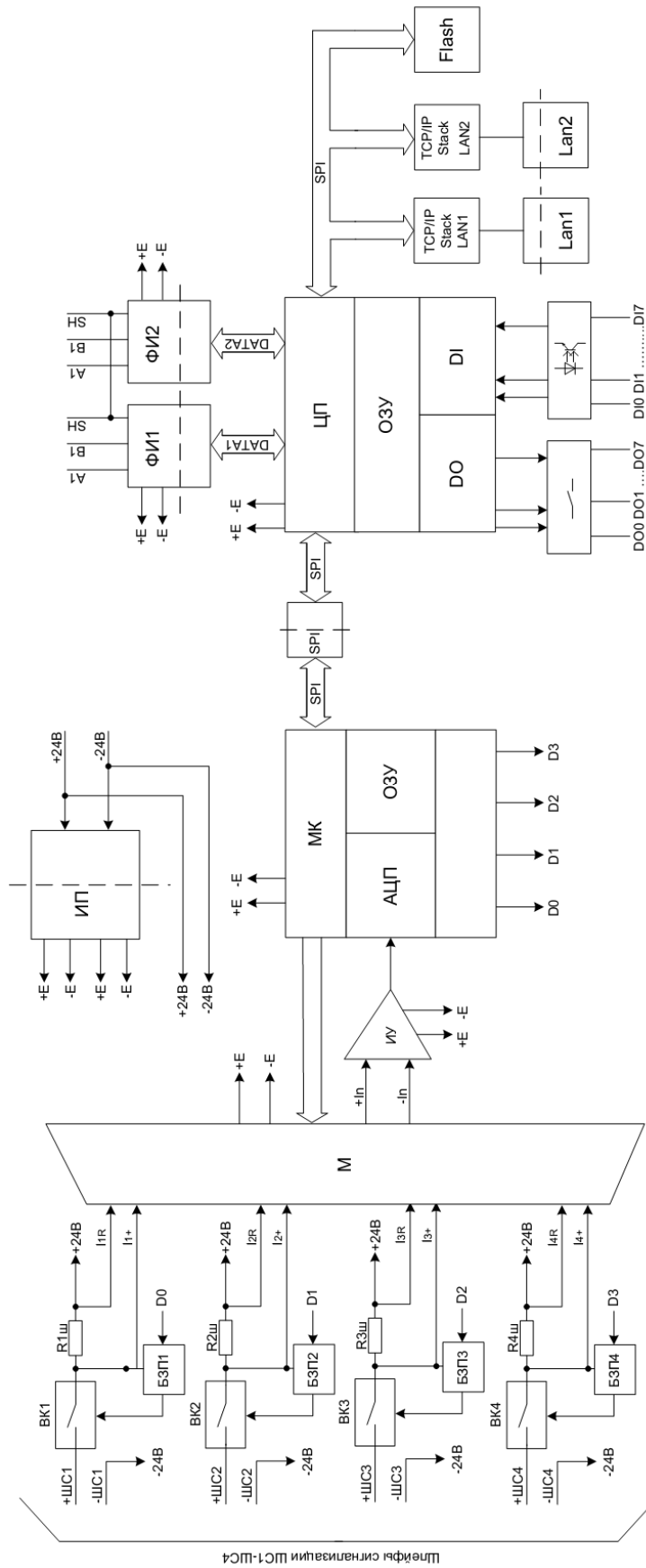


Рисунок 2. Структурная схема контроллера

4.4 Измеренные аналоговые значения тока ШС с выходных ключей ВК1-ВК4, через мультиплексор М, управляемый микроконтроллером МК, поступают на вход инструментального усилителя ИУ. С выхода ИУ сигнал подается на вход 12-разрядного аналого-цифрового преобразователя АЦП, входящего в состав микроконтроллера МК. После получения данных микроконтроллер МК производит их программную фильтрацию, отфильтрованные значения параметров ШС записывает в определенные регистры ОЗУ, доступные для чтения центральному процессору ЦП. При превышении в контролируемых ШС максимального значения тока (более 400 мА), блоки защиты от перегрузки БЗП1-БЗП4, по сигналу микроконтроллера МК обеспечивают отключение ШС на время 0,5 с. По истечению времени блокировки микропроцессор МК проводит повторное включение ШС.

4.5 Энергонезависимая флэш-память используется для хранения конфигурационных параметров контроллера, счетных значений и буфера событий. Параметры конфигурации и содержимое буфера событий передаются по интерфейсу RS-485/Ethernet в аппаратуру среднего/верхнего уровня.

4.6 Световые индикаторы служат для отображения состояния как самого контроллера, так и состояния интерфейса RS-485/Ethernet.

4.7 Центральный процессор ЦП выполняет следующие функции:

- управляет чтением данных из ОЗУ микроконтроллера МК;
- управляет внутренними световыми индикаторами;
- обеспечивает опрос состояния дискретных входов;
- управляет дискретными выходами;
- вводом-выводом данных по интерфейсу Ethernet/RS-485;
- проводит логическую обработку сигналов.

Два последовательных порта центрального процессора ЦП, соединены с формирователями интерфейсов RS-485 ФИ1, ФИ2.

4.8 Твердотельный трансформатор в составе микросхем формирователей интерфейсов обеспечивает гальваническую изоляцию интерфейсных цепей модуля от цепей питания, ШС.

4.9 Источник питания ИП обеспечивает питание цифровых микросхем модуля и гальваническую изоляцию цепей питания модуля от системного источника питания КТС.

4.10 Все цепи контроллера гальванически изолированы от каналов интерфейса RS-485 и внешних цепей питания. Гальваническая развязка обеспечивается применением преобразователей напряжения питания DC/DC и оптронов.

4.11 Согласование нагрузок локальной информационной сети обеспечивается технологическими DIP-переключателями портов интерфейса RS-485, расположенными у разъемов XP 7, XP 8 на лицевой панели контроллера.

4.12 Переключение DIP-переключателей производить инструментом из непроводящего материала. DIP-переключатель в положении ON подключает резистор-терминатор 120 Ом между линиями А и В интерфейса RS-485. Если контроллер является конечным устройством в сети RS-485, указанные DIP-переключатели должны быть установлены в положении ON. Расположение переключателей, разъемов показано на рисунке 1.

## 5 ИСПОЛЬЗОВАНИЕ ПО НАЗНАЧЕНИЮ

### 5.1 Подготовка контроллера к использованию

После получения, длительного хранения или транспортировки контроллера, необходимо провести внешний осмотр транспортной тары и проверить целостность упаковки.

Если целостность тары не нарушена, контроллер следует извлечь из упаковки, провести внешний осмотр на отсутствие механических повреждений и проверить его комплектность.

В случае хранения или транспортирования контроллера при температуре ниже нуля градусов, выдержать его в нормальных условиях в течение 12 часов.

### 5.2 Порядок установки

Перед началом монтажа контроллер следует осмотреть и проверить целостность корпуса, отсутствие повреждений разъемов.

При первоначальной установке контроллера следует выполнить необходимые настройки в соответствии с инструкцией по настройке и программированию согласно приложению В.

Установить контроллер на DIN-рельс, подключить к контроллеру провода соединительных линий ШС и портов дискретного ввода/вывода, питания, интерфейс в соответствии со схемой электрических соединений. Типовые схемы подключения ШС приведены в приложении Б.

Присоединение и отсоединение разъемов модуля должно производиться при отключенном напряжении питания.

### 5.3 Использование контроллера

Подать на контроллер напряжение питания.

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

Порядок контроля работоспособности контроллера, регулирования (настройки), приведения изделия в исходное состояние приведены в приложении В.

Перечень режимов работы, порядок и правила перевода изделия с одного режима работы на другой определяется руководством по эксплуатации на систему, в состав которой входит контроллер, и зависит от варианта прикладного программного обеспечения.

### 5.4 Возможные неисправности и методы их устранения

В случае неисправности контроллера в первую очередь отключить его напряжение питания.

Краткий перечень возможных неисправностей и способы их устранения приведены в таблице 2

Таблица 2

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/Ethernet	Проверить целостность и отсутствие разрывов линии интерфейса связи RS-485/Ethernet

При возникновении прочих более сложных неисправностей их устране-



ние может проводиться только на предприятии - изготовителе подготовленными специалистами.

## 6 УКАЗАНИЕ МЕР БЕЗОПАСНОСТИ

6.1 По способу защиты человека от поражения электрическим током контроллер относится к III классу по ГОСТ 12.2.007.0.

6.2 Конструкция контроллера обеспечивает его пожарную безопасность в аварийном режиме работы и при нарушении правил эксплуатации согласно ГОСТ 12.1.004-91.

6.3 Контроллер не имеет цепей, находящихся под опасным напряжением.

6.4 Монтаж и техническое обслуживание контроллера должны производиться лицами, изучившими настоящее руководство по эксплуатации и имеющими III квалификационную группу по технике безопасности.

## 7 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

7.1 Техническое обслуживание контроллера производится по планово-предупредительной системе, которая предусматривает годовое техническое обслуживание. Работы по годовому техническому обслуживанию включают:

- проверку внешнего состояния контроллера на отсутствие повреждений;
- проверку надежности крепления контроллера, состояния внешних монтажных проводов, контактных соединений;
- удаление загрязнений, пыли и влаги: скопление пыли удаляйте продувкой сухим воздухом и мягкой тканью, влагу – сухой мягкой тканью;
- проверку работоспособности и алгоритма работы контроллера.

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

7.2 При проведении технического обслуживания необходимо соблюдать меры безопасности, указанные в разделе 6 настоящего руководства по эксплуатации.

## 8 ТРАНСПОРТИРОВАНИЕ И ХРАНЕНИЕ

8.1 Контроллер следует транспортировать любым видом транспорта в крытых транспортных средствах (в железнодорожных вагонах, закрытых автомашинах, герметизированных отапливаемых отсеках самолетов и т.д.) в соответствии с правилами транспортирования грузов на соответствующем виде транспорта, на любые расстояния при температуре окружающего воздуха от минус 50 до 50 °С и относительной влажности до 98 % при температуре 35 °С.

8.2 При транспортировании упаковка контроллера должна быть защищена от прямого воздействия атмосферных осадков.

8.3 После погрузки в транспортное средство, ящик должен быть закреплен, с целью исключения возможности его произвольного перемещения.

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

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

## 9 ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

9.1 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.

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

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

9.4 Адрес предприятия изготовителя:

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

## 10 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 комплектность.

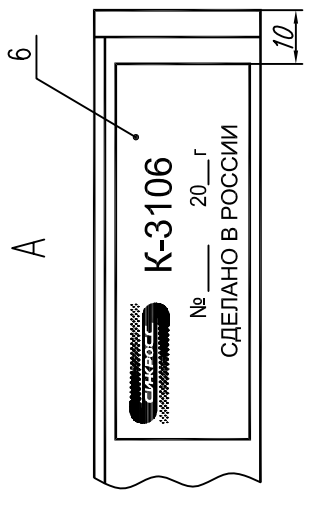
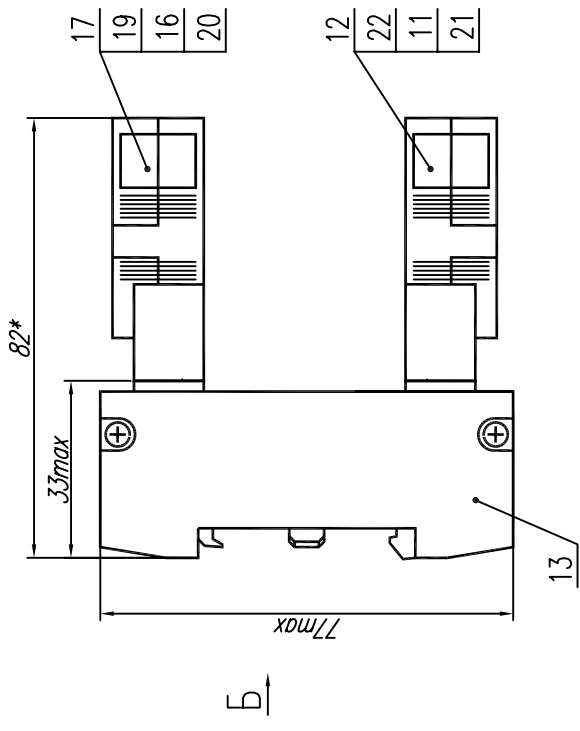
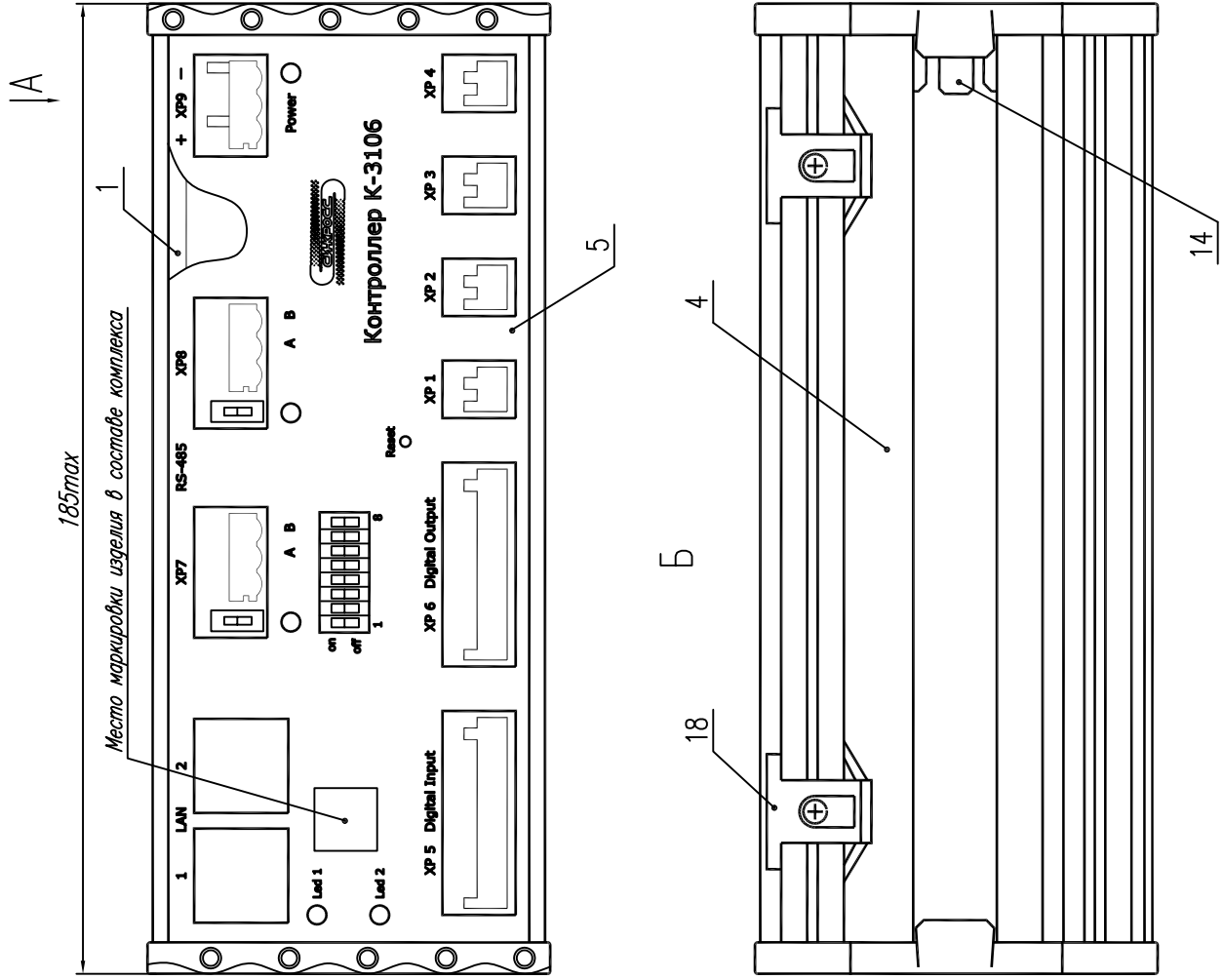
В случае обнаружения повреждений транспортной тары контроллера или комплектности в присутствии грузополучателя составляется соответствующий акт.

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.

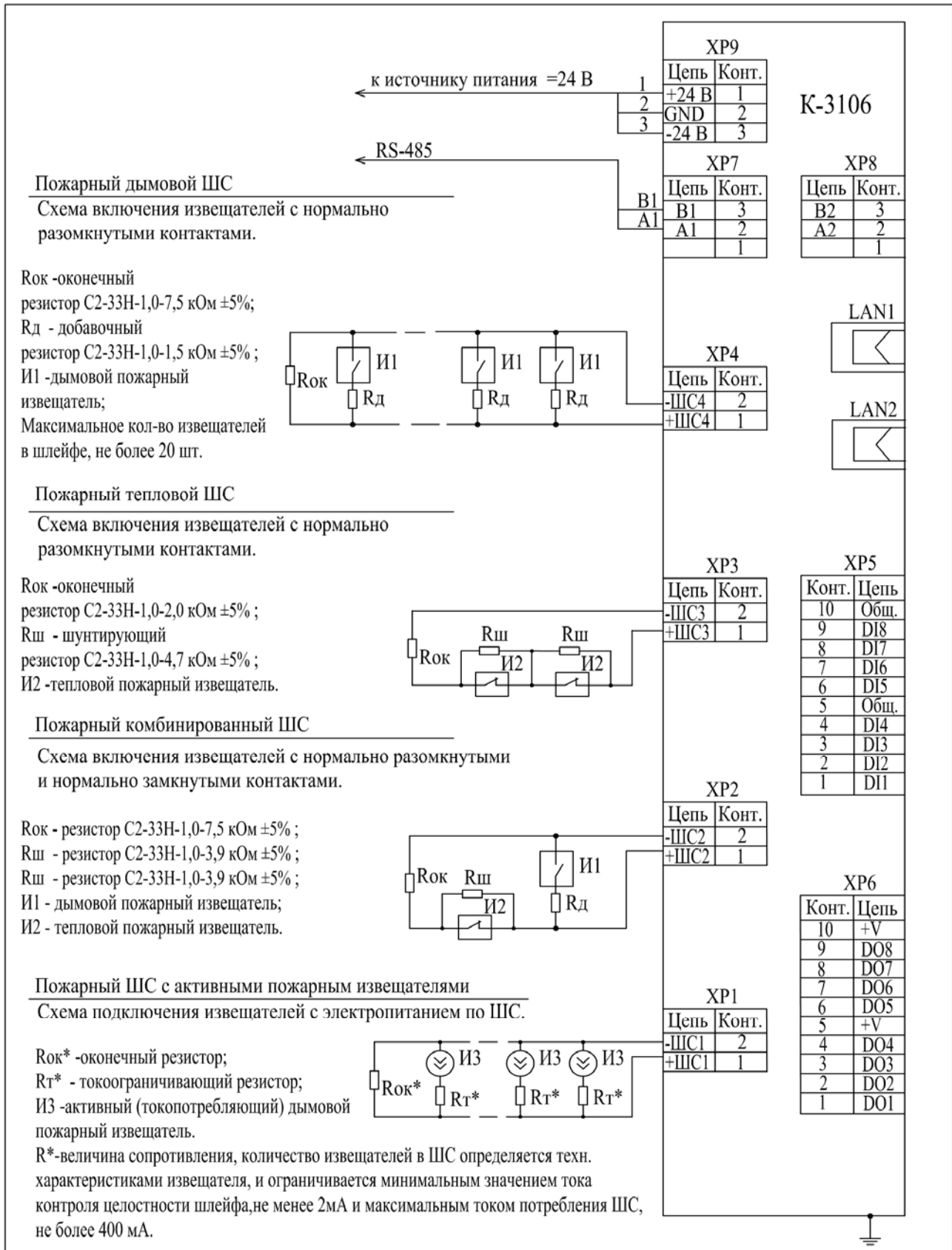
Акт высылается предприятию-изготовителю для устранения выявленных дефектов.

Приложение А



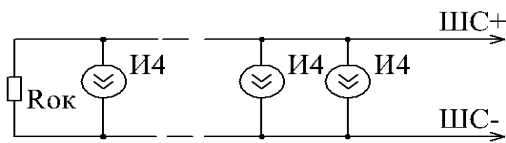
1. Ножевую клемму GND платы поз. 1 соединить с внутренним лепестком контакта поз. 14.
2. \* Размер для справок.

Схема электрическая подключения контроллера К-3106



### Охранный ШС с активными (токопотребляющими) извещателями

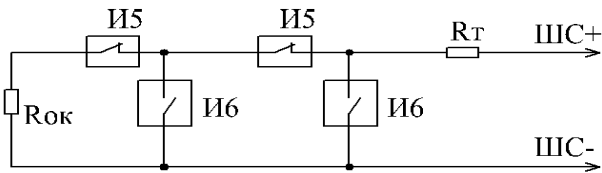
Схема подключения извещателей с электропитанием по ШС.



$R_{ок}$  - оконечный резистор С2-33Н-1,0-7,5 кОм  $\pm 5\%$ ;  
И4 - активный (токопотребляющий) охранный извещатель;  
Максимальное количество токопотребляющих охранных извещателей N для каждого шлейфа определяется по формуле:  
 $N = 400/I_{п}$ , где  $I_{п}$  - максимальный ток потребления одного извещателя в режиме "Тревога", 400 - максимальный ток нагрузки шлейфа, 400 мА.

### Комбинированный охранный ШС

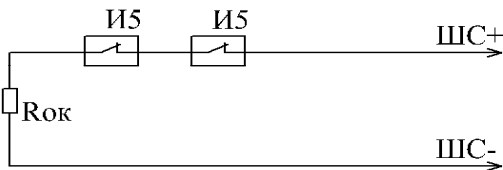
Схема включения извещателей, имеющих на выходе замкнутые и/или разомкнутые контакты, в ШС с оконечным резистором.



$R_{ок}$  - оконечный резистор С2-33Н-1,0-7,5 кОм  $\pm 5\%$ ;  
 $R_{т}$  - токоограничивающий резистор С2-33Н-2,0-470 Ом  $\pm 5\%$ ;  
И5, И6 - охранный извещатель.

### Охранный ШС

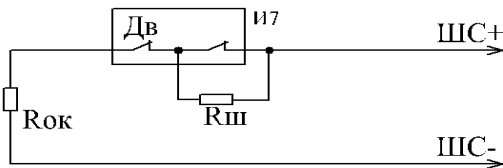
Схема включения извещателей с нормально замкнутыми контактами в контролируемый ШС, с разделением сигналов "Тревога" и "КЗ ШС"



$R_{ок}$  - оконечный резистор С2-33Н-1,0-7,5 кОм  $\pm 5\%$ ;  
И5 - охранный извещатель.

### Охранный ШС

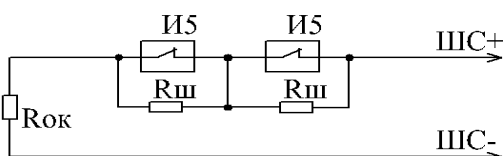
Схема включения охранных извещателей с нормально замкнутыми контактами, с обеспечением контроля их датчиков вскрытия "ДВ" с формированием отдельного извещения о вскрытии корпуса датчика



$R_{ок}$  - оконечный резистор С2-33Н-1,0-2,2 кОм  $\pm 5\%$ ;  
 $R_{ш}$  - шунтирующий резистор С2-33Н-1,0-4,7 кОм  $\pm 5\%$ ;  
И7 - охранный извещатель, с контролем вскрытия корпуса извещателя.

### Охранный ШС

Схема включения извещателей с нормально замкнутыми контактами в контролируемый ШС с обеспечением контоля ШС на обрыв и КЗ



$R_{ок}$  - оконечный резистор С2-33Н-1,0-2,2 кОм  $\pm 5\%$ ;  
 $R_{ш}$  - шунтирующий резистор С2-33Н-1,0-4,7 кОм  $\pm 5\%$ ;  
И7 - охранный извещатель, с контролем вскрытия корпуса извещателя.

## Инструкция по настройке и программированию

### В.1 ВВЕДЕНИЕ

Настоящая инструкция предназначена для ознакомления с контроллером К-3106, принципом его настройки и программирования.

### В.2 ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ

Внешний вид контроллера К-3106 изображен на рисунке В.1.

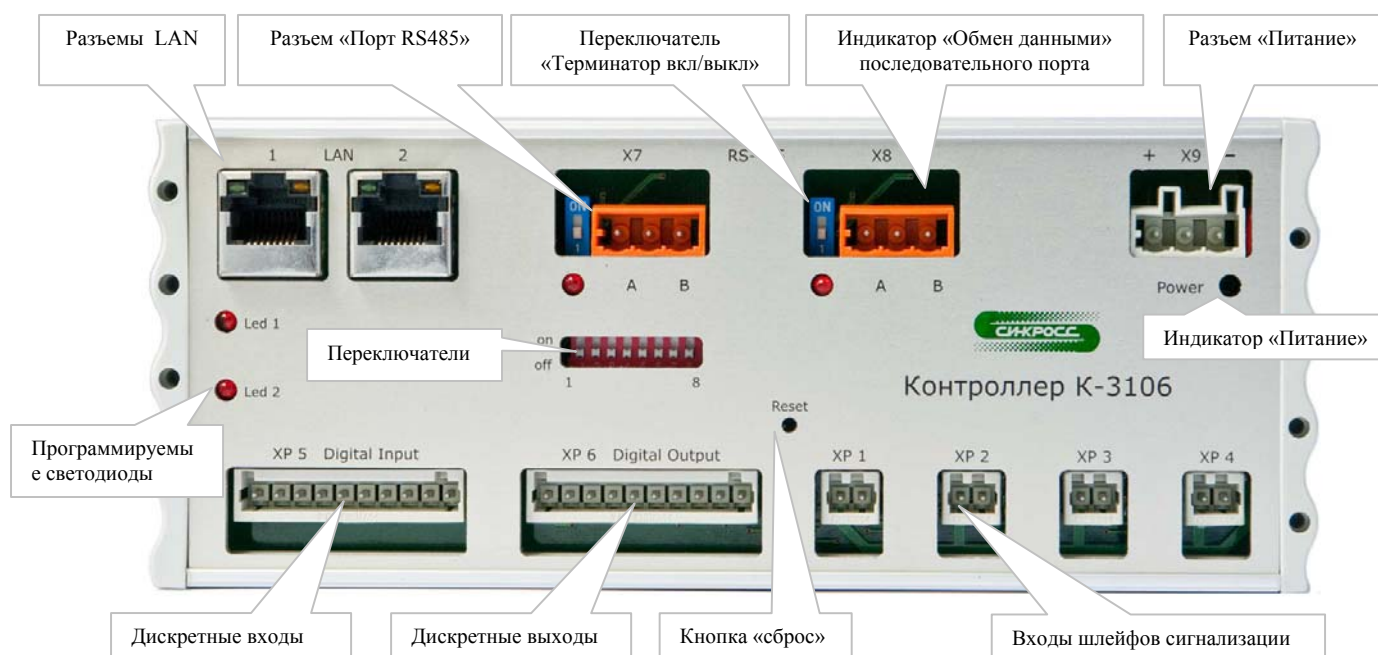


Рисунок В.1. Внешний вид контроллера К-3106

### В.3 РЕЖИМ РАБОТЫ ПОРТОВ

#### В.3.1 Режимы работы последовательные портов

Последовательные порты контроллера К-3106 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 В.3.1.

Таблица В.3.1

№	Функция	Описание	Примечание
1	0x00	Пустая команда	
2	0x03	Чтение регистров	

Продолжение таблицы В.3.1

№	Функция	Описание	Примечание
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	0x50-0x51	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-мастером 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».

### В.3.2 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, 0x50-0x51.

При получении пакета с Modbus функцией 0x6D контроллер на данном порту формирует пакет подтверждения согласно таблице В.3.2. 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 В.3.2.

Алгоритм работы контроллера при получении команд расширенного транзита (функции 0x50-0x51) 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).



Таблица В.3.2

<i>Код</i>	<i>Наименование</i>	<i>Примечания</i>
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

### **В.3.3 Работа в качестве преобразователя Modbus TCP – Modbus RTU**

Контроллер К-3106 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 4 подключений клиентов на порту 1 и до 6 подключений на порту 2. Запросы клиентов Modbus TCP транслируются на последовательные порты согласно таблице транзита каждого порта LAN. Таблица транзита определяет привязку Modbus адресов устройств и последовательных портов контроллера, к которым они подключены.

Если запрашиваемый адрес совпадает с собственным адресом К3106, то запрос обрабатывается контроллером, далее проверяется наличие полученного адреса в таблице транзита. Если устройство с таким адресом найдено в таблице транзита, 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 «Multi Master» в противном случае транзит данных на порт блокируется.

Настройка таблицы транзита осуществляется с помощью ПО конфигурации (см п.В.6.2.2).

## **В.4 ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ**

Программное обеспечение контроллера К-3106 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.

### **В.4.1 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 k3106.inc. Файл k3106.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 k3106.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-3102/6 (см. п.В.6.2.3 Компиляция и запись прикладной программы в контроллер).

#### **В.4.2 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).

#### **В.4.3 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.

### В.4.4 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

```

#### В.4.5 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.

#### В.4.6 Работа с последовательными портами

Программисту доступно 2 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 В.4.1.

Таблица В.4.1. Возможные значения статуса буфера последовательного порта

<i>Обозначение</i>	<i>Значение</i>	<i>Описание</i>
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_TIME EOUT	0xAAA A	«Таймаут ответа». Значение устанавливается если в течении времени таймаута от устройства не получен корректный ответ.
ERROR_ANSWER	0xCCC C	«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 прямо-передающих буферов в разных задачах.*

*2. Не допускаются операции чтения/записи буфера при его статусе NOT\_COMPLETE.*

При наличии свободных прямо-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    jbstat BUF1, wait1, dev_fail, dev_fail, dev_ok      ; переход в
        зависимости
                                                ; от статуса приемного
                                                ; буфера

dev_ok    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

wait1    jbstat  BUF1, wait1, dev1_fail, dev1_fail, dev_ok  ; в порт uart1 через буфер buf1
```

```

dev_ok  SFBufStor BUF1,dev1_answer,0,all ; сохранить ответ ДВВ начиная с
                                           ; адреса

        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   jbstat  BUF1, wait2, dev1_fail, dev1_fail, dev_ok
        SFBufStor BUF1,dev1_answer,0,all
        jmp task 1_2

dev1_fail incr dev_failcount           ; инкрементировать счетчик ошибок
        jmp task1_2

```

#### **V.4.7 Работа с клиентом Modbus TCP**

Программисту доступны по 1 клиентскому сокету Modbus TCP на каждом порту LAN. Чтение и запись регистров с\в серверного устройства выполняется инструкциями SF\_TCPRdreg и SF\_TCPWtreg соответственно.

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

При выполнении инструкции SF\_TCPRdreg или SF\_TCPWtreg происходит соединение с серверным устройством, с последующим обменом данными и разрывом соединения.

В отличие 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 порта LAN 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,IP1,"192.168.23.06",d'502',socket1_stat, 0x05,0x03,
0x0000,0x0a,dev1_answer

m1     jeqrl status_tcp1, NOT_COMPLETE, m1
        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

```

#### **В.4.8 Работа с портами ввода вывода**

Программисту доступно управление 8 дискретными выходами. Для управления выходами используются команды out, outl, oug. В качестве параметров для команды out указывается номер выхода и операция над выходом.

Возможные операции над выходами:

CLEAR	-	выход в состоянии 0;
SET	-	выход в состоянии 1;
TOGGLE	-	инверсия выхода;

Пример: Изменение состояния выхода с частотой 1 Гц

```
Task 1      out DOUT0, TOGGLE
            SFTStrt 1,d'500'          ; запуск таймера 1 с периодом 500 мс
            SFWtTmr 1                 ; ожидание флага таймера 1
            jmp Task 1
```

Для одновременной установки выходов используются команды outl (запись константы в порт вывода) и outr (запись значения регистра в порт вывода).

Для работы с дискретными входами контроллера используются команды:

- movrdi- прочитать значение входов в регистр,
- incopy - копирование состояния входа в бит регистра
- jinsel - переход если вход в состоянии 1
- jinclr - переход если вход в состоянии 0

Описание команд работы с входами контроллера приведено в п. В.6.

#### **В.4.9 Работа с переключателями**

Для работы с переключателями, расположенными на передней панели, используются команды movrsw, swcopy, jswset, jswclr. Описание команд работы с переключателями приведено в п. Г6.

#### **В.4.10 Работа с входами контроля шлейфа**

Контроллер К-3106 имеет в составе 4 входа контроля шлейфа сигнализации с защитой от перегрузки по току. Управление шлейфом и контроль параметров осуществляется при помощи команд интерпретатора.

Подача/снятие питания на/со шлейфов выполняется командами outa, outla, outra. Измеренное значение тока потребления шлейфа хранится во внутренних регистрах контроллера. Чтение значения тока потребления шлейфа осуществляется командой movrai с указанием номера шлейфа и адреса регистра назначения. Значение тока потребления шлейфа выдается в единицах 0,1 мА в диапазоне измерения 0 – 5000. При наличии перегрузки по току, одновременно со снятием питания со шлейфа в регистр записывается значение 0xFFFF.

Описание команд работы со шлейфами сигнализации приведено в п. В.5.

## В.5. СПИСОК КОМАНД

### В.5.1. Системные команды

Таблица В.5.1

<i>№</i>	<i>Наименование</i>	<i>Описание</i>	<i>Флаги</i>
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	

### В.5.2. Словно ориентированные команды

Таблица В.5.2

<i>№</i>	<i>Наименование</i>	<i>Описание</i>	<i>Флаги</i>
		<b>1 Логические команды</b>	
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	orrllr	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
		<b>2 Арифметические команды</b>	
28	addlrnc	Суммирование содержимого регистра и константы	Z,N,C,OV



Продолжение таблицы В.5.2

<b>№</b>	<b>Наименование</b>	<b>Описание</b>	<b>Флаги</b>
29	addrnc	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
		<b>2.3 Команды сдвига</b>	
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
		<b>2.4 Специальные команды</b>	
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	
		<b>3. Бит ориентированные команды</b>	
53	bset	Установить бит	
54	bclr	Очистить бит	
55	btgl	Инверсия бита	
	bcopy	копирование бита m регистра reg2 в бит n регистра reg1	
		<b>4. Команды работы с подпрограммами</b>	
56	call	Вызов подпрограммы	
57	return	Возврат из подпрограммы	
		<b>5. Команды переходов</b>	
58	jmp	Безусловный переход на адрес	

Продолжение таблицы В.5.2

<i>№</i>	<i>Наименование</i>	<i>Описание</i>	<i>Флаги</i>
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	
		<b>6. Команды косвенной адресации</b>	
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	
		<b>7. Команды работы с портами ввода/вывода</b>	
77	out	Установка выхода (clear - выкл, 1-вкл, 2- инверсия)	
78	outl	Запись константы в порт вывода	
79	outr	Запись содержимого регистра в порт вывода	
80	movrdi	Прочитать значение дискретных входов в регистр	
81	incopy	Копирование состояния входа в бит регистра	
82	jinsset	Переход на адрес, если вход в состоянии 1	
83	jinclr	Переход на адрес, если вход в состоянии 0	
84	movrsw	Прочитать значение состояния переключателей в регистр	
85	swcopy	Копирование состояние переключателя в бит регистра	
86	jswset	Переход на адрес, если переключатель в состоянии ON	
87	jswclr	Переход на адрес, если переключатель в состоянии OFF	
88	outa	Подача/снятие питания на/с шлейф сигнализации	
89	outla	Запись константы в порт управления шлейфами сигнализации	

Продолжение таблицы В.5.2

<i>№</i>	<i>Наименование</i>	<i>Описание</i>	<i>Флаги</i>
90	Outra	Запись значения регистра в порт управления шлейфами сигнализации	
91	movrai	Прочитать значение тока шлейфа в регистр	

Описание инструкции приведено в п. В.7.

## **В.6 НАСТРОЙКА**

### **В.6.1 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

- для порта 1 - 192.168.10.1;
- для порта 2 - 192.168.10.2;

Последовательные порты контроллера по умолчанию устанавливаются в режим «Slave» с адресами 0x00.

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

### **В.6.2 Изменение настроек при помощи программы конфигурации**

Настройка контроллера, а также запись прикладной программы, осуществляется при помощи программы конфигурации. Для загрузки программы приложения используется только 1 порт LAN.

#### **В.6.2.1 Установка связи**

Обмен данными между контроллером и программой конфигурации производится по сети Ethernet. Для конфигурирования контроллера используется служебный сокет (порт) с номером 5001.

Для изменения настроек подключите контроллер к локальной сети, подайте питание. Запустите программу конфигурации. В появившемся окне (рисунок В.2) выберете тип контроллера К-3106.

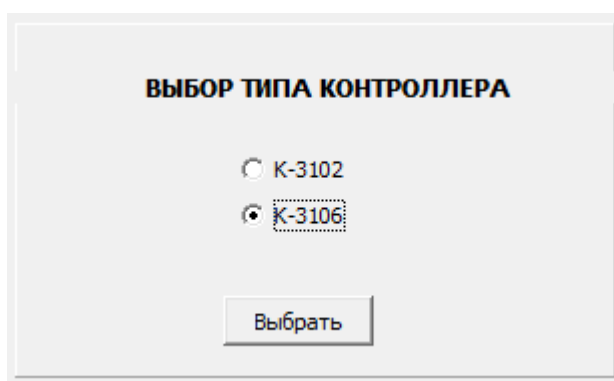


Рисунок В.2.

Внешний вид программы конфигурации после запуска представлен на рисунке В.3.

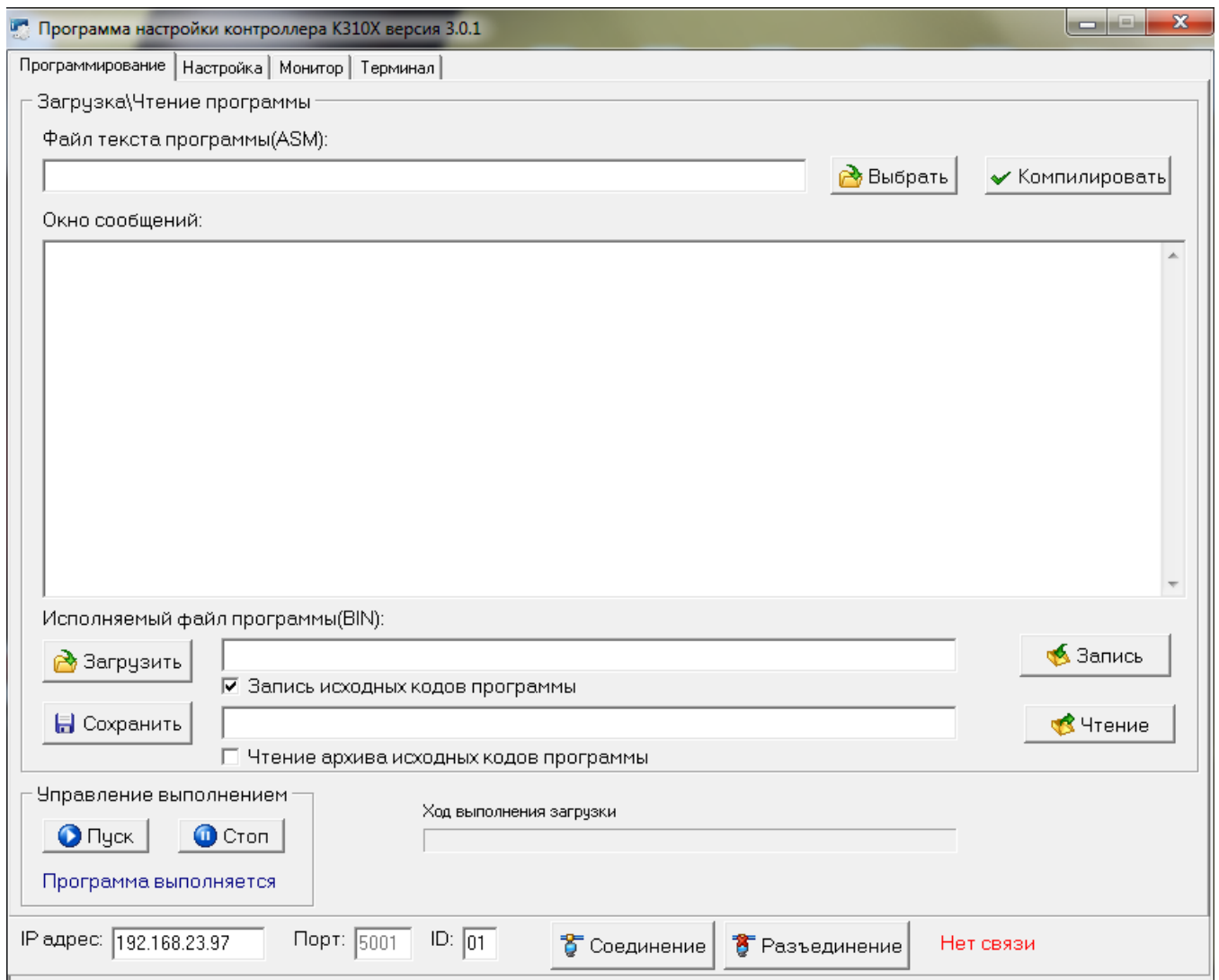


Рисунок В.3. Внешний вид программы конфигурации при запуске.

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»

### В.6.2.2 Настройка параметров

Для установки параметров порта выберите вкладку «Настройка». На экране отобразится окно рисунок В.4.

Программа настройки контроллера К310Х версия 3.0.1

Программирование | **Настройка** | Монитор | Терминал

ПАРАМЕТРЫ ПОРТА LAN1

Таблица транзита данных на порты RS-485

IP адрес:  Порт 1:

Маска:  Порт 2:

MAC адрес:

ID Modbus TCP:  ID:  добавить/удалить

ПАРАМЕТРЫ ПОРТА LAN2

Таблица транзита данных на порты RS-485

IP адрес:  Порт 1:

Маска:  Порт 2:

MAC адрес:

ID Modbus TCP:  ID:  добавить/удалить

ПАРАМЕТРЫ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА 1

1 Slave адрес:  Задержка ответа:  Время старт. синхронизации:

Master адрес:  Таймаут ответа:  Порт функции 0x6D:

Скорость:  MaxMaster:  Таймаут захвата линии:

ПАРАМЕТРЫ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА 2

2 Slave адрес:  Задержка ответа:  Время старт. синхронизации:

Master адрес:  Таймаут ответа:  Порт функции 0x6D:

Скорость:  MaxMaster:  Таймаут захвата линии:

IP адрес:  Порт:  ID:    Соединение установлено

Рисунок В.4. 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.В.6.2, 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».

### В.6.2.3 Компиляция и запись прикладной программы в контроллер

Выберите вкладку «Программирование». На экране отобразится окно рисунок В.5.

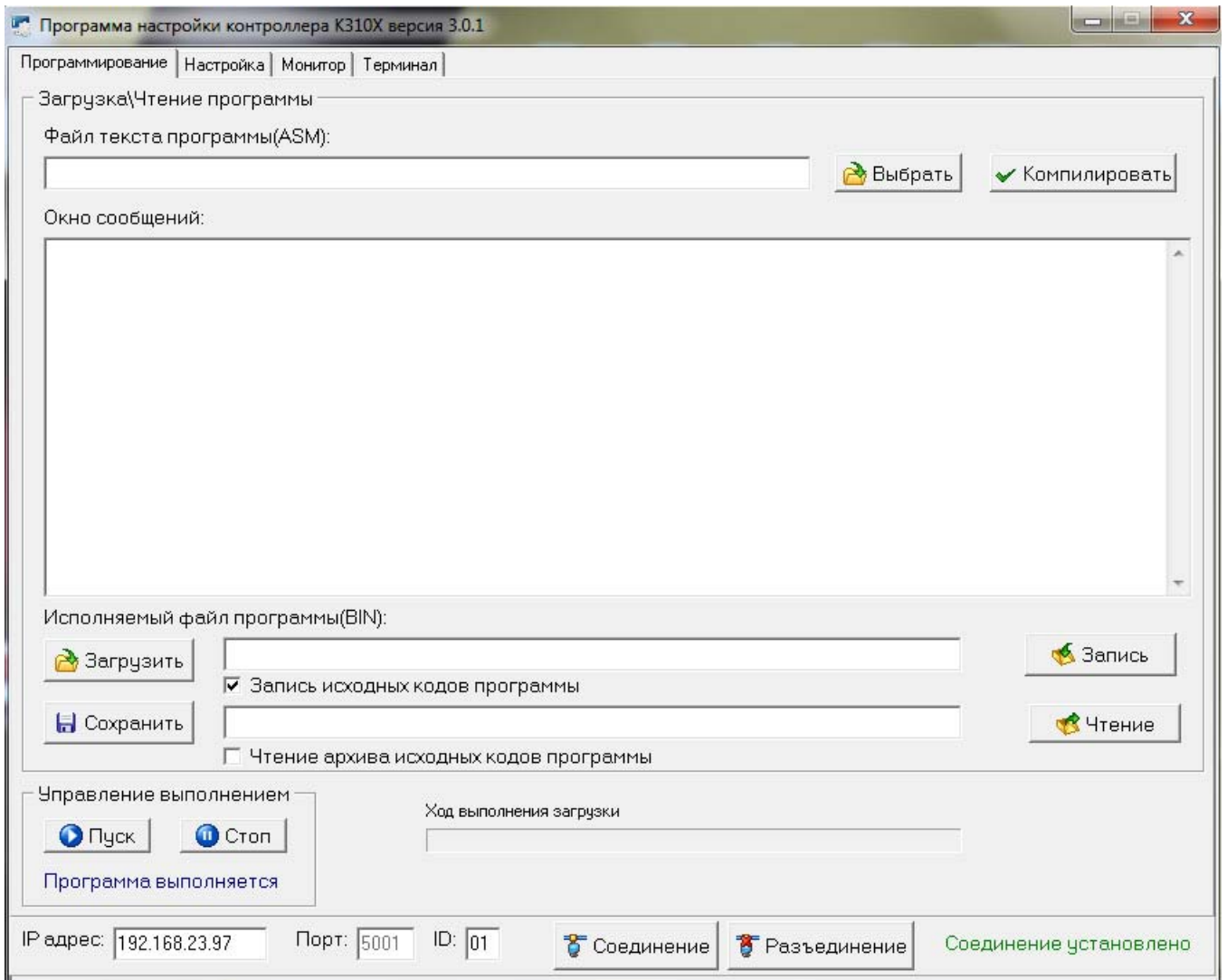


Рисунок В.5 Внешний вид окна «Программирование»

#### В.6.2.3.1 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 и архив исходных кодов программы source.zip

#### В.6.2.3.2 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».

### В.6.2.3.3 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.

### В.6.2.3.3 Монитор

Просмотр состояния памяти данных контроллера осуществляется при помощи вкладки «Монитор» рисунок В.6.

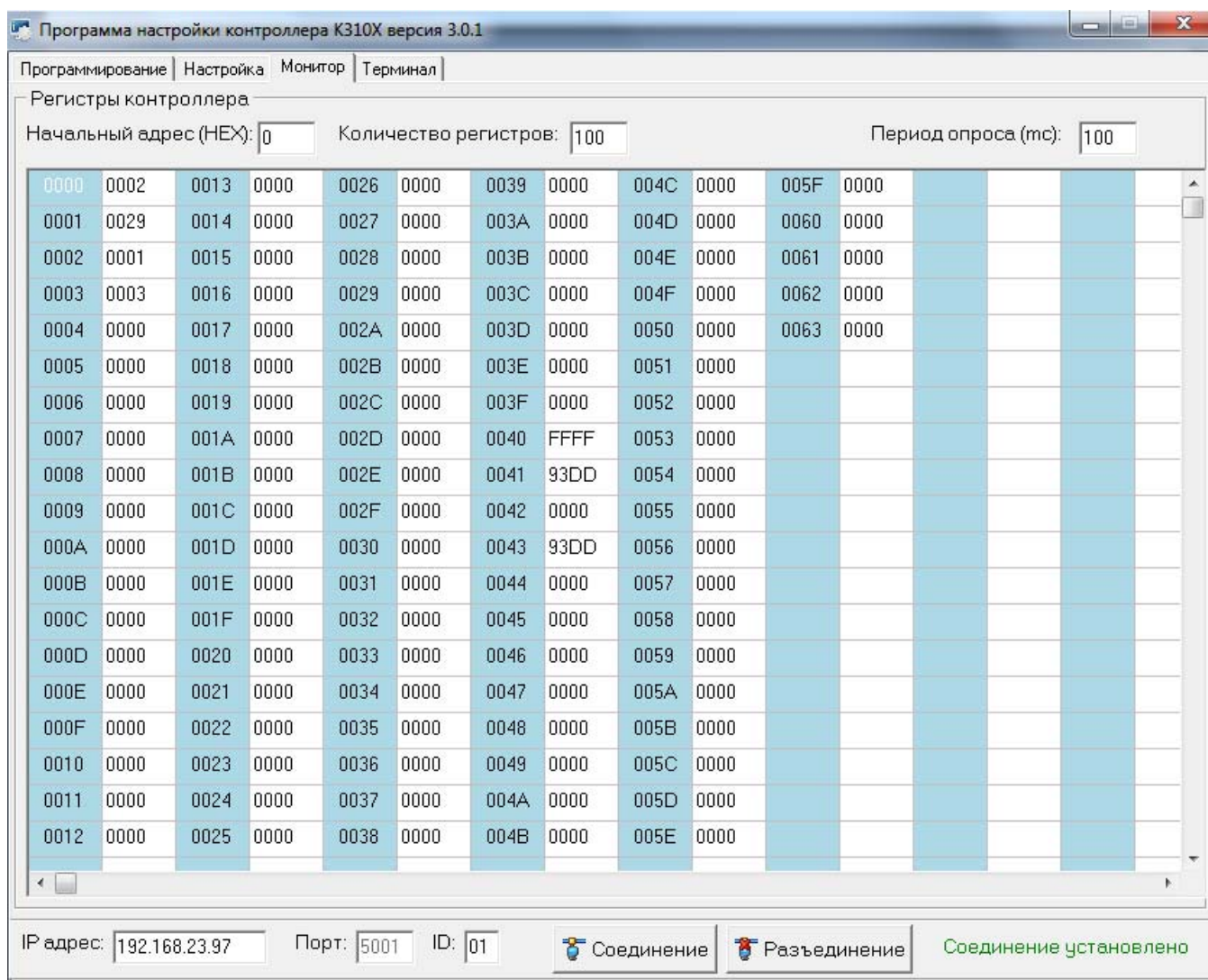


Рисунок В.6 Внешний вид окна «Монитор»

Для установки нового значения регистра двойным щелчком мыши выберете необходимый регистр и в появившемся окне введите новое значение. Подтвердите установку нажатием кнопки «Записать»

### В.6.2.4 Работа с терминалом

Для удаленной работы с терминалом последовательных портов используется вкладка «Терминал» см. рисунок В.7. 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.В.6.2.2).

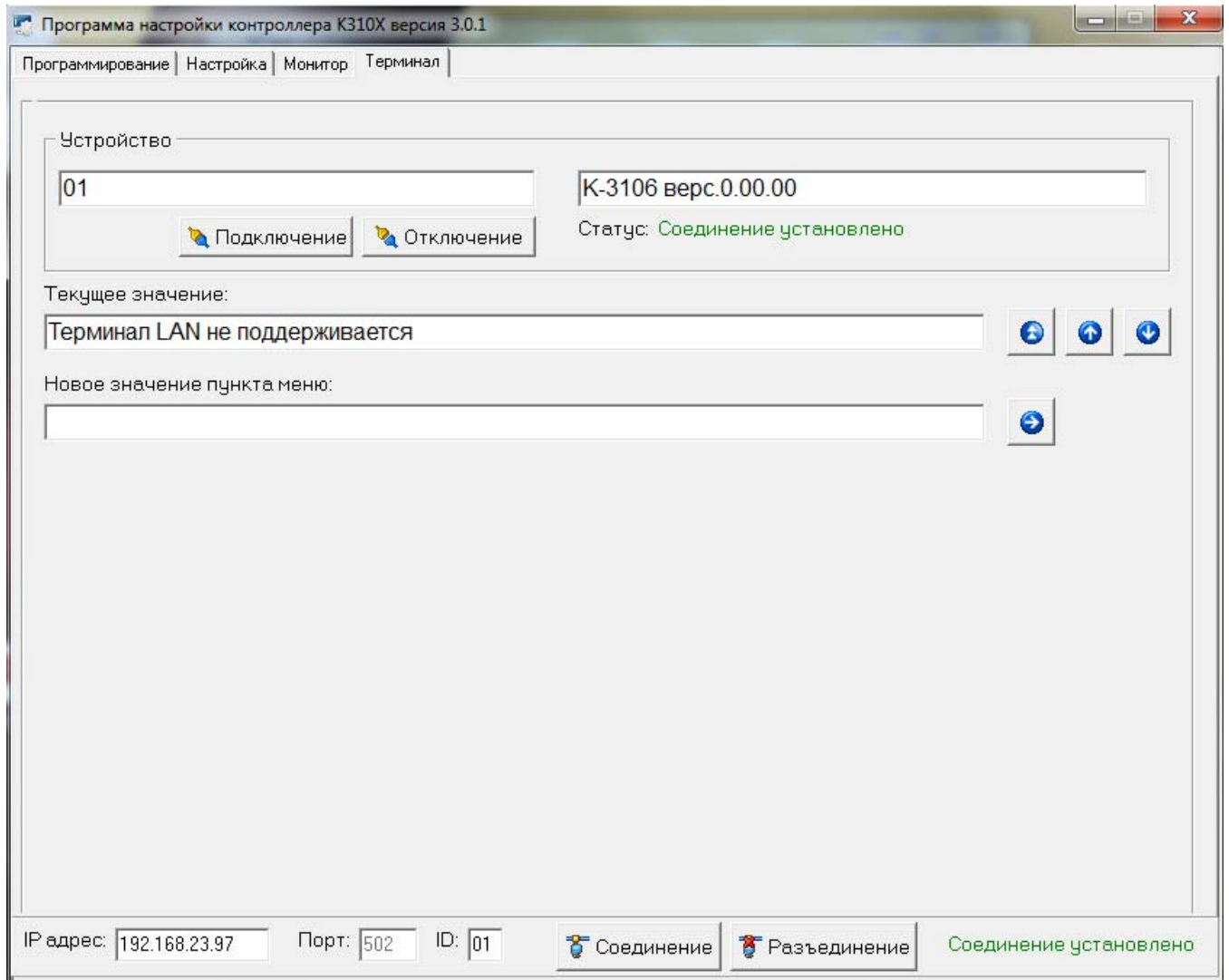


Рисунок В.7 Внешний вид вкладки «Терминал»

Для подключения к удаленному устройству введите адрес в поле «Устройство». Адрес вводится 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

### В.6.3 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.

#### В.6.3.1 Описание пунктов меню терминала

Основное меню терминала имеет вид:

Таблица В.6.1

<i>Пункт меню</i>	<i>Описание</i>
01/05 Настройка порта 1 (RS-485)	Используется для доступа к настройкам последовательного порта
02/05 Настройка порта 2 (RS-485)	
03/05 Настройка порта LAN 1 04/05 Настройка порта LAN 2	Используется для доступа к настройкам порта LAN
05/05 Запись команды	Используется для доступа к сервисным функциям устройства

Подменю COM x

Таблица В.6.2

<i>Пункт меню</i>	<i>Описание</i>
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, 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.

Продолжение таблицы В.6.2

<i><b>Пункт меню</b></i>	<i><b>Описание</b></i>
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

Таблица В.6.3

<i><b>Пункт меню</b></i>	<i><b>Описание</b></i>
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)
MAC адрес	MAC адрес устройства
Применить и выйти	Сохранение настроек
Отмена	Выход в основное меню

Подменю SYSTEM

Таблица В.6.4

<i><b>Пункт меню</b></i>	<i><b>Описание</b></i>
Запись команды	Используется для ввода сервисных команд
Отмена	Выход в основное меню

## **В.7 ОПИСАНИЕ ИНСТРУКЦИЙ**

### **В.7.1. Системные команды**

<b>1. start_task</b> Запуск задачи
Синтаксис: start_task n, label
Операнды: 0<n<6; 0x0000<offset<0x4FFF
Описание: Запускает задачу с номером n с начальным адресом offset
<b>2. end_main_task</b> Завершение основной задачи
Синтаксис: end_main_task
Описание: Останавливает выполнение системной задачи

<b>3. SFTStrt Запуск таймера</b>
Синтаксис: SFTStrt n, time
Операнды: $0 < n < 17$ ; $0x0000 < time < 0xFFFF$
Описание: Резервирует и запускает таймер n с периодом работы time
<b>4. SFTStat Проверка состояния таймера</b>
Синтаксис: SFTStrt n
Операнды: $0 < n < 17$
Флаги: Z
Описание: Проверяет состояние таймера n выставляет флаг Z если таймер завершил отсчет
<b>5. SFWtTmr Ожидание таймера</b>
Синтаксис: SFWtTmr n
Операнды: $0 < n < 17$
Описание: Останавливает выполнение задачи до тех пор, пока таймер n не завершит отсчет
<b>6. SFledon Включение светодиода</b>
Синтаксис: SFledon n
Операнды: $0 < n < 3$
Описание: Включает светодиод n
<b>7. SFledoff Выключение светодиода</b>
Синтаксис: SFledoff n
Операнды: $0 < n < 3$
Описание: Выключает светодиод n
<b>8. SFledinv Инверсия состояния светодиода</b>
Синтаксис: SFledinv n
Операнды: $0 < n < 3$
Описание: Выключает светодиод n
<b>9. SFRdreg Чтение регистров из подчиненного</b>
Синтаксис: SFRdreg port, buf, spath, path, func, offset, nreg
Операнды: $0 < port < 2$ ; $1 < buf < 16$ ; $0 < spath$ ; path=" " (ascii строка); func=0x03 или 0x04; $0x0000 \leq offset \leq 0xFFFF$ ; $0x0000 < nreg \leq 0x007E$
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
<b>10. SFWrreg Запись регистров подчиненного</b>
Синтаксис: SFWrreg port, buf, spath, path, dstadr, srsadr, nreg
Операнды: $0 < port < 2$ ; $1 < buf < 16$ ; spath>0; path=" " (ascii строка); $0x0000 \leq offset \leq 0xFFFF$ ; $0x0000 \leq dstadr \leq 0xFFFF$ ; $0x0000 \leq srsadr \leq 0xFFFF$ ; $0x00 < nreg \leq 0x7E$
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

<b>11. SFWrData Запись данных в подчиненного</b>
Синтаксис: SFWrData port, buf, srsadr, nbyte
Операнды: 0<port<2; 1 <buf<16; 0x0000≤ srsadr ≤0x0FFF; 0x00< nbyte≤0xFF
Описание: Записать в порт port через буфер buf данные, начиная с адреса srsadr количеством байт nbyte.

<b>12. SFBufstat Чтение статуса буфера</b>
Синтаксис: SFBufstat buf, adr
Операнды 1 <buf<16; 0x0000≤ adr ≤0x0FFF
Описание: Прочитать статус буфера buf в регистр с адресом adr. Возможные значения статуса приемо-передающего буфера приведены в таблице 2.

<b>13. SFBufStor Прочитать данные из буфера</b>
Синтаксис: SFBufStor buf, adr, offset, nreg
Операнды: 1 <buf<16; 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.

<b>14. SF_TCPRdreg Прочитать регистры с серверного устройства по протоколу Modbus TCP</b>
Синтаксис: SF_TCPRdreg dir, socket, ip, port, adr status, dev_ adr, func, offset, nreg, answer_ adr
Операнды: 0≤dir<2, 0≤socket<2; 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 с LAN порта dir IP адрес ip порт port. adr_status-регистр статуса запроса answer_ adr -указатель на адрес сохранения ответа.

<b>15. SF_TCPWrreg Записать регистры с серверного устройства по протоколу Modbus TCP</b>
Синтаксис: SF_TCPWrreg dir, socket, ip, port, adr_status, dev_ adr, func, offset, nreg, srsadr
Операнды: : 0≤dir<2, 0<socket<2; 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 с LAN порта dir IP адрес ip порт port. adr_status-регистр статуса запроса, srsadr - указатель на данные

## 7.2. Словно ориентированные инструкции

### 7.2.1 Логические команды

<b>1. andlr Логическое «И» содержимого регистра и константы</b>
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

<b>2. andrr</b> Логическое «И» содержимого регистров
Синтаксис: andrr reg1,reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: $(\text{reg1}) \text{ AND } (\text{reg2}) \rightarrow (\text{reg1})$
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистров

<b>3. andrlr</b> Логическое «И» содержимого регистра и константы с сохранением результата
Синтаксис: andrlr reg1,reg2,literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$
Операция: $(\text{reg2}) \text{ AND } \text{literal} \rightarrow (\text{reg1})$
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистра и константы с сохранением результата

<b>4. andrrr</b> Логическое «И» содержимого регистров с сохранением результата
Синтаксис: andrrr reg1,reg2,reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ; $0x0000 \leq \text{reg3} \leq 0x0FFF$ ;
Операция: $(\text{reg2}) \text{ AND } (\text{reg3}) \rightarrow (\text{reg1})$
Флаги: Z, N
Описание: Логическое побитовое «И» содержимого регистров с сохранением результата

<b>5. orlr</b> Логическое «ИЛИ» содержимого регистра и константы
Синтаксис: orlr reg,literal
Операнды: $0x0000 \leq \text{reg} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$
Операция: $(\text{reg}) \text{ OR } \text{literal} \rightarrow (\text{reg})$
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистра и константы

<b>6. orrr</b> Логическое «ИЛИ» содержимого регистров
Синтаксис: orrr reg1,reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: $(\text{reg1}) \text{ OR } (\text{reg2}) \rightarrow (\text{reg1})$
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистров

<b>7. orrlr</b> Логическое «ИЛИ» содержимого регистра и константы с сохранением результата
Синтаксис: orrlr reg1,reg2,literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$
Операция: $(\text{reg2}) \text{ OR } \text{literal} \rightarrow (\text{reg1})$
Флаги: Z, N
Описание: Логическое побитовое «ИЛИ» содержимого регистра и константы с сохранением результата

<b>8. orrrr</b> 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

<b>9. xorlr</b> 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

<b>10. xorrr</b> 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

<b>11. xorrlr</b> 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

<b>12. xorrrr</b> 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

#### В.7.2.2 Арифметические команды

<b>1. addlrc</b> Суммирование содержимого регистра и константы
Синтаксис: addlrc 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

<b>2. addrnc Суммирование содержимого регистров</b>
Синтаксис: addrnc reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) + (\text{reg3}) \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистров reg2 и reg3 с сохранением результата в регистре reg1
<b>3. addlrc Суммирование содержимого регистра и константы с учетом флага переноса C</b>
Синтаксис: addlrc reg1, reg2, literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg2}) + \text{literal} + C \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистра reg2 и константы literal с учетом флага переноса C с сохранением результата в регистре reg1
<b>4. addrcc Суммирование содержимого регистров с учетом флага переноса C</b>
Синтаксис: addrcc reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) + (\text{reg3}) + C \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Суммирование содержимого регистров reg2 и reg3 с учетом флага переноса C с сохранением результата в регистре reg1
<b>5. sublrc Вычитание константы из содержимого регистра</b>
Синтаксис: sublrc reg1, reg2, literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg2}) - \text{literal} \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Вычитание константы literal из содержимого регистра reg2 и с сохранением результата в регистре reg1
<b>6. subrrnc Вычитание содержимого регистров</b>
Синтаксис: subrrnc reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) - (\text{reg3}) \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Вычитание содержимого регистра reg3 из reg2 с сохранением результата в регистре reg1
<b>7. sublcc Вычитание константы из содержимого регистра с учетом флага заёма</b>
Синтаксис: sublcc reg1, reg2, literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg2}) - \text{literal} - (\sim C) \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Вычитание константы literal из содержимого регистра reg2 с учетом флага заема с сохранением результата в регистре reg1

<b>8. subrrc Вычитание регистров с учетом флага заёма</b>
Синтаксис: subrrc reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) - (\text{reg3}) - (\sim C) \rightarrow (\text{reg1})$
Флаги: Z, N, C, OV
Описание: Вычитание содержимого регистра reg3 из reg2 с сохранением результата в регистре reg1

<b>9. mullr Умножение содержимого регистра на константу</b>
Синтаксис: mullr reg1, reg2, literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg2}) * \text{literal} \rightarrow (\text{reg1})(\text{reg1}+1)$
Флаги: Z
Описание: Умножение содержимого регистра reg2 на константу с сохранением результата в регистры reg1 и reg1+1

<b>10. mulrr Умножение содержимого регистров</b>
Синтаксис: mulrr reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) * (\text{reg3}) \rightarrow (\text{reg1})(\text{reg1}+1)$
Флаги: Z
Описание: Умножение содержимого регистра reg2 на reg3 с сохранением результата в регистры reg1 и reg1+1

<b>11. divlr Деление содержимого регистра на константу</b>
Синтаксис: divlr reg1, reg2, literal
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg2}) / \text{literal} \rightarrow (\text{reg1})(\text{reg1}+1)$
Флаги: Z
Описание: Деление содержимого регистра reg2 на константу с сохранением результата в регистры reg1 и reg1+1

<b>12. divrr Деление содержимого регистров</b>
Синтаксис: divrr reg1, reg2, reg3
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x0000 \leq \text{reg2} \leq 0x2000$ ; $0x0000 \leq \text{reg3} \leq 0x2000$ ;
Операция: $(\text{reg2}) / (\text{reg3}) \rightarrow (\text{reg1})(\text{reg1}+1)$
Флаги: Z
Описание: Умножение содержимого регистра reg2 на reg3 с сохранением результата в регистры reg1 и reg1+1

### В.7.2.3 Команды сдвига

<b>1. rlc Циклический сдвиг влево содержимого регистра через флаг переноса</b>
Синтаксис: rlc reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Циклический сдвиг влево содержимого регистра reg1 через флаг переноса n раз



<b>2. rrc Циклический сдвиг вправо содержимого регистра через флаг переноса</b>
Синтаксис: rrc reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Циклический сдвиг вправо содержимого регистра reg1 через флаг переноса n раз

<b>3. rlc Циклический сдвиг влево содержимого регистра</b>
Синтаксис: rlc reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Циклический сдвиг влево содержимого регистра reg1 n раз

<b>4. rrc Циклический сдвиг вправо содержимого регистра</b>
Синтаксис: rrc reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Циклический сдвиг вправо содержимого регистра reg1 n раз

<b>5. shl Сдвиг влево содержимого регистра</b>
Синтаксис: shl reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Сдвиг влево содержимого регистра reg1 n раз

<b>6. shr Сдвиг вправо содержимого регистра</b>
Синтаксис: shr reg1, n
Операнды: $0x0000 \leq \text{reg1} \leq 0x2000$ ; $0x00 \leq n \leq 0x0F$ ;
Флаги: Z, N
Описание: Сдвиг вправо содержимого регистра reg1 n раз

### В.7.2.3 Специальные команды

<b>1. movl Запись константы в регистр</b>
Синтаксис: movl reg, literal
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: $(\text{reg}) = \text{literal}$
Описание: Запись константы в регистр

<b>2. movr Копирование регистра</b>
Синтаксис: movr reg1, reg2
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Операция: $(\text{reg1}) = (\text{reg2})$
Описание: Запись содержимого регистра reg2 в регистр reg1

<b>3. incr Инкрементирование содержимого регистра</b>
Синтаксис: incr reg
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Операция: $(\text{reg}) = (\text{reg}) + 1$
Флаги: Z, C, N, OV
Описание: Инкрементирование содержимого регистра reg

<b>4. decr Декрементирование содержимого регистра</b>
---

Синтаксис: <code>decr reg</code>
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Операция: $(\text{reg}) = (\text{reg}) - 1$
Флаги: Z, C, N, OV
Описание: Декрементирование содержимого регистра <code>reg</code>

<b>5. not Инверсия содержимого регистра</b>
Синтаксис: <code>not reg</code>
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Операция: $(\text{reg}) = \sim(\text{reg})$
Флаги: Z, N
Описание: Инвертирование регистра <code>reg</code>

<b>6. neg Отрицание +1</b>
Синтаксис: <code>neg reg</code>
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Операция: $(\text{reg}) = \sim(\text{reg}) + 1$
Флаги: Z, N
Описание: Перевод содержимого регистра в дополнительный код

<b>7. swapr Поменять местами байты регистра</b>
Синтаксис: <code>swapr reg</code>
Операнды: $0x0000 \leq \text{reg} \leq 0x2000$ ;
Описание: Поменять местами старший и младший байты регистра

### В.7.3. Команды работы с подпрограммами

<b>1. call Вызов подпрограммы</b>
Синтаксис: <code>call k</code>
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $PC + 1 \rightarrow [SP]$ ; $k \rightarrow PC$ ; $SP + 1 \rightarrow SP$
Описание: Безусловный вызов подпрограммы

<b>2. return Возврат из подпрограммы</b>
Операция: $SP - 1 \rightarrow SP$ ; $[SP] \rightarrow PC$
Описание: Безусловный возврат из подпрограммы

### В.7.4. Команды переходов

<b>1. jmp Безусловный переход на адрес</b>
Синтаксис: <code>jmp k</code>
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Безусловный переход на адрес

<b>2. jz Переход на адрес, если установлен флаг Z</b>
Синтаксис: <code>jz k</code>
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес <code>k</code> если установлен флаг <code>Z</code>

<b>3. jnz Переход на адрес, если флаг Z не установлен</b>
Синтаксис: <code>jnz k</code>

Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если не установлен флаг Z
<b>4. jc Переход на адрес, если установлен флаг C</b>
Синтаксис: jc k
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если установлен флаг C
<b>5. jnc Переход на адрес, если флаг C не установлен</b>
Синтаксис: jnc k
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если не установлен флаг C
<b>6. jn Переход на адрес, если установлен флаг N</b>
Синтаксис: jn k
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если установлен флаг N
<b>7. jnp Переход на адрес, если флаг N не установлен</b>
Синтаксис: jnp k
Операнды: $0x0000 \leq k \leq 0xFFFF$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если не установлен флаг N
<b>8. jbs Переход на адрес, если установлен бит регистра</b>
Синтаксис: jbs k,reg,n
Операнды: $0x0000 \leq k \leq 0xFFFF$ ; $0x0000 \leq reg \leq 0x0FFF$ ; $0x00 \leq n \leq 0x0F$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если установлен бит N регистра reg
<b>9. jbc Переход на адрес, если не установлен бит регистра</b>
Синтаксис: jbc k,reg,n
Операнды: $0x0000 \leq k \leq 0xFFFF$ ; $0x0000 \leq reg \leq 0x0FFF$ ; $0x00 \leq n \leq 0x0F$ ;
Операция: $k \rightarrow PC$
Описание: Выполнить переход на адрес k если не установлен бит N регистра reg
<b>10. cmpleq Пропустить, если значение регистра равно значению константы</b>
Синтаксис: cmpleq reg, literal
Операнды: $0x0000 \leq reg \leq 0x0FFF$ ; $0x0000 \leq literal \leq 0xFFFF$ ;
Операция: если $(reg) = literal$ , то $PC+2 \rightarrow PC$
Описание: Пропустить следующую команду, если содержимое регистра равно значению константы

<b>11. cmplneq Пропустить, если значение регистра не равно значению константы</b>
Синтаксис: cmplneq reg, literal
Операнды: $0x0000 \leq \text{reg} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: если $(\text{reg}) \neq \text{literal}$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если содержимое регистра не равно значению константы

<b>12. cmplgt Пропустить, если значение регистра больше чем значение константы</b>
Синтаксис: cmplgt reg, literal
Операнды: $0x0000 \leq \text{reg} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: если $(\text{reg}) > \text{literal}$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если содержимое регистра больше значения константы

<b>13. cmplt Пропустить, если значение регистра меньше чем значение константы</b>
Синтаксис: cmplt reg, literal
Операнды: $0x0000 \leq \text{reg} \leq 0x0FFF$ ; $0x0000 \leq \text{literal} \leq 0xFFFF$ ;
Операция: если $(\text{reg}) < \text{literal}$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если содержимое регистра меньше значения константы

<b>14. cmpr eq Пропустить, если значения регистров равны</b>
Синтаксис: cmpr eq reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) = (\text{reg2})$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если значения регистров равны

<b>15. cmpr neq Пропустить, если значения регистров не равны</b>
Синтаксис: cmpr neq reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) \neq (\text{reg2})$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если значения регистров не равны

<b>16. cmpr gt Пропустить, если больше</b>
Синтаксис: cmpr gt reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) > (\text{reg2})$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если значение регистра reg1 больше значения регистра reg2

<b>17. cmpr lt Пропустить, если меньше</b>
Синтаксис: cmpr lt reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) < (\text{reg2})$ , то $\text{PC}+2 \rightarrow \text{PC}$
Описание: Пропустить следующую команду, если значение регистра reg1 меньше значения регистра reg2

<b>18. jeqrr Переход, если значения регистров равны</b>
Синтаксис: jeqrr reg1, reg2, label
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) = (\text{reg2})$ , то $\text{label} \rightarrow \text{PC}$
Описание: Переход на метку label, если значение регистра reg1 равно значению регистра reg2

<b>19. jneqrr Переход, если значения регистров не равны</b>
Синтаксис: jneqrr reg1, reg2, label
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) \neq (\text{reg2})$ , то $\text{label} \rightarrow \text{PC}$
Описание: Переход на метку label, если значение регистра reg1 не равно значению регистра reg2

<b>20. jeqrl Переход, если значения регистра равно константе</b>
Синтаксис: jeqrl reg1, literal, label
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) = \text{literal}$ , то $\text{label} \rightarrow \text{PC}$
Описание: Переход на метку label, если значение регистра reg1 равно константе literal

<b>21. jneqrl Переход, если значения регистра не равно константе</b>
Синтаксис: jneqrl reg1, literal, label
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: если $(\text{reg1}) \neq \text{literal}$ , то $\text{label} \rightarrow \text{PC}$
Описание: Переход на метку label, если значение регистра reg1 не равно константе literal

<b>22. jbstat Переход в зависимости от состояния буфера порта</b>
Синтаксис: jbstat buf, label1, label2, label3
Операнды: $0 \leq \text{buf} \leq 16$ ;
Описание: Переход на одну из меток labelx в зависимости от состояния буфера порта

#### В.7.5. Команды косвенной адресации

<b>1. movMR Скопировать содержимое регистра 2 в регистр с адресом лежащим в регистре 1</b>
Синтаксис: movMR reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: $[\text{reg1}] = \text{reg2}$
Описание: Скопировать содержимое регистра reg1 в регистр с адресом лежащим в регистре reg2

<b>2. movRM Скопировать содержимое регистра с адресом лежащим в регистре 1 в регистр 2</b>
Синтаксис: movRM reg1, reg2
Операнды: $0x0000 \leq \text{reg1} \leq 0x0FFF$ ; $0x0000 \leq \text{reg2} \leq 0x0FFF$ ;
Операция: $\text{reg1} = [\text{reg2}]$
Описание: Скопировать содержимое регистра с адресом лежащим в регистре 1 в регистр 2

## В.7.6 Команды работы с портами ввода - вывода

<b>1. out Установка выхода</b>
Синтаксис: out num, state
Операнды: $0 \leq \text{num} < 7$ , $0 \leq \text{state} < 3$
Операция: digital output (num) = state
Описание: установка выхода num в состояние state (0-clear, 1-set, 2- toggle)

<b>2. outl Запись константы в порт выхода</b>
Синтаксис: outl literal
Операнды: $0x0000 \leq \text{literal} \leq 0x00FF$ ;
Операция: digital outputs = literal
Описание: Установка выходов согласно битам младшего байта константы

<b>3. outr Запись значения регистра в порт выхода</b>
Синтаксис: outr address
Операнды: $0x0000 \leq \text{address} \leq 0x0FFF$ ;
Операция: digital outputs = [address]
Описание: Установка выходов согласно битам младшего байта регистра address

<b>4. movrdi Чтение состояния дискретных входов в регистр</b>
Синтаксис: movrdi address
Операнды: $0x0000 \leq \text{address} \leq 0x0FFF$ ;
Операция: [address] = digital inputs
Описание: Чтение состояния дискретных входов в регистр address

<b>5. incopy Копирование состояния входа в бит регистра</b>
Синтаксис: incopy address, regbit, din
Операнды: $0x0000 \leq \text{address} \leq 0x0FFF$ ; $0 \leq \text{regbit} \leq 15$ , $0 \leq \text{din} \leq 7$
Операция: [address], regbit = din
Описание: Копирование состояния входа din в бит regbit регистра address

<b>6. jinset Переход если вход в состоянии 1</b>
Синтаксис: jinset din, label
Операнды: $0 \leq \text{din} \leq 7$ , $0x0000 \leq \text{label} \leq 0x0FFF$ ;
Операция: если din = 1, то label → PC
Описание: Переход если вход в состоянии 1

<b>7. jinclr Переход если вход в состоянии 0</b>
Синтаксис: jinclr din, label
Операнды: $0 \leq \text{din} \leq 7$ , $0x0000 \leq \text{label} \leq 0x0FFF$ ;
Операция: если din = 0, то label → PC
Описание: Переход если вход в состоянии 0

<b>8. movrsw Прочитать состояние переключателя в регистр</b>
Синтаксис: movrsw address
Операнды: $0x0000 \leq \text{address} \leq 0x0FFF$ ;
Операция: [address] = switchs state
Описание: Прочитать состояние переключателя в регистр address

<b>9. swcopy Копирование состояние переключателя в бит регистра</b>
Синтаксис: swcopy adres, regbit, switch
Операнды: $0x0000 \leq \text{adres} \leq 0x0FFF$ ; $0 \leq \text{regbit} \leq 15$ , $0 \leq \text{switch} \leq 7$
Операция: [adres], regbit = switch
Описание: Копирование состояния переключателя switch в бит regbit регистра adres
<b>10. jswset Переход если бит переключателя ON</b>
Синтаксис: jswset switch, label
Операнды: $0 \leq \text{switch} \leq 7$ , $0x0000 \leq \text{label} \leq 0x0FFF$ ;
Операция: если switch = 1, to label → PC
Описание: Переход если переключатель в состоянии ON
<b>11. jswclr Переход если бит переключателя OFF</b>
Синтаксис: jswclr switch, label
Операнды: $0 \leq \text{switch} \leq 7$ , $0x0000 \leq \text{label} \leq 0x0FFF$ ;
Операция: если switch = 0, to label → PC
Описание: Переход если переключатель в состоянии OFF
<b>12. outa управление питанием входа контроля шлейфа (0-clear, 1-set)</b>
Синтаксис: outa num, state
Операнды: $0 \leq \text{num} < 4$ , $0 \leq \text{state} < 2$
Операция: aoutput = state
Описание: управление питанием входа контроля шлейфа (0-clear, 1-set)
<b>13. outla Запись константы в порт контроля шлейфа</b>
Синтаксис: outla literal
Операнды: $0x0000 \leq \text{literal} \leq 0x00FF$ ;
Операция: aoutputs = literal
Описание: Запись константы в порт контроля шлейфа
<b>14. outra Запись значения регистра в порт контроля шлейфа</b>
Синтаксис: outra adres
Операнды: $0x0000 \leq \text{adres} \leq 0x0FFF$ ;
Операция: aoutputs = [adres]
Описание: Запись значения регистра в порт контроля шлейфа
<b>15. movrai Прочитать значение тока шлейфа в регистр</b>
Синтаксис: movrai adres, num
Операнды: $0x0000 \leq \text{adres} \leq 0x0FFF$ ; $0 \leq \text{num} < 4$
Операция:
Описание: Прочитать значение тока шлейфа в регистр adres