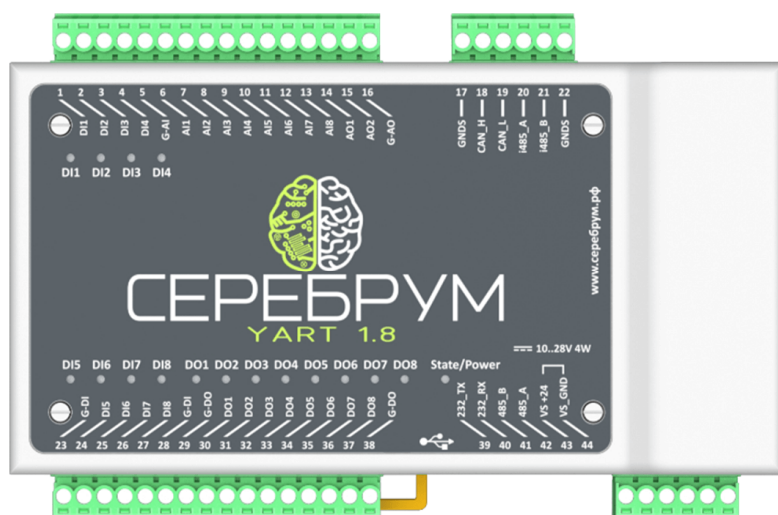


Программируемый логический контроллер

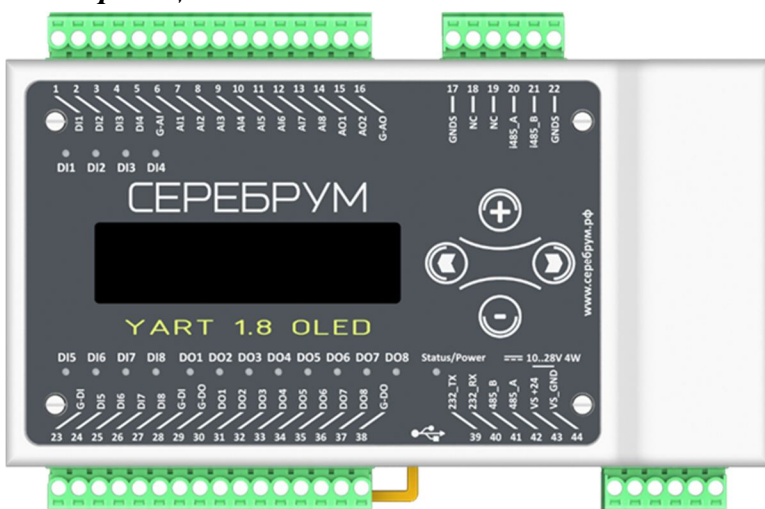
СЕРЕБРУМ YART 1.8

Техническое руководство



DI 8	DO ^{AC} 8 DC	AI ^U 8 I	AO ^U 2 I	32GB < 	
Ethernet 	USB 	RS232	RS485 2	ШИНА модулей I/O	Clock 
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

**модификация с OLED*



DI 8	DO ^{AC} 8 DC	AI ^U 8 I	AO ^U 2 I	32GB < microSD	OLED HMI
Ethernet	USB	RS232	RS485 2	ШИНА модулей I/O	Clock
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



Оглавление

2. Предупреждения	4
3. Список изменений.....	5
4. Общая информация.....	6
5. Технические характеристики	7
5.1 Массогабаритные характеристики.....	7
5.2 Требования к электропитанию.....	7
5.3 Условия эксплуатации.....	7
5.4 Конструкция и особенности устройства и работы контроллера	9
6. Подключение.....	12
7. Начало работы.....	15
7.1 Настройки сети	15
7.2 Работа с 3G модемом Neuro	16
8. Использование OLED пульта	17
8.1 Система меню пульта	17
8.1.1 Меню «Данные ПЛК»	17
8.1.2 Меню «Настройки»	18
8.1.3 Меню «Программа»	19
8.1.4 Меню «Телеметрия».....	19
8.1.5 Меню «Neuro»	20
8.1.6 Меню «Периферия»	21
8.2 Настройка переменных в YART Studio	21
9. Использование модулей расширения.....	23
10. Индикация состояний	26
11. ПЛК, особенности архитектуры и память	27
11.1 Память программируемого контроллера.....	28
11.1.1 Системная память.....	28
11.1.2 Оперативная память	28
11.1.3 Энергонезависимая память (FRAM).....	28
11.1.4 Батарейная память.....	30
11.2 Типы данных, используемых в системе.....	31
12. Телеметрия	32
12.1 Подключение к удаленному серверу телеметрии	33
13. Модули обмена данными.....	37



13.1 Modbus TCP	37
13.2 Modbus RTU.....	40
13.2.1 Modbus RTU Master.....	43
13.3 Пользовательские протоколы.....	45
13.4 Yart Link	47
13.5 USB порт Yart Studio	49
14. Работа с периферийными модулями контроллера.....	50
14.1 Аналоговые входы	50
14.2 Дискретные входы	53
14.3 Аналоговые выходы	56
14.4 Дискретные выходы	58
14.5 Настройка часов.....	60
15. Работа с внешними периферийными модулями	61
15.1 Аналоговые входы	62
15.2 Аналоговые выходы	66
15.3 Дискретные входы	68
15.4 Дискретные выходы	69
16. Архив данных.....	70
17. Транспортировка и хранение.....	72
18. Гарантийные обязательства.....	73



2. Предупреждения



3. Список изменений

4. Общая информация

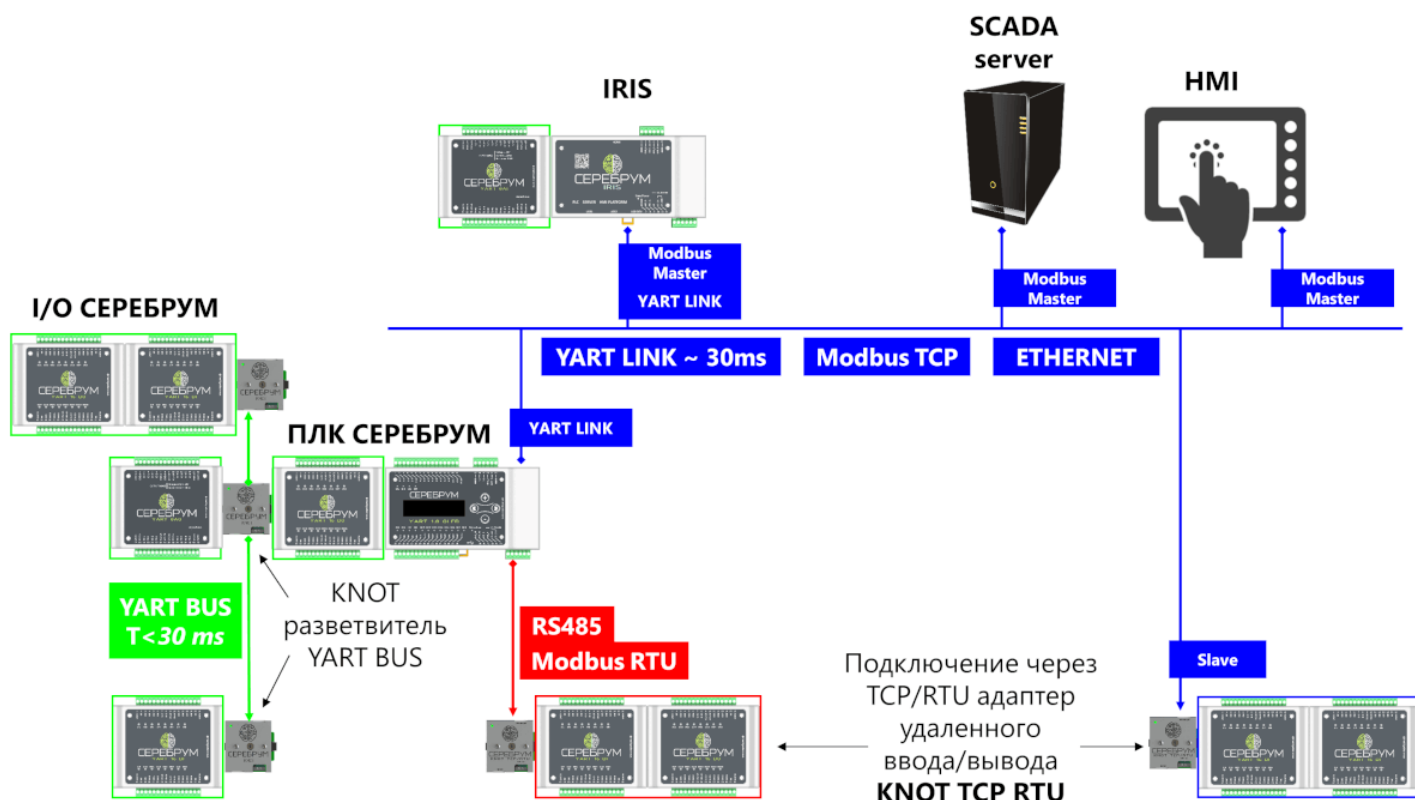
Контроллер YART 1.8 — промышленный логический контроллер (ПЛК) со встроенной системой удаленного мониторинга и управления СЕРЕБРУМ.

YART 1.8 может использоваться в качестве промышленного контроллера в локальных и распределенных системах управления с возможностью подключения удаленных моделей ввода/вывода через Ethernet или RS-485.

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

YART 1.8 поддерживает одновременную работу с серверами телеметрии СЕРЕБРУМ, Modbus TCP Slave, ModbusRTU Master/Slave, Yart-Link и до восьми подключенный модулей расширения через шину YART-BUS.

Физическое соединение с Internet осуществляется через Ethernet или при помощи внешнего модема СЕРЕБРУМ, подключаемого к порту RS-232





5. Технические характеристики

5.1 Массогабаритные характеристики

– габаритные размеры, мм, не более:

а) корпуса контроллера 145 x 78 x 28;

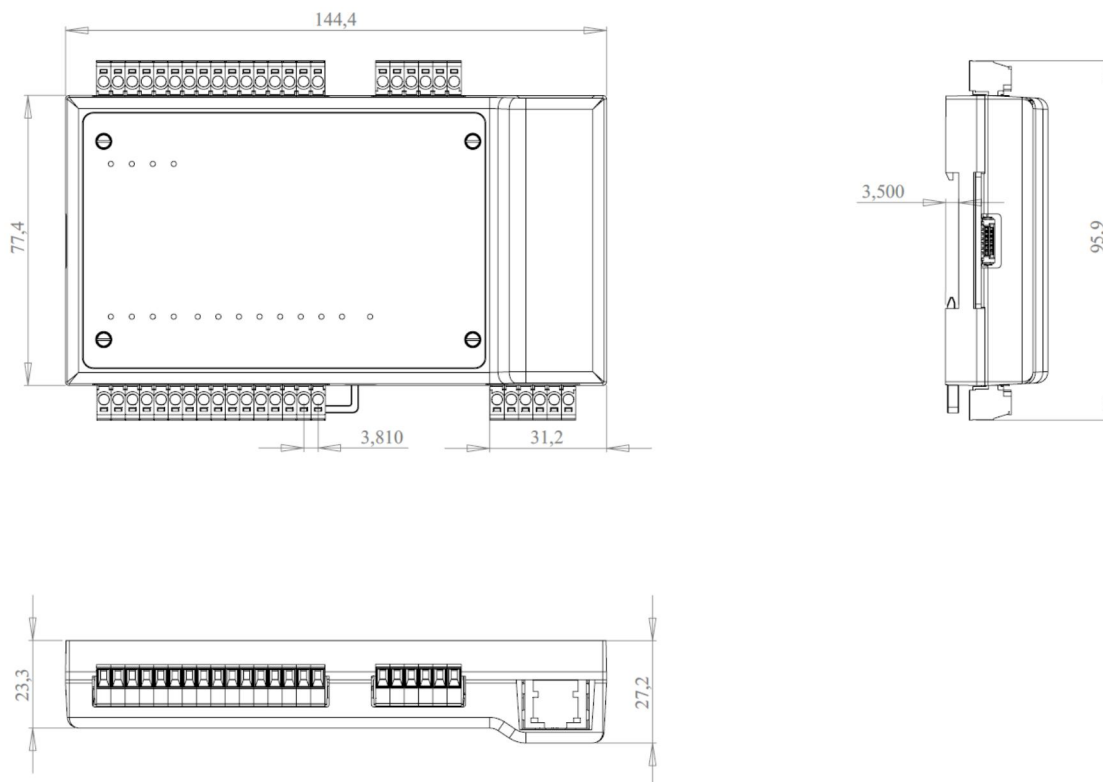
б) с учетом клеммных соединений 145 x 98 x 28.

– масса, кг, не более

0,2;

– монтаж

на DIN-рейку по стандарту DIN EN 50 022.



5.2 Требования к электропитанию

Электропитание контроллера осуществляется от сети постоянного тока напряжением от 10 до 28 В (номинальное – 24 В).

Потребляемая мощность – не более 4 Вт.

5.3 Условия эксплуатации

Контроллер предназначен для эксплуатации в следующих условиях:

– закрытые взрывобезопасные помещения или шкафы электрооборудования без агрессивных паров и газов;

– рабочая среда

воздух;

– диапазон рабочих температур окружающей среды от минус 40 до 55 °С;

– верхний предел относительной влажности

80% при 35 °С и более

низких температурах без

конденсации влаги;



- атмосферное давление от 84 до 106,7 кПа;
- высота над уровнем моря не более 2000 м;
- степень защиты корпуса по ГОСТ 14254-96 IP20;
- вибрация амплитуда не более 0,1 мм
с частотой не более 25 Гц;

допустимая степень загрязнения 1 по ГОСТ Р 51841-2001 (несущественные загрязнения или наличие только сухих непроводящих загрязнений).

5.4 Конструкция и особенности устройства и работы контроллера

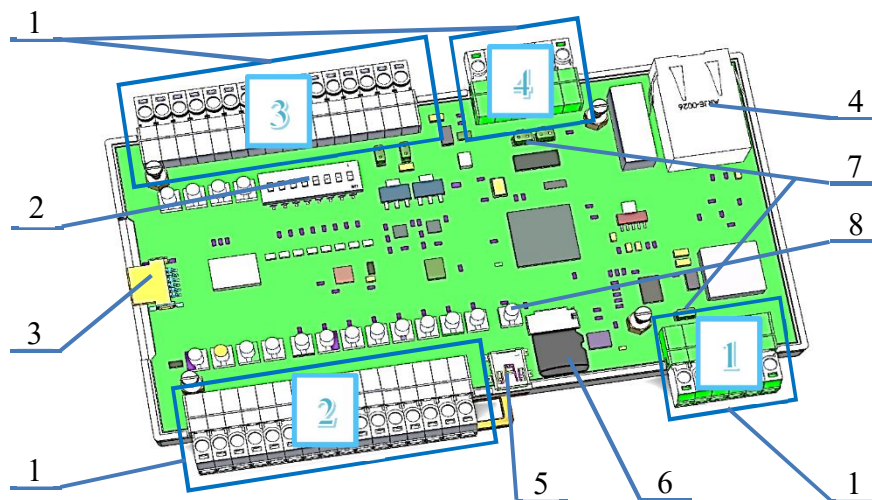


Рисунок 1. Плата контроллера YART-1.8

1. Клеммники
2. Переключатель токового режима аналоговых входов
3. Разъем шины YART-BUS
4. Разъем RJ-45, Ethernet
5. Разъем mini USB для подключения к ПК
6. Слот uSD карты
7. Переключатели «терминаторов» последовательных интерфейсов
8. Индикатор режима работы контроллера



Таблица 1. Технические характеристики YART-1.8

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	640 KB энергонезависимая память (более 500 типов блоков, 5000 вызовов блоков)
	RAM	40 KB память для переменных пользователя
	FRAM	16 KB энергонезависимая память для переменных пользователя
	BRAM	4 KB, энергонезависимая (батарейная) память для переменных пользователя
	RAM BIT	2 KBIT память для переменных пользователя
	FRAM BIT	2 KBIT энергонезависимая память для переменных пользователя
uSD карта памяти		до 32 GB для хранения архива
Выполнения цикла программы		от 1 ms
Входы	Дискретные	8DI x 7-28 VDC, LED индикация, гальванически изолированы от цепей процессора
	Аналоговые	8AI x 16 бит. Режимы - U/I/R/DI (0-10VDC/0-20mA/0-400kOm).
Выходы	Дискретные	8DO x твердотельные реле, VDC/VAC, до 0,5 А кратковременно, 0,1 А



		длительная нагрузка, LED индикация. Гальванически изолированы от цепей процессора
	Аналоговые	2АО x U/I (0-10VDC/0-20mA)
Коммуникационные порты	ETHERNET	Разъем RJ-45, 100/10Base-T автоопределение типа кабеля. YART-LINK, MODBUS TCP, IoT, статус WEB SERVER
	mini-USB	MODBUS RTU, порт отладки
	RS232	MODBUS RTU, протоколы библиотеки, пользовательский
	RS485	MODBUS RTU, протоколы библиотеки, пользовательский
	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 – 8
Часы реального времени (RTC)		С питанием от батарейки (точность хода при 25 °C – не более ±2 с в сутки)

Таблица 2. Технические характеристики OLED:

Параметр	Характеристики
Разрешение экрана:	128x32
Размер видимой области:	57x15 мм
Отображаемые параметры:	- встроенное меню настройки и диагностики ПЛК - до 29 отображаемых пользовательских переменных
Тип дисплея:	OLED
Поддерживаемые типы данных ПЛК:	bool, byte, short, int, float, ip, time, date
Управление от клавиатуры	4 кнопки
Оценочная наработка на отказ:	35000 часов
Температурный диапазон эксплуатации:	-40 +55 °C

6. Подключение

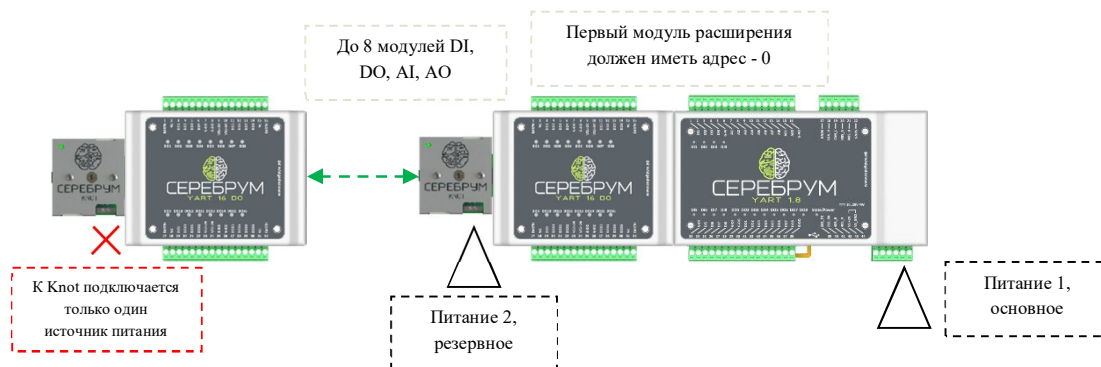


Рисунок 2. Подключение источников питания

Для работы контроллера требуется наличие внешнего питания (см. **Таблицу №1**). Линия питания выведена на клеммы 11-12 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 YART-1.8 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 3. Сигналы клеммника 1

Номер клеммы	Наименование	Назначение
39	232-TX	Линия RS-232, передача
40	232-RX	Линия RS-232, прием
41	485-B	Линия RS-485 (COM3, неизолированный), В
42	485-A	Линия RS-485 (COM3, неизолированный), А
43	VS+24	Питание контроллера, положительный потенциал
44	VS GND	Питание контроллера, общий

Таблица 4. Сигналы клеммника 2

Номер клеммы	Наименование	Назначение
23	G-DI	Общий сигнал дискретных входов
24	DI5	Дискретный вход
25	DI6	Дискретный вход
26	DI7	Дискретный вход
27	DI8	Дискретный вход
28	G-DI	Общий сигнал дискретных входов
29	G-DO	Общий сигнал дискретных выходов
30	DO1	Дискретный выход (транзисторный)
31	DO2	Дискретный выход (транзисторный)
32	DO3	Дискретный выход (транзисторный)
33	DO4	Дискретный выход (транзисторный)
34	DO5	Дискретный выход (транзисторный)
35	DO6	Дискретный выход (транзисторный)
36	DO7	Дискретный выход (транзисторный)
37	DO8	Дискретный выход (транзисторный)
38	G-DO	Общий сигнал дискретных выходов

Таблица 3. Сигналы клеммника 3

Номер клеммы	Наименование	Назначение
1	DI1	Дискретный вход
2	DI2	Дискретный вход
3	DI3	Дискретный вход
4	DI4	Дискретный вход
5	G-AI	Общий сигнал аналоговых входов
6	AI1	Аналоговый вход 1 (U, I R)
7	AI2	Аналоговый вход 2 (U, I R)
8	AI3	Аналоговый вход 3 (U, I R)
9	AI4	Аналоговый вход 4 (U, I R)
10	AI5	Аналоговый вход 5 (U, I R)
11	AI6	Аналоговый вход 6 (U, I R)
12	AI7	Аналоговый вход 7 (U, I R)
13	AI8	Аналоговый вход 8 (U, I R)
14	AO1	Аналоговый выход 1 (I, U)
15	AO2	Аналоговый выход 2 (I, U)
16	G-AO	Общий сигнал аналоговых выходов



Таблица 3. Сигналы клеммника 4

Номер клеммы	Наименование	Назначение
17	GNDS	Общий сигнал последовательных портов
18	CAN-H	*Линия CAN-H (зарезервировано)
19	CAN-L	*Линия CAN-L (зарезервировано)
20	i485-A	Линия RS-485 (COM2, изолированный), А
21	I485-B	Линия RS-485 (COM2, изолированный), В
22	GNDS	Общий сигнал последовательных портов

*-опционально



7. Начало работы

Обмен данными с YART-1.8 в основном осуществляется через интерфейс Ethernet. Для комфортной работы с контроллером уделите внимание корректным настройкам сетевого оборудования и сетевой инфраструктуры в целом.

Подключение к Yart Studio также может быть выполнено через порт USB.

Порт USB гальванически связан с линией питания контроллера. Перед подключением убедитесь, что питание ПЛК имеет гальваническую развязку от основной сети питания установки.

7.1 Настройки сети

Настройка сетевого интерфейса осуществляется помощи Yart Studio.

По умолчанию YART-1.8 получает статический IP адрес равный **192.168.1.254**

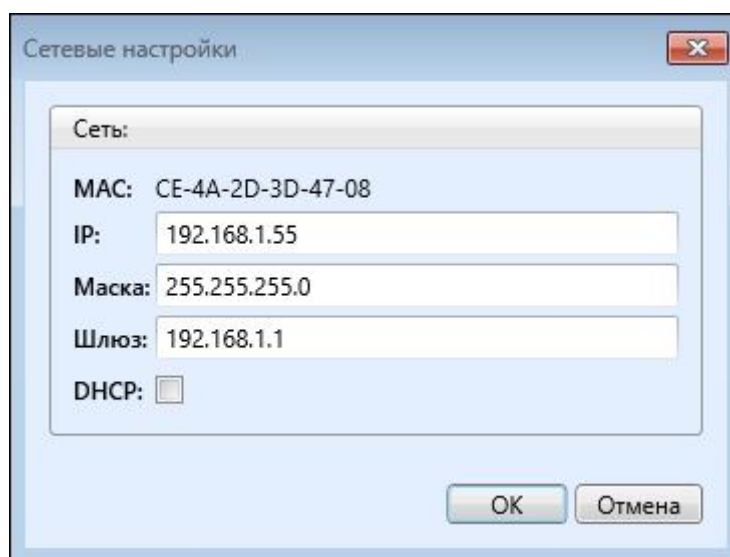


Рисунок 3. Настройка сетевого подключения

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



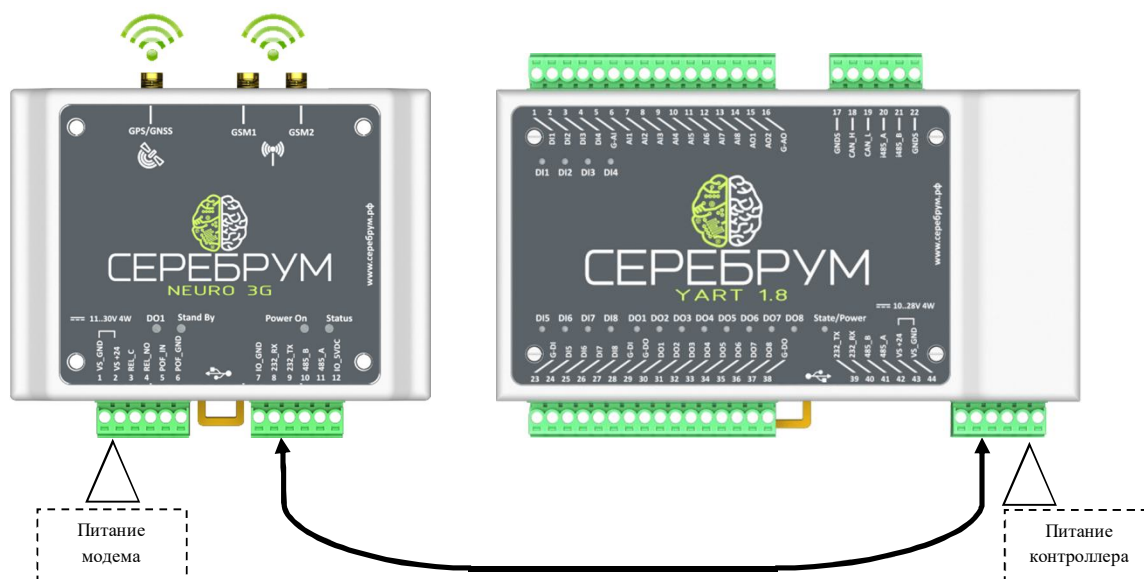
DHCP – поле выбора режима автоматического назначения сетевого адреса. В случае, если в сети не будет найден сервер DHCP адрес контроллера будет равен значению из поля «IP»

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.

7.2 Работа с 3G модемом Neuro

YART-1.8 подключается к 3G модему Neuro только через интерфейс RS-232 (**Рисунок 4**), клеммы 8, 9, 12, клеммник №1.



# Цепь	Клемма Neuro-3G	Клемма YART-1.8
1	7 – IO_GND	44 – VS_GND
2	8 – 232_RX	39 – 232_TX
3	9 – 232_TX	40 – 232_RX

Рисунок 4. Подключение модема Neuro

К одному контроллеру подключается не более одного модема.

При необходимости сброса питания модема подключите линии POF_IN:POF_GND модема к одному из дискретных выходов контроллера или модуля расширения. При наличии активного уровня на входе POF_IN Neuro-3G переходит в режим остановки со сбросом питания GSM-GPRS-3G модуля.

При работе с Neuro соединение с Интернет будет поддерживаться автоматически.



8. Использование OLED пульта

На контроллер YART 1.8 возможна установка пульта с OLED дисплеем и клавиатурой. Пульт дает возможность менять и просматривать системные настройки контроллера и переменные алгоритма пользователя. Всего в пульте может быть не больше 29 переменных пользователя.

Обновление прошивки пульта производится автоматически при обновлении прошивки контроллера.

8.1 Система меню пульта

Корневое меню пульта включает в себя 6 пунктов, перемещение между которыми производится клавишами вверх и вниз. Для выбора пункта меню используется клавиша вправо:



8.1.1 Меню «Данные ПЛК»

В меню содержится данные:

- SN1 и SN2 серийный номер контроллера разбитый на 2 строчки
- Тип ПЛК
- Версия ОС – версия прошивки контроллера
- Версия загрузчика контроллера
- Объем Flash памяти контроллера



8.1.2 Меню «Настройки»

В меню содержатся системные настройки контроллера:

- IP адрес
- Маска сети
- IP адрес шлюза
- Управление режимом DHCP
- Время контроллера

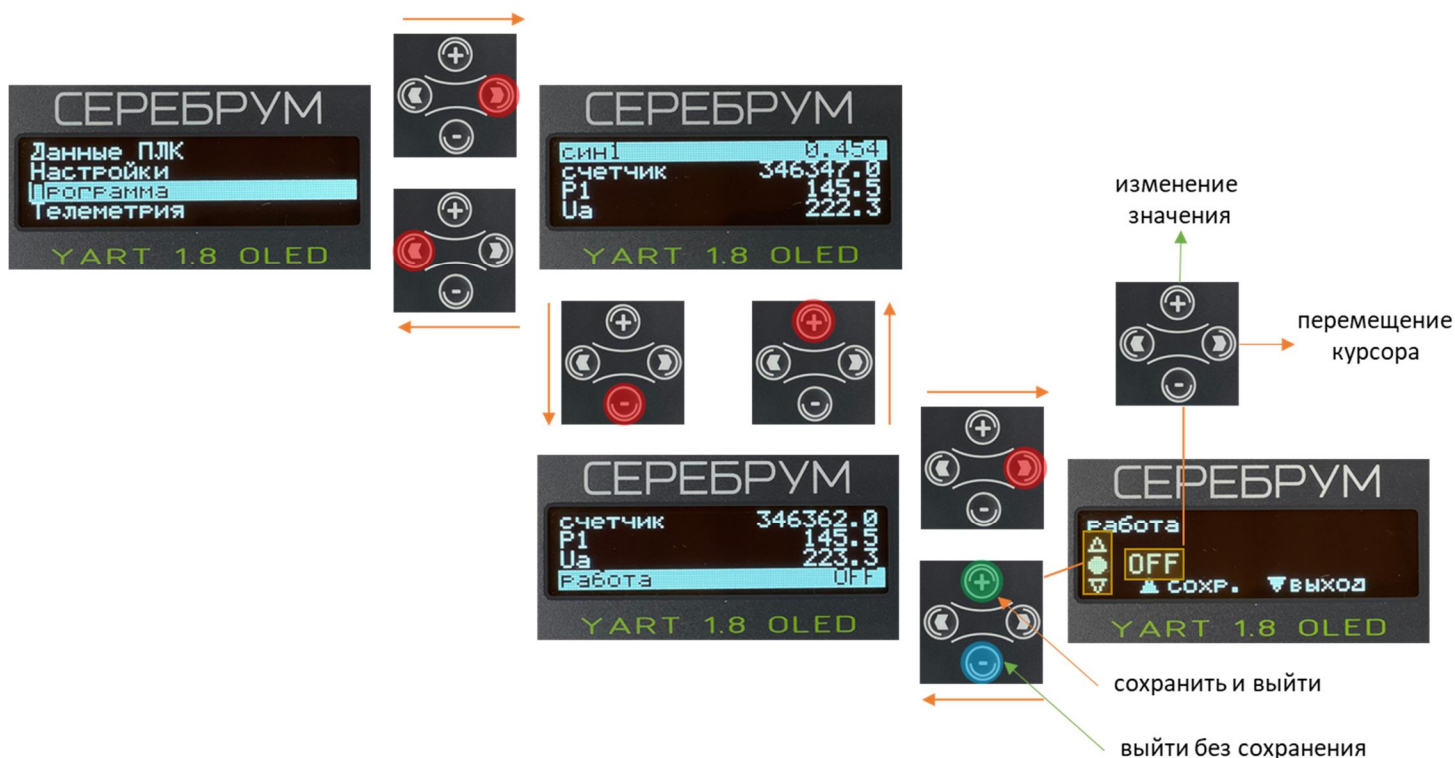
Все данные доступны для редактирования.



8.1.3 Меню «Программа»

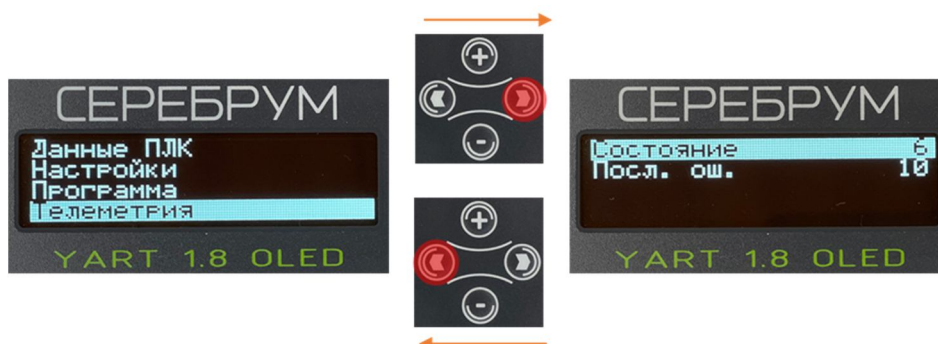
В меню содержатся переменные, заданные пользователем в YART Studio. Если переменные заданы, при старте контроллера производится индикация этого меню.

Пример меню заданного в YART Studio:



8.1.4 Меню «Телеметрия»

В меню отображаются данные о состоянии телеметрии контроллера:



Значения параметра «Состояние»:

- 0 - Процесс инициализации внутренних дескрипторов;
- 1 - Процесс пинга сервера завершился неудачно;
- 2 - Процесс подключения к порту сервера завершился неудачно;
- 3 - Процесс подписки завершился неудачно;
- 4 - Процесс передачи описания или данных завершился неудачно;
- 5 - Процесс завершился неизвестной ошибкой;
- 6 - Процесс завершился успешно;
- 7 - Процесс формирования дескрипторов завершился неудачно;



- 8 - Процесс соединение с сервером;
- 9 - Процесс подписки;
- 10 - Процесс аутентификации;
- 11 - Процесс отправки описания.

Значения «Посл. ош.» (последняя ошибка):

- 0 - Ожидание инициализации системных дескрипторов;
- 1 - Ошибка указателя на дескриптор;
- 2 - Ошибка содержимого дескриптора;
- 3 - Неизвестный режим телеметрии;
- 4 - Нет пинга от сервера;
- 5 - Сервер не подтвердил авторизацию;
- 6 - Ошибка подписки на сервере;
- 7 - Описание передано на сервер, но сервер отказал в регистрации;
- 8 - Соединение разорвано, причина не установлена;
- 9 - Соединение принудительно разорвано;
- 10 - Ошибка публикации на сервере.

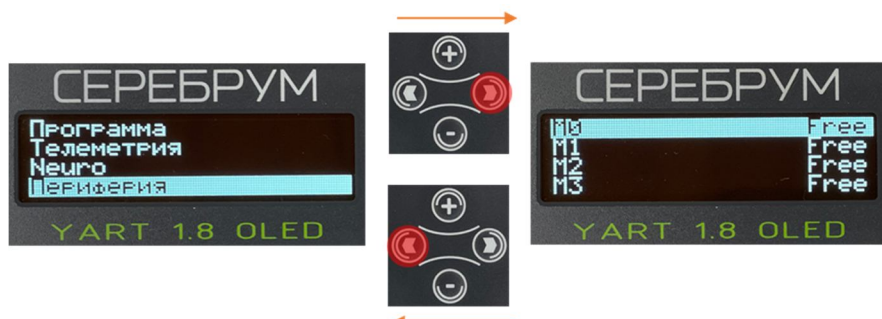
8.1.5 Меню «Neuro»

Если к контроллеру подключен модем NEURO 3G (GSM, GPS/GLONASS), в меню будут отображаться параметры связи. Контроллер должен быть подключен через модем к серверу телеметрии.



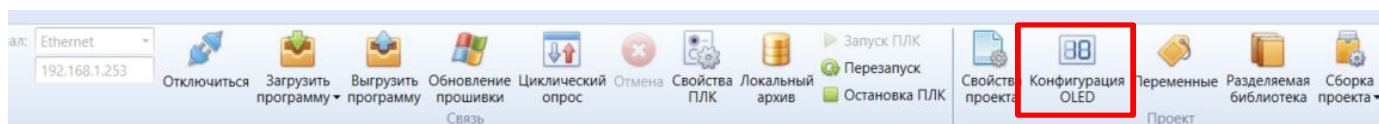
8.1.6 Меню «Периферия»

В меню содержится информация о подключенных модулях расширения. Если при перезапуске контроллера были подключены модули расширения, то напротив их адреса будет индикация типа, например «DO16». Напротив свободных адресов будет надпись «free».

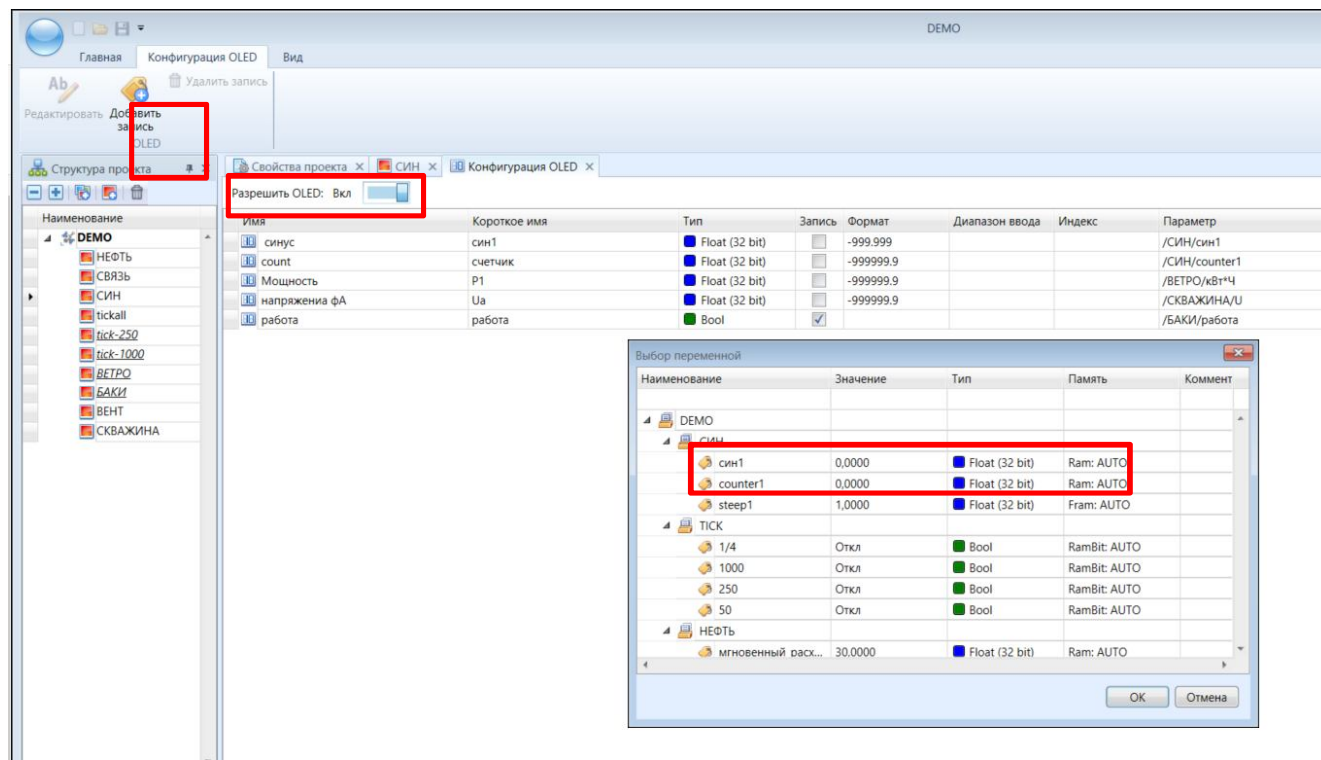


8.2 Настройка переменных в YART Studio

Для настройки переменных доступных в пульте, откройте проект и выберите пункт меню «OLED»



В открывшемся окне необходимо установить «Разрешить OLED» в положение «Вкл.», нажать «Добавить запись» и выбрать переменную:





Откроется окно для настройки отображения выбранной переменной на пульте:

«Параметр» - к какому параметру алгоритма привязан параметр пульта

«Имя» - полное имя переменной для пульта

«Короткое имя»

«Формат» - формат отображения в пульте

«Запись» - устанавливается в «on», если переменная должна изменяться с пульта

После завершения редактирования нажмите «Ok».



9. Использование модулей расширения

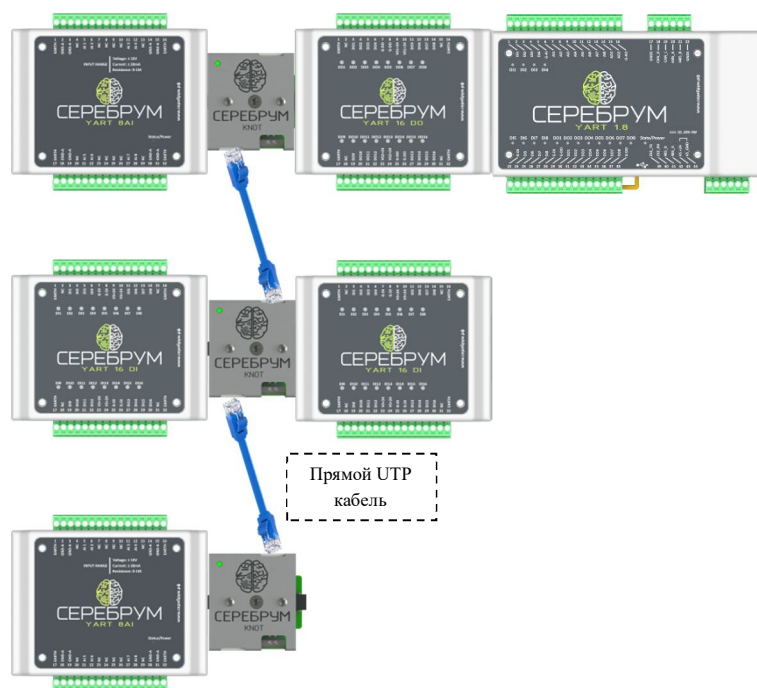


Рисунок 5. Возможная топология шины расширения

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

При проектировании конфигурации шины необходимо учитывать следующие правила:

- Контроллер YART-1.8 поддерживает до восьми различных модулей расширения на одной шине
- Первым к контроллеру должен подключаться модуль расширения с адресом 0
- 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 0 – 7
- Порядок присоединения модулей может быть любым за исключением модуля с адресом 0 (всегда подключается к ПЛК)

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

YART-1.8 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.

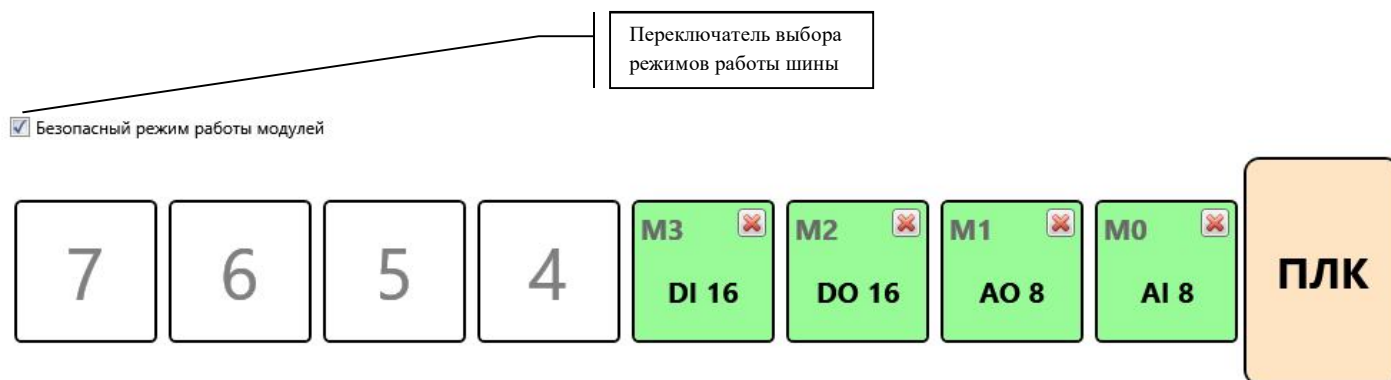


Рисунок 7. Конфигурация модулей расширения

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 YART-1.8 доступны адреса модулей в диапазоне 0 – 7.

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

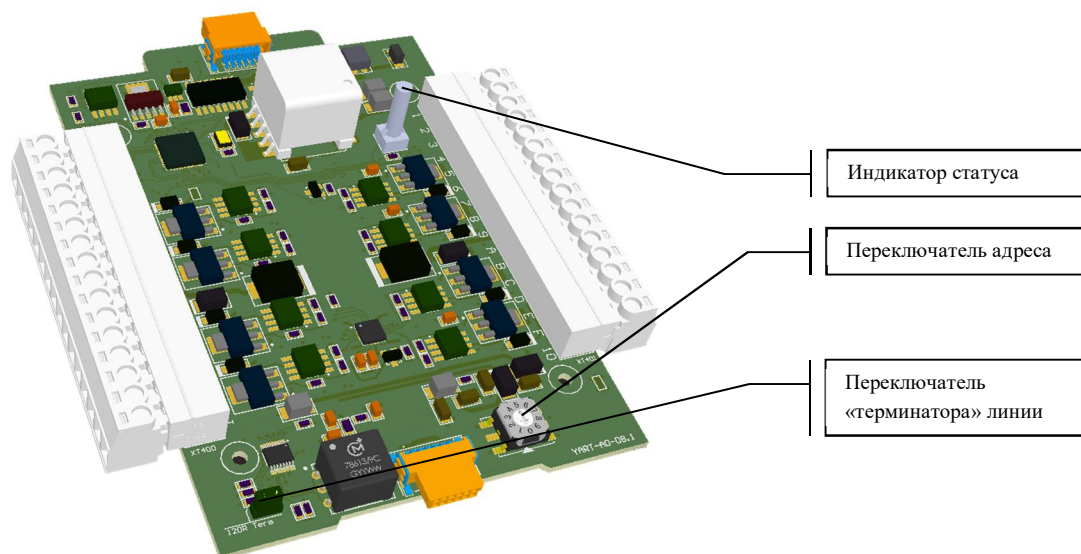


Рисунок 8. Переключатели модулей расширения

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.



10. Индикация состояний

Отображение состояния ПЛК YART-1.8 осуществляется при помощи светодиодного индикатора (**поз. 8 Рис. 1**)

Различные режимы отображения представлены в **Таблице 7**.

Таблица 7. 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



11. ПЛК, особенности архитектуры и память

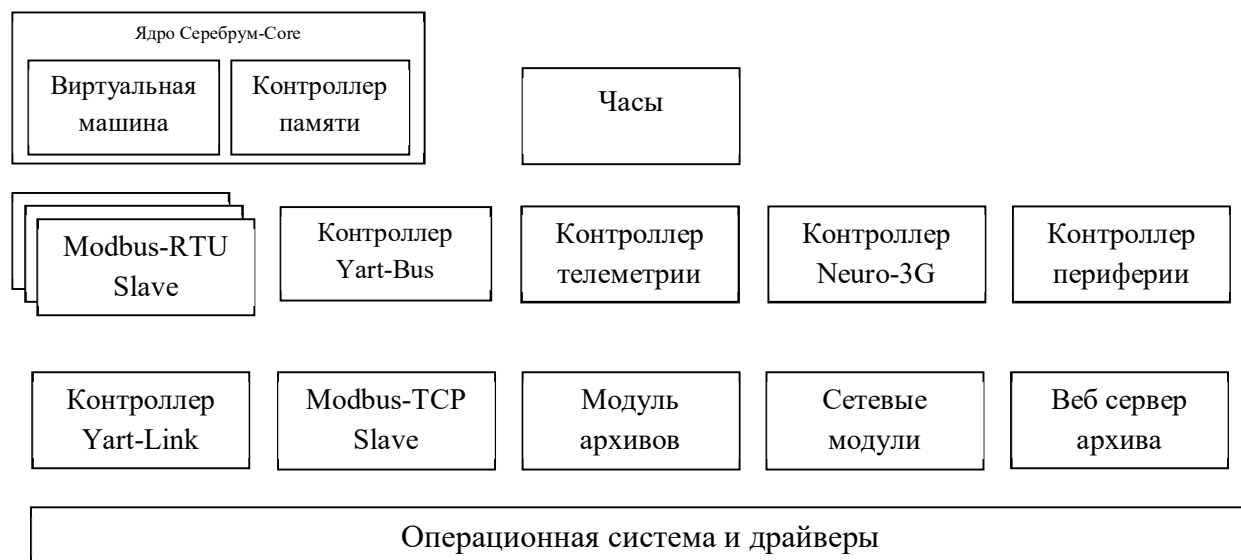


Рисунок 9. Архитектура контроллера YART-1.8

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.

YART-1.8 содержит встроенные периферийный модули для непосредственного подключения к линиям датчиков и управления.



11.1 Память программируемого контроллера

Вся память контроллера (**Таблица 8**) 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

11.1.1 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

11.1.2 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.

11.1.3 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.



11.2 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

12. Телеметрия

Все программируемые контроллеры Серебрум изготавливаются с поддержкой системы удаленного мониторинга и управления. В зависимости от типа используемого оборудования возможно подключение через Ethernet, 2G/3G модем или оба варианта.

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.

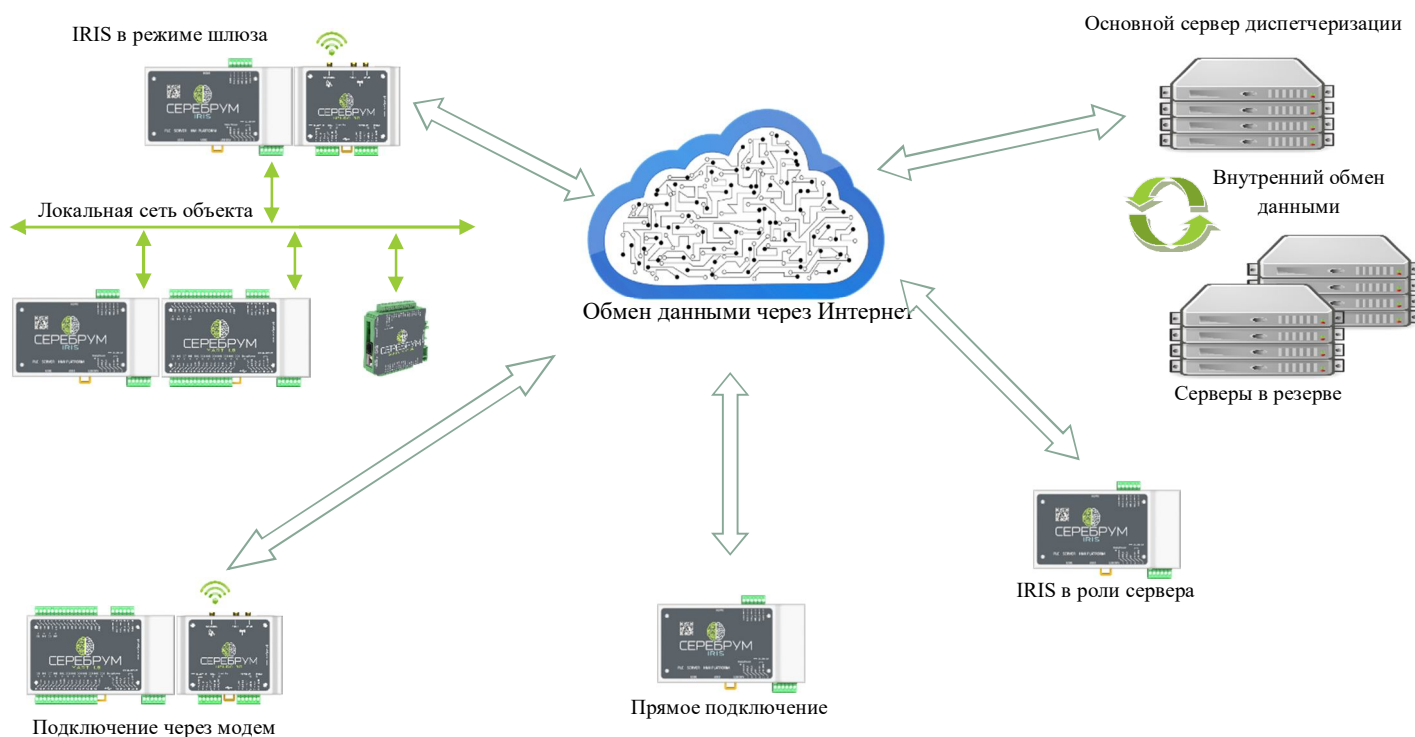


Рисунок 10. Удаленная диспетчеризация и управление Серебрум

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.



12.1 Подключение к удаленному серверу телеметрии

Контроллер YART-1.8 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» или «Модем на RS232», а также укажите настройки соединения с сервером телеметрии

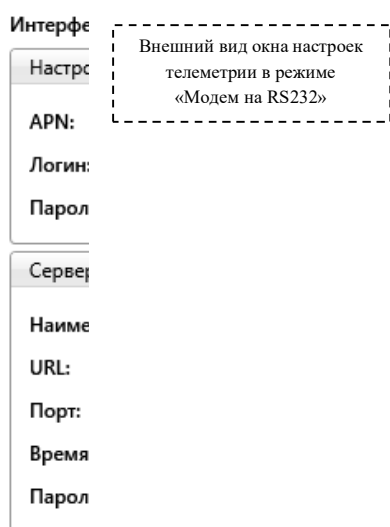
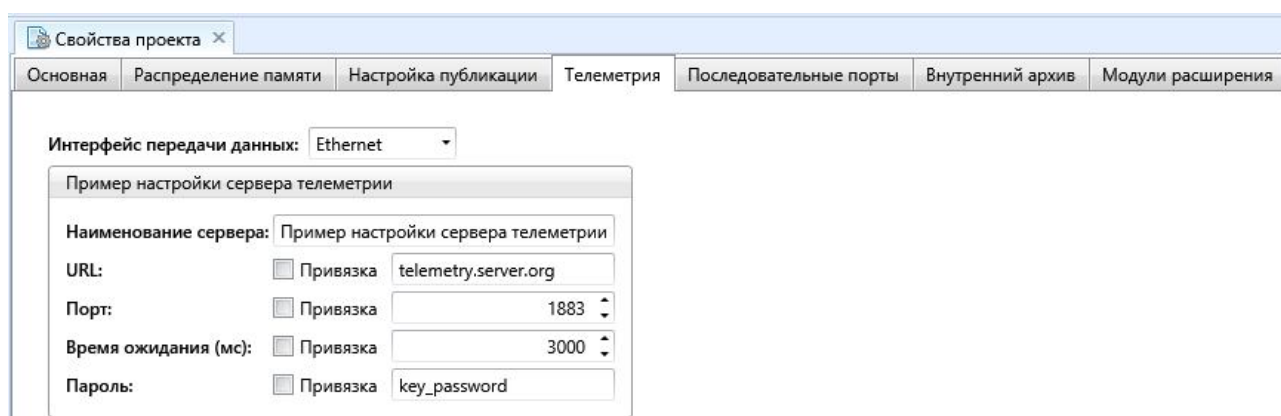


Рисунок 11. Простая настройка телеметрии

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.

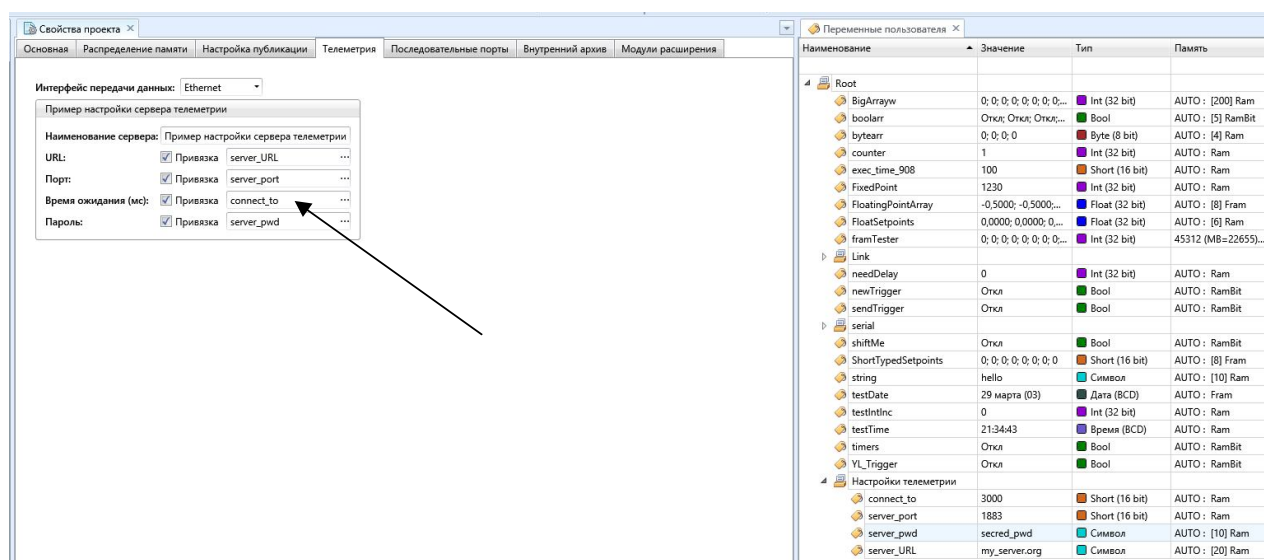


Рисунок 12. Привязка параметров подключения к серверу к переменным

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.

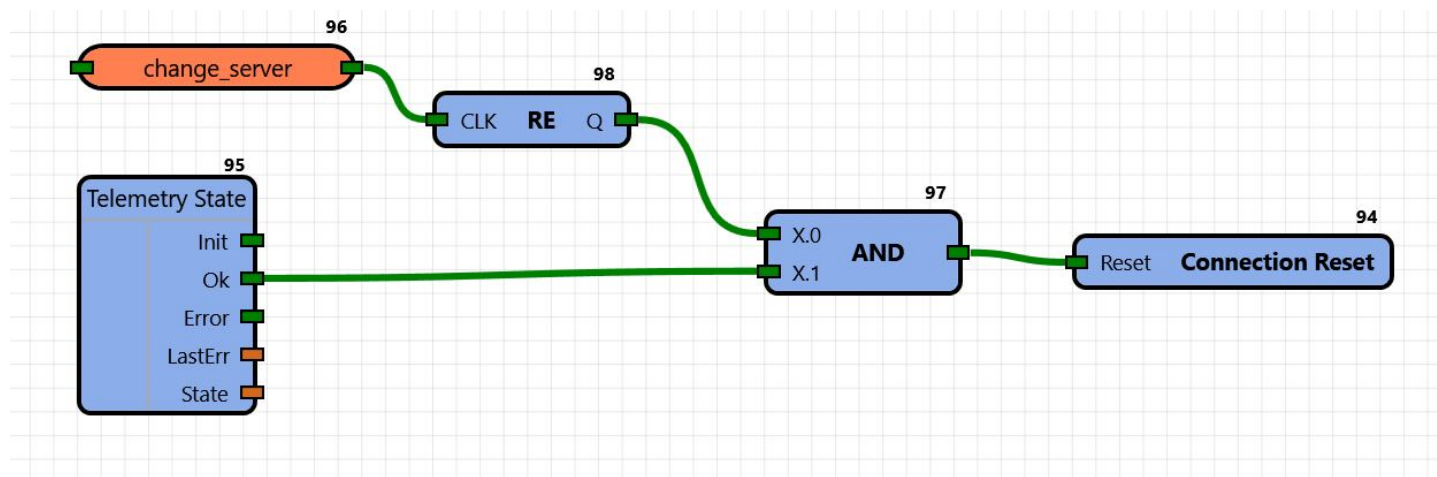


Рисунок 13. Пример команды сброса соединения с сервером



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 считает 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; 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)	Data (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.

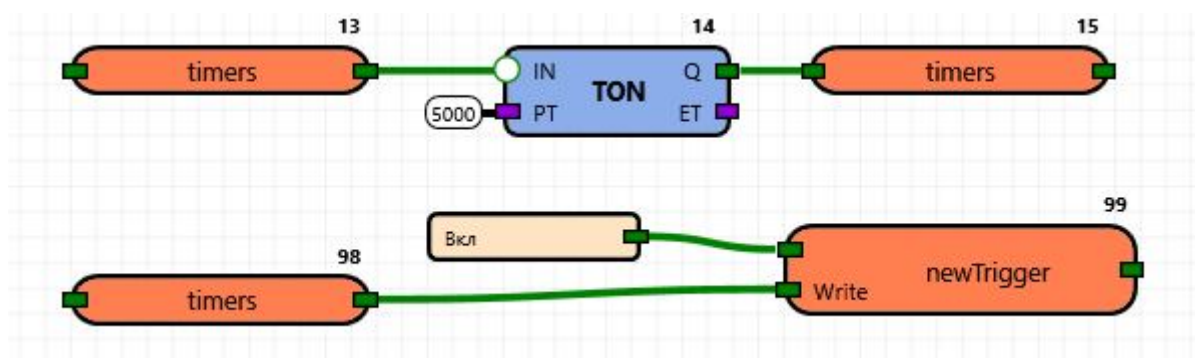


Рисунок 15. Отправка данных на сервер каждые 5 секунд

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.

13. 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.

13.1 Modbus TCP

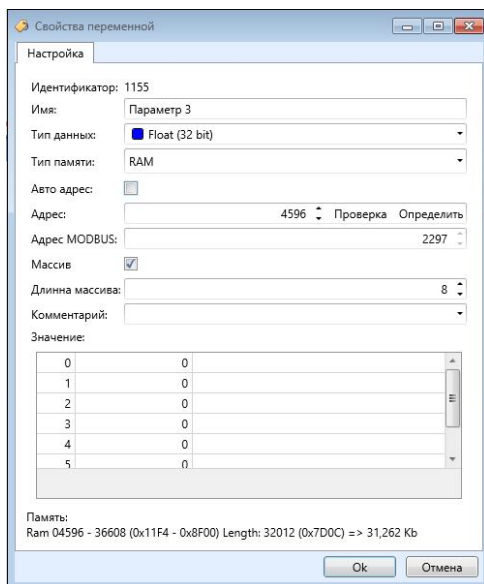
Modbus TCP является основным протоколом обмена для связи ПЛК с системой YartStudio. Кроме того, данный протокол поддерживается большинством производителей промышленного оборудования и используется повсеместно.

В контроллере YART-1.8 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.



Идентификатор:	1155
Имя:	Параметр 3
Тип данных:	Float (32 bit)
Тип памяти:	RAM
Авто адрес:	☐
Адрес:	4596
Адрес MODBUS:	2297
Массив:	☒
Длина массива:	8
Комментарий:	
Значения:	
0	0
1	0
2	0
3	0
4	0
5	0

Память:
Ram 04596 - 36608 (0x11F4 - 0x8F00) Length: 32012 (0x7D0C) => 31,262 Kb



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.

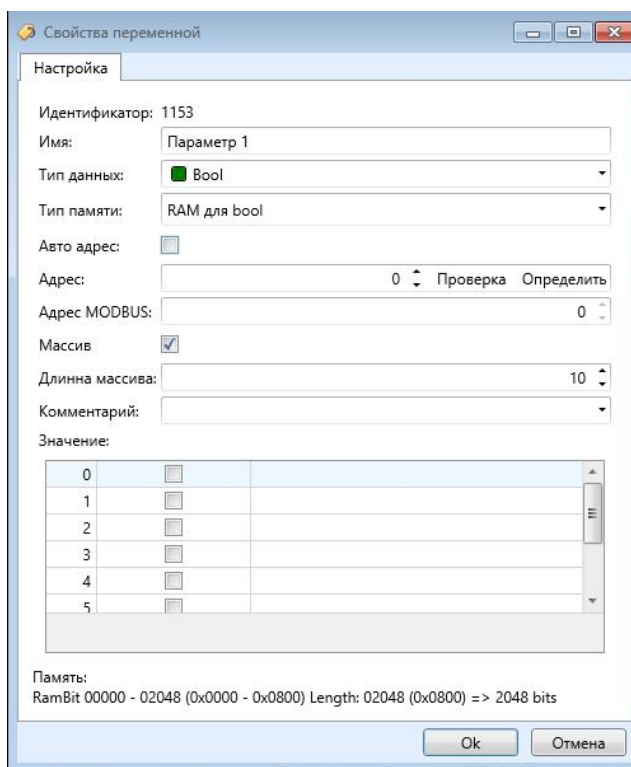


Рисунок 17. Пример с Bool

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.

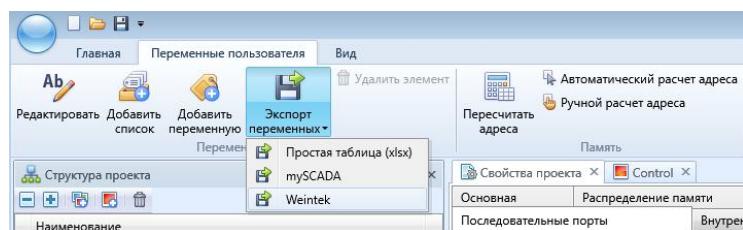


Рисунок 18. Меню экспорта списка переменных

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.



13.2 Modbus RTU

Работа Modbus RTU осуществляется через последовательные порты контроллера.

В **Таблице 10** указано обозначение портов.

Порт COM1 может быть задействован для работы с Neuro-3G и быть недоступен для настройки.

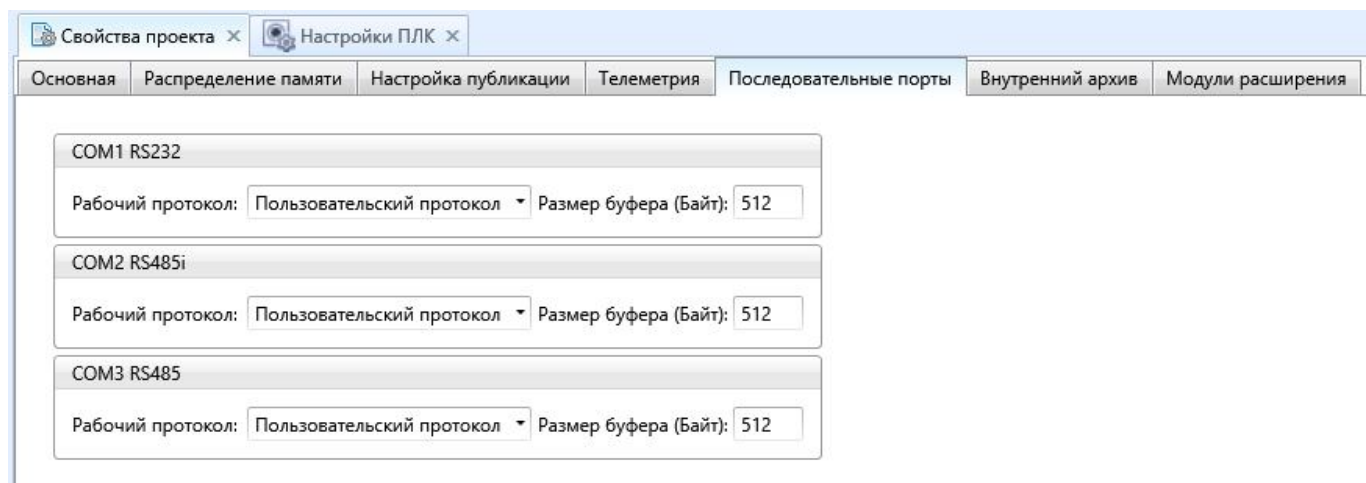


Рисунок 19. Базовая настройка последовательных портов

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

- Modbus RTU Master
- Modbus RTU Slave
- Пользовательский протокол

Таблица 10. Последовательные порты

Именованье	Расположение	Обозначение в программе
COM1	Последовательный порт RS-232, клеммы 39,40	1
COM2	Изолированный порт RS-485, клеммы 20, 21	2
COM3	Неизолированный порт RS-485, клеммы 41, 42	3

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 13.1).

Однако команды доступа к Boot разделам памяти недоступны.

Для правильного использования режима Modbus RTU Slave в прикладной программе необходимо указать локальный адрес устройства и режим работы последовательного порта (**Рис. 20**).

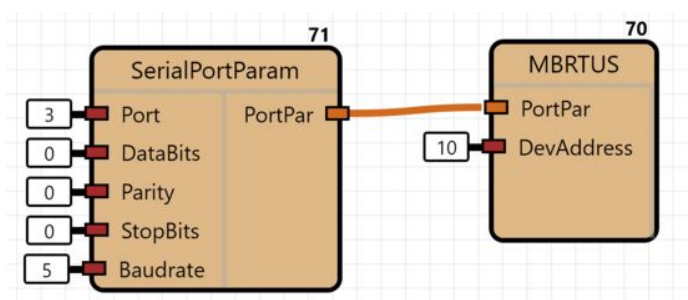


Рисунок 20. Настройка параметров Modbus RTU Slave

В блоке *SerialPortParam YART-1.8* указывается условный номер порта (Таблица 10), 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 YART-1.8 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.

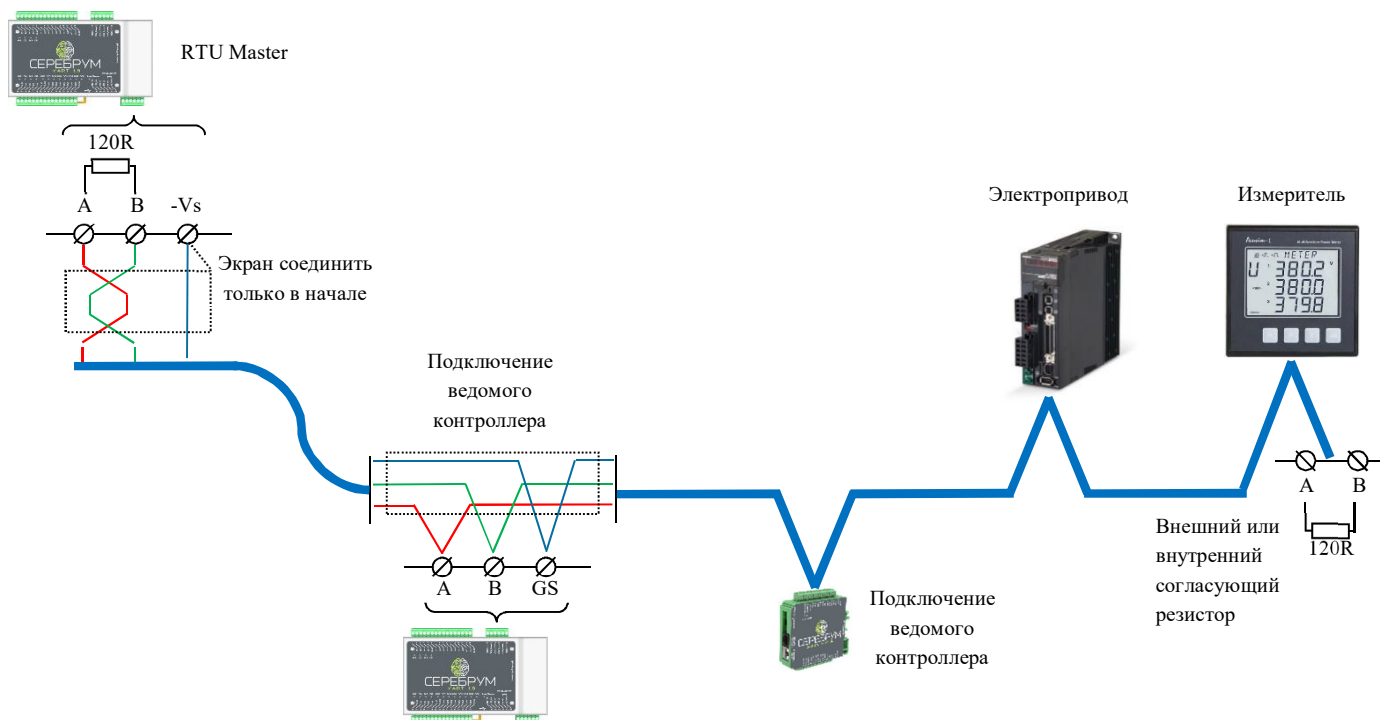


Рисунок 21. Подключение к шине Modbus RTU

На **Рисунке 21** показана приблизительная схема подключения устройств в шине RS-485 (Modbus RTU). В качестве ведущего устройства выступает контроллер YART-1.8 – COM3.

Поскольку в данной конфигурации ведущий физически расположен в самом начале шины, то для порта COM3 включается согласующий резистор 120 Ом. Для этого устанавливается джампер (**поз. 7, Рис. 1**).

Другой контроллер YART-1.8 подключается к шине через порт COM2.

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

13.2.1 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.

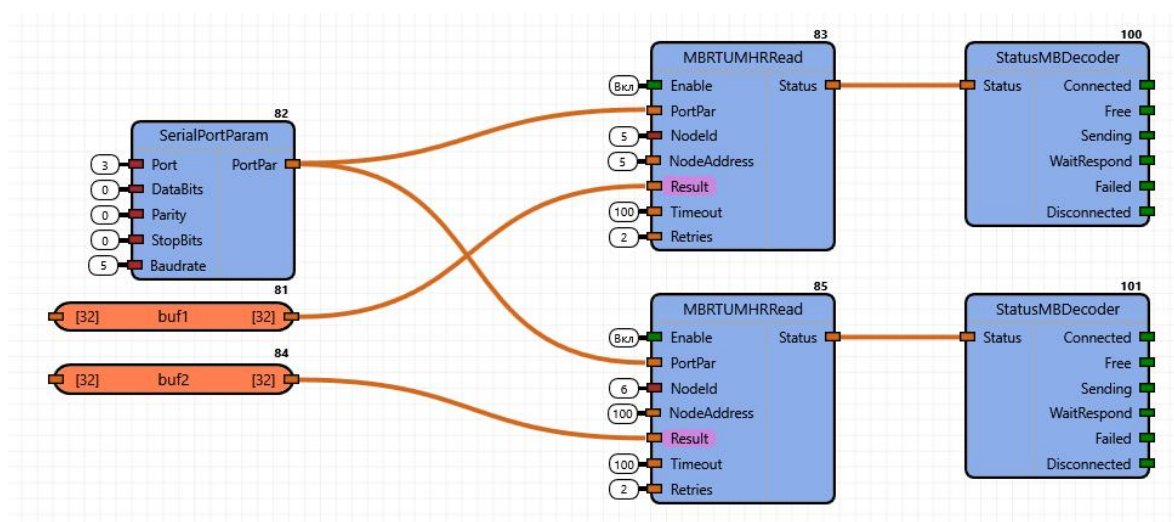


Рисунок 22. Конфигурация обмена Modbus RTU Master

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 – **MBRTUMHRead** 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».

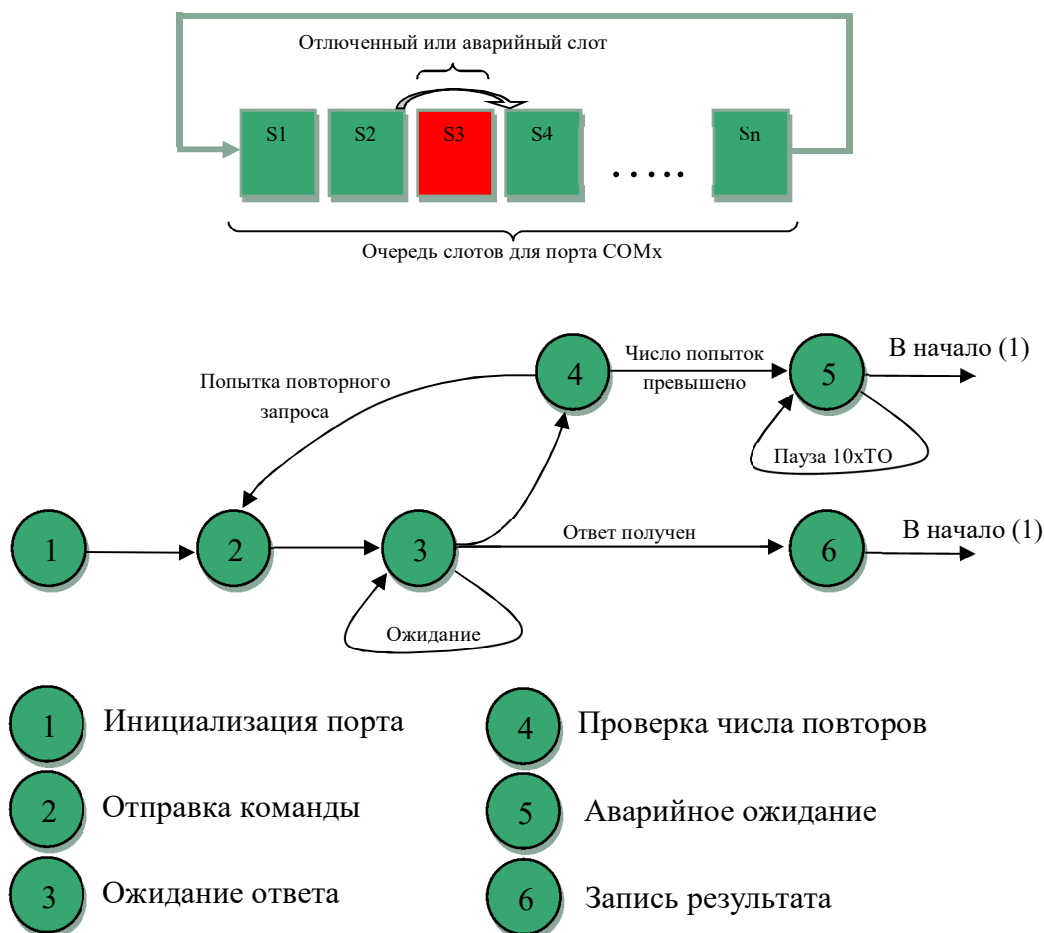


Рисунок 23. Работа слотов и диаграмма состояний коммуникационного слота.

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.



13.3 Пользовательские протоколы

В контроллере YART-1.8 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.



13.4 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**.

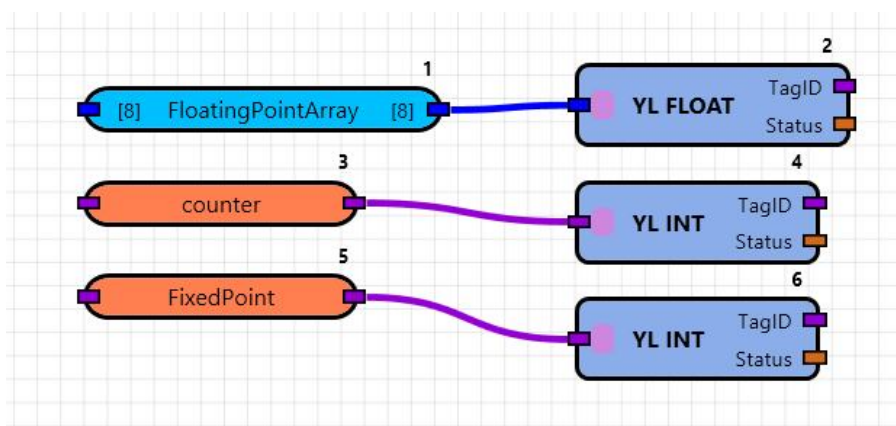


Рисунок 24. Пример программы передатчика YartLink

В процессе работы программы выход блока формирования пакета выдает идентификатор переменной для подписи 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 отpravку 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 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.

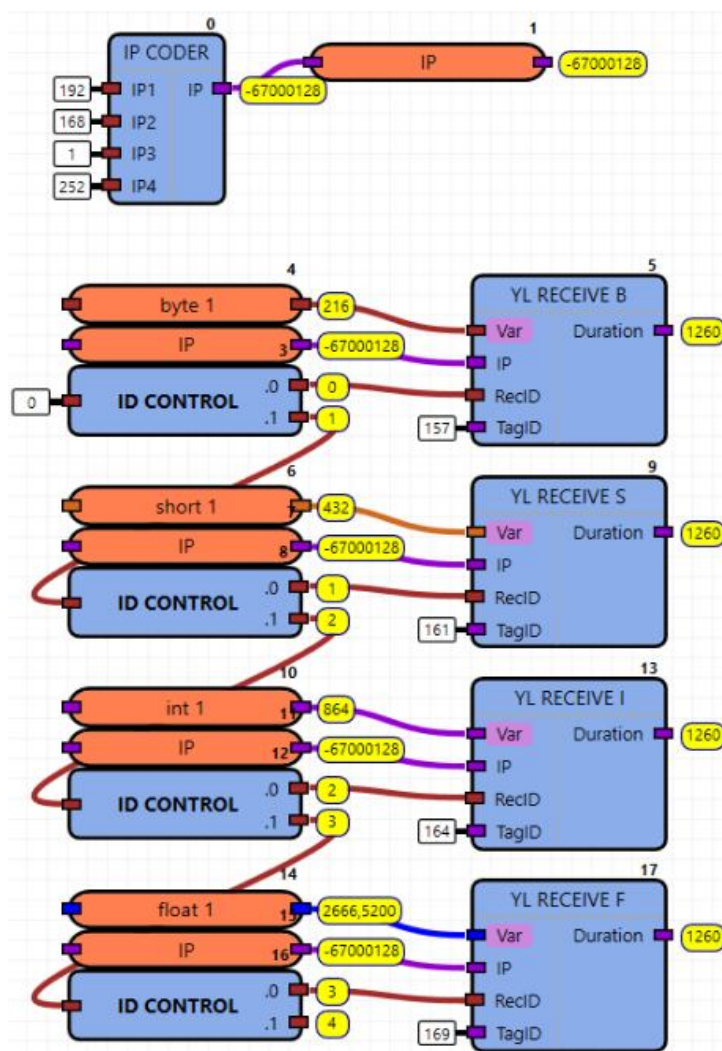


Рисунок 25. Подписка на глобальные публикуемые данные

13.5 USB порт Yart Studio

В ряде случаев сетевое соединение с ПЛК при помощи Ethernet невозможно.

Для прямого подключения к YART-1.8 допускается использование USB подключения через встроенный порт (**поз. 5 Рис. 1**). При подключении к ПК в системе появится дополнительный USB-COM порт.

ОС должна определить новое устройство («Устройство с последовательным интерфейсом», **Рис. 26**).

В процессе установки Yart Studio доступен режим автоматической настройки драйвера для USB порта контроллера.

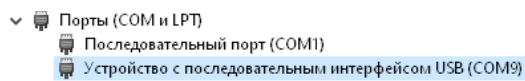


Рисунок 26. Подключение YART-1.8 к ПК через USB порт



Доступ к данным YART-1.8 будет осуществляться через протокол Modbus-RTU по адресу 1.

14. Работа с периферийными модулями контроллера

14.1 Аналоговые входы

В YART-1.8 доступны восемь универсальных аналоговых входов, **Рис. 27**. В зависимости от конфигурации возможны следующие режимы измерения:

- Измерение входного напряжения
- Измерение входного тока
- Измерение подключенного сопротивления
- Имитация дискретных входов

Измерение входного тока осуществляется при помощи подключаемого нагрузочного резистора (**поз. 2, Рис. 1**).

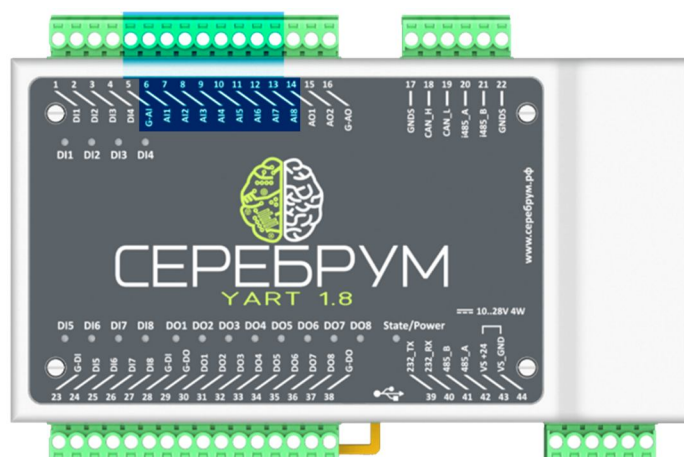


Рисунок 27. Аналоговые входы контроллера

Переключение схемы для измерения напряжения и сопротивления осуществляется автоматически.

На **Рисунке 28** показаны возможные схемы подключения входных сигналов к аналоговым входам:

- к входу AI8 подключен датчик с токовым выходом по двухпроводной схеме
- к входу AI7 подключен датчик с токовым выходом
- вход AI6 используется в режиме эмуляции дискретного входа
- вход AI5 измеряет входное напряжение
- вход AI3 осуществляет измерение внешнего сопротивления



При проектировании схем с использованием аналоговых входов следует уделить внимание следующим особенностям схемы:

- ✓ Линии подключения резистивных датчиков должны быть как можно короче
- ✓ Монтаж проводов соединений с резистивными датчиками должен осуществляться на максимальном расстоянии от силовым кабелей, особенно преобразователей частоты и выпрямителей
- ✓ Общий сигнал токовых датчиков должен быть соединен с клеммой G-AI. В случае использования питания ПЛК клеммы G-AI и VS_GND должны соединяться
- ✓ Аналоговая «земля» гальванически соединена с питанием контроллера. Убедитесь, что между G-AI и VS_GND нет токов выравнивания потенциалов
- ✓ При необходимости измерения сопротивления на расстоянии более 10м рекомендуется использовать преобразователи в токовый сигнал

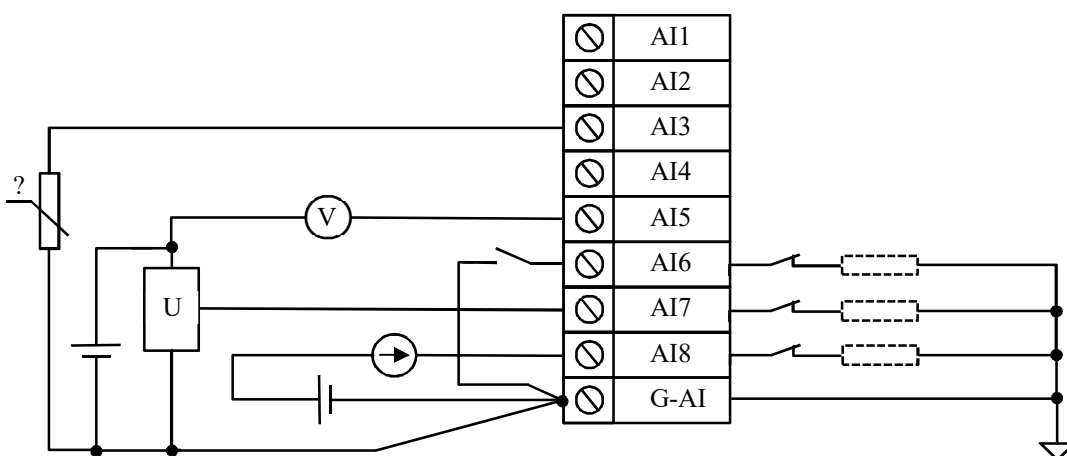


Рисунок 28. Подключение к аналоговым входам

Дальнейшая работа с аналоговыми входами выполняется в Yart Studio. Для каждого из используемых аналоговых входов необходимо выбрать требуемый режим работы, **Рис. 29**.

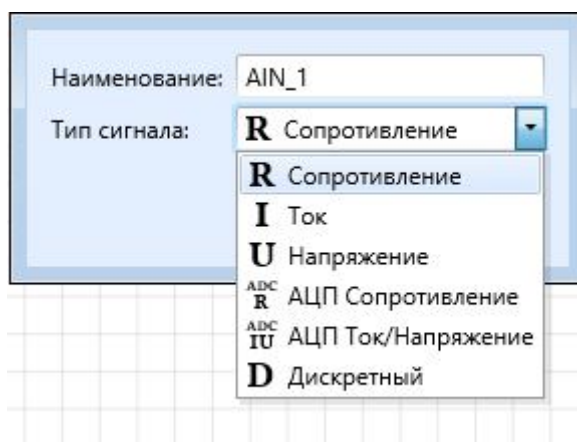


Рисунок 29. Установка режимов аналогового входа



Так, для примера выше для AI7 и AI8 необходимо установить режим «I Ток», AI6 – «D Дискретный», AI5 – «U Напряжение» и AI3 – «R Сопротивление».

Выход каждого канала – число с плавающей точкой в формате измеряемой величины, **Рис. 30.**

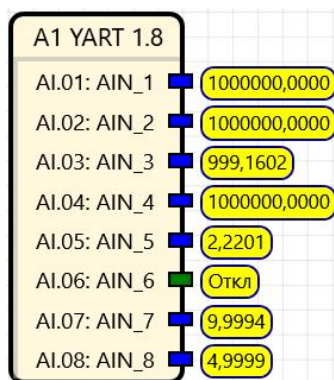


Рисунок 30. Установка режимов аналогового входа

Токовые каналы выдают значения в миллиамперах, напряжение выражается в Вольтах, величина сопротивления выражается в Омах. При использовании аналогового входа в качестве дискретного выход такого канала имеет тип Bool.

В ряде случаев удобно использовать необработанный код АЦП, для этого в списке режимов работы добавлены «ADC R АЦП Сопротивление» и «ADC IU АЦП Ток/Напряжение».



14.2 Дискретные входы

К контроллеру YART-1.8 может подключаться до восьми линий дискретных сигналов с номинальным напряжением «лог. 1» 24В. Каждый из входов (Рис. 31) может использоваться в режиме автоматического низкоскоростного счетчика импульсов (максимальная частота подсчета 100 Гц).

Входы DI2:DI4 и DI7:DI8 используются как входы для двух встроенных (аппаратных) модулей счета/захвата, позволяющих обрабатывать сигналы от высокоскоростных датчиков скорости.

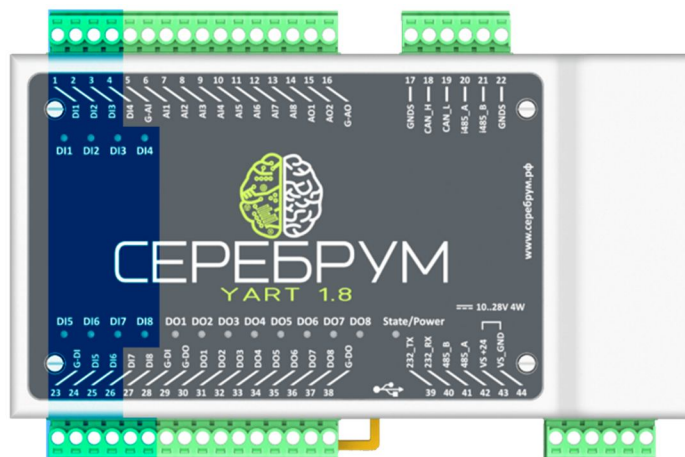


Рисунок 31. Дискретные входы контроллера

Подключение внешних сигналов к входам DI_x должно выполняться по схеме, **Рис. 32**

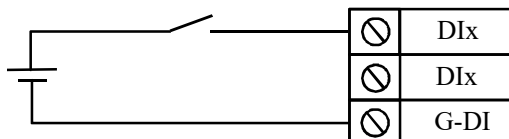


Рисунок 32. Схема подключения дискретных входов

Настройка режимов работы дискретных входов осуществляется при помощи Yart Studio (**Рис. 33**)

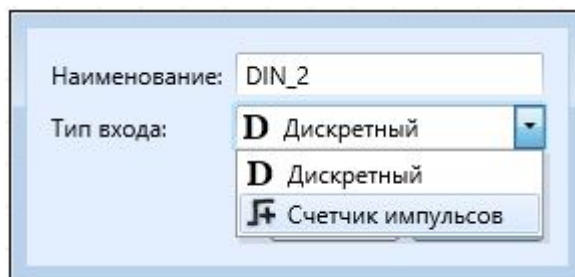


Рисунок 33. Дискретные входы контроллера

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.

Необходимо отметить, что работа счетчика импульсов не зависит от текущего времени выполнения прикладной программы пользователя.

При использовании аппаратных счетчиков конфигурация осуществляется при помощи набора функциональных блоков, **Рис. 34**.

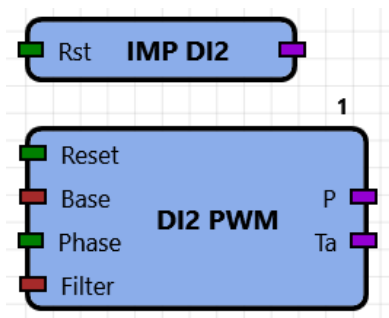


Рисунок 34. Функциональные блоки аппаратных счетчиков

Каждый из двух счетчиков настраивается для работы в одном из трех режимов

Инкрементальный энкодер. В этом режиме на каждый счетчик подключаются два сигнала: А:В к входам DI2:4 или DI7, **Рис. 35**.

В зависимости от направления вращения вала значение счетчика будет увеличиваться/уменьшаться пропорционально скорости вращения.

Аппаратные счетчики имеют разрядность 16 бит, поэтому в режиме энкодера требуется дальнейшая обработка для предотвращения переполнения значения.

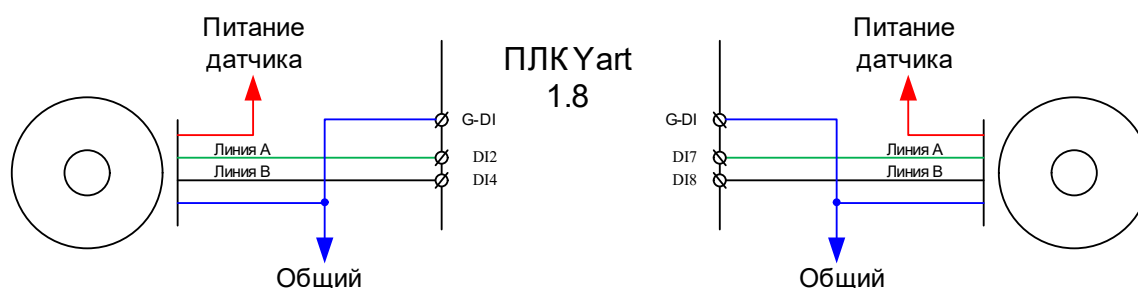


Рисунок 35. Подключение инкрементальных энкодеров

Счетчик импульсов – режим похож на режим энкодера кроме отсутствия определения направления. Контроллер считает число переходов сигнала из 0 в 1 (в зависимости от выбранной полярности). Подключение осуществляется по схеме, **Рис. 36**.

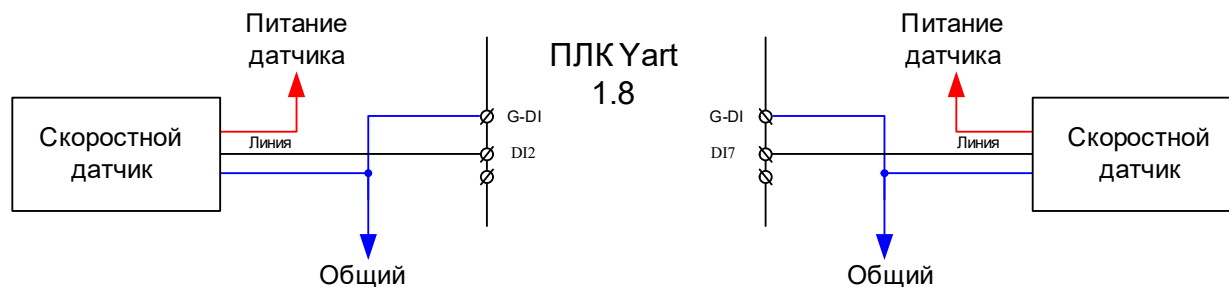


Рисунок 36. Подключение скоростных датчиков

Режим захвата. В данном режиме контроллер определяет длительность активного состояния на входе (Рис. 37).

На выходе формируется два значения: длительность активного состояния (T_a) и период (P).

Разрешающая способность определяется значением Base: 1 мкс, 10 мкс, 100 мкс, 1000 мкс.

Таким образом измеряемая длительность ограничивается 65.535 мс, 655.35 мс, 6553.5 мс и 65535 мс соответственно.

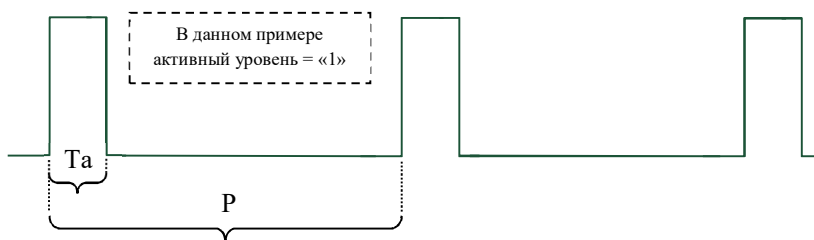


Рисунок 37. Работа модуля захвата

Значения T_a и P обновляются на каждом периоде сигнала.

Все дискретные входы контроллера оснащаются светодиодными индикаторами – при активном состоянии входа светодиод светится.



14.3 Аналоговые выходы

YART 1.8 оснащен двумя независимыми аналоговыми токовыми выходами, **Рис. 38**. На плате контроллера установлены переключатели, позволяющие получать напряжение пропорциональное величине тока за счет нагрузочного резистора, **Рис. 39**.

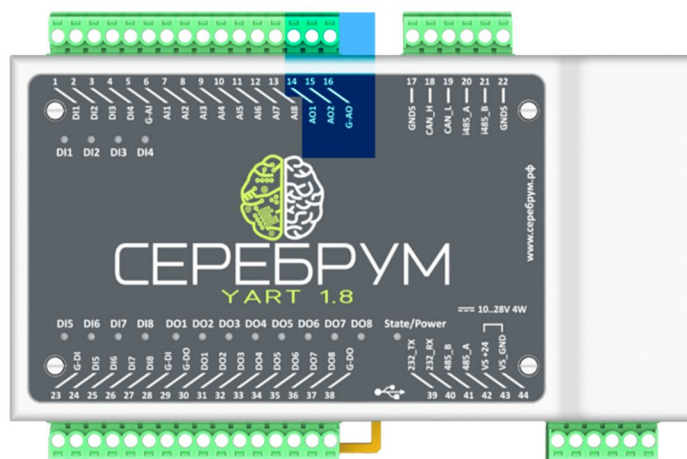


Рисунок 38. Аналоговые выходы контроллера

При работе аналоговых выходов с подключенными нагрузочными резисторами (выход по напряжению) необходимо учитывать входное сопротивление приемника. В случае устройств с низким входным сопротивлением целесообразно использовать токовый режим выхода.

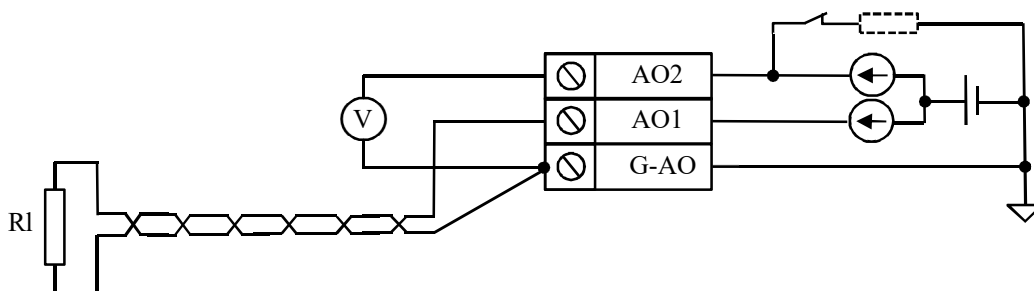


Рисунок 39. Подключение аналоговых выходов

В примере (**Рис. 39**) выход AO2 – использует встроенный нагрузочный резистор, таким образом формируется падение напряжения (0-10В) между клеммами G-AO и AO2.

Выход AO1 – токовый (0..20мА), в примере нагрузочный резистор установлен непосредственно на приемнике токового сигнала (ПЛК, электропривод и т.д.).

Токовые выходы YART 1.8 работают от встроенного источника питания напряжением 15В. Таким образом максимальное сопротивление нагрузки не должно превышать 750 Ом.

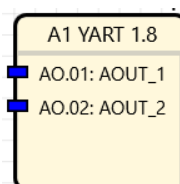


Рисунок 40. Подключение аналоговых выходов

Для использования аналоговых выходов достаточно перенести блок «Выход»-«Аналоговый» поля «Состав оборудования» в рабочую область программы (**Рис. 40**).

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



14.4 Дискретные выходы

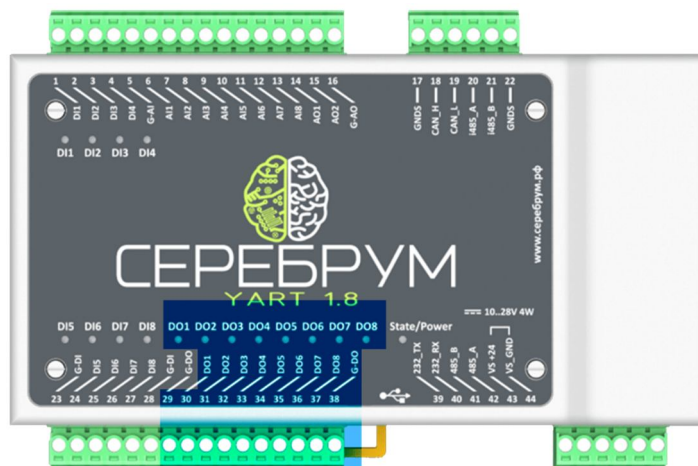


Рисунок 41. Дискретные выходы контроллера

В состав YART 1.8 входят восемь дискретных выходов на основе твердотельных реле. Вся группа выходов имеет групповую гальваническую развязку с выводов общей линии на клеммы 29 и 38.

На **Рисунке 42** показана предполагаемая схема подключения нагрузки к выходам контроллера.

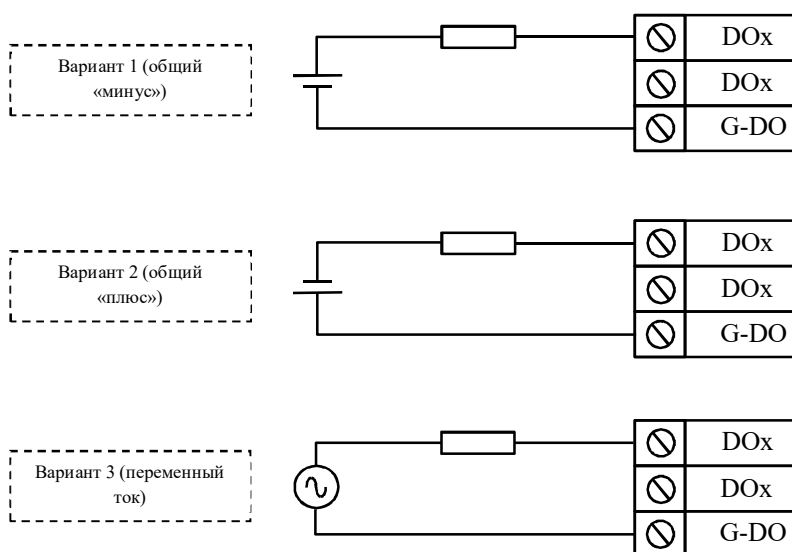


Рисунок 42. Подключение нагрузки к дискретным выходам контроллера

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 43**.

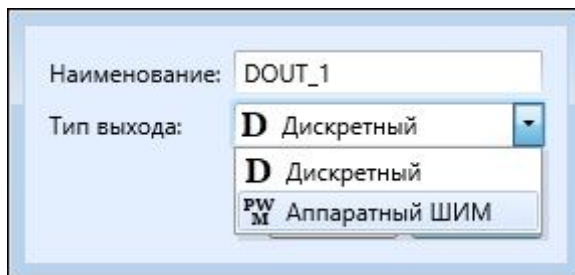


Рисунок 43. Конфигурация дискретного выхода

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 44** показан пример использования ШИМ.

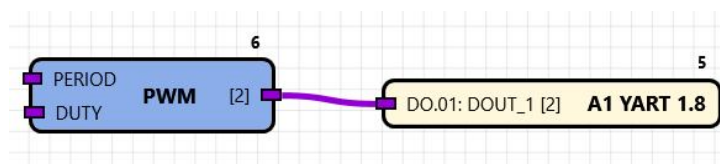


Рисунок 44. Формирование ШИМ на выходе ПЛК

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



14.5 Настройка часов

В YART 1.8 встроены часы реального времени с резервным питанием от литиевой батареи 3В (CR2032). Работа часов не зависит от выполнения прикладной программы пользователя за исключением использования команд программной установки времени.

Значение текущего времени/даты доступно в программе пользователя (блоки GET DATE, GET_TIME).

При подключении к ПЛК текущее время/дата контроллера доступно для чтения и установки.

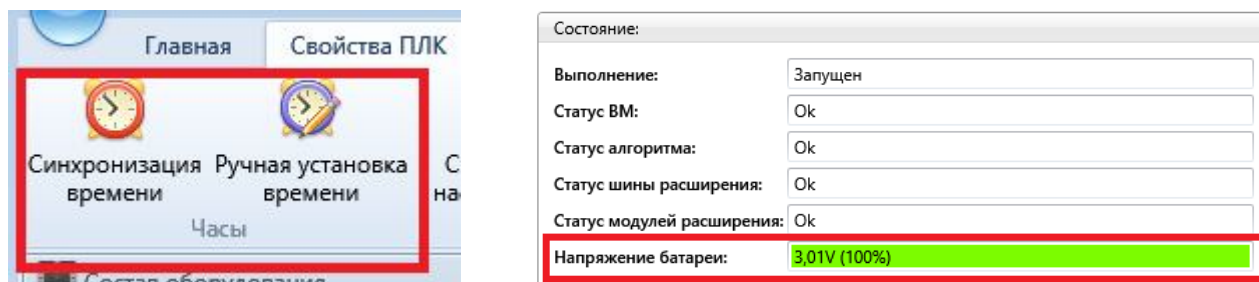


Рисунок 45. Подключение нагрузки к дискретным выходам контроллера

Состояние батареи отображается в поле «Напряжение батареи» раздела «Состояние».

При падении напряжения батареи ниже 2.5В батарею необходимо заменить. Для замены снимите верхнюю крышку контроллера, выкрутите болты крепления платы и снимите плату с пластикового основания. Замените батарею на новую и осуществите сборку контроллера в обратной последовательности.

Расчетный срок эксплуатации новой батареи – 20 лет.

Величина напряжения доступна для контроля из пользовательской программы через блок U BATTERY, Рисунок 46.



Рисунок 46. Подключение нагрузки к дискретным выходам контроллера



15. 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.

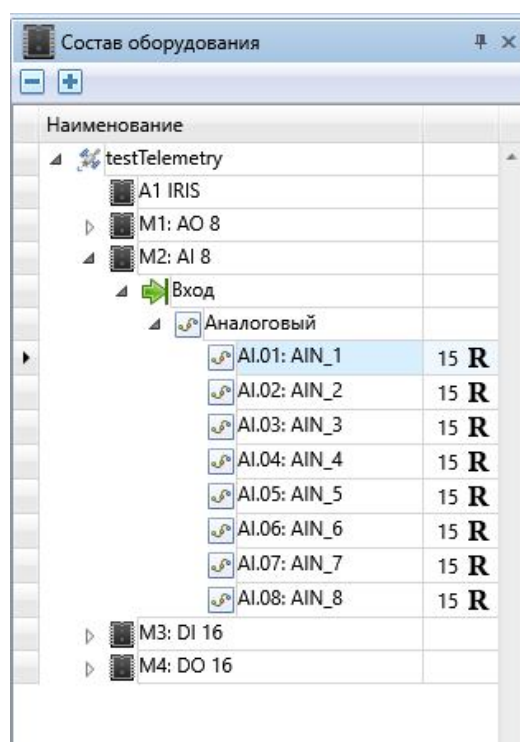


Рисунок 47. Панель настройки периферийных модулей



15.1 Аналоговые входы

Модуль AI8 (**Рис. 48, 49**) основан на 12-разрядном АЦП со встроенным программируемым усилителем.

Подключение модуля рекомендуется выполнять в соответствии со схемой (**Рис. 48**). При этом клеммы «Earth», 1, 16, 17, 32 должны подключаться к контуру заземления шкафа непосредственно в месте установки модуля.

Необходимо подключение всех клемм заземления.

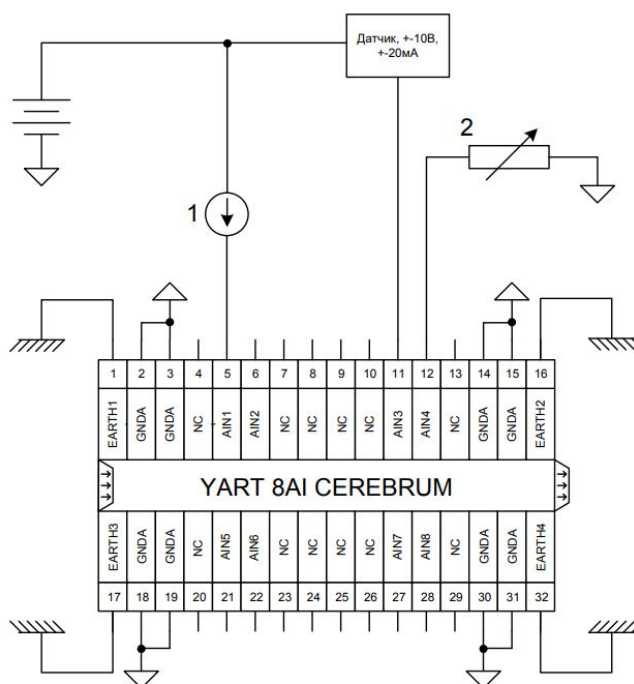


Рисунок 48. Модуль 8AI

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.

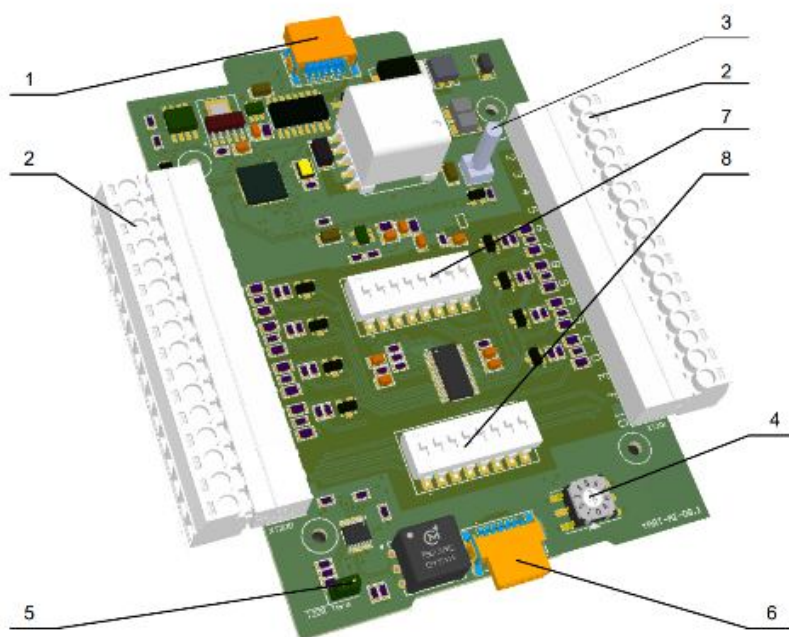


Рисунок 49. Модуль 8AI

- 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 11. 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.

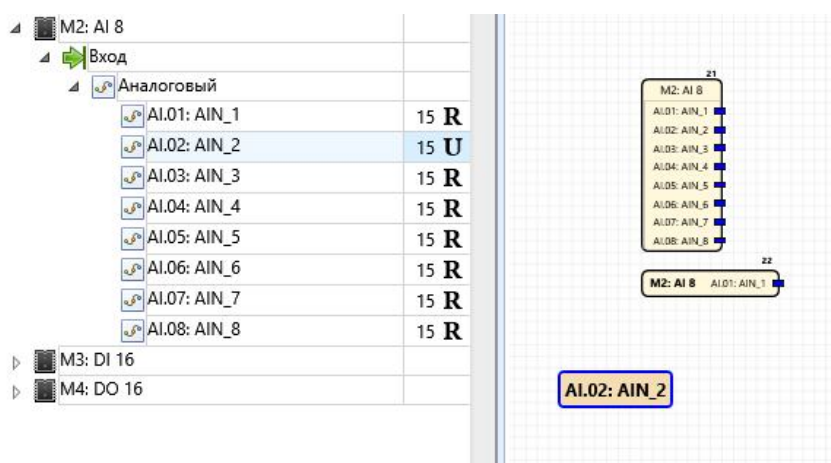


Рисунок 50. Функциональные блоки аналоговых входов

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.



15.2 Аналоговые выходы

Модуль 8АО (**Рис. 51**) 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 51**). 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.

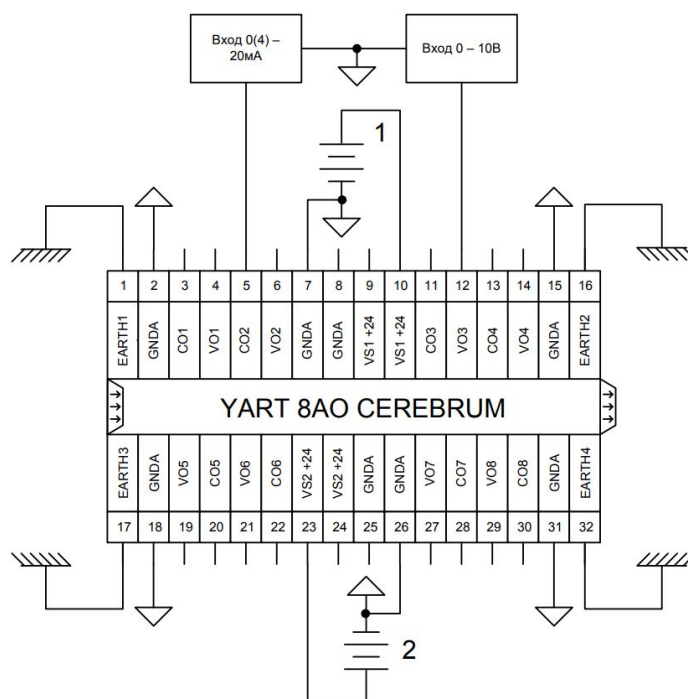


Рисунок 51. Модуль 8АО

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 п. 9.1, **Рис. 31**.

На вход блока (**Рис. 52**) 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

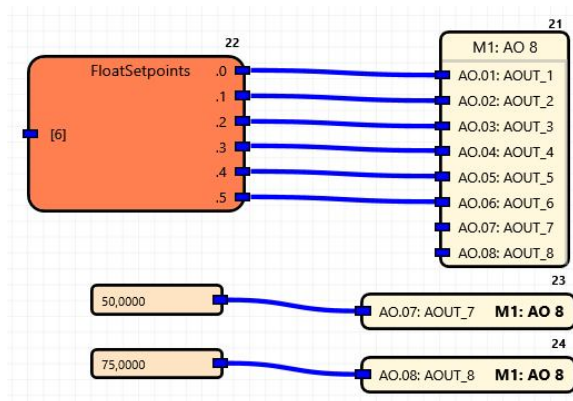


Рисунок 52. Использование модулей аналоговых выходов.



15.3 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.

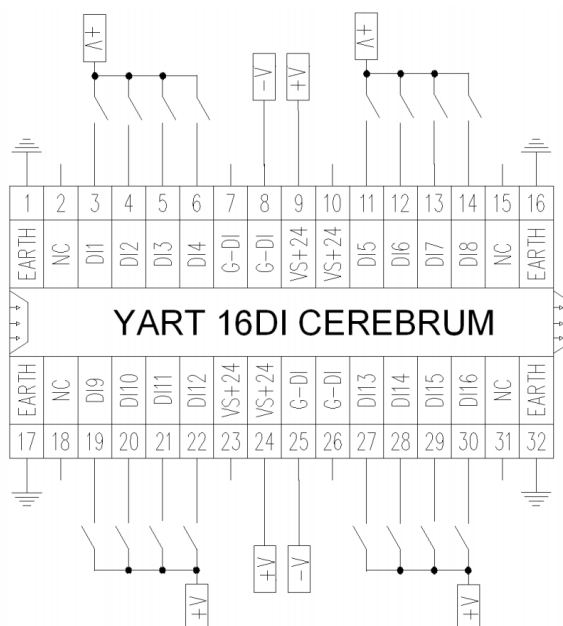


Рисунок 53. Модуль 16DI

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

Клеммы 1, 16, 17, 32 (**Рис. 53**) 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. 54**).

Блоки допускается размещать в любом месте программы.

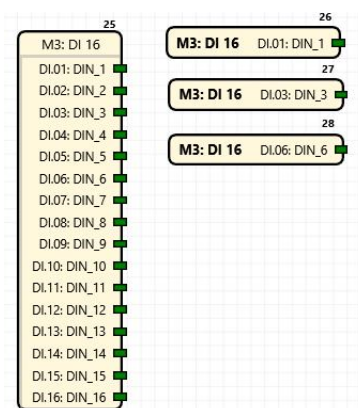


Рисунок 54. Блоки дискретного ввода



15.4 Дискретные выходы

Модуль 16DO содержит 16 выходных каналов с максимальной токовой нагрузкой 1А. Для каждого выходного блока из восьми линий (**Рис. 55**) 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. 55**) должны быть присоединены к внутреннему контуру заземления.

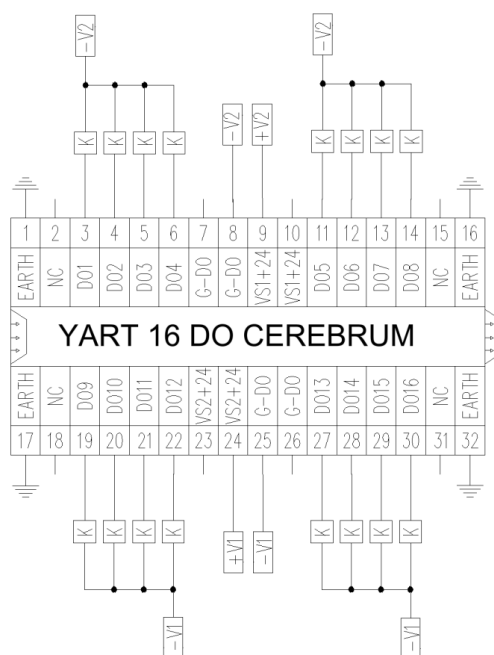


Рисунок 55. Модуль 16DO

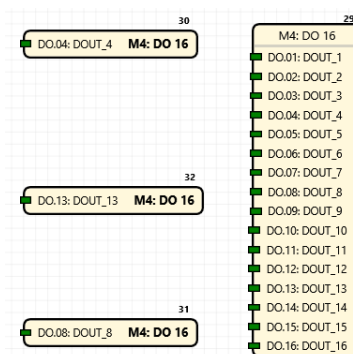


Рисунок 56. Функциональные блоки DO

Значения дискретных выходов задаются при помощи функциональных блоков DO (**Рис. 56**). 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.



16. Архив данных

Встроенный слот для uSD карты (**поз. 6 Рис. 1**) позволяет сформировать локальный архив исторических данных контроллера.

В процессе выполнения прикладной программы данные, указанные в окне "Внутренний архив" (**Рис. 57**) будут автоматически сохранены на карту.

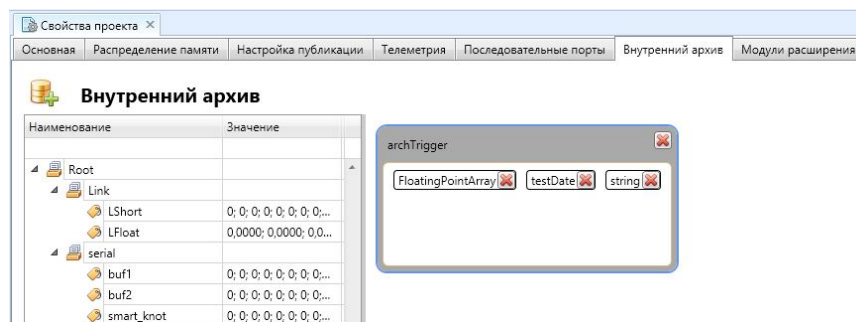


Рисунок 57. Настройка встроенного архива данных

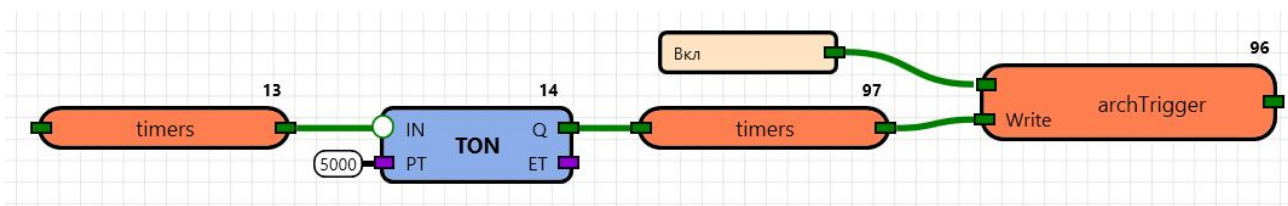


Рисунок 58. Сохранение данных в архив

Программа архива постоянно проверяет флаги записи в архив и переносит на uSD карту данные в случае "1" в флаге управления. На **Рисунке 58** показан пример программы для сохранения данные в архив каждые 5 секунд.

После того, как данные сохранены флаг записи сбрасывается в "0".

Для работы с контроллером uSD карта должна иметь один раздел данных, отформатированный в формате FAT32.

Перенос архивной информации с карты на ПК осуществляется через YartStudio.

В разделе "Локальный архив" необходимо выбрать интервал времени архива и параметры считывания.

Архивные данные считываются через интерфейс Ethernet.

При необходимости архивные данные доступны непосредственно на ПЛК по установленному IP адресу

Cerebrum YART: System Status

Archive

PLC ID: YA1004900293336511132363738
 FW Ver.: 01.04
 HW Type.: CE01
 Current time: 04:13:05 22.01.2000
 Battery Voltage: 3.03 V
 SD Card Free: 4194303 MB
 SD Free Clust: -1
 SD Clust Size: 64
 Application Name: Новый проект (ver. 1.0.0)

Cerebrum YART: Archive

Current time: 04:15:09 22.01.2000

DESC.DAT

[A_04_25_CSV](#) Date 25.04.2019; Size: 99107
[A_04_28_CSV](#) Date 28.04.2019; Size: 57051
[A_05_02_CSV](#) Date 02.05.2019; Size: 29141
[A_05_03_CSV](#) Date 03.05.2019; Size: 5863
[A_05_24_CSV](#) Date 24.05.2019; Size: 82708
[A_05_25_CSV](#) Date 25.05.2019; Size: 169797
[A_05_26_CSV](#) Date 26.05.2019; Size: 180902
[A_05_27_CSV](#) Date 27.05.2019; Size: 156620
[A_05_30_CSV](#) Date 30.05.2019; Size: 76119
[A_05_31_CSV](#) Date 31.05.2019; Size: 176345
[A_06_01_CSV](#) Date 01.06.2019; Size: 129167
[A_06_02_CSV](#) Date 02.06.2019; Size: 150245
[A_06_03_CSV](#) Date 03.06.2019; Size: 104412
[A_06_17_CSV](#) Date 17.06.2019; Size: 7435
[A_07_07_CSV](#) Date 07.07.2019; Size: 34240
[A_07_29_CSV](#) Date 29.07.2019; Size: 91301
[A_08_01_CSV](#) Date 01.08.2019; Size: 115883

Рисунок 59. Web страница контроллера с данными SD карты и архивом

Архив представляет собой список файлов, по одному файлу на каждый день в году. Скачанные с контроллера файлы могут быть расшифрованы при помощи YartStudio.

Также файлы архива доступны через Web интерфейс (Рис. 59).

Для работы с архивом через Yart Studio подключитесь к ПЛК через Ethernet и перейдите в раздел «Локальный архив». Кнопка «Выгрузить архив» запускает процесс передачи данных. В появившемся окне необходимо выбрать интересующий период времени нажать «ОК» (Рис. 60).

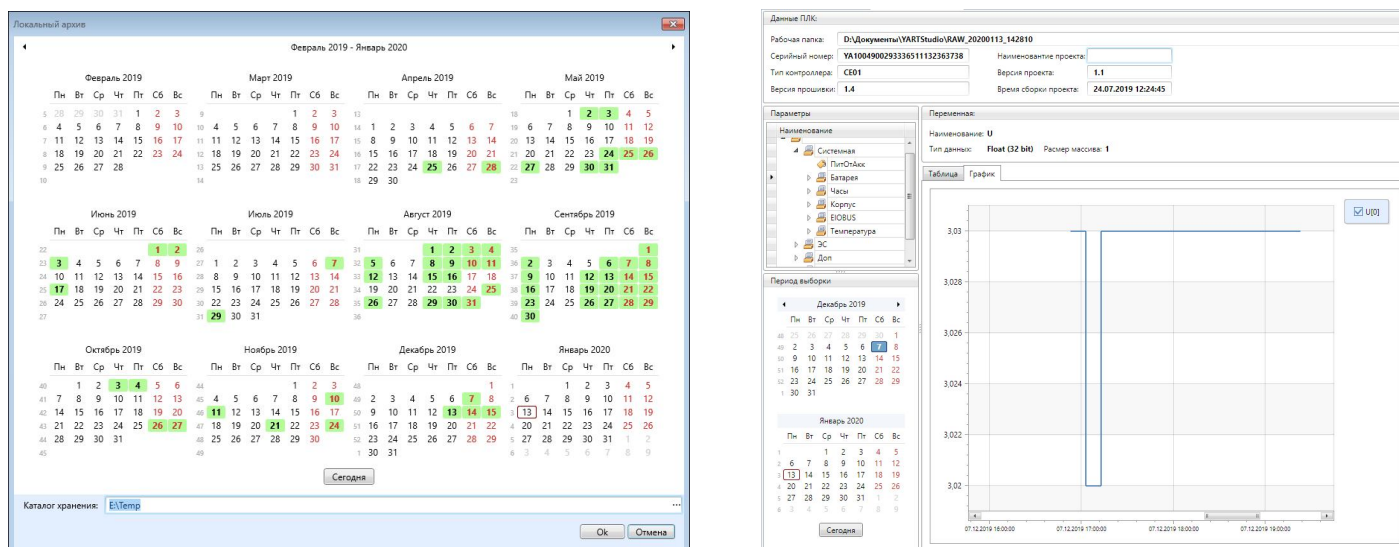


Рисунок 60. Пользовательский интерфейс архива

После завершения загрузки данные архива доступны для дальнейшей обработки. Нажмите «Обработка выгруженных данных» чтобы открыть папку с данными.



17. 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



18. 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.