

**Блоки РД с интерфейсом Modbus RTU.**

**DRM88R, DRM21R, DSU44R, DDL44R**

**Протокол информационного обмена данными блоков с системами телемеханики.**

**Содержание.**

1. Общие сведения.
2. Контроль и управление блоком.
3. Настройки интерфейса.
4. Конфигурация блоков.
5. Сценарии для внутренней логики.
6. ПО для управления и настройки блоков.

## 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](http://ModBus.org).
- 1.8** Режим функционирования блока – **«Slave»** (подчинённый).
- 1.9** Режим передачи информации – **«RTU»** (бинарный режим).
- 1.10** Используемые функции (команды) обмена информацией:
- код функции – **01 Read Coils**;
  - код функции – **02 Read Discrete Inputs**;
  - код функции – **03 Read Holding Registers (HR)**;
  - код функции – **04 Read Input registers (IR)**.;
  - код функции – **05 Write Single Coil**;
  - код функции – **06 Write Single Register**;
  - код функции – **16 Write Multiple registers**.
- 1.11** Адрес блока – согласно протоколу **MODBUS**. По умолчанию все блоки 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 блока **0** в регистр **0**, но при этом на шине должен быть только один блок. 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 команды внутренней логики**.

## 2. Контроль и управление блоком.

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 блока и датчиков. И на выходные данные, воздействующие на выходы блока.

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 в два байта. 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;

Блоки поддерживают только целочисленные значения. Значения с плавающей запятой блоки не поддерживают. При необходимости передачи десятичных значений нужно в блоке умножить значение на 1000 и передать в контроллер целое число, которое будет в 1000 раз больше. А в контроллере это число разделить на 1000. В результате получится число с тысячными долями.

## 2.1. Управление выходами блоков.

Для управления выходами блоков 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			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
<b>Co 0</b>	0...1	данные	данные	данные	данные
<b>Co 1</b>	0...1	Реле канал <b>1</b>	Реле канал <b>1</b>	Выход 1	Выход 1
<b>Co 2</b>	0...1	Реле канал <b>2</b>	данные	Выход 2	Выход 2
<b>Co 3</b>	0...1	Реле канал <b>3</b>	данные	Выход 3	Выход 3
<b>Co 4</b>	0...1	Реле канал <b>4</b>	данные	Выход 4	Выход 4
<b>Co 5</b>	0...1	Реле канал <b>5</b>	данные	Выход общ	Выход общ
<b>Co 6</b>	0...1	Реле канал <b>6</b>	данные	данные	данные
<b>Co 7</b>	0...1	Реле канал <b>7</b>	данные	данные	данные
<b>Co 8</b>	0...1	Реле канал <b>8</b>	данные	данные	данные
<b>Co 9 ... 23</b>	0...1	данные	данные	данные	данные

Значение 0 соответствует отключенному состоянию реле, а значение 1 соответствует включенному состоянию реле.

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

Эти регистры доступны для чтения и записи.

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
HR 5	0...1023	Все реле	реле	Яркость 1	Яркость 1
HR 6	0...1023	-	-	Яркость 2	Яркость 2
HR 7	0...1023	-	-	Яркость 3	Яркость 3
HR 8	0...1023	-	-	Яркость 4	Яркость 4
HR 9	0...100	-	-	Яркость общая	Яркость общая

Регистр HR5 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.

Яркость соответствует выходному уровню ШИМ для модулей DDL и DSU. Значение 0 соответствует минимальному уровню ШИМ и на выходе будет постоянный 0. Значение 1023 соответствует максимальному уровню ШИМ и на выходе будет постоянный 1. Значение 511 будет соответствовать 50% скважности ШИМ, длительность 0 и 1 будет одинаковой. В модулях может стоять интегратор, который преобразует сигнал ШИМ в уровень напряжения 0-10В. Преобразование линейное и 50% ШИМ будет соответствовать 5В напряжения.

## 2.2. 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			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
DI 0	0...1	-	-	-	-
DI 1	0...1	вход 1	вход 1	вход 1	вход 1
DI 2	0...1	вход 2	вход 2	вход 2	вход 2
DI 3	0...1	вход 3	-	вход 3	вход 3
DI 4	0...1	вход 4	-	вход 4	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	-	-	-

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.

04 Input registers (IR).

Регистры **Input registers (IR)** хранят состояние аналоговых входов. Эти регистры можно только читать командами **Modbus**. Каждый вход опрашивается микросхемой АЦП, которая выдает значение от **0** до **40960**.

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			
		DRM88R	DRM21R	DDL44R	DSU44R
IR 0	-32768 +32767	IR0	IR0	-	-
IR 1	0...40960	АЦП 1	АЦП 1	АЦП 1	АЦП 1
IR 2	0...40960	АЦП 2	АЦП 2	АЦП 2	АЦП 2
IR 3	0...40960	АЦП 3	IR 3	АЦП 3	АЦП 3
IR 4	0...40960	АЦП 4	IR 4	АЦП 4	АЦП 4
IR 5	0...40960	АЦП 5	IR 5	-	-
IR 6	0...40960	АЦП 6	IR 6	-	-
IR 7	0...40960	АЦП 7	IR 7	-	-
IR 8	0...40960	АЦП 8	IR 8	-	-
IR 9	-32768 +32767	IR 9	IR 9	-	-
IR10	-32768 +32767	IR10	IR10	-	-
IR11	-32768 +32767	ADC1*K/N+B	ADC1*K/N+B	ADC1*K/N+B	ADC1*K/N+B
IR12	-32768 +32767	ADC2*K/N+B	ADC2*K/N+B	ADC2*K/N+B	ADC2*K/N+B
IR13	-32768 +32767	ADC3*K/N+B	IR13	ADC3*K/N+B	ADC3*K/N+B
IR14	-32768 +32767	ADC4*K/N+B	IR14	ADC4*K/N+B	ADC4*K/N+B
IR15	-32768 +32767	ADC5*K/N+B	IR15	-	-
IR16	-32768 +32767	ADC6*K/N+B	IR16	-	-
IR17	-32768 +32767	ADC7*K/N+B	IR17	-	-
IR18	-32768 +32767	ADC8*K/N+B	IR18	-	-
IR19	-32768 +32767	IR19	IR19	-	-
IR20	-32768 +32767	IR20	IR20	-	-
IR21	0...1	DI1	DI1	DI1	DI1
IR22	0...1	DI2	DI2	DI2	DI2
IR23	0...1	DI3	IR23	DI3	DI3
IR24	0...1	DI4	IR24	DI4	DI4
IR25	0...1	DI5	IR25	-	-
IR26	0...1	DI6	IR26	-	-
IR27	0...1	DI7	IR27	-	-
IR28	0...1	DI8	IR28	-	-
IR 9 ... IR999	-32768 +32767	IR9... IR999	IR9... IR999	IR9... IR999	IR9... IR999

В регистрах IR21 ... IR28 находится состояние дискретных входов DI1 ... DI8.

Каждый вход внутри имеет подтягивающий резистор 4,7кОм к напряжению +5В. Затем через 22кОм приходит на ножку микросхемы АЦП. Модуль будет реагировать на любое входное напряжение от 0.1В до 5В. 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.

### 3. Настройки интерфейса.

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

Эти регистры доступны для чтения и записи.

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
HR 0	1...255	Адрес	Адрес	Адрес	Адрес
HR 1	0...0xFFFF	Настройки	Настройки	Настройки	Настройки

Параметры по умолчанию:

Адрес модуля:	1
Скорость:	9600 бит/сек
Бит данных:	8 бит
Чётность:	Нет
Стоповых бит:	2

Адрес можно поменять в регистре 0. Параметры можно поменять в регистре 1. После изменения адреса, модуль нужно отключить и снова включить. Адрес устройства изменится только после перезапуска устройства.

Примечание. Для активации параметров по умолчанию запустите модуль, нажав кнопку или замкнув переключатель. Переключатель находится под лицевой панелью модуля.

Настройка порта Modbus RTU (старшие 8 bit – options + младшие 8 bit – baudrate)

Options:		Baudrates:	
2STOPS	0x0000	9600	0x0000
1STOPS	0x0100	19200	0x0001
PARITY_EVEN	0x0200	38400	0x0002
PARITY_ODD	0x0400	57600	0x0003
PARITY_NO	0x0000	115200	0x0004

Например, 0x0104 = четность нет, 1 стоп бит и 115200

Установка адреса.

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. После изменения адреса и параметров, модуль нужно отключить и снова включить.

Для задания адреса и других настроек можно воспользоваться программой RD Control Modbus v3.0 или MODBUS POLL с официального сайта.

## 4. Конфигурация блоков.

### 4.1. Информационные регистры.

#### 04 Input registers (IR).

Регистры **Input registers (IR)** хранят состояние режимов. Эти регистры можно только читать командами **Modbus**.

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
IR 9000	0...65535	номер ревизии ПО	номер ревизии ПО	номер ревизии ПО	номер ревизии ПО
IR 9001	0...65535	номер ревизии ПО	номер ревизии ПО	номер ревизии ПО	номер ревизии ПО
IR 9002	0...1	Версия ПО Release -0 / Debug -1			
IR 9003	0...255	Тип устройства: 11 – DRM88R	Тип устройства:	Тип устройства:	Тип устройства:
IR 9004	1...31	Дата: день	Дата: день	Дата: день	Дата: день
IR 9005	1...7	Дата: неделя	Дата: неделя	Дата: неделя	Дата: неделя
IR 9006	1...12	Дата: месяц	Дата: месяц	Дата: месяц	Дата: месяц
IR 9007	0...99	Дата: год	Дата: год	Дата: год	Дата: год
IR 9008	0...23	Время: часы	Время: часы	Время: часы	Время: часы
IR 9009	0...59	Время: минуты	Время: минуты	Время: минуты	Время: минуты
IR 9010	0...59	Время: секунды	Время: секунды	Время: секунды	Время: секунды
IR 9011	0...65535	SN	SN	SN	SN
IR 9012	0...65535	SN	SN	SN	SN
IR 9013	0...65535	SN	SN	SN	SN
IR 9014	0...65535	SN	SN	SN	SN
IR 9015	0...65535	SN	SN	SN	SN
IR 9016	0...65535	SN	SN	SN	SN
IR 9017	0...1		Детектор АС		
IR 9018	0...5	Код ошибки I2C	Код ошибки I2C	Код ошибки I2C	Код ошибки I2C
IR 9019	0...1	Состояние кнопки	Состояние кнопки	Состояние кнопки	Состояние кнопки

Информационные регистры для идентификации блока: номер ревизии, версия, тип и серийный номер.

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

Детектор АС – это наличие переменного напряжения (1) или отсутствие напряжения (0).

Код ошибки I2C должен быть всегда 0, это значит нет ошибок. Цифра от 1 до 5 говорит о наличии ошибки микросхем на шине I2C: ЕЕПРОМ, АЦП, ЦАП, дисплей и других. Если есть ошибка, то модуль подлежит ремонту.

**4.2. Установка параметров конфигурации.**

*03 (0x03) Read Holding Registers (HR),*

*06 (0x06) Write Single Register, 16 (0x10) Write Multiple registers.*

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
HR 2	0...100	-	Время вкл реле	-	-
HR 3	0...100	-	Время выкл реле	-	-
HR4	-100 ... 0 ... +100	Коррекция часов	Коррекция часов	Коррекция часов	Коррекция часов
HR40	-32768 +32767	Канал 1, К	Канал 1, К	Канал 1, К	Канал 1, К
HR41	-32768 +32767	Канал 1, N	Канал 1, N	Канал 1, N	Канал 1, N
HR42	-32768 +32767	Канал 1, В	Канал 1, В	Канал 1, В	Канал 1, В
HR43	-32768 +32767	Канал 2, К	Канал 2, К	Канал 2, К	Канал 2, К
HR44	-32768 +32767	Канал 2, N	Канал 2, N	Канал 2, N	Канал 2, N
HR45	-32768 +32767	Канал 2, В	Канал 2, В	Канал 2, В	Канал 2, В
HR46	-32768 +32767	Канал 3, К	-	Канал 3, К	Канал 3, К
HR47	-32768 +32767	Канал 3, N	-	Канал 3, N	Канал 3, N
HR48	-32768 +32767	Канал 3, В	-	Канал 3, В	Канал 3, В
HR49	-32768 +32767	Канал 4, К	-	Канал 4, К	Канал 4, К
HR50	-32768 +32767	Канал 4, N	-	Канал 4, N	Канал 4, N
HR51	-32768 +32767	Канал 4, В	-	Канал 4, В	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	0...8	Калькулятор канал	Калькулятор канал	Калькулятор канал	Калькулятор канал
HR65	-32768 +32767	Калькулятор Параметр 1	Калькулятор Параметр 1	Калькулятор Параметр 1	Калькулятор Параметр 1
HR66	-32768 +32767	Калькулятор Параметр 2	Калькулятор Параметр 2	Калькулятор Параметр 2	Калькулятор Параметр 2
HR67	-32768 +32767	Калькулятор АЦП 1	Калькулятор АЦП 1	Калькулятор АЦП 1	Калькулятор АЦП 1
HR68	-32768 +32767	Калькулятор АЦП 2	Калькулятор АЦП 2	Калькулятор АЦП 2	Калькулятор АЦП 2
HR69	0...255		усреднение АЦП		
HR93	1...31	Уст дата: День	Уст дата: День	Уст дата: День	Уст дата: День
HR94	1...7	Уст дата: неделя	Уст дата: неделя	Уст дата: неделя	Уст дата: неделя
HR95	1...12	Уст дата: Месяц	Уст дата: Месяц	Уст дата: Месяц	Уст дата: Месяц
HR96	18...118	Уст дата: Год	Уст дата: Год	Уст дата: Год	Уст дата: Год
HR97	0...23	Уст время: Часы	Уст время: Часы	Уст время: Часы	Уст время: Часы
HR98	0...59	Уст время: RTC минут			
HR99	0...59	Уст время: Секунд	Уст время: Секунд	Уст время: Секунд	Уст время: Секунд

Эти регистры доступны для чтения и записи.

У блока DRM21R есть детектор перехода нуля сети. С помощью него можно за синхронизировать переключение контактов реле в момент перехода нуля напряжения сети. Это позволит переключать мощную активную нагрузку с минимальной искрой на контактах реле и, тем самым, значительно продлить срок службы реле. Поскольку реле инерционное, то для каждого реле можно установить время включения и выключения реле. Время устанавливается в регистрах HR2 и HR3 в миллисекундах.

Полученные значения АЦП можно преобразовать по формуле:  $X = \frac{ADC * K}{N} + B$ ; Результат расчета помещается в регистры IR11 – IR18. Коэффициенты хранятся в регистрах HR40 – HR63. Для расчета этих коэффициентов формулы уравнения прямой необходимо использовать две точки. Коэффициенты рассчитываются по формуле.

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

$N = \text{ADC}2 - \text{ADC}1$ ;

$B = (\text{ADC}1 * \text{Температура}2 - \text{ADC}2 * \text{Температура}1) / (\text{ADC}1 - \text{ADC}2)$ ;

В блоках добавлен калькулятор для автоматического расчета этих коэффициентов. В регистрах HR64 – HR68.

Последовательность действий следующая.

1) подключить датчик.

2) в регистр HR64 записать номер канала (1 ... 8 в зависимости от модификации), к которому подключен датчик.

3) установить датчик в калибровочную камеру.

4) после стабилизации значений вписать значение первого параметра в регистр HR65 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для первого параметра в регистр HR67.

5) изменить величину климатического параметра.

6) после стабилизации значений вписать значение второго параметра в регистр HR66 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для второго параметра в регистр HR68. Затем модуль рассчитает коэффициенты и перепишет эти параметры в регистры коэффициентов номера канала, указанного в HR64. После этого в регистрах IR11 ... IR17 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.

В регистры HR93 – HR99 можно установить новое значение даты и времени. Т.к. в блоках нет батарейки, то при отключении питания часы сбросятся. Для постоянной работы часов необходимо использовать внешний ИБП. Для коррекции хода часов необходимо в регистр HR4 вписать поправку с плюсом или минусом. Эти поправка добавляется к счетчику миллисекундного таймера.

Регистр HR69 – 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. 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 и выходы.

Сценарии могут использовать данные со входов, управлять выходами и использовать регистры для хранения необходимых значений.

### 01 Read Coils, 05 Write Single Coil.

Регистры **Coils** хранят состояние выхода и пользовательские данные. Эти регистры доступны для чтения и записи. Значение данных может быть только 0 или 1.

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
<b>Co 0</b>	0...1	данные	данные	данные	данные
<b>Co 1</b>	0...1	Реле канал <b>1</b>	Реле канал <b>1</b>	Выход 1	Выход 1
<b>Co 2</b>	0...1	Реле канал <b>2</b>	данные	Выход 2	Выход 2
<b>Co 3</b>	0...1	Реле канал <b>3</b>	данные	Выход 3	Выход 3
<b>Co 4</b>	0...1	Реле канал <b>4</b>	данные	Выход 4	Выход 4
<b>Co 5</b>	0...1	Реле канал <b>5</b>	данные	Выход общ	Выход общ
<b>Co 6</b>	0...1	Реле канал <b>6</b>	данные	данные	данные
<b>Co 7</b>	0...1	Реле канал <b>7</b>	данные	данные	данные
<b>Co 8</b>	0...1	Реле канал <b>8</b>	данные	данные	данные
<b>Co 9 ... 23</b>	0...1	данные	данные	данные	данные

04 Input registers (IR).

Регистры **Input registers (IR)** хранят состояние аналоговых входов. Кроме входных данных в этих регистрах можно хранить собственные данные, т.е. результат работы сценария. Значения **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 в ЕЕПРОМ. Часть адресов жестко зафиксированы под определенные входные данные.

Регистр	Диапазон данных	Назначение			
		<u>DRM88R</u>	<u>DRM21R</u>	<u>DDL44R</u>	<u>DSU44R</u>
IR 0	-32768 +32767	IR0	IR0	-	-
IR 1	0...40960	АЦП 1	АЦП 1	АЦП 1	АЦП 1
IR 2	0...40960	АЦП 2	АЦП 2	АЦП 2	АЦП 2
IR 3	0...40960	АЦП 3	IR 3	АЦП 3	АЦП 3
IR 4	0...40960	АЦП 4	IR 4	АЦП 4	АЦП 4
IR 5	0...40960	АЦП 5	IR 5	-	-
IR 6	0...40960	АЦП 6	IR 6	-	-
IR 7	0...40960	АЦП 7	IR 7	-	-
IR 8	0...40960	АЦП 8	IR 8	-	-
IR 9	-32768 +32767	IR 9	IR 9	-	-
IR10	-32768 +32767	IR10	IR10	-	-
IR11	-32768 +32767	ADC1*К/N+B	ADC1*К/N+B	ADC1*К/N+B	ADC1*К/N+B
IR12	-32768 +32767	ADC2*К/N+B	ADC2*К/N+B	ADC2*К/N+B	ADC2*К/N+B
IR13	-32768 +32767	ADC3*К/N+B	IR13	ADC3*К/N+B	ADC3*К/N+B
IR14	-32768 +32767	ADC4*К/N+B	IR14	ADC4*К/N+B	ADC4*К/N+B
IR15	-32768 +32767	ADC5*К/N+B	IR15	-	-
IR16	-32768 +32767	ADC6*К/N+B	IR16	-	-
IR17	-32768 +32767	ADC7*К/N+B	IR17	-	-
IR18	-32768 +32767	ADC8*К/N+B	IR18	-	-
IR19	-32768 +32767	IR19	IR19	-	-
IR20	-32768 +32767	IR20	IR20	-	-
IR21	0...1	DI1	DI1	DI1	DI1
IR22	0...1	DI2	DI2	DI2	DI2
IR23	0...1	DI3	IR23	DI3	DI3
IR24	0...1	DI4	IR24	DI4	DI4
IR25	0...1	DI5	IR25	-	-
IR26	0...1	DI6	IR26	-	-
IR27	0...1	DI7	IR27	-	-
IR28	0...1	DI8	IR28	-	-
IR29 ... IR999	-32768 +32767	IR9... IR999	IR9... IR999	IR9... IR999	IR9... IR999

В регистрах с **21 – 28** выводится состояние реле в дискретном виде, и оно совпадает с **Discrete Inputs DI1 ... DI8**.

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

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 2660 зарезервировано место для записи сценариев с 0 по 127. Все указанные в таблице значения сохраняются в ЕЕПРОМ. Все неиспользуемые адреса не записываются и не сохраняются.

Сценарии внутри блока выполняются последовательно от 0 до 127. 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 (15) GOTO 0. Или пропустить пустые сценарии этой же командой.

<i>Адрес</i>							<i>Описание регистра</i>
R0	R1	R2	R3	R4	...	R19	
HR100	HR101	HR102	HR103	HR104	...	HR119	Сценарий № 0 и его параметры
HR120	HR121	HR122	HR123	HR124	...	HR139	Сценарий № 1 и его параметры
HR140	HR141	HR142	HR143	HR144	...	HR159	Сценарий № 2 и его параметры
HR160	HR161	HR162	HR163	HR164	...	HR179	Сценарий № 3 и его параметры
HR180	HR181	HR182	HR183	HR184	...	HR199	Сценарий № 4 и его параметры
...	...	...	...	...	...	...	...
HR2600	HR2601	HR2602	HR2603	HR2604	...	HR2619	Сценарий № 125 и его параметры
HR2620	HR2621	HR2622	HR2623	HR2624	...	HR2639	Сценарий № 126 и его параметры
HR2640	HR2641	HR2642	HR2643	HR2644	...	HR2659	Сценарий № 127 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».

<i>Значение</i>	<i>Тип данных</i>	<i>диапазон</i>	<i>Чтение запись</i>
0	Const – константа, фиксированное число.	(-32768 ... +32767)	Только чтение.
1	Coils (Co) – 01 регистр Реле.	(0 ... 23)	Чтение и запись.
2	Discrete Input (DI) – 02 регистр дискретных входов	(0 ... 8)	Только чтение.
3	Holding Reg (HR) – 03 регистр параметров.	(0 ... 2660)	Чтение и запись.
4	Input Reg (IR) – 04 регистр аналоговых входов.	(0...999)	Чтение и запись.
5	Timer – регистр таймера обратного отсчета.	(0...15)	Чтение и запись.
6	PC счетчик команд для перехода на другой сценарий.	(0 ... 127)	Только 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.

<i>Номер</i>	<i>Таймер</i>
0	Таймер 0
1	Таймер 1
2	Таймер 2
3	Таймер 3
4	Таймер 4
5	Таймер 5
6	Таймер 6
7	Таймер 7

<i>Номер</i>	<i>Таймер</i>
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 №0 ... №127*

В каждом сценарии в адресе от 100 до 119 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>Тип сценария</i>
<b>0</b>	<b>NOP</b> - Пустая команда. Не производит никаких действий
<b>1</b>	<b>FORM</b> - Формула
<b>2</b>	<b>THRS</b> - Пороговое реле
<b>3</b>	<b>NTHRS</b> - Пороговое инверсное реле
<b>4</b>	<b>TRG</b> - Триггер
<b>5</b>	<b>EQU</b> - Выход = входу
<b>6</b>	<b>TIMER</b> - Таймер
<b>7</b>	<b>JAL</b> - Управление Жалюзи
<b>8</b>	<b>PID</b> - ПИД регулятор
<b>9</b>	<b>RTC</b> – часы и дата
<b>10</b>	<b>MATH</b> - Арифметические операции
<b>11</b>	<b>BITS</b> - Побитовые операции
<b>12</b>	<b>MOV</b> - Присвоение R2 = R4
<b>13</b>	<b>IFG</b> - Логические операции IF (R2 условие R5) с переходом
<b>14</b>	<b>IFE</b> - Логические операции IF (R2 условие R5) с функцией присвоения
<b>15</b>	<b>GOTO</b> – Переход
<b>16</b>	<b>CALL</b> – Переход в подпрограмму
<b>17</b>	<b>RETURN</b> – Выход из подпрограммы
<b>18</b>	<b>PTR</b> – указатель для создания массивов переменных
<b>19</b>	<b>FOR</b> – создание циклов
<b>20</b>	<b>TIMER</b> – Таймер
<b>21</b>	<b>KD</b> – кнопка с диммированием
<b>22 – 200</b>	Зарезервировано. Не использовать.
<b>201 - 65535</b>	<b>NOP</b> - Пустая команда. Не производит никаких действий

**FORM - Формула:**

Сценарий позволяет перевести значение входных данных в другое значение после пересчета по заданной формуле.

регистр	параметр
R0	1 = Формула
R1	Тип 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	Функция: формула, по которой производится расчет
R6	Коэффициент K
R7	Коэффициент N
R8	Коэффициент B

Входные данные: R4, R5;

Выходные данные: R1, R2;

Функция: R5;

Коэффициенты: R6, R7, R8.

Сценарий «Формула» производит расчет по заданной формуле.

Значение	функция
0	$R2 = K * R4 / N + B$
1	$R2 = (K / N) / R4 + B$
2	$R2 = (R4 + B) * (K / N)$

Если делитель (знаменатель) получается 0, то программа подставляет вместо нуля единицу и производит расчет, не возвращая ошибку.

Формат записи: FORM (R1) (R2) = (R5) (X=(R3) (R4), K= (R6), N= (R7), B= (R8))

Например, **FORM IR 5 = (K / N) \* X + B (X= IR 10, K= 20, N= 30, B= -200)**

Сценарий	R1	R2	R3	R4	R5	R6	R7	R8
1 - FORM	1-Coils 2 3-HR 4-IR 5-Tim	Число	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число	0 - $(K / N) * X + B$ 1 - $(K / N) / X + B$ 2 - $(K / N) * (X + B)$	Число Тип - Const	Число Тип - Const	Число Тип - Const
Формула	тип	результат	тип	Исходное значение	формула	K	N	B

**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	2 - Пороговое реле
R1	Входной регистр источника данных (всегда IR)
R2	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Регистр с пороговым значением (тип указан выше)
R4	Гистерезис, (Константа)
R5	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R6	Выходной регистр
R7	Выходной регистр (Coils). Если не используется, то записать 0.

Входные данные: R1;

Выходные данные: R5, R6, R7;

Функция: нет;

Коэффициенты: R2, R3, R4.

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((R1>R3) && (flag=0)) flag=1; R6=1; R7=1;`

Если значение больше заданного, то включает реле.

`if((R1<(R3-R4)) && (flag=1)) flag=0; R6=0; R7=0;`

Если значение меньше заданного, то выключает реле.

Входное значение всегда берется из IR, при этом могут быть значения АЦП, пересчитанные значения, дискретные значения или пользовательские данные. Регистр с пороговым значением может быть, как константа, так и данные с любого типа регистров.

Формат записи: THRS (R5) (R6) = Coil (R7) = 0 or 1, In IR (R1) th (R2) (R3) ± (R4)

Например, **THRS IR 5 = Coil 6 = 0 or 1, In IR 1 th HR 110 ± 2**

Сценарий	R1	R2	R3	R4	R5	R6	R7
2 - THRS	Число Тип - IR	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R2	Число Тип - Const	0 1-Coils 2 3-HR 4-IR 5-Tim	Число Тип в R5	Число Тип - Coils
Пороговое инверсное реле	Входной регистр IR	Тип	Пороговое значение	Гистерезис Const	Тип	Выходной регистр	Coils выходной регистр

**NTHRS - Пороговое реле, инверсное значение:**

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	3 - Пороговое реле
R1	Входной регистр источника данных (всегда IR)
R2	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Регистр с пороговым значением (тип указан выше)
R4	Гистерезис, (Константа)
R5	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R6	Выходной регистр
R7	Выходной регистр (Coils). Если не используется, то записать 0.

Входные данные: R1;

Выходные данные: R5, R6, R7;

Функция: нет;

Коэффициенты: R2, R3, R4.

Этот сценарий точно такой же, как и предыдущий только выходная команда инверсная.

Логика работы сценария «Пороговое реле инверсное» следующая.

`if((R1>R3) && (flag=0)) flag=1; R6=0; R7=0;`

Если значение больше заданного, то выключает реле.

`if((R1<(R3-R4)) && (flag=1)) flag=0; R6=1; R7=1;`

Если значение меньше заданного, то включает реле.

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.

Входное значение всегда берется из IR, при этом могут быть значения АЦП, пересчитанные значения, дискретные значения или пользовательские данные. Регистр с пороговым значением может быть, как константа, так и данные с любого типа регистров.

Формат записи: **NTHRS (R5) (R6) = Coil (R7) = 1 or 0, In IR (R1) th (R2) (R3) ± (R4)**

Например, **NTHRS IR 5 = Coil 6 = 1 or 0, In IR 1 th HR 110 +- 2**

Сценарий	R1	R2	R3	R4	R5	R6	R7
2 - THRS	Число Тип - IR	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R2	Число Тип - Const	1-Coils 3-HR 4-IR 5-Tim	Число Тип в R5	Число Тип - Coils
Пороговое инверсное реле	Входной регистр IR	Тип	Пороговое значение	Гистерезис	Тип	Выходной регистр	Coils выходной регистр

**TRG - Триггер:**

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

регистр	параметр
R0	4 = Триггер
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	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R5	Выходной регистр (всегда IR)
R6	Выходной регистр (Coils). Если не используется, то записать 0.

Входные данные: R1, R2;

Выходные данные: R4, R5, R6;

Функция: нет;

Параметры: R3.

Логика работы сценария «Триггер» следующая:

При четном нажатии в регистры Reg и Coils записываются значения 1. При нечетном нажатии в регистры Reg и Coils 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 (R4) (R5) = Coil (R6) = (R1) (R2) th (R3)

Например, **TRG IR 5 = Coil 6 = 0 or 1, In HR 110 th 2**

Сценарий	R1	R2	R3	R4	R5	R6
4 - TRG	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R1	Число Тип - Const	0-нет 1-Coils 2-нет 3-HR 4-IR 5-Tim	Число Тип в R4	Число Тип - Coils
Триггер	Тип	Входной регистр	Пороговое значение	Тип	Выходной регистр	Coils выходной регистр

**EQU - Состояние выхода = состоянию входа:**

Сценарий присваивает значение входного регистра выходному регистру.

регистр	параметр
R0	5 R4 = R2
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной регистр источника данных (тип указан выше)
R3	Инверсия
R4	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R5	Выходной регистр
R6	Выходной регистр (Coils). Если не используется, то записать 0.

Входные данные: R1, R2;

Выходные данные: R4, R5, R6;

Параметры: R3.

Логика работы сценария «Вход = выходу» следующая:

If(R5 <> R2) R5 = R2; R6 = R5;

Если инверсное значение:

If(R5 <> (0xFFFF-R2+2)) R5 = 0xFFFF - R2+2; R6 = 0xFFFF - R2+2;

Инверсия 0 будет -1+2=1, инверсия 1 будет -2+2=0, инверсия 2 будет -3+2=-1 и так далее.

Логика выполняет однократное действие, событие и не держит выход реле в одном состоянии. Поэтому после включения реле его можно выключить записью в регистр Coils. Если Coils не используется, то в него нужно записать 0.

Входное значение берется из любого регистра.

Формат записи: EQU (R4) (R5) = Coil (R6) = (R3) (R1) (R2)

Например, **EQU HR 1 = Coil 3 = Inv HR 2**

Сценарий	R 1	R 2	R 3	R 4	R 5	R 6
5 - EQU	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R1	0-dir 1-Inv	1-Coils 3-HR 4-IR 5-Tim	Число Тип в R4	Число Тип - Coils
Выход равен входу	Тип	Входной регистр	Инверсия	Тип	Выходной регистр	Coils выходной регистр

**TIMER - Таймер.**

Сценарий позволяет выполнять события по таймеру.

регистр	параметр
R0	6 - 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) (R3) TO (R4) (R5) = (R6) (R7)

Формат записи, если (R4=6): TIMG (R1) = (R2) (R3) TO (R4) (R5)

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

Сценарий	R1	R2	R3	R4	R5	R6	R7
6 - 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

**JAL - Управление Жалюзи.**

Сценарий позволяет организовать процесс управления приводом, в частности жалюзи.

регистр	Параметр
R0	7 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) (R2) 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
7 - 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 интегрирующе дифференцирующее регулирование нагревательным элементом.

регистр	Параметр
R0	8 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
8 - 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

**RTC – часы и дата**

Сценарий позволяет выполнить действия при наступлении заданного времени

регистр	параметр
R0	9 – 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
9 - 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

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

Сценарий производит арифметические действия над двумя входными регистрами.

регистр	параметр
R0	10 - R2 = R4 (операция) R7
R1	Тип 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 операнд 1
R5	Функция: 0- «+», 1- «-», 2- «*», 3- «/», 4- «%», 5- «++», 6- «--»
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Входной операнд 2

Входные данные: R3, R4, R6, R7;

Выходные данные: R1, R2;

Функция: R5;

Логика работы сценария «арифметическая операция» следующая: R2 = R4 (операция) R7; R2=R4 + R7;

Функция может быть:

значение	функция
0	«+» сложение
1	«-» вычитание
2	«*» умножение
3	«/» деление
4	«%» остаток от деления
5	«++» прибавление к результату
6	«--» вычитание из результата

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**

Сценарий	R1	R2	R3	R4	R5	R6	R7
10 - MATH	1-Coils 3-HR 4-IR 5-Tim	Число: 0...2660	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	0 - + 1 - - 2 - * 3 - / 4 - % 5 - ++ 6 - --	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767
арифметика	Тип	Вых регистр	Тип	Операнд 1	Операция	Тип	Операнд 2

**BITS - Побитовые сценарии R2 = R4 (операция) R7:**

Сценарий производит побитовые действия над двумя входными регистрами.

регистр	параметр
R0	11 - R2 = R4 (операция) R7
R1	Тип 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 операнд 1
R5	Функция: 0- «~», 1- «&», 2- « », 3- «^», 4- «<<», 5- «>>»
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Входной операнд 2

Входные данные: R3, R4, R6, R7;

Выходные данные: R1, R2;

Функция: R5;

Логика работы сценария «Побитовые сценарии» следующая: R2 = ~ R4 или R2 = R4 (операция) R7.

Функция может быть:

значение	функция
0	«~» побитная инверсия только над первым операндом
1	«&» логическая И
2	« » логическая ИЛИ
3	«^» Побитовое исключающее ИЛИ
4	«<<» побитовый сдвиг влево
5	«>>» побитовый сдвиг вправо

Сохранение результата в «константу» и «Discrete Inputs» невозможно.

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

Пример записи: 00: **BITS HR 20 = IR 10 & HR30**

Сценарий	R1	R2	R3	R4	R5	R6	R7
11 - BITS	1-Coils 3-HR 4-IR 5-Tim 6-нет	Число: 0...2660	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767	0 - ~ 1 - & 2 -   3 - ^ 4 - << 5 - >>	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767
Побитовая логика	Тип	Вых регистр	Тип	Операнд 1	Операция	Тип	Операнд 2

**MOV - Присвоение R2 = R4:**

Сценарий присваивает значение входного регистра выходному регистру.

регистр	параметр
R0	12 – R2 = R4
R1	Тип 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 операнд

Входные данные: R3, R4;

Выходные данные: R1, R2;

Логика работы сценария «Присвоение» следующая: R2 = R4. Значение из R4 записывается в R2.

Формат записи: MOV (R1) (R2) = (R3) (R4)

Пример записи: 00: **MOV IR 2 = HR 4**

Сценарий	R1	R2	R3	R4
		Тип в пар 1		Тип в пар 3
12 - MOV	1-Coils 3-HR 4-IR 5-Tim	Число: 0...2660	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число: -32768 ... +32767
Присвоение	Тип	Вых регистр	Тип	Вх регистр

**IFG - Логические операции IFG (R2 условие R5):**

Сценарий выполняет логическое условие ЕСЛИ (IF).

регистр	параметр
R0	13 - IFG (R2 условие R5) переход
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	Переход (возможные значения 0-127)

Входные данные: R1, R2, R4, R5;

Выходные данные: R6;

Функция: R3;

Логика работы сценария «Логические операции» следующая: IFG (R2 условие R5) тогда выполнится переход CALL на адрес указанного сценария, иначе выполнится следующая операция. При переходе запоминается адрес выхода, чтобы после выполнения подпрограммы вернуться командой RETURN. Для выполнения одного из действий дополнительно используйте команду перехода GOTO. Если номер сценария указан больше 127, то будет переход на номер 0.

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

Например:

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

3: MOV IR25 = 1

4: GOTO 6

5: MOV IR25 = 0

Формат записи: IFG ( (R1) (R2) (R3) (R4) (R5) ) THEN GOTO (R6)

Пример записи: 00: **IFG (IR3 > HR5) THEN GOTO 25**

Сценарий	R1	R2	R3	R4	R5	R6
13 - IFG	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	Число: 0...127
Логическая операция	Тип	Вх операнд 1		Тип	Вх операнд 2	Переход

**IFE - Логические операции IF (R2 условие R5):**

Сценарий выполняет логическое условие ЕСЛИ (IF).

регистр	параметр
R0	14 - IFE (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:

IFE (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: IFE ( (R1) (R2) (R3) (R4) (R5) ) THEN (R6) (R7) = (R8) (R9)

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

Сценарий	R1	R2	R3	R4	R5	R6	R7	R8	R9
14 - 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		Тип	Вх операнд 2	Тип	Вых регистр	Тип	Вх регистр

**GOTO - Переход:**

Сценарий выполняет переход на другую команду.

регистр	параметр
R0	15 - GOTO
R1	Переход (возможные значения 0-127)

Входные данные: нет;

Выходные данные: R1;

Функция: нет;

Команда GOTO переход позволяет перепрыгнуть несколько сценариев. Сценарии выполняются последовательно от 0 до 127 и снова повторяются. Команда GOTO может выполняться совместно с условием IF. Если номер сценария указан больше 127, 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
15 - GOTO	Число: 0...127
Переход	Номер счетчика команд

**CALL – Переход в подпрограмму:**

Сценарий выполняет переход на подпрограмму.

регистр	Параметр
R0	16 – CALL
R1	Переход (возможные значения 0-127)

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 127, то будет переход на номер 0.

Формат записи: CALL (R1)

Пример записи: 00: **CALL 121**

Сценарий	R1
16 - CALL	Число: 0...127
Вызов подпрограммы	Номер счетчика команд

**RETURN – Выход из подпрограммы:**

Сценарий выполняет выход из подпрограммы.

регистр	Параметр
R0	17 – 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
17 - RETURN
Выход из подпрограммы

***PTR – указатель для создания массивов переменных***

Сценарий позволяет записывать данные в массив данных.

регистр	параметр
R0	18 – Type[TypeR3] = TypeR4
R1	Тип выходного 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) (R3) ] = (R4) (R5)

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

Сценарий	R1	R2	R3	R4	R5
	выбор	выбор	Тип в пар 1	выбор	Тип в пар 4
18 - PTR	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	Вых регистр	Тип	Вх регистр

### FOR – циклы

Сценарий позволяет создать циклы.

регистр	Параметр
R0	19 – 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;

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,

25: FOR (IR35=3; IR35<=7; IR35++) GOTO 29; цикл от 3 до 7

26: MATH IR35 = IR35+25; операции внутри цикла, прибавление значения

27: PTR IR[IR35] = IR35; операции внутри цикла, заполнение массива

28: GOTO 25; переход в начало цикла

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

Формат записи: FOR (IR (R1) = (R2) (R3) TO (R2) (R4)) RET (R6)

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

Сценарий	R1	R2	R3	R4	R5
	Тип IR	выбор	Тип в пар 2	Тип в пар 2	выбор
19 – 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

**KDO – кнопка с удержанием.**

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

Регистр	параметр
R0	20 – кнопка с удержанием
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Номер 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) (R2) = (R7) dsec -> (Coil R3) -> (R4) (R5)=(R6)

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

Сценарий	R1	R2	R3	R4	R5	R6	R7
20 - 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	21 – кнопка с удержанием
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Номер 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) (R2) = (R5) dsec -> (Coil R3) -> (Coil R4)

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

Сценарий	R1	R2	R3	R4	R5
21 - 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

## 6. ПО для управления и настройки блоков

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

1. программа ModbusPoll 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 ModbusPoll interface with several data tables. The top row contains four tables for modules: DRM88R\_KNB (Tx = 74: Err = 1: ID = 1: F), DRM88R\_Input (Tx = 79: Err = 4: ID), DRM88R\_KNB (Tx = 77: Err = 0: ID), and DRM88R\_KNB (Tx = 53: Err = 1: ID = 1: F = 03: SR = 1000ms). The bottom table is for DRM88R\_HR100\_scen (Tx = 70: Err = 2: ID = 1: F = 03: SR = 1000ms).

Alias	00000	Alias	00010	Alias	00020
0	Адрес	0	0	0	0
1	Парам порта	0	0	0	0
2	Время вкл	0	0	0	0
3	Время выкл	0	0	0	0
4	Кор часов	0	0	0	0
5	Реле	0	0	0	0
6		-1			
7		-1			
8		-1			
9		-1			

  

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	1
5	B	0	K	0	0
6	K	0	N	0	0
7	N	0	B	0	0
8	B	0	K	0	0
9	K	0	N	0	0

  

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

2. программа RDControl Modbus 3 позволяет управлять и настраивать модули РД. Программа оптимизирована для работы с модулями Разумный дом и может расшифровывать полученные данные. Программа не может создавать сценарии.

The screenshot shows the RD Control ModBus interface. The left pane shows the connection list with '1 DRM108EM COM7' selected. The main pane shows the configuration for '1 DRM108EM COM7' with the device name 'DRM8816M' and description '8- канальный релейный исполнительный модуль'. The 'Все регистры Extended mode' tab is active, displaying a table of registers.

Адрес	Содержимое	Значение	Описание
0	0x0000	<input type="checkbox"/> 0	канал 0
1	0x0000	<input type="checkbox"/> 0	канал 1
2	0x0001	<input checked="" type="checkbox"/> 1	канал 2
3	0x0001	<input checked="" type="checkbox"/> 1	канал 3
4	0x0001	<input checked="" type="checkbox"/> 1	канал 4
5	0x0001	<input checked="" type="checkbox"/> 1	канал 5
6	0x0001	<input checked="" type="checkbox"/> 1	канал 6
7	0x0000	<input type="checkbox"/> 0	канал 7

3. программа RDControl Modbus 4 позволяет управлять и настраивать модули и создавать сценарии. Программа оптимизирована для работы с модулями Разумный дом и может расшифровывать полученные данные.

#### Работа с программой RDControl Modbus 4.

После запуска программы необходимо:

- создать соединение в меню Соединение, указав необходимые параметры.
- создать или открыть решение. Для этого:
- Открыть раздел в меню: Файл -> Создать решение или Открыть решение.
- Выбрать папку и написать имя файла.
- Откроется окно с тремя закладками: Устройства, Каналы, Сценарии.
- Если устройства подключены, то нажать кнопку Поиск устройств.

Если устройства не подключены, нажать кнопку Добавить устройство. И в открывшемся окне выбрать тип устройства, его адрес и параметры соединения.

- В закладке Устройства будет отображен список подключенных устройств.
- В закладке Каналы будут отображаться все входы и выходы с необходимым количеством каналов.
- В закладке сценарии будут отображаться все 128 строк сценариев.

Продолжение следует...

Создание документа близится к завершению.