

Модуль исполнительный диммерный: DDM845R, DDM8410R

Сценарии для внутренней логики

Содержание.

1. Общие сведения
2. Адреса сценариев.
3. Источники данных.
4. Типы данных.
5. Пользовательские данные.
6. Типы сценариев.
7. Примеры сценариев.

1. Общие сведения.

Для автономной работы блока без контроллера можно использовать встроенные сценарии. Протокол MODBUS подразумевает в сети одного мастера Master, которым может являться контроллер и множество подчиненных Slave, которым является данный модуль. Самостоятельно посылать команды в сеть модуль не может. 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.

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

Для чтения и записи сценариев используются регистры Holding Registers (HR). Эти регистры доступны для чтения и записи. Регистры HR0 – HR99 зафиксированы под определенные параметры. С адреса HR100 до адреса HR2660 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, тогда последним сценарием можно использовать переход в начало (15) GOTO 0. Можно пропустить пустые сценарии командой GOTO x(номер следующего сценария).

2. Адреса сценариев.

Адрес							Описание регистра
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

3. 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 ... 23)	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...999)	Int (-32768 ... +32767)	Чтение и запись.
5	Не используется			

6	Счетчик команд (PC) для перехода на другой сценарий.	(0 ... 127)	Int (-32768 ... +32767)	Только запись.
---	--	-------------	-------------------------	----------------

4. Типы данных.

Данные 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;
- bool – бинарное. Максимальные значения: 0 ... 1;

Большинство данных использует тип *int16* это двухбайтовое 16 битное число со знаком. Диапазон значений -32768 ... +32767. При работе с логикой или дискретными входами, выходами модуль использует значение *bool – false* (0) или *true* (1). При переводе из типа *int* значение 0 будет переведется в *false* (0), любое другое значение, отличное от 0 будет переводится в *true* (1). Модули поддерживают только целочисленные значения. Значения с плавающей запятой модули не поддерживают. При необходимости передачи десятичных значений нужно в модуле 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. Символьные значения модули не поддерживают.

5. 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 128 регистров с адресами от IR100 до IR227. Эти данные представляют собой массив ОЗУ и не сохраняются в ЕЕПРОМ. При отключении питания сбрасываются в 0. Часть адресов с IR0 по IR70 жестко зафиксированы под определенные входные данные.

2) Для чтения и записи настроек используются регистры Holding Registers (HR). Эти регистры доступны для чтения и записи. Эти регистры записываются в ЕЕПРОМ и при повторном включении питания восстанавливаются.

Значение	Тип данных	Размещение	Тип данных	Чтение запись
IR100 ... IR227	пользовательские данные	ОЗУ не сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.
HR100... HR2000	Данные сценариев	ЕЕПРОМ сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.

Данные можно сохранять и в области сценариев в тех регистрах, которые не используют сценарии.

6. Типы сценариев.

Доступны сценарии №0 ... №127. В каждом сценарии в адресе от HR100 до HR119 записываются тип и параметры сценария. Далее адрес будет обозначаться R0 – тип, записанный в регистр HR100 (для сценария 0), R1 – параметр 1, записанный в регистр HR101, R2 – параметр 2, записанный в регистр HR102 и так далее.

Регистр R0 – Тип сценария

Значение	Тип сценария	Обозначение
0	NOP - Пустая команда. Не производит никаких действий	
1	FORM - Формула	
2	THRS - Пороговое реле	HYSTERESIS
3	NTHRS - Пороговое инверсное реле	HYSTERESIS
4	TRG - Триггер	
5	EQU - Выход = входу	
6	TIMER - Таймер	TIME
10	MATH - Арифметические операции «+» сложение	ADD
10	MATH - Арифметические операции «-» вычитание	SUB
10	MATH - Арифметические операции «*» умножение	MUL
10	MATH - Арифметические операции «/» деление	DIV
10	MATH - Арифметические операции «%» остаток от деления	MOD
10	MATH - Арифметические операции «++» прибавление к результату	CTU
10	MATH - Арифметические операции «--» вычитание из результата	CTD
11	BITS - Побитовые операции «~» побитная инверсия	NOT
11	BITS - Побитовые операции «&» логическая И	AND
11	BITS - Побитовые операции « » логическая ИЛИ	OR
11	BITS - Побитовые операции «^» Побитовое исключающее ИЛИ	XOR
11	BITS - Побитовые операции «<<<» побитовый сдвиг влево	SHL
11	BITS - Побитовые операции «>>>» побитовый сдвиг вправо	SHR
12	MOV - Присвоение R2 = R4	MOVE
13	IFG - Логические операции IF (R2 == R5) с переходом, если равно	EQ
13	IFG - Логические операции IF (R2 != R5) с переходом, если не равно	NE
13	IFG - Логические операции IF (R2 > R5) с переходом, если больше	GT
13	IFG - Логические операции IF (R2 < R5) с переходом, если меньше	LT
13	IFG - Логические операции IF (R2 >= R5) с переходом, если больше или равно	GE
13	IFG - Логические операции IF (R2 <= R5) с переходом, если меньше или равно	LE
13	IFG - Логические операции IF (R2 ! R5) с переходом, логическая операция НЕ	
13	IFG - Логические операции IF (R2 && R5) с переходом, логическая операция И	
13	IFG - Логические операции IF (R2 R5) с переходом «логическая операция ИЛИ	
14	IFE - Логические операции IF (R2 условие R5) с функцией присвоения	
15	GOTO – Переход	
16	CALL – Переход в подпрограмму	CAL
17	RETURN – Выход из подпрограммы	
21	KEYDIM – кнопка диммирования	
29	SMOOTH – плавное изменение	
30 - 65535	NOP - 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 К
R7	Коэффициент N
R8	Коэффициент В

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
3 - 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

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 номер 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

KD – кнопка диммирования.

Сценарий позволяет включать и выключать кратковременным нажатием кнопки и диммировать долгим нажатием кнопки.

Регистр	параметр
R0	21 – кнопка диммирования
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR
R2	Номер входного регистра источника данных
R3	Выходной регистр HR, для плавного управления (регистры 0, 1, 2, 3)
R4	Время изменения яркости на единицу, мс (оптимально 10)
R5	Время удержания кнопки, мс (оптимально 1000)

Входные данные: R1, R2;

Выходные данные: R3;

Параметры: R4, R5

Логика работы сценария следующая: при кратковременном нечетном нажатии на кнопку (указанную в R1) включится канал (указанный в R2), при кратковременном четном нажатии на кнопку (указанную в R1) выключится канал (указанный в R2). При нечетном удержании кнопки будет плавное увеличение яркости, при четном удержании будет плавное уменьшение яркости. Время удержания кнопки до перехода в режим изменения яркости указывается в R4 в миллисекундах. Оптимальное значение 1 сек – 1000. Время увеличения яркости на единицу (влияет на скорость увеличения яркости) указывается в R3 в миллисекундах. Оптимально 10 мс.

Формат записи: KD (R3) = (R1) (R2) -> (R4)msec, (R5)msec,

Пример записи: 00: **KD HR 2 = DI 2 -> 10msec, 1000msec**

Сценарий	R1	R2	R3	R4	R5
21 - KDB	1-Coils 2-DI 3-HR 4-IR	Число: 0 ... 999	Число: 0 ... 4	Число: 0 ... 32767	Число: 0 ... 32767
Кнопка	Тип	вход	Выход	Время	Время

SMOOTH – плавное изменение.

Сценарий позволяет плавно изменять уровень на выходе.

регистр	параметр
R0	29 – плавное изменение
R1	Входной регистр источника данных IR(x)
R2	Значение входного регистра, по которому срабатывает алгоритм
R3	Выходной Holding регистр, для плавного управления выходом (регистры от 40 до 43)
R4	Направление изменения выходного регистра: 0 - увеличение, 1 - уменьшение
R5	Минимальное или максимальное значение выходного регистра (в зависимости от значения регистра HRxx4)
R6	Время изменения на единицу, мс

Входные данные: R1, R2;

Выходные данные: R3;

Параметры: R4, R5, R6

Сценарий позволяет плавно изменять уровень на выходе. Входной регистр источника данных IR(x), обычно результат от алгоритма типа "Триггер" или "Состояние выхода = состоянию входа".

Формат записи: SMOOTH (R3) = (R1) (R2) -> (R4), (R5), (R6)ms

Сценарий	R1	R2	R3	R4	R5
29 - SMOOTH	Число: 0 ... 999	Число: 0 ... 999	Число: 0 ... 23	Число: 0 ... 23	Число: 0 ... 255
Кнопка	вход	вход	Реле 1	Реле 2	Время 0,1S

7. Примеры сценариев.

1. Сценарий для подключения входов – кнопок диммирования к выходам модуля.

Логика работы сценария следующая: при кратковременном нечетном нажатии на кнопку включится соответствующий канал, при кратковременном четном нажатии на кнопку выключится соответствующий канал. При нечетном удержании кнопки будет плавное увеличение яркости, при четном удержании будет плавное уменьшение яркости. Время удержания кнопки до перехода в режим изменения яркости указывается в миллисекундах в регистрах HRxx5. Оптимальное значение 1 сек – 1000. Время увеличения яркости на единицу в миллисекундах в регистрах HRxx4. Оптимально 5 - 10 миллисекунд.

Регистр	Значение	Описание
HR100	21	Команда «Кнопка диммирования»
HR101	2	Тип регистра: 2 - Discrete Input
HR102	0	Номер регистра входа: DI0
HR103	0	Номер канала выхода: 0
HR104	5	Время изменения яркости на единицу, мс
HR105	1000	Время удержания кнопки, мс
HR120	21	Команда «Кнопка диммирования»
HR121	2	Тип регистра: 2 - Discrete Input
HR122	1	Номер регистра входа: DI1
HR123	1	Номер канала выхода: 1
HR124	5	Время изменения яркости на единицу, мс
HR125	1000	Время удержания кнопки, мс
HR140	21	Команда «Кнопка диммирования»
HR141	2	Тип регистра: 2 - Discrete Input
HR142	2	Номер регистра входа: DI2
HR143	2	Номер канала выхода: 2
HR144	5	Время изменения яркости на единицу, мс
HR145	1000	Время удержания кнопки, мс
HR160	21	Команда «Кнопка диммирования»
HR161	2	Тип регистра: 2 - Discrete Input
HR162	3	Номер регистра входа: DI3
HR163	3	Номер канала выхода: 3
HR164	5	Время изменения яркости на единицу, мс
HR165	1000	Время удержания кнопки, мс