8 канальный релейный модуль DRM88R v1.0

Описание регистров Modbus и встроенных сценариев.

3.4. Управление модулем.

Протокол обмена данными Modbus подразумевает наличие в сети мастера, которым является контроллер и около 250 подчиненных. Данный модуль является подчиненным и может только отвечать на запросы мастера. Самостоятельно посылать команды в сеть модуль не может. Поэтому описанные ниже сценарии могут работать только внутри самого модуля, используя собственные входы и выходы.

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

01 (0x01) Read Coils, 05 (0x05) Write Single Coil.

Регистры Coils хранит состояние реле. Эти регистры доступны для чтения и записи. Из этого регистра можно читать состояние реле. Запись в этот регистр переключает реле. Значение может быть 0 – реле выключено или 1 – реле включено. Запись в этот регистр значения больше 1 будет возвращать значение 1.

Рег	*Назначение*
0	Пользовательские данные
1	Реле канал 1
2	Реле канал 2
3	Реле канал 3
4	Реле канал 4
5	Реле канал 5
6	Реле канал 6
7	Реле канал 7
8	Реле канал 8
9 … 23	Пользовательские данные
24…65535	Не используется

02 (0x02) Read Discrete Inputs.

Регистры Discrete Input (DI) хранит состояние дискретных входов. Эти регистры можно только читать командами Modbus. Из этого регистра можно читать состояние дискретных входов. Состояние дискретного входа определяется из аналогового входа АЦП. Порог переключения 1000. Если значение АЦП меньше 1000, то возвращает 0, если значение больше 1000, то возвращает 1.

Рег	*Назначение*
0	Не используется
1	Дискретный вход 1
2	Дискретный вход 2
3	Дискретный вход 3
4	Дискретный вход 4
5	Дискретный вход 5
6	Дискретный вход 6
7	Дискретный вход 7
8	Дискретный вход 8
9…65535	Не используется

04 Input registers (IR).

Регистры Input registers (IR) хранят состояние аналоговых входов. Эти регистры можно только читать командами Modbus. 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. Для чтения и записи доступно 1000 регистров с адресами от 0 до 999. Эти данные представляют собой массив ОЗУ и не сохраняются в ЕЕПРОМ. Часть адресов жестко зафиксированы под определенные входные данные.

При необходимости значения АЦП можно преобразовать по формуле:

$$X = \frac{ADC*K}{N} + B;$$

Это используется, например, при подключении датчика температуры, когда вместо значений АЦП надо выводить температуру. Новые значения сохраняются в регистрах с IR11 по IR18. В регистрах с 21 – 28 выводится состояние реле в дискретном виде. По сути оно совпадает с Discrete Inputs.

Рег	*назначение*
0	Пользовательские данные IR0
1	канал 1, ADC диапазон значений 0 … 40960
2	канал 2, ADC диапазон значений 0 … 40960
3	канал 3, ADC диапазон значений 0 … 40960
4	канал 4, ADC диапазон значений 0 … 40960
5	канал 5, ADC диапазон значений 0 … 40960
6	канал 6, ADC диапазон значений 0 … 40960
7	канал 7, ADC диапазон значений 0 … 40960
8	канал 8, ADC диапазон значений 0 … 40960
9	Пользовательские данные IR9
10	Пользовательские данные IR10
11	Канал №1, X = k(HR40)/k_scale(HR41)+b(HR42)
12	Канал №2, X = k(HR43)/k_scale(HR44)+b(HR45)

13	Канал №3, X = k(HR46)/k_scale(HR47)+b(HR48)
14	Канал №4, X = k(HR49)/k_scale(HR50)+b(HR51)
15	Канал №5, X = k(HR52)/k_scale(HR53)+b(HR54)
16	Канал №6, X = k(HR55)/k_scale(HR56)+b(HR57)
17	Канал №7, X = k(HR58)/k_scale(HR59)+b(HR60)
18	Канал №8, X = k(HR61)/k_scale(HR62)+b(HR63)
19	Пользовательские данные IR19
20	Пользовательские данные IR20
21	канал 1, дискретное состояние, диапазон значений 0 … 1
22	канал 2, дискретное состояние, диапазон значений 0 … 1
23	канал 3, дискретное состояние, диапазон значений 0 … 1
24	канал 4, дискретное состояние, диапазон значений 0 … 1
25	канал 5, дискретное состояние, диапазон значений 0 … 1
26	канал 6, дискретное состояние, диапазон значений 0 … 1
27	канал 7, дискретное состояние, диапазон значений 0 … 1
28	канал 8, дискретное состояние, диапазон значений 0 … 1
29	Пользовательские данные IR29
…	Пользовательские данные IR30 – IR998
999	Пользовательские данные IR999
9000	Старший номер ревизии ПО
9001	Младший номер ревизии ПО
9002	Версия ПО Release -0 / Debug -1
9003	Тип устройства: 11 (0x000B) – DRM88R

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

Для чтения и записи параметров и сценариев используются регистры Holding Registers (HR). Эти регистры доступны для чтения и записи. Часть регистров используются под определенные параметры. Все указанные в таблице значения сохраняются в ЕЕПРОМ. Все неиспользуемые адреса игнорируются, не сохраняются в памяти и не записываются.

Адрес	*Описание регистра*
0	Адрес устройства на шине ModBus RTU и ModBus TCP
1	Настройки порта Modbus RTU
2	Код ошибки I2C
3	Состояние кнопки
4-39	Не используется
40	Канал №1, Коэффициент K
41	Канал №1, Коэффициент N
42	Канал №1, Коэффициент B
43	Канал №2, Коэффициент K
44	Канал №2, Коэффициент N
45	Канал №2, Коэффициент B
46	Канал №3, Коэффициент K
47	Канал №3, Коэффициент N
48	Канал №3, Коэффициент B
49	Канал №4, Коэффициент K
50	Канал №4, Коэффициент N
51	Канал №4, Коэффициент B
52	Канал №5, Коэффициент K
53	Канал №5, Коэффициент N
54	Канал №5, Коэффициент B
55	Канал №6, Коэффициент K
56	Канал №6, Коэффициент N
57	Канал №6, Коэффициент B
58	Канал №7, Коэффициент K
59	Канал №7, Коэффициент N
60	Канал №7, Коэффициент B
61	Канал №8, Коэффициент K
62	Канал №8, Коэффициент N
63	Канал №8, Коэффициент B
64-93	Не используется
94-99	Зарезервирован для RTC
100 -119	Сценарий № 0 и его параметры
120 -139	Сценарий № 1 и его параметры
…	Сценарий № 2 … 126 и его параметры
2640 – 2659	Сценарий № 127 и его параметры

С адреса 100 до адреса 2660 Зарезервировано место для записи сценариев с 0 по 127. Для каждого сценария зарезервировано по 20 регистров.
0 сценарий записывается в регистры 100-119,
1 сценарий в регистры 120-139,
2 сценарий в регистры 140-159 и так далее.

Эти же регистры можно использовать как переменные, сохраняемые в ЕЕПРОМ. Но для этого в 0 регистре должно быть записано значение 0, команда NOP. В этом случае следующие 19 регистров не будут восприниматься как команды и программа их будет игнорировать.

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

Значение	*Тип сценария*
0	NOP - Пустая команда. Не производит никаких действий
1	FORM - Формула
2	THRS - Пороговое реле
3	NTHRS - Пороговое инверсное реле
4	TRG - Триггер
5	EQU - Выход = входу
6	TIMER - Таймер
7	JAL - Управление Жалюзи
8	PID - ПИД регулятор
9	Зарезервирован (RTC – часы и дата)
10	MATH - Арифметические операции
11	BITS - Побитовые операции
12	MOV - Присвоение R2 = R4
13	IF - Логические операции IF (R2 условие R5) с переходом
14	IF - Логические операции IF (R2 условие R5) с функцией присвоения
15	GOTO - Переход
16	CALL – Переход в подпрограмму
17	RETURN – Выход из подпрограммы
18	PTR – указатель для создания массивов переменных
19	FOR – создание циклов

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*
0	Const – константа, фиксированное число (-32768 … +32767). Только чтение.
1	Coils – 01 регистр Реле, который содержит состояние реле (0 … 23). Доступно чтение и запись.
2	Discrete Input (DI) – 02 регистр - Дискретные входы (0 … 8). Только чтение.
3	Holding Reg (HR) – 03 регистр параметров (0 … 12900). Доступно чтение и запись.
4	Input Reg (IR) – 04 регистр Аналоговые входы. Доступно чтение и запись.
5	Timer – регистр таймера обратного отсчета.
6	Goto - Переход на другой сценарий. (0 … 127). Используется в некоторых сценариях.

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

Номер	*Таймер*
0	Таймер 0
1	Таймер 1
…	Таймер 2 … 14
15	Таймер 15

FORM - Формула:

регистр	параметр
R0	1 = Формула
R1	Тип выходного регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R2	Выходной регистр - Y
R3	Тип регистра: 0, 1, 2, 3, 4, 5
R4	Входной регистр – X
R5	Функция: формула, по которой производится рассчет
R6	Коэффициент K
R7	Коэффициент N
R8	Коэффициент B

Сценарий «Формула» производит расчет по заданной формуле. Данные берет из входного регистра R2, а результат сохраняет в выходной регистр R4.

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

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

THRS - Пороговое реле:

регистр	параметр
R0	2 - Пороговое реле
R1	Входной регистр источника данных (всегда IR)
R2	Тип регистра: 0, 1, 2, 3, 4, 5
R3	Регистр с пороговым значением (тип указан выше)
R4	Гистерезис, (Константа)
R5	Тип выходного регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R6	Выходной регистр
R7	Выходной регистр (Coils). Если не используется, то записать 0.

Логика работы сценария «Пороговое реле» следующая.
If((R1>R3) && (flag=0)) flag=1; R5=1; R6=1;
Если значение больше заданного, то включает реле.
if((R1<(R3-R4)) && (flag=1)) flag=0; R5=0; R6=0;
Если значение меньше заданного, то выключает 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.

NTHRS - Пороговое реле, инверсное значение:

регистр	параметр
R0	3 - Пороговое реле
R1	Входной регистр источника данных (всегда IR)
R2	Тип регистра: 0, 1, 2, 3, 4, 5
R3	Регистр с пороговым значением (тип указан выше)
R4	Гистерезис, (Константа)
R5	Тип выходного регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R6	Выходной регистр
R7	Выходной регистр (Coils). Если не используется, то записать 0.

Этот сценарий точно такой же, только выходная команда инверсная.
Логика работы сценария «Пороговое реле инверсное» следующая.
If((R1>R3) && (R5=1)) R5=0; R6=0;
Если значение больше заданного, то выключает реле.
if((R1<(R3-R4)) && (R5=0)) R5=1; R6=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.

TRG - Триггер:

регистр	параметр
R0	4 = Триггер
R1	Тип регистра: 0, 1, 2, 3, 4, 5
R2	Входной регистр источника данных (тип указан выше)
R3	Пороговое значение
R4	Тип выходного регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R5	Выходной регистр (всегда IR)
R6	Выходной регистр (Coils). Если не используется, то записать 0.

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.

EQU - Состояние выхода = состоянию входа:

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

Логика работы сценария «Вход = выходу» следующая:
If(R2 <> R5) R5 = R2; R6 = R5;
Если инверсное значение:
If(R2 <> R5) R5 = 0xFFFF - R2; R6 = R5;
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 значение, отличное от каналов реле.
Входное значение берется из любого регистра.

TIMER - Таймер.

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

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. Внимание, это может привезти 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.

JAL - Управление Жалюзи.

регистр	Параметр
R0	7 JAL
R1	Тип регистра: 0, 1, 2, 3, 4, 5
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)

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.

PID - ПИД регулятор.

регистр	Параметр
R0	8 PID
R1	Выходной регистр – канал реле
R2	Входной регистр – измеренное значение (IR)
R3	Тип регистра: 0, 1, 2, 3, 4, 5
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)

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.

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

регистр	параметр
R0	10 - R2 = R4 (операция) R7
R1	Тип регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R2	Выходной регистр
R3	Тип регистра: 0…5
R4	Входной операнд 1
R5	Функция: 0- «+», 1- «-«, 2- «*», 3- «/», 4- «%», 5- «++», 6- «--»
R6	Тип регистра
R7	Входной операнд 2

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

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

регистр	параметр
R0	11 - R2 = R4 (операция) R7
R1	Тип регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R2	Выходной регистр
R3	Тип регистра: 0…5
R4	Входной операнд 1
R5	Функция: 0- «~», 1- «&», 2- «|», 3- «^», 4- «<<», 5- «>>»
R6	Тип регистра
R7	Входной операнд 2

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.

MOV - Присвоение R2 = R4:

регистр	параметр
R0	12 – R2 = R4
R1	Тип регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R2	Выходной регистр
R3	Тип регистра: 0…5
R4	Входной операнд

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

IF - Логические операции IF (R2 условие R5):

регистр	параметр
R0	13 - IF (R2 условие R5) переход
R1	Тип регистра: 0…5
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- «||»
R4	Тип регистра: 0…5
R5	Входной операнд 2
R6	Переход (возможные значения 0-127)

Логика работы сценария «Логические операции» следующая:
IF (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: IR25 = 0
6: R2 = R4 & R7

IF - Логические операции IF (R2 условие R5):

регистр	параметр
R0	14 - IF (R2 условие R5) тогда R6=R8 иначе R6=R9
R1	Тип регистра: 0…5
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- «||»
R4	Тип регистра: 0…5
R5	Входной операнд 2
R9	Тип регистра: 0…5
R7	Выходной регистр
R8	Тип регистра: 0…5
R9	Входной операнд если верно

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

GOTO - Переход:

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

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

CALL – Переход в подпрограмму:

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

Логика работы сценария «Переход» следующая: CALL номер сценария.
Эта команда работает так же, как и GOTO, но запоминает номер своего сценария. Команда переход позволяет перепрыгнуть несколько сценариев. Может выполняться совместно с условием. Если номер сценария указан больше 127, то будет переход на номер 0.

RETURN – 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; подпрограмма, вызываемая из основной программы
101: MATH IR38 = DI4 * IR7
102: RETURN; возврат из подпрограммы

PTR – указатель для создания массивов переменных

регистр	параметр
R0	18 – Type[TypeR3] = TypeR4
R1	Тип выходного регистра: 1-Coils, 3-Holding Reg, 4-Input Reg, 5-Timer
R2	Тип регистра указателя: 0…5
R3	Номер регистр
R4	Тип входного регистра: 0…5
R5	Входной операнд

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.

FOR – создание циклов

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…5
R3	Регистр начала цикла
R4	Регистр конца цикла
R5	Адрес выхода из цикла

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

Адрес устройства на шине ModBus RTU. По умолчанию 1. Адрес устройства изменится только при перезапуске устройства.

Настройки 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
		230400	0x0005

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

В регистрах с HR40 по HR63 хранятся коэффициенты для пересчета значения АЦП в физические значения: градусы, проценты, Люксы и т.п.

Чтобы получить значения в градусах, нужно пересчитать биты АЦП в градусы по формуле T = K * X / N + B. Коэффициенты необходимо рассчитывать для конкретного датчика и диапазона. Значение в градусах уже после расчёта можно получить в регистрах IR11 – IR18.