

Модули исполнительные релейные: DRM88R

Сервисная инструкция по настройке модулей

Содержание.

1. Общие сведения.
2. Настройки интерфейса.
3. Контроль и управление модулем.
4. Управление выходами модуля.
Контроль входов.
5. Информационные регистры.
6. Установка параметров конфигурации.
7. Сценарии.
8. ПО для конфигурирования.

1. Общие сведения

1.1 Протокол физического стыка – EIA/TIA-485-A (RS-485), двухпроводный, полудуплексный без гальванической развязки.

1.2 Количество бит данных по умолчанию – 8.

1.3 Количество стоповых бит по умолчанию – 2.

1.4 Бит чётности по умолчанию – отсутствует.

1.5 Скорость передачи данных по умолчанию – 9600 бит/сек.

1.6 Протокол логического обмена – «Modbus RTU».

1.7 Поддержка функций и команд обеспечивается в полном соответствии с синтаксисом запроса и ответа определенным в документе «MODBUS Application Protocol Specification v1.1b». Полное описание протокола находится на официальном сайте: ModBus.org.

1.8 Режим функционирования модуля – «Slave» (подчинённый).

1.9 Режим передачи информации – «RTU» (бинарный режим).

1.10 Используемые функции (команды) обмена информацией:

код функции	Название	Диапазон адресов	Диапазон регистров
01	Read Coils	0 - 65535	20001 - 29999
02	Read Discrete Inputs (DI)	0 - 65535	10001 - 19999
03	Read Holding Registers (HR)	0 - 65535	40001 - 49999
04	Read Input registers (IR)	0 - 65535	30001 - 39999
05	Write Single Coil	0 - 65535	20001 - 29999
06	Write Single Register (HR)	0 - 65535	30001 - 39999
15	Write Multiple Coil	0 - 65535	20001 - 29999
16	Write Multiple registers (HR)	0 - 65535	30001 - 39999

1.11 Адрес модуля – согласно протоколу MODBUS. По умолчанию все модули имеют адрес «1». Для протокола MODBUS адрес можно поменять только записью в регистр 0 другого адреса. Если адрес не известен, то запись нужно производить широковещательной командой по адресу модуля 0 в регистр 0, но при этом на шине должен быть только один модуль. Адрес устройства изменится только при перезапуске устройства.

1.12. Перевод значения регистра в единицы измерения указаны в каждой ячейке таблицы.

1.13 Тип, номер регистра и назначение регистра указаны в каждой ячейке таблицы.

1.14. Все неиспользуемые регистры возвращают фиксированные значения и не записываются.

1.15 Для групп информационных сигналов обмена выделены следующие группы данных:

- группа регистров управления;
- группа регистров настройки интерфейса;
- группа регистров конфигурации;
- группа команд внутренней логики - сценариев.

2. Настройки интерфейса.

03 Read Holding Registers (HR), 06 Write Single Register, 16 Write Multiple registers.

Эти регистры доступны для чтения и записи.

Регистр	Описание регистра	Диапазон	По умолчанию
HR0	Адрес устройства на шине ModBus RTU	1...247	1
HR1	Modbus RTU port settings [8 bit - options, 8 bit - baudrate]	0...0x55	0

2.1. Установка адреса.

Адрес можно поменять только записью в регистр HR0 другого адреса. Если адрес не известен, то запись нужно производить широковещательной командой по адресу модуля 0 в регистр HR0, но при этом на шине должен быть только один модуль.

2.2. Настройки порта Modbus RTU

Параметры можно поменять в регистре HR1. После изменения адреса, модуль нужно отключить и снова включить. Адрес устройства изменится только после перезапуска устройства.

Options:		Baudrates:	
8-N-2	0x0000	9600	0x0000
8-N-1	0x0100	19200	0x0001
8-E-2	0x0200	38400	0x0002
8-E-1	0x0300	57600	0x0003
8-O-2	0x0400	115200	0x0004
8-O-1	0x0500	230400	0x0005

Старшие 8 bit – options + младшие 8 bit – baudrate. Например, 0x0104 = четность нет, 1 стоп бит и 115200

2.3. Параметры по умолчанию:

Адрес и параметры можно сбросить по умолчанию, запустив модуль с нажатой кнопкой. Через секунду после включения кнопку можно отпустить. Кнопка находится на плате под лицевой панелью модуля. Кнопку можно использовать во время работы и задействовать её в алгоритмах работы. Состояние кнопки можно прочитать в регистре IR9019 и в регистре D10.

Параметры по умолчанию:

Адрес модуля:	1 (меняется в регистре HR0)
Скорость:	9600 бит/сек (меняется в регистре HR1)
Бит данных:	8 бит (не меняется)
Чётность:	Нет
Стоповых бит:	2

Конфигурация и сценарии работают в ОЗУ, но сохраняются во flash памяти. При включении загружаются из flash в ОЗУ. Адрес и параметры сохраняются сразу. Остальные настройки сохраняются раз в минуту. Счетчики и наработка часов сохраняются раз в 8 часов. Для принудительного сохранения из ОЗУ во flash нужно записать 1 в регистр HR92. При чтении будет возвращаться количество циклов перезаписи flash.

Регистр	Описание регистра	Запись	Чтение
HR 92	Сохранение из ОЗУ во flash	Любое значение	Счетчик записей

3. Контроль и управление модулем.

Протокол обмена данными Modbus подразумевает наличие в сети мастера, которым является контроллер и 252 подчиненных. Данные модули являются подчиненным и могут только отвечать на запросы мастера.

Данные для управления делятся на входные, полученные со входов модуля. И на выходные данные, воздействующие на выходы модуля.

Управление модулем по протоколу ModBus осуществляется чтением и записью в регистры: Coils (Co), Discrete Input (DI), Holding Registers (HR), Input Registers (IR). Далее будут использоваться сокращенные названия регистров Co, DI, HR, IR. Адреса любых регистров начинаются с 0 и заканчиваются 65535. Перечень и описание регистров указано ниже.

Данных любых регистров передаются двумя байтами. В зависимости от типов данных их максимальные значения могут быть следующие:

- Signed - знаковое целое. Максимальные значения: -32768 ... +32767;
- Unsigned - беззнаковое целое. Максимальные значения: 0 ... +65535;
- Hex - шестнадцатеричное. Максимальные значения: 0x0000 ... 0xFFFF;
- значения с запятой. Значения передаются в тысячных долях. Интерфейс передает только целые значения. Значению с запятой 1.000 соответствует 1000 в Modbus. Значению с запятой 0.001 соответствует 1 в Modbus. В контроллере это число нужно разделить на 1000. В результате получится число с тысячными долями. Значения с плавающей запятой модули не поддерживают.

4.1. Управление выходами модулей.

Для управления выходами модулей могут использоваться регистры Coils и регистры Holding.

01 Read Coils, 05 Write Single Coil.

Регистры Coils хранят состояние выхода. Эти регистры доступны для чтения и записи. Из этого регистра можно читать состояние выхода. Запись в этот регистр переключает выход.

Регистр	Диапазон данных	Назначение
Coil 1	0...1	включение канала 1
Coil 2	0...1	включение канала 2
Coil 3	0...1	включение канала 3
Coil 4	0...1	включение канала 4
Coil 5	0...1	включение канала 5
Coil 6	0...1	включение канала 6
Coil 7	0...1	включение канала 7
Coil 8	0...1	включение канала 8
Coil 9	0...1	включение канала 9
Coil 10	0...1	включение канала 10
Coil 11	0...1	включение канала 11
Coil 12	0...1	включение канала 12
Coil 13	0...1	включение канала 13
Coil 14	0...1	включение канала 14

Значение 0 соответствует отключенному состоянию реле, а значение 1 соответствует включенному состоянию реле. Coils 15 - Coils 23 – флаги, ячейки ОЗУ с дискретным значением. Могут использоваться в сценариях.

При чтении регистров Coils 24 ... Coils 65535 модуль вернет ошибку "Illegal Data Adress".

03 Read Holding Registers (HR), 06 Write Single Register, 16 Write Multiple registers.

Эти регистры доступны для чтения и записи.

Регистр	Описание регистра	Диапазон значений	После сброса
HR9	Все реле (0...255)	0...16383	0

Регистр HR9 для блока DRM88R позволяет управлять сразу несколькими выходами. Каждый бит в этом регистре управляет своим выходом. 0 бит – 1 выход реле, 1 бит – 2 выход и т.д., Например, значение HR5 = 0x01 включит 1 выход, 0x03 – включит 1 и 2 выход, 0x14 включит 5 и 4 выходы, остальные выключит.

4.2. Контроль входов.

Для контроля состояния входов используются функции *Discrete Inputs* и *Input registers*

02 Discrete Inputs (DI).

Регистры Discrete Input (DI) хранят состояние дискретных входов. Эти регистры можно только читать командами Modbus. Из этого регистра можно читать состояние дискретных входов.

Регистр	Диапазон данных	Назначение
DI 0		Кнопка
DI 1	0...1	вход 1
DI 2	0...1	вход 2
DI 3	0...1	вход 3
DI 4	0...1	вход 4
DI 5	0...1	вход 5
DI 6	0...1	вход 6
DI 7	0...1	вход 7
DI 8	0...1	вход 8
DI 9	0...1	вход 9
DI 10	0...1	вход 10
DI 11	0...1	вход 11

Значение 0 соответствует минимальному напряжению на входе, а значение 1 соответствует максимальному напряжению. Если на вход подключить кнопку, то при замыкании кнопки на общий, на входе будет минимальное напряжение и DI будет показывать 0. А при размыкании кнопки, с помощью подтягивающего резистора входное напряжение поднимется до максимального и DI покажет 1. Т.е. при нажатой кнопке – 0, при отпущенной – 1.

В регистре IR19 можно прочесть сразу все дискретные входы. Каждый вход на отдельном бите. IN1 устанавливается в бите 0, IN8 устанавливается в бите 7.

Регистр	Диапазон данных	Назначение
IR31	0...2047	Дискретные входы побитно

Входы могут работать в импульсном режиме.

- 1) Счетчик импульсов входов. Максимальное значение счетчика каждого входа 64 бита.
- 2) Частотомер импульсов на входе. Подсчитывает количество импульсов за определенный промежуток времени.

04 Input registers (IR).

Регистр	Диапазон данных	Назначение
IR 100 – IR 103 IR 140 – IR 143	0...18446744073709551616	Счетчик входов по 4 регистра на канал. Каналы 1 - 11
IR 200 IR 210	0...65535	Частотомер Каналы 1 - 11
IR 300 – IR 303 IR 340 – IR 343	0...18446744073709551616	Счетчик после пересчета по 4 регистра на канал. Каналы 1 - 11
IR 400 IR 410	0...65535	Частотомер после пересчета Каналы 1 - 11

- 3) Период времени для подсчета импульсов.

03 (0x03) Read Holding Registers (HR), 06 (0x06) Write Single Register, 16 (0x10) Write Multiple registers.

Регистр	Диапазон данных	Назначение
HR 2840 – HR 2850	100 ... 1000	Период в миллисекундах

- 4) Коэффициенты для пересчета счетчика импульсов в единицы измерения, например, в кубометры.
 5) Коэффициенты для пересчета частотомера в единицы измерения, например, кубометры в час.
 03 (0x03) Read Holding Registers (HR), 06 (0x06) Write Single Register, 16 (0x10) Write Multiple registers.

Регистр	Диапазон данных	Назначение
HR 2700 – 2710	-32768 ... +32767	Коэффициент К множитель для частотомера 1-11
HR 2720 – 2730	-32768 ... +32767	Коэффициент N делитель для частотомера 1-11
HR 2740 – 2750	-32768 ... +32767	Коэффициент N делитель для счетчика 1-11
HR 2760 – 2803	0...18446744073709551616	Смещение для счетчика 1-11

- 6) Обнуление всех счетчиков. Для обнуления необходимо записать в регистр HR 2860 значение 2860.

Регистр	Диапазон данных	Назначение
HR 2860	2860	Обнуление всех счетчиков

Что бы входы считали нужно отключать АЦП этих входов.

Регистр	Диапазон данных	Назначение
HR 5	0 ... 1023	Отключение каждого АЦП

04 Input registers (IR).

Регистры Input registers (IR) хранят состояние аналоговых входов. Эти регистры можно только читать командами Modbus. Каждый вход опрашиваются микросхемой АЦП 12бит, которая выдает значение от 0 до 4096.

При необходимости значения АЦП можно преобразовать по формуле: $X = \frac{ADC * K}{N} + B$; Результат расчета помещается в регистры IR11 – IR18.

Регистр	Диапазон данных	Назначение
IR 0	-	-
IR 1	0...4096	АЦП 1
IR 2	0...4096	АЦП 2
IR 3	0...4096	АЦП 3
IR 4	0...4096	АЦП 4
IR 5	0...4096	АЦП 5
IR 6	0...4096	АЦП 6
IR 7	0...4096	АЦП 7
IR 8	0...4096	АЦП 8
IR 9	0...4096	АЦП 9
IR 10	0...4096	АЦП 10
IR11	-32768 +32767	ADC1*K/N+B
IR12	-32768 +32767	ADC2*K/N+B
IR13	-32768 +32767	ADC3*K/N+B
IR14	-32768 +32767	ADC4*K/N+B
IR15	-32768 +32767	ADC5*K/N+B
IR16	-32768 +32767	ADC6*K/N+B
IR17	-32768 +32767	ADC7*K/N+B
IR18	-32768 +32767	ADC8*K/N+B

IR19	-32768 +32767	ADC9*K/N+B
IR20	-32768 +32767	ADC10*K/N+B
IR31	0...1023	Дискретные входы побитно
IR32 ... IR999	-32768 +32767	переменные

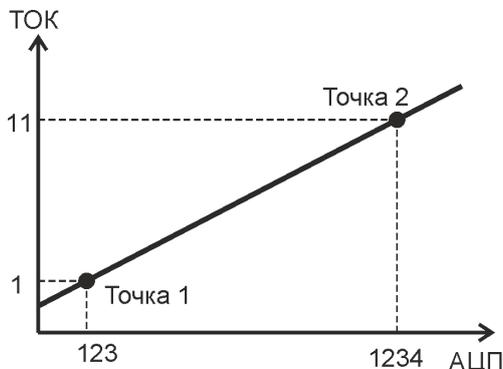
Каждый вход внутри имеет подтягивающий резистор 22кОм к напряжению +5В. Затем через 22кОм приходит на ножку микросхемы АЦП. Модуль будет реагировать на любое входное напряжение от 0 до 5В для входов 1...8 и от 0 до 3,3В для входов 9 и 10. Вход 11 измерять аналоговое значение не может, он только дискретный. После измерения входного сигнала АЦП, в сценариях можно установить любой уровень срабатывания входного сигнала. Входы имеют защиту от превышения напряжения до 25В. Можно подать напряжение от 5 до 25В. Вход будет всегда показывать максимум, но блок не сгорит. Выше 25В на вход блока подавать нельзя.

4.3. Коэффициенты аналоговых входов.

Регистр	Диапазон данных	Назначение
HR40	-32768 +32767	Канал 1, К
HR41	-32768 +32767	Канал 1, N
HR42	-32768 +32767	Канал 1, В
HR43	-32768 +32767	Канал 2, К
HR44	-32768 +32767	Канал 2, N
HR45	-32768 +32767	Канал 2, В
HR46	-32768 +32767	Канал 3, К
HR47	-32768 +32767	Канал 3, N
HR48	-32768 +32767	Канал 3, В
HR49	-32768 +32767	Канал 4, К
HR50	-32768 +32767	Канал 4, N
HR51	-32768 +32767	Канал 4, В
HR52	-32768 +32767	Канал 5, К
HR53	-32768 +32767	Канал 5, N
HR54	-32768 +32767	Канал 5, В
HR55	-32768 +32767	Канал 6, К
HR56	-32768 +32767	Канал 6, N
HR57	-32768 +32767	Канал 6, В
HR58	-32768 +32767	Канал 7, К
HR59	-32768 +32767	Канал 7, N
HR60	-32768 +32767	Канал 7, В
HR61	-32768 +32767	Канал 8, К
HR62	-32768 +32767	Канал 8, N
HR63	-32768 +32767	Канал 8, В
HR64	-32768 +32767	Канал 9, К
HR65	-32768 +32767	Канал 9, N
HR66	-32768 +32767	Канал 9, В
HR67	-32768 +32767	Канал 10, К
HR68	-32768 +32767	Канал 10, N
HR69	-32768 +32767	Канал 10, В
HR70	0...8	Калькулятор: канал
HR71	-32768 +32767	Калькулятор: Параметр 1
HR72	-32768 +32767	Калькулятор: Параметр 2
HR73	-32768 +32767	Калькулятор: АЦП 1
HR74	-32768 +32767	Калькулятор: АЦП 2
HR75	1...255	усреднение АЦП

Полученные значения АЦП можно преобразовать по формуле: $X = \frac{ADC * K}{N} + B$; Результат расчета помещается в регистры IR11 – IR20. Коэффициенты хранятся в регистрах HR40 – HR69. Для расчета этих коэффициентов формулы уравнения прямой необходимо использовать две точки.

Точки измерения могут быть любые. Наклон линии может быть любой: вниз, вверх, в плюс или в минус. Значение АЦП и входного напряжения так же может быть любое как в плюс, так и в минус.



Коэффициенты рассчитываются по формуле.

$K = \text{Температура}2 - \text{Температура}1$;

$N = ADC2 - ADC1$;

$B = \frac{ADC1 * \text{Температура}2 - ADC2 * \text{Температура}1}{ADC1 - ADC2}$;

В блоках добавлен калькулятор для автоматического расчета этих коэффициентов. В регистрах HR70 – HR74.

03 Read Holding Registers (HR), 06 Write Single Register.

Регистр	Адрес	Диапазон	Описание регистра
HR 70	40071	1...10	номер канала
HR 71	40072	-32768...32767	Параметр 1
HR 72	40073	-32768...32767	Параметр 2 (запись в этот регистр запускает расчет и сохранение коэффициентов в указанный HR70 канал)
HR 73	40074	-32768...32767	результат АЦП 1 (только чтение)
HR 74	40075	-32768...32767	результат АЦП 2 (только чтение)

Последовательность действий следующая.

- 1) подключить датчик.
- 2) в регистр HR70 записать номер канала (1 ... 10), к которому подключен датчик.
- 3) установить датчик в калибровочную камеру.
- 4) после стабилизации значений вписать значение первого параметра в регистр HR71 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для первого параметра в регистр HR73.
- 5) изменить величину климатического параметра.
- 6) после стабилизации значений вписать значение второго параметра в регистр HR72 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для второго параметра в регистр HR74. Затем модуль рассчитает коэффициенты и перепишет эти параметры в регистры коэффициентов номера канала, указанного в HR70. После этого в регистрах IR11 ... IR20 будут выводиться значения в заданных физических величинах.

Для повышения точности показаний нужно, чтобы диапазон изменения физической величины был в максимальном диапазоне АЦП от 0 до 4095. Для разных типов датчиков на входах модуля могут быть запаяны разные элементы с разными номиналами. Вход может быть настроен для измерения напряжения, сопротивления или тока. По умолчанию блок настроен на измерение напряжения.

Регистр HR75 – Усреднение АЦП используется для уменьшения шумов и увеличения точности показаний. Значение может быть от 1 до 255. Полученное значение АЦП складывается указанное количество раз и делится на это количество. Скорость измерения уменьшается в это же количество раз.

5. Информационные регистры.

04 Input registers (IR).

Регистры *Input registers (IR)* хранят информацию о модуле. Эти регистры можно только читать командами Modbus.

Информационные регистры служат для идентификации модуля и контроля внутреннего состояния.

<i>Регистр</i>	<i>Диапазон данных</i>	<i>Назначение</i>
IR 9000	0...65535	номер ревизии ПО
IR 9001	0...65535	номер ревизии ПО
IR 9002	0...1	Версия ПО Release -0 / Debug -1
IR 9003	0...255	Тип устройства: 11 (DRM88Rv1), 17 (DRM88Rv2), 7 (DRM88Rv3v4)
IR 9004	1...31	Дата: день месяца
IR 9005	1...7	Дата: неделя
IR 9006	1...12	Дата: месяц
IR 9007	0...99	Дата: год
IR 9008	0...23	Время: часы
IR 9009	0...59	Время: минуты
IR 9010	0...59	Время: секунды
IR 9011	0...65535	Серийный номер
IR 9012	0...65535	Серийный номер
IR 9013	0...65535	Серийный номер
IR 9014	0...65535	Серийный номер
IR 9015	0...65535	Серийный номер
IR 9016	0...65535	Серийный номер
IR 9017	-	Детектор перехода нуля
IR 9018	-	-
IR 9019	0...1	Состояние кнопки
IR 9020	0...65535	Случайное число
IR 9021	0...65535	Счетчик наработки часов
IR 9022	0...65535	Циклы перезаписи Flash
IR 9023	0...65535	Количество переключений реле ~I
IR 9024	0...65535	Количество переключений реле =I
IR 9025	0...65535	Размер файла конфигурации, байт

В регистрах дата и время хранится текущее состояние часов. Регистры часов можно использовать как для контроля, так и для сценариев.

6. Установка параметров конфигурации.

Параметры устанавливаются в: 03 Read Holding Registers (HR), 06 Write Single Register, 16 Write Multiple registers. Эти регистры доступны для чтения и записи.

6.1. Установка даты и времени.

Регистр	Диапазон	Описание регистра
HR 93	1...31	Установка даты - День месяца
HR 94	1...7	Установка даты - День недели
HR 95	1...12	Установка даты - Месяц
HR 96	2018...2118	Установка даты - Год, 2019 или 19
HR 97	0...23	Установка времени - Часы
HR 98	0...59	Установка времени - Минуты
HR 99	0...59	Установка времени - Секунды

В регистры HR94 – HR99 можно установить новое значение даты и времени. Для установки даты и времени необходимо записать в регистр HR99 новое значение. При отключении питания часы сбросятся.

Прочитать текущее время и дату можно из регистров: IR9004 – IR9010.

6.2. Коррекция часов.

Регистр	Диапазон	Описание регистра
HR 4	-127...0...127	Коррекция работы часов

Часы за синхронизированы от внешнего кварцевого генератора. Часы можно немного замедлить или ускорить, записав в HR4 значение с минусом или плюсом.

6.3. Настройки входов.

Регистр	Диапазон	Описание регистра	После сброса
HR 5	0...1023	Отключение АЦП поканально. 0 – все включено	255
HR 6	-4000...4000	Порог преобразования АЦП в дискретный	1000
HR 7	0...255	Замедление счетчика	10

Поскольку АЦП работает медленнее, чем дискретные входы, то для работы счетных входов нужно отключить АЦП. Каждый вход можно отключить независимо от других. Младший бит отключает АЦП 1, старший бит отключает АЦП 8. 0 - АЦП включен, 1 – АЦП отключен.

Значение АЦП преобразуется в дискретные значения и выводятся в регистры IR 21 – IR 28. Можно задать пороговое значение в регистре HR 6. Оптимальное значение 1000. Если значение будет с минусом, тогда результат будет инверсный. Гистерезис нулевой. В этом режиме должен быть включен АЦП для заданного канала.

Счетчик работает с частотой до 750 Гц. На такой частоте будет сильно замечен дребезг контактов. Чтобы не было дребезга контактов и не было ложных срабатываний нужно уменьшить частоту сигнала. Для этого нужно подключить конденсаторы на входы или установить замедление счетчика в регистре HR7. При значении 0 частота 750Гц, при значении 80 частота примерно 10Гц.

Регистр	Диапазон	Описание регистра	После сброса
HR 2691	0...3	Тип входа	1

В этом регистре можно установить взаимосвязь входов и выходов.

0 – Входы и выходы не зависимы, сценарии отключены.

1 – Сценарии включены. Взаимодействие входов и выходов через сценарии.

2 – Все входы настроены как кнопки. При четном нажатии включается соответствующий канал, при нечетном выключается. Для переключения контролируется выход. Первый вход переключает первый выход, восьмой вход переключает восьмой выход. Связь жесткая и поменять нельзя.

3 – Все входы настроены как переключатели. При замыкании входа включается соответствующий канал, при размыкании входа выключается выход. Связь входа с выходом жесткая и поменять нельзя.

6.4. Перезагрузка.

Регистр	Диапазон	Описание регистра
HR 5678	-	Переагрузка модуля

Запись любого числа в регистр HR5678 произведет перезагрузку модуля.

6.5. Сохранение конфигурации.

Сохранение конфигурации производится из ОЗУ в Flash память микроконтроллера. При включении питания данные из Flash записываются в ОЗУ. Запись во Flash производится один раз в минуту или принудительно командой записи в регистр HR92 значения 1. При чтении регистра HR92 модуль будет отдавать количество циклов записи во Flash.

Регистр	Диапазон	функция	Описание регистра
HR 92	-	Запись	Запись во Flash
HR 92	0-65535	чтение	Количество циклов записи

Количество циклов записи во Flash ограничено значением 10000. Поэтому запись нужно производить после завершения редактирования всех сценариев.

Состояние реле сохраняются в регистры Backup. Это ячейки ОЗУ, питающиеся от батарейки. При условии, что батарейка подключена, при включении питания восстановятся ранее установленные состояния. При использовании батарейки часы останавливаются, но сохраняют предыдущее значение времени и даты.

6.6. Доступ к данным.

Эти регистры доступны для чтения и записи.

Регистр	Диапазон и тип данных	Адрес	Назначение
HR100 – HR1652	Сценарии	40101	Сценарии
HR2980 - HR2995	0 ... +32767	42981	Доступ к Timer 0 – Timer15
HR3000 - HR3999	-32768 +32767	43001	Доступ к IR0 – IR999

7. Сценарии для внутренней логики.

Для автономной работы блока без контроллера можно использовать встроенные сценарии. Самостоятельно посылать команды в сеть модули не могут. Поэтому описанные ниже сценарии могут работать только внутри самого модуля, используя собственные входы, выходы, таймеры, часы реального времени и регистры хранения.

С регистра HR100 записываются данные для сценариев. В модуле зарезервировано место для 64 сценариев. Для каждого сценария используется по 13 регистров R0 – R12. Шаг записи сценариев 20.

Для хранения промежуточных данных используются регистры переменных. Значения регистров находятся в области регистров Input registers (IR) из сценариев можно как читать, так и записывать. По протоколу Modbus эти регистры можно только читать. Для чтения и записи доступно 1000 регистров с адресами от IR0 до IR999. Эти данные представляют собой массив ОЗУ и не сохраняются во FLASH.

Для чтения и записи сценариев используются регистры Holding Registers (HR). Эти регистры доступны для чтения и записи. Часть регистров зафиксированы под определенные параметры. С адреса 100 до адреса 1379 зарезервировано место для записи сценариев с 0 по 63. Все указанные в таблице значения сохраняются во FLASH. Все неиспользуемые адреса не записываются и не сохраняются.

Сценарии внутри блока выполняются последовательно от 0 до 63. Затем циклично повторяется с нулевого сценария. При большом количестве сценариев может ощущаться задержка. Если сценариев используется мало, то последним сценарием можно использовать переход в начало GOTO 0. Или пропустить пустые сценарии этой же командой.

03 (0x03) Read Holding Registers (HR), 06 (0x06) Write Single Register, 16 (0x10) Write Multiple registers.

Адрес							Описание регистра
R0	R1	R2	R3	R4	...	R12	
HR100	HR101	HR102	HR103	HR104	...	HR112	Сценарий № 0 и его параметры
HR120	HR121	HR122	HR123	HR124	...	HR132	Сценарий № 1 и его параметры
HR140	HR141	HR142	HR143	HR144	...	HR152	Сценарий № 2 и его параметры
HR160	HR161	HR162	HR163	HR164	...	HR172	Сценарий № 3 и его параметры
HR180	HR181	HR182	HR183	HR184	...	HR192	Сценарий № 4 и его параметры
...
HR1360	HR1361	HR1362	HR1363	HR1364	...	HR1372	Сценарий № 63 и его параметры

Источники данных.

Сценарии могут работать с входными и выходными источниками данных. Источник данных может быть разного типа и записывается он в ячейку «Тип регистра».

Значение	Тип данных	диапазон	Тип данных	Чтение запись
0	Const – константа, фиксированное число.	(-32768 ... +32767)	Int (-32768 ... +32767)	Только чтение.
1	Coils (Co) – 01 регистр Реле.	(0 ... 4)	Bool (0...1)	Чтение и запись.
2	Discrete Input (DI) – 02 регистр дискретных входов.	(0 ... 8)	Bool (0...1)	Только чтение.
3	Holding Reg (HR) – 03 регистр параметров.	(0 ... 2660)	Int (-32768 ... +32767)	Чтение и запись.
4	Input Reg (IR) – 04 регистр аналоговых входов.	(0...9030)	Int (-32768 ... +32767)	Чтение и запись.
5	Timer – регистр таймера обратного отсчета.	(0...15)	Int (-32768 ... +32767)	Чтение и запись.

6	PC счетчик команд для перехода на другой сценарий. Используются команды GOTO, CALL	(0 ... 77)	Int (-32768 ... +32767)	Только запись.
---	--	------------	-------------------------	----------------

Типы данных.

Большинство данных использует тип *int16* это двухбайтовое 16 битное число со знаком. Диапазон значений -32768 ... +32767. При работе с логикой или дискретными входами, выходами реле используется значение *bool – false* (0) или *true* (1). При переводе из типа *int* значение 0 будет переводиться в *false* (0), любое другое значение, отличное от 0 будет переводиться в *true* (1). Другие типы данных, например, символьные значения или значения с плавающей запятой модули не используют.

Пользовательские данные.

Большинство команд используют входные данные и результат помещают в выходные данные. Эти данные могут быть как физические входы или выходы блока, так и пользовательские данные. Пользовательские данные могут использоваться как переменные для промежуточных расчетов.

Эти данные делятся на два типа и располагаются в двух областях:

1) Регистры Input registers (IR). Для чтения и записи доступно 1000 регистров с адресами от IR0 до IR999. Эти данные представляют собой массив ОЗУ и не сохраняются во FLASH. При отключении питания сбрасываются в 0. В эти регистры можно записывать массивы статистических данных.

2) Для чтения и записи настроек используются регистры Holding Registers (HR). Используется диапазон данных сценариев. В области сценариев нулевой регистр должен быть 0, а остальные 12 регистров могут быть любые. В эту область R1 – R12 можно записывать значения настроек. Эти регистры доступны для чтения и записи. Эти регистры записываются во FLASH и при повторном включении питания восстанавливаются.

Значение	Тип данных	Размещение	Тип данных	Чтение запись
IR0 ... IR999	пользовательские данные	ОЗУ не сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.
HR100... HR2000	Данные сценариев	FLASH сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.

Таймеры.

Timer – регистр таймера обратного отсчета. В этом регистре каждую 0,1 секунду значение уменьшается на 1. После того, как таймер доходит до 0 счет останавливается. Доступно 16 таймеров (0 ... 15). Доступно чтение и запись. Разрядность таймера 16 бит, значение таймера от 0 до 65535.

Номер	Таймер
0	Таймер 0
1	Таймер 1
2	Таймер 2
3	Таймер 3
4	Таймер 4
5	Таймер 5
6	Таймер 6
7	Таймер 7

Номер	Таймер
8	Таймер 8
9	Таймер 9
10	Таймер 10
11	Таймер 11
12	Таймер 12
13	Таймер 13
14	Таймер 14
15	Таймер 15

Сценарии.

Доступны сценарии №0 ... №63. В каждом сценарии в адресе от 100 до 112 записываются тип и параметры сценария. Далее адрес будет обозначаться R0 – тип, записанный в регистр 100 (для сценария 0), R1 – параметр 1, записанный в регистр 101, R2 – параметр 2, записанный в регистр 102 и так далее.

Регистр R0 – Тип сценария

<i>Значение</i>	<i>Обозначение</i>	<i>Описание</i>
0	NOP	Пустая команда. Не производит никаких действий.
1	MATH	Целочисленные арифметические и битовые операции: 0 – “=” – равно; 1 - “~” – инверсия; 2 - “++” – инкремент; 3 - “--” – декремент; 4 - “+” – сложение; 5 - “-” – вычитание; 6 - “*” – умножение; 7 - “/” – деление; 8 - “%” – остаток от деления; 9 - “+=” – сложение с предыдущим; 10 - “-=” – вычитание из предыдущего; 11 - “&” – бинарная И; 12 - “ ” – бинарная ИЛИ; 13 - “^” – бинарная ИСКЛЮЧАЮЩЕЕ ИЛИ; 14 - “<<” – побитовый сдвиг влево; 15 - “>>” – побитовый сдвиг вправо; 16 - “~=” – бинарная инверсия; 17 - “POW” – X в степени Y; 18 - “SQRT” – квадратный корень; 19 - “MIN” – выбор минимального значения; 20 - “MED” – расчет среднего значения; 21 - “MAX” – выбор максимального значения; 22 - “LIMIT” – ограничение в указанном диапазоне; 23 - “ABS” – абсолютное значение.
2	FLOAT	Математические операции с дробными значениями: 1 – SIN – Синус; 2 - COS – Косинус; 3 - TAN – Тангенс; 4 - ASIN – Арксинус; 5 - ACOS – Арккосинус; 6 - ATAN – Арктангенс; 7 - SINH - Синус гиперболический; 8 - COSH - Косинус гиперболический; 9 - TANH - Тангенс гиперболический; 10 - EXP – Экспонента; 11 - LN - Логарифм натуральный; 12 - LOG - Логарифм десятичный; 13 - POW - Y в степени X; 14 - SQRT - Корень квадратный.
3	PTRW	Указатель для записи массивов переменных.
4	PTRR	Указатель для чтения массивов переменных.

5	FOR	Создание циклов.
6	SEL	Бинарный выбор, мультиплексор. Возвращает К-е значение из входных переменных.
7	IF	Логические операции IF (R2 условие R5) с функцией присвоения и переходом: 0 – “==” – если равно; 1 – “!=” – если не равно; 2 – “>” – если больше; 3 – “<” – если меньше; 4 – “>=” – если больше или равно; 5 – “<=” – если меньше или равно; 6 – “!” – если не верно, false; 7 – “&&” – если true оба операнда; 8 – “ ” – если true один из операндов.
8	IFAND	Логические операции IF двойная с И if((R1.2 R3 Const R4)&&(R5.6 R7 Const R8)) then R9.10 = R11.12 0 – “==” – если равно; 1 – “!=” – если не равно; 2 – “>” – если больше; 3 – “<” – если меньше; 4 – “>=” – если больше или равно; 5 – “<=” – если меньше или равно; 6 – “!” – если не верно, false; 7 – “&&” – если true оба операнда; 8 – “ ” – если true один из операндов.
9	IFOR	Логические операции IF двойная с ИЛИ if((R1.2 R3 Const R4) (R5.6 R7 Const R8)) then R9.10 = R11.12 0 – “==” – если равно; 1 – “!=” – если не равно; 2 – “>” – если больше; 3 – “<” – если меньше; 4 – “>=” – если больше или равно; 5 – “<=” – если меньше или равно; 6 – “!” – если не верно, false; 7 – “&&” – если true оба операнда; 8 – “ ” – если true один из операндов.
10	GOTO	Переход.
11	CALL	Переход в подпрограмму.
12	RETURN	Выход из подпрограммы.
13	THRS	Пороговое реле.
14	TRG	Триггер.
15	KEY_DO	Кнопка двойная без переключения.
16	KEY_DB	Кнопка двойная с переключением.
17	TIMER	Таймер.
18	RTC	Часы и дата.
19	JAL	Управление Жалюзи.
20	PID	ПИД регулятор.
21 - 65535	NOP	Пустая команда. Не производит никаких действий.

MATH - Арифметические операции R2 = R4 (операция) R7:

Сценарий производит арифметические действия над двумя входными операндами и присваивает к выходному результату.

Регистр	R0	R1	R2	R3	R4	R5	R6	R7
Параметр	1 MATH	Тип	Рег результата	Тип	операнд 1	Операция	Тип	операнд 2
Диапазон	1	1, 3, 4, 5	0...2660	0 - 5	-32768 ... +32767	0 - 23	0 - 5	-32768 ... +32767

Входные данные: R3, R4, R6, R7; Выходные данные: R1, R2; Операция: R5;

Логика работы сценария «арифметическая операция» следующая: R1.2 = R3.4 (операция) R6.7;

Регистр R5 - Операция:

значение	Операция	Действия
0	"=" – равно	R1.2 = R3.4
1	"~" – инверсия	R1.2 = ~R3.4
2	"++" – инкремент	R1.2 = R1.2 + 1
3	"--" – декремент	R1.2 = R1.2 - 1
4	"+" – сложение	R1.2 = R3.4 + R6.7
5	"-" – вычитание	R1.2 = R3.4 - R6.7
6	"*" – умножение	R1.2 = R3.4 * R6.7
7	"/" – деление	R1.2 = R3.4 / R6.7 целое число
8	"%" – остаток от деления	R1.2 = R3.4 % R6.7 остаток
9	"+=" – сложение с предыдущим	R1.2 = R1.2 + R3.4
10	"-=" – вычитание из предыдущего	R1.2 = R1.2 - R3.4
11	"&" – бинарная И	R1.2 = R3.4 & R6.7
12	" " – бинарная ИЛИ	R1.2 = R3.4 R6.7
13	"^" – бинарная ИСКЛЮЧАЮЩЕЕ ИЛИ	R1.2 = R3.4 ^ R6.7
14	"<<" – побитовый сдвиг влево	R1.2 = R3.4 << R6.7(количество бит)
15	">>" – побитовый сдвиг вправо	R1.2 = R3.4 >> R6.7(количество бит)
16	"~=" – бинарная инверсия	R1.2 = R3.4 = (0xFFFF-R6.7+2)
17	"POW" – X в степени Y	R1.2 = R3.4 в степени R6.7
18	"SQRT" – квадратный корень	R1.2 = квадратный корень из R3.4
19	"MIN" – выбор минимального значения	R1.2 = (R3.4 > R6.7) ? R6.7 : R3.4
20	"MED" – расчет среднего значения	R1.2 = (R3.4 + R6.7) / 2
21	"MAX" – выбор максимального значения	R1.2 = (R3.4 > R6.7) ? R3.4 : R6.7
22	"LIMIT" – ограничение в диапазоне	R1.2 = R3.4 [R1.2] R6.7
23	"ABS" – абсолютное значение	R1.2 = abs(R3.4), например, abs(-2)=2

Сохранение результата в константу и Discrete Inputs невозможно.

Формат записи: MATH (R1) (R2) = (R3) (R4) (R5) (R6) (R7)

Пример записи: **MATH HR 20 = IR 10 + HR 30**

FLOAT - Математические операции с дробными значениями

Сценарий производит математические операции с дробными значениями над двумя входными операндами и присваивает к выходному результату.

Регистр	R0	R1	R2	R3	R4	R5	R6	R7
Параметр	1 MATH	Тип	Рег результата	Тип	операнд 1	Операция	Тип	операнд 2
Диапазон	1	1, 3, 4, 5	0...2660	0 - 5	-32768 ... +32767	0 - 14	0 - 5	-32768 ... +32767

Входные данные: R3, R4, R6, R7; Выходные данные: R1, R2; Операция: R5;

Сценарий рассчитывает тригонометрическую функцию угла в радианах. Сценарий работает со значениями меньше единицы. Поскольку интерфейс передает только целочисленные значения, то передаваемые значения делятся на 1000.

Для значения 1.0 необходимо ввести 1000. Результат так же увеличен на 1000. Значения вводятся в Радианах. Например, для 90 градусов будет 1,5708 радиан. SIN(1.571 радиан) = 1.000. В регистр R4 необходимо ввести 1571. В регистре R2 будет значение 1000

$\pi * 1$ радиан = 3,14*1 радиан. Поскольку 1 радиан = 57°17'44,8" (57 градусов 17 минут 44,8 секунд) , это означает $3,14 * 57^{\circ}17'44,8'' = 180^{\circ}$

Логика работы сценария «арифметическая операция» следующая: R1.2 = SIN (R3.4);

Регистр R5 - Операция:

значение	Операция	Действия
0	"="	R1.2 = R6.7
1	SIN – Синус	R1.2 = SIN (R3.4)
2	COS – Косинус	R1.2 = COS (R3.4)
3	TAN – Тангенс	R1.2 = TAN (R3.4)
4	ASIN – Арксинус	R1.2 = ASIN (R3.4)
5	ACOS – Арккосинус	R1.2 = ACOS (R3.4)
6	ATAN – Арктангенс	R1.2 = ATAN (R3.4)
7	SINH - Синус гиперболический	R1.2 = SINH (R3.4)
8	COSH - Косинус гиперболический	R1.2 = COSH (R3.4)
9	TANH - Тангенс гиперболический	R1.2 = TANH (R3.4)
10	EXP – Экспонента	R1.2 = EXP (R3.4)
11	LN - Логарифм натуральный	R1.2 = LN (R3.4)
12	LOG - Логарифм десятичный	R1.2 = LOG (R3.4)
13	POW - Y в степени X	R1.2 = R3.4 в степени R6.7
14	SQRT - Корень квадратный	R1.2 = квадратный корень из R3.4

Формат записи: FLOAT (R1) (R2) = (R3) (R4) (R5) (R6) (R7)

Пример записи: **FLOAT HR 20 = SIN IR 10**

PTRW – указатель для записи массивов переменных

Сценарий позволяет записывать данные в массив данных.

регистр	Описание
R0	(TypeR1)[(TypeR2)R3] = (TypeR4)R5
R1	Тип вых регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Тип регистра указателя: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Номер регистр
R4	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R5	Входной операнд

Входные данные: R4, R5;

Выходные данные: R1, R2, R3;

Указатель используется для создания массивов. Можно записывать в регистр с номером, указанным в другом регистре. Например, IR[IR23]=IR86, в регистре IR23 указывается номер регистра в массиве.

Формат записи: PTR (R1) [(R2.3)] = (R4.5)

Пример записи: 00: PTR IR [HR 7] = HR 15

Сценарий	R0	R1	R2	R3	R4	R5
параметр	PTRW	Тип вых регистра	Тип указателя	регистр	Тип вх регистра	Вх регистр
Значение	3	1, 3, 4, 5	0 - 5	-32768 ... +32767	0 - 5	-32768 ... +32767

PTRR – указатель для чтения массивов переменных

Сценарий позволяет читать данные из массива данных.

регистр	параметр
R0	4 – (TypeR4)R5 = (TypeR1)[(TypeR2)R3]
R1	Тип входного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Тип регистра указателя: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Номер регистр
R4	Тип выходного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R5	Выходной операнд

Выходные данные: R4, R5;

Входные данные: R1, R2, R3;

Указатель используется для создания массивов. Можно читать их регистра с номером, указанным в другом регистре. Например, IR86 = IR[IR23], в регистре IR23 указывается номер регистра в массиве.

Формат записи: PTRS (R4.5) = (R1) [(R2.3)]

Пример записи: 00: **PTRS HR 15 = IR [HR 7]**

Сценарий	R0	R1	R2	R3	R4	R5
параметр	PTRR	Тип вх рег	Тип указателя	Вх регистр	Тип	Вых регистр
Значение	4	1-Coils 3-HR 4-IR 5-Tim	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767
Указатель		Тип	Тип	Вх регистр	Тип	Вых регистр

FOR – циклы

Сценарий позволяет создать циклы.

регистр	Параметр
R0	5 – FOR (IR1=R3; IR1<=R4; IR1++) GOTO R5
R1	Регистр цикла (всегда IR)
R2	Тип регистра значений начала и конца цикла: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Регистр начала цикла
R4	Регистр конца цикла
R5	Адрес выхода из цикла

Входные данные: R1, R2, R3, R4;

Выходные данные: R5;

R0 5 – FOR (IR1=R3; IR1<=R4; IR1++) GOTO R5

Сценарий позволяет зациклить часть сценариев в указанном диапазоне изменения переменной. Регистр R1 будет прибавляться на единицу (инкрементироваться) в диапазоне от указанного в R3 до указанного в R4. Пока значение R1 внутри диапазона, то будут выполняться следующие за этим сценарием команды.

В конце сценариев необходимо добавить команду перехода GOTO в начало цикла. Например,

01: FOR (IR35=3; IR35<=7; IR35++) GOTO 05; цикл от 3 до 7

02: MATH IR55 = IR55+25; операции внутри цикла, прибавление значения

03: PTRW IR[IR35] = IR55; операции внутри цикла, заполнение массива

04: GOTO 01; переход в начало цикла

05: IR29=IR28; следующая команда после окончания выполнения цикла

Формат записи: FOR (IR (R1) = (R2.3) TO (R2.4)) GOTO (R6)

Пример записи: 00: **FOR (IR 20 = HR 30 TO HR 40) GOTO 98**

Сценарий	R0	R1	R2	R3	R4	R5
		Тип IR	выбор	Тип в пар 2	Тип в пар 2	выбор
5 – FOR		Число: 0...2660	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	Число: -32768 ... +32767	Число: -32768 ... +32767
Цикл		Регистр цикла	Тип	начало	конец	Адр выхода

SEL – Бинарный выбор. Мультиплексор. Тернарный оператор.

Сценарий присваивает к выходному регистру один из двух, трех или четырех входных регистров по указанному номеру.

Регистр	Параметр
R0	6 SEL - Бинарный выбор.
R1	Тип выходного регистра: 3-Holding Reg, 4-Input Reg
R2	Выходной регистр
R3	Тип регистра выбора: 1-Coils, 2-Discrete Input, 3-Holding Reg, 4-Input Reg
R4	Регистр выбора: 0 или 1 или 2 или 3
R5	Тип входного регистра: 0-Const, 1-Coils, 2-Discrete Input, 3-Holding Reg, 4-Input Reg, 5-Timer
R6	Входной регистр при 0
R7	Тип входного регистра: 0-Const, 1-Coils, 2-Discrete Input, 3-Holding Reg, 4-Input Reg, 5-Timer
R8	Входной регистр при 1
R9	Тип входного регистра: 0-Const, 1-Coils, 2-Discrete Input, 3-Holding Reg, 4-Input Reg, 5-Timer
R10	Входной регистр при 2
R11	Тип входного регистра: 0-Const, 1-Coils, 2-Discrete Input, 3-Holding Reg, 4-Input Reg, 5-Timer
R12	Входной регистр при 3

Выходные данные: R1, R2;

Условие: R3, R4;

Входные данные: R5, R6, R7, R8, R9, R10, R11, R12;

Сценарий присваивает к выходному регистру R2 значение одного из четырех входных регистров R6, R8, R10 или R12 по указанному номеру R4.

Если R4 = 0, тогда R2 = R6.

Если R4 = 1, тогда R2 = R8.

Если R4 = 2, тогда R2 = R10.

Если R4 = 3, тогда R2 = R12.

Формат записи: R1.2 = (R3.4)? R5.6 : R7.8 : R9.10 : R11.12

Пример записи: **IR 10 = SEL (DI 1) ? HR 22 : HR 23 : HR 24 : HR 25**

IF - Логическая операция

Сценарий выполняет логическое условие ЕСЛИ (IF).

регистр	параметр
R0	7 - IF (R2 условие R5) тогда R6=R8 иначе R6=R9
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R4	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R5	Входной операнд 2
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer, 6-PC
R7	Выходной регистр
R8	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R9	Входной операнд если верно

Входные данные: R1, R2, R4, R5;

Выходные данные: R6, R7, R8, R9;

Функция: R3;

Логика работы сценария «Логические операции» следующая:

IF (R2 условие R5) тогда R7=R9. Действие выполняется однократно и не будет постоянно присваивать значение при верном условии. Присвоение выполнится снова только когда условие станет неверно и снова верно.

значение	функция
0	«==» если равно
1	«!=» если не равно
2	«>» если больше
3	«<» если меньше
4	«>=» если больше или равно
5	«<=» если меньше или равно
6	«!» логическая операция НЕ
7	«&&» логическая операция И
8	« » логическая операция ИЛИ

Формат записи: IF ((R1.2) (R3) (R4.5)) THEN (R6.7) = (R8.9)

Пример записи: 00: IFE (IR3 > HR5) THEN IR 25 = HR 125

Сценарий	R1	R2	R3	R4	R5	R6	R7	R8	R9
7 - IFE	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: 0...2660	0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767
Логическая операция	Тип	Вх операнд 1	функция	Тип	Вх операнд 2	Тип	Вых регистр	Тип	Вх регистр

IFAND - Логическая операция двойная с И

Сценарий выполняет логическое условие: ЕСЛИ (условие) И ЕСЛИ (условие) ТОГДА равно.

if((R1.2 R3 Const R4)&&(R5.6 R7 Const R8))then R9.10 = R11.12:

регистр	параметр
R0	8 - IFAND
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R4	Входной операнд 2
R5	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R6	Входной операнд 3
R7	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R8	Входной операнд 4
R9	Тип регистра: 1-Coils, 3- HR, 4-IR, 5-Timer, 6-PC
R10	Выходной регистр
R11	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R12	Входной операнд если верно

Входные данные: R1, R2, R4, R5, R6, R8, R11, R12;

Выходные данные: R9, R10;

Функция: R3, R7;

Логика работы сценария «Логические операции» следующая:

IF ((R1.2 R3 Const R4)&&(R5.6 R7 Const R8)) тогда R9.10=R11.12. Действие выполняется однократно и не будет постоянно присваивать значение при верном условии. Присвоение выполнится снова только когда условие станет неверно и снова верно.

значение	функция
0	«==» если равно
1	«!=» если не равно
2	«>» если больше
3	«<» если меньше
4	«>=» если больше или равно
5	«<=» если меньше или равно
6	«!» логическая операция НЕ
7	«&&» логическая операция И
8	« » логическая операция ИЛИ

Формат записи: IFAND ((R1.2 R3 Const R4)&&(R5.6 R7 Const R8)) THEN R9.10=R11.12

Пример записи: 00: IFAND (IR3 > HR5) && (IR7 > HR8) THEN IR 25 = HR 125

IFOR - Логическая операция двойная с ИЛИ

Сценарий выполняет логическое условие: ЕСЛИ (условие) ИЛИ ЕСЛИ (условие) ТОГДА равно.

if((R1.2 R3 Const R4) || (R5.6 R7 Const R8)) then R9.10 = R11.12:

регистр	параметр
R0	9 - IFOR
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R4	Входной операнд 2
R5	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R6	Входной операнд 3
R7	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R8	Входной операнд 4
R9	Тип регистра: 1-Coils, 3- HR, 4-IR, 5-Timer, 6-PC
R10	Выходной регистр
R11	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R12	Входной операнд если верно

Входные данные: R1, R2, R4, R5, R6, R8, R11, R12;

Выходные данные: R9, R10;

Функция: R3, R7;

Логика работы сценария «Логические операции» следующая:

ЕСЛИ ((R1.2 R3 Const R4) || (R5.6 R7 Const R8)) ТОГДА R9.10=R11.12. Действие выполняется однократно и не будет постоянно присваивать значение при верном условии. Присвоение выполнится снова только когда условие станет неверно и снова верно.

значение	функция
0	«==» если равно
1	«!=» если не равно
2	«>» если больше
3	«<» если меньше
4	«>=» если больше или равно
5	«<=» если меньше или равно
6	«!» логическая операция НЕ
7	«&&» логическая операция И
8	« » логическая операция ИЛИ

Формат записи: IFAND ((R1.2 R3 Const R4) || (R5.6 R7 Const R8)) THEN R9.10=R11.12

Пример записи: 00: IFAND (IR3 > HR5) || (IR7 > HR8) THEN IR 25 = HR 125

GOTO - Переход:

Сценарий выполняет переход на другую команду.

регистр	параметр
R0	10 - GOTO
R1	Переход (возможные значения 0-77)

Входные данные: нет;

Выходные данные: R1;

Команда GOTO переход позволяет перепрыгнуть несколько сценариев. Сценарии выполняются последовательно от 0 до 77 и снова повторяются. Команда GOTO может выполняться совместно с условием IF. Если номер сценария указан больше 77, то будет переход на номер 0.

Например:

2: If (IR2 > 30) переход на 5, иначе выполнится следующая операция

3: MATH R7 = R4 * R7

4: GOTO 6

5: MATH R7 = R4 - R7

6: BITS R12 = R4 & R7

Формат записи: GOTO (R1)

Пример записи: 00: **GOTO 125**

Сценарий	R1
10 - GOTO	Число: 0...77
Переход	Номер счетчика команд

CALL – Переход в подпрограмму:

Сценарий выполняет переход на подпрограмму.

регистр	Параметр
R0	11 – CALL
R1	Переход (возможные значения 0-77)

Входные данные: нет;

Выходные данные: R1;

Функция: нет;

Логика работы сценария «Переход» следующая: CALL номера сценария.

Эта команда работает так же, как и GOTO, но запоминает номер своего сценария. Команда позволяет перейти на подпрограмму. В конце подпрограммы используется команда RETURN. Может выполняться совместно с условием. Если номер сценария указан больше 77, то будет переход на номер 0.

Формат записи: CALL (R1)

Пример записи: 00: **CALL 121**

Сценарий	R1
11 - CALL	Число: 0...77
Вызов подпрограммы	Номер счетчика команд

RETURN – Выход из подпрограммы:

Сценарий выполняет выход из подпрограммы.

регистр	Параметр
R0	12 – RETURN

Логика работы сценария «Переход» следующая: RETURN.

Команда ставится в конце сценария подпрограммы и делает переход на следующий шаг, откуда был сделан вызов CALL. Например:

2: If (IR2 > 30) переход на 100, иначе выполнится следующая операция

3: MATH IR28 = IR38 * IR7

4: MATH IR7 = DI4 - IR7

5: BITS R12 = R4 & R7

6: MOV Coil3=R12

99: GOTO 0; переход в начало алгоритма

100: MOV IR7 = 25

101: MATH IR38 = DI4 * IR7

102: RETURN; возврат из подпрограммы

Формат записи: RETURN

Пример записи: 00: **RETURN**

Сценарий
12 - RETURN
Выход из подпрограммы

THRS - Пороговое реле:

Сценарий позволяет получить дискретное значение из аналогового значения. Является аналогом триггера Шмидта. Переключение происходит при установленном пороге. Разница в порогах включения и выключения устанавливается в регистре гистерезис.

регистр	параметр
R0	13 - Пороговое реле
R1	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Выходной регистр
R3	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R4	Входной регистр
R5	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R6	Регистр с пороговым значением
R7	Гистерезис, (Константа)
R8	инверсия (0-нет, 1-инверсия)

Входные данные: R3, R4;

Выходные данные: R1, R2;

Коэффициенты: R5, R6, R7, R8.

Логика выполняет однократное действие, т.е. событие и не держит выход в одном состоянии. Поэтому после включения реле его можно выключить записью в регистр Coils.

Логика работы сценария «Пороговое реле» следующая.

if((R3.4>R5.6) && (flag=0)) flag=1; R1.2=1

Если значение больше заданного, то включает реле.

if((R3.4<(R5.6-R7)) && (flag=1)) flag=0; R1.2=0;

Если значение меньше заданного, то выключает реле.

Входное значение может быть значение АЦП, пересчитанные значения, дискретные значения или пользовательские данные.

Формат записи: THRS (R1.2) = 0 or 1, In (R3.4) th (R5.6) ± (R7) inv R8

Например, **THRS IR 5 = 0 or 1, In IR 1 th HR 110 ± 2 inv 0**

TRG - Триггер:

Сценарий позволяет сделать переключение выхода с выключенного состояния на включенное и обратно при кратковременном нажатии кнопки.

регистр	параметр
R0	14 = Триггер
R1	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Выходной регистр
R3	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R4	Входной регистр
R5	Пороговое значение, константа

Входные данные: R3, R4;

Выходные данные: R1, R2;

Параметры: R5.

// 0= тип=14, 1= Тип, 2= вых регистр, 3= тип, 4= вх регистр, 5= порог

Логика работы сценария «Триггер» следующая:

При четном нажатии в регистры записываются значения 1. При нечетном нажатии в регистры записываются значения 0. Пороговое значение – это уровень, при котором происходит переключение. Для дискретного входа записать значение 1. Логика выполняет однократное действие, т.е. событие и не держит выход в одном состоянии. Поэтому после включения реле его можно выключить записью в регистр Coils.

Формат записи: TRG (R1.2) =0 or 1, In (R3.4) th (R5)

Например, **TRG IR 5 = 0 or 1, In HR 110 th 2**

KDO – кнопка с удержанием.

Сценарий позволяет переключить один канал реле кратковременным нажатием кнопки и выключить несколько каналов реле, долгим нажатием кнопки.

Регистр	параметр
R0	15 – кнопка с удержанием
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Номер входного регистра
R3	Номер реле Coil при кратковременном нажатии
R4	Тип регистра: 1-Coils, 3-HR, 13-HR, 23-HR, 4-IR, 5-Timer
R5	Номер выходного регистра при удержании кнопки
R6	Значение
R7	Время удержания кнопки (в 0.1 сек)

Входные данные: R1, R2;

Выходные данные: R3, R4, R5, R6;

Параметры: R7

Сценарий позволяет переключить с выключенного состояния на включенное и обратно канал реле, указанный в R3 при кратковременном нажатии кнопки.

Позволяет изменить значение выхода, указанного в R4 при удержании кнопки больше времени в R5.

Для реле DRM88R можно в регистр HR5 записать число, меняющее сразу состояние всех 8 каналов.

Если в R4 указан 3 (HR), то сценарий запишет в R5 значение R6. Изменит все каналы.

Если 13 (HR), то сценарий только выключит указанные каналы, не меняя другие каналы (R5 &= ~R6). При R6=15 (0b00001111) выключит только 1, 2, 3, 4 каналы, остальные оставит неизменными.

Если в R4 указан 23 (HR), то сценарий только включит указанные каналы, не меняя другие каналы (R5 |= R6). При R6=51 (0b00110011) включит только 1, 2, 5, 6 каналы, остальные оставит неизменными.

Формат записи: KDB (R1.2) = (R7) dsec -> (Coil R3) -> (R4.5)=(R6)

Пример записи: 00: **KDO DI 2 = 10dsec -> Co 2 -> Co 3**

Сценарий	R1	R2	R3	R4	R5	R6	R7
15 - KDB	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: 0 ... 999	Число: 0 ... 23	1-Coils 3-HR 13-HR 23-HR 4-IR 5-Tim	Число: 0 ... 23	Число: 0 ... 255	Число: 0 ... 255
Кнопка	Тип	вход	Реле 1	Тип	Реле 2	Значение	Время 0,1S

KDB - кнопка с удержанием.

Сценарий позволяет переключить один канал реле кратковременным нажатием кнопки и переключить другой канал реле, долгим нажатием кнопки.

регистр	параметр
R0	16 – кнопка с удержанием
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Номер входного регистра
R3	Номер реле Coil при кратковременном нажатии
R4	Номер реле Coil при удержании кнопки
R5	Время удержания кнопки (в 0.1 сек)

Входные данные: R1, R2;

Выходные данные: R3, R4;

Параметры: R5

Сценарий позволяет переключить с выключенного состояния на включенное и обратно канал реле, указанный в R3 при кратковременном нажатии кнопки. И переключить с выключенного состояния на включенное и обратно канал реле, указанный в R4 при удержании кнопки больше времени в R5.

Формат записи: KDB (R1.2) = (R5) dsec -> (Coil R3) -> (Coil R4)

Пример записи: 00: **KDB DI 2 = 10dsec -> Co 2 -> Co 3**

Сценарий	R1	R2	R3	R4	R5
16 - KDB	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: 0 ... 999	Число: 0 ... 23	Число: 0 ... 23	Число: 0 ... 255
Кнопка	Тип	вход	Реле 1	Реле 2	Время 0,1S

TIMER - Таймер.

Сценарий позволяет выполнять события по таймеру.

регистр	параметр
R0	17 - Таймер
R1	Номер таймера
R2	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Максимальное значение таймера
R4	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer, 6-PC
R5	Выходной регистр или переход на другой сценарий
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Входной сценарий

Входные данные: R6, R7;

Выходные данные: R4, R5;

Параметры: R1, R2, R3.

Логика работы сценария «Таймер» следующая:

Сценарий опрашивает переменную Timer с указанным номером. Доступно 16 таймеров от 0 до 15. Эта переменная уменьшается на 1 каждые 0.1 сек. Когда переменная доходит до 0, то в переменную Timer записывается новое значение таймера из регистра R3 и выполняет команду. После этого переменная таймера снова начинает обратный отсчет. Максимальное значение таймера можно взять из любого регистра, включая значение самого таймера, это может привести к зацикливанию.

Если в регистре R5 указан тип 6, то произойдет переход на сценарий с адресом, указанным в R6. Если в регистре R5 указан тип от 0 до 5, то выполнится команда присвоения: R5 = R7.

Формат записи, если (R4<6): TIME (R1) = (R2.3) TO (R4.5) = (R6.7)

Формат записи, если (R4=6): TIMG (R1) = (R2.3) TO (R4.5)

Например, **TIM 2 = HR 50 TO Coil 4 = HR 25**

Сценарий	R1	R2	R3	R4	R5	R6	R7
17 - TIM	Число Тип - Timer	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R2	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim 6-PC	Число Тип в R4	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R6
Таймер	Номер таймера TIMER	тип	Значение таймера	тип	Вых регистр	тип	Входной сценарий

RTC – часы и дата

Сценарий позволяет выполнить действия при наступлении заданного времени

регистр	параметр
R0	18 – RTC
R1	тип: 0 – нет; 1 – Ежемесячно; 2 – Еженедельно; 3 – Ежедневно; 4 - Каждый час; 5 - Каждую минуту
R2	День – для «ежемесячно» или маска недели – для «еженедельно»;
R3	часы (0-23),
R4	минуты (0-59),
R5	секунды (0-59),
R6	Тип регистра: 1, 3, 4, 5, 6
R7	Выходной регистр
R8	Тип регистра: 0...5
R9	Входной операнд

Входные данные: R8, R9;

Выходные данные: R6, R7;

Параметры: R1 – R5.

Сценарий позволяет выполнить сценарий присвоения или переход CALL при наступлении указанного времени и даты.

1 – Ежемесячно: выполняет действия каждый месяц в указанный день, час, минуту и секунду.

2 – Еженедельно: выполняет действия каждую неделю в указанные дни недели, час, минуту и секунду.

3 – Ежедневно: выполняет действия каждый день в указанный час, минуту и секунду.

4 – Каждый час: выполняет действия каждый час в указанную минуту и секунду.

5 – Каждую минуту: выполняет действия каждую минуту в указанную секунду.

Например, для типа: 2 – Еженедельно нужно записать следующие значения:

R0 = 9;

R1 = 2 тип;

R2 = 1 маска недели;

R3 = 10 час;

R4 = 11 минута;

R5 = 12 секунда;

R6 = тип 6 переход;

R7 = 25 адрес перехода

Например, для типа: 4 – каждый час необходимо указать:

R0 = 9;

R1 = 4 тип;

R2 = 0;

R3 = 20 час;

R4 = 21 минута;

R5 = 0 секунда;

R6 = 1 тип Coils;

R7 = 1 номер реле;

R8 = 0 константа;

R9 = 1 включение реле.

Формат записи: RTC (R1), D=(R2), H=(R3) : (R4) : (R5) TO (R6) (R7) = (R8)

Пример записи: **RTC Ежемесячно, D=2, H=3 : 40 : 50 TO IR 7 = 28**

Сценарий	R1	R2	R3	R4	R5	R6	R7	R8
18 - RTC	0 – нет; 1 – Ежемесячно; 2 – Еженедельно; 3 – Ежедневно; 4 - Каждый час; 5 - Каждую минуту	1...31	0...23	0...59	0...59	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim 6-PC	Число: 0...2660	Число: -32768 ... +32767
Часы	Тип таймера	День или неделя	часы	минуты	секунды	Тип регистра:	Выходной регистр	Значение, записываемое в регистр

Маска дней недели (для таймера типа - Еженедельно):

Байт	Значение
1	ПН
2	ВТ
4	СР
8	ЧТ
16	ПТ
32	СБ
64	ВС

JAL - Управление Жалюзи.

Сценарий позволяет организовать процесс управления приводом, в частности жалюзи.

регистр	Параметр
R0	19 JAL
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной регистр источника данных (тип указан выше)
R3	Пороговое значение
R4	Канал реле для открытия
R5	Канал реле для закрытия
R6	Время срабатывания (0.1 сек)

Входные данные: R1, R2;

Выходные данные: R4, R5;

Параметры: R3, R6.

При изменении входного регистра больше 0 происходит включение канала реле открытия, выдержка заданного времени и выключение этого реле. При изменении входного регистра равном 0 происходит включение канала реле закрытия, выдержка заданного времени и выключение этого реле.

Формат записи: JAL IN (R1.2) threshold (R3) Coil Open (R4), Coil Close (R5), t=(R6) sec

Пример записи: **JAL In IR 20 Thr 30 Coil Open 4, Coil Close 5, t=30 sec**

Сценарий	R1	R2	R3	R4	R5	R6
19 - JAL	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R1	Число: Тип - Const 0...65535	Число Тип - Coils	Число Тип - Coils	Число: Тип - Const 0...255
Управление приводом	тип	Входной регистр	Пороговое значение	Канал реле для открытия	Канал реле для закрытия	Время срабатывания

PID - ПИД регулятор.

Сценарий позволяет организовать пропорционально интегрирующее дифференцирующее регулирование нагревательным элементом.

регистр	Параметр
R0	20 PID
R1	Выходной регистр – канал реле
R2	Входной регистр – измеренное значение (IR)
R3	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R4	Входной регистр – заданное значение (уставка)
R5	KP - Коэффициент пропорциональной составляющей (константа)
R6	KI – Коэффициент интегрирующей составляющей (константа)
R7	KD - Коэффициент дифференцирующей составляющей (константа)
R8	CycleTime – время, сек (константа)
R9	P - Пропорциональная составляющая (IR)
R10	I - Интегрирующая составляющая (IR)
R11	D - дифференцирующая составляющая (IR)
R12	MV – Результат - выделяемой мощности нагревателем (IR)

Входные данные: R2, R3, R4;

Выходные данные: R1;

Результат: R12;

Параметры: R5, R6, R7, R8.

DE = SP - PV; Разность между измеренным и заданным значением;

P = Kp*DE; Пропорциональная составляющая;

I = I + Ki*DE*CycleTime; Интегрирующая составляющая;

D = Kd*(DE - DE_last)/CycleTime; дифференцирующая составляющая;

DE_last = DE; предыдущее значение разности значений;

MV = P + I + D; Результат выделяемой мощности нагревателем.

Формат записи: PID Coil (R1), Meas IR (R2), Set (R3) (R4) (KP=(R5), KI=(R6), KD=(R7))

Пример записи: **PID Coil 2, Meas IR 2, Set IR 3 (KP=1, KI=2, KD=3)**

Параметры 9,10,11,12 это промежуточные данные поэтому не используются.

Сценарий	R1	R2	R3	R4
20 - PID	Число Тип - Coils	Число Тип - IR	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R3
ПИД регулятор	Канал реле	Изм значение	тип	Заданное значение

R5	R6	R7	R8	R9	R10	R11	R12
Число: Тип - Const	Число: Тип - Const	Число: Тип - Const	Число: Тип 0 - Const	Число: -32768 ... +32767	Число: -32768 ... +32767	Число: -32768 ... +32767	Число: -32768 ... +32767
KP	KI	KD	Time	P	I	D	MV

8. ПО для управления и настройки блоков

Управлять модулями, настроить модули и написать сценарии можно записью в регистры Modbus необходимых значений с помощью различных программ:

8.1. программа ModbusPoll

Программа позволяет читать и записывать в регистры любые значения. Программа универсальная, поэтому её придется настраивать самостоятельно и специфические данные она не расшифровывает.

The screenshot shows the Modbus Poll software interface with several data tables for different modules:

- DRM88R_KNB (Tx = 74):**

Alias	00000
0	Адрес
1	Парам порта
2	Время вкл
3	Время выкл
4	Кор часов
5	Реле
6	
7	
8	
9	
- DRM88R_KNB (Tx = 79):**

Alias	00000
0	
1	Реле 1
2	Реле 2
3	Реле 3
4	Реле 4
5	Реле 5
6	Реле 6
7	Реле 7
8	Реле 8
9	
- DRM88R_KNB (Tx = 77):**

Alias	00000
0	
1	Вход 1
2	Вход 2
3	Вход 3
4	Вход 4
5	Вход 5
6	Вход 6
7	Вход 7
8	Вход 8
9	
- DRM88R_Input (Tx = 63):**

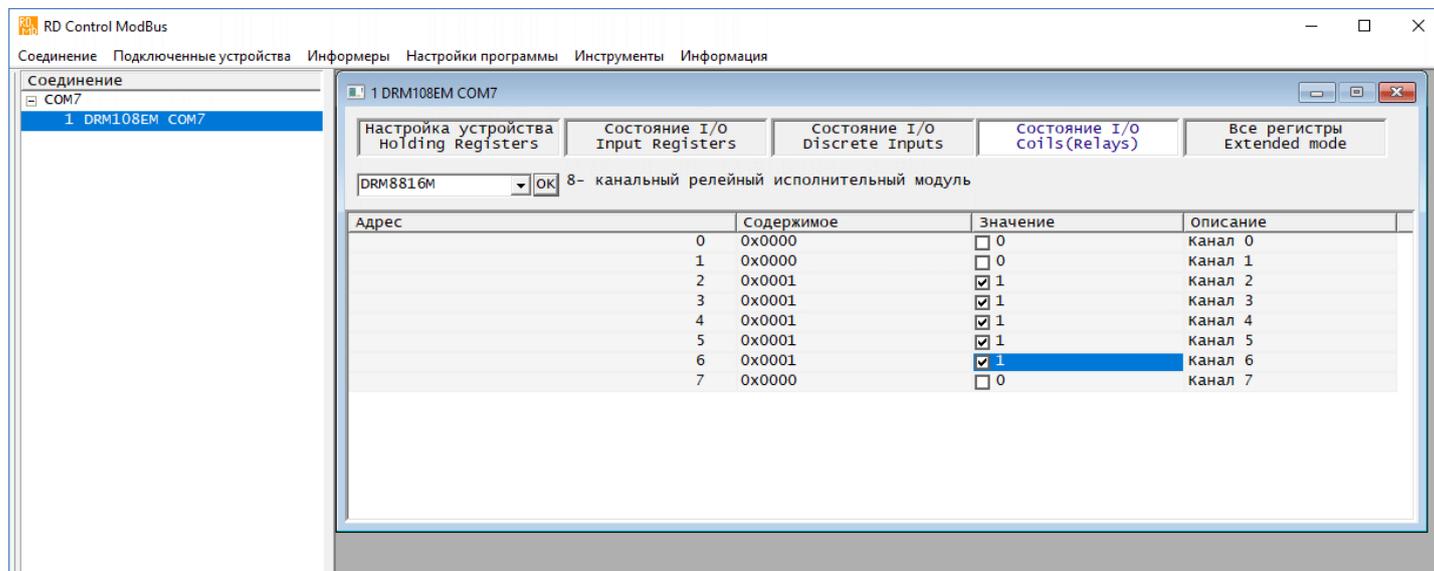
Alias	00000	Alias	00010	Alias	00020
0					
1	Вход 1	-24586	K*X/N+B	0	DISCRET1
2	Вход 2	-24586	K*X/N+B	0	DISCRET2
3	Вход 3	-24586	K*X/N+B	0	DISCRET3
4	Вход 4	-24586	K*X/N+B	0	DISCRET4
5	Вход 5	-24586	K*X/N+B	0	DISCRET5
6	Вход 6	-24586	K*X/N+B	0	DISCRET6
7	Вход 7	-24586	K*X/N+B	0	DISCRET7
8	Вход 8	-24586	K*X/N+B	0	DISCRET8
9					
- DRM88R_KNB (Tx = 53):**

Alias	00040	Alias	00050	Alias	00060
0	K	0	N	0	B
1	N	0	B	0	K
2	B	0	K	0	N
3	K	0	N	0	B
4	N	0	B	0	
5	B	0	K	0	
6	K	0	N	0	
7	N	0	B	0	
8	B	0	K	0	
9	K	0	N	0	
- DRM88R_HR100_sцен (Tx = 70):**

Alias	00100	Alias	00120	Alias	00140	Alias	00160	Alias	00180	Alias	00200
0	Тип	0	KD = 21	0	Таймер	0	IF	0		0	
1	Регистр 1	0	DI = 2	0	номер	0	тип	0		0	
2	Регистр 2	0	Inp 1	0	тип рег	0	вх операнд	0		0	
3	Регистр 3	0	Coil 1	0	значен	0	функ	0		0	
4	Регистр 4	0	Coil 2	0	тип рег	0	тип	0		0	
5	Регистр 5	0	Time = 10	0	вх рег	0	вх операнд	0		0	
6	Регистр 6	0		0	тип вх рег	0	тип вх	0		0	
7	Регистр 7	0		0	вх рег	0	выход	0		0	
8	Регистр 8	0		0		0	тип	0		0	
9	9	0		0		0	вх	0		0	
10		0		0		0		0		0	
11		0		0		0		0		0	
12		0		0		0		0		0	

8.2. программа RDControl Modbus 3

Программа позволяет управлять и настраивать модули РД. Программа оптимизирована для работы с модулями Разумный дом и может расшифровывать полученные данные. Программа не может создавать сценарии.



8.3. программа RDControl Modbus 4

Программа позволяет управлять и настраивать модули и создавать сценарии. Программа оптимизирована для работы с модулями Разумный дом и может расшифровывать полученные данные и создавать сценарии для работы блока.

