

**ПРЕОБРАЗОВАТЕЛЬ ИЗМЕРИТЕЛЬНЫЙ  
МНОГОФУНКЦИОНАЛЬНЫЙ  
АЕТ**

**Протокол информационного обмена  
49501860.70018-019001**

## 1 Основные положения

1.1 Настоящий документ распространяется на преобразователи измерительные многофункциональные АЕТ100, АЕТ200, АЕТ300, АЕТ400 (далее – преобразователь), работающие в составе автоматизированной системы диспетчерского управления.

Преобразователь обеспечивает измерение параметров трехпроводных и четырехпроводных электрических сетей переменного тока и выдачу результатов по интерфейсу RS-485 в виде выходного кода.

Протокол обмена MODBUS-RTU, MODBUS-ASCII.

1.2 MODBUS - это протокол обмена данными между ведущим и ведомым устройствами. Ведущее устройство управляет всей последовательной деятельностью путем избирательного опроса одного или нескольких ведомых устройств. Протокол допускает одно ведущее устройство и до 247 ведомых устройств на общей линии. Каждому устройству присваивается адрес, чтобы отличать его от других подключенных устройств.

Устройства соединяются, используя технологию – «главный/подчиненный», при которой только одно устройство (главный) может инициировать передачу (сделать запрос). Другие устройства (подчиненные) передают запрашиваемые главным устройством данные или производят запрашиваемые действия. Типичное главное устройство включает в себя ведущий (HOST) процессор и панели программирования. Типичное подчиненное устройство – преобразователь.

Главный может адресоваться к индивидуальному подчиненному или может инициировать широкую передачу сообщения на все подчиненные устройства. Подчиненное устройство возвращает сообщение в ответ на запрос, адресуемый именно ему. Ответы не возвращаются при широковещательном запросе от главного.

1.3 Используется два способа передачи: ASCII и RTU. Пользователь выбирает необходимый режим вместе с другими параметрами (скорость передачи, режим паритета и т.д.) во время конфигурирования каждого преобразователя.

## 2 Режим ASCII

2.1 При использовании ASCII - режима каждый байт сообщения передается как два ASCII символа. Главное преимущество этого способа - время между передачей символов может быть до 1 сек. без возникновения ошибок при передаче.

Формат каждого байта в ASCII-режиме:

Система кодировки: Шестнадцатеричная, ASCII-символы 0 - 9, A - F

Назначение бит:

1 стартовый бит

7 бит данных, младшим битом вперед

1 бит паритета; нет бита паритета

1 стоповый бит, если есть паритет; 2 стоповых бита, если нет паритета

Контрольная сумма: Longitudinal Redundancy Chek (LRC)

## 3 Режим RTU

3.1 При использовании RTU - режима каждый байт сообщения содержит два 4-х битных шестнадцатеричных числа.

Каждое сообщение передается непрерывным потоком.

Формат каждого байта в RTU-режиме:

Система кодировки: 8-ми битная двоичная, шестнадцатеричная 0 - 9, A - F

Две шестнадцатеричные цифры содержатся в каждом 8-ми битном байте сообщения.

Назначение бит:

- 1 стартовый бит
- 8 бит данных, младшим значащим разрядом вперед
- 1 бит паритета; нет бита паритета
- 1 стоповый бит, если есть паритет; 2 стоповых бита, если нет паритета

Контрольная сумма: Cyclical Redundancy Check (CRC)

## 4 Содержание MODBUS сообщения

### 4.1 ASCII фрейм

4.1.1 В ASCII-режиме сообщение начинается с "двоеточия" (":" - ASCII 3A hex) и заканчивается последовательностью "возврат каретки - перевод строки" (CRLF - ASCII 0D и 0A hex).

Допустимые символы для передачи - это шестнадцатеричные цифры 0 - 9, A - F. Мониторы устройств, находящихся в сети, непрерывно отслеживают символ "двоеточие". Когда он принят, каждое устройство декодирует следующее поле сообщения (поле адреса) и т.д.

Интервал между символами сообщения не должен превышать 1 сек. Если интервал больше, то принимающее устройство распознает это как ошибку. Типичный фрейм сообщения показан на рисунке 1.

Старт	Адрес	Функция	Данные	LRC	Конец
1 символ (":")	2 символа	2 символа	N символов	2 символа	2 символа (CR LF)

Рисунок 1

### 4.2 RTU фрейм

4.2.1 В RTU режиме сообщение начинается с интервала тишины равного времени передачи 3.5 символов при данной скорости передачи в сети. Первым полем передается адрес устройства.

Вслед за последним передаваемым символом также следует интервал тишины продолжительностью не менее 3.5 символов. Новое сообщение должно начинаться не раньше этого интервала.

Таким образом, если новое сообщение начнется раньше интервала длительностью 3.5 символа, принимающее устройство воспримет его как продолжение предыдущего сообщения. В этом случае устанавливается ошибка, так как будет несовпадение контрольных сумм. Типичный фрейм сообщения показан на рисунке 2.

Старт	Адрес	Функция	Данные	CRC	Конец
T1-T2-T3-T4	8 бит	8 бит	N x 8 бит	16 бит	T1-T2-T3-T4

Рисунок 2

### 4.3 Содержание адресного поля

4.3.1 Адресное поле фрейма содержит два символа (ASCII) или 8 бит (RTU). Допустимый адрес передачи находится в диапазоне 0 - 247. Каждому подчиненному устройству присваивается адрес в пределах от 1 до 247.

Адрес 0 используется для широковещательной передачи, его распознает каждое устройство. Когда MODBUS протокол используется на более высоком уровне сети, широковещательная передача может не поддерживаться или может быть реализована другими методами.

## 4.4 Содержание поля функции

4.4.1 Поле функции фрейма содержит два символа (ASCII) или 8 бит (RTU). Диапазон числа от 1 до 255.

Когда подчиненный отвечает главному, он использует поле кода функции для фиксации ошибки. В случае нормального ответа подчиненный повторяет оригинальный код функции. Если имеет место ошибка, возвращается код функции с установленным в 1 старшим битом.

Например, сообщение от главного подчиненному - прочитать группу регистров, имеет следующий код функции:

0000 0100 (04 hex)

Если подчиненный выполнил затребованное действие без ошибки, он возвращает такой же код. Если имеет место ошибка, то он возвращает:

1000 0100 (84 hex)

В добавление к изменению кода функции, подчиненный размещает в поле данных уникальный код, который говорит главному, какая именно ошибка произошла, или причину ошибки (См. раздел 6).

## 4.5 Содержание поля данных

4.5.1 Поле данных в сообщении от главного к подчиненному содержит дополнительную информацию, которая необходима подчиненному для выполнения указанной функции. Оно может содержать адреса регистров или выходов, их количество, счетчик передаваемых байтов данных.

Например, если главный запрашивает у подчиненного прочитать группу регистров (код функции 04), поле данных содержит адрес начального регистра и количество регистров.

## 4.6 Содержание поля контрольной суммы

4.6.1 В MODBUS сетях используются два метода контроля ошибок передачи. Содержание поля контрольной суммы зависит от выбранного способа передачи.

4.6.2 Когда используется ASCII-режим, поле контрольной суммы содержит два ASCII-символа. Контрольная сумма является результатом вычисления Longitudinal Redundancy Check (LRC), сделанного над содержимым сообщения, исключая ":" и пару CRLF.

4.6.3 Когда используется RTU-режим, поле контрольной суммы содержит 16-ти битовую величину. Контрольная сумма является результатом вычисления Cyclical Redundancy Check (CRC), сделанного над содержимым сообщения.

CRC добавляется к сообщению последним полем младшим байтом вперед.

## 4.7 Формат передачи символов.

4.7.1 Передача символов идет младшим битом вперед, как показано на рисунках 3 – 6.

старт	1	2	3	4	5	6	7	паритет	стоп
-------	---	---	---	---	---	---	---	---------	------

Рисунок 3. ASCII фрейм с контролем четности

старт	1	2	3	4	5	6	7	стоп	стоп
-------	---	---	---	---	---	---	---	------	------

Рисунок 4. ASCII фрейм без контроля четности

старт	1	2	3	4	5	6	7	8	паритет	стоп
-------	---	---	---	---	---	---	---	---	---------	------

Рисунок 5. RTU фрейм с контролем четности

старт	1	2	3	4	5	6	7	8	стоп	стоп
-------	---	---	---	---	---	---	---	---	------	------

Рисунок 6. RTU фрейм без контроля четности

Примечание – Возможно использование одного стоп-бита без бита паритета - устанавливается при конфигурировании преобразователя.

## 4.8 Методы контроля ошибок.

4.8.1 Стандартная MODBUS сеть использует два метода контроля ошибок: контроль паритета (even/odd) и контрольная сумма. Обе эти проверки генерируются в головном устройстве. Подчиненное устройство проверяет каждый байт и все сообщение в процессе приема. Если подчиненный обнаружил ошибку передачи, то он не формирует ответ главному. В случае отсутствия ошибок приёма данных подчинённое устройство (преобразователь) начинает передачу не позднее 5 мс от момента завершения приёма данных от головного устройства.

### 4.8.2 Контроль паритета

Пользователь может конфигурировать устройства на проверку четного или нечетного паритета (even/odd).

Например, 8 бит RTU-режима содержат следующую информацию:

1100 0101

Общее количество единиц - 4. Если используется четный паритет, то бит паритета будет равен 0, и общее количество единиц будет по-прежнему четным числом. Если используется нечетный паритет, то бит паритета будет равен 1, тогда общее количество единиц вместе с битом паритета будет равно 5, т.е. нечетному числу.

### 4.8.3 Контрольная сумма LRC

4.8.3.1 Метод LRC проверяет содержание сообщения, исключая начальный символ ":" и пару CRLF.

LRC - это 1 байт. LRC вычисляется передающим устройством и добавляется в конец сообщения. Принимающее устройство вычисляет LRC в процессе приема сообщения и сравнивает его с контрольной суммой, принятой от главного. Если есть несовпадение, то имеет место ошибка.

#### 4.8.3.2 Алгоритм генерации LRC:

1) Сложить все байты сообщения, исключая стартовый символ ":" и конечные CRLF, складывая их так, чтобы перенос отбрасывался.

2) Отнять получившееся значение от числа FF(Hex) - это является первым дополнением.

3) Прибавить к получившемуся значению 1 - это второе дополнение.

#### 4.8.3.3 Размещение LRC в сообщении:

Когда 8-ми битное поле LRC (2 ASCII символа) передается в сообщении, то старший символ будет передан первым, а за ним - младший. Пример сообщения для значения LRC 61 hex(0110 0001) приведен на рисунке 7.

":"	Адрес	Функция	Счетчик байт	Байт	Байт	Байт	Байт	Ст. LRC	Мл. LRC	CR	LF
								6	1		

Рисунок 7

### 4.8.4 Контрольная сумма CRC

4.8.4.1 Контрольная сумма CRC состоит из двух байт. 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.

Счетчик контрольной суммы предварительно инициализируется числом FFFF hex. Только восемь бит данных используются для вычисления контрольной суммы CRC. Старт и стоп биты, бит паритета, если он используется, не учитываются в контрольной сумме.

Во время генерации CRC каждый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с текущим содержимым регистра контрольной суммы. Результат сдвигается в направлении младшего бита, с заполнением нулем старшего бита. Если младший бит равен 1, то производится ИСКЛЮЧАЮЩЕЕ ИЛИ содержимого регистра контрольной суммы и определенного числа. Если младший бит равен 0, то ИСКЛЮЧАЮЩЕЕ ИЛИ не делается.

Процесс сдвига повторяется восемь раз. После последнего (восьмого) сдвига, следующий байт складывается с текущей величиной регистра контрольной суммы, и процесс сдвига повторяется восемь раз как описано выше. Конечное содержание регистра и есть контрольная сумма CRC.

#### 4.8.4.2 Алгоритм генерации CRC:

- 1) 16-ти битный регистр загружается числом FFFF hex (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 A001 hex.
- 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.

#### 4.8.4.3 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 1241 hex приведен на рисунке 8.

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

Рисунок 8

## 5 Поддерживаемые преобразователем функции

5.1 Поддерживаемые преобразователем функции приведены в таблице 1.

Таблица 1

Код функции	Описание	Чтение/запись, максимальное количество регистров
04	Чтение из регистров	100/100 регистров

### 5.2 Чтение из регистров

5.2.1 Функция «Чтение из регистров» используется для считывания регистров преобразователя.

5.2.2 Запрос содержит номер начального регистра и количество регистров для чтения.

Пример

Запрос для чтения регистра 0007h с подчиненного устройства 1

Запрос

Имя поля	Пример (Hex)
----------	-----------------

Адрес подчиненного	01
Функция	04
Начальный адрес ст.	00
Начальный адрес мл.	07
Кол-во регистров ст.	00
Кол-во регистров мл.	01
Контрольная сумма	--

5.2.3 Данные регистров в ответе передаются как два байта на регистр. Для каждого регистра первый байт содержит старшие биты, второй байт содержит младшие биты.

За одно обращение может считываться до 200 байт. Ответ дается, когда все данные укомплектованы.

Пример

Ответ на запрос, представленный в 5.2.2

Ответ

Имя поля	Пример (Hex)
----------	-----------------

Адрес подчиненного	01
Функция	04
Счетчик байт	02
Данные (регистр 0007h) ст.	00
Данные (регистр 0007h) мл.	2A
Контрольная сумма	--

## 6 Сообщения об ошибках

6.1 Список кодов ошибок приведен в таблице 2.

Таблица 2

Код	Название ошибки	Описание
01	Недопустимая функция	Запрашиваемая функция не поддерживается ведомым устройством
02	Недопустимый адрес данных	Полученный адрес данных не допустим для ведомого устройства
03	Недопустимое значение данных	Полученные значения данных не допустимы для ведомого устройства
04	Ошибка при выполнении запроса	

## 7 Регистры преобразователя

7.1 Адреса регистров, устанавливаемые по умолчанию, приведены в таблице 3.

Таблица 3

Имя регистра	Адрес	Чтение/Запись
Действующее значение напряжения фазы А ( $U_a$ )	0x0000	+/-
Действующее значение напряжения фазы В ( $U_b$ )	0x0001	+/-
Действующее значение напряжения фазы С ( $U_c$ )	0x0002	+/-
Действующее значение напряжения нулевой последовательности ( $U_o$ )	0x0003	+/-
Действующее значение силы тока фазы А ( $I_a$ )	0x0004	+/-
Действующее значение силы тока фазы В ( $I_b$ )	0x0005	+/-
Действующее значение силы тока фазы С ( $I_c$ )	0x0006	+/-
Действующее значение силы тока нулевой последовательности ( $I_o$ )	0x0007	+/-
Действующее значение междуфазного напряжения ( $U_{ab}$ )	0x0008	+/-
Действующее значение междуфазного напряжения ( $U_{bc}$ )	0x0009	+/-
Действующее значение междуфазного напряжения ( $U_{ca}$ )	0x000A	+/-
Активная мощность фазы А ( $P_a$ )	0x000B	+/-
Активная мощность фазы В ( $P_b$ )	0x000C	+/-
Активная мощность фазы С ( $P_c$ )	0x000D	+/-
Активная мощность трехфазной системы ( $P$ )	0x000E	+/-
Реактивная мощность фазы А ( $Q_a$ )	0x000F	+/-
Реактивная мощность фазы В ( $Q_b$ )	0x0010	+/-
Реактивная мощность фазы С ( $Q_c$ )	0x0011	+/-
Реактивная мощность трехфазной системы ( $Q$ )	0x0012	+/-
Полная мощность фазы А ( $S_a$ )	0x0013	+/-
Полная мощность фазы В ( $S_b$ )	0x0014	+/-
Полная мощность фазы С ( $S_c$ )	0x0015	+/-
Полная мощность трехфазной системы ( $S$ )	0x0016	+/-
Реактивная мощность фазы А ( $Q^{\prime}a$ )	0x0017	+/-
Реактивная мощность фазы В ( $Q^{\prime}b$ )	0x0018	+/-
Реактивная мощность фазы С ( $Q^{\prime}c$ )	0x0019	+/-
Частота ( $f$ )	0x001A	+/-

### Примечания

1 Адреса регистров могут быть изменены в адресном пространстве от 0x0000 до 0xFFFF с помощью программы «SetComplex» при конфигурировании преобразователя.

2 Программа «SetComplex» входит в комплект поставки преобразователя.