
Многофункциональное измерительное устройство ESM

Описание протокола обмена Modbus
[ПРОЕКТ]

Содержание

1	Общая информация	4
1.1	Интерфейсы	4
1.2	Структура запросов	5
1.3	Поддерживаемые команды	8
1.4	Примечания	9
1	Чтение информации о приборе.....	10
2	Чтение результатов текущих измерений	11
2.1	Стандартные команды	11
2.2	Пользовательские команды	11
2.3	Формат данных	14
3	Чтение гармоник и интергармоник.....	16
4	Аутентификация	17
4.1	Запрос ключа авторизации	17
4.2	Подготовка запроса авторизации.....	17
4.3	Отправка запроса авторизации	17
4.4	Чтение результата авторизации	18
5	Чтение энергии по тарифам.....	20
6	Журналы	22
6.1	Установка начального и конечного значения времени для чтения записей	22
6.2	Чтение записей	23
6.3	Завершение процедуры чтения записей	24
6.4	Структура типов журналов	25
6.5	Структура временных интервалов	25
6.6	Журнал профиля	26
6.6.1	Структура записи.....	26
6.6.2	Флаги состояния полчасовок	26
6.6.3	Получения списка параметров.....	27
6.7	Структура записи журнала энергий.....	27
6.8	Структура журнала состояний дискретных сигналов	27
6.9	Структура записи журнала событий	28
6.10	Структура расширенной записи журнала событий	29
6.11	Разбор записи журнала событий	29
7	Дискретные сигналы (DIO).....	31
7.1	Запрос состояний	31
7.2	Запрос информации о дискретных сигналах	31
8	Синхронизация времени	33
8.1	Запрос текущего времени	33
8.2	Корректировка времени	33
8.3	Установка времени.....	34
9	Диагностические данные	36
9.1	Описание диагностических регистров	36

1 Общая информация

Описание протокола обмена Modbus в ESM позволяет реализовать поддержку многофункционального измерительного устройства ESM в программном обеспечении, предназначенном для считывания и хранения данных со счетчиков электроэнергии, приборов учета и других измерительных устройств.



Примечание: Данное описание составлено на основе прошивки для ESM версии 0.0.0.1 от 15.06.2018

1.1 Интерфейсы

Устройства ESM могут применяться в составе различных автоматизированных систем, включая системы технологического управления, системы учета электроэнергии, системы мониторинга качества электроэнергии и др.

Устройства ESM могут иметь различные наборы интерфейсов (см. рис. 1), предназначенные для связи с верхним уровнем, модулями индикации и другими внешними устройствами. Независимо от модификации, в каждом ESM присутствуют порты RS-485 и Ethernet.

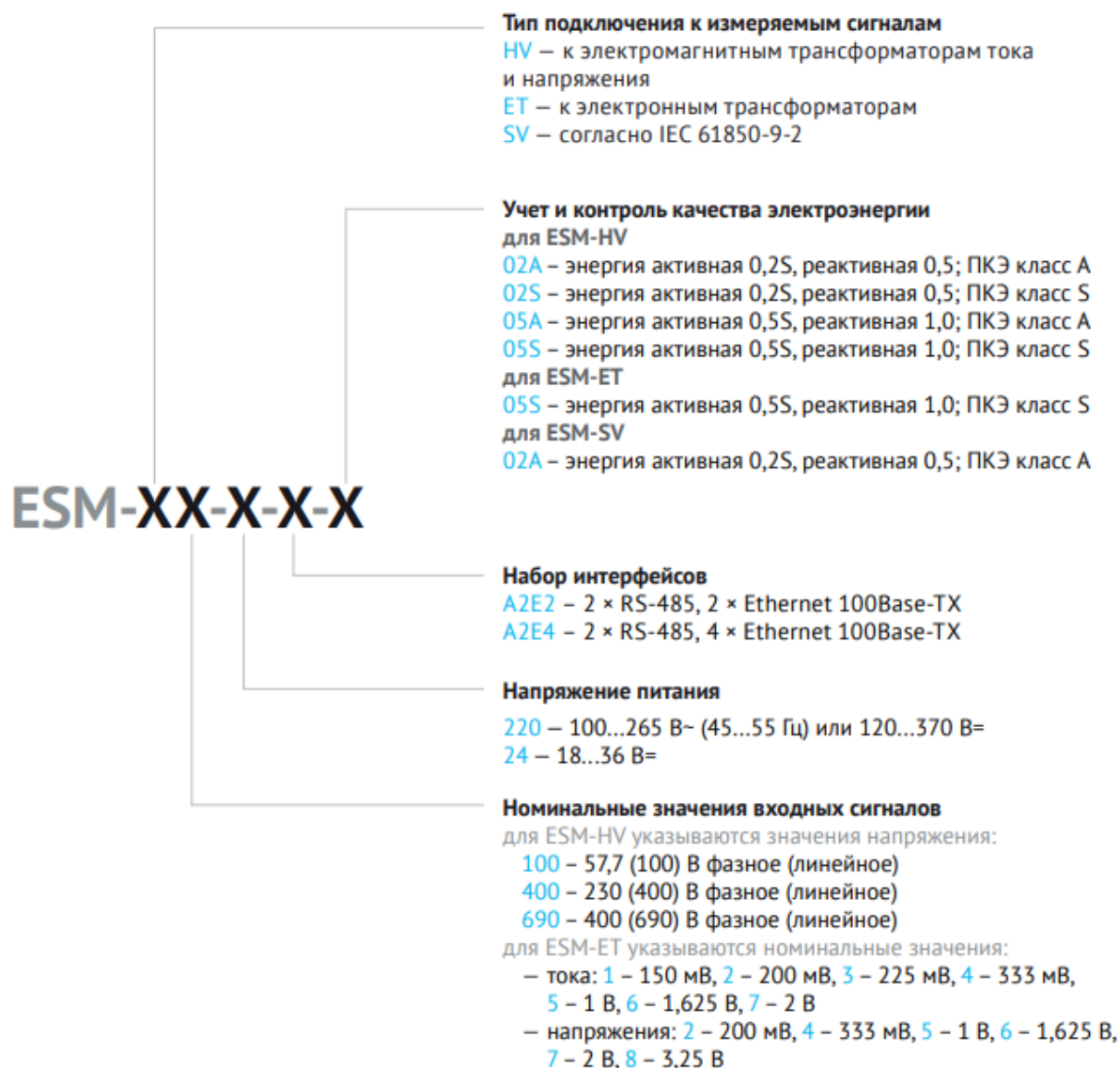


Рисунок 1. Структура условного обозначения устройств ESM.

1.2 Структура запросов

Структура фреймов запросов и ответов соответствует структуре, представленной в описании протокола Modbus (Modbus - это торговая марка, принадлежащая компании Schneider Electric), приведенном на сайте www.modbus.org.

По стандарту Modbus кадр каждого сообщения состоит из элемента данных протокола - Application Data Unit (ADU), который включает элемент данных приложения - Protocol Data Unit (PDU):

ADU = Адрес + PDU + Контроль ошибок

PDU = Код функции + Данные

Уровень элементов данных приложения (PDU) не зависит от способа передачи данных.

Уровень элементов данных протокола (ADU), предусматривает дополнительные поля к PDU, позволяющие организовать передачу сообщений в определенных сетях.

Имеющиеся в ESM интерфейсы позволяют организовать обмен данными по следующим вариантам протокола Modbus:

Modbus RTU (Remote Terminal Unit – «удаленный терминал») – компактный двоичный режим передачи данных. Для Modbus RTU стандарт Modbus предусматривает применение физического интерфейса RS-485, RS-422 или RS-232. В случае ESM используется двухпроводной интерфейс RS-485.

Сообщения Modbus RTU передаются в виде кадров (см. табл. 1), для каждого из которых известно начало и конец.

Таблица 1

Имя поля	Длина (бит)	Описание
Начало	28	Пауза (тишина) продолжительностью не менее 3,5 шестнадцатеричных символов.
Адрес (Slave ID)	8	Адрес ведомого устройства, которому адресован запрос.
Код функции	8	Код функции определяет тип выполняемой операции.
Данные	n x 8	Поле данных содержит параметры запроса, результаты ответа и тп.
CRC	16	Контрольная сумма.
Конец	28	Пауза (тишина) продолжительностью не менее 3,5 шестнадцатеричных символов.

Признаком начала кадра является пауза (тишина) продолжительностью не менее 3,5 шестнадцатеричных символов. Кадр должен передаваться непрерывно. Если при передаче кадра обнаруживается пауза продолжительностью более 1,5 шестнадцатеричных символа, то считается, что кадр содержит ошибку и должен быть отклонен принимающим модулем. Эти величины пауз должны строго соблюдаться при скоростях ниже 19200 бит/с, однако при более высоких скоростях рекомендуется использовать фиксированные значения паузы, 1,75 мс и 750 мкс соответственно.

Адрес – адрес ведомого устройства, к которому адресован запрос. В случае Modbus RTU каждое устройство, подключенное к одной и той же магистрали должно иметь свой уникальный сетевой адрес, отличный от других устройств. Может иметь значения h01...hFE (1...254). Адрес, равный 00h, используется как общий, на него отвечают все счетчики в магистрали. Он предназначен для индивидуальной работы с одним счетчиком, к примеру, для определения значения его текущего адреса.

Код функции – определяет тип запроса к устройству. Имеет размер 1 байт.

Поле данных – при запросе описывает запрашиваемые параметры, их количество, номер записи и тд; при ответе – содержит в себе значения запрашиваемых параметров, либо коды ошибок, если запрос был составлен некорректно.

CRC (англ. Cyclic redundancy check) – контрольная сумма (CRC-16-ANSI так же известную как CRC-16-IBM) состоит из двух байт, предназначена для проверки целостности передаваемых данных. Контрольная сумма вычисляется передающим устройством и добавляется в конец сообщения. Принимающее устройство вычисляет контрольную сумму в процессе приема и сравнивает ее с полем CRC принятого сообщения.

Алгоритм генерации CRC:

- 1) 16-ти битный 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 0.
- 4) Если младший бит 0: повторяется шаг 3 (сдвиг).

Если младший бит 1: делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа A001h.

- 5) Шаги 3 и 4 повторяются восемь раз.

- 6) Повторяются шаги со второго по пятый для следующего байта сообщения. Это повторяется до тех пор, пока все байты сообщения не будут обработаны.

- 7) Финальное содержание регистра CRC и есть контрольная сумма.

Размещение CRC в сообщении:

При передаче 16 бит контрольной суммы CRC в сообщении, сначала передается младший байт, затем старший. Пример сообщения для значения CRC равной 1241h:

Адрес	Функция	Счетчик байт	Байт	Байт	Байт	Байт	Мл. CRC	Ст. CRC
							41	12

Modbus TCP – вариант Modbus использующийся для передачи данных по сетям TCP/IP, стандартный порт для таких соединений 502. Формат кадра Modbus TCP имеет отличия от кадра Modbus RTU, пример смотрите в табл. 2.

Имя поля	Длина (байт)	Описание
Номер транзакции (Transaction id)	2	Служит для контроля транзакций запрос/ответ. Инициализирует ведущей. Ведомый копирует из запроса в ответ.
Идентификатор протокола (Protocol id)	2	Равен 0 для Modbus TCP.
Длина	2	Количество следующих дальше в этом кадре байт.
Адрес (Unit id)	1	Адрес ведомого устройства, которому адресован запрос. В ESM для Modbus TCP адрес равен 01h.
Код функции	1	Код функции определяет тип выполняемой операции.
Данные	n	Данные в ответе, или запросе.

Modbus RTU over TCP – тот же Modbus RTU, но передаваемый по TCP/IP, используя сеть Ethernet, а не последовательную (serial) линию.

От Modbus TCP отличие будет в том, что отсутствуют номер транзакции, идентификатор протокола, длина, вместо UnitID используется SlaveID и в конце добавится контрольная сумма. В итоге «экономия» 4 байта на каждой посылке.

Служебные утилиты ES Конфигуратор и ESM Test ведут обмен с ESM по Modbus RTU. В ES Конфигураторе есть возможность принудительного переключения на протокол Modbus на время работы с устройством.

В дальнейшем приводится описание на уровне элементов данных приложения (PDU).

1.3 Поддерживаемые команды

Команда	Название	Описание
Стандартные Modbus-команды		
0x01	read coil	Чтение дискретной информации
0x02	read discrete inputs	
0x03	read holding registers	Чтение измерений
0x04	read input registers	
0x05	write single coil	Телеуправление
0x2B	read ID	Чтение описания прибора
Пользовательские Modbus-команды		
0x64	service read	Чтение служебной информации
0x66	read meters	Чтение измерений
0x67	read tariff	Чтение тарифных данных
0x6A	read harmonics	Чтение гармоник
0x6B	read profiles	Чтение профилей и журналов
0x6C	read info	Чтение информации о приборе
0x6D	read dio info	Чтение информации о дискретных сигналах

Адреса регистров стандартных команд настраивается через ПО «ES Конфигуратор». Формат ответа в соответствии с протоколом Modbus.

Для пользовательских команд адресация статичная. В ответе младший байт передаётся первым.

1.4 Примечания



Внимание! Программное обеспечение постоянно совершенствуется и дополняется новыми функциональными настройками. Производитель оставляет за собой право вносить изменения и улучшения в ПО и аппаратную часть устройства без уведомления потребителей.

1 Чтение информации о приборе

Для чтения информации об устройстве авторизация не требуется. Пример запроса приведен в табл. 3.

Таблица 3

Код функции 6Ch	Параметры -
---------------------------	----------------

Структура ответа приведена в табл. 4.

Таблица 4

Имя поля	Тип	Описание, значения в DEC
serial	UInt32	Серийный номер
software	UInt32	Номер прошивки: A.B.C.D, где A, B, C и D – полученные значения байт
hardware	Byte	Номер аппаратной части: N/10, где N полученное число
iface	Byte	Наборы интерфейсов: 0 - "A2E2"; 1 - "A2E4".
power	Byte	Напряжение питания: 0 - "220 В"; 1 - "24 В".
dsp	Byte	Исполнение по вводу измеряемых параметров: 1 - "HV"; 2 - "ET"; 3 - "SV".
support61850	Byte	Наличие поддержки МЭК 61850-8-1 0 - "без поддержки"; 1 - "есть поддержка".
nominalU	Byte	Номинальное напряжение: 5 - "57,7/100 В"; 23 - "230/400 В"; 40 - "400/690 В".
nominalI	Byte	Номинальный ток: 5 - "5 А";
nominalF	Byte	Номинальная частота: 50 - "50 Гц"; 60 - "60 Гц".
KU	float	Коэффициент трансформации по напряжению
KI	float	Коэффициент трансформации по току
KP	float	Коэффициент трансформации по мощности
etType	Byte	Тип для исполнения ET: XY, где X и Y – это обозначения в соответствии с кодом заказа по току и напряжению соответственно.
accuracy	Byte	Класс точности: 0 - "02A - Wp: 0.2S, Wq: 0.5, ПКЭ класс A"; 1 - "02S - Wp: 0.2S, Wq: 0.5, ПКЭ класс S"; 2 - "05A - Wp: 0.5S, Wq: 1.0, ПКЭ класс A"; 3 - "05S - Wp: 0.5S, Wq: 1.0, ПКЭ класс S".
reserv	UInt32	Резерв

2 Чтение результатов текущих измерений

2.1 Стандартные команды

Чтение результатов текущих измерений не требует процедуры авторизации и происходит согласно стандартным функциям протокола Modbus: 03h – чтение регистров хранения и 04h – чтение регистров ввода. Команды для ESM-а полностью равнозначны. Ответ согласно протоколу Modbus. Структура запросов приведена в табл. 5 и 6.

Таблица 5

Код функции 03h	Адрес регистра 2 байта	Количество регистров 2 байта
---------------------------	---------------------------	---------------------------------

Таблица 6

Код функции 04h	Адрес регистра 2 байта	Количество регистров 2 байта
---------------------------	---------------------------	---------------------------------

Регистры настраиваются с помощью конфигулятора. Перечень доступных параметров приведен в табл. 8.

2.2 Пользовательские команды

Запрос результатов текущих измерений возможен также по команде 66h. Для команды 66h структура регистров не настраивается и остается всегда постоянной, в соответствии с табл. 8.

Пример запроса приведен в табл. 7.

Таблица 7

Код функции 66h	Адрес регистра 2 байта	Количество регистров 2 байта
---------------------------	---------------------------	---------------------------------

Стартовый адрес регистров измерений 128. Размер регистра – 4 байта. Порядок регистров в соответствии с таблицей 8.

Команда 66h используется для запроса только аналоговых величин, для получения дискретных сигналов см п. 7 (Дискретные сигналы (DIO)).

```

// Дискретные сигналы -----
idDIOFirst = 0,           // 64 DIO
idDIOLast = 63,

// дополнительные регистры -----
idExtFirst = 64,          // 64 внешних регистра (4 байта)
idExtLast = 127,

// RMS (быстрые или медленные) -----

public float Ua, Ub, Uc, U;           // фазное напряжение
public float Uab, Ubc, Uca, Ul;       // линейное напряжение
public float Ia, Ib, Ic, I;           // ток
public float Pa, Pb, Pc, P;           // активная мощность
public float Qa, Qb, Qc, Q;           // реактивная мощность
public float Sa, Sb, Sc, S;           // полная мощность
public float Fa, Fb, Fc, F;           // частота

public float U1a, U1b, U1c, U1;       // фазное напряжение первой гармоники
public float U1ab, U1bc, U1ca, U1l;   // линейное напряжение первой гармоники
public float I1a, I1b, I1c, I1;       // ток первой гармоники
public float P1a, P1b, P1c, P1;       // активная мощность первой гармоники
public float Q1a, Q1b, Q1c, Q1;       // реактивная мощность первой гармоники
public float S1a, S1b, S1c, S1;       // полная мощность первой гармоники

public float FiU1a, FiU1b, FiU1c;     // фаза напряжения первой гармоники
public float FiI1a, FiI1b, FiI1c;     // фаза тока первой гармоники
public float FiUab, FiUbc, FiUca;     // угол между фазными напряжениями
public float FiIa, FiIb, FiIc;        // угол между током и напряжением
public float FiSa, FiSb, FiSc;        // эквивалентный угол между током и напряжением
                                     // (рассчитывается из P,Q,S RMS)
public float FiIz, FiIp, FiIn;         // угол между током и напряжением нулевой,
                                     // прямой, обратной последовательности
public float Ca, Cb, Cc, C;           // коэффициент мощности (P/S, косинус угла между
                                     // током и напряжением)
public float TGa, TGb, TGc, TG;       // коэффициент мощности (Q/P, тангенс угла между
                                     // током и напряжением)

public float Uz, Up, Un;               // напряжение нулевой, прямой, обратной
                                     // последовательности (сумма векторов)
public float Iz, Ip, In;               // ток нулевой, прямой, обратной
                                     // последовательности (сумма векторов)
public float Pz, Pp, Pn;               // активная мощность нулевой, прямой, обратной
                                     // последовательности
public float Qz, Qp, Qn;               // реактивная мощность нулевой, прямой, обратной
                                     // последовательности
public float Sz, Sp, Sn;               // полная мощность нулевой, прямой, обратной
                                     // последовательности

public float KuUz, KuUn, KuIz, KuIn;   // коэффициенты несимметрии токов напряжений по
                                     // нулевой и обратной последовательностям

public float KdUa, KdUb, KdUc, KdU;   // коэффициент несинусоидальности фазного
                                     // (линейного) напряжения
public float KdIa, KdIb, KdIc, KdI;   // коэффициент несинусоидальности по току

public float THDpa, THDpb, THDpc, THDp; // суммарный коэффициент гармонических
                                     // искажений по активной мощности

```

```

public float THDpa, THDpb, THDpc, THDp; // суммарный коэффициент гармонических
                                         // искажений по активной мощности
public float THDqa, THDqb, THDqc, THDq; // суммарный коэффициент гармонических
                                         // искажений по реактивной мощности
public float THDsa, THDsb, THDsc, THDs; // суммарный коэффициент гармонических
                                         // искажений по полной мощности

// дополнительные измерения
public float Uzr, Izr; // RMS параметры нулевой последовательности (В
                       // ESM-ET с дополнительных входов)

public float Pzr, Qzr, Szr;
public float DFa, DFb, DFc, DF; // росcof (производная частоты)
public float Psta, Pstb, Pstc, Pst; // кратковременная доза фликера (10 мин)
public float Plta, Pltb, Pltc, Plt; // долговременная доза фликера (2 часа)
public float Bx, By, Bz, B; // показания датчика магнитного поля
public float T; // температура
public float Vbat; // напряжение батарейки

public float rsrv1, rsrv2, rsrv3, rsrv4;
public float rsrv5, rsrv6, rsrv7, rsrv8;
public float rsrv9;

public float idDWp, idDWq, idDWs; // приращение энергии за 200 мс

// энергия -----
public ulong Wpp, Wpm, Wqp, Wqm; // P+, P-, Q+, Q-
public ulong Wq1, Wq2, Wq3, Wq4; // Q1,2,3,4
public ulong Wsp, Wsm; // S+, S-

// энергия основной гармоники
public ulong W1pp, W1pm, W1qp, W1qm;
public ulong W1q1, W1q2, W1q3, W1q4;
public ulong W1sp, W1sm;

// энергия прямой последовательности
public ulong WFpp, WFpm, WFqp, WFqm;
public ulong WFq1, WFq2, WFq3, WFq4;
public ulong WFsp, WFsm;

// некачественная энергия
public ulong WBpp, WBpm, WBqp, WBqm;
public ulong WBq1, WBq2, WBq3, WBq4;
public ulong WBsp, WBsm;

```

2.3 Формат данных

- **Целочисленные величины (short)**

В этом формате передаются измерения, запрашиваемые по 03h и 04h командам.

Для перевода масштабируемых величин необходимо значение регистра умножить на квант соответствующей величины:

Квант	Описание, единица измерения	Величина кванта		
		I _{ном} = 5 А		
		Уном.ф = 57,7 В	Уном.ф = 220 В	Уном.ф = 380 В
I	Квант тока, А	0,001		
U	Квант напряжения, В	0,01	0,04	0,08
P/Q/S/W	Квант мощности, энергии, Вт(/ч)/Вар(/ч)/ВА	0,01	0,04	0,08
F	Квант частоты, Гц	0,001		
cosφ, tgφ	Квант cosφ, tgφ	0,001		
φ	Квант φ	0,01		
t	Квант температуры, °C	0,1		

- **Величины в формате с плавающей запятой (float)**

Регистры содержат значения в форме мантиссы и показателя степени. Представление утверждено в стандарте [IEEE 754](#).

Все данные передаются с учетом коэффициента трансформации и единиц измерения, заданных в приборе.

Запрос коэффициентов и единиц измерения:

Таблица 9

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
00h	Стартовый адрес регистра
6Eh	
00h	Количество регистров
09h	
D1h	Контрольная сумма (CRC)
D9h	

Ответ:

Таблица 10

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
12h	18 – длина ответа
4 байта (float)	Коэффициент трансформации по напряжению
4 байта (float)	Коэффициент трансформации по току
4 байта (float)	Коэффициент трансформации по мощности; если 0, то коэффициент по мощности равен произведению коэффициентов по току и напряжению
2 байта (short)	Единицы измерения напряжения: 0 – В; 1 – кВ.
2 байта (short)	Единицы измерения тока: 0 – А; 1 – кА;
2 байта (short)	Единицы измерения мощности: 0 – Вт/вар/ВА; 1 – кВт/квар/кВА; 2 – МВт/Мвар/МВА.
6Fh	Контрольная сумма (CRC)
81h	

- **Целочисленные величины (long и long long)**

Эти форматы используются для передачи значений энергии. Для перевода в Вт*ч/вар*ч/ВА*ч полученную величину необходимо домножить на 0,001.

3 Чтение гармоник и интергармоник

Запрос:

Код функции 6Ah	Стартовый номер гармоники 2 байта	Количество гармоник 2 байта
---------------------------	--------------------------------------	--------------------------------

Номер гармоники:

Значение	Номер
00h	1 гармоника
01h	2 гармоника
...	...
31h	50 гармоника
32h	0,5 интергармоника
33h	1,5 интергармоника
...	...
63h	49,5 интергармоника

В одном запросе передаётся максимум пять гармоник.

Ответ:

Код функции 6Ah	Длина ответа 1 байт	Разделитель 1 байт	Данные N * 48 байт
---------------------------	------------------------	-----------------------	-----------------------

N – количество запрошенных гармоник (не более 5);

Длина ответа – количество байт данных;

Разделитель – всегда 00h;

Данные – вектора тока и напряжения соответствующих гармоник. Один блок данных имеет следующую структуру:

Ua		Ub		Uc		Ia		Ib		Ic	
Mag	Ang	Mag	Ang	Mag	Ang	Mag	Ang	Mag	Ang	Mag	Ang

Mag – амплитуда вектора гармоники, 4 байта (float);

Ang – угол вектора гармоники, 4 байта (float).

4 Аутентификация

Процедура аутентификации необходима для доступа к результатам измерений, конфигурационным данным ESM. Аутентификация совершается поэтапно:

4.1 Запрос ключа авторизации

Запрос ключа совершается с помощью кода функции 64h со следующими параметрами (см. табл. 11)

Таблица 11

Код функции 64h	Параметры 00h 01h 00h 08h
---------------------------	-------------------------------------

В ответе будет ключ из 16-и байт.

4.2 Подготовка запроса авторизации

В соответствии с симметричным алгоритмом блочного шифрования AES (Advanced Encryption Standard) происходит создание объекта-шифратора. В качестве ключа берется хеш-сумма пароля доступа к устройству, взятая по алгоритму MD5 (MD5 Message-Digest Algorithm в соответствии с [rfc1321](#)). В качестве вектора инициализации принимается значение **null**. С помощью полученного объекта шифратора происходит шифрование принятого от устройства ключа авторизации.

4.3 Отправка запроса авторизации

Для отправки преобразованного ключа авторизации, полученного от устройства, используется код запроса, к которому добавляется регистр из двух байт, отвечающий за запрашиваемый уровень доступа (младший байт первый), плюс подготовленный ключ авторизации (см. табл. 12). Уровни доступа приведены в табл. Таблица 13.

Таблица 12

Код функции 65h	Параметры 00h 00h 00h 09h 12h	Номер уровня доступа 2 байта	Зашифрованный паролем доступа ключ авторизации
---------------------------	-----------------------------------------	---------------------------------	------------------------------------------------

Таблица 13

Номер уровня доступа	Описание	Пароль по умолчанию
1	Уровень пользователя.	111111
2	Уровень администратора.	222222
3	Уровень производителя устройства.	

4.4 Чтение результата авторизации

В случае неуспешной авторизации будет возвращен код ошибки.

Таблица 14

Код ошибки	Описание
4	Введен неверный пароль
6	Запрошен несуществующий уровень доступа
7	Ключ авторизации устарел
8	Превышено число попыток ввода пароля, авторизация заблокирована

Пример функции авторизации приведен в табл. 15.

```

private void Authorize()
{
    int len;
    byte[] data = new byte[256];
    byte[] getKey = { 0x01, 0x64, 0x00, 0x01, 0x00, 0x08 };

    len = getKey.Length;
    getKey.CopyTo(data, 0);
    ModbusTransaction(data, ref len);

    if(len != 21)
        throw new Modbus.IOException("Авторизация не удалась!\r\nНе удалось запросить
ключ авторизации.");

    byte[] authKey = data.Skip(3).Take(16).ToArray();

    MD5 md5 = MD5.Create();
    byte[] aesKey = md5.ComputeHash(Encoding.Default.GetBytes(config.pass));
    byte[] authReq = new byte[16];

    Aes aes = Aes.Create();
    aes.KeySize = 128;
    aes.BlockSize = 128;
    aes.Padding = PaddingMode.None;
    aes.Mode = CipherMode.ECB;
    ICryptoTransform encrypter = aes.CreateEncryptor(aesKey, null);
    encrypter.TransformBlock(authKey, 0, 16, authReq, 0);

    byte[] writeAuth = { 0x01, 0x65, 0x00, 0x00, 0x00, 0x09, 0x12 };

    writeAuth.CopyTo(data, 0);
    data[7] = (byte)config.accessLevel;
    data[8] = 0;
    authReq.CopyTo(data, 9);

    len = 7 + 2 + 16;
    ModbusTransaction(data, ref len);

    if(len != 8 || ((data[1] & 0x80) != 0))
    {
        string s = "Авторизация не удалась!";

        if((data[1] & 0x80) != 0)
        {
            string error;

            switch(data[2])
            {
                case 4: error = "Введен неправильный пароль!"; break;
                case 6: error = "Запрошен несуществующий уровень доступа!"; break;
                case 7: error = "Ключ авторизации устарел!"; break;
                case 8: error = "Превышено число попыток ввода пароля, авторизация
заблокирована!"; break;
                default: error = "Код ошибки: " + data[2]; break;
            }

            s += "\r\n" + error;
        }

        throw new Modbus.IOException(s);
    }
}

```

5 Чтение энергии по тарифам

Чтение энергии по тарифам не требует процедуры авторизации и происходит по 67h функции Modbus.

Структура запроса приведена в табл. 16.

Таблица 16

Код функции 67h	Вид энергии 1 байт	Тип энергии 1 байт	Метка времени 4 байта
---------------------------	-----------------------	-----------------------	--------------------------

Доступные виды энергии:

Значение	Вид
00h	Текущая
01h	Срез за сутки
02h	Срез за месяц
03h	Срез за год

Типы энергии:

Значение	Тип	Примечание
00h	Тариф 1	
01h	Тариф 2	
02h	Тариф 3	
03h	Тариф 4	
04h	Тариф 5	
05h	Тариф 6	
06h	Тариф 7	
07h	Тариф 8	
FFh	Суммарная	Доступно только для текущей энергии
FEh	Текущий тариф	
FDh	Первой гармоники	
FCh	Прямой последовательности	
FBh	Некачественная	

Метка времени указывается для срезов энергии в формате unix time; для текущей энергии поле время любое.

Ответ:

Таблица 17

Код функции 67h	Разделитель 1 байт	Длина 1 байт	Данные 80 байт
---------------------------	-----------------------	-----------------	-------------------

Разделитель – всегда 0;

Данные:

Обозначение	Описание	Формат
WPr	Энергия активная принятая	Long long
WPm	Энергия активная отданная	Long long
WQp	Энергия реактивная принятая	Long long
WQm	Энергия реактивная отданная	Long long
WQ1	Энергия реактивная в первом квадранте	Long long
WQ2	Энергия реактивная во втором квадранте	Long long
WQ3	Энергия реактивная в третьем квадранте	Long long
WQ4	Энергия реактивная в четвертом квадранте	Long long
WSp	Энергия полная принятая	Long long
WSm	Энергия полная отданная	Long long

Единицы измерения: мВт*ч.

6 Журналы

6.1 Установка начального и конечного значения времени для чтения записей

Эта команда устанавливает диапазон времени, для последующего чтения. При успешном выполнении команды, ESM запоминает начало и конец диапазона и сбрасывает счетчик считанных записей. Если начало и конец диапазона установить в 0, то чтение произойдет для всего журнала. При успешном выполнении команды и до окончания операции чтения (команда 2) по любым другим каналам чтение журналов становится невозможным (также невозможно чтение других типов журнала). При отсутствии команд чтения в течение 10 секунд блокировка снимается.

Таблица 18. Команда установки интервала чтения записей.

Код функции 6Bh	Установка времени 1 байт	Тип журнала 1 байт	Время начала диапазона чтения 4 байта	Время окончания диапазона чтения 4 байта
---------------------------	-----------------------------	-----------------------	------------------------------------------	---------------------------------------------

Таблица 19. Ответ на команду установки интервала чтения записей.

Код функции 6Bh	Установка времени 1 байт	Тип журнала 1 байт
---------------------------	-----------------------------	-----------------------

Пример запроса установки интервала чтения записей и ответ приведены ниже.

Запрос:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
00h	0 – установка времени
00h	Тип журнала (TMagazineType)
80h	UTC время начала диапазона чтения (младший байт первый)
C1h	
85h	Например, 01.01.2016 = 1451606400 = 0x5685C180 = 80h C1h 85h 56h
56h	
00h	UTC время конца диапазона чтения (младший байт первый)
A0h	
AEh	
56h	
31h	Контрольная сумма (CRC)
ADh	

Ответ:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
00h	0 – установка времени
00h	Тип журнала (TMagazineType)
70h	Контрольная сумма (CRC)
04h	

Коды ошибок:

Коды ошибок	Описание
1	Некорректная длина команды (ожидается 14 байт).
2	Конец диапазона меньше начала.
3	Уже идет чтение по другому каналу / повторная установка с другим типом журнала.

6.2 Чтение записей

При чтении ESM перебирает все записи в журнале и высылает в ответ только те, которые попадают в запрошенный диапазон. После передачи всех записей в ответе на данную команду вернется 0 в поле число записей. Номер запрашиваемой записи необходим для обработки повторных запросов (в случаях, когда не удалось принять ответ от ESM) и всегда должен быть равен числу принятых записей. Подготовка ответа может занять до 2 секунд (зависит от числа записей, которые требуется пропустить, чтобы найти попадающие в запрошенный диапазон).

Значения по профилю выдаются в квантах, смотрите таблицу преобразования масштабированных величин.

Таблица 20. Команда чтения записей.

Код функции	Чтение записей	Тип журнала	Число принятых записей
6Bh	1 байт	1 байт	2 байта

Таблица 21. Ответ на команду чтения записей.

Код функции 6Bh	Чтение записей 1 байт	Тип журнала 1 байт	Число записей в ответе 2 байта	Общее число отправленных записей 2 байта	Записи (кол-во записей х длину записи)
------------------------	--------------------------	-----------------------	-----------------------------------	---------------------------------------------	-------------------------------------------------

Пример запроса на чтение записей и ответ приведены ниже.

Запрос:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
00h	1 – чтение записей
00h	Тип журнала (TMagazineType)
00h	Число принятых записей
00h	
5Ah	Контрольная сумма (CRC)
D0h	

Ответ:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
00h	1 – чтение записей
00h	Тип журнала (TMagazineType)
01h	Число записей в ответе (младший байт первый)
00h	
01h	Общее число отправленных записей (младший байт первый)
00h	
...	Записи: число байт = число записей * размер записи
70h	Контрольная сумма (CRC)
04h	

Коды ошибок:

Коды ошибок	Описание
1	Некорректная длина команды (ожидается 8 байт).
2	Не установлен диапазон времени / тип журнала не соответствует заданному по команде 0.
3	Нарушена последовательность чтения (номер запрошенной записи не совпадает с числом принятых или отправленных записей).
4	Ошибка чтения журнала, возможно аппаратный сбой.

6.3 Завершение процедуры чтения записей

После окончания чтения журналы остаются заблокированными в течение 10 секунд. Для того, чтобы разрешить чтение другого типа журнала или чтение по другим каналам, необходимо отправить команду 3.

Таблица 22. Команда на завершение чтения записей.

Код функции	Окончание чтения	Тип журнала
6Bh	02h	(1 байт)

Таблица 23. Ответ на команду завершения чтения записей.

Код функции	Окончание чтения	Тип журнала
6Bh	02h	(1 байт)

Пример запроса на чтение записей и ответ приведены ниже.

Запрос:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
02h	2 – окончание чтения
00h	Тип журнала (TMagazineType)
71h	Контрольная сумма (CRC)
64h	

Ответ:

Значение	Описание параметра
01h	Адрес устройства
6Bh	Команда
02h	2 – окончание чтения
00h	Тип журнала (TMagazineType)
71h	Контрольная сумма (CRC)
64h	

Коды ошибок:

Коды ошибок	Описание
1	Некорректная длина команды (ожидается 6 байт)
2	Блокировка произведена по другому каналу (снять блокировку можно только по каналу, который начал чтение) / тип журнала не соответствует заданному по команде 0.

6.4 Структура типов журналов

№	Тип журнала	Тип структуры	Описание
0	Profile1	TProfileRecord	Профиль №1
1	Profile2	TProfileRecord	Профиль №2
2	ProfileDays	TEnergyRecord	Накопленная энергия за сутки
3	ProfileMonths	TEnergyRecord	Накопленная энергия за месяц
4	DIO	TDIORecord	Изменения состояний дискретных сигналов
5	Auth	TEventRecord	Успешная авторизация
6	AuthFail	TEventRecord	Ошибка авторизации
7	EnergyReset	TEventRecord	Сброс данных по накопленной энергии
8	ProfilesReset	TEventRecord	Сброс данных по профилям
9	Power	TEventRecord	Сброс, включение, выключение устройства
10	CoeffChange	TEventRecord	Изменение коэффициентов трансформации
11	Voltage	TEventRecordEx	Перенапряжение, конец перенапряжения, провал напряжения, конец провала напряжения
12	Time	TEventRecordEx	Синхронизация времени
13	FlashUpdate	TEventRecordEx	Обновление прошивки, изменение настроек, изменение пароля, изменение калибровочных коэффициентов
14	CorrectChange	TEventRecord	Изменение поправочных коэффициентов
15	MemoryTest	TEventRecord	Тестирование энергонезависимой памяти

6.5 Структура временных интервалов

```

typedef enum
{
    LenSeconds1 = 0,
    LenSeconds2,
    LenSeconds3,
    LenSeconds4,
    LenSeconds5,
    LenSeconds6,
    LenSeconds10,
    LenSeconds12,
    LenSeconds15,
    LenSeconds20,
    LenSeconds30,
    LenMinutes1,
    LenMinutes2,
    LenMinutes3,
    LenMinutes4,
    LenMinutes5,
    LenMinutes6,
    LenMinutes10,
    LenMinutes12,
    LenMinutes15,
    LenMinutes20,
    LenMinutes30,
    LenHours1,
    LenHours2,
    LenHours3,
    LenHours4,
    LenHours6,
    LenHours8,
    LenHours12,
    LenDay,
    LenMonth,    // не используется!
    LenYear,     // не используется!
}TProfileTimeLength;

```

6.6 Журнал профиля

6.6.1 Структура записи

```

typedef struct
{
    unsigned long V1;           // Настраиваемый параметр 1
    unsigned long V2;           // Настраиваемый параметр 2
    unsigned long V3;           // Настраиваемый параметр 3
    unsigned long V4;           // Настраиваемый параметр 4

    unsigned long utcTime;      // UTC время начала получасовки

    unsigned short flags:11;    // флаги состояния получасовки
    unsigned short length:5;    // размер записи (номер в TProfileTimeLength)

    unsigned short crc;         // для контроля состояния памяти
}TProfileRecord;

```

6.6.2 Флаги состояния получасовок

Флаг	Значение	Описание
PROFILE_FLAG_BROKEN	01h	Запись не полная (было отключение питания)
PROFILE_FLAG_QUALITY	02h	Зафиксированы нарушения качества
PROFILE_FLAG_SYNC	04h	Произошла синхронизация времени
PROFILE_FLAG_STREAMS_UNSYNC	08h	Входящие потоки данных не синхронизированы

6.6.3 Получения списка параметров

Запрос перечня параметров профилей осуществляется по команде из табл. 24 . В ответе возвращается 8 регистров, первые четыре содержат информацию о параметрах первого профиля, вторые четыре – второго.

Запрос:

Таблица 24

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
01h	Стартовый адрес регистра
7Eh	
00h	Количество регистров
08h	
10h	Контрольная сумма (CRC)
20h	

Ответ:

Таблица 25

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
10h	16 – длина ответа
4 x 2 байта	Четыре параметра первого профиля; соответствие см. в табл. 8
4 x 2 байта	Четыре параметра второго профиля; соответствие см. в табл. 8
88h	Контрольная сумма (CRC)
D0h	

6.7 Структура записи журнала энергий

```
typedef struct
{
    unsigned long Pp;
    unsigned long Pm;
    unsigned long Q1;
    unsigned long Q2;
    unsigned long Q3;
    unsigned long Q4;

    unsigned long utcTime;

    unsigned short flags:11; // флаги состояния профиля
    unsigned short length:5; //размер записи (номер в TProfileTimeLength)

    unsigned short crc;      // для контроля состояния памяти
}TEnergyRecord;
```

6.8 Структура журнала состояний дискретных сигналов

```

typedef struct
{
    unsigned char DIO[8];           // состояние всех DIO
    unsigned char quality[8];       // качество всех DIO
    unsigned short ID:3;            // тип DIO
    unsigned short index:6;         // номер DIO
    unsigned short flagOff:1;       // DIO изменился, пока устройство было выключено
    unsigned short flagExtUnsync:1; // внешнее устройство не синхронизированно
    unsigned short flagExtNoTime:1; // время внешнего события не известно
    unsigned short flagReserv:4;    //
    unsigned short ms;              // время изменения ТС (старший бит - неактуальность
метки)
    unsigned long utcTime;
    unsigned long data;
}TDIORecord;

```

Доступные типы DIO:

Индекс	Наименование
0	Не определен
1	Дискретный выход
2	Дискретный вход
3	Уставка
4	Логическое выражение
5	Подписка на GOOSE-сообщение
6	Диагностика

Разбор структуры data:

```

switch(ID)
{
    case 0: comment = "NONE"; break; // тип 0 зарезервирован

    case 1:
        if((r.data & 0xFF0000) != 0)
            comment = "DO Pulse"; // встроенный дискретный выход

        else
            comment = $"DO Ext{(r.data & 0xFF) + 1}-{((r.data >> 8) & 0xFF) + 1}"; break; //номер внешнего устройства и номер DO в устройстве;

    case 2: comment = $"DI Ext{(r.data & 0xFF) + 1}-{((r.data >> 8) & 0xFF) + 1}";
    break; //номер внешнего устройства и номер DI в устройстве;
    case 3: comment = $"Setpoint {Meters[(ID)((r.data - 128) % ((int)Dws + 1))].Name}"; break;
    case 4: comment = "Logic"; break;
    case 5: comment = "GOOSE"; break;
    case 6:
        uint msgIndex = r.data & 0xFF;

        if(msgIndex < Service.Diagnostic.Errors.Length)
            comment = $"Diagnostic {Service.Diagnostic.Errors[msgIndex]} -> {((r.data & 0x80000000) == 0 ? "ВЫКЛ" : "ВКЛ")}";
        else
            comment = "Diagnostic bad record!"; break;

    default: comment = "ER"; break;
}

```

6.9 Структура записи журнала событий

```
typedef struct
{
    unsigned long utcTime;
    unsigned short ms;
    unsigned char data1;
    unsigned char data2;
}TEventRecord;
```

6.10 Структура расширенной записи журнала событий

```
typedef struct
{
    unsigned long utcTime;
    unsigned short ms;
    unsigned char data1;
    unsigned char data2;
    unsigned long dataEx;
}TEventRecordEx;
```

6.11 Разбор записи журнала событий

```

switch(type)
{
    case eMagazine.FlashUpdate:
        switch (r.data1)
        {
            case 0: e = new TEvent("Обновление прошивки"); e.data = "Новая версия: " +
String.Join(".", BitConverter.GetBytes(r.dataEx).Select(x => x.ToString()).ToArray()) ;
break;

            case 1: e = new TEvent("Изменение настроек"); break;
            case 2: e = new TEvent("Изменение пароля"); e.data = "Уровень " + r.data2;
break;

            case 3: e = new TEvent("Изменение калибровочных коэффициентов"); break;
                    default: e = new TEvent("Flash UNKNOWN!"); break;
        }
        break;

    case eMagazine.Time:
        e = new TEvent("Синхронизация времени");
        e.data = "Смещение: " + (((Int64)(int)r.dataEx << r.data1) * 1.0 /
0x100000000).ToString("0.000") + " с";
        break;

    case eMagazine.Voltage:
        switch(r.data2)
        {
            case 1: e = new TEvent("Перенапряжение"); break;
            case 2: e = new TEvent("Конец перенапряжения"); break;
            case 3: e = new TEvent("Провал напряжения"); break;
            case 4: e = new TEvent("Конец провала напряжения"); break;
            default: e = new TEvent("Voltage Error!"); break;
        }

        if(r.data1 == 0) e.data = "Фаза А";
        if(r.data1 == 1) e.data = "Фаза В";
        if(r.data1 == 2) e.data = "Фаза С";

        if(r.data2 == 2) e.data += "; Umax: " + BitConverter.ToSingle(BitConverter.Get-
Bytes(r.dataEx), 0);
        if(r.data2 == 4) e.data += "; Umin: " + BitConverter.ToSingle(BitConverter.Get-
Bytes(r.dataEx), 0);
        break;

        default: e = new TEvent("Неизвестное событие!"); break;
    }

    e.time = new DateTime(1970, 1, 1).AddSeconds(r.utcTime).AddMilliseconds(r.ms);

```

7 Дискретные сигналы (DIO)

Максимально в ESM доступно 64 DIO. Тип и параметры DIO настраиваются через конфигуратор.

7.1 Запрос состояний

Запрос

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
00h	Стартовый адрес
C8h	
00h	Количество регистров
08h	
F0h	Контрольная сумма (CRC)
3Ah	

Ответ:

Значение	Описание параметра
01h	Адрес устройства
64h	Команда
10h	16 – длина ответа
8 байт	Битовая строка: состояние всех DIO (0 – ОТКЛ, 1 - ВКЛ)
8 байт	Битовая строка: качество всех DIO (0 – invalid, 1 – valid)
F0h	Контрольная сумма (CRC)
3Ah	

7.2 Запрос информации о дискретных сигналах

Запрос информации о всех настроенных дискретных сигналах осуществляется по команде **6Dh**:

Значение	Описание параметра
01h	Адрес устройства
6Dh	Команда
00h	Стартовый адрес DIO
0Ch	Контрольная сумма (CRC)
90h	

Для получения информации о всех DIO первый запрос должен начинаться с нулевого адреса, следующий с номера последнего полученного DIO плюс 1 и т.д., пока в ответ не придет сообщения с 0 в поле длины.

Ответ:

Код функции 6Dh	Длина ответа 1 байт	Данные 1...250 байт
---------------------------	------------------------	------------------------

Поле данные состоит из блоков, каждый из которых содержит информацию об одном дискретном сигнале. Структура блока:

Описание	Размер	Возможные значения
Порядковый номер DIO	1 байт	0...63
Тип DIO	1 байт	1 – Дискретный выход; 2 – Дискретный вход; 3 – Уставка; 4 – Логическое выражение; 5 – GOOSE; 6 – Диагностика.
Длина информации о DIO	1 байт	8 – Дискретный выход; 2 – Дискретный вход; 20 – Уставка; 2 – Логическое выражение; 0 – GOOSE; 4 – Диагностика.
Длина названия DIO	1 байт	0...128
Информация о DIO	0...20 байт	См. таблицу 26
Название DIO	0...128 байт	Название в виде строки

Таблица 26

Тип	Размер (байт)	Значение
Дискретный выход	1	0 – внешний дискретный выход; 1 – встроенный дискретный выход;
	1	номер блокирующего DIO + 1; 0 – не используется;
	1	время удержания по modbus, сек.
	1	Резерв
	2	логическое выражение для авто ТУ
	1	номер внешнего устройства
	1	номер дискретного выхода внутри внешнего устройства
Дискретный вход	1	номер внешнего устройства
	1	номер DI внутри внешнего устройства
Уставка	2	Параметр уставки (см табл. 8)
	2	задержка срабатывания уставки - в 0,1 с
	1	0 – уставка с учетом знака; 1 – уставка работает по модулю.
	1	0 - уставка по понижению, 1 - по повышению, 2 - по выходу из диапазона, 3 и все остальное - по входу в диапазон
	2	Резерв
	4 (float)	Верхняя граница диапазона
	4 (float)	Нижняя граница диапазона
	4 (float)	Гистерезис
Логическое выражение	2	Настройки логического выражения
Диагностика	4	Битовая маска настроенных диагностических сообщений (расшифровку см. в табл. 28)

8 Синхронизация времени

8.1 Запрос текущего времени

Запрос времени совершается с помощью кода функции 64h со следующими параметрами (см. табл. 27)

Таблица 27

Код функции	Параметры
64h	00h 46h 00h 09h

Пример чтения и разбора метки времени представлен ниже:

```
public void readTime()
{
    byte[] data = new byte[256];
    byte[] cmd = { 0x01, 0x64, 0x00, 0x46, 0x00, 0x09 };

    if(Modbus.Request(cmd, data) != 23)
    {
        error = true;
        return;
    }

    error = false;

    DateTime now = DateTime.UtcNow;

    long ticks = BitConverter.ToInt32(data, 3) * TimeSpan.TicksPerSecond;
    long frac = BitConverter.ToInt32(data, 7);

    ticks += (frac * TimeSpan.TicksPerSecond + 0x100000000 / 2) / 0x100000000;

    currentTime = new DateTime(1970, 1, 1).AddTicks(ticks);

    shift = currentTime.Subtract(now).TotalSeconds;
    msec = (int)(frac * 1000 / 0x100000000);

    dayShift = BitConverter.ToInt64(data, 11) * 1.0 / 0x100000000;
    syncCount = BitConverter.ToInt16(data, 19);
}
```

8.2 Корректировка времени

```

private string correctTime()
{
    double d;
    string s = tbCorrectTime.Text.Replace(",", CultureInfo.CurrentCulture.NumberFormat.CurrencyDecimalSeparator);
    s = s.Replace(".", CultureInfo.CurrentCulture.NumberFormat.CurrencyDecimalSeparator);

    if(!double.TryParse(s, out d))
        throw new IOException("Некорректное смещение!");
    long delta = (long)(d * 0x100000000 + 0.5 * Math.Sign(d));

    byte[] cmd = { 0x01, 0x65, 0x00, 0x4F, 0x00, 0x04, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
    byte[] data = new byte[256];

    BitConverter.GetBytes(delta).CopyTo(cmd, 7);

    if(Modbus.Request(cmd, data) != 8 || ((data[1] & 0x80) != 0))
    {
        if(data[2] == 0x04) throw new IOException("Недостаточный уровень доступа!\r\nУстановите уровень доступа не менее 1 в настройках.");
        if(data[2] == 0x05) throw new IOException("Превышено максимальное время коррекции. Для коррекции времени можно увеличить уровень авторизации в настройках.");

        throw new IOException("Не удалось скорректировать время!");
    }

    return "Время скорректировано";
}

```

8.3 Установка времени

```

private string setTime()
{
    DateTime time;

    if(rbSystemTime.Checked)
        time = DateTime.UtcNow;
    else
    {
        time = dateSet.Value;
        time = time.Add(timeSet.Value.TimeOfDay);
        time = TimeZoneInfo.ConvertTimeToUtc(time, TimeZoneInfo);
    }

    uint sec = (uint)time.Subtract(new DateTime(1970, 1, 1)).TotalSeconds;
    uint frac = (uint)((time.TimeOfDay.Ticks % TimeSpan.TicksPerSecond) * 0x100000000
/ TimeSpan.TicksPerSecond);

    byte[] cmd = { 0x01, 0x65, 0x00, 0x46, 0x00, 0x04, 0x08, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00 };
    byte[] data = new byte[256];

    BitConverter.GetBytes(sec).CopyTo(cmd, 7);
    BitConverter.GetBytes(frac).CopyTo(cmd, 11);

    if(Modbus.Request(cmd, data) != 8 || ((data[1] & 0x80) != 0))
    {
        if(data[2] == 0x04) throw new IOException("Недостаточный уровень до-
ступа!\r\nУстановите уровень доступа не менее 1 в настройках.");
        if(data[2] == 0x05) throw new IOException("Превышено максимальное время
коррекции. Для установки времени можно увеличить уровень авторизации в настройках.");

        throw new IOException("Не удалось установить время!");
    }

    time = TimeZoneInfo.ConvertTime(time, TimeZoneInfo);
    return "Время установлено: " + time.ToLongTimeString() + " " + time.ToLong-
DateString();
}

```

9 Диагностические данные

9.1 Описание диагностических регистров

Ошибки прибора:

Таблица 28

Номер бита	Ошибка
0	Ошибка энергонезависимой памяти
1	Неисправность микросхемы Ethernet
2	Неисправность внутренних часов
3	Напряжение батареи меньше 2,5 В
4	Превышено число попыток ввода пароля
5	Ошибка SD карты
6	Отсутствует синхронизация времени
7	Ошибка опроса внешних устройств
8	Неисправность внешнего ТУ
9	Резерв 1
10	Резерв 2
11	Резерв 3
12	Резерв 4
13	Резерв 5
14	Резерв 6
15	Резерв 7
16	Неисправность DSP
17	Неисправность АЦП
18	Неисправность микросхемы Ethernet-SV
19	Отсутствует поток SV
20	Отсутствует синхронизация потока SV
21	Используется резервный поток SV
22	Ошибка датчика магнитного поля
23	Зафиксировано внешнее магнитное поле
24	Ошибка АЦП монитора питания
25	Сбой напряжения +12В
26	Сбой напряжения -12В
27	Ошибка микросхемы DSP-ID
28	DSP PRP: Отсутствует LAN A
29	DSP PRP: Отсутствует LAN B
30	DSP PRP: Перепутаны порты A/B
31	DSP PRP: Перепутаны порты B/A