



Программируемый логический контроллер

СЕРЕБРУМ YART 1.8

Техническое руководство



DI 6	DO 4 DC	AI ^U 4 I	AO ^U 1 I	32GB < 	
Ethernet 			RS485	ШИНА модулей I/O	
MODBUS TCP RTU	Telemetry GSM/ Ethernet	UDP YART LINK	USER protocol	OPC UA	REST API
среда YART Studio	языки FBD C	отладка VIRTUAL PLC	отладка VISUAL элементы	AUTO PID	CEREBRUM CLOUD
TTL- HTL	SIN/ COS	SSI	EnDat	HTL	

Оглавление

2. Предупреждения	4
3. Список изменений.....	5
4. Общая информация.....	6
5. Технические характеристики	7
6. Подключение	10
7. Начало работы	12
7.1 Настройки сети	13
8. Использование модулей расширения.....	14
9. Индикация состояний	17
10. ПЛК, особенности архитектуры и память	18
10.1 Память программируемого контроллера.....	19
10.1.1 Системная память	19
10.1.2 Оперативная память.....	19
10.1.3 Энергонезависимая память (FRAM)	19
10.1.4 Батарейная память.....	20
10.2 Типы данных, используемых в системе	22
11. Телеметрия.....	23
11.1 Подключение к удаленному серверу телеметрии	24
12. Модули обмена данными	28
12.1 Modbus TCP.....	28
12.2 Modbus RTU	31
12.2.1 Modbus RTU Master.....	34
12.3 Пользовательские протоколы.....	36
12.4 Yart Link.....	38
13. Работа с периферийными модулями контроллера.....	41
13.1 Аналоговые входы.....	41
13.2 Дискретные входы	43
13.3 Подключение датчиков положения	44
13.4 Аналоговый выход	46
13.5 Дискретные выходы	47
14. Работа с внешними периферийными модулями	49
14.1 Аналоговые входы.....	50
14.2 Аналоговые выходы	54

14.3 Дискретные входы	56
14.4 Дискретные выходы	57
15. Транспортировка и хранение	58
16. Гарантийные обязательства	59



2. Предупреждения



3. Список изменений

4. Общая информация

Контроллер Green Motion — промышленный логический контроллер (ПЛК) со встроенной системой удаленного мониторинга и управления СЕРЕБРУМ.

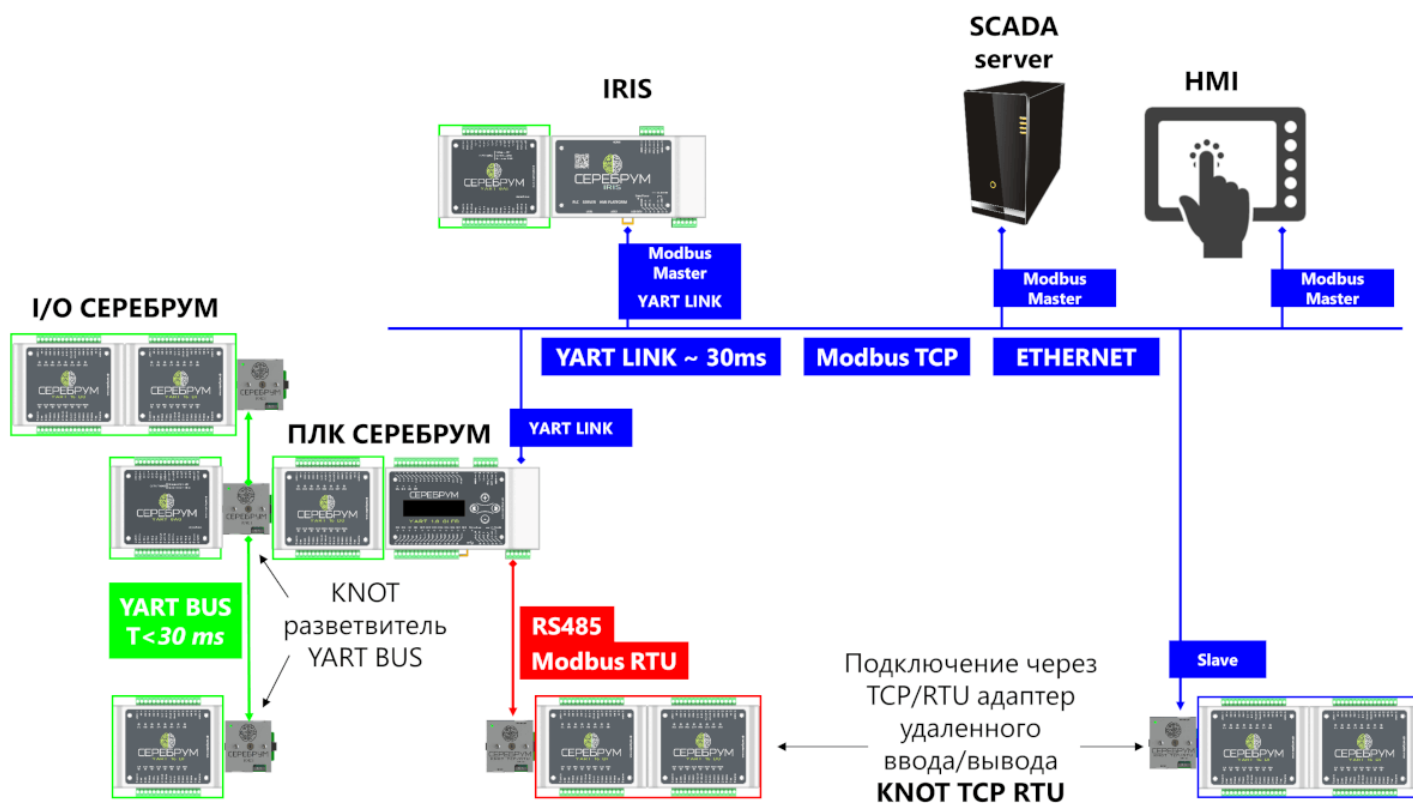
Green Motion может использоваться в качестве промышленного контроллера в локальных и распределенных системах управления с возможностью подключения удаленных моделей ввода/вывода через Ethernet или RS-485.

Важной особенностью контроллера является наличие периферийных модулей для подключения промышленных датчиков скорости и положения. Универсальный порт энкодера поддерживает работу с TTL, HTL, SinCos, SSI и EnDAT датчиками. Второй порт физически соединен с линиями дискретного входа и поддерживает подключение HTL энкодеров.

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

Green Motion поддерживает одновременную работу с серверами телеметрии СЕРЕБРУМ, Modbus TCP Slave, ModbusRTU Master/Slave, Yart-Link и до семи подключенный модулей расширения через шину YART-BUS.

Физическое соединение с Internet осуществляется через Ethernet.



5. Технические характеристики

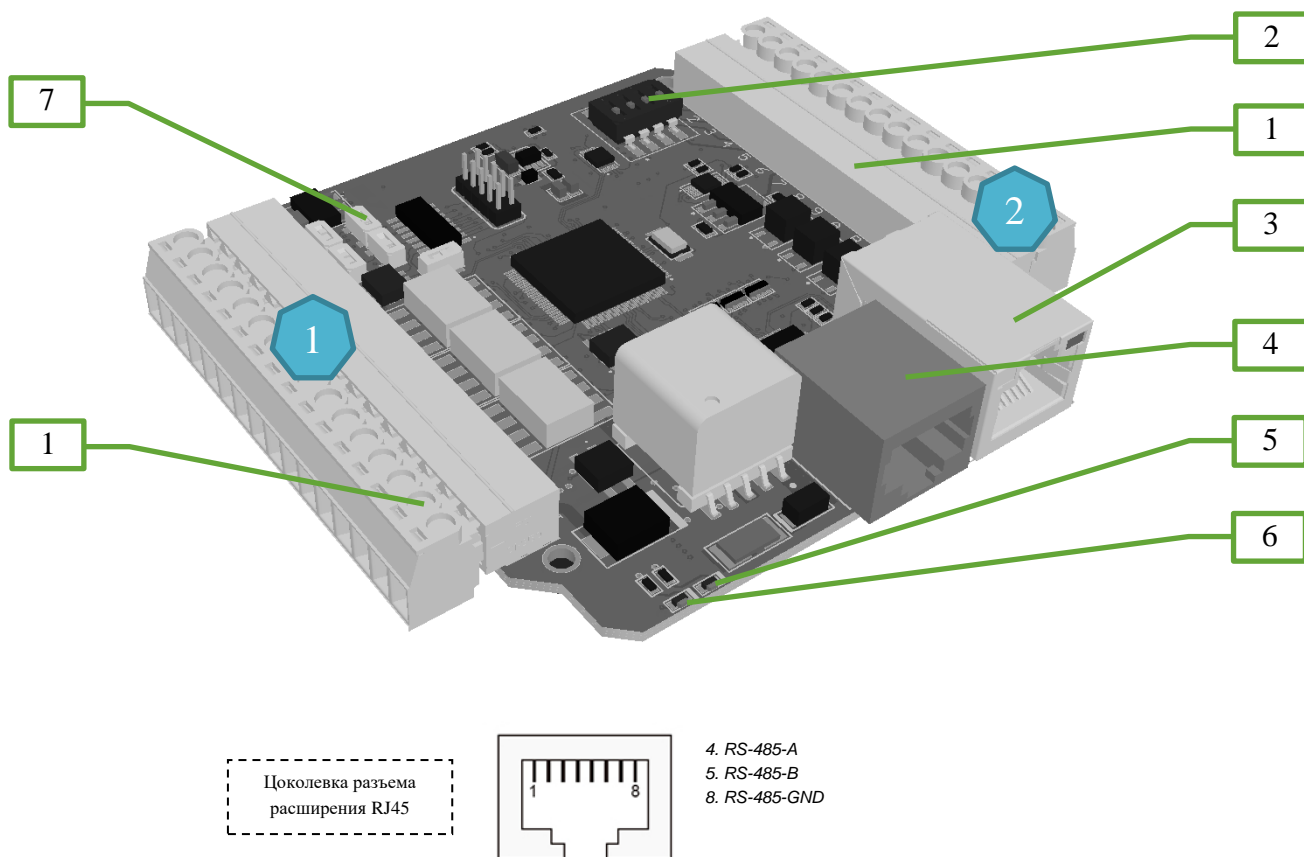


Рисунок 1. Плата контроллера GREEN MOTION

1. Клеммники
2. Переключатель токового режима аналоговых входов
3. Разъем RJ-45, Ethernet
4. Разъем RJ-45 последовательного порта и шины YART BUS
5. LED2 – Программируемый индикатор
6. LED1 – Индикатор статуса контроллера
7. Переключатель порта энкодера №1



Таблица 1. Технические характеристики GREEN MOTION

Параметр		Характеристики
Напряжение питания		от 10 до 28 VDC
Тип разъемов клеммника		Разъемные, под винт, максимальное сечение провода 1.5 мм ²
Программирование		ФБД, C-YART, среда программирования YART Studio
Память	Программ	1218 KB энергонезависимая память (более 500 типов блоков, 1200 вызовов блоков)
	RAM	40 KB память для переменных пользователя
	FRAM	16 KB энергонезависимая память для переменных пользователя
	BRAM	отсутствует
	RAM BIT	2 KBIT память для переменных пользователя
	FRAM BIT	2 KBIT энергонезависимая память для переменных пользователя
uSD карта памяти		отсутствует
Выполнения цикла программы		от 1 ms
Входы	Дискретные	6DI x 7-28 VDC, гальванически изолированы от цепей процессора
	Аналоговые	4AI x 12 бит. Режимы - U/I (0-10VDC/0-20mA)
Выходы	Дискретные	4DO x оптотранзисторы, ток коммутации до 0,15 А. Гальванически изолированы



	Аналоговые	1АО x I (0-20mA), 10 бит
Коммуникационные порты	ETHERNET	Разъем RJ-45, 100/10Base-T автоопределение типа кабеля. YART-LINK, MODBUS TCP, IoT
	mini-USB	Отсутствует
	Порт энкодера	Инкрементальные: SinCos, TTL, HTL Абсолютные: SSI, EnDat
	RS485	Отсутствует
	RS485I	MODBUS RTU, протоколы библиотеки, пользовательский, гальванически изолирован от цепей процессора
	YART-PORT	Шина модулей расширения, период опроса всех модулей менее 30 ms, максимальное число подключаемых модулей – 7
Часы реального времени (RTC)		Отсутствует

6. Подключение

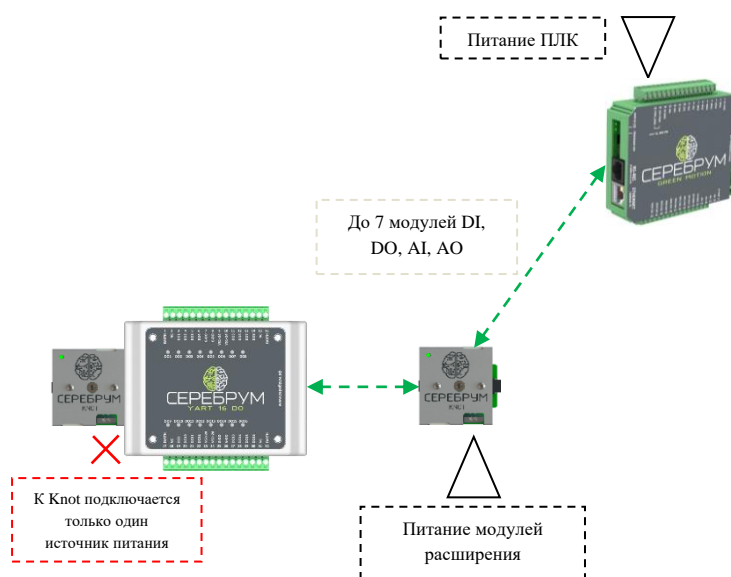


Рисунок 2. Подключение источников питания

Для работы контроллера требуется наличие внешнего питания (см. **Таблицу №1**). Линия питания выведена на клеммы 15-16 клеммника 1. Выводы разъема подключения модулей расширения – YART-BUS не содержат цепи питания – модули расширения контроллера должны запитываться отдельно, через коммутаторы Knot.

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

! **Внимание**

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

Таблица 2. Сигналы разъема RJ45 (Поз. 4, Рис. 1)

Номер клеммы	Наименование	Назначение
1	YART-BUS-1	Линия YART BUS
2	YART-BUS-2	Линия YART BUS
4	485-A	Линия RS-485-A
5	485-D	Линия RS-485-B
8	485-GND	Общий линии RS-485 (изолирован от входного питания)

Таблица 3. Сигналы клеммника 1

Номер клеммы	Наименование	Назначение
1	DI-GND	Общий сигнал дискретных входов. Развязан от питания
2	ING1	Порт энкодера 1, вход 1 (линия А)
3	ING2	Порт энкодера 1, вход 2 (линия А)
4	ING3	Порт энкодера 1, вход 3 (линия В)
5	ING4	Порт энкодера 1, вход 4 (линия В)
6	INS1	Порт энкодера 1, линия синхронизации
7	INS2	Порт энкодера 1, линия синхронизации
8	DI1	Дискретный вход 1. Линия А энкодера 2
9	DI2	Дискретный вход 2. Линия В энкодера 2
10	DI3	Дискретный вход 3
11	DI4	Дискретный вход 4
12	DI5	Дискретный вход 5
13	DI6	Дискретный вход 6
14	DI-GND	Общий сигнал дискретных входов. Развязан от питания
15	24V PWR	Вход питания контроллера
16	PWR_GND	Общий питания контроллера

Таблица 4. Сигналы клеммника 2

Номер клеммы	Наименование	Назначение
17	AI GND	Общий аналоговых входов
18	AI1	Аналоговый вход 1 (U, I)
19	AI2	Аналоговый вход 2 (U, I)
20	AI3	Аналоговый вход 3 (U, I)
21	AI4	Аналоговый вход 4 (U, I)
22	AI GND	Общий аналоговых входов
23	AO GND	Общий аналогового выхода
24	AO OUT	Аналоговый выход, 0-20мА
25	DO1 EM	Дискретный выход 1, эмиттер
26	DO1 COL	Дискретный выход 1, коллектор
27	DO2 EM	Дискретный выход 2, эмиттер
28	DO2 COL	Дискретный выход 2, коллектор
29	DO3 EM	Дискретный выход 3, эмиттер
30	DO3 COL	Дискретный выход 3, коллектор
31	DO4 EM	Дискретный выход 4, эмиттер
32	DO4 COL	Дискретный выход 4, коллектор



7. Начало работы

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

По умолчанию контроллер имеет IP адрес: **192.168.1.254**

При утрате IP адреса и невозможности установить соединение необходимо произвести сброс настроек контроллера. Для этого необходимо выполнить следующую последовательность действий:

1. Отключите питание от контроллера
2. Снимите крышку корпуса.
3. Замкните внутренние клеммы «RST», расположенные рядом с переключателем режимов режимов энкодера (**Рис. 3**) и подайте питание контроллера.
4. Снимите питание, перемычку и снова подайте питание. Настройки контроллера будут сброшены.

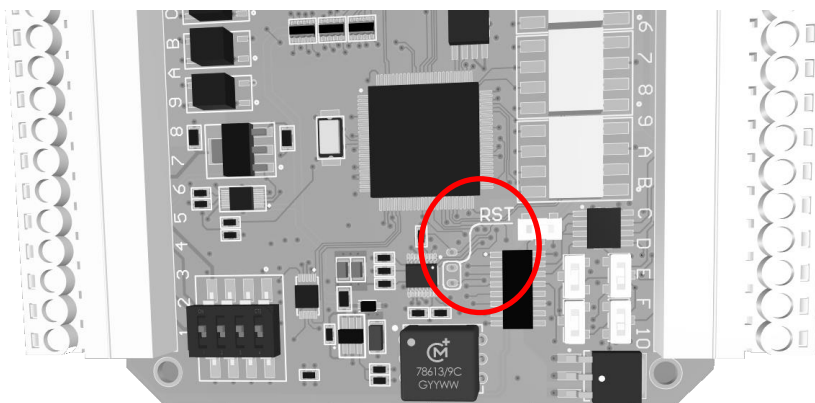


Рисунок 3. Клеммы сброса настроек



7.1 Настройки сети

Настройка сетевого интерфейса осуществляется помощи Yart Studio.

По умолчанию Green Motion получает статический IP адрес равный **192.168.1.254**

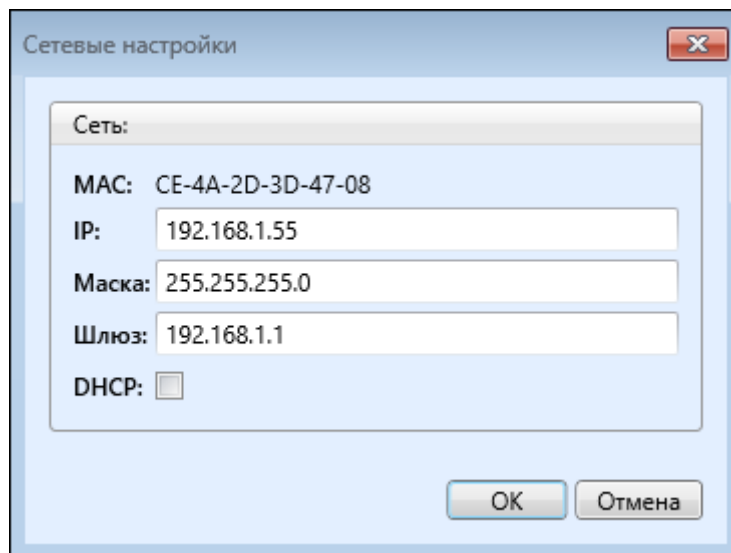


Рисунок 4. Настройка сетевого подключения

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

В отобразившемся окне (**Рис. 4**) 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

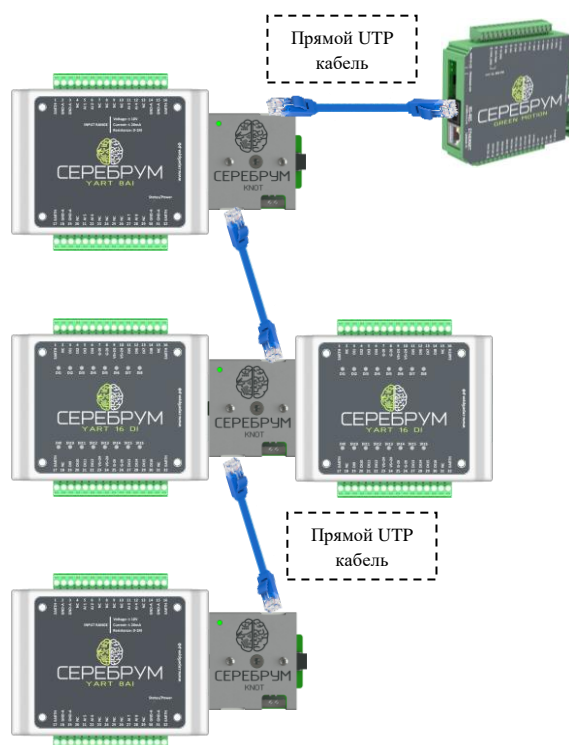


Рисунок 5. Возможная топология шины расширения

Пример возможной конфигурации шины YART-BUS представлен на **Рисунке 5**.

При проектировании конфигурации шины необходимо учитывать следующие правила:

- Контроллер GREEN MOTION 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 6. Список определенных модулей

GREEN MOTION 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.

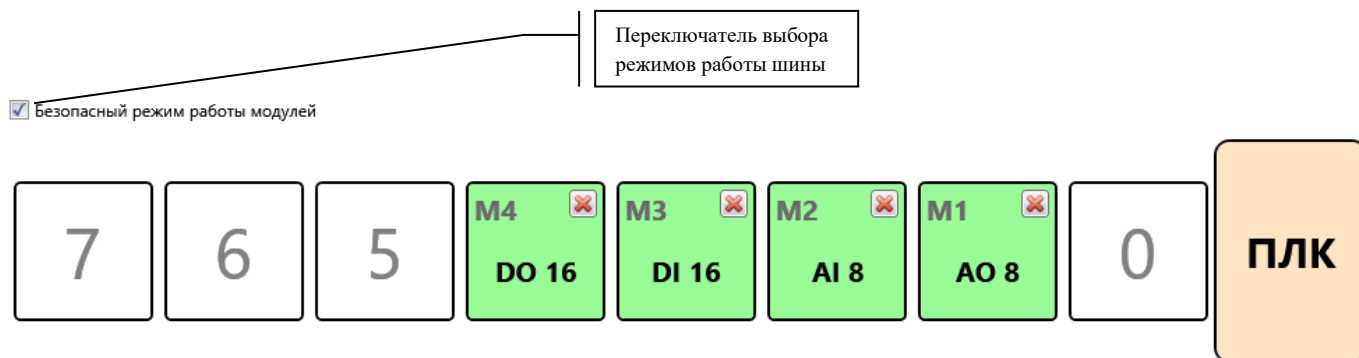


Рисунок 7. Конфигурация модулей расширения

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 GREEN MOTION доступны адреса модулей в диапазоне 1 – 7.

Пример модуля расширения и расположение переключателей показан на **Рисунке 8**.

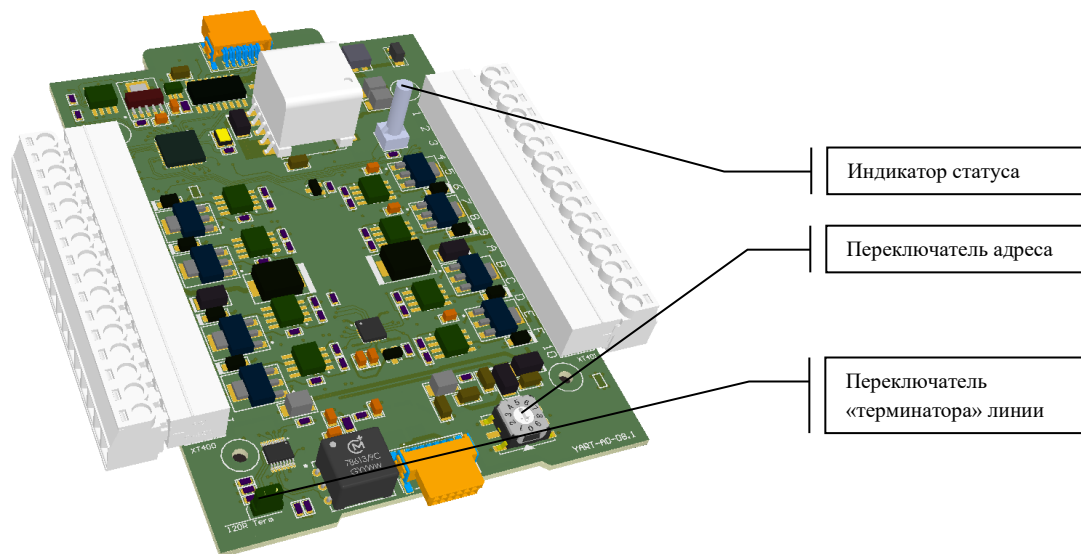


Рисунок 8. Переключатели модулей расширения

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 GREEN MOTION осуществляется при помощи светодиодного индикатора (поз. 6 Рис. 1)

Различные режимы отображения представлены в **Таблице 5**.

Таблица 5. 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



10. ПЛК, особенности архитектуры и память

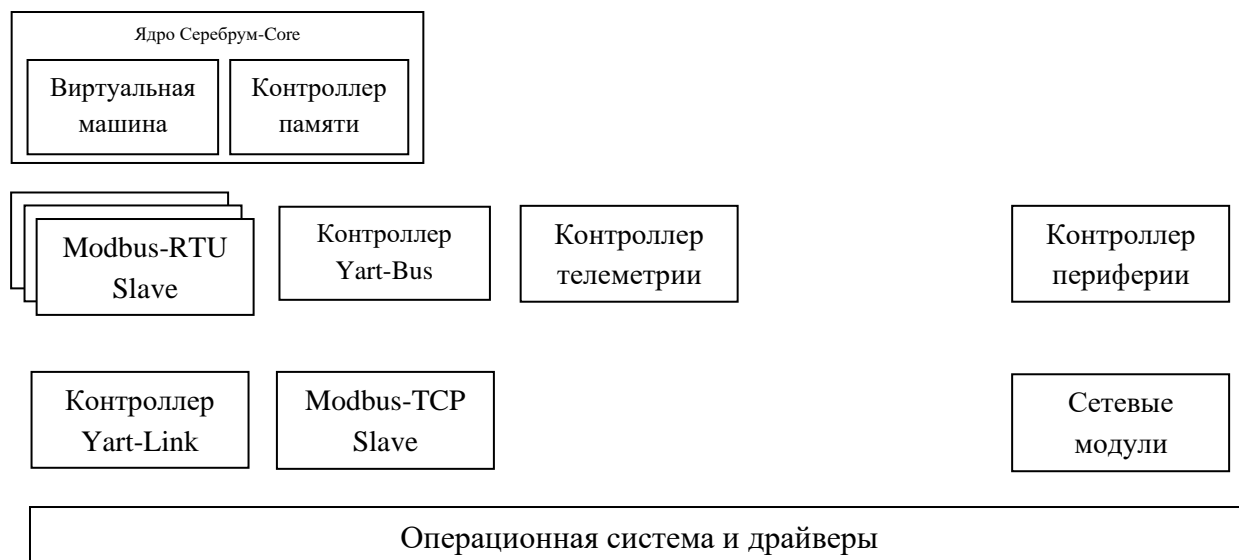


Рисунок 9. Архитектура контроллера GREEN MOTION

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.

GREEN MOTION содержит встроенные периферийный модули для непосредственного подключения к линиям датчиков и управления.



10.1 Память программируемого контроллера

Вся память контроллера (**Таблица 6**) 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

Контроллер Green Motion не оснащен батареей, поэтому использование BRAM в прикладных программах пользователя нецелесообразно.

Страница 21 из 59



10.2 Типы данных, используемых в системе

- **Bool** – Битовая переменная.

Данные переменные хранятся в Bool сегментах (Таблица 8).

В C-Yart доступны предопределенные 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. Телеметрия

Все программируемые контроллеры Серебрум изготавливаются с поддержкой системы удаленного мониторинга и управления. В зависимости от типа используемого оборудования возможно подключение через Ethernet.

Все подключенные контроллеры имеют встроенную возможность передавать через защищенное соединение данные на удаленный сервер, принимать команды и отображать в пространстве программы пользователя данные о состоянии сетевого подключения.

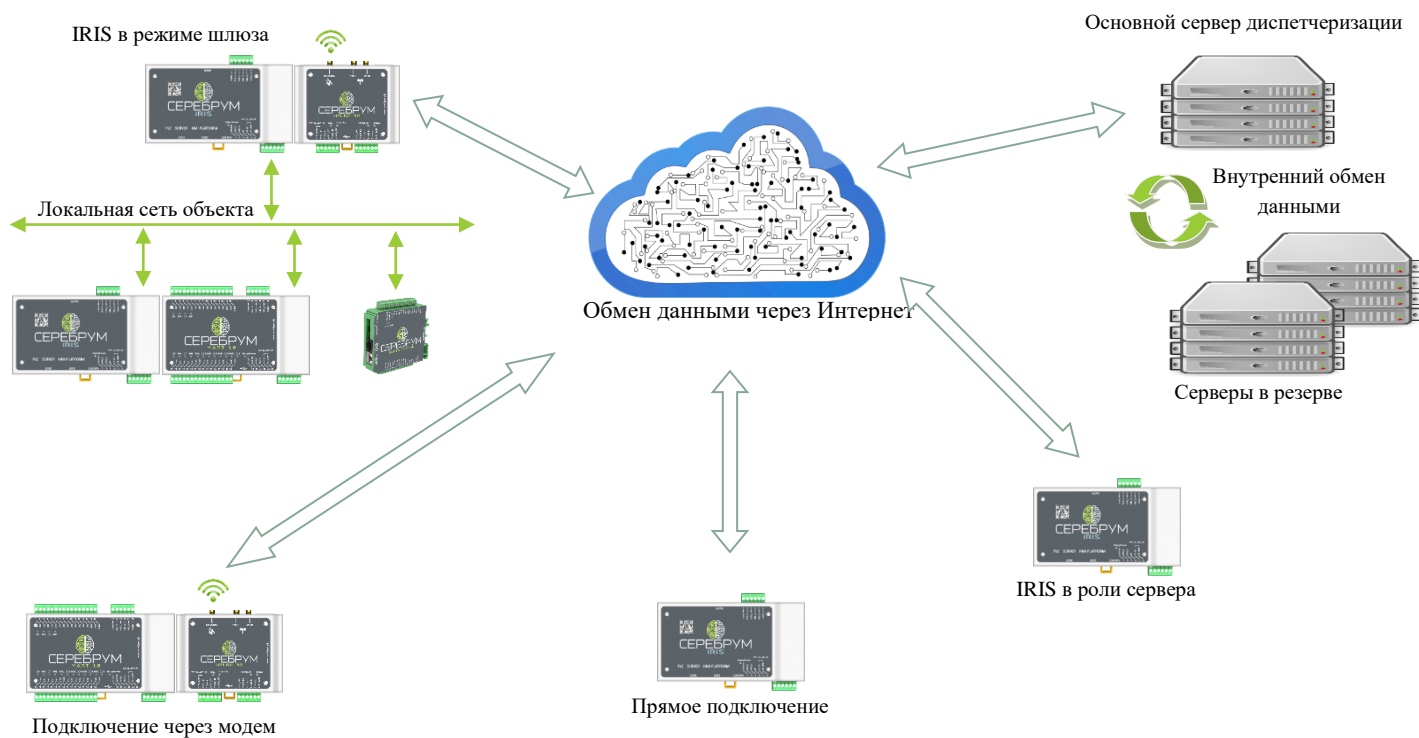


Рисунок 10. Удаленная диспетчеризация и управление Серебрум

Инфраструктура системы телеметрии Серебрум (Рис. 10) состоит из выделенного сервера и множества удаленных контроллеров. При необходимости вся система может быть развернута в локальной сети или сети предприятия.

Головная часть состоит из одного или нескольких серверов, которые могут быть связаны друг с другом для обеспечения горячего резервирования данных.



11.1 Подключение к удаленному серверу телеметрии

Контроллер GREEN MOTION поддерживает работу в режиме удаленного модуля телеметрии. При этом подключение к Интернет может быть осуществлено через встроенный интерфейс Ethernet, а также через внешний модем Neuro.

Выбор режима работы осуществляется при настройке подключения в Yart Studio или динамически при использовании программного интерфейса настроек.

Для настройки телеметрии необходимо выполнить ряд действий.

Шаг 1. В окне настройки проекта (**Рис. 11**) выберите интерфейс передачи данных «Ethernet», а также укажите настройки соединения с сервером телеметрии

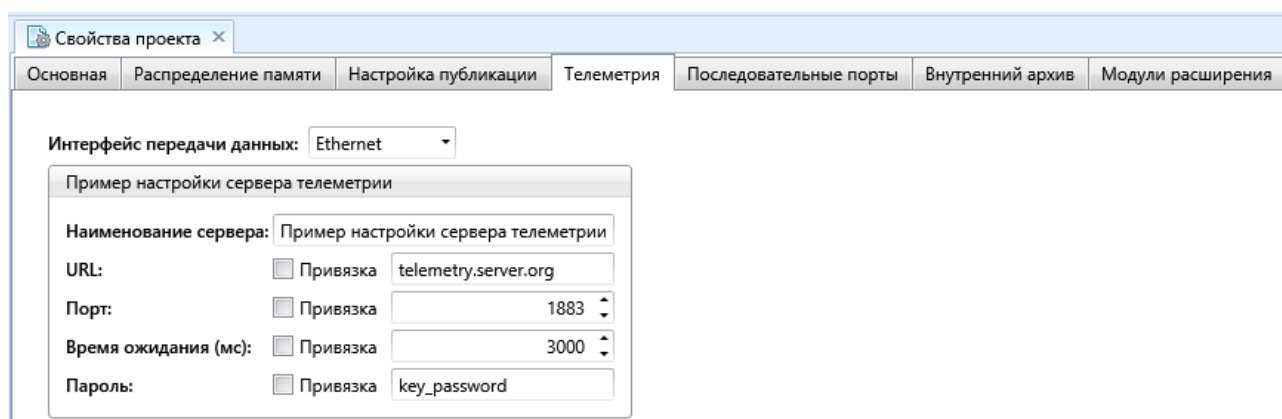


Рисунок 11. Простая настройка телеметрии

В том случае если контроллер еще не был зарегистрирован в системе – оставьте значение поля «Пароль» равным «*hitchhiker*».

В поле URL укажите адрес основного сервера телеметрии.

Порт – 1883, время ожидания – рекомендуется использовать значение 3000

В данном примере показана базовая настройка, не предполагающая смену рабочих серверов при работе контроллера.

Для динамического управления подключениями следует использовать режим привязок.

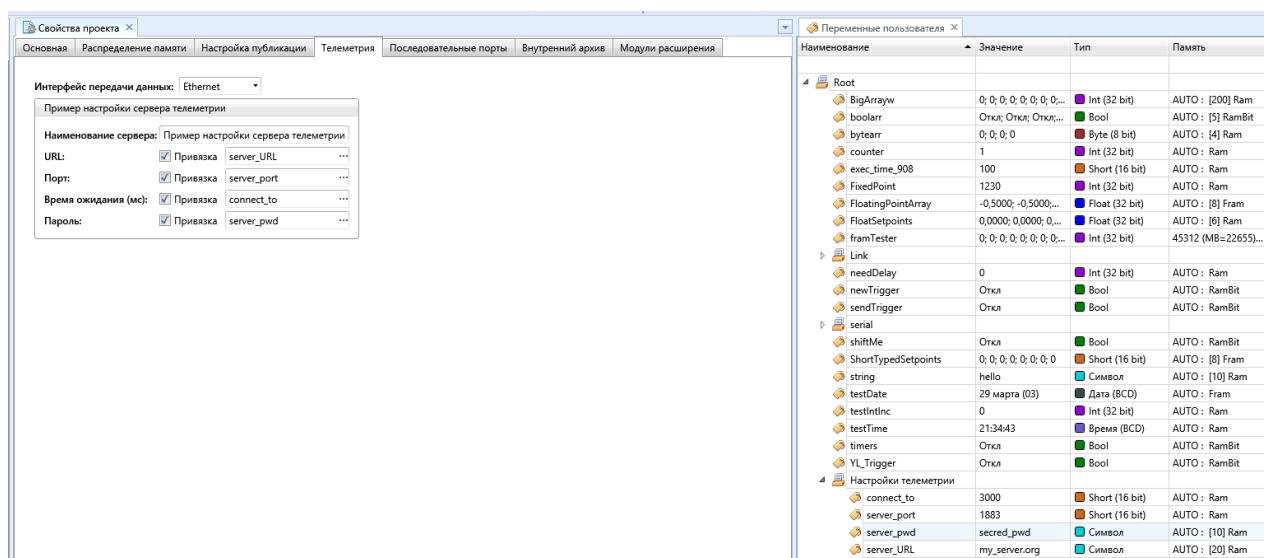


Рисунок 12. Привязка параметров подключения к серверу к переменным

На **Рисунке 12** показан пример использования переменных для настройки подключения к серверу. Значения переменных объявлены заранее, однако в процессе работы их можно изменить и дать команду повторного подключения.

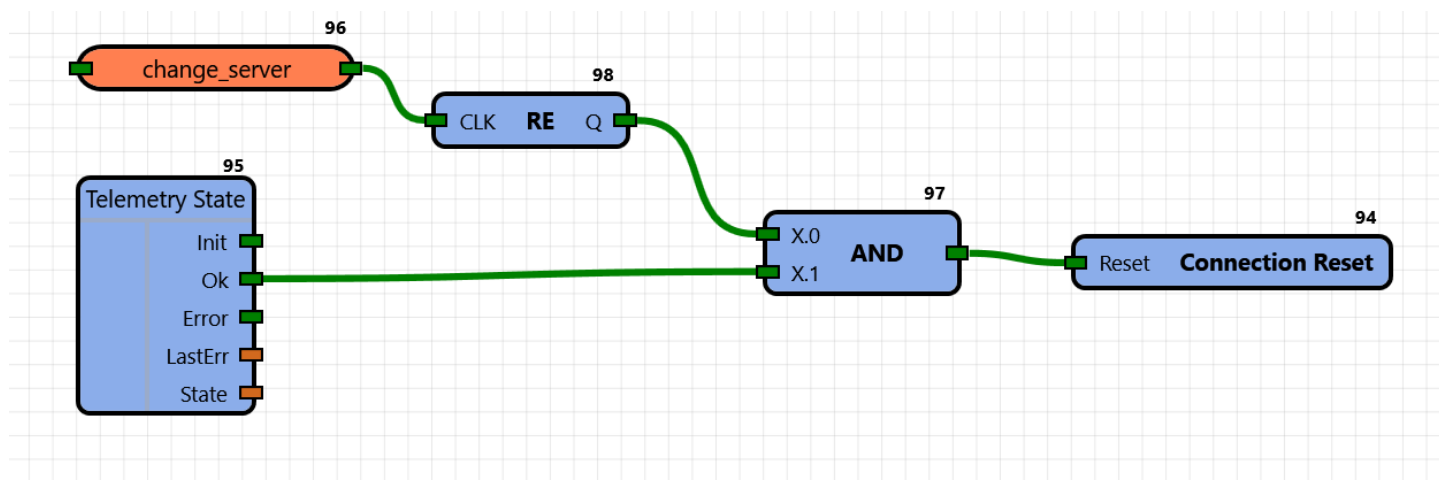


Рисунок 13. Пример команды сброса соединения с сервером

Перед каждой попыткой соединения с сервером программа телеметрии будет считывать значения переменных, указанных в виде ссылок (**Рис. 12**). Если соединение установить не удалось, программа снова считывает эти данные и повторит попытку.


Если требуется установить соединение с другим сервером, то в приложении необходимо предусмотреть место, где переменные **server_URL**, **server_pwd**, **server_port** и **connect_to** будут изменены.

Для сброса и обновления данных подключения работающего соединения следует использовать блок “Connection Reset”, **Рисунок 13**.

На этом примере соединение сбросится и новые данные будут скопированы из привязанных переменных при наличии успешного соединения (блок Telemetry State) и нарастающего фронта “change_server”.

Параметры подключения можно изменять удаленно, поместив данные основных и резервных серверов в список публикуемых переменных.

Шаг 2. Определение списков удаленных переменных

 Публикация параметров

Наименование	Значение	Тип	Память	Ко
Root				
Link				
serial				
sendTrigger	Откл	Bool	RamBit: AUTO	
FixedPoint	1230	Int (32 bit)	Ram: AUTO	
FloatingPointArray	-0,5000; -0,5000; 1,...	Float (32 bit)	Fram: AUTO [8]	
timers	Откл	Bool	RamBit: AUTO	
newTrigger	Откл	Bool	RamBit: AUTO	
counter	1	Int (32 bit)	Ram: AUTO	
shiftMe	Откл	Bool	RamBit: AUTO	
ShortTypedSetpoints	0; 0; 0; 0; 0; 0; 0	Short (16 bit)	Fram: AUTO [8]	
YL_Trigger	Откл	Bool	RamBit: AUTO	
testIntInc	0	Int (32 bit)	Ram: AUTO	
framTester	0; 0; 0; 0; 0; 0; 0; 0;...	Int (32 bit)	Fram: 45312 (MB=...	
testDate	29 марта (03)	Дата (BCD)	Fram: AUTO	
needDelay	0	Int (32 bit)	Ram: AUTO	
BigArrayw	0; 0; 0; 0; 0; 0; 0; 0;...	Int (32 bit)	Ram: AUTO [200]	
testTime	21:34:43	Время (BCD)	Ram: AUTO	
boolarr	Откл; Откл; Откл;...	Bool	RamBit: AUTO [5]	
bytearr	0; 0; 0; 0	Byte (8 bit)	Ram: AUTO [4]	
string	hello	Символ	Ram: AUTO [10]	
FloatSetpoints	0,0000; 0,0000; 0,0...	Float (32 bit)	Ram: AUTO [6]	
exec_time_908	100	Short (16 bit)	Ram: AUTO	
Настройки телеме...				

Подписка:

FixedPoint FloatingPointArray
ShortTypedSetpoints testDate
BigArrayw testTime boolarr
bytearr string FloatSetpoints

newTrigger

FixedPoint FloatingPointArray
counter ShortTypedSetpoints
testDate testTime boolarr
bytearr string

Рисунок 14. Параметры публикации

Все публикуемые переменные условно разделяются на два множества: переменные в подписке и переменные для публикации.

В списке «Подписка» (Рис. 14) находятся переменные, которые допускается изменять со стороны сервера – настройки, параметры работы и прочие данные.

Шаг 3. Публикация данных

Отправка данных на сервер осуществляется по флагам передачи. Флагом является любая переменная типа bool, которая объявлена в таком качестве. Для этого требуется перенести нужную переменную на правое поле – появится серый контейнер, куда аналогично можно помещать переменные, значения которых будут отправлены при наличии «1» в бите соответствующего флага.



В примере выше в качестве флага используется переменная “newTrigger”. После того, как программа присвоит данной переменной значение «1» на текущий сервер будут отправлены значения переменных:

FixedPoint, FloatingPointArray, counter, ShortTypedSetpoints, testDate, testTime, boolarr, bytearr, string.

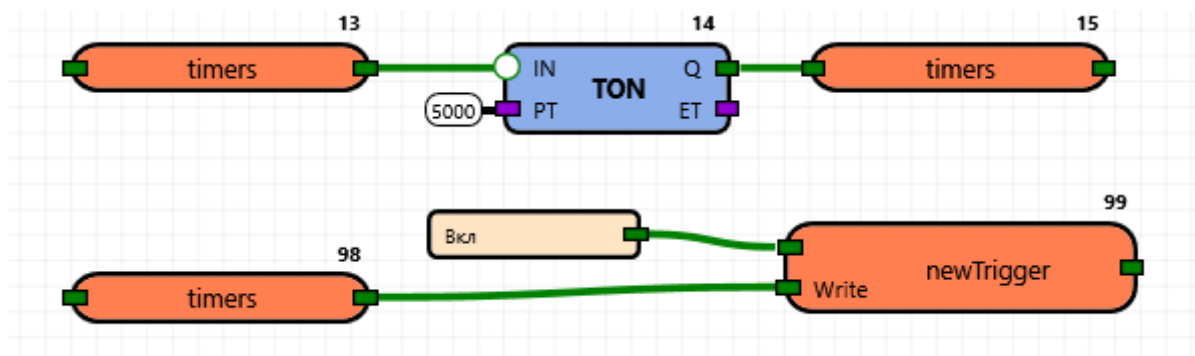


Рисунок 15. Отправка данных на сервер каждые 5 секунд

В качестве примера приложение (Рис. 15) отправляет данные на сервер через каждые 5000мс.

Обратите внимание, что переменной newTrigger присваивается только значение «1» - после успешной отправки данных значение переменной-флага автоматически сбросится в «0».

Переменные, перечисленные в списке «Подписка», меняются автоматически, если от сервера пришла команда на их изменение.

В случае, когда контроллеру необходимо «знать» о изменении тех или иных переменных в прикладной программе, необходимо предусмотреть дополнительные функции контроля за счет специального системного блока SubscribeEvent – данный блок выдает информацию о идентификаторе записанной переменной и флаг активации.

Одни и те же переменные могут быть перечислены в обоих множествах (подписка и передача) – контроллер будет отправлять на сервер значение тех переменных, которые также могут быть изменены удаленно.



12. Модули обмена данными

Как видно из **Рис. 9** 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.

12.1 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 GREEN MOTION 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.

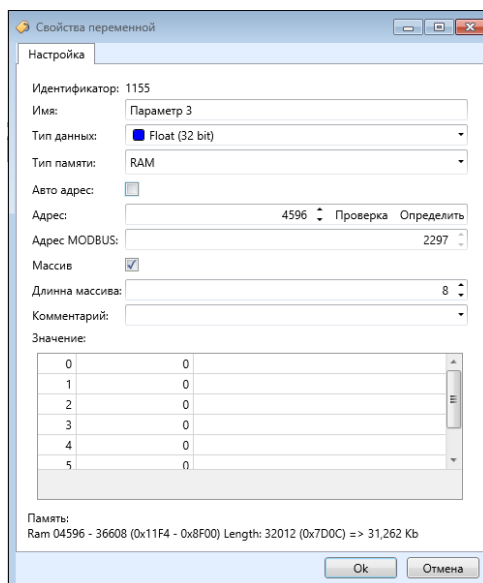


Рисунок 16. Редактирование переменных

Для точного определения любой переменной в Modbus пространстве необходимо использовать окно редактора переменных (**Рис. 16**).



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. 16**) этот адрес равен 4596 (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.

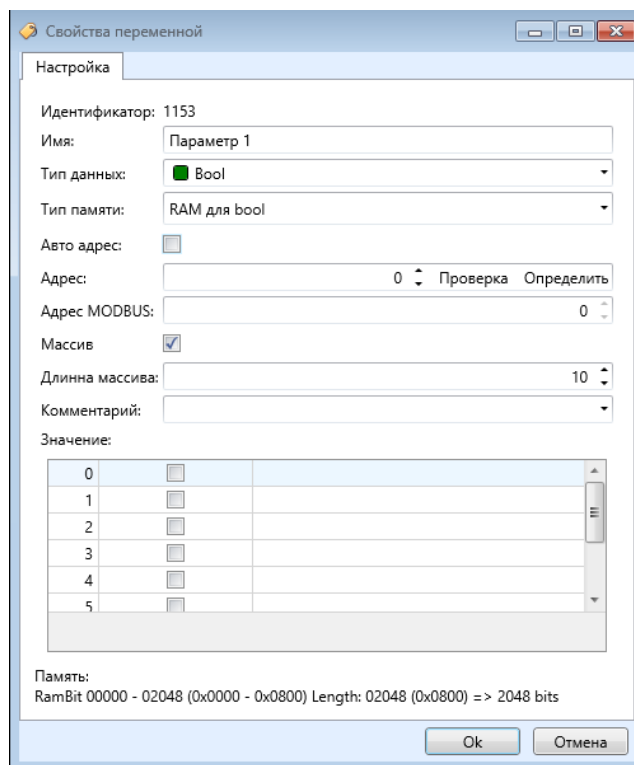


Рисунок 17. Пример с Bool

На **Рис. 17** 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 использовании 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.

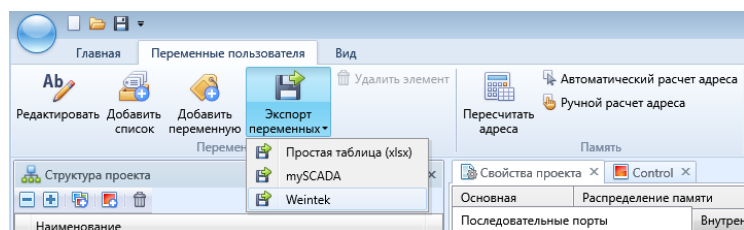


Рисунок 18. Меню экспорта списка переменных

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.

12.2 Modbus RTU

Работа Modbus RTU осуществляется через последовательные порты контроллера.

В **Таблице 7** указано обозначение портов.

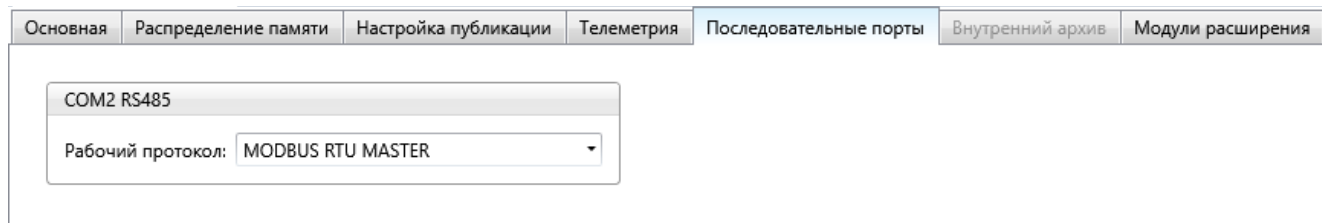


Рисунок 19. Базовая настройка последовательных портов

Для каждого порта допускается три рабочих режима, каждый из которых будет рассмотрен далее:

- Modbus RTU Master
- Modbus RTU Slave
- Пользовательский протокол

Таблица 7. Последовательные порты

Именование	Расположение	Обозначение в программе
COM1	Отсутствует	X
COM2	Изолированный порт RS-485, разъем RJ45	2
COM3	Отсутствует	X

В режиме Modbus RTU Slave последовательный порт предоставляет доступ сторонним устройствам к записи/чтению данных памяти контроллера.

Система адресации и порядок распределения памяти полностью аналогичен Modbus TCP Slave (раздел 12.1).

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. 19**).

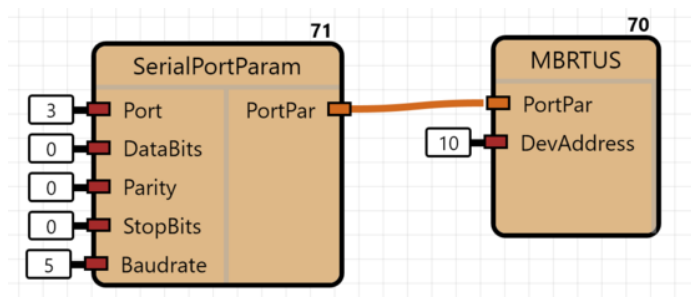


Рисунок 20. Настройка параметров Modbus RTU Slave

В блоке *SerialPortParamGREEN MOTION* указывается условный номер порта (Таблица 7), 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 GREEN MOTION автоматически сгенерирует и отправит ответ. Участия прикладной программы не требуется.

Участок кода, показанный на примере, Рис. 20, 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.

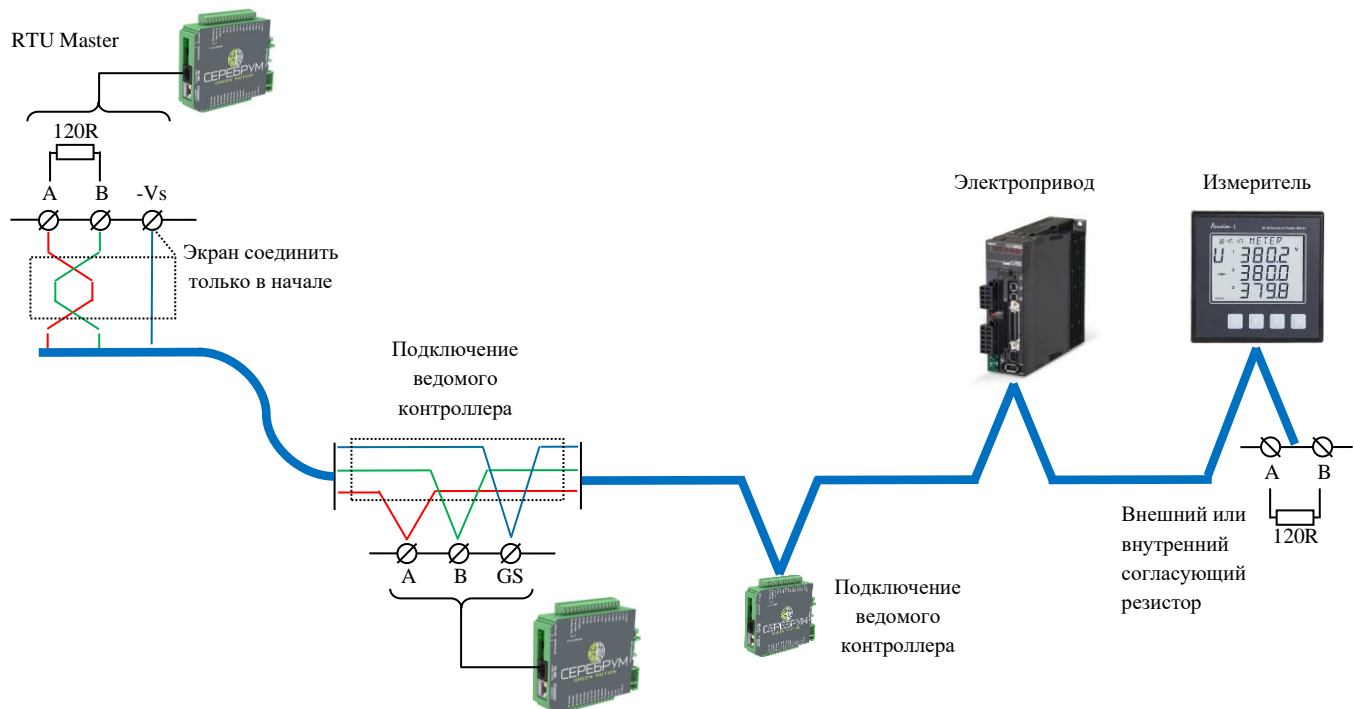


Рисунок 21. Подключение к шине Modbus RTU

На **Рисунке 21** показана приблизительная схема подключения устройств в шине RS-485 (Modbus RTU). В качестве ведущего устройства выступает контроллер GREEN MOTION.

Другой контроллер GREEN MOTION подключается к шине через порт RS-485.

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

12.2.1 Modbus RTU Master

Если COM порт сконфигурирован в проекте в качестве ведущего устройства, то автоматически инициализируется программа обмена по сети Modbus RTU.

Конфигурация данной программы осуществляется через прикладную программу пользователя.

На **Рисунке 22** приведен пример программы настройки обмена через порт Modbus RTU Master.

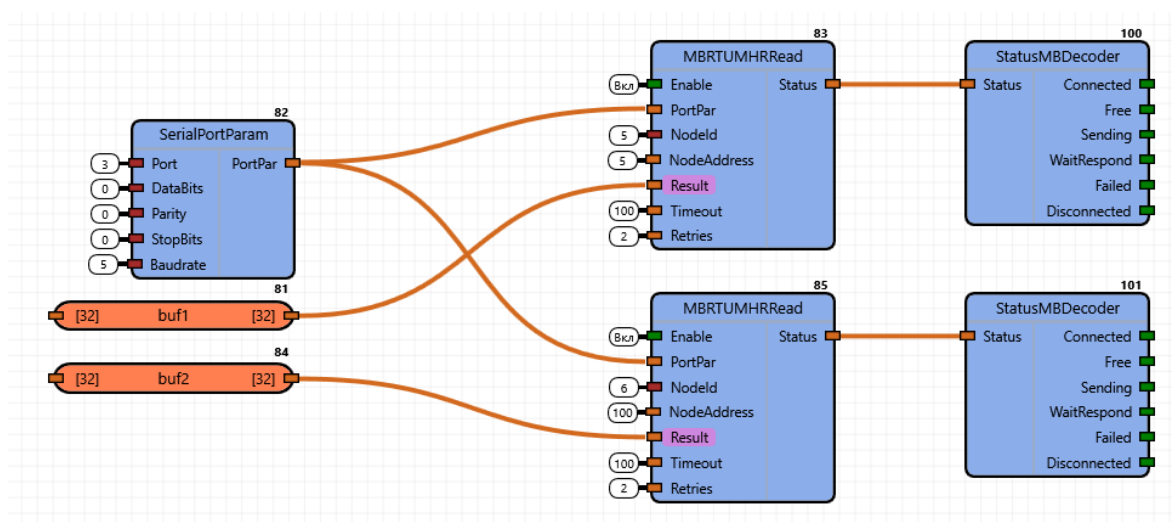


Рисунок 22. Конфигурация обмена Modbus RTU Master

Для каждого коммуникационного слота (команда для обмена) может быть определена своя уникальная настройка последовательного порта. В ряде случаев такой подход позволяет организовать несколько RTU сетей, работающих параллельно.

В примере настройки порта не изменяются и используется порт COM3, 19200 бод в формате 8N1 (блок 82 – *SerialPortParam*).

Команда 83 – *MBRTUMHRRRead* конфигурирует коммуникационный слот для чтения группы Holding регистров из устройства по сетевому адресу 5 (*NodeId*). Адрес первого регистра на удаленном устройстве тоже 5 (*NodeAddress*). Данные будут прочитаны в пользовательскую переменную buf1.

Обратите внимание, что длина запроса берется равной длине пользовательской переменной buf1, т. е. 32 слова.

Для данной команды определен таймаут отклика ведомого устройства (Timeout) 100 мс. Если отклик не будет получен, то программа Modbus RTU Master повторит запрос 2 (Retries) раза.

Статус выполнения команды расшифровывается в блоке *StatusMBDecoder*.

Аналогично настроена и команда 85, за исключением того, что обращение производится к устройству по адресу 6, начальный адрес регистра 100.

Все запросы делятся по используемым ими портам. Каждый порт обрабатывается индивидуально. Запросы, запрошенные в программе для порта, следуют поочередно (**Рис. 23**).

При необходимости запрос может быть пропущен, если вход Enable при начале обработки запроса равен «0».

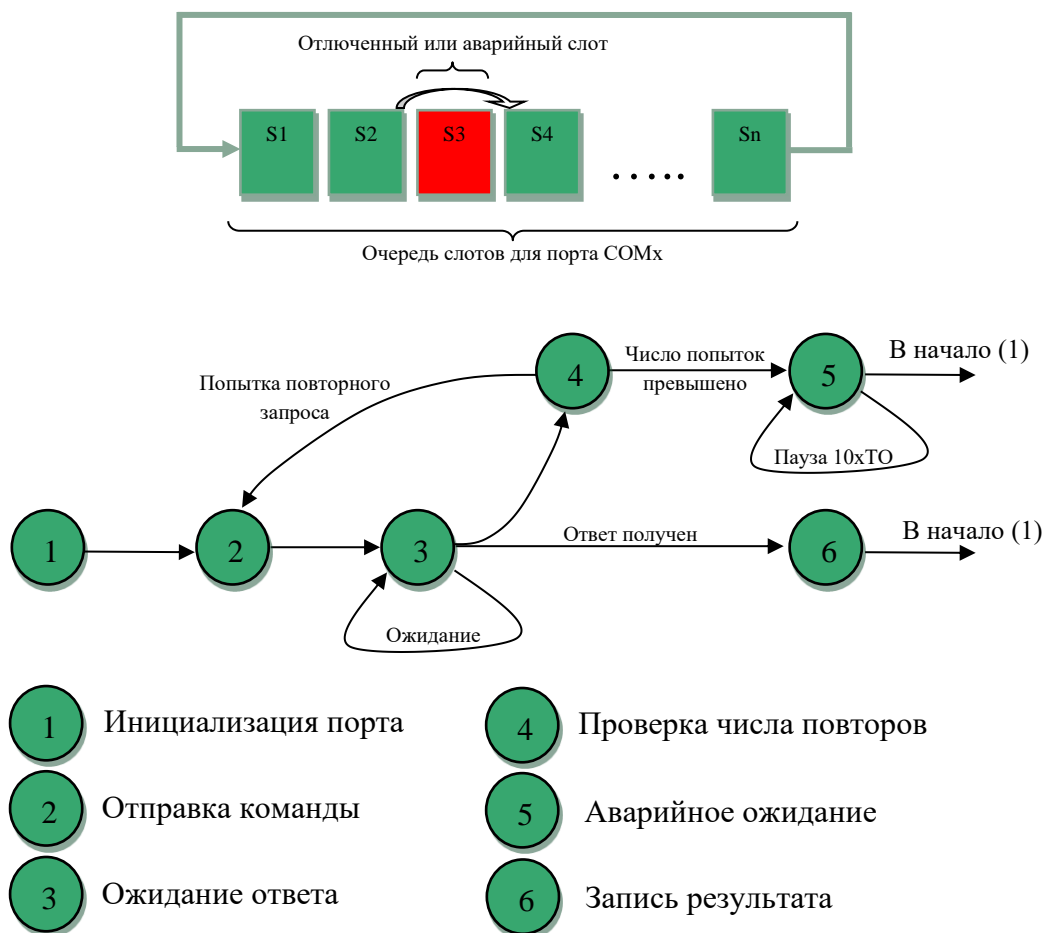


Рисунок 23. Работа слотов и диаграмма состояний коммуникационного слота.

Также слот пропускается в том случае, если все попытки получить отклик от удаленного устройства были неудачными (состояние 5). Пауза, в течение которой слот не будет задействован как правило равна десяти таймаутам для данной команды.

При успешном завершении процедуры обмена (состояние 6) программа перейдет к обслуживанию следующего по очереди слота.

Обработка очереди запросов прекращается если контроллер переходит в состояние Стоп. Кроме того, очередь сбрасывается при каждом новом запуске контроллера.

Запрос полностью удаляется из очереди в случае, когда Enable равняется False более чем пять секунд. Поскольку суммарная длина очереди запросов ограничена 32 – таким образом возможно организовать обмен с большей суммарной длиной очереди.



12.3 Пользовательские протоколы

В контроллере GREEN MOTION предусмотрен API для самостоятельного программирования протоколов обмена через последовательные порты.

При необходимости использования собственного протокола (или какого-либо из поставляемых в составе библиотеки функциональных блоков) необходимо в окне основного режима СОМ портов (**Рис. 19**) выбрать пункт «Пользовательский протокол».

API рассчитан на использование языка C-Yart и содержит следующие функции:

Попытка открытия порта с заданными аргументами.

port_open (port, baud, flags)

В процессе выполнения данной команды операционная система проверяет правильность конфигурации порта (режим «Пользовательский протокол»), доступность порта (порт не уже не открыт) и конфигурирует порт в соответствии с заданными в команде параметрами.

Значение параметра port должно быть равно коду порта (**Табл. 10**).

baud – значение скорости обмена: 9600, 19200, и т.д.

Поддерживаются следующие значения:

- 1200 бод
- 2400 бод
- 4800 бод
- 9600 бод
- 19200 бод
- 38400 бод
- 57600 бод
- 115200 бод

flags - битовое поле конфигурации

биты 31-3	бит 2	бит 1	бит 0
зарезервировано	STOPB	Parity	

STOPB: 1 – два стоповых бита; 0 – 1 стоповый бит

Parity: 0 – нет четности; 01 – Нечетный режим; 10 – Четный режим

В случае успешного открытия порта функция вернет значение 1.



В случае ошибки значения будут:

- -1 – порт уже открыт
- -2 – некорректный режим порта

Для закрытия порта используется функция ***port_close(port)***

port – аналогично ***port_open***

Все соединения сделанные ранее разрываются, параметры сбрасываются. Если порт находился в режиме приема или передачи данных, то работа будет остановлена.

port_check (port) – проверяет текущий режим порта. Функция возвращает «1», если порт открыт и «0», если порт закрыт.

port_send (port, address, len) – функция отправки сообщения через последовательный порт

port – индекс порта для отправки сообщения. Перед отправкой порт должен быть открыт

address – адрес в пространстве оперативной памяти (**Табл. 8**) где находится сообщение

len – длина сообщения в байтах

Функция возвращает:

- 0 - отправка началась
- -1 – ошибочный порт
- -5 – Отправка не удалась, передатчик занят
- -6 – Ошибка работы порта

port_receive (port, address, len)

port – индекс порта для получения сообщения. Перед приемом порт должен быть открыт

address – адрес в пространстве оперативной памяти (**Табл. 8**) где будет находиться сообщение

len – длина сообщения в байтах

Для информации о текущем статусе приема или передачи предусмотрены функции:

port_transmit_status (port) – проверка очереди передачи

port_receive_status (port) – проверка очереди приема

Функции возвращают число байт, оставшихся к приему или передаче.

0 – означает, что операция завершена

Отрицательное число обозначает ошибку.

stop_port_receive(port) – останавливает прием данных. ***port_receive_status()*** будет возвращать 0.



12.4 Yart Link

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

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. 24**.

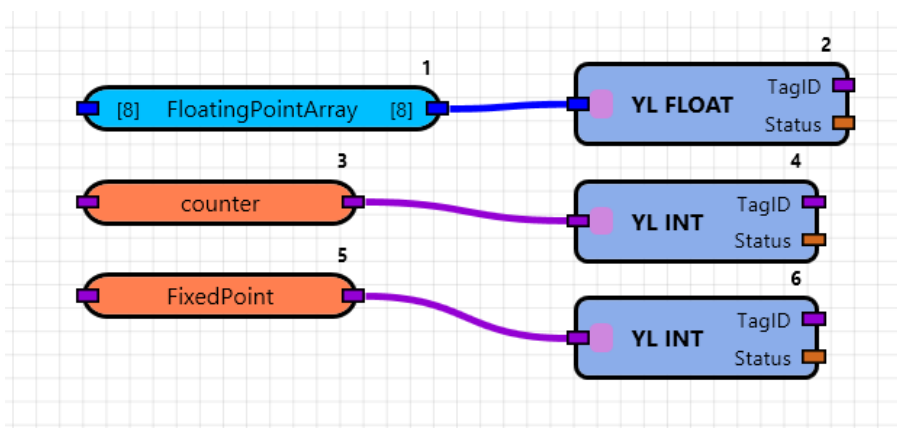


Рисунок 24. Пример программы передатчика YartLink

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. 25**) 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 predetermined 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.

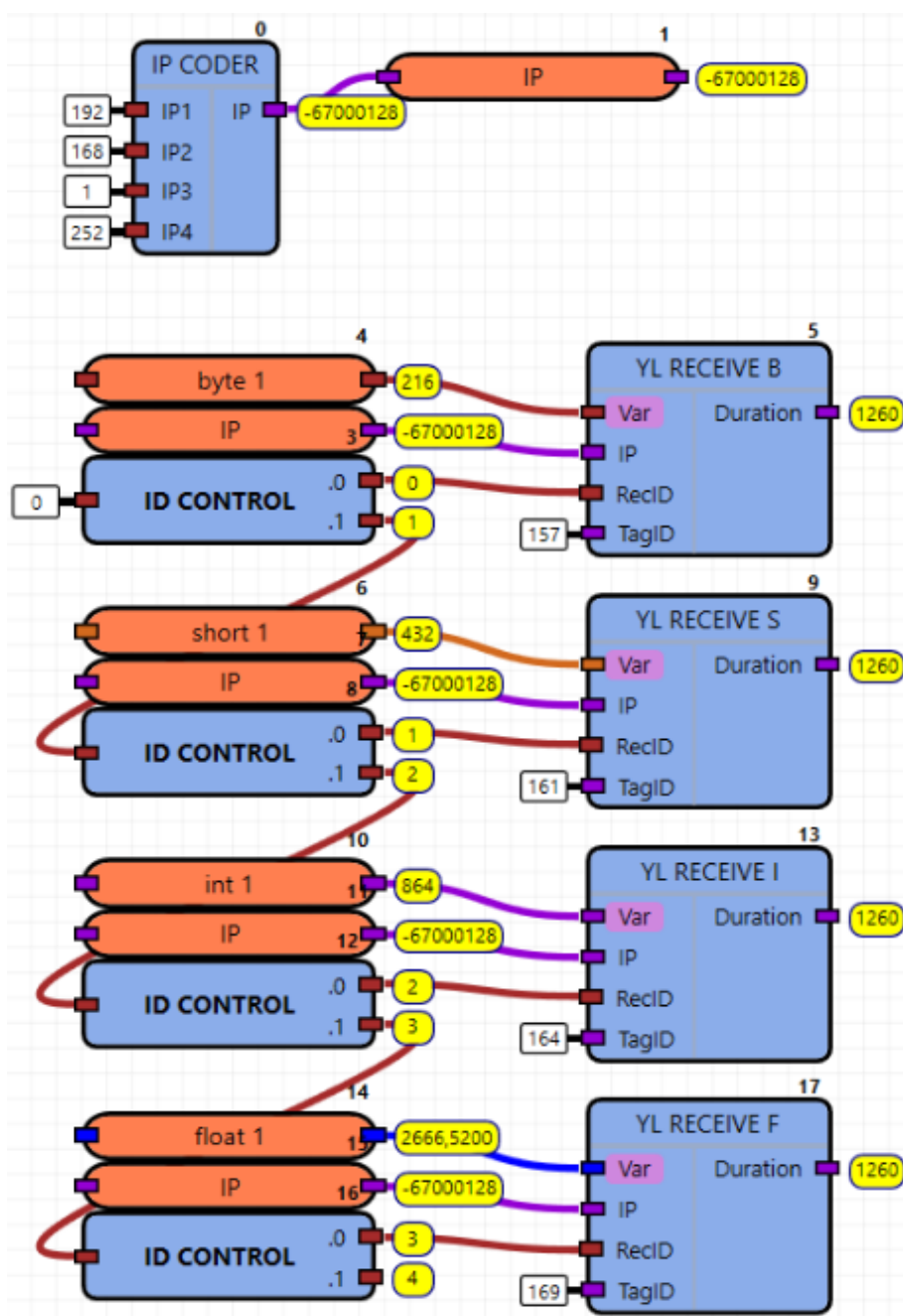


Рисунок 25. Подписка на глобальные публикуемые данные



13. Работа с периферийными модулями контроллера

13.1 Аналоговые входы

В GREEN MOTION доступны четыре аналоговых входа, в зависимости от конфигурации которых возможны следующие режимы измерения:

- Измерение входного напряжения
- Измерение входного тока

Измерение входного тока осуществляется при помощи подключаемых нагрузочных резисторов (поз. 2, Рис. 1).

На **Рисунке 26** показана схема подключения входных сигналов к аналоговым входам

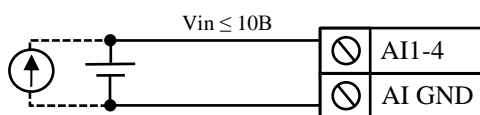


Рисунок 26. Подписка на глобальные публикуемые данные

Дальнейшая работа с аналоговыми входами выполняется в Yart Studio. Для каждого из используемых аналоговых входов необходимо выбрать требуемый режим работы, **Рис. 27**.

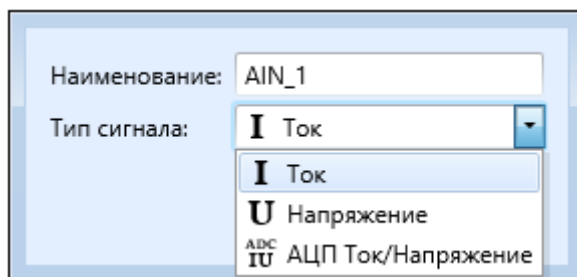


Рисунок 27. Установка режимов аналогового входа

Так, для примера выше для AI1 и AI2 необходимо установить режим «I Ток», AI3 и AI4 — «U Напряжение».

Выход каждого канала — число с плавающей точкой в формате измеряемой величины, **Рис. 28**.

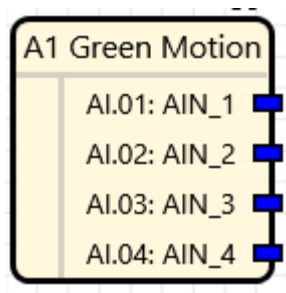


Рисунок 28. Установка режимов аналогового входа

Токовые каналы выдают значения в миллиамперах, напряжение выражается в Вольтах.

В ряде случаев удобно использовать необработанный код АЦП, для этого в списке режимов работы добавлен вариант «ADC IU АЦП Ток/Напряжение».

13.2 Дискретные входы

К контроллеру GREEN MOTION может подключаться до шести линий дискретных сигналов с номинальным напряжением «лог. 1» 24В. Каждый из входов может использоваться в режиме автоматического низкоскоростного счетчика импульсов (максимальная частота подсчета 100 Гц).

Входы DI1:DI2 также используются как входы для встроенного модуля обработки сигнала от инкрементального энкодера.

Подключение внешних сигналов к входам DIx должно выполняться по схеме, **Рис. 29**

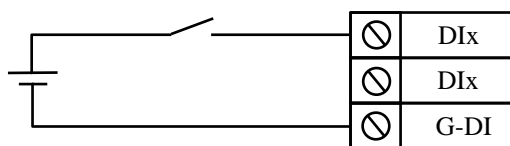


Рисунок 29. Схема подключения дискретных входов

Настройка режимов работы дискретных входов осуществляется при помощи Yart Studio (**Рис. 30**)

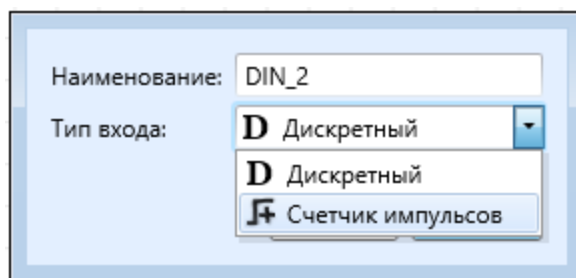


Рисунок 30. Дискретные входы контроллера

Режим «Дискретный» - обычный режим работы. Выход блока отражает текущее состояние дискретного входа.

В режиме «Счетчик импульсов» выходом блока будет являться число Int, равное числу детектированных переходов от лог. 0 к лог. 1. Для корректной работы входа в таком режиме частота на входе не должна превышать 100 Гц.

13.3 Подключение датчиков положения

К контроллеру Green Motion допускается одновременное подключение двух датчиков положения.

Вспомогательный датчик (вспомогательный в данном случае – это условное название) подключается к паре дискретных входов DI1:DI2. Для подключения основного датчика положения предусмотрен порт энкодера №1.

Выбор типа используемого энкодера осуществляется при помощи переключателей, расположенных на плате контроллера (Поз. 7, Рис. 1 и Рис. 31).

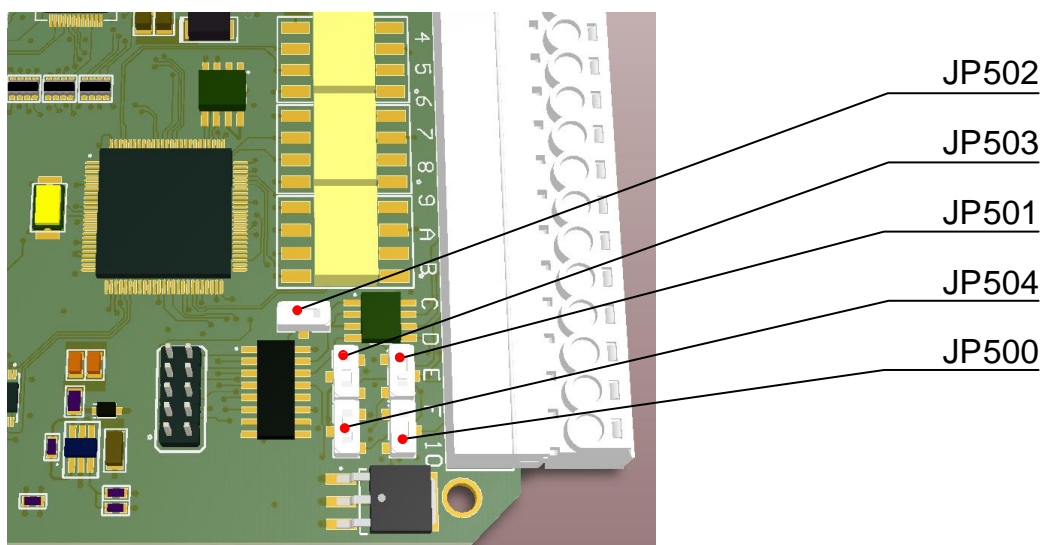


Рисунок 31. Переключатели режимов работы энкодера

В режиме инкрементального энкодера с HTL выходом линии А и В подключаются к клеммам ING1:ING3, Рисунок 32.

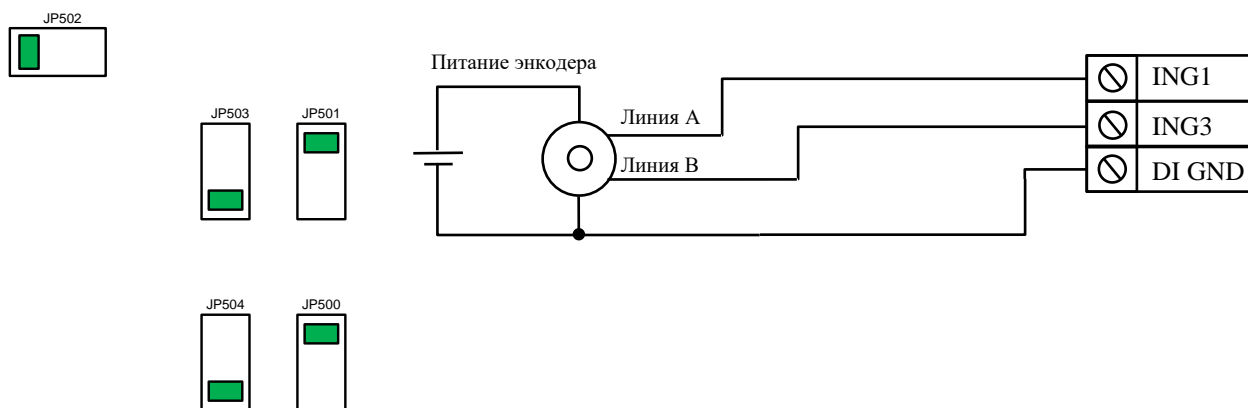


Рисунок 32. Подключение HTL энкодеров

Для подключения TTL и SinCos энкодеров схема и конфигурация порта энкодера показана на **Рис. 33**.

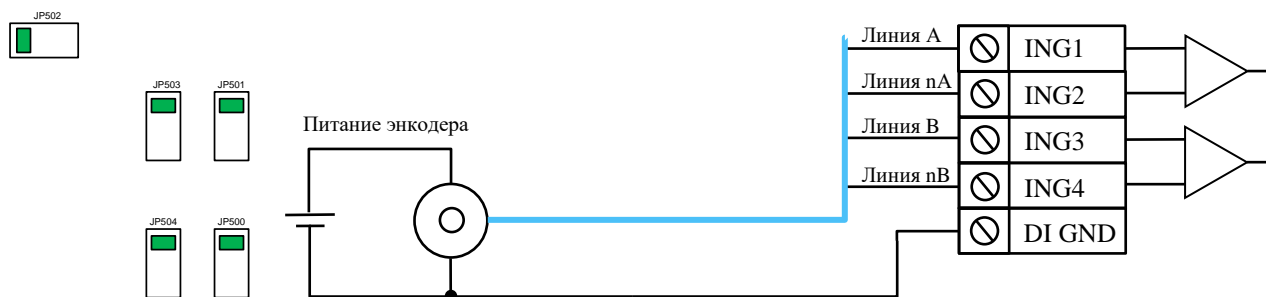


Рисунок 33. Подключение TTL и Sin/Cos энкодеров

Подключение абсолютных энкодеров показано на **Рисунках 34, 35**.

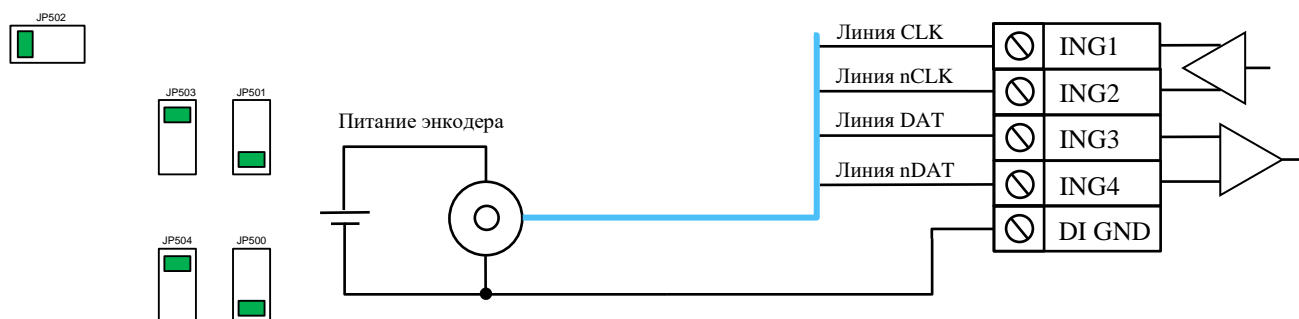


Рисунок 34. Подключение SSI энкодеров

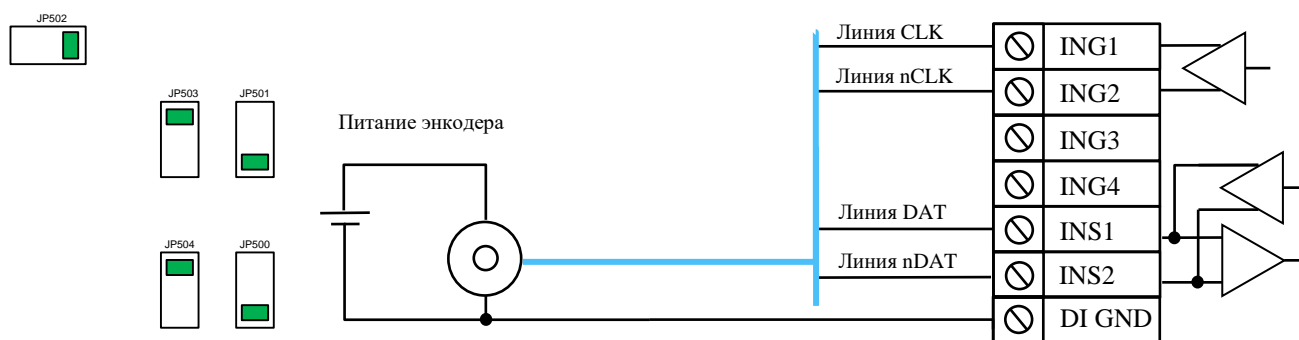


Рисунок 35. Подключение SSI энкодеров



13.4 Аналоговый выход

Единственный аналоговый выход, расположен на клеммнике 2 и подключается в соответствии с **Рис. 36**. Выход работает только в токовом режиме 0-20мА

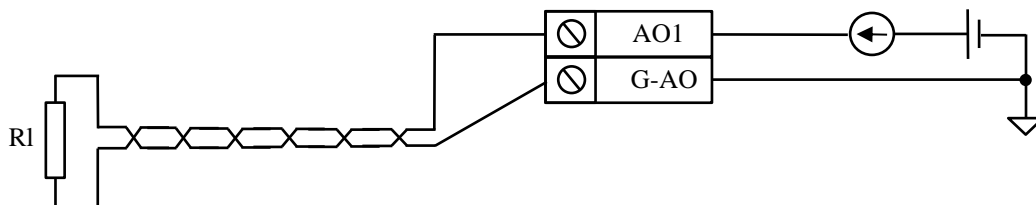


Рисунок 36. Подключение аналогового выхода

Токовый выход Green Motion работают от встроенного источника питания напряжением 12В. Таким образом максимальное сопротивление нагрузки не должно превышать 600 Ом.

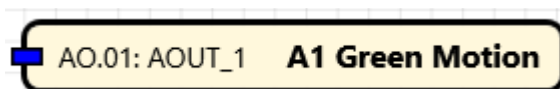


Рисунок 37. Использование аналогового выхода

Для использования аналоговых выходов достаточно перенести блок «Выход»-«Аналоговый» поля «Состав оборудования» в рабочую область программы (**Рис. 37**).

Величина выхода задается в формате с плавающей точкой и составляет 100.0 для максимального значения в 20 мА (10В при подключении встроенного нагрузочного резистора).



13.5 Дискретные выходы

В состав Green Motion входят четыре дискретных выхода на основе оптотранзисторов. Каждый оптотранзистор имеет индивидуальную изоляцию.

На **Рисунке 38** показана предполагаемая схема подключения нагрузки к выходам контроллера.

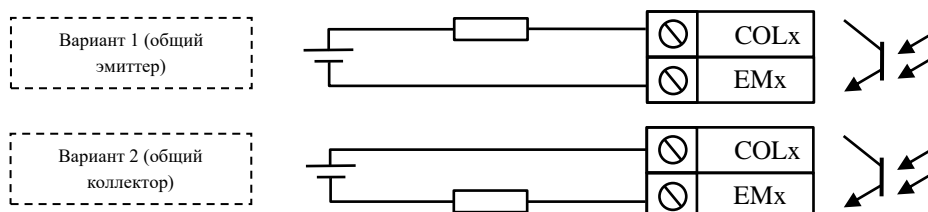


Рисунок 38. Подключение нагрузки к дискретным выходам контроллера

Каждый дискретный выход контроллера поддерживает два режима работы:

- дискретный выход
- выход ШИМ

Настройка режима осуществляется в Yart Studio в разделе «Состав оборудования» - «Выход» - «Дискретный», **Рисунок 39**.

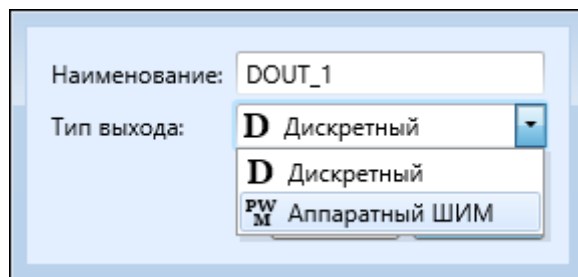


Рисунок 39. Конфигурация дискретного выхода

Режим работы «Дискретный» предполагает прямую трансляцию значения Bool на физический выход контроллера: False – выключено, True – включено.

В режиме «Аппаратный ШИМ» на вход функционального блока подается массив из двух Int. Для удобства формирования ШИМ в состав библиотеки включены блоки формирования массива управления (PWM, 2PWM, 3PWM).

На **Рисунке 40** показан пример использования ШИМ.

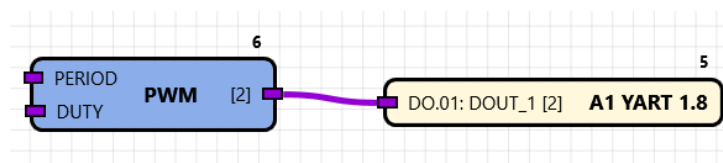


Рисунок 40. Формирование ШИМ на выходе ПЛК

Значение PERIOD соответствует периоду ШИМ и задается в миллисекундах. DUTY – величина длительности активного уровня сигнала (в мс).

В дополнение к основным четырем выходам пользователю доступен встроенный LED индикатор. Он управляется также как и обычный дискретный выход контроллера (Рис. 41).

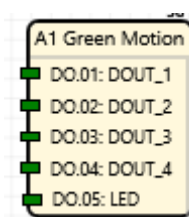


Рисунок 41. Дискретные выходы ПЛК



14. Работа с внешними 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. 7**).

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.

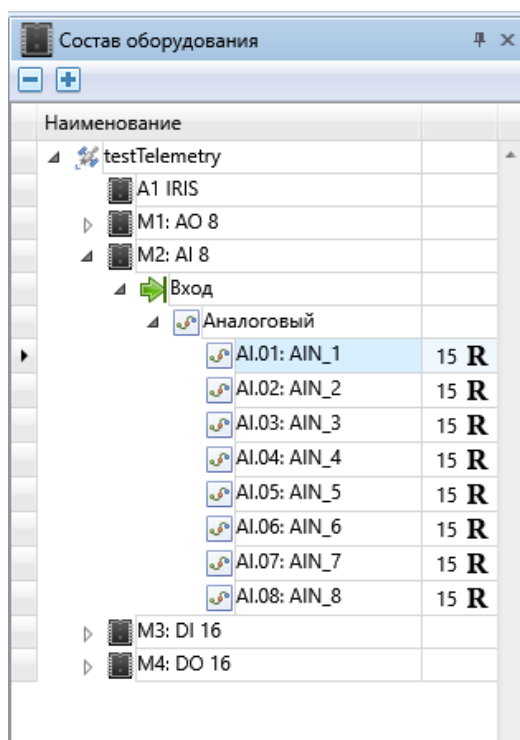


Рисунок 42. Панель настройки периферийных модулей

14.1 Аналоговые входы

Модуль AI8 (Рис. 43, 44) основан на 12-разрядном АЦП со встроенным программируемым усилителем.

Подключение модуля рекомендуется выполнять в соответствии со схемой (Рис. 44). При этом клеммы «Earth», 1, 16, 17, 32 должны подключаться к контуру заземления шкафа непосредственно в месте установки модуля.

Необходимо подключение всех клемм заземления.

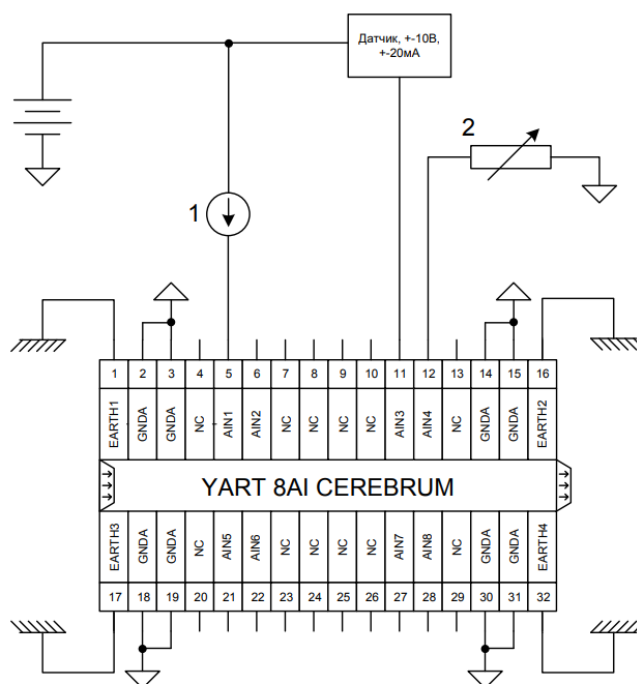


Рисунок 43. Модуль 8AI

На схеме (Рис. 43) 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.

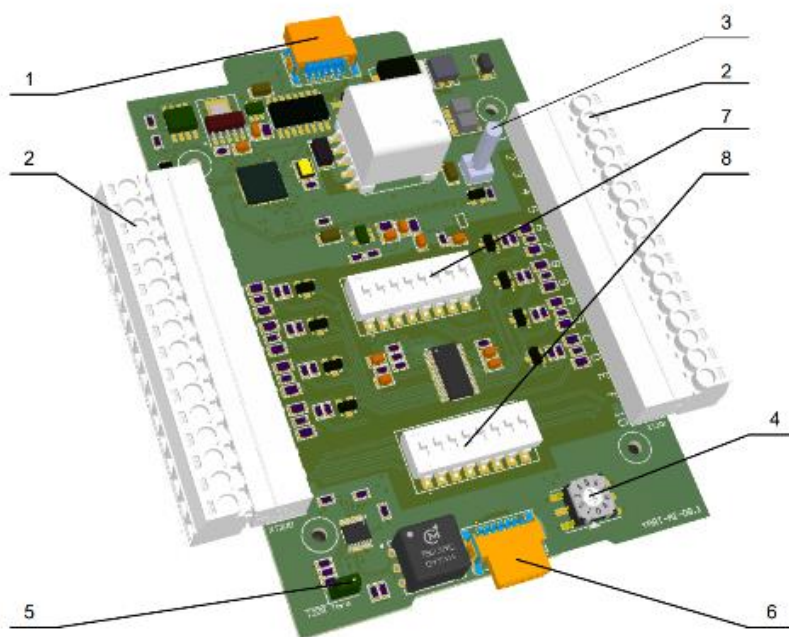


Рисунок 44. Модуль 8AI

- 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 – переключатель 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 8):

- измерение напряжения
- измерение тока
- измерение сопротивления

Таблица 8. 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 AI8 в прикладной программе (**Рисунок 35**) необходимо использовать индивидуальный функциональный блок A_{ix} или весь модуль сразу.

Добавление входа в программу осуществляется путем перетаскивания из выпадающего списка каналов (**Рис. 50**).

Пункт списка «Аналоговый» добавит все каналы одновременно.

Возможно добавления каждого канала по отдельности, выбирая элементы AI.xx:AIN_x.

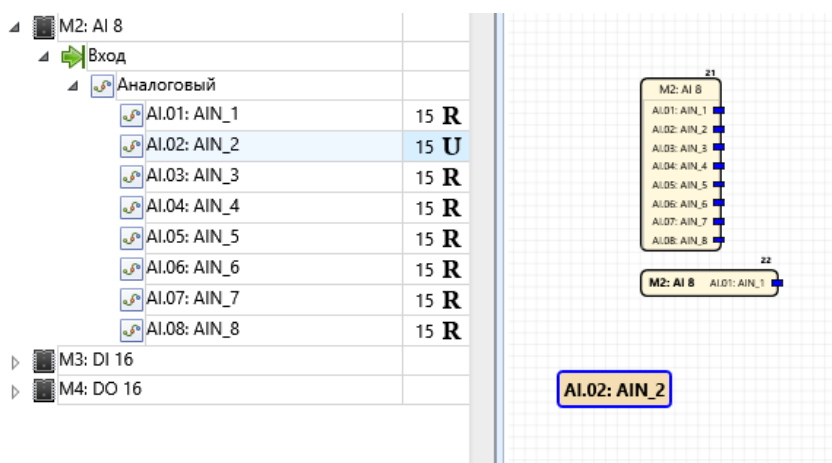


Рисунок 45. Функциональные блоки аналоговых входов

Выходы программных блоков модуля AI 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.

14.2 Аналоговые выходы

Модуль 8АО (**Рис. 46**) 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 46**). 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.

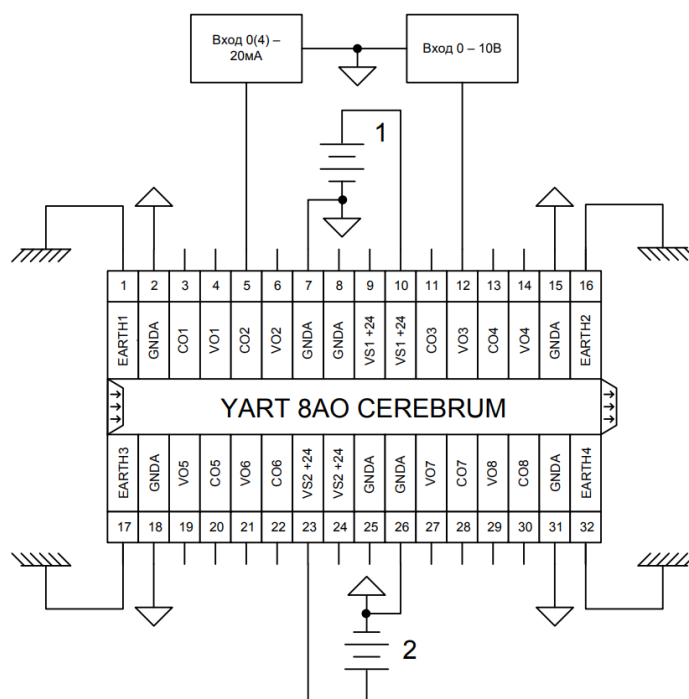


Рисунок 46. Модуль 8АО

Клеммы COx – токовые выходы 0 -20мА

Клеммы VOx – выходы напряжения 0 -10В

Для использования аналогового выхода в программе соответствующий блок необходимо разместить на программном поле, аналогично выходу ПЛК, **Рис. 37**.

На вход блока (**Рис. 47**) подается число в формате с плавающей точкой в диапазоне 0 -100.0

0.0 соответствует выходному току/напряжению 0мА/0.0В

100 соответствует выходному току/напряжению 20мА/10.0В

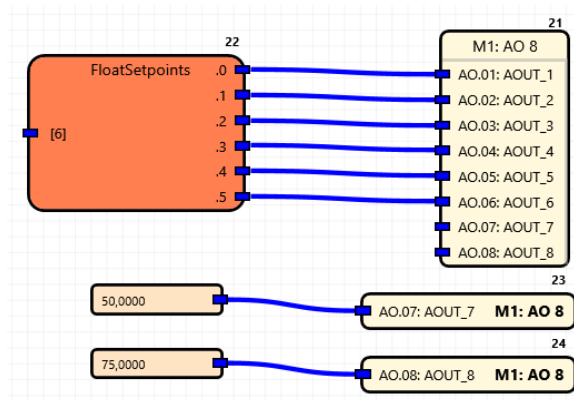


Рисунок 47. Использование модулей аналоговых выходов.

14.3 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.

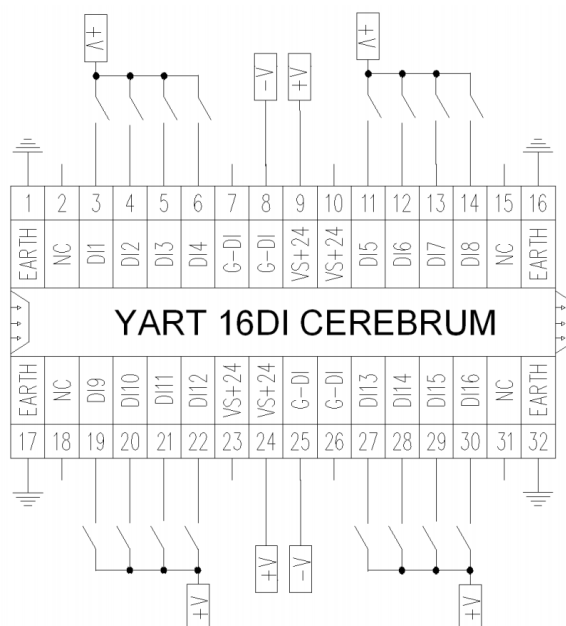


Рисунок 48. Модуль 16DI

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

Клеммы 1, 16, 17, 32 (**Рис. 48**) 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. 49**).

Блоки допускается размещать в любом месте программы.

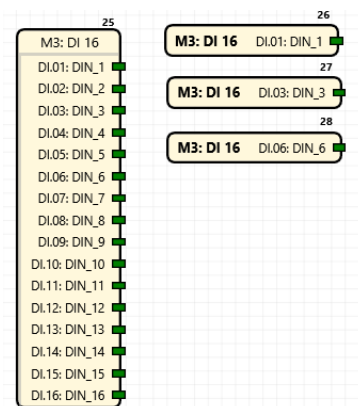


Рисунок 49. Блоки дискретного ввода

14.4 Дискретные выходы

Модуль 16DO содержит 16 выходных каналов с максимальной токовой нагрузкой 1А. Для каждого выходного блока из восьми линий (**Рис. 51**) 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. 51**) должны быть присоединены к внутреннему контуру заземления.

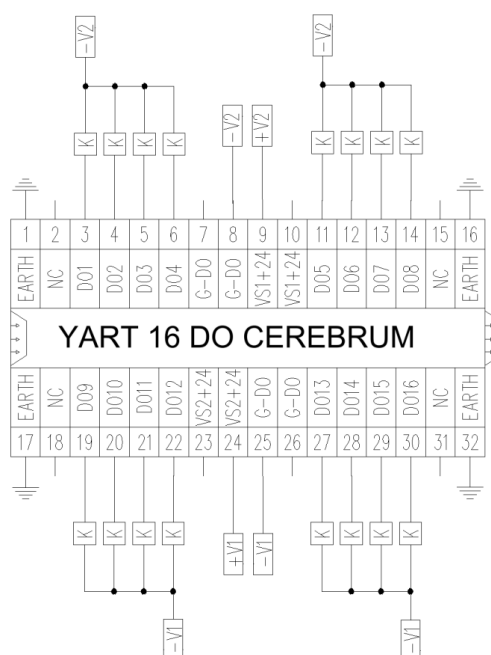


Рисунок 50. Модуль 16DO

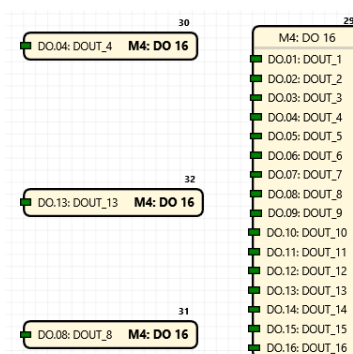


Рисунок 51. Функциональные блоки DO

Значения дискретных выходов задаются при помощи функциональных блоков DO (**Рис. 51**). 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.



15. 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

16. 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.