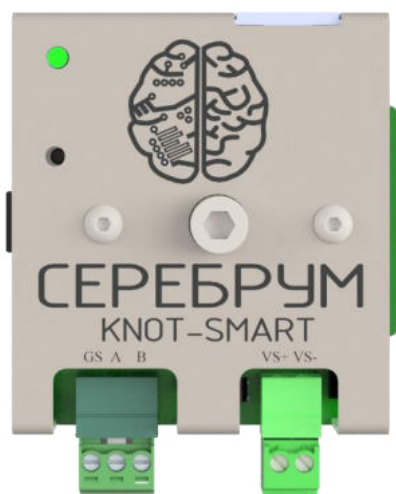


Программируемый модуль удаленного ввода-вывода СЕРЕБРУМ SMART KNOT

Техническое руководство



| | | | | |
|--|-----------------------------|----------------------------------|--------------------------------------|-------------------------------|
| Ethernet  | MODBUS TCP RTU | UDP YART LINK | RS485 | ШИНА модулей I/O |
| среда YART Studio | языки FBD C | отладка VIRTUAL PLC | отладка VISUAL элементы | AUTO PID |

1. Оглавление

| | |
|--|----|
| 2. Предупреждения | 3 |
| 3. Список изменений..... | 4 |
| 4. Общая информация..... | 5 |
| 5. Технические характеристики | 6 |
| 6. Подключение | 8 |
| 7. Начало работы | 10 |
| 7.1 Настройки сети | 11 |
| 8. Использование модулей расширения..... | 12 |
| 9. Индикация состояний | 15 |
| 10. ПЛК, особенности архитектуры и память | 16 |
| 10.1 Память программируемого контроллера..... | 17 |
| 10.1.1 Системная память | 17 |
| 10.1.2 Оперативная память..... | 17 |
| 10.1.3 Энергонезависимая память (FRAM) | 18 |
| 10.1.4 Батарейная память..... | 19 |
| 10.2 Типы данных, используемых в системе | 20 |
| 11. Модули обмена данными | 21 |
| 11.1 Modbus TCP..... | 21 |
| 11.2 Modbus RTU | 24 |
| 11.4 YART Link..... | 26 |
| 12. Работа с подключаемыми периферийными модулями..... | 29 |
| 12.1 Аналоговые входы..... | 30 |
| 12.2 Аналоговые выходы | 34 |
| 12.3 Дискретные входы..... | 36 |
| 12.4 Дискретные выходы | 37 |
| 13. Транспортировка и хранение | 38 |
| 14. Гарантийные обязательства | 39 |

2. Предупреждения

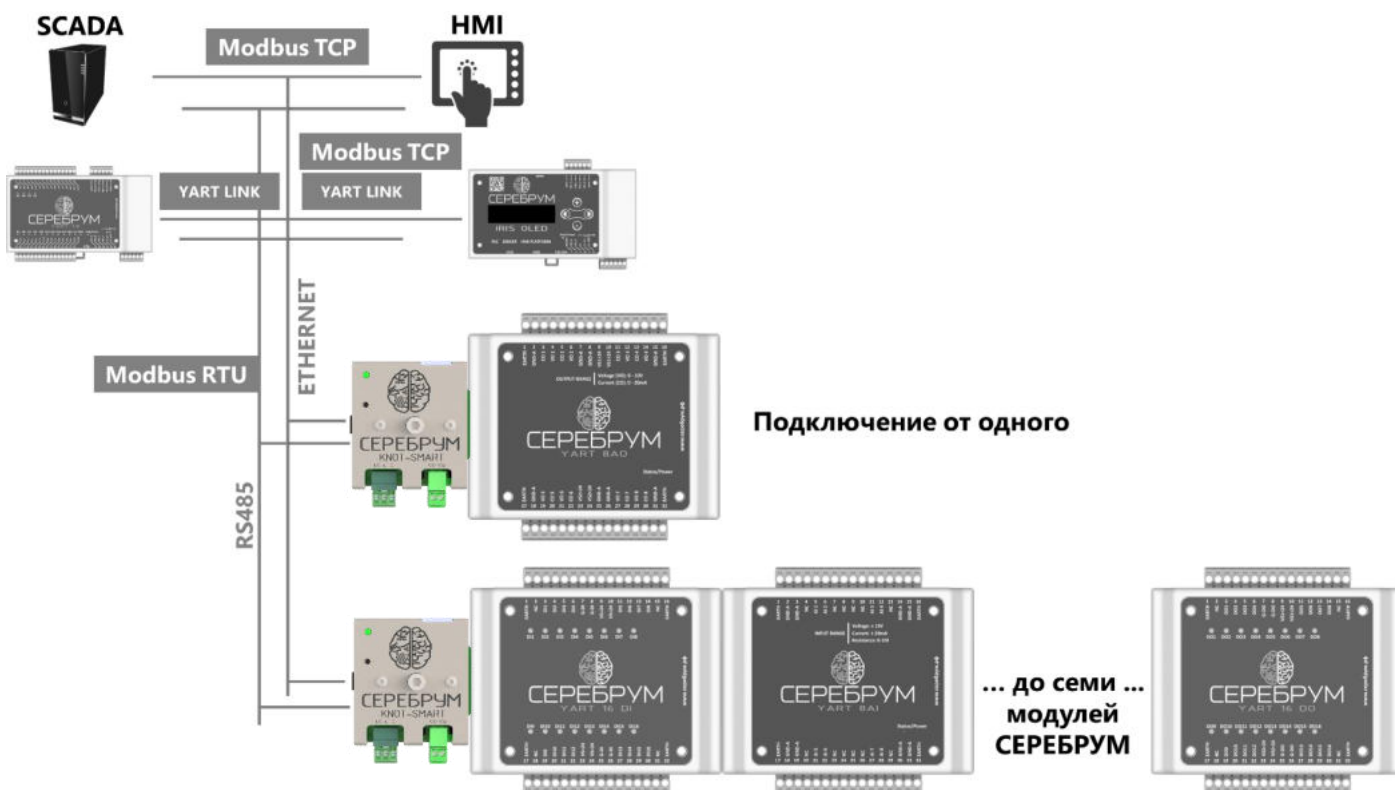
3. Список изменений

4. Общая информация

Контроллер SMART KNOT — головной модуль каждой станции удаленного ввода-вывода с возможностью программирования в среде YART Studio. Набор функциональных блоков SMART KNOT позволяет решать множество задач управления и обработки входных сигналов.

SMART KNOT может использоваться в качестве промышленного контроллера в локальных и распределенных системах управления с каналами передачи данных через Ethernet и/или RS-485.

Программирование и настройка контроллера выполняется в среде YART Studio, соединение с которой осуществляется через интерфейс Ethernet.



За счет возможности программирования SMART KNOT позволяет осуществить обработку полученных данных с линий модулей расширения, записать полученные значения в единый массив и осуществить передачу максимально эффективным способом через Modbus RTU Slave, Modbus TCP Slave или YART Link.

5. Технические характеристики

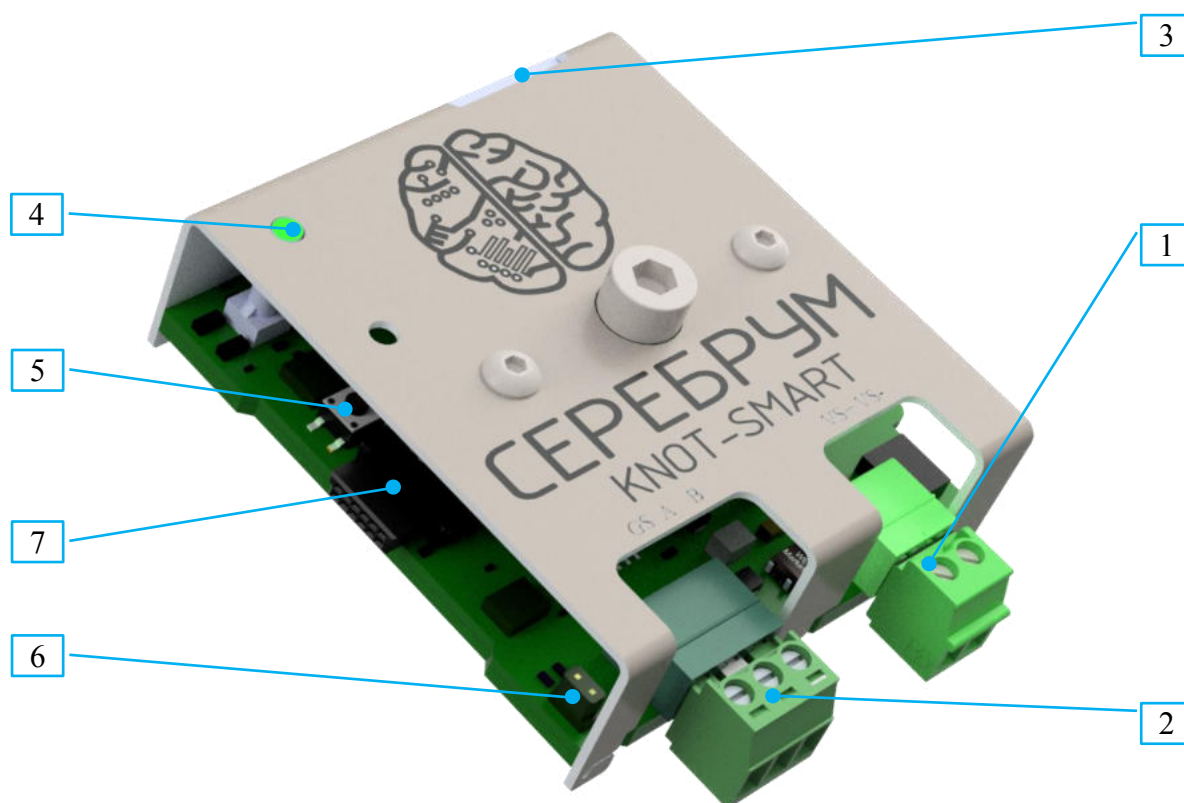


Рисунок 1. Контроллер SMART KNOT

1. Клеммник питания модуля
2. Клеммник последовательного порта
3. Разъем RJ-45, Ethernet
4. Индикатор статуса
5. Кнопка сброса сетевых настроек
6. «Терминатор» последовательного порта
7. Разъем подключения модулей расширения (YART BUS)

Таблица 1. Технические характеристики SMART KNOT

| Параметр | | Характеристики |
|----------------------------|-----------|--|
| Напряжение питания | | от 10 до 28 VDC |
| Тип разъемов клеммника | | Разъемные, под винт, максимальное сечение провода 1.5 мм ² |
| Программирование | | ФБД, С-YART, среда программирования YART Studio |
| Память | Программ | 74 КВ энергонезависимая память (более 50 типов блоков, 500 вызовов блоков) |
| | RAM | 3 КВ память для переменных пользователя |
| | FRAM | 16 КВ энергонезависимая память для переменных пользователя |
| | BRAM | Отсутствует |
| | RAM BIT | 2 КБИТ память для переменных пользователя |
| | FRAM BIT | 2 КБИТ энергонезависимая память для переменных пользователя |
| uSD карта памяти | | Отсутствует |
| Выполнения цикла программы | | от 1 ms |
| Коммуникационные порты | ETHERNET | Разъем RJ-45, 100/10Base-T; YART-LINK, MODBUS TCP |
| | RS485 | MODBUS RTU Slave |
| | YART-PORT | Шина модулей расширения, период опроса всех модулей менее 30 ms, максимальное число подключаемых модулей – 7 |

6. Подключение

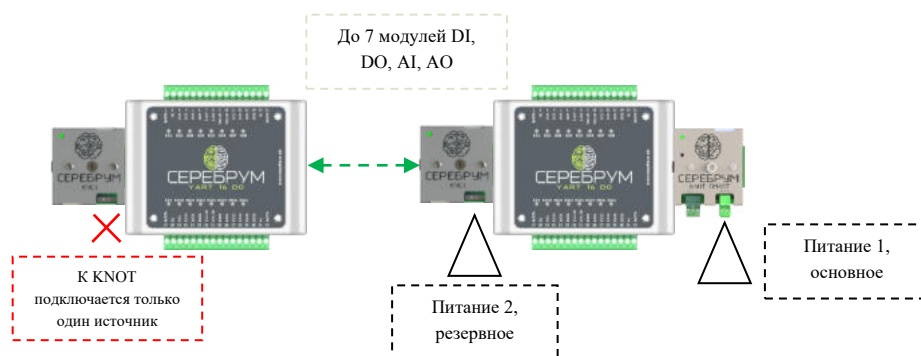


Рисунок 2. Подключение источников питания

Для работы контроллера требуется наличие внешнего питания (см. **Таблицу №1**). Линия питания выведена на клеммы 1-2 клеммника 1, а также на выводы разъема подключения модулей расширения – YART-BUS.

На **Рисунке 2** показана схема питания контроллера.

Основной источник питания подключается к клеммнику 1. Модули расширения, подключенные к SMART KNOT будут работать от основного источника питания через шину расширения YART-BUS.

При необходимости организации резервного питания или при отсутствии возможности подключения к клеммнику №1 питание может осуществляться через модуль KNOT, подключаемый к шине расширения ПЛК.

Место подключения не регламентируется.

! **Внимание**

При одновременном использовании нескольких модулей KNOT внешнее питание должно подключаться только к одному из них.

Таблица 2. Сигналы клеммника 1

| Номер клеммы | Наименование | Назначение |
|--------------|--------------|------------------------------------|
| 1 | VS+ | Линия питания контроллера, «плюс» |
| 2 | VS- | Линия питания контроллера, «общий» |

Таблица 3. Сигналы клеммника 2

| Номер клеммы | Наименование | Назначение |
|--------------|--------------|--|
| 1 | RS485-A | Порт RS-485, Линия А |
| 2 | RS485-B | Порт RS-485, Линия В |
| 3 | RS485-G | Порт RS-485, Общий. Порт гальванически изолирован от цепей питания |

7. Начало работы

Обмен данными с SMART KNOT осуществляется через интерфейс Ethernet. Для работы с контроллером уделите внимание корректным настройкам сетевого оборудования и сетевой инфраструктуры в целом.

Сброс сетевых настроек осуществляется при помощи кнопки сброса (поз. 5 Рис. 1). Для сброса необходимо удерживать кнопку сброса в процессе включения. При следующем включении сетевые настройки сбросятся в значения по умолчанию.

7.1 Настройки сети

Настройка сетевого интерфейса осуществляется помощи YART Studio.

По умолчанию SMART KNOT получает статический IP адрес равный **192.168.1.254**

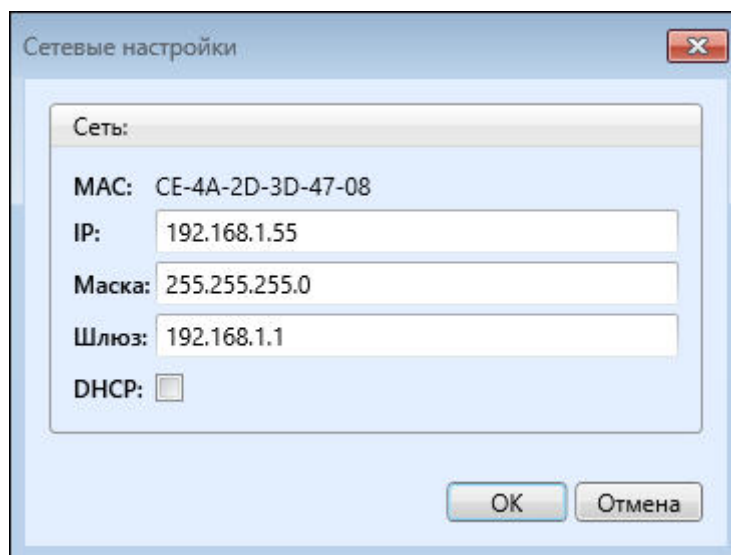


Рисунок 3. Настройка сетевого подключения

Для осуществления настройки сетевого подключения перейдите в панель «Свойства ПЛК» - «Сетевые настройки».

В отобразившемся окне (**Рис. 3**) расположены настройки основного сетевого подключения.

MAC – отображает уникальный аппаратный адрес контроллера.

IP – текущий IP адрес контроллера.

Маска – маска подсети

Шлюз – шлюз для подключения к другим сетям

DHCP – поле выбора режима автоматического назначения сетевого адреса. В случае, если в сети не будет найден сервер DHCP адрес контроллера будет равен значению из поля «**IP**»

Новые настройки сетевого подключения станут активными только после полной перезагрузки контроллера.

Перезагрузка осуществляется из основного меню YART Studio (пункт «Перезапуск»  **Перезапуск**), либо сбросом питания контроллера.

8. Использование модулей расширения

SMART KNOT не оснащен встроенными линиями ввода-вывода. Подключение внешних устройств осуществляется только при помощи внешних модулей расширения, присоединяемых к разъему YART-BUS.

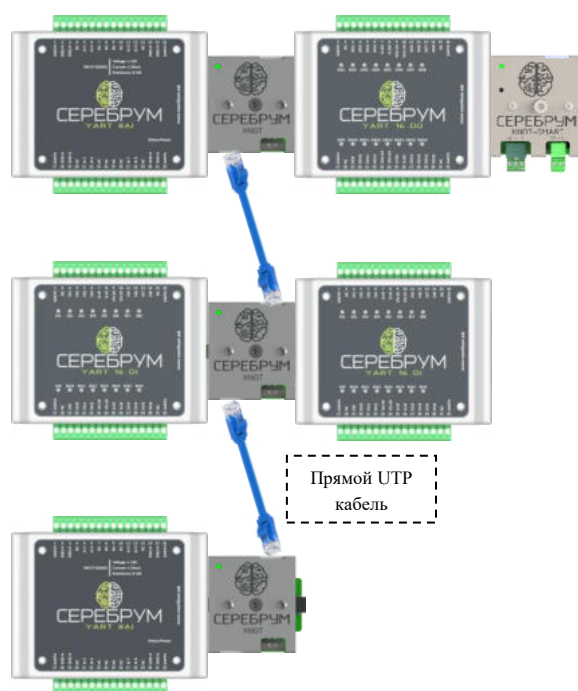


Рисунок 4. Возможная топология шины расширения

Пример возможной конфигурации шины YART-BUS представлен на **Рисунке 4**

При проектировании конфигурации шины необходимо учитывать следующие правила:

- Контроллер SMART KNOT поддерживает до семи различных модулей расширения на одной шине
- Суммарная длина всех соединений не должна превышать 5 метров
- На оконечном модуле должен быть включен согласующий «терминатор»
- Учитывайте падение напряжения между модулями. В среднем на каждом модуле теряется 0.1В
- При низком значении напряжения питания удаленный модуль может отключиться
- Диапазон разрешенных адресов для конфигурации модулей расширения 1 – 7
- Порядок присоединения модулей может быть любым

Программная конфигурация модулей осуществляется при программе YARTStudio.

При каждом запуске контроллер осуществляет сканирование и идентификацию всех возможных модулей. В процессе работы найденные модули опрашиваются, а результат работы сохраняется в специализированных регистрах процессора.

На рисунке ниже показан пример автоматически определенных модулей.



Рисунок 5. Список определенных модулей

SMART KNOT может работать в двух режимах опроса модулей расширения:

- В **безопасном режиме работы** любая ошибка опроса модулей расширения вызовет аварийную остановку прикладной программы. При этом все выходы модулей будут переведены в безопасное состояние (нуль).
- В **обычном режиме работы** ошибка обмена с модулем расширения не вызывает остановку выполнения прикладной программы, но состояние ошибки фиксируется в соответствующем регистре процессора для анализа и диагностики.

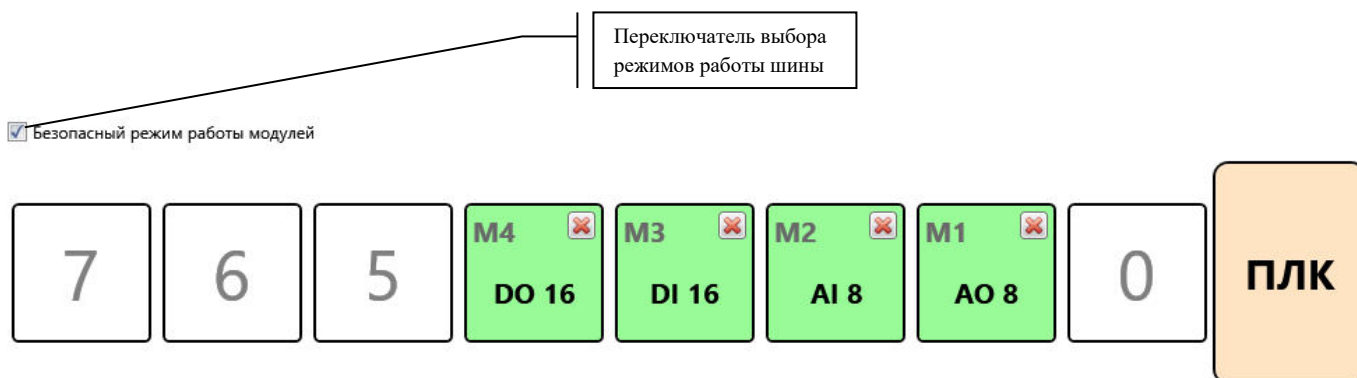


Рисунок 6. Конфигурация модулей расширения

Выбор режима работы шины расширения осуществляется в окне настроек проекта – модули расширения.

Каждый модуль расширения должен иметь уникальный адрес, который назначается вращающимся переключателем, расположенным на плате модуля.

Для SMART KNOT доступны адреса модулей в диапазоне 1 – 7.

Пример модуля расширения и расположение переключателей показан на **Рисунке 7**.

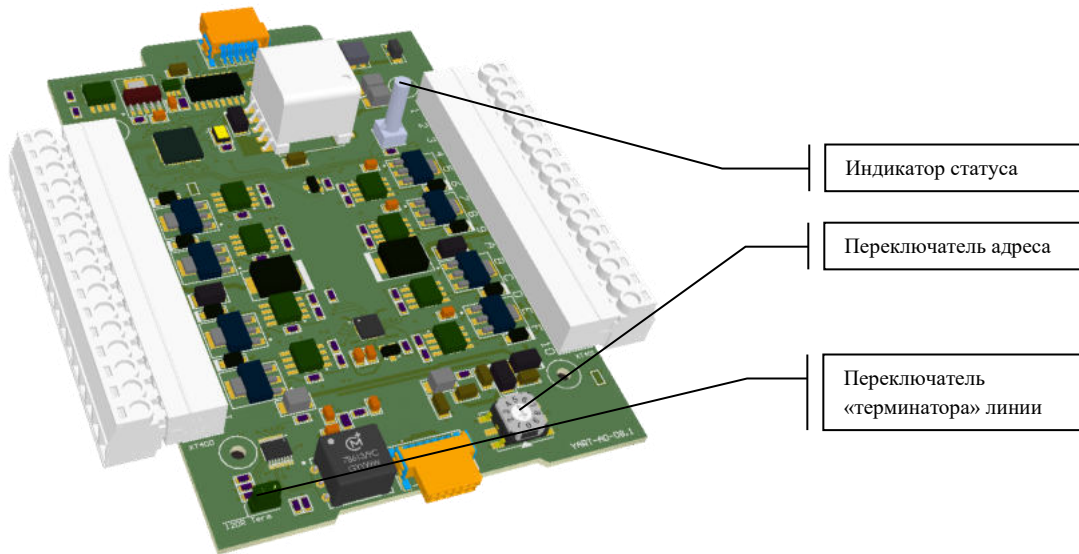


Рисунок 7. Переключатели модулей расширения

Перед началом работы убедитесь, что переключатели адресов модулей расширения установлены в правильные позиции, совпадений между ними нет.

В процессе работы в окне статуса модулей расширения YART Studio будет отображаться текущее состояние шины расширения.

Для успешной работы отображаемое состояние модулей должно в точности соответствовать проектному.

Каждый модуль расширения оснащается статусным индикатором. Для исправного модуля расширения данный индикатор должен всегда быть включен.

9. Индикация состояний

Отображение состояния ПЛК SMART KNOT осуществляется при помощи светодиодного индикатора (поз. 4 Рис. 1)

Различные режимы отображения представлены в **Таблице 4**.

Таблица 4. Состояния встроенного индикатора

| № | Режим индикатора | Состояние ПЛК |
|---|-------------------------------|--|
| 1 | Горит зеленым | Выполнение прикладной программы (режим Run) |
| 2 | Мигает зеленым (t = 1 сек) | Выполнение прикладной программы остановлено |
| 3 | Мигает зеленым (t = 0.25 сек) | Аварийная остановка прикладной программы |
| 4 | Красный светодиод | Мигание красного светодиода отображает прием и передачу данных через Ethernet и/или RS-485. При передаче данных светодиод моргает. |

10. ПЛК, особенности архитектуры и память

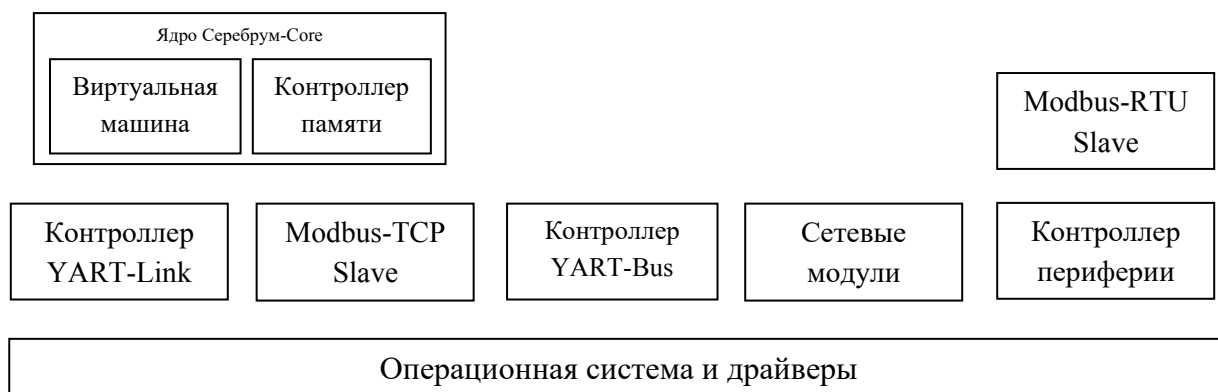


Рисунок 8. Архитектура контроллера SMART KNOT

В основе вычислительного ядра контроллера лежит система *Серебрум-Core*, состоящая из виртуальной машины и контроллера виртуальной памяти.

Все взаимодействие между модулями осуществляется через общее пространство памяти, в которой определены участки периферии, данных пользователя и настроек. Более подробно структура памяти рассматривается ниже.

Независимо от виртуальной машины ПЛК выполняет задачи обработки внешних интерфейсов, приёма и передачи данных, а также поддерживает функционирование встроенного сервера телеметрии.

При изменении настроек загруженного проекта все сопутствующие программы автоматически перенастраиваются.

Часть программных модулей настраиваются и обрабатываются непосредственно из программы пользователя. Например, работа программ обмена по Modbus конфигурируется через специальный API, что ускоряет и упрощает процесс настройки.

SMART KNOT содержит встроенные периферийный модули для непосредственного подключения к линиям датчиков и управления.

10.1 Память программируемого контроллера

Вся память контроллера (**Таблица 5**) условно разделена на две основных части:

- Память программы
- Память данных

В памяти программы хранится исполняемый код виртуальной машины, настройки проекта, список переменных, а также настройки обмена и телеметрии.

При каждом старте контроллера содержимое памяти программы проверяется и, если все поля верны - даётся разрешение на исполнение кода виртуальной машины.

Память данных содержит четыре основных сегмента

10.1.1 Системная память

В системной памяти хранятся все настройки периферии контроллера, а также текущее значение данных, полученных от модулей ввода.

Чтение и запись в системную память осуществляют специальные блоки программы пользователя, обеспечивающие корректную работу с аппаратными ресурсами ПЛК

10.1.2 Оперативная память

Основной блок памяти, предназначенный для хранения данных прикладной программы, настроек программ окружения и данных последовательных портов.

Пространство оперативной памяти автоматически сбрасывается при рестарте контроллера. Исключение составляют лишь переменные, для которых определены начальные значения.

Нижнее пространство оперативной памяти выделено для хранения значений булевых переменных.

Помимо обычного режима доступа (байт, слово, двойное слово) булевые переменные имеют битовый режим доступа.

10.1.3 Энергонезависимая память (FRAM)

FRAM даёт возможность сохранения данных программы виртуальной машины при отсутствии основного питания контроллера.

Доступ к FRAM осуществляется аналогично оперативной памяти, однако скорость работы контроллера при этом будет снижена.

При разработке прикладных программ следует избегать постоянной перезаписи значений FRAM, поскольку контроллеру требуется дополнительное время на индексацию данных энергонезависимой памяти.

Верхние 256 байт FRAM выделены под хранение энергонезависимых булевых переменных аналогично булевым переменным оперативной памяти.

10.1.4 Батарейная память

В SMART KNOT нет встроенного источника автономного питания, поэтому батарейная память недоступна для использования в прикладной программе.

Таблица 5. Карта памяти контроллера

| | | | | |
|-------|------------|--|--------|--|
| 0000h | System, 1k | Настройки Данные в/в Данные системы Управление телеметрией | 0000h | Память программ |
| 03FFh | | | | |
| 1000h | RAM, 3k | Встроенные данные Данные пользователя | | Настройки проекта |
| 1A00h | | | | Список переменных Значения констант |
| 1BFFh | | 256 байт Bool | | Данные телеметрии Настройки |
| | | | | |
| | | | | Настройки ВМ Бинарный код ВМ |
| | | | | |
| | | | | |
| B000h | FRAM, 15k | 256 байт Bool | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| E8FFh | | | | Контрольная сумма |
| | | Батарейная память | | |
| | | | | |
| | | | | |
| | | | | |
| | | | 127FFh | |

10.2 Типы данных, используемых в системе

- **Bool** – Битовая переменная.
Данные переменные хранятся в Bool сегментах (Таблица 8).
В C-YART доступны predefined слова «*true*», «*false*» соответствующие значениям «1» и «0» соответственно.
- **Byte** – Один байт данных. Беззнаковый тип в диапазоне чисел 0 – 255.
- **Short** – Знаковый 16 – разрядный тип данных. Может принимать значения в диапазоне [–32768, +32767]
- **Int** - Знаковый 32 – разрядный тип данных. Может принимать значения в диапазоне [–2 147 483 648, +2 147 483 647]
- **Float** – вещественные 32-разрядные числа в формате с плавающей точкой (IEEE 754-1985)
- **Символ** – ASCII символ в кодировке IBM CP866. Поддерживаются строки длиной до 254 символов
- **Дата** – внутренний тип данных в BCD формате даты: DD.MM.YY
Нумерация лет в рамках диапазона [0, 99]
- **Время** – внутренний формат времени в BCD формате HH:MM:SS

11. Модули обмена данными

Как видно из **Рис. 8** все интерфейсные блоки реализованы отдельно от виртуальной машины что дает возможность независимой (асинхронной) работы коммуникационных модулей. Конфигурация режимов работы и получение статуса связи выполняется посредством внутреннего API, а данные для передачи передаются в/из пространство памяти контроллера независимо от работы основной программы.

В зависимости от модификации контроллера состав интерфейсных модулей может изменяться, однако основные принципы конфигурации и обмена остаются неизменными.

11.1 Modbus TCP

Modbus TCP является основным протоколом обмена для связи ПЛК с системой YARTStudio. Кроме того, данный протокол поддерживается большинством производителей промышленного оборудования и используется повсеместно.

В контроллере SMART KNOT доступен только режим Modbus-TCP-Slave.

Modbus TCP Slave активен всегда и доступен через стандартный порт 502 (TCP). Количество одновременно работающих внешних подключений к контроллеру - 5.

Адресация Modbus привязана к физическим адресам контроллера, однако ряд системных адресов защищен от записи для обеспечения безопасной работы контроллера.

Наиболее удобный способ получения информации об адресации – это использование YARTStudio.

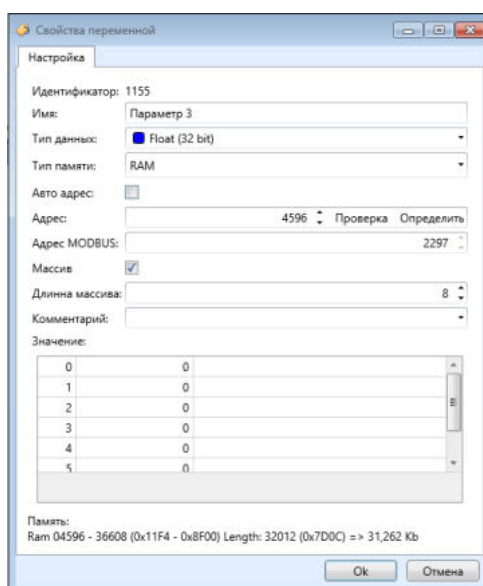


Рисунок 9. Редактирование переменных

Для точного определения любой переменной в Modbus пространстве необходимо использовать окно редактора переменных (**Рис. 9**).

YARTStudio допускает два режима физического размещения переменных: автоматический и ручной (переключатель «Авто адрес»).

В автоматическом режиме размещения компилятор виртуальной машины последовательно размещает переменные в памяти данных. В этом режиме нет возможности однозначной идентификации физического адреса Modbus.

В ручном режиме пользователь сам выбирает тот адрес, где будет размещена переменная. При этом система предлагает пользователю возможность автоматизированного назначения адресов и проверки введенного адреса на пересечение с уже объявленными адресами переменных.

В поле «Адрес» указывается текущий начальный адрес размещения переменной. Кнопки «Проверить» и «Определить» позволяют осуществить проверку использованного адреса на предмет пересечений, и выбрать новый свободный в случае необходимости.

Ниже, в поле «Modbus» указан начальный адрес, по которому данная переменная будет доступна через Modbus протокол. В примере (**Рис. 9**) этот адрес равен 2297 (Holding Register).

Необходимо отметить, что стандартная спецификация Modbus не предполагает чтение переменных типа Float, более того разные платформы могут иметь различающийся формат записи чисел с плавающей точкой. Чтение Float можно реализовать как чтение двух последовательных Holding регистров и дальнейшее их объединение.

В случае линейки контроллеров Серебрум данные преобразования реализованы автоматически – достаточно прочитать/записать требуемое количество слов.

Все данные в памяти контроллера размещаются последовательно, без промежутков, поэтому в приведенном примере массив «Параметр 3» доступен по Modbus адресам 2297 – 2312.

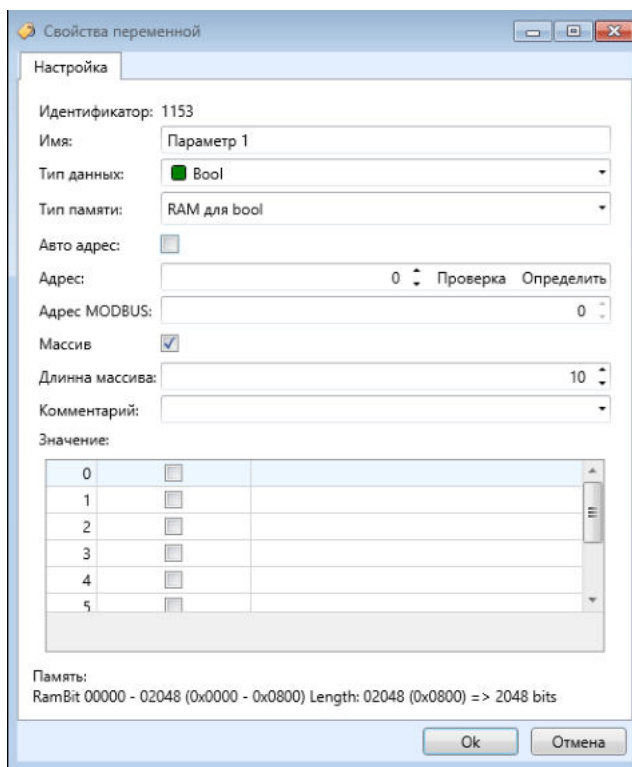


Рисунок 10. Пример с Bool

На **Рис. 10** определен массив переменных типа bool. Обратите внимание, что адрес указан как 0, поскольку доступ к bool памяти организован при помощи команд 01, 05, 15 – Coils Read/Write.

Изменение режима адресации переменной никак не сказывается на использование данной переменной в других программных модулях. Эта переменная может быть использована в модуле Modbus-RTU, телеметрии, архиве и т. д.

YARTStudio позволяет осуществить экспорт списка переменных для последующей интеграции в продукты других компаний.

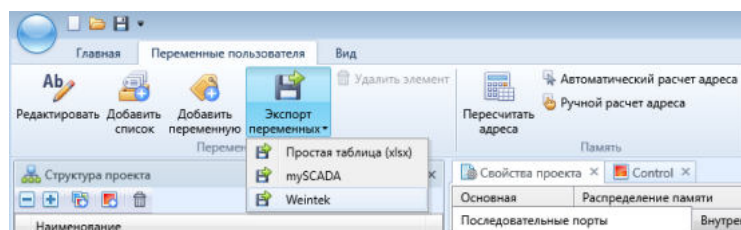


Рисунок 11. Меню экспорта списка переменных

В пункте основного меню «Экспорт переменных» можно выбрать формат выходных данных и осуществить экспорт. В зависимости от выбранного формата определенные в ручном режиме переменные будут сохранены в файл на диск.

11.2 Modbus RTU

Работа Modbus RTU осуществляется через последовательный порт контроллера только в режиме Slave.

Система адресации и порядок распределения памяти полностью аналогичен Modbus TCP Slave (раздел 11.1).

Однако команды доступа к Bool разделам памяти недоступны.

Для правильного использования режима Modbus RTU Slave в прикладной программе необходимо указать локальный адрес устройства и режим работы последовательного порта (Рис. 12).

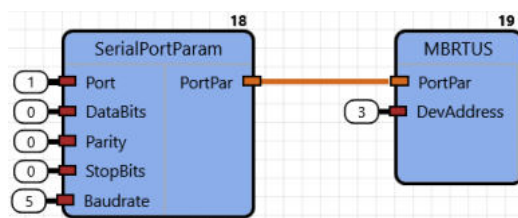


Рисунок 12. Настройка параметров Modbus RTU Slave

В блоке *SerialPortParam* указывается условный номер порта (для SMART KNOT он всегда равен 1), и параметры обмена:

DataBits – Код числа бит в слове обмена

Parity – Режим четности

StopBits – Число Стоп бит

Baudrate – Скорость обмена

Подробная информация по командам содержится в описании программирования ПЛК Серебрум

В примере (Рис. 20) **DataBits** = 0 (8 бит), **Parity** = 0 (нет бита четности), **StopBits** = 0 (один стоп бит), **Baudrate** = 7 (скорость обмена 57600 бод).

При получении корректных Modbus RTU запросов на указанный порт SMART KNOT автоматически сгенерирует и отправит ответ. Участия прикладной программы не требуется.

Участок кода, показанный на примере, Рис. 12, будет выполнен лишь в самом начале работы программы, соответственно параметры работы последовательного порта устанавливаются на всем протяжении времени жизни программы.

При необходимости изменить сетевой адрес или режим соединения потребуется перезагрузка прикладной программы контроллера.

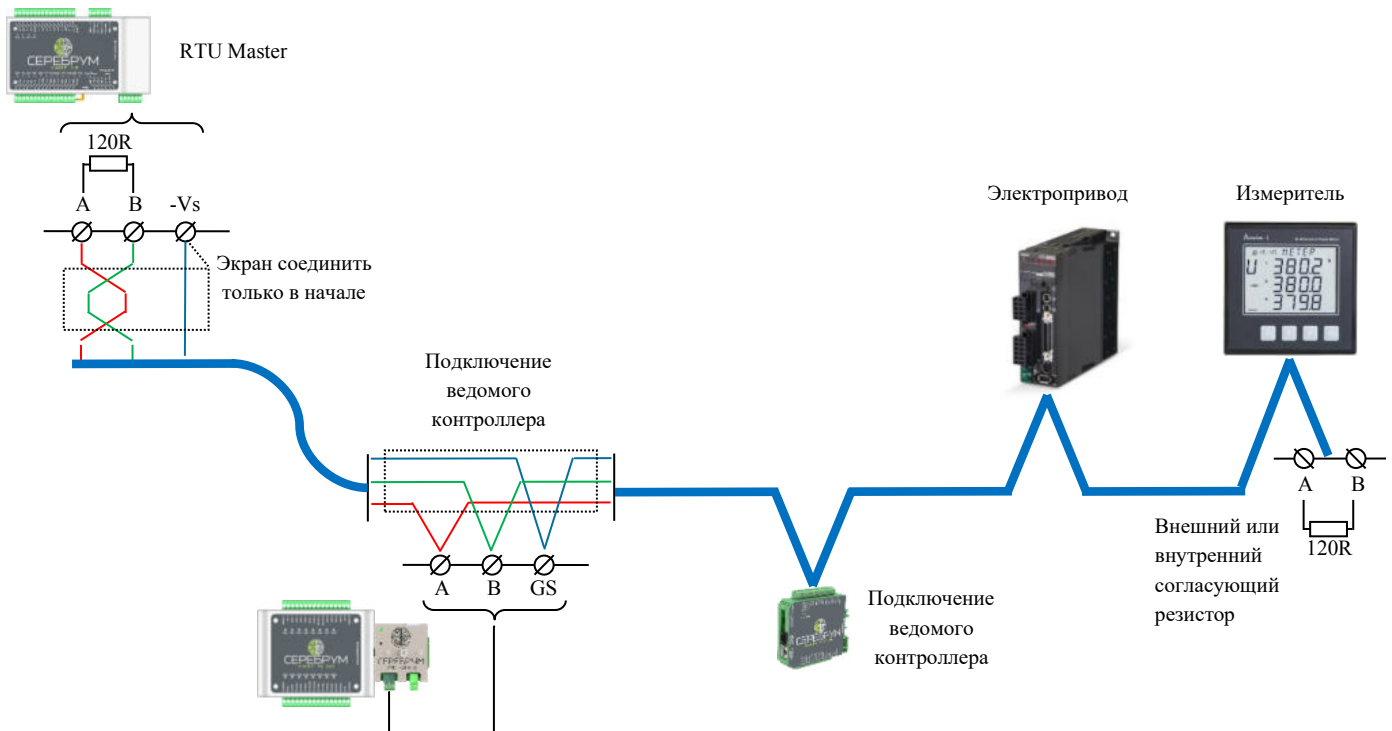


Рисунок 13. Подключение к шине Modbus RTU

На **Рисунке 13** показана приблизительная схема подключения устройств в шине RS-485 (Modbus RTU). В качестве ведущего устройства выступает контроллер YART 1.8

Поскольку в данной конфигурации ведущий физически расположен в самом начале шины, то для порта включается согласующий резистор 120 Ом.

На конечном устройстве (в данном примере это Измеритель) также должен быть подключен согласующий резистор.

11.4 YART Link

YARTLink — это протокол межконтроллерного обмена данными Серебрум. Он основан на UDP датаграммах и позволяет разворачивать децентрализованные решения на основе ПЛК Серебрум.

Использование YART Link дают возможность множеству контроллеров одновременно получать данные от SMART KNOT без необходимости отправки запросов на данные. Такой подход существенно экономит сетевой трафик.

Аналогичным образом несколько контроллеров SMART KNOT могут использовать один большой управляющий пакет от ведущего устройства, в котором будут храниться данные для всех используемых выходных каналов.

Поскольку YART Link поддерживает данные в формате с плавающей точкой, а также массивы – распределение выходных значений по каналам не потребуют больших вычислительных затрат.

При этом один источник может передавать данные множеству получателей без потери производительности и загрузки сетевого интерфейса.

Получатели данных, в свою очередь, "подписываются" на публичную рассылку от нужного источника. Данные приходят асинхронно от выполнения основной программы. Факт получения регистрируется временной меткой, по которой прикладная программа может сделать вывод о достоверности текущего значения.

Настройка работы YARTLink осуществляется отдельно на источнике и приемниках данных.

В источнике данных формируется датаграмма для передачи в сеть. Размер датаграммы ограничен и равен 1024 байт. В процессе работы программы пользователя датаграмма заполняется через блоки YL xx, где xx - тип данных переменной на входе блока, **Рис. 14**.

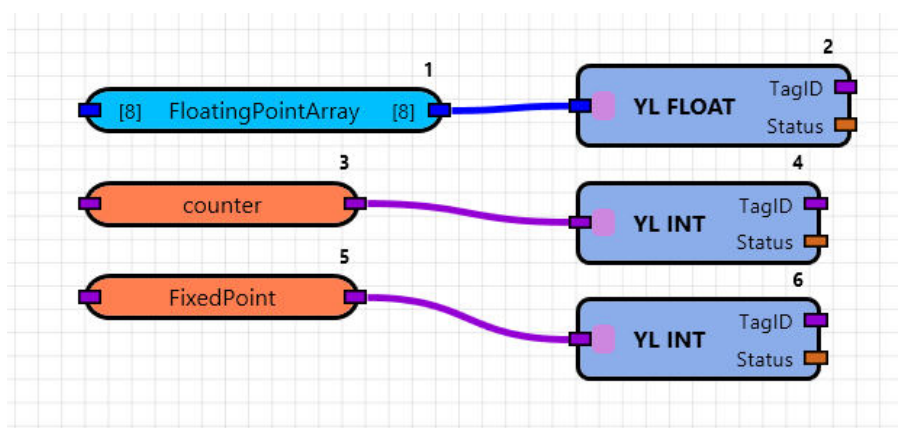


Рисунок 14. Пример программы передатчика YARTLink

В процессе работы программы выход блока формирования пакета выдает идентификатор переменной для подписки и статус выполнения операции.

Статус операции подготовки может принимать 3 значения:

- 0 – завершено без ошибок
- 1 – не корректный идентификатор переменной
- 2 – тип переменной не поддерживается
- 3 – пакет отправки переполнен

Отправка подготовленного пакета осуществляется или периодически, блоком "*YL SEND PERIODIC*", либо блоком "*YL SEND*", который начинает отправку блока при наличии "1" на входе.

Фактически датаграмма будет отправлена синхронно, после окончания выполнения такта программы пользователя, в котором был активирован флаг передачи сообщения.

Датаграмма отправляется в широковещательном режиме - дальнейшая обработка будет осуществлена на каждом подключенном устройстве.

Для получения опубликованных данных на приемном контроллере должна быть сконфигурирована подписка.

Для подписки (пример на **Рис. 15**) необходимо указать IP адрес передающего контроллера, идентификатор переменной, которая передается (выход Status блоков *YL XX*), а также ссылка на переменную куда будет записано значение.

Важной особенностью является то, что типы данных и длина должно полностью совпадать.

Выход Duration показывает время с момента получения последних данных от источника.

!

При реализации ответственных объектов данный параметр необходимо контролировать и переводить систему в безопасное состояние в случае, если данное время превысило предопределенный порог.

Каждая подписка должна иметь свой уникальный идентификатор (RecID), программируемый в диапазоне 0-63. В примере используется специализированный функциональный блок "ID CONTROL", формирующий код RecID на основе предыдущего.

Как видно из параметра RecID всего в контроллере допускается до 64 подписок на глобальные переменные одновременно.

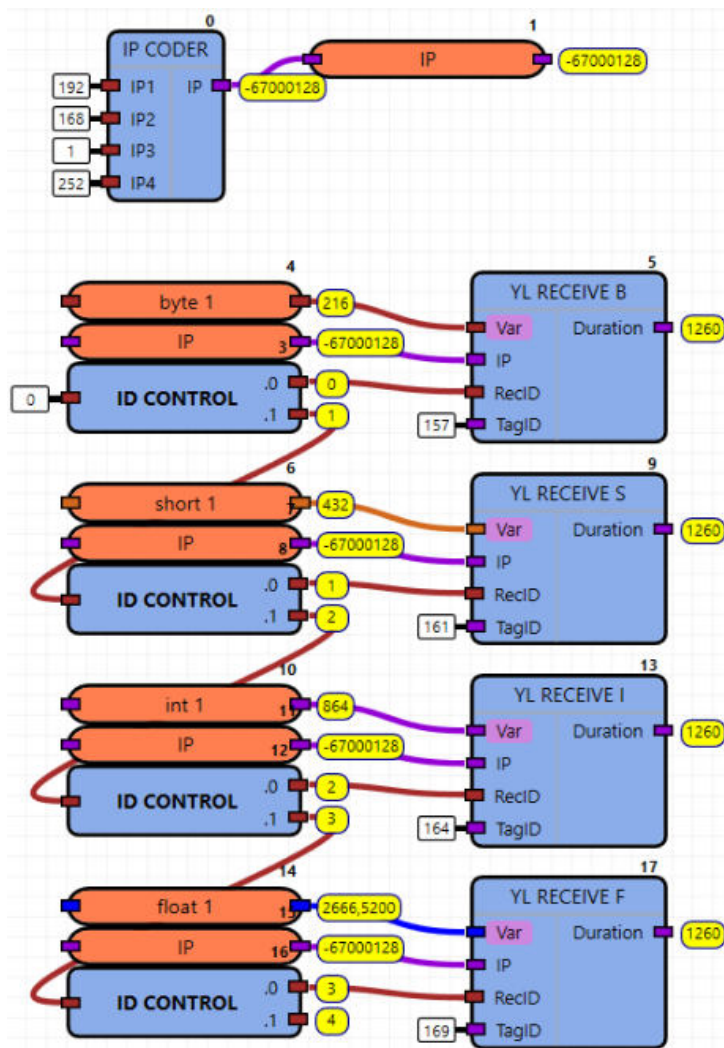


Рисунок 15. Подписка на глобальные публикуемые данные

12. Работа с подключаемыми периферийными модулями

Настройка работы внешних периферийных модулей производится в разделе «Состав оборудования». В данном разделе отображаются модули, добавленные в разделе настройки проекта – «Модули расширения» (**Рис. 6**).

Для модулей аналогового ввода необходим также выбор основного режима работы.

В процессе работы контроллера производится опрос внешних модулей расширения: модули ввода опрашиваются до старта очередного цикла программы, в модули вывода данные заносятся после выполнения программы.

Поскольку скорость выполнения прикладной программы может превышать физические характеристики периферийных модулей (например AI8) – фактические данные измерений будут обновляться не на каждом такте исполнения прикладной программы.

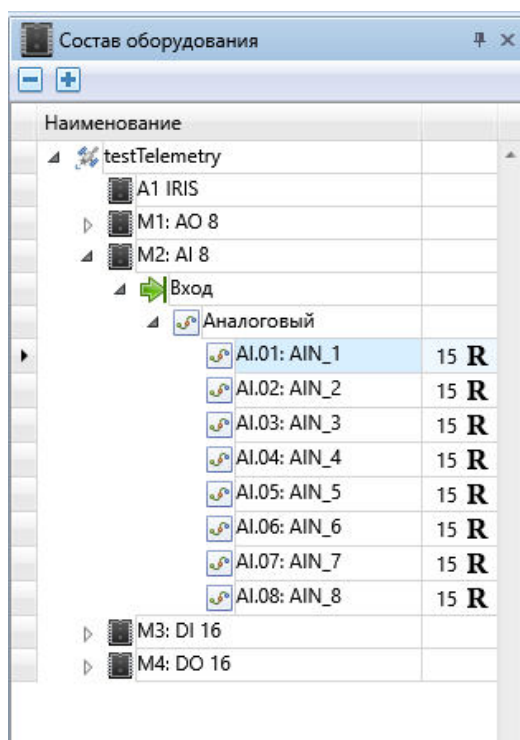


Рисунок 16. Панель настройки периферийных модулей

12.1 Аналоговые входы

Модуль AI8 (**Рис. 17, 18**) основан на 12-разрядном АЦП со встроенным программируемым усилителем.

Подключение модуля рекомендуется выполнять в соответствии со схемой (**Рис. 17**). При этом клеммы «Earth», 1, 16, 17, 32 должны подключаться к контуру заземления шкафа непосредственно в месте установки модуля.

Необходимо подключение всех клемм заземления.

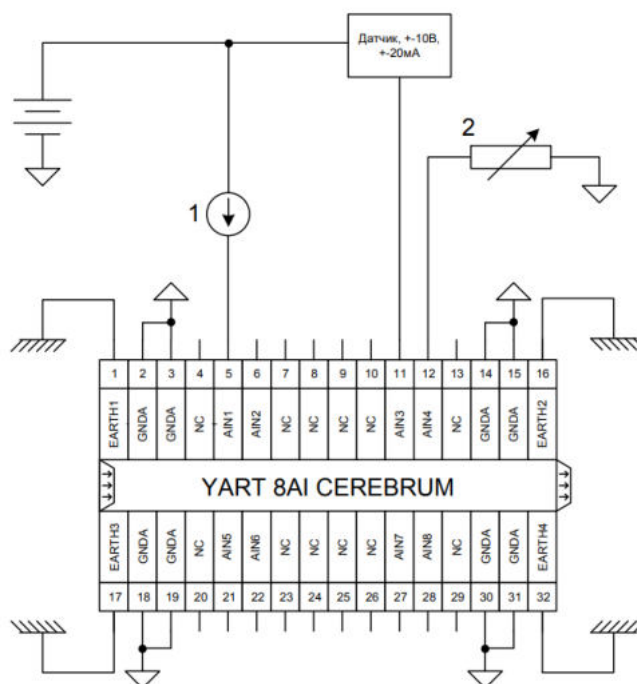


Рисунок 17. Модуль 8AI

На схеме (**Рис. 48**) цифрами обозначены:

- 1 – Токовый датчик, подключенный по 2-проводной схеме
- 2 – Измеряемое внешнее сопротивление (датчик температуры, дискретный вход)

Все каналы модуля имеют групповую гальваническую развязку от напряжения питания шины расширения и контроллера.

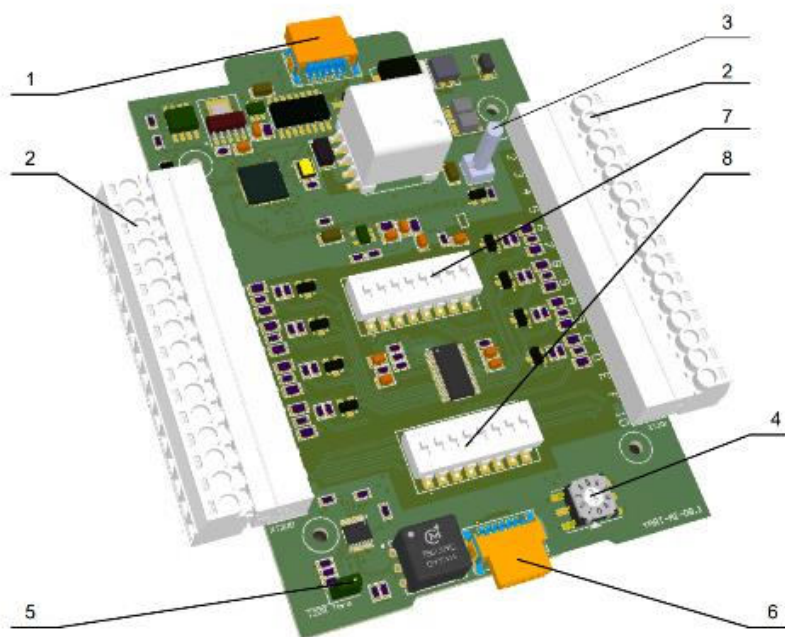


Рисунок 18. Модуль 8AI

- 1 – разъем для подключения к контроллеру или «ведущему» модулю расширения
- 2 – группа клемм для подключения внешних соединителей
- 3 – индикатор состояния работы модуля
- 4 – переключатель адреса в шине модулей расширения
- 5 – перемычка включения терминатора линии передачи данных
- 6 – разъем для подключения последующих модулей расширения
- 7 – переключатели 1 активации режима измерения сопротивления
- 8 – переключатели 2 активации режима измерения тока

Кроме того, входной каскад АЦП имеет внешние переключатели для выбора режима работы каналов. Доступно три основных режима (Таблица 11):

- измерение напряжения
- измерение тока
- измерение сопротивления

Таблица 6. Режимы работы модуля AI8

| Режим работы | Переключатель 1 | Переключатель 2 |
|-------------------------|-----------------|-----------------|
| Тестовый режим | Включен | Включен |
| Измерение сопротивления | Включен | Выключен |
| Измерение тока | Выключен | Включен |
| Измерение напряжения | Выключен | Выключен |

Каждый канал настраивается индивидуально. В зависимости от настроек программируемого усилителя изменяется входной диапазон измерений, в рамках которого работает 12-разрядный АЦП. Специальные блоки программного обеспечения ПЛК выполняют дальнейший пересчет полученных кодов АЦП, предоставляя пользователю уже обработанные данные (в физических величинах: В, мА, Ом).

Обратите внимание, что при измерении биполярного напряжения (или тока) 12 разрядов распределяются на весь диапазон измерения, т. е. 11 разрядов АЦП на положительное напряжение и 11 разрядов на отрицательное напряжение.

Также предусмотрены режимы, когда выходом блока AI8 являются необработанные данные АЦП.

Дальнейшие настройки выполняются в YARTStudio в соответствии с Таблицей 11

Измерение напряжения

Доступны восемь режимов:

- 0-1.25В / $\pm 0-1.25В$
- 0-2.5В / $\pm 0-2.5В$
- 0-5.0В / $\pm 0-5.0В$

Изменение тока

Все указанные диапазоны приведены с учетом подключаемого нагрузочного резистора 250 Ом.

- 0-20.0mA / $\pm 0-20.0mA$
- 0-10.0mA / $\pm 0-10.0mA$
- 0-5.0mA / $\pm 0-5.0mA$
- 0-2.5mA / $\pm 0-2.5mA$

Измерение входного сопротивления / дискретный вход

Измерение сопротивления осуществляется за счет измерения падения напряжения на входе канала при подаче нормированного тока возбуждения.

Схема измерения не является дифференциальной, поэтому подвержена влиянию синфазной помехи, возникающей на длинных кабелях.

При необходимости измерения сопротивления на расстояниях более 10 метров, а также при работе каналов измерения рядом с мощными силовыми установками рекомендуется использовать преобразователи сопротивление/ток.

Доступно три диапазона измерения:

- 0-4k
- 0-18k
- 0-1M

Канал измерения сопротивления способен имитировать дискретный вход. Программа автоматически определяет состояние низкого входного сопротивления, формируя «0» на выходе.

Для использования данных модуля АІ8 в прикладной программе (**Рисунок 19**) необходимо использовать индивидуальный функциональный блок Аіх или весь модуль сразу.

Добавление входа в программу осуществляется путем перетаскивания из выпадающего списка каналов (**Рис. 19**).

Пункт списка «Аналоговый» добавит все каналы одновременно.

Возможно добавления каждого канала по отдельности, выбирая элементы АІ.хх:АІN_х.

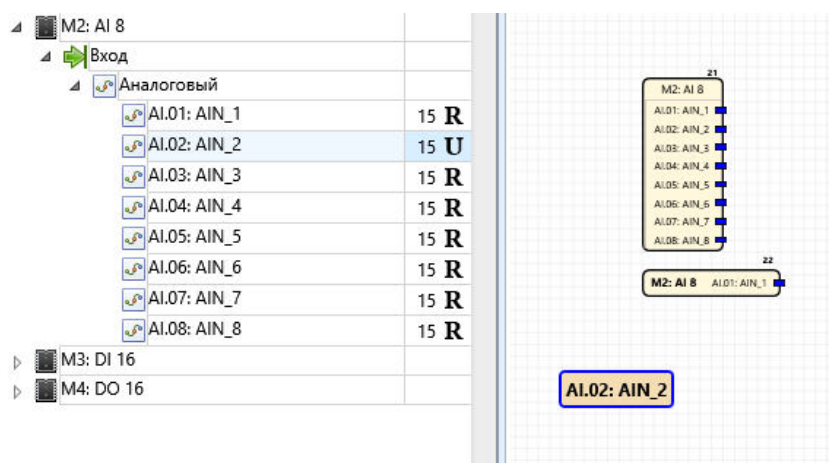


Рисунок 19. Функциональные блоки аналоговых входов

Выходы программных блоков модуля АІ формируют выходное значение в формате с плавающей точкой.

Значения выходов соответствуют измеренным физическим величинам в зависимости от выбранного режима работы (R, U, I, DI).

В режиме DI выход имеет тип Bool.

В состав встроенной библиотеки включены уже готовые функциональные блоки для осуществления дальнейших преобразований: датчики температуры, давления и т. д.

12.2 Аналоговые выходы

Модуль 8АО (**Рис. 20**) формирует восемь независимых каналов напряжения и тока. Формирователи тока и напряжения независимые и могут работать одновременно (каждый выходной канал имеет две клеммы: ток и напряжение).

Для работы токовых каналов модулю необходимо внешнее питание (поз. 1 и 2, **Рис 20**). Для устойчивой работы модуля достаточно запитать один вход – второй может оставаться в резерве.

Все каналы модуля имеют групповую гальваническую развязку от напряжения питания шины расширения и контроллера.

Для фильтрации помех предусмотрены клеммы 1, 17, 32, 16 (Earth), которые должны присоединяться к заземляющему контуру системы управления.

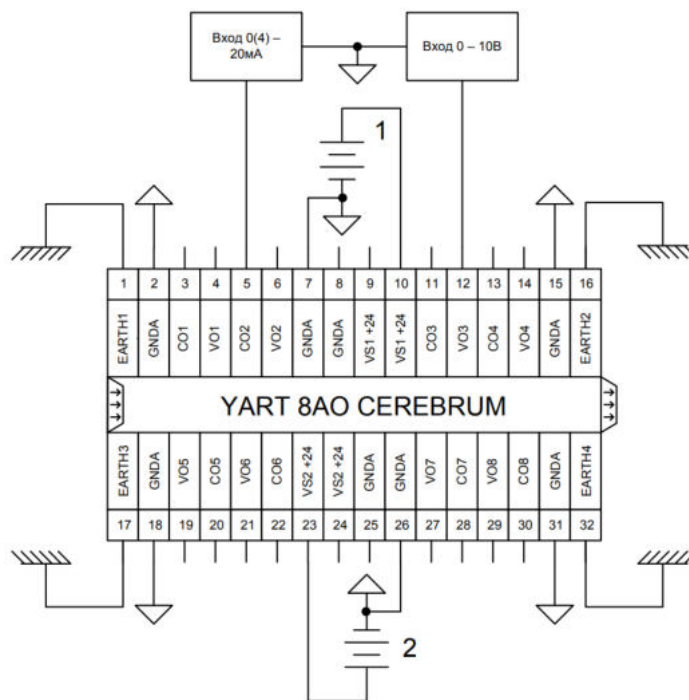


Рисунок 20. Модуль 8АО

Клеммы COx – токовые выходы 0 -20mA

Клеммы VOx – выходы напряжения 0 -10V

Для использования аналогового выхода в программе соответствующий блок необходимо разместить на программном поле, аналогично **Рис. 19**.

На вход блока (**Рис. 21**) подается число в формате с плавающей точкой в диапазоне 0 -100.0

0.0 соответствует выходному току/напряжению 0mA/0.0V

100 соответствует выходному току/напряжению 20mA/10.0V

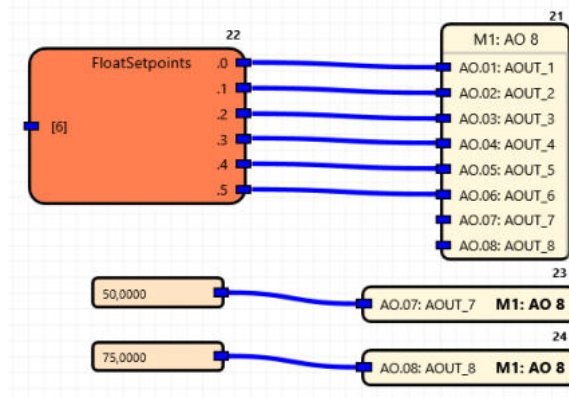


Рисунок 21. Использование модулей аналоговых выходов.

12.3 Дискретные входы

Модуль 16DI содержит 16 входных каналов для дискретных сигналов. Работа входных каскадов модуля осуществляется от внешнего источника питания. При этом все каналы модуля гальванически развязаны от напряжения питания шины расширения.

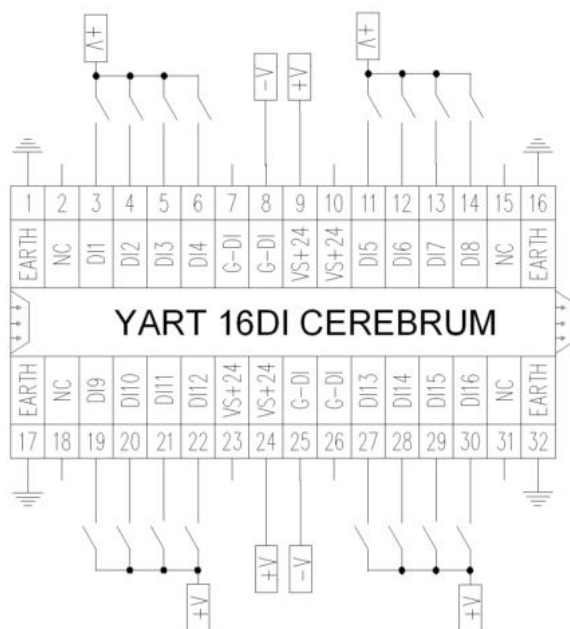


Рисунок 22. Модуль 16DI

Для устойчивой работы модуля достаточно присоединить питание к одной паре клемм.

Клеммы 1, 16, 17, 32 (Рис. 22) должны быть присоединены к внутреннему контуру заземления.

Аналогично модулям 8AI 8АО модуль 16DI представлен в виде функционального блока, доступного для размещения на поле программы (Рис. 23).

Блоки допускается размещать в любом месте программы.

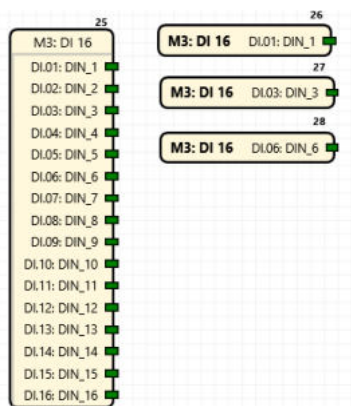


Рисунок 23. Блоки дискретного ввода

12.4 Дискретные выходы

Модуль 16DO содержит 16 выходных каналов с максимальной токовой нагрузкой 1А. Для каждого выходного блока из восьми линий (Рис. 24) предусмотрен отдельный вход питания.

При этом все каналы модуля гальванически развязаны от напряжения питания шины расширения.

Выходы модуля имеют встроенную защиту от перегрузки.

Клеммы 1, 16, 17, 32 (Рис. 24) должны быть присоединены к внутреннему контуру заземления.

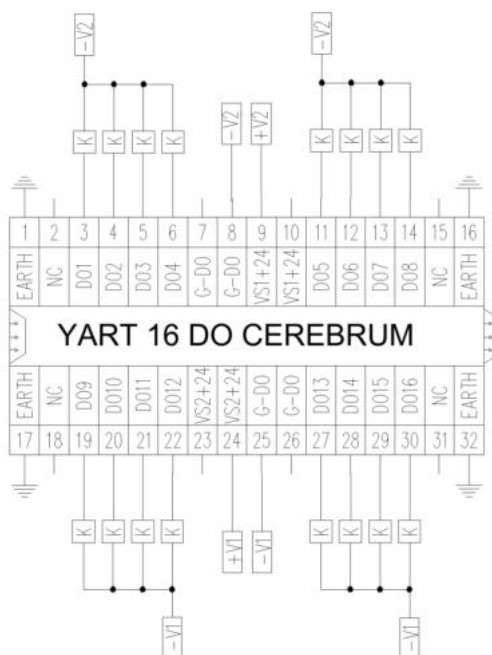


Рисунок 24. Модуль 16DO

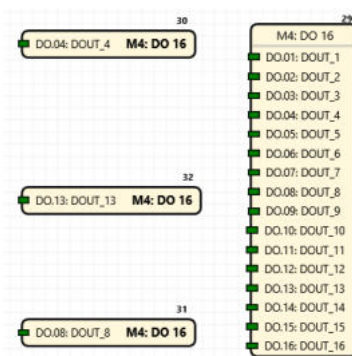


Рисунок 25. Функциональные блоки DO

Значения дискретных выходов задаются при помощи функциональных блоков DO (Рис. 25). Данные блоки могут быть размещены в любом месте программы. Компилятор YARTStudio автоматически проверяет однократное использование каждого выхода.

При многократной попытке записи в один и тот же канал будет сгенерирована ошибка компиляции.

13. Транспортировка и хранение

Контроллеры транспортируются в заводской упаковке в транспортной таре любым видом транспорта с защитой от дождя и снега. Крепление тары в транспортных средствах должно производиться согласно правилам, действующим на соответствующих видах транспорта.

Условия транспортирования должны соответствовать условиям 5 по ГОСТ 15150-69 при температуре окружающего воздуха от минус 40 до 50 °С с соблюдением мер защиты от ударов и вибраций.

Пребывание в условиях транспортирования – не более 3 месяцев.

Условия хранения в заводской упаковке на складе изготовителя и потребителя должны соответствовать условиям 1 по ГОСТ 15150-69. Наличие в воздухе агрессивных примесей не допускается.

После транспортирования при отрицательных температурах контроллеры перед включением необходимо выдержать в нормальных условиях не менее 24 ч

14. Гарантийные обязательства

Изготовитель гарантирует соответствие контроллера требованиям ТУ при соблюдении условий эксплуатации, транспортирования, хранения и монтажа.

Гарантийный срок– 36 месяцев со дня продажи.

В случае выхода контроллера из строя в течение гарантийного срока при соблюдении пользователем условий эксплуатации, транспортирования, хранения и монтажа предприятие-изготовитель обязуется осуществить его бесплатный ремонт или замену.