

Модули исполнительные релейные: DRM21R, DRM88R

Сервисная инструкция по настройке модулей

Содержание.

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

1. Общие сведения

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

1.2 Количество бит данных по умолчанию – 8.

1.3 Количество стоповых бит по умолчанию – 2.

1.4 Бит чётности по умолчанию – отсутствует.

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

1.6 Протокол логического обмена – Modbus RTU.

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

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

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

1.10 Используемые функции (команды) обмена информацией:

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

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

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

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

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

1.15 Для групп информационных сигналов обмена выделены следующие группы данных:

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

2. Настройки интерфейса.

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

Эти регистры доступны для чтения и записи.

<i>Регистр</i>	<i>Адрес</i>	<i>Описание регистра</i>	<i>Диапазон</i>	<i>По умолчанию</i>
HR0	40001	Адрес устройства на шине ModBus RTU	1...247	1
HR1	40002	Настройки последовательного порта	0...0x55	0

2.1. Установка адреса.

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

2.2. Настройки порта Modbus RTU

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

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

Старшие 8 bit – *options* + младшие 8 bit – *baudrate*.

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

2.3. Параметры по умолчанию:

Адрес и параметры можно сбросить по умолчанию, запустив модуль с замкнутой кнопкой или переключкой «DEF». Через секунду после включения кнопку можно отпустить.

Переключку можно использовать во время работы и задействовать её в алгоритмах работы. Состояние переключки - кнопки можно прочитать в регистре IR9019.

Параметры по умолчанию:

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

Конфигурация и сценарии, при включении загружаются из flash в ОЗУ работают в ОЗУ. После изменения сохраняются во flash памяти. Для сохранения из ОЗУ во flash нужно записать любое значение в регистр HR92 или запись произойдет самостоятельно раз минуту, при условии изменения сохраняемых параметров.

<i>Регистр</i>	<i>Адрес</i>	<i>Описание регистра</i>	<i>Запись</i>	<i>Чтение</i>
HR 92	40093	Сохранение из ОЗУ во flash	Любое значение	Счетчик записей

3. Контроль и управление модулем.

Протокол обмена данными Modbus подразумевает наличие в сети мастера, которым является контроллер и 247 подчиненных. Данные модули являются подчиненным и могут только отвечать на запросы мастера.

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

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

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

- Signed - знаковое целое. Максимальные значения: -32768 ... +32767;
- Unsigned - беззнаковое целое. Максимальные значения: 0 ... +65535;
- Hex - шестнадцатеричное. Максимальные значения: 0x0000 ... 0xFFFF;
- Bool - бинарное. Максимальные значения: 0 ... 1;

- значения с запятой. Значения передаются в тысячных долях. Интерфейс передает только целые значения. Значению с запятой 1.000 соответствует 1000 в Modbus. Значению с запятой 0.001 соответствует 1 в Modbus. В контроллере это число нужно разделить на 1000. В результате получится число с тысячными долями. Значения с плавающей запятой модули не поддерживают.

4. Управление выходом модулей.

Для управления выходом модулей можно использовать регистры Coils.

4.1. Включение реле.

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.

<i>Регистр</i>	<i>Адрес</i>	<i>Диапазон данных</i>	<i>DRM21R</i>	<i>DRM88R</i>
Coil 1	20002	0...1	Канал реле	Реле канал 1
Coil 2	20003	0...1	Флаги для сценария	Реле канал 2
Coil 3	20004	0...1	Флаги для сценария	Реле канал 3
Coil 4	20005	0...1	Флаги для сценария	Реле канал 4
Coil 5	20006	0...1	Флаги для сценария	Реле канал 5
Coil 6	20007	0...1	Флаги для сценария	Реле канал 6
Coil 7	20008	0...1	Флаги для сценария	Реле канал 7
Coil 8	20009	0...1	Флаги для сценария	Реле канал 8
Coil 9 – Coil 23	20010	0...1	Флаги для сценария	Флаги для сценария

При записи 1 реле включится. При записи 0 – реле выключится.

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

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

<i>Регистр</i>	<i>Адрес</i>	<i>Диапазон данных</i>	<i>DRM21R</i>	<i>DRM88R</i>
HR 34	40035	0-255	-	Все реле

Эти регистры доступны для чтения и записи. Регистр HR34 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 3 и 5 выходы, остальные выключит.

4.2. Установка задержки включения реле.

Включение и выключение реле синхронизировано от детектора перехода сети через ноль. Это уменьшает уровень искры, излучаемые помехи, продлевает срок службы контактов.

Задержка зависит от инерционности якоря реле. Она может быть разная для разных типов реле и для циклов переключения. В регистрах HR2 можно установить задержку включения и выключения реле от момента определения перехода сети через ноль.

<i>Регистр</i>	<i>Адрес</i>	<i>Диапазон</i>	<i>Описание регистра</i>	<i>DRM21R</i>	<i>DRM88R</i>
HR 2	40003	1...255	Задержка включения реле	3	3
HR 3	40004		Задержка выключения реле	73	40

5. Входы.

К клеммам 1 - 8 подключаются дискретные или аналоговые датчики относительно G.

Выход +5В используется для питания датчиков, например, датчика влажности. Максимальный ток выхода «+5В» - 20мА.

Каждый вход внутри модуля подтянут резистором сопротивлением 4.7 или 22 кОм к напряжению +5В. Поэтому контактные датчики и кнопки необходимо подключать относительно G.

Входное аналоговое напряжение может быть в диапазоне 0 - 5 В. При подключении другого напряжения, например, 0 - 10В, необходимо использовать резисторный делитель из двух резисторов, сопротивлением по 1 кОм.

5.1. Дискретные входы.

02 (0x02) Read Discrete Inputs.

Регистры Discrete Input (DI) возвращают состояние дискретных входов. Эти регистры можно только читать командами Modbus.

<i>Регистр</i>	<i>Адрес</i>	<i>Диапазон данных</i>	<i>Назначение DRM21R</i>	<i>Назначение DRM88R</i>
DI 1	10002	0...1	Дискретный вход 1	Дискретный вход 1
DI 2	10003	0...1	Дискретный вход 2	Дискретный вход 2
DI 3	10004	0...1	-	Дискретный вход 3
DI 4	10005	0...1	-	Дискретный вход 4
DI 5	10006	0...1	-	Дискретный вход 5
DI 6	10007	0...1	-	Дискретный вход 6
DI 7	10008	0...1	-	Дискретный вход 7
DI 8	10009	0...1	-	Дискретный вход 8

В DRM21R состояние дискретного входа определяется из логического уровня. Скорость реакции на логические уровни гораздо больше, чем на аналоговые входы. В DRM88R 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. Скорость переключения ниже, чем у дискретных. Для максимальной скорости нужно усреднение АЦП сделать равным 1 в регистре HR75.

5.2. Аналоговые входы.

04 Input registers (IR).

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

<i>Рег</i>	<i>Адрес</i>	<i>Диапазон данных</i>	<i>Назначение DRM21R</i>	<i>Назначение DRM88R</i>
IR 1	30002	0 ... 4096	АЦП канал 1	АЦП канал 1
IR 2	30003	0 ... 4096	АЦП канал 2	АЦП канал 2
IR 3	30004	0 ... 4096		АЦП канал 3
IR 4	30005	0 ... 4096		АЦП канал 4
IR 5	30006	0 ... 4096		АЦП канал 5
IR 6	30007	0 ... 4096		АЦП канал 6
IR 7	30008	0 ... 4096		АЦП канал 7
IR 8	30009	0 ... 4096		АЦП канал 8
IR 11	30012	-32768...32767	$Y=X*K/N+B$ канал 1	$Y=X*K/N+B$ канал 1
IR 12	30013	-32768...32767	$Y=X*K/N+B$ канал 2	$Y=X*K/N+B$ канал 2
IR 13	30014	-32768...32767		$Y=X*K/N+B$ канал 3

IR 14	30015	-32768...32767		$Y=X*K/N+B$ канал 4
IR 15	30016	-32768...32767		$Y=X*K/N+B$ канал 5
IR 16	30017	-32768...32767		$Y=X*K/N+B$ канал 6
IR 17	30018	-32768...32767		$Y=X*K/N+B$ канал 7
IR 18	30019	-32768...32767		$Y=X*K/N+B$ канал 8

АЦП и входные цепи могут шуметь и выходные данные будут постоянно меняться. Для увеличения точности показаний можно использовать усреднение. Коэффициент усреднения задается в регистре HR75. И рассчитывается по формуле:

$$АЦП_{ср} = \frac{АЦП_1 + \dots + АЦП_n}{n};$$

Складывается указанное количество измерений АЦП и делится на их количество.

При этом повышается точность, уменьшается колебание значений, но увеличивается время измерения значения. Так же увеличивается время реагирования на дискретные входы в регистре DI1, DI8.

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

Регистр	Адрес	Диапазон	Описание регистра
HR 75	40076	1...255	Коэффициент усреднения АЦП

При отключенных входах, на них будет присутствовать +5 В (входы подтянуты резистором 4.7 или 22 кОм к напряжению +5В) и дискретные входы будут показывать 1. При замыкании датчика на общий G, на входах будет 0 и дискретные входы будут показывать 0. Таким образом, если на вход подключена кнопка, то при нажатой кнопке вход покажет состояние 0.

5.3. Коэффициенты аналоговых входов.

Каждый аналоговый вход выводит значение АЦП в регистры IR1 – IR8. Затем эти значения пересчитываются в физические величины по формуле уравнения прямой:

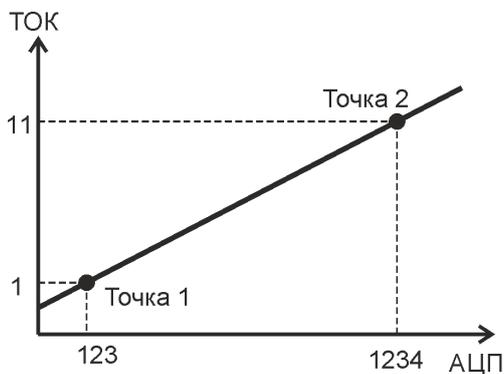
03 Read Holding Registers (HR), 06 Write Single Register, 16 Write Multiple registers.

Регистр	Адрес	Диапазон	Регистры DRM21R	Регистры DRM88R
HR 40	40041	-32768...32767	Канал №1, Коэффициент К	Канал №1, Коэффициент К
HR 41	40042	-32768...32767	Канал №1, Коэффициент N	Канал №1, Коэффициент N
HR 42	40043	-32768...32767	Канал №1, Коэффициент В	Канал №1, Коэффициент В
HR 43	40044	-32768...32767	Канал №2, Коэффициент К	Канал №2, Коэффициент К
HR 44	40045	-32768...32767	Канал №2, Коэффициент N	Канал №2, Коэффициент N
HR 45	40046	-32768...32767	Канал №2, Коэффициент В	Канал №2, Коэффициент В
HR 46	40047	-32768...32767	-	Канал №3, Коэффициент К
HR 47	40048	-32768...32767	-	Канал №3, Коэффициент N
HR 48	40049	-32768...32767	-	Канал №3, Коэффициент В
HR 49	40050	-32768...32767	-	Канал №4, Коэффициент К
HR 50	40051	-32768...32767	-	Канал №4, Коэффициент N
HR 51	40052	-32768...32767	-	Канал №4, Коэффициент В
HR 52	40053	-32768...32767	-	Канал №5, Коэффициент К
HR 53	40054	-32768...32767	-	Канал №5, Коэффициент N
HR 54	40055	-32768...32767	-	Канал №5, Коэффициент В
HR 55	40056	-32768...32767	-	Канал №6, Коэффициент К
HR 56	40057	-32768...32767	-	Канал №6, Коэффициент N
HR 57	40058	-32768...32767	-	Канал №6, Коэффициент В
HR 58	40059	-32768...32767	-	Канал №7, Коэффициент К
HR 59	40060	-32768...32767	-	Канал №7, Коэффициент N
HR 60	40061	-32768...32767	-	Канал №7, Коэффициент В
HR 61	40062	-32768...32767	-	Канал №8, Коэффициент К

HR 62	40063	-32768...32767	-	Канал №8, Коэффициент N
HR 63	40064	-32768...32767	-	Канал №8, Коэффициент B

Полученные значения АЦП можно преобразовать по формуле: $\text{Значение} = \frac{ADC * K}{N} + B$; Результат расчета помещается в регистры IR11 – IR18. Коэффициенты хранятся в регистрах HR40 – HR45. Для расчета этих коэффициентов формулы уравнения прямой необходимо использовать две точки.

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



5.4. Калькулятор коэффициентов.

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

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

Регистр	Диапазон	Описание регистра
HR 70	1...8	номер канала
HR 71	-32768...32767	Параметр 1
HR 72	-32768...32767	Параметр 2 (запись в этот регистр запускает расчет и сохранение коэффициентов в указанный HR70 канал)
HR 73	-32768...32767	результат АЦП 1 (только чтение)
HR 74	-32768...32767	результат АЦП 2 (только чтение)

Последовательность действий следующая.

- 1) подключить датчик.
- 2) в регистр HR70 записать номер канала (1 ... 8), к которому подключен датчик.
- 3) установить датчик в калибровочную камеру.
- 4) после стабилизации значений вписать значение первого параметра в регистр HR71 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для первого параметра в регистр HR73.
- 5) изменить величину климатического параметра.
- 6) после стабилизации значений вписать значение второго параметра в регистр HR72 и нажать ввод. Вместе с записью значения запишется текущее значение АЦП для второго параметра в регистр HR74. Затем модуль рассчитает коэффициенты и перепишет эти параметры в регистры коэффициентов номера канала, указанного в HR70. После этого в регистрах IR11 ... IR18 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.

6. Информационные регистры.

04 Input registers (IR).

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

Информационные регистры служат для идентификации модуля и контроля внутреннего состояния.

<i>Регистр</i>	<i>Адрес</i>	<i>Диапазон данных</i>	<i>Назначение</i>
IR 9000	39001	0...65535	номер версии ПО
IR 9001	39002	0...65535	номер версии ПО
IR 9002	39003	0...1	Тип ПО
IR 9003	39004	0...255	Тип устройства: 7 (DRM88R), 18 (DRM21R)
IR 9004	39005	1...31	Дата: день месяца
IR 9005	39006	1...7	Дата: неделя
IR 9006	39007	1...12	Дата: месяц
IR 9007	39008	0...99	Дата: год
IR 9008	39009	0...23	Время: часы
IR 9009	39010	0...59	Время: минуты
IR 9010	39011	0...59	Время: секунды
IR 9011	39012	0...65535	Серийный номер
IR 9012	39013	0...65535	Серийный номер
IR 9013	39014	0...65535	Серийный номер
IR 9014	39015	0...65535	Серийный номер
IR 9015	39016	0...65535	Серийный номер
IR 9016	39017	0...65535	Серийный номер
IR 9017	39018	0...1	Детектор перехода нуля сети для реле
IR 9018	39019	0	Код ошибки
IR 9019	39020	0...1	Состояние кнопки
IR 9020	39021	0...65535	Случайное число
IR 9021	39022	0...65535	Счетчик наработки часов
IR 9031	39032	0...65535	Счетчик замыканий для реле 1
IR 9032	39033	0...65535	Счетчик замыканий для реле 2
IR 9033	39034	0...65535	Счетчик замыканий для реле 3
IR 9034	39035	0...65535	Счетчик замыканий для реле 4
IR 9035	39036	0...65535	Счетчик замыканий для реле 5
IR 9036	39037	0...65535	Счетчик замыканий для реле 6
IR 9037	39038	0...65535	Счетчик замыканий для реле 7
IR 9038	39039	0...65535	Счетчик замыканий для реле 8

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

7. Установка параметров конфигурации.

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

7.1. Установка даты и времени.

Регистр	Адрес	Диапазон	Описание регистра
HR 93	40094	1...31	Установка даты - День месяца
HR 94	40095	1...7	Установка даты - День недели
HR 95	40096	1...12	Установка даты - Месяц
HR 96	40097	2020...2099	Установка даты - Год, 2020 или 20
HR 97	40098	0...23	Установка времени - Часы
HR 98	40099	0...59	Установка времени - Минуты
HR 99	40100	0...59	Установка времени - Секунды

В регистры HR94 – HR99 можно установить новое значение даты и времени. Для установки даты и времени необходимо записать в регистр HR99 новое значение. Т.к. в модулях нет батарейки, то при отключении питания часы сбросятся. Для постоянной работы часов необходимо впасть батарейку или использовать внешний ИБП. Прочитать текущее время и дату можно из регистров: IR9004 – IR9010.

7.2. Коррекция часов.

Регистр	Адрес	Диапазон	Описание регистра
HR 4	40005	-1300...0...1300	Коррекция работы часов

Часы за синхронизированы от внутреннего RC генератора, поэтому точность не высокая. Часы можно немного замедлить или ускорить, записав в HR4 значение с минусом или плюсом.

7.3. Сохранение во flash память.

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

Регистр	Адрес	Диапазон	функция	Описание регистра
HR 92	400093	-	Запись	Запись во Flash
HR 92	400093	0...65535	Чтение	Количество циклов записи

Количество циклов записи во Flash ограничено значением 100000.

7.4. Перезагрузка.

Регистр	Диапазон	Описание регистра
HR 5678	Любое значение	Переагрузка модуля

Запись любого числа в регистр HR5678 произведет перезагрузку модуля.

8. Сценарии для внутренней логики.

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

С регистра HR100 записываются данные для сценариев. В модуле зарезервировано место для 100 сценариев. Для каждого сценария используется по 13 регистров R0 – R12. Шаг записи сценариев 20.

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

Для чтения и записи сценариев используются регистры Holding Registers (HR). Эти регистры доступны для чтения и записи. Часть регистров зафиксированы под определенные параметры. С адреса 100 до адреса 2100 зарезервировано место для записи сценариев с 0 по 99. Все указанные в таблице значения сохраняются во FLASH. Все неиспользуемые адреса не записываются и не сохраняются.

Сценарии внутри блока выполняются последовательно от 0 до 99. Затем циклично повторяется с нулевого сценария. При большом количестве сценариев может ощущаться задержка. Если сценариев используется мало, то последним сценарием можно использовать переход в начало GOTO 0. Или пропустить пустые сценарии этой же командой GOTO.

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

Адрес							Описание регистра
R0	R1	R2	R3	R4	...	R12	
HR100	HR101	HR102	HR103	HR104	...	HR112	Сценарий № 0 и его параметры
HR120	HR121	HR122	HR123	HR124	...	HR132	Сценарий № 1 и его параметры
HR140	HR141	HR142	HR143	HR144	...	HR152	Сценарий № 2 и его параметры
HR160	HR161	HR162	HR163	HR164	...	HR172	Сценарий № 3 и его параметры
HR180	HR181	HR182	HR183	HR184	...	HR192	Сценарий № 4 и его параметры
...
HR2080	HR2081	HR2082	HR2083	HR2084	...	HR2092	Сценарий № 99 и его параметры

Источники данных.

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

Значение	Тип данных	диапазон	Тип данных	Чтение запись
0	Const – константа, фиксированное число.	(-32768 ... +32767)	Int (-32768 ... +32767)	Только чтение.
1	Coils (Co) – 01 регистр Реле.	(0 ... 4)	Bool (0...1)	Чтение, запись.
2	Discrete Input (DI) – 02 дискретные входы.	(0 ... 8)	Bool (0...1)	Только чтение.
3	Holding Reg (HR) – 03 регистр параметров.	(0 ... 2660)	Int (-32768 ... +32767)	Чтение и запись.
4	Input Reg (IR) – 04 регистр аналоговых входов.	(0...9030)	Int (-32768 ... +32767)	Чтение и запись.
5	Timer – регистр таймера обратного отсчета.	(0...15)	Int (-32768 ... +32767)	Чтение и запись.
6	GOTO для перехода на другой сценарий.	(0 ... 127)	Int (-32768 ... +32767)	Только запись.
7	CALL для перехода на другой сценарий.	(0 ... 127)	Int (-32768 ... +32767)	Только запись.

Типы данных.

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

Пользовательские данные.

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

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

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

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

<i>Значение</i>	<i>Тип данных</i>	<i>Размещение</i>	<i>Тип данных</i>	<i>Чтение, запись</i>
IR0 ... IR999	пользовательские данные и входы	ОЗУ не сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.
Coil0 ... Coil23	Флаги и выходы	ОЗУ не сохраняются	bool (0 ... 1)	Чтение и запись.
HR82 ... HR91	пользовательские данные	FLASH сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.
HR100 ... HR1612	Данные сценариев	FLASH сохраняются	Int16 (-32768 ... +32767)	Чтение и запись.
HR2980 - HR2995	Доступ к Timer 0-15	ОЗУ	0 ... +65535	Чтение и запись.
HR3000 - HR3999	Доступ к IR0 – IR999	ОЗУ	-32768 +32767	Чтение и запись.

Таймеры.

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

Доступно чтение и запись. Разрядность таймера 16 бит, значение таймера от 0 до 65535.

<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.

В каждом сценарии в адресе от 100 до 112 записываются тип и параметры сценария. Далее адрес будет обозначаться R0 – тип, записанный в регистр 100 (для сценария 0), R1 – параметр 1, записанный в регистр 101, R2 – параметр 2, записанный в регистр 102 и так далее.

Регистр R0 – Тип сценария

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

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

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

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

регистр	Описание
R0	1 - Математическая операция
R1	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Регистр результата
R3	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R4	Операнд 1
R5	Операция
R6	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Операнд 2

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

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

Регистр R5 - Операция:

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

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

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

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

FLOAT - Математические операции с дробными значениями

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

регистр	Описание
R0	2 - Математическая операция
R1	Тип выходного регистра: 1-Coils, 3-HR, 4-IR, 5-Timer
R2	Регистр результата
R3	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R4	Операнд 1
R5	Операция
R6	Тип входного регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Операнд 2

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

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

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

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

Регистр R5 - Операция:

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

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

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

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

PTRW – указатель для записи массивов переменных

Сценарий позволяет записывать данные в массив данных.

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

Входные данные: R4, R5;

Выходные данные: R1, R2, R3;

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

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

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

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

PTRR – указатель для чтения массивов переменных

Сценарий позволяет читать данные из массива данных.

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

Выходные данные: R4, R5;

Входные данные: R1, R2, R3;

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

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

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

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

FOR – циклы

Сценарий позволяет создать циклы.

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

Входные данные: R1, R2, R3, R4;

Выходные данные: R5;

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

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

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

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

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

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

04: GOTO 01; переход в начало цикла

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

Формат записи: FOR (IR (R1) = (R2.3) TO (R2.4)) GOTO (R5)

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

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

SEL – Бинарный выбор. Мультиплексор.

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

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

Выходные данные: R1, R2;

Условие: R3, R4;

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

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

Если R4 = 0, тогда R2 = R6.

Если R4 = 1, тогда R2 = R8.

Если R4 = 2, тогда R2 = R10.

Если R4 = 3, тогда R2 = R12.

Формат записи: R1.2 = SEL(R3.4) 0:R5.6; 1:R7.8; 2:R9.10; 3:R11.12

Пример записи: **IR 10 = SEL (DI 1) 0:HR 22; 1:HR 23; 2:HR 24; 3:HR 25**

IF - Логическая операция

Сценарий выполняет логическое условие ЕСЛИ (IF).

регистр	параметр
R0	7 - IF (R2 условие R5) тогда R6=R8 иначе R6=R9
R1	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Входной операнд 1
R3	Функция: 0- «==», 1- «!=», 2- «>», 3- «<», 4- «>=», 5- «<=», 6- «!», 7- «&&», 8- « »
R4	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R5	Входной операнд 2
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer, 6-GOTO, 7-CALL
R7	Выходной регистр
R8	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R9	Входной операнд если верно
R10	Действие: 0 – однократно; 1 – регулярно; 2 – однократно, иначе; 3 - регулярно, иначе
R11	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R12	Входной операнд если не верно

Входные данные: R1, R2, R4, R7, R8, R11, R12; Выходные данные: R6, R7; Функция: R3;
 Логика работы сценария «Логические операции» следующая:
 IF (R1.2 условие(R3) R4.5) THEN R6.7=R8.9 ELSE R6.7=R11.12. Действие выполняется: если R10=0, то однократно или если R10=1, то регулярно без использования ELSE. Действие выполняется: если R10=2, то однократно или если R10=3, то регулярно с использованием ELSE.

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

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

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

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

IFAND - Логическая операция двойная с И

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

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

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

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

Выходные данные: R9, R10;

Функция: R3, R7;

Логика работы сценария «Логические операции» следующая:

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

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

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

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

IFOR - Логическая операция двойная с ИЛИ

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

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

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

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

Выходные данные: R9, R10;

Функция: R3, R7;

Логика работы сценария «Логические операции» следующая:

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

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

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

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

GOTO - Переход:

Сценарий выполняет переход на другую команду.

регистр	параметр
R0	10 - GOTO
R1	Переход (возможные значения 0-99)

Входные данные: нет;

Выходные данные: R1;

Команда GOTO переход позволяет перепрыгнуть несколько сценариев. Сценарии выполняются последовательно от 0 до 99 и снова повторяются. Команда GOTO может выполняться совместно с условием IF. Если номер сценария указан больше 99, 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 0**

Сценарий	R1
10 - GOTO	Число: 0...99
Переход	Номер счетчика команд

CALL – Переход в подпрограмму:

Сценарий выполняет переход на подпрограмму.

регистр	Параметр
R0	11 – CALL
R1	Переход (возможные значения 0-99)

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 99, то будет переход на номер 0.

Формат записи: CALL (R1)

Пример записи: 00: **CALL 33**

Сценарий	R1
11 - CALL	Число: 0...99
Вызов подпрограммы	Номер счетчика команд

RETURN – Выход из подпрограммы:

Сценарий выполняет выход из подпрограммы.

регистр	Параметр
R0	12 – RETURN

Логика работы сценария «Переход» следующая: RETURN.

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

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

3: MATH IR28 = IR38 * IR7

4: MATH IR7 = DI4 - IR7

5: BITS R12 = R4 & R7

6: MOV Coil3=R12

99: GOTO 0; переход в начало алгоритма

100: MOV IR7 = 25

101: MATH IR38 = DI4 * IR7

102: RETURN; возврат из подпрограммы

Формат записи: RETURN

Пример записи: 00: **RETURN**

Сценарий
12 - RETURN
Выход из подпрограммы

THRS - Пороговое реле:

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

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

Входные данные: R3, R4;

Выходные данные: R1, R2;

Коэффициенты: R5, R6, R7, R8.

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

Логика работы сценария «Пороговое реле» следующая.

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

Если значение больше заданного, то включает реле.

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

Если значение меньше заданного, то выключает реле.

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

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

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

TRG - Триггер:

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

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

Входные данные: R3, R4;

Выходные данные: R1, R2;

Параметры: R5.

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

Логика работы сценария «Триггер» следующая:

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

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

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

KDO – кнопка с удержанием.

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

Регистр	параметр
R0	15 – кнопка с удержанием
R1	Тип регистра: 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R2	Номер входного регистра
R3	Номер реле Coil при кратковременном нажатии
R4	Тип регистра: 1-Coils, 3-HR, 13-HR, 23-HR, 4-IR, 5-Timer
R5	Номер выходного регистра при удержании кнопки
R6	Значение, присеваемое регистру R5.
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: KDO: (R1.2) = (R7) dsec -> (Coil R3) -> (R4.5)=(R6)

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

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

KDB - кнопка с удержанием.

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

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

Входные данные: R1, R2;

Выходные данные: R3, R4;

Параметры: R5

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

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

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

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

TIMER - Таймер.

Сценарий позволяет выполнять события по таймеру.

регистр	параметр
R0	17 - Таймер
R1	Номер таймера
R2	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R3	Максимальное значение таймера
R4	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer, 6-GOTO, 7-CALL
R5	Выходной регистр или переход на другой сценарий
R6	Тип регистра: 0- Const, 1-Coils, 2-DI, 3- HR, 4-IR, 5-Timer
R7	Входной сценарий

Входные данные: R6, R7;

Выходные данные: R4, R5;

Параметры: R1, R2, R3.

Логика работы сценария «Таймер» следующая:

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

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

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

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

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

Сценарий	R1	R2	R3	R4	R5	R6	R7
17 - TIM	Число Тип - Timer	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R2	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim 6-GOTO 7-CALL	Число Тип в R4	0-Const 1-Coils 2-DI 3-HR 4-IR 5-Tim	Число Тип в R6
Таймер	Номер таймера TIMER	тип	Значение таймера	тип	Вых регистр	тип	Входной сценарий

RTC – часы и дата

Сценарий позволяет выполнить действия при наступлении заданного времени

регистр	параметр
R0	18 – RTC
R1	тип: 0 – нет; 1 – Ежемесячно; 2 – Еженедельно; 3 – Ежедневно; 4 - Каждый час; 5 - Каждую минуту
R2	День – для «ежемесячно» или маска недели – для «еженедельно»;
R3	часы (0-23),
R4	минуты (0-59),
R5	секунды (0-59),
R6	Тип регистра: 1-Coils, 3- HR, 4-IR, 5-Timer, 6-GOTO, 7-CALL
R7	Выходной регистр
R8	Тип регистра: 0...5
R9	Входной операнд

Входные данные: R8, R9;

Выходные данные: R6, R7;

Параметры: R1 – R5.

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

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

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

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

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

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

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

R0 = 9;

R1 = 2 тип;

R2 = 1 маска недели;

R3 = 10 час;

R4 = 11 минута;

R5 = 12 секунда;

R6 = тип 6 переход;

R7 = 25 адрес перехода

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

R0 = 9;

R1 = 4 тип;

R2 = 0;

R3 = 20 час;

R4 = 21 минута;

R5 = 0 секунда;

R6 = 1 тип Coils;

R7 = 1 номер реле;

R8 = 0 константа;

R9 = 1 включение реле.

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

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

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

Маска дней недели (для таймера типа - Еженедельно):

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

JAL - Управление Жалюзи.

Сценарий позволяет организовать процесс управления приводом, в частности жалюзи.

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

Входные данные: R1, R2;

Выходные данные: R4, R5;

Параметры: R3, R6.

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

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

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

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

PID - ПИД регулятор.

Сценарий позволяет организовать пропорционально интегрирующее дифференцирующее регулирование нагревательным элементом.

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

Входные данные: R2, R3, R4;

Выходные данные: R1;

Результат: R12;

Параметры: R5, R6, R7, R8.

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

P = Kp*DE; Пропорциональная составляющая;

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

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

DE_last = DE; предыдущее значение разности значений;

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

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

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

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

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

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

8. ПО для управления и настройки блоков

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

8.1. программа ModbusPoll

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

The screenshot shows the Modbus Poll interface with several data windows. The top row contains five windows for registers: Tx = 74, Tx = 79, Tx = 77, Tx = 63, and Tx = 53. Each window displays a table with columns for Alias, value, and status. The bottom window shows a larger table for registers with columns for Alias, value, and type (e.g., Таймер, номер, тип рег, etc.).

8.2. программа RDControl Modbus 4

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

The screenshot shows the RDCMB4.1 interface. On the left, there is a tree view for 'Устройства' (Devices) with '1 (DRM88Rv3)' selected. The main area is titled 'Сценарии' (Scenarios) and contains a table with columns: №, Сценарии, Входные данные, and Комментарии. The table lists 23 scenarios, including triggers, thresholds, math operations, and logic-based actions. At the bottom, there are buttons for 'Считать' (Read) and 'Записать' (Write), and a status bar indicating 'Настройки порта: COM4, 9600, 8, None, 2'.