

ООО «Новая автоматика»
302025 Россия, Орловская область, г.Орёл,
Московское шоссе, дом № 137, корпус 4, помещение 20
Тел.: +7(977)946-45-03, 8(804)333-74-73
E-mail : info-na @ mail.ru
[http:// www.new-automatics.ru](http://www.new-automatics.ru)

МКЗ

**УСТРОЙСТВО УПРАВЛЕНИЯ И ЗАЩИТЫ
K57.01.00.00.00**

Описание протокола Modbus

Версия 2.07

СОДЕРЖАНИЕ:

1. О документе.....	3
1.1 Содержание документа	3
1.2 Ссылки	3
1.3 Термины и сокращения.....	3
2. Описание реализации.....	3
2.1 Интерфейс.....	3
2.2 Адреса устройств Modbus	3
2.3 Команды.....	3
2.4 Адресация.....	3
2.5 Режим передачи	4
3. Описание регистров.....	5
3.1 Чтение состояния дискретных выходов (Read Coils).....	5
3.2 Чтение состояния дискретных входов (Read Discrete Inputs).	7
3.3 Регистры задания параметров объекта (Holding Register).	9
3.4 Регистры чтения данных объекта (Input Register).....	17
4. Исключительные ситуации	23
5. Задержки между пакетами	23
6. Генерация контрольной суммы	24
6.1 Контрольная сумма LRC.....	24
6.2 Контрольная сумма CRC.....	25

Что нового:

Версия 2.00

В аппаратную часть МК3.03 добавлен модуль измерения и контроля температуры

Новые параметры установочного меню 242-245:

- Регистр 242 - Модуль термодатчика
- Регистр 243 - Тип термодатчика
- Регистр 244 - Температура включения, °С
- Регистр 245 - Температура отключения, °С

Регистры состояния:

- Регистр 351 - Сопротивление цепи датчика, Ом
- Регистр 352 - Состояние термодатчика
- Регистр 353 - Температура, °С
- Регистр 354 - Состояние терморегулятора

Версия 2.07

Увеличено время таймеров задержки включения/отключения в Регистрах 108 и 109 с 600 до 1200 секунд.
В Регистре 127 добавлен новый тип датчика тока - Т04-200(2000:1)

1. О документе

1.1 Содержание документа

Документ описывает реализацию протокола Modbus (режим передачи RTU и ASCII) в устройствах управления и защиты трехфазных асинхронных электродвигателей серии МК3 (далее по тексту - контроллер) производства ООО «Новая автоматика». Содержится вся необходимая информация для программистов при подключении контроллеров к SCADA системам или при создании распределенных систем автоматике.

1.2 Ссылки

Данный документ ссылается на следующие документы:

1. Modbus Application Protocol Specification v1.1b (www.modbus.org)
2. Modbus messaging on TCP/IP implementation Guide Rev 1.0 (www.modbus.org)
3. Modbus over Serial Line Specification & Implementation guide V1.0 (www.modbus.org)

1.3 Термины и сокращения

- RS-232 – стандарт EIA/TIA-232;
RS-485 – стандарт EIA/TIA-485 Standard.

2. Описание реализации

2.1 Интерфейс

Контроллер имеет последовательные интерфейсы RS-485 и/или RS-232. Для организации сети из двух и более приборов можно использовать преобразователь интерфейсов 232/485. Интерфейс RS-485 позволяет объединить в сеть до 128 устройств на линии длиной до 1200 м. Контроллер является ведомым (slave) устройством, отвечающим на команды с соответствующим адресом в пакете протокола.

По последовательным интерфейсам поддерживается протокол верхнего уровня Modbus с форматом пакета RTU или ASCII в полном соответствии с документом «Modbus over Serial Line Specification & Implementation guide V1.0». Поддерживаются скорости передачи от 2400 бит/с до 256000 бит/с с контролем четности, 7 или 8 бит данных, 1 или 2 стоп-бита. Физический интерфейс, скорость соединения и сетевой адрес задаются при программировании контроллера. Максимальное время ожидания ответа составляет не более 100 мс.

2.2 Адреса устройств Modbus

Все устройства серии поддерживают команды Modbus в полном соответствии с синтаксисом запроса и ответа, определенным в документе «Modbus Application Protocol Specification v1.1b». Поддерживаются запросы к конкретным устройствам по их адресам, широковещательный режим не поддерживается. Адрес устройства может быть от 01h до F7h. Диапазон адресов F8h-FFh зарезервирован в стандарте Modbus.

2.3 Команды

Микроконтроллер поддерживает следующие команды:

- **01h** Чтение состояния дискретных выходов (Read Coils)
- **02h** Чтение состояния дискретных входов (Read Discrete Inputs)
- **03h** Чтение регистров настройки (Read Holding Registers)
- **04h** Чтение входных регистров (Read Input Register)
- **05h** Установка единичного дискретного выхода (Write Single Coil)
- **06h** Запись регистра настройки (Write Single Register)
- **10h** Запись нескольких регистров настройки (Write Multiple Registers)

Наличие команд 07h, 08h, 0Bh, 0Ch, 0Fh, 11h, 14h, 15h, 16h, 17h, 18h не обязательна. Поддержка конкретным устройством команды из приведенного выше списка отражается отдельно.

2.4 Адресация

Адреса запрашиваемых регистров и битов по протоколу Modbus и адреса в памяти устройства имеют однозначное табличное соответствие, но не совпадают. Таблица соответствия адресов задается программистом и должна быть отражена в документации к устройству. По одному интерфейсу может быть доступно не более 32767 байт адресуемых регистров переменных, 32767 байт регистров входов и 8192 байт, доступных через битовых переменные.

Основным способом передачи данных по протоколу Modbus является чтение или запись регистров. Реализация протокола поддерживает как побайтную адресацию, так и пословную. Адресация битовых массивов данных полностью соответствует стандарту Modbus. Максимально возможное количество битов, передаваемых в одном пакете, не может быть более 256.

2.5 Режим передачи

В протоколе Modbus существуют два режима передачи. Это ASCII (American Standard Code for Information Interchange) и RTU (Remote Terminal Unit). Режим выбирается пользователем, в зависимости от используемого в сети оборудования. Для каждой сети Modbus должен использоваться только один режим. Использование смешанных режимов в одной сети не допускается.

Структура кадра сообщения в режиме ASCII

Семибитовый код ASCII был разработан как универсальный код для отображения символов английского языка для телетайпов и является принятым в США стандартом для представления символов английского языка и управляющих символов, например, CR (возврат каретки) и LF (перевод строки). Наименования этих символов сохранились со времен телетайпов и сейчас просто указывает на конец кадра.

Преимуществом данного режима является то, что если в качестве ведомого устройства включить монитор, то можно увидеть на экране понятный человеку отформатированный код, который послан ведущим устройством на экран монитора. На Рисунке 1 показана структура сообщения Modbus в режиме ASCII. Его начало обозначается символом «:»(3Ah), а конец - последовательностью CR/LF (два символа ASCII - 0Dh, 0Ah).

Начало	Адрес устройства	Код функции	Данные	Контрольная сумма (LRC)	Конец
1 символ (:)	2 символа	2 символа	N символов	2 символа	2 символа (CR+LF)

Рисунок 1 Структура кадра сообщения Modbus ASCII

Любой символ ASCII представляется 7 битами. Символы должны быть либо цифрами от 0 до 9, либо буквами от A до F, так как предполагается, что данные представляются в шестнадцатеричном формате, но отображаются в виде символов ASCII. Например, код функции 03 будет отображаться двумя ASCII-символами: «0»(30h) и «3»(33h). То же самое относится и к содержимому поля данных.

Одним из преимуществ режима ASCII является то, что он не предъявляет жестких требований к синхронизации. Допускается временной промежуток между символами до 1 секунды - только по истечении его генерируется сообщение о превышении лимита времени.

Структура кадра сообщения в режиме RTU

При работе в режиме RTU синхронизация имеет более важное значение, чем в режиме ASCII. В этом варианте специальный начальный символ отсутствует. Вместо этого кадр сообщения начинается с маркерного интервала, длительность которого равна времени передачи четырех символов. После истечения этого интервала передается адрес устройства, затем код функции и собственно данные. Имеются и другие отличия от кадра сообщения в режиме ASCII, как это показано на рис. 2.

Начало	Адрес устройства	Код функции	Данные	Контрольная сумма CRC	Конец
Интервал, равный времени передачи 4-х символов	8 бит	8 бит	N*8 бит	16 бит	Интервал, равный времени передачи 4-х символов

Рис. 2. Структура кадра сообщения Modbus RTU

Вместо контрольной суммы LRC (Longitudinal Redundancy Check - продольный контроль по избыточности) в режиме RTU используется контрольная сумма CRC (Cyclic Redundancy Check - циклический контроль по избыточности). Конец кадра отмечается маркерным интервалом, равным времени передачи четырех символов.

RTU-сообщения должны посылаться в виде непрерывного потока, и появление значительного временного интервала между смежными символами рассматривается как окончание сообщения. Сообщения в этом режиме весьма компактны и более эффективны, чем ASCII, с точки зрения их передачи. Поэтому режим RTU является более популярным.

3. Описание регистров.

3.1 Чтение состояния дискретных выходов (Read Coils).

Чтение дискретных выходов производится командой **01h (Read Coils)**.

Функция позволяет пользователю получить статус (1/0) логических ячеек. Широковещательный режим не поддерживается. Помимо полей адреса и функции, сообщение требует, чтобы информационное поле содержало логический адрес первой ячейки и число ячеек, статус которых необходимо получить.

Если возвращаемое количество выходов не кратно восьми, то оставшиеся старшие биты в последнем байте сообщения будут установлены в «0».

Адрес (dec)	Описание	Состояние
000	Реле включения двигателя К1	1: Включено 0: Отключено
001	Реле универсальное К2	
002	Реле универсальное К3	
003	Реле универсальное К4	
004	Сброс аварии	1 (Включено): команда сброса 0 (Отключено): не влияет
005	Сброс (перезагрузка) контроллера	
006	Постановка объекта на охрану	1 (Включено): команда постановки 0 (Отключено): не влияет
007	Снятие объекта с охраны	1 (Включено): команда снятия 0 (Отключено): не влияет
008-049	Резерв	

Примечание - при чтении выходов по адресам 004 – 007 их значение всегда будет нулевым.

Чтение состояния 8 выходов с адреса 000.

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	01	Функция	01
Начальный адрес старший	00	Счетчик байт	01
Начальный адрес младший	00	Данные выхода 007 - 000	02
Количество выходов старший	00	CRC	D0 49
Количество выходов младший	08		
CRC	3D CC		

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 31	Функция	30 31
Начальный адрес старший	30 30	Счетчик байт	30 31
Начальный адрес младший	30 30	Данные выхода 007 - 000	30 32
Количество выходов старший	30 30	LRC	46 42
Количество выходов младший	30 38	Конец кадра (CR+LF)	0D 0A
LRC	46 36		
Конец кадра (CR+LF)	0D 0A		

Статус выходов 007 - 000 = 0x02h = 00000010.

Читая слева направо, видим, что выход 001 (Реле К2) в состоянии «Включено», остальные в состоянии «Отключено».

Чтение состояние одного выхода по адресу 003 (Реле K4).

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	01	Функция	01
Начальный адрес старший	00	Счетчик байт	01
Начальный адрес младший	03	Данные выхода 003	00
Количество выходов старший	00	CRC	51 88
Количество выходов младший	01		
CRC	0D CA		

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 31	Функция	30 31
Начальный адрес старший	30 30	Счетчик байт	30 31
Начальный адрес младший	30 33	Данные выхода 003	30 30
Количество выходов старший	30 30	LRC	46 44
Количество выходов младший	30 31	Конец кадра (CR+LF)	0D 0A
LRC	46 41		
Конец кадра (CR+LF)	0D 0A		

В младшем разряде – состояние бита: 0 (реле K4 отключено), в остальных нули, независимо от их состояния. При запросе любого единичного выхода в младшем разряде будет его состояние, а в остальных нули.

Установка единичного дискретного выхода производится командой **05h** (Write Single Coil).

Запрос содержит адрес устройства, номер функции, адрес выхода (2 байта) и состояние, в которое его необходимо установить (2 байта).

Значение FF 00 (hex) – состояние «**Включено**», значение 00 00 (hex) – состояние «**Отключено**». Любое другое значение неверно и не влияет на выход.

Запись состояния «Включено» дискретного выхода по адресу 003 (Реле K4). Нормальный ответ повторяет запрос.

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	05	Функция	05
Начальный адрес старший	00	Начальный адрес старший	00
Начальный адрес младший	03	Начальный адрес младший	03
Значение выхода старший	FF	Значение выхода старший	FF
Значение выхода младший	00	Значение выхода младший	00
CRC	7C 3A	CRC	7C 3A

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 35	Функция	30 35
Начальный адрес старший	30 30	Начальный адрес старший	30 30
Начальный адрес младший	30 33	Начальный адрес младший	30 33
Значение выхода старший	46 46	Значение выхода старший	46 46
Значение выхода младший	30 30	Значение выхода младший	30 30
LRC	46 38	LRC	46 38
Конец кадра (CR+LF)	0D 0A	Конец кадра (CR+LF)	0D 0A

Примечание: Изменение состояния выхода по адресу 000 «Реле включения двигателя К1» возможно только при режиме работы - «Управление по RS-485». В других режимах попытки записи значений будут игнорироваться.

Изменение состояния выходов по адресам 001...003 «Реле универсальное К2...К4» возможно только при установке функций каждого реле - «Управляется по RS-485».

Сброс аварии, перезагрузка контроллера, постановка на охрану и снятие объекта с охраны осуществляется записью состояния «Включено» в соответствующие регистры. Значение этих выходов при чтении всегда нулевое.

3.2 Чтение состояния дискретных входов (Read Discrete Inputs).

Чтение дискретных входов производится командой **02h (Read Discrete Inputs)**.

Данная функция позволяет пользователю получить состояние («Замкнут» или «Разомкнут») дискретных входов и аварийных битовых сигналов адресуемого контроллера. Широковещательный запрос не поддерживается.

Запрос содержит адрес устройства, номер функции, начальный адрес (2 байта) и количество требуемых входов (2 байта). Статус входов в ответном сообщении передается как один выход на бит.

Если возвращаемое количество входов не кратно восьми, то оставшиеся старшие биты в последнем байте сообщения будут установлены в 0. Счетчик байт содержит количество байт, передаваемых в поле данных.

Адрес (dec)	Описание	Состояние
050	Дискретный вход DI.1	0: вход разомкнут 1: вход замкнут
051	Дискретный вход DI.2	
052	Дискретный вход DI.3	
053	Дискретный вход DI.4	
054	Дискретный вход DI.5	
055	Дискретный вход DI.6	
056	Дискретный вход DI.7	
057	Дискретный вход DI.8	
058	Сигнал повышения напряжения	0: нет аварийного сигнала 1: есть аварийный сигнал
059	Сигнал понижения напряжения	
060	Сигнал перекоса фаз по напряжению	
061	Сигнал повышения тока	
062	Сигнал понижения тока	
063	Сигнал перекоса фаз по току	
064	Сигнал неисправности аналогового входа	
065	Сигнал о замыкании (утечке) на корпус	
066-099	Резерв	

Чтение 3 входов с адреса 053.

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	02	Функция	02
Начальный адрес старший	00	Счетчик байт	01
Начальный адрес младший	35	Данные входов 055 – 053	02
Количество входов старший	00	CRC	20 49
Количество входов младший	03		
CRC	28 05		

Статус входов 055 – 053 = 0x02h = 0000 0010.

Читая с нулевого бита, определяем состояние входов 055 – 053.

53 – «0», 54 – «1», 55 – «0», далее идут нули, независимо от состояния входов.

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 32	Функция	30 32
Начальный адрес старший	30 30	Счетчик байт	30 31
Начальный адрес младший	33 35	Данные входов 055 – 053	30 32
Количество входов старший	30 30	LRC	46 41
Количество входов младший	30 33	Конец кадра (CR+LF)	0D 0A
LRC	43 35		
Конец кадра (CR+LF)	0D 0A		

Чтение 8 входов с адреса 050.

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	02	Функция	02
Начальный адрес старший	00	Счетчик байт	01
Начальный адрес младший	32	Данные входов 057 – 050	10
Количество входов старший	00	CRC Lo	A0 44
Количество входов младший	08		
CRC	D8 03		

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 32	Функция	30 32
Начальный адрес старший	30 30	Счетчик байт	30 31
Начальный адрес младший	33 32	Данные входов 057 – 050	31 30
Количество входов старший	30 30	LRC	45 43
Количество входов младший	30 38	Конец кадра (CR+LF)	0D 0A
LRC	43 33		
Конец кадра (CR+LF)	0D 0A		

Статус входов 057 – 050 = 0x10h = 0001 0000. В младшем бите состояние регистра 050.

Чтение 16 входов с адреса 050.

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	02	Функция	02
Начальный адрес старший	00	Счетчик байт	02
Начальный адрес младший	32	Данные входов 057 – 050	10
Количество входов старший	00	Данные входов 065 – 058	40
Количество входов младший	10	CRC	B5 88
CRC	D8 09		

Статус входов 057 – 050 = 0x10h = 0001 0000. В младшем бите состояние регистра 050.

Статус входов 065 – 058 = 0x40h = 0100 0000. В младшем бите состояние регистра 058.

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 32	Функция	30 32
Начальный адрес старший	30 30	Счетчик байт	30 32
Начальный адрес младший	33 32	Данные входов 057 – 050	31 30
Количество входов старший	30 30	Данные входов 065 – 058	34 30
Количество входов младший	31 30	LRC	41 42
LRC	42 42	Конец кадра (CR+LF)	0D 0A
Конец кадра (CR+LF)	0D 0A		

3.3 Регистры задания параметров объекта (Holding Register).

Чтение регистров производится командой **03h (Read Holding Register)**.

Данная функция позволяет получить двоичное содержимое 16-ти разрядных регистров адресуемого контроллера. Адресация позволяет получить за каждый запрос до 125 регистров. Широковещательный режим не допускается.

Адресуемый контроллер посылает в ответе свой адрес, код выполненной функции и информационное поле. Информационное поле содержит 2 байта, описывающих количество возвращаемых байт данных. Длина каждого регистра данных – 2 байта. Первый байт данных в посылке является старшим байтом регистра, второй – младшим.

Адрес (dec)	Название	Диапазон допустимых значений		
		Тип переменной	Диапазон значений	Реальное значение
100	Команды диагностики. Не использовать!			
101	Режим работы		0 – автомат по датчикам уровня, 1 – автомат по таймеру и датчику dL (dH), 2 – управление по линии связи RS-485, 3 – SMS-управление.	
102	Функция управления процессом		0 – прямая (налив), 1 – обратная (дренаж).	
103	Максимальная уставка давления (уровня), (бар)(м)(°C)(м3/ч)(л/с)(%) (МПа)	int	10 : 65000	0,10 : 650,00
104	Минимальная уставка давления (уровня), (бар)(м)(°C)(м3/ч)(л/с)(%) (МПа)		10 : 65000	0,10 : 650,00
105	Время работы таймера, минут	char	1:180	
106	Задержка пуска после подачи питания, секунд		0:180	
107	Таймер аварийного отключения, минут		0:240	
108	Таймер задержки включения, секунд		0:1200	
109	Таймер задержки отключения, секунд		0:1200	
110	Секунды	char	0:59	
111	Минуты		0:59	
112	Часы		0:23	
113	Дата		1:31	
114	Месяц		1:12	
115	Год		int	2020 : 2100
116	Коррекция хода часов реального времени за 10 суток, секунд	signed char	-100 : +100	

117	Сброс пользовательских моточасов	20 - сброс моточасов		
118	Сброс пользовательского счетчика количества запусков двигателя	25 - сброс счетчика		
119	Установка заводских значений	50 - сброс на заводские уставки		
120	Тип датчиков уровня (давления)	0 - дискретные датчики уровня (давления), 1 - аналоговый датчик давления (уровня).		
121	Тип сигнала аналогового датчика	0 – (0...20мА), 1 – (4...20мА).		
122	Диапазон аналогового датчика, (бар)(м)(°C)(м ³ /ч)(л/с)(%)	int	100 : 65000	1,00 : 650,00
123	Единица измерения сигнала аналогового датчика	char	0 - бар, 1 - метры, 2 - °C, 3 - м ³ /час,	4 - л/сек, 5 - %, 6 - МПа.
124	Фиксированное значение 1 аналогового датчика давления (уровня), мА	int	0 : 2000	0,00 : 20,00
125	Фиксированное значение 2 аналогового датчика давления (уровня), мА	int	0 : 2000	0,00 : 20,00
126	Проверка дискретного датчика «сухого» хода	0 – не проверяется, 1 – проверяется всегда, 2 – проверяется после запуска.		
127	Тип датчика (трансформатора) тока	0 – тип Т03-120 (измеряемый ток до 120А), 1 – с унифицированным токовым выходом 5А. 2 – тип Т04-200 (измеряемый ток до 200А)		
128	Диапазон трансформатора тока с унифицированным токовым выходом(5А), А	int	50 : 10000	5,0 : 1000,0
129	Сигнал «Внешнее управление» (E.Run)	0 – не используется, 1 – разрешение работы по датчикам уровня, 2 – принудительное включение нагрузки.		
130	Разрешить вход внешней аварии (E.Error)	0 – нет, 1 – да.		
131	Использование охранной сигнализации			
132	Функция универсального реле К2	0 – работа (питание подано), 1 – авария, 2 – двигатель Включен/Отключен, 3 – состояние входа DI.1, 4 – состояние входа DI.2, 5 – состояние входа DI.3, 6 – состояние входа DI.4, 7 – состояние входа DI.5, 8 – состояние входа DI.6, 9 – состояние входа DI.7, 10 – состояние входа DI.8, 11 – ручной / автоматический режим, 12 – несанкционированный доступ, 13 – достижение фиксир. значения в mA-1, 14 – достижение фиксир. значения в mA-2, 15 – управляется от ПК по RS-485, 16 – управляется Недельным Таймером, 17 – переключение «Звезда-Треугольник», 18 – реле не используется, 19 – управляется встроенным терморегулятором, 20 – минимальный уровень воды, 21 – максимальный уровень воды, 22 – аварийный уровень воды (перелив), 23 – кратковременное включение (3 сек.) при таймауте соединения с ПК или команд из СМС.		
133	Функция универсального реле К3			
134	Функция универсального реле К4			

135	Выбор переключателя режима «Ручной/Автомат»	0 – кнопка на лицевой панели МКЗ, 1 – дискретный вход DI.X с установленной функцией входа - «Auto».		
136	Вес импульса счетчика воды, литров	int	1 : 10000	
137	Подключаемый сигнал входа DI.1	0 – вход не используется, 1 – датчик верхнего уровня (dH), 2 – датчик нижнего уровня (dL), 3 – датчик «сухого» хода (dS), 4 – датчик аварийного уровня (dAV), 5 – датчик охранной сигнализации (Alarm), 6 – сигнал внешнего управления (E.Run), 7 – сигнал внешней аварии (E.Ergr), 8 – переключатель «Ручной/Автомат» (Auto), 9 – сигнал «Пуск/Стоп» (в ручном режиме), 10 – сигнал «Пуск» (в ручном режиме), 11 – сигнал «Стоп» (в ручном режиме), 12 – сигнал сброса аварии, 13 – импульсный сигнал со счетчика воды, 14 – импульсный сигнал с электросчетчика. 15 – Разрешение работы Недельного Таймера 1, 16 – Разрешение работы Недельного Таймера 2, 17 – Разрешение работы Недельного Таймера 3, 18 – Разрешение работы Недельного Таймера 4 , 19 – Разрешение работы всех Нед. Таймеров.		
138	Подключаемый сигнал входа DI.2			
139	Подключаемый сигнал входа DI.3			
140	Подключаемый сигнал входа DI.4			
141	Подключаемый сигнал входа DI.5			
142	Подключаемый сигнал входа DI.6			
143	Подключаемый сигнал входа DI.7			
144	Подключаемый сигнал входа DI.8			
145	Тип контакта входов	Бит		
		0	Дискретный вход DI.1	0 – н.з. 1 – н.о.
		1	Дискретный вход DI.2	
		2	Дискретный вход DI.3	
		3	Дискретный вход DI.4	
		4	Дискретный вход DI.5	
		5	Дискретный вход DI.6	
		6	Дискретный вход DI.7	
		7	Дискретный вход DI.8	
8-15	Резерв			
146	Максимальный ток (перегрузка), А	int	5 : 10000	0,5 : 1000,0
147	Минимальный ток (недогрузка), А		0 : 10000	0,0 : 1000,0
148	Значение перекоса фаз по току в %	char	0 : 40 (0 – отключает проверку)	
149	Максимальное напряжение, В	int	2300 : 2700	230,0 : 270,0
150	Минимальное напряжение, В		1600 : 2200	160,0 : 220,0
151	Значение перекоса фаз по напряжению, В		0 : 400	0,0 : 40,0
152	Разрешаемое количество пусков в час	char	0...100 (0 – отключает проверку)	
153	Время блокировки пускового тока, секунд	char	1 : 90	
154	Время срабатывания аварии, секунд		1 : 20	
155	Время выдержки после аварии, минут		1 : 60	
156	Время срабат. датчика «сухого» хода, секунд		1 : 120	
157	Время выдержки после «сухого» хода, минут		1 : 60	
158	Время выдержки после внешней аварии, минут		0 : 60 (0 – без выдержки времени)	
159	Блокировка включения после повторяющихся аварий в течение часа	0 – нет, 1 – да		
160	Количество повторяющихся подряд аварий в час для срабатывания блокировки	1 : 30		

161	Функция датчика аварийного уровня dAV (в автоматическом режиме работы)		0 – не используется, 1 – аварийный останов до снятия сигнала, 2 – аварийный останов с выдержкой времени, 3 – обработка сигнала «Перелив».	
162	Защита от заклинивания при длительном простое	Время простоя, часов	0 : 100 (0 – отключает защиту)	
163		Время, на которое будет запущен двигатель после простоя, секунд	1 : 30	
164	Проверка замыкания (утечки) на корпус		0 – нет, 1 – да	
165	Выбор протокола связи		0 – GSM-модем (AT-команды), 1 – Modbus RTU, 2 – Modbus ASCII	
166	Скорость передачи, бит/сек.		0 – 2400, 1 – 4800, 2 – 9600, 3 – 14400, 4 – 19200, 5 – 38400,	6 – 56000, 7 – 57600, 8 – 115200, 9 – 128000, 10 – 256000
167	Адрес устройства в сети Modbus		char	1 : 247
168	Максимальный сетевой таймаут (соединение по Modbus), секунд		int	0 : 600
169	Действие при таймауте соединения по Modbus		0 – ничего не предпринимать, 1 – авария по таймауту.	
170	Задействовать передачу sms		0 – нет, 1 – да	
171	Максимальный сетевой таймаут для управления по SMS, минут		int	0 : 360
172	Действие при таймауте соединений (управление по SMS)		0 – ничего не предпринимать, 1 – авария по таймауту	
173	Проверка номера телефона, с которого приходит управляющая sms		0 – нет, 1 – да	
174	Проверка превышения времени доставки sms более чем на 15 минут			
175 . . . 186	D1 ... D12	Номер телефона сотовой связи, на который будут отправляться sms	11 или 12 цифр, 0...9 Формат номера: +x (xxx) xxx-xx-xx По умолчанию код государства: +7 (РФ) При использовании 11-значного номера в ячейку D12(186) необходимо записать знач. 10 (0x0Ah)	
187 . . . 198	N1 ... N12	Идентификационное имя прибора при отправке sms, по умолчанию установлено «ШУ Оникс-001»	12 символов типа char (0x20h...0xFFh), соответствует кодировке UNICODE 16 Bit	

199	Автоматическая рассылка sms с показаниями счетчиков воды, времени наработки и количества запусков двигателя	Включение функции авторассылки sms	0 – нет, 1 – да		
200		Дни недели, в которые будут рассылаться sms с показаниями счетчиков	Бит		
			0	воскресенье	0 – снять, 1 – установить
			1	понедельник	
			2	вторник	
			3	среда	
			4	четверг	
			5	пятница	
6	суббота				
7	---				
201		Время отправки sms с показаниями счетчиков, часы	char	0 : 23	
202		Время отправки sms с показаниями счетчиков, минуты		0 : 59	
203	Недельный таймер 1	Включение таймера	char	0 – отключен, 1 – включен	
204		Выходное реле, управляемое данным таймером	char	1 – универсальное реле К2, 2 – универсальное реле К3, 3 – универсальное реле К4	
205		Дни недели	Бит		
			0	воскресенье	0 – снять, 1 – установить
			1	понедельник	
			2	вторник	
			3	среда	
			4	четверг	
			5	пятница	
6		суббота			
7	---				
206		Время включения, часы	char	0 : 23	
207		Время включения, минуты	char	0 : 59	
208		Время отключения, часы	char	0 : 23	
209		Время отключения, минуты	char	0 : 59	
210 – 216	Недельный таймер 2 (аналогичен таймеру 1)				
217 – 223	Недельный таймер 3 (аналогичен таймеру 1)				
224 – 230	Недельный таймер 4 (аналогичен таймеру 1)				
231	Связь	Количество бит данных	char	7 : 8	
232		Четность		0 – none (нет), 1 – odd (нечетный), 2 – even (четный)	
233		Количество стоп-бит		1 : 2	

234	Счетчик электроэнергии, импульсов на кВт*ч		int	1 : 10000	
235	Период измерения расхода воды, секунд		int	5 : 600	
236	Расчет КПД системы	Тип счетчика электроэнергии	char	0 – внешний электросчетчик с импульсным выходом, 1 – внутренний	
237		Высота подъема воды	char	0 – только константа, 1 – константа+значение датчика давления (уровня) dP	
238		Константа высоты подъема, м	int	10 : 2000	1,0 : 200,0
239		Время расчета КПД, минут	char	1 : 60	
240	Включение/отключение звуковых сигналов МКЗ		char	0 – отключить сигналы, 1 – включить сигналы	
241	Значение минимального коэффициента мощности cos(φ) для отключения двигателя (0 – отключает эту защиту)		char	0 : 100	0,00 : 1,00
242	Модуль термо-датчика	Функция модуля	char	0 – Не используется, 1 – Терморегулятор, 2 – Защита от перегрева	
243		Тип датчика температуры	char	1 – РТС-термистор, 2 – Pt100 (a=1.385), 3 – ТСМ 50М (a=1.428), 4 – ТСМ 100М (a=1.428), 5 – ТСП 100П (a=1.391), 6 – Н.З. термоконтакт, 7 – Н.О. термоконтакт	
244		Температура включения, °C	signed int	-500 : 2500	-50,0 : +250,0
245		Температура отключения, °C			
246	Делитель цены импульса счетчика воды		char	1 : 100	
247	Время антидребезга контактов счетчика воды, мс		int	5 : 250	0,5 : 25,0
248	Время антидребезга контактов счетчика электроэнергии, мс				
249	Номинальный ток, А				
250	Тип питающей сети		0 – однофазная, 1 – трехфазная		
251 - 299	Резерв				

Запись в регистры производится командой **06h (Write Single Register)**

В случае успешного выполнения функции ответное сообщение идентично запросу.
При попытке записи значений вне допустимого диапазона, будет записано минимальное или максимальное значение этого диапазона.

Чтение 3 регистров с адреса 101.

Запрос RTU		Ответ RTU	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Адрес	01	Адрес	01
Функция	03	Функция	03
Начальный адрес старший	00	Счетчик байт	06
Начальный адрес младший	65	Данные регистра 101 старший	00
Количество регистров старший	00	Данные регистра 101 младший	00
Количество регистров младший	03	Данные регистра 102 старший	00
CRC	15 D4	Данные регистра 102 младший	00
		Данные регистра 103 старший	01
		Данные регистра 103 младший	90
		CRC	20 89

Запрос ASCII		Ответ ASCII	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 33	Функция	30 33
Начальный адрес старший	30 30	Счетчик байт	30 36
Начальный адрес младший	36 35	Данные регистра 101 старший	30 30
Количество регистров старший	30 30	Данные регистра 101 младший	30 30
Количество регистров младший	30 33	Данные регистра 102 старший	30 30
LRC	39 34	Данные регистра 102 младший	30 30
Конец кадра (CR+LF)	0D 0A	Данные регистра 103 старший	30 31
		Данные регистра 103 младший	39 30
		LRC	36 35
		Конец кадра (CR+LF)	0D 0A

Запись регистра по адресу 108.
Нормальный ответ контроллера повторяет запрос.

Запрос RTU		Ответ RTU	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Адрес	01	Адрес	01
Функция	06	Функция	06
Начальный адрес старший	00	Начальный адрес старший	00
Начальный адрес младший	6C	Начальный адрес младший	6C
Данные регистра 108 старший	00	Данные регистра 108 старший	00
Данные регистра 108 младший	0A	Данные регистра 108 младший	0A
CRC	C9 D0	CRC	C9 D0

Запрос ASCII		Ответ ASCII	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 36	Функция	30 36
Начальный адрес старший	30 30	Начальный адрес старший	30 30
Начальный адрес младший	36 43	Начальный адрес младший	36 43
Данные регистра 108 старший	30 30	Данные регистра 108 старший	30 30
Данные регистра 108 младший	30 41	Данные регистра 108 младший	30 41
LRC	38 33	LRC	38 33
Конец кадра (CR+LF)	0D 0A	Конец кадра (CR+LF)	0D 0A

Запись нескольких регистров настройки производится командой **10h (Write Multiple Registers)**

Данная функция позволяет записать до 64 16-ти разрядных регистров настройки. В случае успешного выполнения функции ответное сообщение состоит из первых 6 байт запроса.

При попытке записи значений вне допустимого диапазона, будет записано минимальное или максимальное значение этого диапазона.

Запись 5 регистров с адреса 101(значения 0, 0, 400, 300, 10).

Запрос RTU		Ответ RTU	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Адрес	01	Адрес	01
Функция	10	Функция	10
Начальный адрес старший	00	Начальный адрес старший	00
Начальный адрес младший	65	Начальный адрес младший	65
Количество регистров старший	00	Количество регистров старший	00
Количество регистров младший	05	Количество регистров младший	05
Количество байт в регистрах	0A	CRC	10 15
Данные регистра 101 старший	00		
Данные регистра 101 младший	00		
Данные регистра 102 старший	00		
Данные регистра 102 младший	00		
Данные регистра 103 старший	01		
Данные регистра 103 младший	90		
Данные регистра 104 старший	01		
Данные регистра 104 младший	2C		
Данные регистра 105 старший	00		
Данные регистра 105 младший	0A		
CRC	E5 63		

Запрос ASCII		Ответ ASCII	
<i>Название поля</i>	<i>(Hex)</i>	<i>Название поля</i>	<i>(Hex)</i>
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	31 30	Функция	31 30
Начальный адрес старший	30 30	Начальный адрес старший	30 30
Начальный адрес младший	36 35	Начальный адрес младший	36 35
Количество регистров старший	30 30	Количество регистров старший	30 30
Количество регистров младший	30 35	Количество регистров младший	30 35
Количество байт в регистрах	30 41	LRC	38 35
Данные регистра 101 старший	30 30	Конец кадра (CR+LF)	0D 0A
Данные регистра 101 младший	30 30		
Данные регистра 102 старший	30 30		
Данные регистра 102 младший	30 30		
Данные регистра 103 старший	30 31		
Данные регистра 103 младший	39 30		
Данные регистра 104 старший	30 31		
Данные регистра 104 младший	32 43		
Данные регистра 105 старший	30 30		
Данные регистра 105 младший	30 41		
LRC	42 33		
Конец кадра (CR+LF)	0D 0A		

3.4 Регистры чтения данных объекта (Input Register).

Чтение регистров производится командой **04 (Read Input Register)**.

Данная функция позволяет получить двоичное содержимое 16-ти разрядных регистров адресуемого контроллера. Адресация позволяет получить за каждый запрос до 125 регистров. Широковещательный режим не допускается. Адресуемый контроллер посылает в ответе свой адрес, код выполненной функции и информационное поле. Информационное поле содержит 2 байта, описывающих количество возвращаемых байт данных. Длина каждого регистра данных – 2 байта. Первый байт данных в посылке является старшим байтом регистра, второй – младшим. С адреса 400 (0190h) находится Журнал аварий станции. Количество записей – 20 (20*17 = 340 регистров, 680 байт данных).

Запись содержит поля: код аварии, дата, месяц, год, час, мин, значений токов и напряжений по фазам на момент аварии и время сброса аварии (нули, если авария еще не сброшена).

Записи располагаются в хронологическом порядке, начиная с последней по времени аварии. При возникновении очередной аварии происходит сдвиг вниз на одну запись. Последняя авария всегда находится в первой записи.

Адрес	Название	Описание	
300	Управление (чтение и запись)	бит	
		0	Реле включения двигателя К1 1: Вкл, 0: Откл
		1	Реле универсальное К2 1: Включено
		2	Реле универсальное К3 1: Включено
		3	Реле универсальное К4 1: Включено
		4	Сброс аварии 1: команда сброса
		5	Сброс (перезагрузка) контроллера 1: команда сброса
		6	Постановка объекта на охрану 1: команда постановки
7	Снятие объекта с охраны 1: команда снятия		
8-F	Резерв		
<p>Примечание – данный регистр дублирует функции управления, описанные в разделе 3.1 «Чтение и запись состояния дискретных выходов». Управлять нагрузкой возможно как чтением и записью битовых регистров по адресу 000 – 007 (функция 01 и 05), так и чтением и записью одного регистра по адресу 300.</p>			
301	Режим работы МК3	0: Ручное (местное) управление	1: Автоматический режим
302	Состояние МК3	0: Работа МК3 приостановлена 1: Проверка перед запуском двигателя 2: Запуск двигателя 3: Ожидание срабатывания сигнала «Внешнее управление» 4: Ожидание нажатия кнопки «ПУСК» 5: Ожидание нажатия кнопки «СТОП» 6: Ожидание верхнего уровня 7: Ожидание нижнего уровня 8: Ожидание срабатывания датчика dL (Режим – По таймеру) 9: Ожидание срабатывания датчика dH (Режим – По таймеру) 10: Двигатель включен на время работы по таймеру 11: Задержка пуска 12: Задержка останова 13: Двигатель включен сигналом «Внешнее управление» 14: Задержка пуска после подачи питания 15: Ожидание команды включения (по линии связи) 16: Ожидание команды отключения (по линии связи) 17: Ожидание команды включения из SMS 18: Ожидание команды отключения из SMS 19: Диагностика МК3 после подачи питания 20: Ожидание срабатывания датчика dS (сухой ход) 21: Ожидание снятия сигнала «Внешняя авария» (E.Error) 22: Ожидание снятия сигнала с датчика аварийного уровня (dAV) 23: Запущен режим диагностики по RS-485 24: Принудительный запуск двигателя при длительном простое 25: Принудительный запуск двигателя по сигналу «Внешнее управл.» 26: Ожидание понижения температуры 27: Ожидание восстановления датчика температуры	

303	Состояние охранной сигнализации	0: отключена в установочном меню МКЗ 1: поставлена на охрану 2: ожидание срабатывания датчика двери (при постановке на охрану) 3: несанкционированный доступ 4: снята с охраны 5: ожидание ввода пароля для снятия с охраны (20 секунд после открытия двери, затем – несанкционированный доступ)		
304	Состояние дискретных входов и аварийных сигналов	бит		
		0	Дискретный вход DI.1	0: разомкнут 1: замкнут
		1	Дискретный вход DI.2	
		2	Дискретный вход DI.3	
		3	Дискретный вход DI.4	
		4	Дискретный вход DI.5	
		5	Дискретный вход DI.6	
		6	Дискретный вход DI.7	
		7	Дискретный вход DI.8	
		8	Сигнал повышения напряжения	0: нет аварийного сигнала 1: есть аварийный сигнал
		9	Сигнал понижения напряжения	
		10	Сигнал перекоса фаз по напряжению	
		11	Сигнал повышения тока	
		12	Сигнал понижения тока	
		13	Сигнал перекоса фаз по току	
		14	Сигнал неисправности аналогового входа	
15	Сигнал о замыкании (утечке) на корпус			
<p>Примечание – данный регистр дублирует функцию чтения дискретных входов, описанную в разделе 3.2 «Чтение состояния дискретных входов». Возможно как чтение одиночных битовых регистров по адресам 50 – 65 (функция 02), так и чтение одного регистра по адресу 304 (функция 04).</p>				
305	Код аварии	0: Нет аварии 1: Неправильное чередование фаз или напряжение одной из фаз меньше 50В 2: Повышение напряжения выше заданной уставки 3: Понижение напряжения ниже заданной уставки 4: Перекос фаз по напряжению 5: Повышение тока выше заданной уставки 6: Понижение тока ниже заданной уставки 7: Перекос фаз по току 8: «Сухой» ход (срабатывание входа dS) 9: Внешняя авария (срабатывание входа E.Egor) 10: Неправильное срабатывание дискретных датчиков уровня (давления) 11: Превышено время продолжительности работы двигателя 12: Блокировка работы при подряд возникающих авариях 13: Внутренняя авария МКЗ (нет связи с измерительным модулем) 14: Превышение количества пусков в час в автоматическом режиме 15: Отказ аналогового датчика давления (уровня) 0...20(4...20)ма 16: Таймаут соединения с ПК (ПЛК) по RS-485 17: Таймаут соединения GSM 18: Не заданы дискретные входы DI.x для датчиков уровня (dL и/или dH) 19: Срабатывание датчика аварийного уровня dAV (перелив) 20: Замыкание (утечка) на корпус обмоток электродвигателя 21: Отказ часов реального времени (RTC) 22: Понижение коэффициента мощности cos(φ) ниже заданной уставки 23: Перегрев (повышение температуры выше заданной уставки) 24: Отказ датчика температуры или измерительной цепи		
306	Оставшееся время выдержки при аварии в секундах	int	0 : 3600	

307	Напряжение фазы А(L1), В		int	0 : 3000	0,0 : 300,0
308	Напряжение фазы В(L2), В				
309	Напряжение фазы С(L3), В				
310	Среднее напряжение по фазам, В				
311	Перекас по напряжению, В			0 : 1000	0,0 : 100,0
312	Ток фазы А(L1), А			0 : 10000	0,0 : 1000,0
313	Ток фазы В(L2), А				
314	Ток фазы С(L3), А				
315	Средний ток по фазам, А				
316	Перекас по току, %		0 : 1000	0,0 : 100,0	
317	Состояние аналогового датчика 0...20mA (4...20mA)		0 – исправен, 1 – обрыв, 2 – короткое замыкание.		
318	Значение аналогового датчика, (бар)(м)(°С)(м3/ч)(л/с)(%) (МПа)		int	0 : 10000	0,00 : 100,00
319	Значение аналогового датчика в mA			0 : 2500	0,00 : 25,00
320	Время работы по таймеру (сколько прошло) в секундах		int	0 : 20000	
321	Время работы по таймеру (сколько осталось) в секундах			0 : 20000	
322	Время наработки двигателя (сбрасываемое), часов		int	0 : 65500	
323	Кол-во пусков двигателя (сбрасываемое)				
324	Время наработки двигателя общее, часов				
325	Кол-во пусков двигателя общее				
326	Счетчик воды с импульсным выходом, сумматор	тысячи м ³	int	0 : 999	
327		сотни м ³		0 : 999	
328		литры		0 : 999	
329	Счетчик воды с импульсным выходом, значение расхода воды	литров в минуту		0 : 65500	
330		м ³ в час		0 : 65500	0,0 : 6550,0
331	Счетчик электроэнергии с импульсным выходом, сумматор	тысячи кВт*час		int	0 : 999
332		сотни кВт*час	0 : 999		
333		Вт*час	0 : 999		
334	Коэффициент мощности двигателя cos(φ)	Фаза А(L1)	char	-100 : +100	-1,00 : +1,00
335		Фаза В(L2)			
336		Фаза С(L3)			
337		Среднее значение			
338	Полная мощность двигателя S, кВа		int	0 : 65535	0,00 : 655,35
339	Активная мощность двигателя P, кВт				
340	Реактивная мощность двигателя Q, кВАр				
341	Фактически затраченная энергия E _{факт} по значениям электросчетчика, МДж			0 : 65535	0,0 : 6553,5
342	Энергия расчетная E _{рас} , по прошедшему объему воды и заданной высоте подъема, МДж				
343	КПД работы системы, % (E _{рас} / E _{факт})			0 : 1000	0,0 : 100,0
344	Удельное энергопотребление, кВт / м ³			0 : 65535	0,000 : 65,535
345	Не используется				
346					

347	Счетчик воды с импульсным выходом, сумматор в литрах	int	0 : 999 999 999	
348				
349	Счетчик электроэнергии с импульсным выходом, сумматор в Вт*ч	int	0 : 999 999 999	
350				
351	Сопrotивление цепи термодатчика, Ом	int	0 : 65500	0,0 : 6500,0
352	Состояние термодатчика	char	0: не используется, 1: в норме, 2: в обрыве, 3: короткое замыкание, 4: перегрев	
353	Температура, °C	signed int	-500 : 2500	-50,0 : +250,0
354	Состояние терморегулятора	char	0: реле отключено, 1: реле включено, 2: не используется, 3: не подключен термодатчик, 4: не задано исполн. реле	
355-399	Резерв			

Журнал аварий, 20 записей, 20*17 = 340 регистров, 680 байт данных.

Адрес	Название		Диапазон допустимых значений			
			Тип переменной	Диапазон значений	Реальное значение	
Dec						
400	Код аварии	0 – пустая запись	char	0 : 20		
401	Число	Время возникновения аварии (0 – пустая запись)		1:31		
402	Месяц			1:12		
403	Год			int	2020 : 2100	
404	Часов			char	0:23	
405	Минут				0:59	
406	Ток фазы А (L1), А		int	0 : 10000	0,0 : 1000,0	
407	Ток фазы В (L2), А					
408	Ток фазы С (L3), А					
409	Напряжение фазы А (L1), В		int	0 : 4000	0,0 : 400,0	
410	Напряжение фазы В (L2), В					
411	Напряжение фазы С (L3), В					
412	Число	Время сброса аварии (0 – авария пока не сброшена)	char	1 : 31		
413	Месяц			1 : 12		
414	Год		int	2020 : 2100		
415	Часов		char	0 : 23		
416	Минут			0 : 59		
417 – 433	Запись №2					
434 - 450	Запись №3					
451 - 467	Запись №4					

468 – 484	Запись №5			
485 – 501	Запись №6			
502 – 518	Запись №7			
519 – 535	Запись №8			
536 – 552	Запись №9			
553 – 569	Запись №10			
570 – 586	Запись №11			
587 – 603	Запись №12			
604 – 620	Запись №13			
621 – 637	Запись №14			
638 - 654	Запись №15			
655 – 671	Запись №16			
672 – 688	Запись №17			
689 – 705	Запись №18			
706 – 722	Запись №19			
723 - 739	Запись №20			

С адреса 800 (320h) находится **ЖУРНАЛ РАБОТЫ** - время включенного состояния нагрузки каждый час в сутках. Учитывается время работы за текущие и прошедшие сутки. Количество записей – 2*24. В записи – время включенного состояния нагрузки в секундах за текущий час.

Адрес	Название		Тип переменной	Значение
800	Текущие сутки	Время работы с 00-00 по 01-00	int	0 : 3600 секунд
801		Время работы с 01-00 по 02-00		
802		Время работы с 02-00 по 03-00		
803		Время работы с 03-00 по 04-00		
804		Время работы с 04-00 по 05-00		
805		Время работы с 05-00 по 06-00		
806		Время работы с 06-00 по 07-00		
807		Время работы с 07-00 по 08-00		
808		Время работы с 08-00 по 09-00		
809		Время работы с 09-00 по 10-00		
810		Время работы с 10-00 по 11-00		
811		Время работы с 11-00 по 12-00		
812		Время работы с 12-00 по 13-00		
813		Время работы с 13-00 по 14-00		
814		Время работы с 14-00 по 15-00		
815		Время работы с 15-00 по 16-00		
816		Время работы с 16-00 по 17-00		
817		Время работы с 17-00 по 18-00		
818		Время работы с 18-00 по 19-00		
819		Время работы с 19-00 по 20-00		
820		Время работы с 20-00 по 21-00		
821		Время работы с 21-00 по 22-00		
822		Время работы с 22-00 по 23-00		
823		Время работы с 23-00 по 00-00		

824 - 847	Прошедшие сутки	Время работы с 00-00 по 01-00 ... Время работы с 23-00 по 00-00	int	0 : 3600 секунд		
848	Общее время включенного состояния за ТЕКУЩИЕ сутки, минут		int	0 : 1440		
849	Общее время включенного состояния за ПРОШЕДШИЕ сутки, минут					
850	Общее время включенного состояния за ТЕКУЩИЕ сутки, %			0 : 1000	0,0 : 100,0	
851	Общее время включенного состояния за ПРОШЕДШИЕ сутки, %					
852	Время включенного состояния за ПОСЛЕДНИЙ час, секунд		int	0 : 3600		

С адреса 900 (384h) находится информация о версии и дате ПО.

Адрес	Название		Тип переменной	Значение
900	Версия ПО		int	100...1000 (168 = v.1.68)
901	Дата ПО	День	char	0...31
902		Месяц		0...12
903		Год	int	2020...2100

Чтение 3 регистров с адреса 300 (0x012Ch) (полученные значения 2, 0, 4).

Запрос RTU		Ответ RTU	
Название поля	(Hex)	Название поля	(Hex)
Адрес	01	Адрес	01
Функция	04	Функция	04
Начальный адрес старший	01	Счетчик байт	06
Начальный адрес младший	2C	Данные регистра 300 старший	00
Количество регистров старший	00	Данные регистра 300 младший	02
Количество регистров младший	03	Данные регистра 301 старший	00
CRC	70 3E	Данные регистра 301 младший	00
		Данные регистра 302 старший	00
		Данные регистра 302 младший	04
		CRC	18 90

Запрос ASCII		Ответ ASCII	
Название поля	(Hex)	Название поля	(Hex)
Начало кадра (:)	3A	Начало кадра (:)	3A
Адрес	30 31	Адрес	30 31
Функция	30 34	Функция	30 34
Начальный адрес старший	30 31	Счетчик байт	3036
Начальный адрес младший	32 43	Данные регистра 300 старший	30 30
Количество регистров старший	30 30	Данные регистра 300 младший	30 32
Количество регистров младший	30 33	Данные регистра 301 старший	30 30
LRC	43 42	Данные регистра 301 младший	30 30
Конец кадра (CR+LF)	0D 0A	Данные регистра 302 старший	30 30
		Данные регистра 302 младший	30 34
		LRC	45 46
		Конец кадра (CR+LF)	0D 0A

4. Исключительные ситуации

МКЗ поддерживает сообщения информирования клиента (мастера) Modbus об исключительных ситуациях (Exception). Формат возвращаемых пакетов полностью соответствует документу «Modbus Application Protocol Specification v1.1a». Сообщения об исключительных ситуациях возникают только на запросы, адресованные данному устройству с правильным значением LRC или CRC пакета.

Код ошибки	Название	Описание
01	Неподдерживаемая команда	Возникает только при запросе с номером команды, которую не поддерживает данное устройство.
02	Неподдерживаемый адрес данных	Возникает только при запросе с адресом данных, которых нет в таблицах соответствия между адресами Modbus и внутренней памятью устройства
03	Неверное количество данных	В запросе содержится значения недопустимые для сервера. Например, запрос количества регистров более чем 120.

Когда контроллер обнаруживает одну из этих ошибок, он посылает ответное сообщение, содержащее адрес, код функции, код ошибки и контрольную сумму. Для указания на то, что ответное сообщение – это уведомление об ошибке, старший бит поля кода функции устанавливается в 1.

Запрос RTU

Адрес	Функция	Старший байт адреса	Младший байт адреса	Старший байт числа ячеек	Младший байт числа ячеек	CRC
01	03	00	12	00	06	65 CD

Этот запрос требует от устройства с адресом 01 данных 6 регистров с адреса 18. Но, например, это устройство имеет максимальный адрес 0x0016, а запрашиваемое количество данных превышает диапазон адреса и является ошибочным. Соответственно, будет сгенерировано следующее ответное сообщение:

Ответ RTU

Адрес	Функция	Код исключительной ситуации	CRC
01	83	03	XX XX

Значение в поле функции равно оригинальному значению с установленным в единицу старшим битом. Код исключительной ситуации 03 указывает на ошибочное количество данных.

$$0x83 = 10000011$$

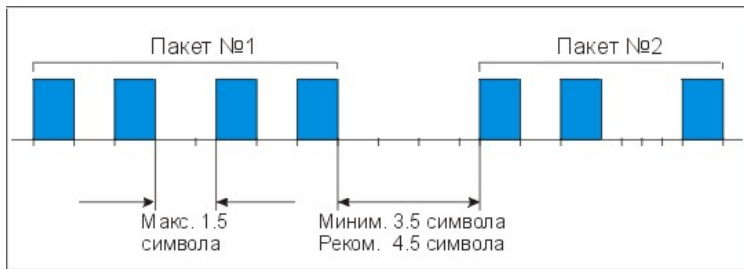
5. Задержки между пакетами

Временные задержки между пакетами и символами пакетов полностью соответствуют «Modbus over Serial Line Specification & Implementation guide V1.0».

В ASCII-режиме сообщение начинается с символа "двоеточие" (":" ASCII, 3A hex), и заканчивается символами "Возврат каретки/Перевод строки" (CR/LF, ASCII 0D и 0A hex). Опрашиваемые устройства в сети непрерывно отслеживают символ "двоеточие". После него начинается прием тела сообщения до символов CR/LF. Интервалы между символами могут быть до 1 секунды. Если интервал больше, принимающее устройство считает это ошибкой и игнорирует сообщение.

В RTU-режиме сообщение начинается с интервала тишины, равного времени передачи 3.5 символов при данной скорости передачи в сети. Затем первым символом передается адрес устройства и остальные байты пакета. Между символами одного пакета может быть задержка длиной не более полутора символов. Между пакетами должна быть задержка не менее 3,5 символов. Рекомендуется начать передавать следующий пакет не ранее чем через 4,5 символа после получения последнего бита предыдущего пакета. Если в интервале между 1,5 символами и 3,5 символами после прихода последнего символа, приходит первый символ следующего пакета, сбрасываются оба пакета.

Комментарий: задержка длиной в символ - это время необходимое, для того чтобы передать 8 бит данных при данной скорости передачи и параметрах соединения.



Рекомендуемые задержки между пакетами для режима RTU

Скорость интерфейса при параметрах 8N1	Минимальное время между символами в пакете	Минимальная задержка между пакетами	Рекомендуемая задержка между пакетами
2400 бит/с	6,3 мс	14,6 мс	18,8 мс
4800 бит/с	3,2 мс	7,3 мс	9,4 мс
9600 бит/с	1,6 мс	3,6 мс	4,7 мс
14400 бит/с	1,0 мс	2,4 мс	3,1 мс
19200 бит/с	0,8 мс	1,8 мс	2,3 мс
38400 бит/с	0,4 мс	0,9 мс	1,2 мс
57600 бит/с	0,3 мс	0,6 мс	0,8 мс
115200 бит/с	0,2 мс	0,3 мс	0,4 мс

6 Генерация контрольной суммы

6.1 Контрольная сумма LRC.

LRC это однобайтное число, которое вычисляется передающим устройством и добавляется в конец ASCII-сообщения. При расчете LRC исключаются начальный символ ":" и конечные символы CR/LF. Принимающее устройство также вычисляет LRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы пришедшего сообщения. Если суммы не совпали - то имеет место ошибка.

LRC вычисляется сложением последовательности байтов сообщения, отбрасывая все переносы, и затем двойным дополнением результата.

Алгоритм генерации LRC:

1. Сложить все байты сообщения, исключая стартовый символ ':' и конечные символы CR/LF.
2. Обнулить все биты больше 7, т.е. оставить младший байт.
3. Сделать результирующий байт отрицательным, чтобы получить LRC.

Размещение LRC в сообщении

Когда 7-ми битная последовательность LRC (2 символа ASCII) передается в сообщении, то старший символ будет передан первым, младший - вторым. Например, передача LRC 61 hex(0110 0001):

':'	Адрес	Функция	Сч-к байт	Байт	Байт	Байт	Байт	LRC Ст.	LRC Мл.	CR	LF
-----	-------	---------	-----------	------	------	------	------	---------	---------	----	----

Пример функции на языке C, реализующей генерацию LRC

```
void main (void)
{
    unsigned char Array[128];          // Массив данных, содержащий ответ
    ...
    ...
    // В массиве Array[] содержатся 8 байт ответного сообщения
    // Добавляем к массиву Array[] 9-й байт контрольной суммы
    Array[9]= Create_LRC(Array, 8);
    ...
    // Преобразовываем значения массива в символы ASCII, добавляем символы
    // начала и конца сообщения и отправляем ответ.
    ...
} // End

unsigned char Create_LRC( unsigned char *Buffer, unsigned short NBytes )
{
    unsigned char nLRC=0;
    unsigned short i=0;

    for (i=0; i< NBytes; i++){ nLRC += *Buffer++;}
    return (unsigned char)(-nLRC);

} // End
```

6.2 Контрольная сумма CRC

CRC - это 16-ти разрядная величина, т.е. два байта. CRC вычисляется передающим устройством и добавляется к RTU-сообщению. Принимающее устройство также вычисляет CRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы пришедшего сообщения. Если суммы не совпали - то имеет место ошибка.

Используются два метода вычисления CRC: циклический или табличный.

При циклическом методе 16-ти битовый регистр CRC предварительно загружается числом FFFF hex. Процесс начинается с добавления байтов сообщения к текущему содержимому регистра. Для генерации CRC используются только 8 бит данных. Старт и стоп биты, бит паритета, если он используется, не учитываются в CRC.

В процессе генерации CRC, каждый 8-ми битовый символ складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра. Результат сдвигается в направлении младшего бита, с заполнением 0 старшего бита. Младший бит извлекается и проверяется. Если младший бит равен 1, то содержимое регистра складывается с определенной ранее, фиксированной величиной, по ИСКЛЮЧАЮЩЕМУ ИЛИ. Если младший бит равен 0, то ИСКЛЮЧАЮЩЕЕ ИЛИ не делается.

Этот процесс повторяется, пока не будет сделано 8 сдвигов. После последнего (восьмого) сдвига, следующий байт складывается с содержимым регистра и процесс повторяется снова. Финальное содержание регистра, после обработки всех байтов сообщения и есть контрольная сумма CRC.

Алгоритм генерации CRC:

1. 16-ти битовый регистр загружается числом FFFFh (все 1), и используется далее как регистр CRC.
2. Первый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра CRC. Результат помещается в регистр CRC.
3. Регистр CRC сдвигается вправо (в направлении младшего бита) на 1 бит, старший бит заполняется нулем.
(Если младший бит 0): Повторяется шаг 3 (сдвиг)
4. (Если младший бит 1): Делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа A001 hex.
5. Шаги 3 и 4 повторяются восемь раз.
6. Повторяются шаги со 2 по 5 для следующего сообщения. Это повторяется до тех пор пока все байты сообщения не будут обработаны.
7. Финальное содержание регистра CRC и есть контрольная сумма.

При табличном вычислении все возможные величины CRC загружены в два массива. Один массив содержит все 256 возможных комбинаций CRC для старшего байта поля CRC, другой массив содержит данные для младшего байта. Индексация CRC в этом случае обеспечивает быстрое выполнение вычислений новой величины CRC для каждого нового байта из буфера сообщения.

Размещение CRC в сообщении

При передаче 16 бит контрольной суммы CRC в сообщении, сначала передается младший байт, затем старший. Например, если CRC равна 1288 hex:

Адрес	Функция	Счетчик байт	Данные	Данные	Данные	Данные	CRC Lo	CRC Hi
							88	12

Пример функции на языке C, реализующей генерацию CRC циклическим методом

```
void main (void)
{
    unsigned char CRC_Hi, CRC_Lo;    // Старший и младший байты контрольной суммы
    unsigned char Array[128];       // Массив данных, содержащий ответ
    ...
    ...
    // В массиве Array[] содержатся 8 байт ответного сообщения
    Create_CRC (8);
    // Добавляем к массиву Array[] 9 и 10 байты контрольной суммы (CRC_Lo и CRC_Hi)
    Array[9]= CRC_Lo;
    Array[10]= CRC_Hi;
    // Отправляем ответ
    ...
} // End

void Create_CRC(unsigned char NBytes)// Подсчет CRC по количеству байт массива Array[]
{
    unsigned int Register = 0xFFFF;
    unsigned int Lsb;
    unsigned char i, j;

    for (i=0; i<NBytes; i++)
    {
        Register^=Array[i];
        for ( j=0; j<8; j++)
        {
            Lsb = Register & 0x0001;
            Register = Register >> 1; // Сдвигаем на 1 вправо
            if (Lsb) Register^=0xA001; // Если мл. бит 0 - исключаемое ИЛИ с числом 0xA001
        }
    }
    CRC_Hi = Register >> 8;
    CRC_Lo = Register;
} // End
```

Пример функции на языке C, реализующей генерацию CRC табличным методом

```
void main (void)
{
    unsigned char CRC_Hi, CRC_Lo;    // Старший и младший байты контрольной суммы
    unsigned char Array[128];       // Массив данных, содержащий ответ
    ...
    // В массиве Array[] содержатся 8 байт ответного сообщения
    Create_CRC_TAB(Array, 8);
    // Добавляем к массиву Array[] 9 и 10 байты контрольной суммы (CRC_Lo и CRC_Hi)
    Array[9]= CRC_Lo;
    Array[10]= CRC_Hi;
    // Отправляем ответ
    ...
} // End

unsigned short Create_CRC_TAB(unsigned char *Buffer, unsigned char NBytes)
{
    unsigned char uchCRCHI = 0xFF, uchCRCLo = 0xFF;
    unsigned int uIndex;

    while( NBytes-- )
    {
        uIndex = uchCRCHI ^ *Buffer++;
        uchCRCHI = uchCRCLo ^ auchCRCHI[uIndex];
        uchCRCLo = auchCRCLo[uIndex];
    }
    CRC_Hi= uchCRCHI;
    CRC_Lo= uchCRCLo;
} // End

const unsigned char auchCRCHI[256]=
{
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
    0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40
};

const unsigned char auchCRCLo[256]=
{
    0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,0x04,
    0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,0x08,0xC8,
    0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,0x1D,0x1C,0xDC,
    0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,0x11,0xD1,0xD0,0x10,
    0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,0x37,0xF5,0x35,0x34,0xF4,
    0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,0x3B,0xFB,0x39,0xF9,0xF8,0x38,
    0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,
    0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,
    0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,
    0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,
    0x78,0xB8,0xB9,0x79,0xBB,0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,
    0xB4,0x74,0x75,0xB5,0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,
    0x50,0x90,0x91,0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,
    0x9C,0x5C,0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,
    0x88,0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
    0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,0x40
};
};
```