# Single-phase Electronic Meter

# SX1-A31N

# AMR Protocol Specifications

SPEC. NO. : **MED-A0028A**

| | |
|---|---|
| Date : | February 8, 2012 |
| Changed : | Rev.A, November 1st, 2012 |

| | |
|---|---|
| Drawn : | K. Nuttapol |
| Checked : | |
| Approved : | |

**MITSUBISHI ELECTRIC AUTOMATION (THAILAND) CO., LTD**

## SX1 PROTOCOL SPECIFICATION

The communications messages are consist of two protocol. First is RS-485 protocol, its cover whole of message and the second is inner protocol we named MEATH protocol. Protocol layers are show in figure 1.1-1 and specifications of each are describe in following sections.

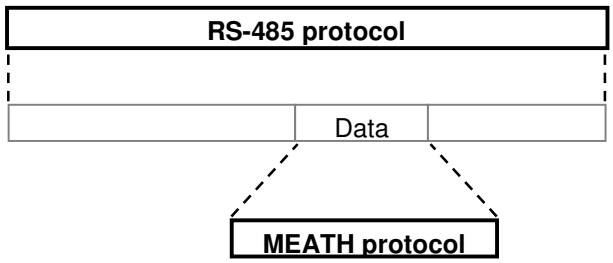



**Figure 1.1-1** Protocol stack layer

### 1. RS-485 Protocol

#### 1.1 Specifications

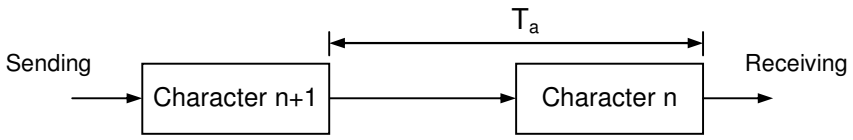
**Table 1.1-1** RS-485 communication and RS-485 protocol specifications

| | |
|---|---|
| Type of transmission | Asynchronous serial bit, Half duplex |
| Physical interface | RS – 485 (2wires) line driver. |
| Transmission speed | 19,200 bps. |
| Protocol standard | RS-485 Protocol |
| Technique | Single Master / Multi Slaves |
| Packet size | 51 bytes (Fixed) |
| Error detection | **CCITT CRC16 (cyclical redundancy check)**<br>Polynomial = 0x1021<br>Non-reflect algorithm (MSB first)<br>Initial value – 0xFFFF<br>**Note :** Testing value = "123456789"<br>CRC value = 0x29B1<br>See annex A for more details |
| Data format | **Binary**<br>LSB first<br>ST \| B0 \| B1 \| B2 \| B3 \| B4 \| B5 \| B6 \| B7 \| SP<br>ST : Start bit ( 1bit / logic 0)<br>SP : Stop bit ( 1bit / logic 1)<br>B0 – B7 : Binary data (8bits) |

## 1.2 Communication Timing

### 1.2.1 Time between each character

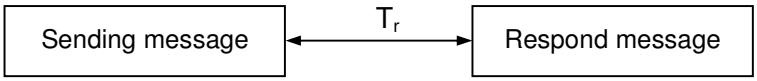
Every byte within a packet must precede the previous byte by take the time less than 1,500 ms. If time is more than, all of data will be flush out from the receive buffer.




$T_a$ < 1,500ms.

### 1.2.2 Time between each packet (Response time)

The response time from a master request to a slave response or in case of sent data in multiple packet. Any new packet of data will be wait 200 ≤ $t_r$ ≤ 1,500ms before send to receiver.




200ms. ≤ $T_r$ ≤ 1,500ms.

S/W target for response waiting time = 210 ms.

S/W target for response timeout = 1,500 ms.

S/W target for inter character timeout = 1,400 ms.

## 1.3 Protocol Structure

RS-485 message structure for every message type (Send message, Receive message, and Receive connect message) are the same, this show in figure 1.3-1.

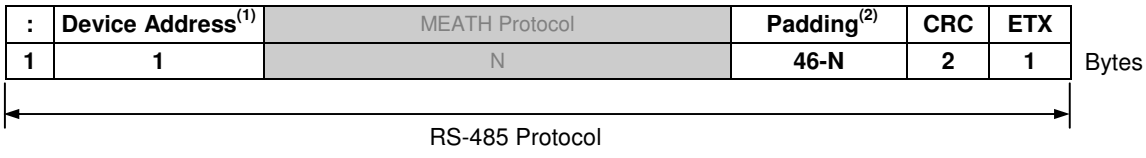



**Figure 1.3-1** RS-485 message structure

**(1)Device address (RS-485 address)**

Data size : 1 byte
Data format : Binary
Default device address : See annex B

**(2)Padding**

Data size : 46-N byte
Data format : Fill in with **#** ASCII

**Remark** Special character ASCII code in communication message

**:** = 0x3A

# = 0x23

ETX = 0x03

### 1.4 Protocol Analyze

For get data from received message, first analyze RS-485 protocol (outer protocol layer) to get MEATH protocol layer (inner protocol layer), and second analyze the MEATH protocol for get the target data. MEATH protocol will describe on section 2.

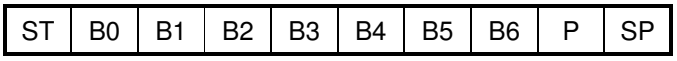
### 1.5 Error Detection

When CRC is incorrect, the meter will reject the command and will not reply to the master.

## 2. MEATH Protocol Specifications

### 2.1 Specifications

**Table 2.1-1** MEATH protocol specifications

| | |
|---|---|
| Protocol standard | MEATH Protocol (Custom) |
| Error detection | **Block Check Character (BCC)**<br>See annex A for more detail |
| Data format | **ASCII**<br>LSB first<br>  <br>ST : Start bit (1 bit/ logic 0)<br>SP : Stop bit (1 bit/ logic 1)<br>P : Parity bit (1 bit/ even)<br>B0 – B6: Character data (7 bit) |

## 2.2 Communication Sequence

As figure 2.2-1 sequence of message is starts from connect message to establish the communication. Then will read target data (such as kWh energy, line voltage and current) and when all target data reading finished program will send disconnect command to terminate the communication.

**Sequence of message**

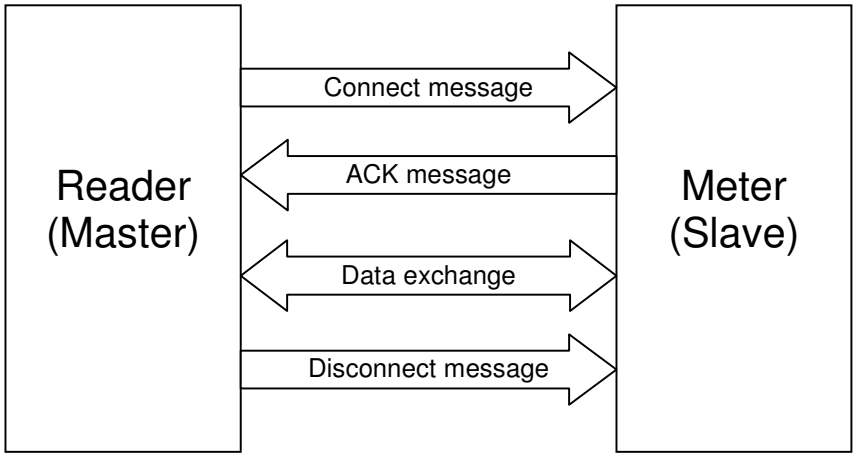



**Figure 2.2-1** Message sequence

**Sequence of data exchange**

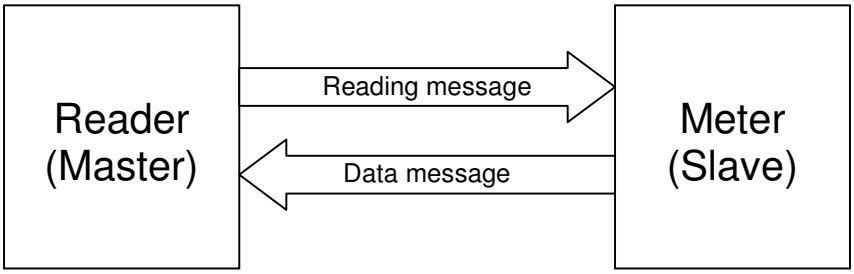



**Figure 2.2-2** Data exchange sequence

The sequence of data exchange will start from the reader sends reading command message to the meter. After meter receives the command message it will send the data message to the reader. For reading command meter will send data message back one by one for each command.

### 2.3 Protocol Structure

MEATH protocol will start after “Device address” (“Device address” is on RS-485 protocol layer) and finish at “BCC”, this show in figure 2.3-1.

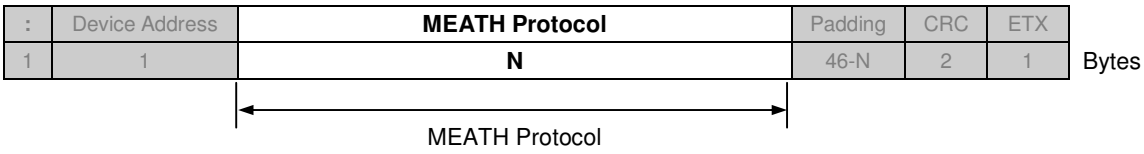



**Figure 2.3-1** MEATH protocol structure

### 2.4 MEATH protocol message

#### 2.4.1 Connect / Disconnect Message

##### 2.4.1.1 Connect Message

**Send** : Connect message

| SOH | P | 1 | STX | ( | Data | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|

Data size : 20 bytes

Data (Binary): D2 53 B4 B8 35 D4 D7 CF D7 C9 D2 C5 53 50 D2 CF CA C5 C3 D4

**Receive** : ACK message

| ACK |
|---|

Data size : 1 byte

Data (Binary) : 06

##### 2.4.1.2 Disconnect message

**Send** : Disconnect message

| SOH | B | 0 | ETX | BCC |
|---|---|---|---|---|

### 2.4.2 Meter Information

#### 2.4.2.1 Meter ID* reading

**Send** : Reading message

| SOH | R | 2 | STX | 0 | 0 | ( | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|---|

**Receive** : Data message

| STX | 0 | 0 | ( | Data | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|

Data size : 7 bytes

Data format : ASCII

Unit : -

*See annex B

### 2.4.3 Energy Value

#### 2.4.3.1 kWh energy reading

**Send** : Reading message

| SOH | R | 2 | STX | D | 7 | ( | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|---|

**Receive** : Data message

| STX | D | 7 | ( | Data | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|

Data size : 9 bytes

Data format : ASCII

Unit : Wh

### 2.4.4 PQM Value

#### 2.4.4.1 RMS line voltage reading

**Send** : Reading message

| SOH | R | 2 | STX | D | 0 | ( | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|---|

**Receive** : Data message

| STX | D | 0 | ( | Data | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|

Data size : 5 bytes

Data format : ASCII

Unit : 10 mV

#### 2.4.4.2 RMS current reading

**Send** : Reading message

| SOH | R | 2 | STX | D | 2 | ( | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|---|

**Receive** : Data message

| STX | D | 2 | ( | Data | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|

Data size : 5 bytes

Data format : ASCII

Unit : 10 mA

**See annex C for examples of data.**

## Annex A
## BCC and CRC calculation

### 1. Block check character (BCC) calculation

**1) Rule for generating the longitudinal parity block check character**

1.1) Block check character

1.1.1 The block check character shall be composed of 7 bits plus an even parity bit

1.1.2 Each of the first 7 bits of the block check character shall be the modulo 2 binary sum of every element in the same bit 0 to bit 6 column of the successive character of the transmitted block.

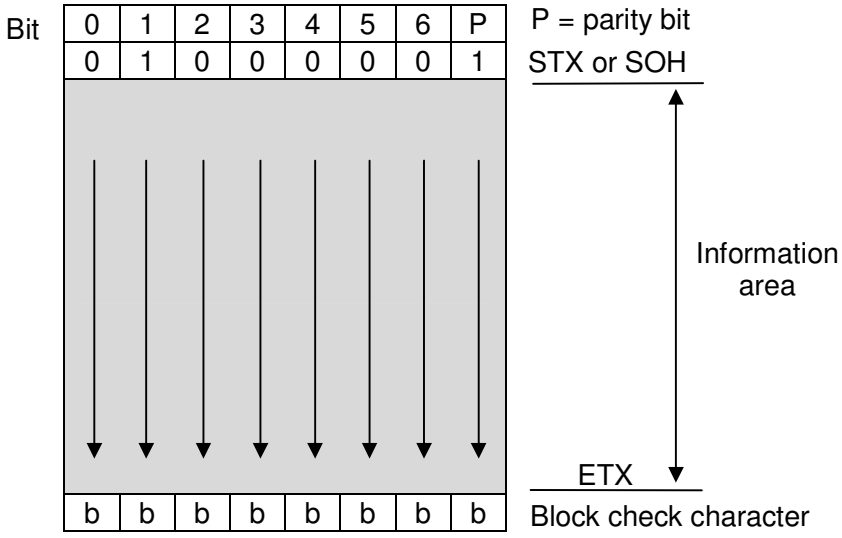
1.1.3 The longitudinal parity of each column of the block, including the block check character, shall be even.

1.2) Summation

1.2.1 The summation to obtain the block check character shall be started by the first appearance of either SOH (Start of Heading) or STX (Start of Text).

1.2.2 The starting character shall not be included in the summation.

1.2.3 If an STX character appears after the summation has been started by SOH, the STX character shall be included in the summation as if it were a text character.



**2) Example of block check character calculation**

Assume that the read command message has the stream of character like this;

HHU →

| SOH | R | 2 | STX | D | 7 | ( | ) | ETX | BCC |
|---|---|---|---|---|---|---|---|---|---|

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | P | P = parity bit | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | SOH | 0x81 |
| | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | R | 0xD2 |
| | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 2 | 0xB2 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | STX | 0x82 |
| | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | D | 0x44 |
| | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 7 | 0xB7 |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | ( | 0x28 |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | ) | 0xA9 |
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ETX | 0x03 |
| | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | BCC | 0x93 |

From the figure, the highlight area is the field of character for generating block check character. Each bit of block check character is the longitudinal even parity of each column of the block, including the block check character, shall be even. So from the example the block check character from the calculation will be (93h).

**2. Cyclical Redundancy Check (CRC-16) calculation**

The Cyclical Redundancy Check (CRC-16) field is two bytes, containing a 16–bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, that's means the communication error are occurs. Program will flush message and do not operate any operation and any response message.

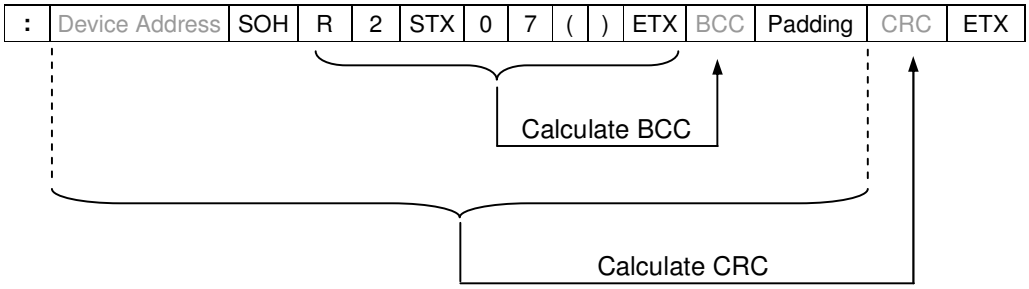
**Placing the CRC into the Message**

When the 16–bit CRC (two 8–bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte. For example, if the CRC value is 1241 hex (0001 0010 0100 0001) the CRC value will be place into message as follow.

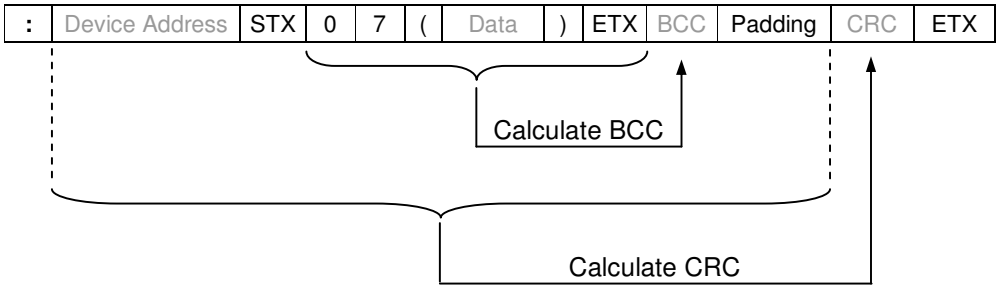
| : | Device Address | Data Field | Reserve | $CRC_{Lo}$ | $CRC_{Hi}$ | ETX |
|---|---|---|---|---|---|---|
| | | | | 41 | 12 | |

BCC and CRC calculation field.

Send




Receive



## Annex B

## RS-485 Address

The meter RS-485 address assign as follows

- If 3rd digits from the last digit of ID no. that showing on nameplate is even number, the meter RS-485 address will be ***"0xx"*** by ***"xx"*** is the last 2 digits of ID no. If the last 2 digits of ID no. are ***"00"***, meter RS-485 address will be ***"200"***.

- If 3rd digits from the last digit of ID no. that showing on nameplate is odd number, the meter RS-485 address will be ***"1xx"*** by ***"xx"*** is the last 2 digits of ID no.

**Example:**

| ID no. | 3rd digit from last digit of ID no. | RS-485 address |
|---|---|---|
| 0275***348*** | ***Odd*** | ***148*** |
| 0275***448*** | ***Even*** | ***48*** |
| 0275***548*** | ***Odd*** | ***148*** |
| 0276***000*** | ***Even*** | ***200*** |
| 0276***100*** | ***Odd*** | ***100*** |
| 0700***030*** | ***Even*** | ***30*** |

## Annex C
## Example Message

### 1. Example of Connect message

**Send:** Connect message

3A 23 81 50 B1 82 28 D2 53 B4 B8 35 D4 D7 CF D7 C9 D2 C5 53 50 D2
CF CA C5 C3 D4 A9 03 9A 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 04 50 03

**Receive:** ACK message

3A 23 06 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 CC D1 03

### 2. Example of Meter ID reading

**Send :** Reading message

3A 23 81 D2 B2 82 30 30 28 A9 03 60 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 96 DC 03

**Receive:** Data message

3A 23 82 30 30 28 ***B7 39 30 30 B2 33 35*** A9 03 B8 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 0A F2 03

**Reading result:** Meter ID is ***7900235***

### 3. Example of kWh energy reading

**Send:** Reading message

```
3A 23 81 D2 B2 82 44 B7 28 A9 03 93 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 0B BD 03
```

**Receive:** Data message

```
3A 23 82 44 B7 28 30 30 30 30 B2 39 33 B4 39 A9 03 44 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 A7 CC 03
```

**Reading result:** Energy = ***000029349*** Wh

= **29.349 kWh**

### 4. Example of RMS line voltage reading

**Send:** Reading message

```
3A 23 81 D2 B2 82 44 30 28 A9 03 14 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 5F 5B 03
```

**Receive:** Data message

```
3A 23 82 44 30 28 B2 B1 B8 B2 B2 A9 03 4D 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 B7 19 03
```

**Reading result:** RMS line voltage = ***21822*** × 10mV

= **218.22 V**

**5. Example of RMS current reading**

**Send:** Reading message

3A 23 81 D2 B2 82 44 B2 28 A9 03 96 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 2D D3 03

**Receive:** Data message

3A 23 82 44 B2 28 ***30 30 30 B8 33*** A9 03 CF 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 B3 1B 03

**Reading result:** RMS current = ***00083*** × 10mA

= **0.83 A**