

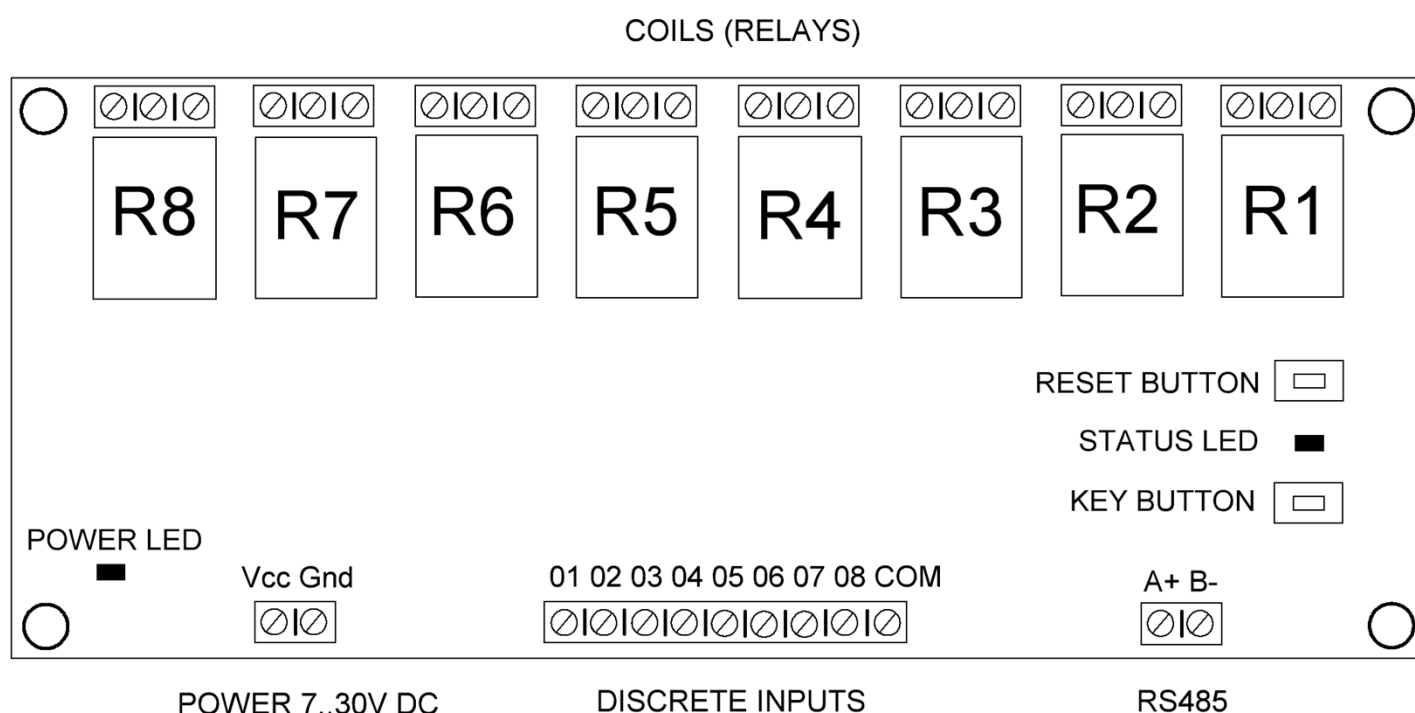
Modbus_Relay X8_V1.2

Инструкция пользователя

(альтернативная версия прошивки от VovikDu)

Modbus_Relay X8_V1.2 предназначен для использования в системах автоматизации производственных процессов в роли удалённой станции ввода-вывода с поддержкой промышленного протокола обмена данными MODBUS RTU, а также может использоваться как набор из восьми независимых программируемых реле.

Общий вид устройства изображён ниже:



Где:

- Coils (Relays)** – восемь реле. «Сухой» перекидной контакт (CO), ток активной нагрузки до 10А;
- Discrete Inputs** – восемь дискретных (логических входов). Входные цепи гальванически развязаны с общей схемой через оптопары. Входное напряжение 7..30В.
- Power** – вход питания устройства. Максимальное допустимое нестабилизированное входное напряжение постоянного тока 30 вольт.
- RS485** – вход дифференциальной пары полудуплексного стандарта связи типа RS485. Защищены от кратковременных перенапряжений в линии.



Внимание! Цепи дифференциальной пары не изолированы гальванически от контроллера. При работе устройства с длинными, а также сильно зашумленными наводками линиями используйте дополнительные средства защиты от перенапряжений и сверхтоков.

Органы индикации на плате:

Power LED – светодиод индикации наличия питающего напряжения.

Status LED – светодиод статуса. Может принимать следующие значения:

В режиме независимых программируемых реле – постоянно мигает с частотой 1 раз в секунду.

В режиме станции удалённого ввода-вывода горит во время отправки ответа на запрос.

В режиме возврата к заводским настройкам горит пока нажата кнопка сброса настроек.

Запрос состояния последних четырёх реле R5 – R8:

сетевой адрес	команда	с какого реле считать		количество реле		контрольная сумма CRC16	
		0x00	0x04	0x00	0x04	0x7C	0x08
0x01	0x01	0x00	0x04	0x00	0x04	0x7C	0x08

Ответ: (случай когда R8 включено, остальные реле выключены)

сетевой адрес	команда	байт прочитано	состояния реле	контрольная сумма CRC16	
0x01	0x01	0x01	0x08	0x50	0x4E

Таблица состояний:

бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	байт
-	-	-	-	R8	R7	R6	R5	R8 – R5
0	0	0	0	1	0	0	0	0x08

0x02 Read Discrete Inputs

Функция позволяет получить значения состояния любых из восьми дискретных входов.

Запрос состояния всех восьми дискретных входов IN1 – IN8:

сетевой адрес	команда	с какого входа считать		количество входов		контрольная сумма CRC16	
		0x00	0x00	0x00	0x08	0x79	0xCC
0x01	0x02	0x00	0x00	0x00	0x08	0x79	0xCC

Ответ: (случай когда на вход IN1 подано питание, остальные входы обесточены)

сетевой адрес	команда	байт прочитано	состояния входов	контрольная сумма CRC16	
0x01	0x02	0x01	0x01	0x60	0x48

Таблица состояний:

бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	байт
IN8	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN8 – IN1
0	0	0	0	0	0	0	1	0x01

Запрос состояния дискретного входа IN3:

сетевой адрес	команда	с какого входа считать		количество входов		контрольная сумма CRC16	
		0x00	0x02	0x00	0x01	0x18	0x0A
0x01	0x02	0x00	0x02	0x00	0x01	0x18	0x0A

Ответ: (случай когда на IN3 подано напряжение)

сетевой адрес	команда	байт прочитано	состояния входов	контрольная сумма CRC16	
0x01	0x02	0x01	0x01	0x60	0x48

Таблица состояний:

бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	байт
-	-	-	-	-	-	-	IN3	IN3
0	0	0	0	0	0	0	1	0x01

0x03 Read Holding Registers

Функция позволяет получить содержимое любых регистров памяти устройства следующих друг за другом. В регистрах хранятся все доступные настройки. Описание содержимого регистров описано ниже в таблице;

Запрос значений содержимого четырёх регистров начиная с нулевого:

сетевой адрес	команда	с какого регистра		количество регистров		контрольная сумма CRC16	
		0x00	0x00	0x00	0x04	0x44	0x09
0x01	0x03	0x00	0x00	0x00	0x04	0x44	0x09

Ответ: (содержимое регистров 0 - 3)

сетевой адрес	команда	байт прочитано	регистр №0	регистр №1	регистр №2	регистр №3	контрольная сумма CRC16	
0x01	0x03	0x08	0x0180	0x0001	0x0000	0x0080	0xE3	0xB3

номер регистра	значение по умолчанию, hex		Описание
	MSB	LSB	
0x00	MSB	0xC0	Скорость связи в бод/с x100; 192x100 = 19200 бод/сек Контроллер поддерживает скорости в диапазоне 300..921600 бод/сек
	LSB	0x00	
0x01	MSB	0x00	Старший полубайт: проверка чётности. 0-откл, 1-нечёт, 2-чёт; Младший полубайт: стоп-бит. 0-x1, 1-x0.5, 3-x1.5, 4-x2; Сетевой адрес устройства, 1..254
	LSB	0x01	
0x02	MSB	0x64	Таймаут потери связи с ведущим устройством, x10ms; 100x10ms = 1000ms (1s). значение 0 отключает контроль связи; Состояние реле при потере связи с ведущим устройством, (R8-R1);
	LSB	0x00	
0x03	MSB	0x00	Состояние дискретных входов Состояние релейных выходов
	LSB	0x00	

Настройки программируемых реле

номер регистра	значение по умолчанию, hex		Описание
	MSB	LSB	
0x04	MSB	0x00	Настройка для пары IN1-R1 Старший полубайт: единицы измерения. 0-секунды, 2-минуты, 3-часы; Младший полубайт: функция реле 0 - REPEAT, повторение сигнала на входе; 1 - TON, задержка включения реле; 2 - TOF, задержка выключения реле; 3 - TOGGLE, импульсное реле, смена значения по переднему фронту управляющего импульса; 4 - PULSE, генератор импульсов скважностью 50% и периодом T; Уставка времени, 0..255
	LSB	0x00	
0x05	MSB	0x00	Настройка для пары IN2-R2 см. описание значений выше
	LSB	0x00	
0x06	MSB	0x00	Настройка для пары IN3-R3 см. описание значений выше
	LSB	0x00	
0x07	MSB	0x00	Настройка для пары IN4-R4 см. описание значений выше
	LSB	0x00	
0x08	MSB	0x00	Настройка для пары IN5-R5 см. описание значений выше
	LSB	0x00	
0x09	MSB	0x00	Настройка для пары IN6-R6 см. описание значений выше
	LSB	0x00	
0x0A	MSB	0x00	Настройка для пары IN7-R7 см. описание значений выше
	LSB	0x00	
0x0B	MSB	0x00	Настройка для пары IN8-R8 см. описание значений выше
	LSB	0x00	



Считывая регистр №3 можно за один запрос одновременно получить значения всех дискретных входов и релейных выходов.

Содержимое всех регистров кроме третьего сохраняются в энергонезависимой памяти устройства. Таким образом абсолютно все настройки не теряются в момент отключения питания.

0x05 Write Single Coil

Функция позволяет включить/отключить одно конкретное реле.

Запрос на включение реле R5:

сетевой адрес	команда	номер реле		0xFF00 - включить	0x0000 - выключить	контрольная сумма CRC16	
0x01	0x05	0x00	0x04	0xFF	0x00	0xCD	0xFB

Ответ на включение реле R5: (повторяет запрос, эхо-ответ)

сетевой адрес	команда	номер реле		0xFF00 - включить	0x0000 - выключить	контрольная сумма CRC16	
0x01	0x05	0x00	0x04	0xFF	0x00	0xCD	0xFB

0x06 Write Single Register

Функция позволяет записать значение в конкретный регистр. Описание содержимого регистров см. выше в таблице.



Рассмотрим пример программирования реле №6:
R6 должно включиться с задержкой в 15 секунд после подачи напряжения на вход IN6.
За работу реле R6 отвечает содержимое регистра №9 (см таблицу регистров выше).
Единица измерения – секунды, первый полубайт = 0;
функция реле – TON, второй полубайт = 1;
Задержка составляет 15 секунд или **0F** в HEX.
Итого в регистр №9 необходимо записать число **0x010F**

Запрос на запись значения 0x010F в регистр №9:

сетевой адрес	команда	номер регистра	значение для записи	контрольная сумма CRC16	
0x01	0x06	0x0009	0x010F	0x18	0x5C

Ответ на запись значения 0x010F в регистр №9: (повторяет запрос, эхо-ответ)

сетевой адрес	команда	номер регистра	записанное значение	контрольная сумма CRC16	
0x01	0x06	0x0009	0x010F	0x18	0x5C



Запись значений в регистр №3 позволяет выставить значения сразу всех восьми реле. При этом в ответе старший байт будет содержать актуальные значения дискретных входов. Таким образом одним коротким запросом можно одновременно выставить все реле и получить значения дискретных входов.

Запрос на запись значения 0x0003 в регистр №3 (включить R1 и R2, остальные выключить):

сетевой адрес	команда	номер регистра	значение для записи	контрольная сумма CRC16	
0x01	0x06	0x0003	0x0003	0x39	0x5B



Обратите внимание: На этот раз ответ не идентичен запросу, так как некоторые дискретные входы под питанием. В примере ответа ниже IN2 находится под напряжением.

Ответ на запись значения 0x0003 в регистр №3 (включить R1 и R2, остальные выключить):

сетевой адрес	команда	номер регистра	записанное значение	контрольная сумма CRC16	
0x01	0x06	0x0003	0x0203	0x89	0x6A



Внимание! В случае если подобный ответ будет расцениваться ведущим устройством в сети как ошибка, следует отказаться от прямой записи значений в регистр №3, и использовать стандартные функции чтения состояния реле (0x01), чтения дискретных входов (0x02), записи состояния одного реле (0x05) и записи нескольких реле (0x0F).

0x0F Write Multiple Coil

Функция позволяет выставить значения сразу нескольких реле идущих по порядку друг за другом.

Запрос на отключение R3 и включение R4, остальные оставить без изменений:

сетевой адрес	команда	с какого реле считать		количество реле		количество байт	состояние реле	контрольная сумма CRC16	
0x01	0x0F	0x00	0x02	0x00	0x02	0x01	0x02	0x26	0x96

Ответ на отключение R3 и включение R4, остальные оставить без изменений:

сетевой адрес	команда	с какого реле считать		количество реле		контрольная сумма CRC16	
		0x00	0x02	0x00	0x02	0x75	0xCA
0x01	0x0F	0x00	0x02	0x00	0x02	0x75	0xCA

Таблица состояний:

бит 7	бит 6	бит 5	бит 4	бит 3	бит 2	бит 1	бит 0	байт
-	-	-	-	-	-	R4	R3	R4 – R3
0	0	0	0	0	0	1	0	0x02

0x11 Report Slave ID

Функция позволяет идентифицировать устройство в сети получив его vendor ID и product ID.

Запрос:

сетевой адрес	команда	контрольная сумма CRC16	
0x01	0x11	0xC0	0x2C

Ответ:

сетевой адрес	команда	количество байт	Device ID	Product ID	Статус RUN индикатора	контрольная сумма CRC16	
0x01	0x11	0x05	0x01FE	0x08C0	0xFF	0x75	0xCA



Команда «**Report Slave ID**» является сервисной, и помогает ведущему устройству сканируя шину однозначно идентифицировать всех участников сети, а также понять какой перечень функций доступен для работы с тем или иным ведомым устройством.

0x16 Mask Write Register

Функция позволяет заменить в регистре лишь конкретные биты, оставив остальные без изменений.



Регистр №1 является составным. Старший байт отвечает за проверку четности и длительности стоп-бита. Младший содержит адрес устройства. Функция «**Mask Write Register**» позволяет нам сменить сетевой адрес устройства, никак не повлияв на другие настройки. Для упрощенного понимания как работает маска достаточно запомнить два простых правила:
«1» в маске **AND** **запрещает** менять бит, **«0»** маске **AND** **разрешает** менять бит
 Маска **OR** содержит биты предназначенные к замене

Заменяем адрес устройства на 5, не меняя настройки чётности и длительности стоп-бита:

значение регистра, hex	0x11	0x01
значение регистра, bin	0001 0001	0000 0001
маска AND, bin	1111 1111	0000 0000
маска OR, bin	число не имеет значения так как этот байт защищён маской AND от изменений	0000 0101
результат, bin	0001 0001	0000 0101

Запрос:

сетевой адрес	команда	номер регистра	маска AND	маска OR	контрольная сумма CRC16	
0x01	0x16	0x0001	0xFF00	0xFF05	0x3E	0x21

Ответ: (повторяет запрос, эхо-ответ)

сетевой адрес	команда	номер регистра	маска AND	маска OR	контрольная сумма CRC16	
0x01	0x16	0x0001	0xFF00	0xFF05	0x3E	0x21