

**ПРЕОБРАЗОВАТЕЛЬ ИЗМЕРИТЕЛЬНЫЙ  
МНОГОФУНКЦИОНАЛЬНЫЙ  
АЕТ**

**Протокол информационного обмена**

**Часть 4. MODBUS-TCP/IP**

**47113964.505100.054-03 90 03-4**

## 1 Основные положения

1.1 Настоящий документ распространяется на преобразователи измерительные многофункциональные АЕТ100, АЕТ200, АЕТ300, АЕТ400 (далее – преобразователь), имеющие интерфейс Ethernet.

Преобразователь обеспечивает измерение параметров трехпроводных и четырехпроводных электрических сетей переменного тока и выдачу результатов измерения в сеть Ethernet по протоколу информационного обмена MODBUS TCP/IP.

1.2 В тексте используются следующие сокращения:

ADU - Application Data Unit (Элемент Данных Прикладного уровня)

IP - Internet Protocol

MBAP - MODBUS Application Protocol (Протокол MODBUS Прикладного уровня)

OSI - Open System Interconnection (взаимодействия открытых систем)

PDU - Protocol Data Unit (Элемент Данных Протокола)

TCP - Transport Control Protocol

1.3 Протокол MODBUS TCP/IP обеспечивает обмен между устройствами в сети Ethernet, используя модель Клиент – Сервер со стеком протоколов TCP/IP, где преобразователь выступает в роли MODBUS Server.

Модель Клиент – Сервер основывается на четырех типах сообщений:

MODBUS Request –запрос данных с сервера;

MODBUS Confirmation – сообщение, подтверждающее прием клиентом ответа от сервера;

MODBUS Indication – сообщение, подтверждающее, что запрос клиента принят сервером;

MODBUS Response – данные сервера.

Модель Клиент – Сервер изображена на рисунке 1.

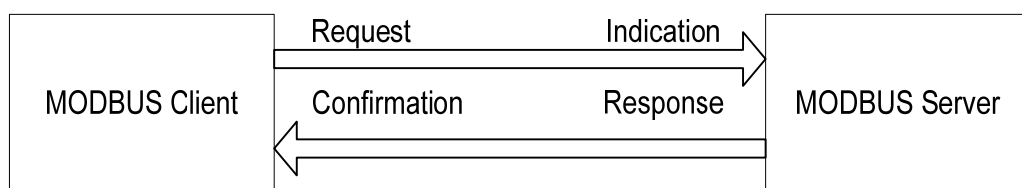


Рисунок 1

## 2 Данные прикладного уровня MODBUS TCP/IP (ADU)

2.1 Структура пакета MODBUS TCP/IP ADU представлена на рисунке 2.

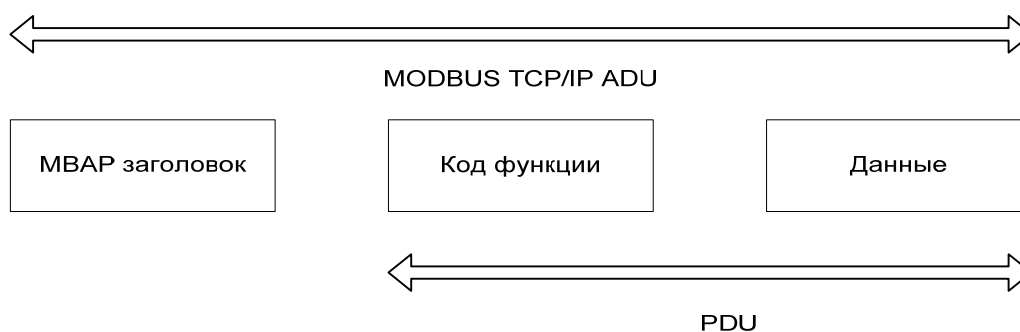


Рисунок 2

### 2.2 МВАР заголовок

МВАР заголовок предназначен для определения в стеке протоколов TCP/IP протокола MODBUS TCP/IP.

Назначение полей МВАР заголовка приведено в таблице 1.

Таблица 1

Поле	Длина, байт	Описание	Клиент	Сервер
Transaction Identifier (Идентификатор Транзакции)	2	Идентификация MODBUS запроса/ответа	Определяется клиентом	Копируется из полученного запроса
Protocol Identifier (Идентификатор Протокола)	2	0 = протокол MODBUS	Определяется клиентом	Копируется из полученного запроса
Length (Длина)	2	Количество последующих байт	Определяется клиентом (запрос)	Определяется сервером (ответ)
Unit Identifier (Идентификатор устройства)	1	Идентификатор устройства, подключенного к последовательной или другой линии данных	Определяется клиентом	Копируется из полученного запроса

#### Примечания

- 1 Transaction Identifier – порядковый номер запроса Клиента;
- 2 Protocol Identifier – используется для идентификации MODBUS протокола в разных сетях;
- 3 Length – счетчик байт передаваемого пакета начиная с поля Unit Identifier;
- 4 Unit Identifier – идентификатор запрашиваемого устройства, используемый в последовательном канале MODBUS. Данное поле применяется для доступа к устройствам, подключенным через шлюз Ethernet/последовательный канал MODBUS. В преобразователях это поле соответствует адресу устройства.

### 2.3 Код функции

В поле «Код функции» передается код запрашиваемой функции, соответствующий кодам функций согласно протоколу MODBUS-RTU. Описание протокола приведено в приложении А.

### 2.4 Данные

Поле данных формируется согласно протоколу MODBUS-RTU.

### 2.5 Элемент данных протокола (PDU)

Поле «Код функции» и поле «Данные» объединены в элемент данных протокола общий для MODBUS TCP/IP и MODBUS-RTU, MODBUS-ASCII.

МВАР заголовок совместно с PDU образует ADU передаваемый по стеку протоколов TCP/IP.

## 3 Структура стека протокола MODBUS TCP/IP в сети Ethernet

3.1 Структура стека протокола MODBUS TCP/IP в сети Ethernet в соответствии с моделью (OSI) приведена в таблице 2.

Таблица 2

Уровень	Реализация
Прикладной	ADU
Представления	-
Сеансовый	-
Транспортный	TCP (порт 502)
Сетевой	IP
Канальный	802.3 (Ethernet)
Физический	Экранированная и неэкранированная витая пара

## Приложение А

### Протокол информационного обмена MODBUS-RTU

#### А.1 Режим RTU

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). При работе в стеке протокола MODBUS TCP/IP контрольная сумма не используется.

#### А.2 Содержание MODBUS сообщения

А.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 А.1.

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 А.1

#### А.2.2 Содержание адресного поля

Адресное поле фрейма содержит 8 бит. 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.

### А.2.3 Содержание поля функции

Поле функции фрейма содержит 8 бит. 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. п. А.3.4).

### А.2.4 Содержание поля данных

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.

### А.2.5 Содержание поля контрольной суммы

А.2.5.1 В MODBUS сетях, работающих в RTU – режиме, поле контрольной суммы содержит 16-ти битовую величину. Контрольная сумма является результатом вычисления Cyclic Redundancy Check (CRC), сделанного над содержимым сообщения.

CRC добавляется к сообщению последним полем младшим байтом вперед.

А.2.5.2 При работе в стеке протокола MODBUS TCP/IP контрольная сумма не используется.

### А.2.6 Формат передачи символов

А.2.6.1 Передача символов идет младшим битом вперед, как показано на рисунках А.2, А.3.

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

Рисунок А.2. RTU фрейм с контролем четности

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

Рисунок А.3. RTU фрейм без контроля четности

## А.2.7 Методы контроля ошибок

А.2.7.1 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 15 мс от момента завершения приёма данных от головного устройства.

### А.2.7.2 Контроль паритета

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

Например, 8 бит RTU-режима содержат следующую информацию: 1100 0101.

В данном примере 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.

### А.2.7.3 Контрольная сумма CRC

А.2.7.3.1 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.

#### А.2.7.3.2 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.

#### А.2.7.3.3 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 А.4.

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

Рисунок А.4

### А.3 Поддерживаемые преобразователем функции

А.3.1 Поддерживаемые преобразователем функции приведены в таблице А.1.

Таблица А.1

Код функции	Описание	Чтение/запись, максимальное количество регистров
04	Чтение из регистров	100/100 регистров
16	Запись данных в последовательность регистров	100/100 регистров

#### А.3.2 Чтение из регистров

А.3.2.1 Функция «Чтение из регистров» используется для считывания регистров преобразователя.

А.3.2.2 Запрос содержит номер начального регистра и количество регистров для чтения.

Пример – Запрос для чтения регистра 0007h с подчиненного устройства 1

Имя поля	Значение (Hex)
Адрес подчиненного	01
Функция	04
Начальный адрес (старший байт)	00
Начальный адрес (младший байт)	07
Кол-во регистров (старший байт)	00
Кол-во регистров (младший байт)	01
Контрольная сумма	--

А.3.2.3 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 А.3.2.2

Имя поля	Значение (Hex)
Адрес подчиненного	01
Функция	04
Счетчик байт	02
Данные (регистр 0007h) (старший байт)	00
Данные (регистр 0007h) (младший байт)	2A
Контрольная сумма	--

### А.3.3 Запись данных в последовательность регистров

А.3.3.1 Функция «Запись данных в последовательность регистров» используется для записи данных в регистры преобразователя.

А.3.3.2 Запрос содержит номер начального регистра, количество регистров, счетчик байт и сами данные для записи.

Пример – Запись текущего времени (синхронизация часов преобразователя)

Имя поля	Значение (Hex)	
Адрес подчиненного	01	
Функция	10	
Начальный адрес (старший байт)	00	
Начальный адрес (младший байт)	1E	
Количество регистров (старший байт)	00	
Количество регистров мл.	04	
Счетчик байт	08	
Данные (старший байт)	06	(миллисекунды)
Данные (младший байт)	1A	
Данные (старший байт)	08	(часы)
Данные (младший байт)	27	(минуты)
Данные (старший байт)	0A	(месяц)
Данные (младший байт)	BA	(дни недели : дни месяца)
Данные (старший байт)	00	
Данные (младший байт)	0A	(год)
Контрольная сумма	--	

А.3.3.3 Нормальный ответ содержит адрес подчиненного, код функции, начальный адрес и количество регистров.

Пример – Ответ на запрос, представленный в А.3.3.2

Имя поля	Значение (Hex)
Адрес подчиненного	01
Функция	10
Начальный адрес (старший байт)	00
Начальный адрес мл.	1E
Количество регистров (старший байт)	00
Количество регистров мл.	04
Контрольная сумма	--

## А.3.4 Сообщения об ошибках

А.3.4.1 Список кодов ошибок приведен в таблице А.2.

Таблица А.2

Код	Название ошибки	Описание
1	Недопустимая функция	Запрашиваемая функция не поддерживается ведомым устройством
2	Недопустимый адрес данных	Полученный адрес данных не допустим для ведомого устройства
3	Недопустимое значение данных	Полученные значения данных не допустимы для ведомого устройства
4	Ошибка при выполнении запроса	

#### А.4 Регистры преобразователя

А.4.1 Адреса регистров, устанавливаемые по умолчанию, приведены в таблице А.3.

Таблица А.3

Имя регистра	Адрес	Чтение/Запись	Примечание
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 ( $Q_{fa}$ )	0x0017	+/-	
Реактивная мощность фазы В ( $Q_{fb}$ )	0x0018	+/-	
Реактивная мощность фазы С ( $Q_{fc}$ )	0x0019	+/-	
Частота ( $f$ )	0x001A	+/-	
Среднее значение фазных напряжений ( $U_{ф. ср}$ )	0x001B	+/-	
Среднее значение силы фазных токов ( $I_{ср}$ )	0x001C	+/-	
Среднее значение междуфазных напряжений ( $U_{ср}$ )	0x001D	+/-	
Коэффициент мощности фазы А ( $PF_a$ )	0x001E	+/-	
Коэффициент мощности фазы В ( $PF_b$ )	0x001F	+/-	
Коэффициент мощности фазы С ( $PF_c$ )	0x0020	+/-	
Коэффициент мощности трехфазной системы ( $PF$ )	0x0021	+/-	
Коэффициент мощности фазы А по первой гармонике ( $DPF_a$ )	0x0022	+/-	
Коэффициент мощности фазы В по первой гармонике ( $DPF_b$ )	0x0023	+/-	
Коэффициент мощности фазы С по первой гармонике ( $DPF_c$ )	0x0024	+/-	
Коэффициент мощности трехфазной системы по первой гармонике ( $DPF$ )	0x0025	+/-	
Время ( $t_{ms}$ )	0x0026	+/+	Опция RTC
Время ( $t_{hm}$ )	0x0027	+/+	
Время ( $t_{md}$ )	0x0028	+/+	
Время ( $t_y$ )	0x0029	+/+	

## Примечания

1 Адреса регистров могут быть изменены в адресном пространстве от 0x0000 до 0xFFFF с помощью программы для конфигурирования преобразователя.

2 Программа конфигурирования входит в комплект поставки преобразователя.

3 Для времени применены следующие обозначения:

tms – миллисекунды;

thm – старший байт - часы; младший байт - минуты;

tmd – старший байт - месяц; младший байт - дни недели 1...7 (3 старших бита); дни месяца 1...31 (5 младших бит);

ty – старший байт - всегда 0; младший байт – год.

По умолчанию в преобразователе установлено время по Гринвичу (GMT).

4 Регистры времени tmd и ty могут быть объединены в один регистр ty с помощью программы для конфигурирования преобразователя.

Для этого размер регистра ty необходимо установить равным трем байтам:

- старший байт – год;

- второй байт – месяц;

- младший байт – дни недели 1...7 (3 старших бита); дни месяца 1...31 (5 младших бит).

5 Опция RTC – наличие в преобразователе часов реального времени (Real Time Clock).