**Revised Renogy logo 22417**

**Rover 20A/40A Charge Controller—MODBUS Protocol**

**1. MODBUS RTU Communication Protocol Format and Command Parsing:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Start character | Address field | Function code | Data | Error check | End character |

**1.1 Format:**

**1.2 Descriptions:**

(Data below suffixed with an "H" are hexadecimal, and the others are decimal)

1. Start character: >10ms
2. Address field: one byte, range: 01H **to** F7H (decimal 1 **to** 247). 00H is a broadcast address to which all slaves respond but do not return commands
3. Function code: 1 byte

|  |  |  |  |
| --- | --- | --- | --- |
| Command name | Accessed data type | Function code | Error code |
| Read a Single or Multiple Word register(s) | 2 bytes | 03H | 83H |
| Write a Single Word Register | 2 bytes | 06H | 86H |
| Write N Word Registers in a Row | 2 bytes | 10H | 90H |
| Reset to Factory Defaults | No accessed data | 78H | F8H |
| Clear History | No accessed data | 79H | F9H |

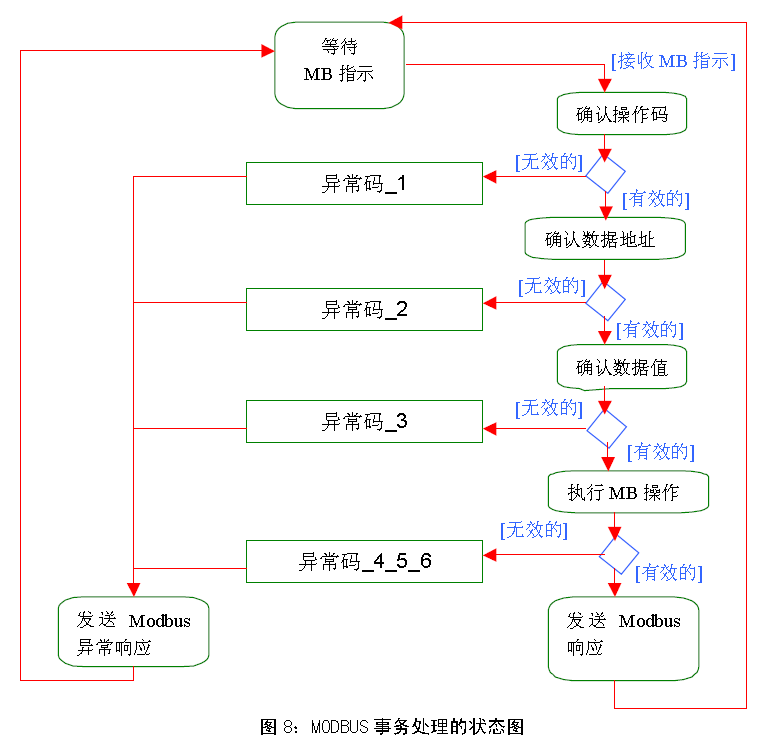
1. Data: N bytes
2. Error check: 2 bytes. It's the CRC checksum of the device address, function code and each byte of the data.
3. End character: >10ms

Note:

1) The data address and the data itself are of 2 bytes, with the high byte sent first and then the low byte; for CRC, the low byte is sent first, and the high byte is sent next.

2) The error code is the error response function code returned by the client when there is some error in the frame data sent by the server; error cod = function code | 80H

**1.3 Process flow chart**



Awaits MB instruction

[Receives MB instruction]

Exception code\_1

Confirms data address

Fig. 8 Modbus process flow chart

Executes MB operation

Confirms operation code

Confirms data value

Exception code\_1

Sends Modbus exception response

Exception code\_2

Exception code\_3

Exception code\_4\_5\_6

Sends Modbus response

[Invalid]

[Valid]

[Invalid]

[Invalid]

[Invalid]

[Valid]

[Valid]

[Valid]

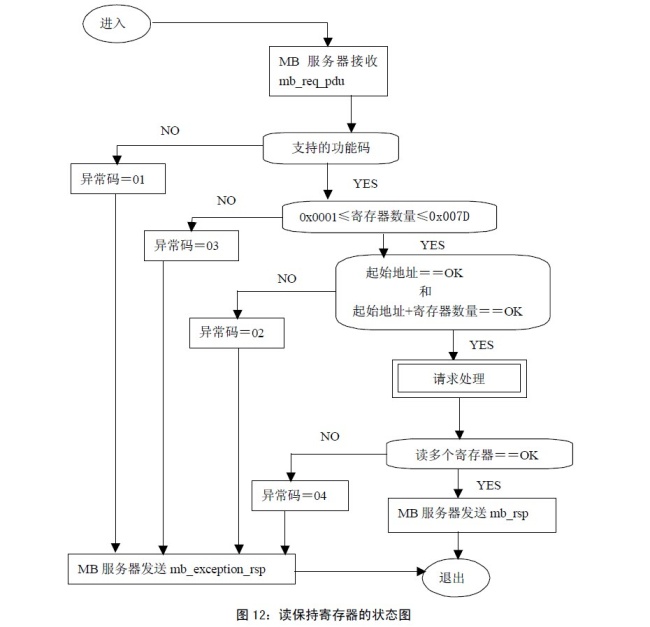
Exception code descriptions

1. 01H — Function code not supported
2. 02H — PDU start address is not correct or PDU start address + data length
3. 03H — Data length in reading or writing register is too large
4. 04H — Client fails to read or write register — not used
5. 05H — Data check code sent by server is not correct — not used

Note: the server's reception of an exception code returned by the controller indicates that the controller had received the command sent by the server, but the command was erroneous, thus the server should resend the command.

1.3.1 Flow chart of reading register

Entry



MB server receives

mb\_req\_pdu

Supported function code

Exception code = 01

Exception code = 02

Exception code = 03

Exception code = 04

Exit

Requests processing

Reads multiple registers == OK

Start address == OK

and

Start address + No. of registers == OK

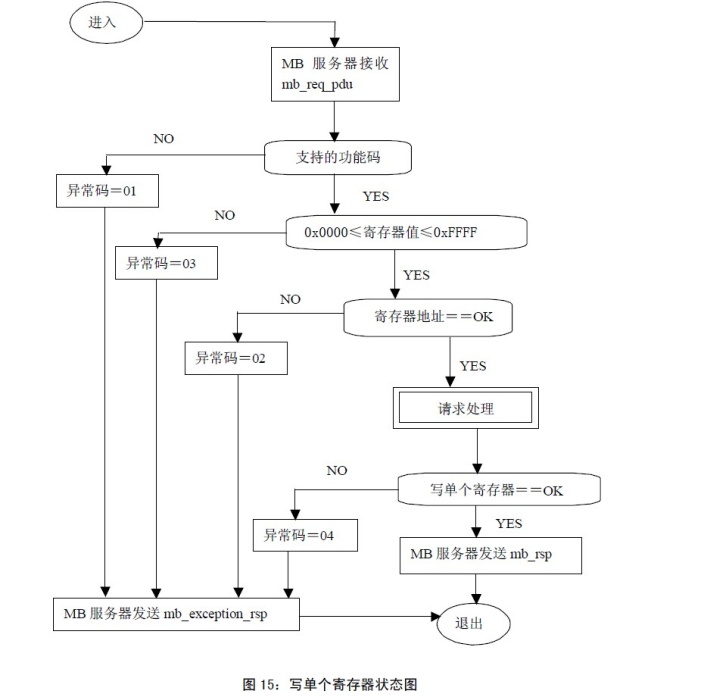
MB server sends mb\_rsp

MB server sends mb\_exception\_rsp

0x0001 ≤ No. of registers ≤ 0x007D

**Fig. 12 Flow chart of reading holding register**

1.3.2 Flow chart of writing a single register



Entry

MB server receives

mb\_req\_pdu

Supported function code

Exception code = 01

Exception code = 03

Exception code = 02

Exception code = 04

MB server sends mb\_rsp

Writes a single register == OK

MB server sends mb\_exception\_rsp

Exit

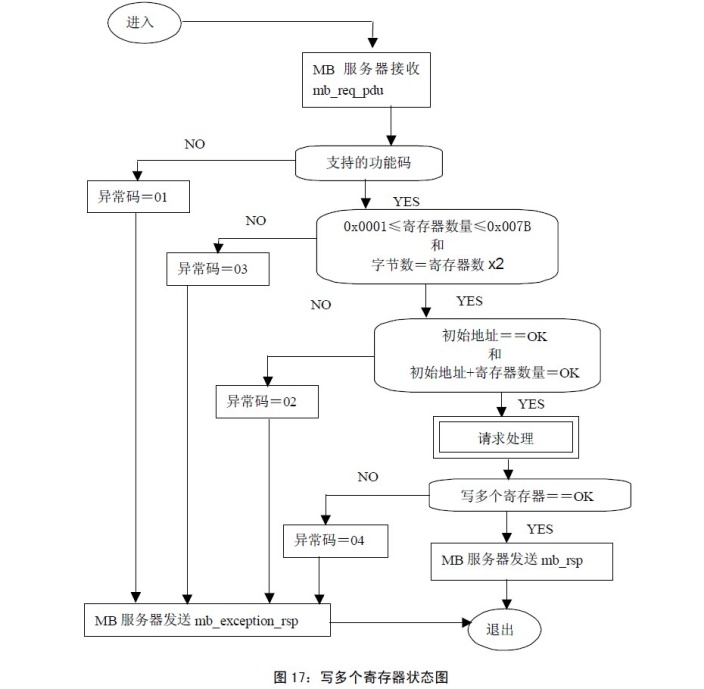
**Fig. 15 Flow chart of writing a single register**

Requests processing

Register address == OK

0x0000 ≤ Register value ≤ 0xFFFF

1.3.3 Flow chart of writing N registers in a row



MB server sends mb\_exception\_rsp

Exit

MB server sends mb\_rsp

Exception code = 04

Writes multiple registers == OK

Requests processing

Exception code = 02

Exception code = 03

Exception code = 01

Supported function code

MB server receives

mb\_req\_pdu

Entry

Start address == OK

and

Start address + No. of registers == OK

0x0001 ≤ No. of registers ≤ 0x007B

and

No. of bytes = No. of registers × 2

**Fig. 17 Flow chart of writing multiple registers**

**1.4 Example:**

**1.4.1 Read register**

**Request:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 03H |
| Start address | WORD | 0000H **to** FFFFH |
| No. of read words | WORD | 0001H **to** 007DH |
| Check code | WORD | CRC checksum of all the above bytes |

**Normal response**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 03H |
| Data length | BYTE | 01H **to** FAH |
| Data content | WORD | Data read out (High byte sent first, low byte sent next) |
| ... | WORD | Data read out (High byte sent first, low byte sent next) |
| Check code | WORD | CRC checksum of all the above bytes |

**Exception response:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Error code | BYTE | 83H |
| Exception code | BYTE | N (N=1, 2, 3, 4) |
| Check code | WORD | CRC checksum of all the above bytes |

**1.4.2 Write a single register**

**Request**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 06H |
| Start address | WORD | 0000H **to** FFFFH |
| Write data in | WORD | 0000H **to** FFFFH |
| Check code | WORD | CRC checksum of all the above bytes |

**Normal response**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 06H |
| Start address | WORD | 0000H **to** FFFFH |
| Write data in | WORD | 0000H **to** FFFFH |
| Check code | WORD | CRC checksum of all the above bytes |

**Exception response:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Error code | BYTE | 86H |
| Exception code | BYTE | N (N=1, 2, 3, 4) |
| Check code | WORD | CRC checksum of all the above bytes |

**1.4.3 Write N registers in a row**

**Request:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 10H |
| Start address | WORD | 0000H **to** FFFFH |
| No. of written bytes | WORD | 0001H **to** 007DH |
| No. of written words | BYTE | One time of the No. of bytes |
| Data content | WORD | Data written in (High byte sent first, low byte sent next) |
| ... | WORD | Data written in (High byte sent first, low byte sent next) |
| Check code | WORD | CRC checksum of all the above bytes |

**Normal response**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 10H |
| Start address | WORD | 0000H **to** FFFFH |
| No. of written bytes | WORD | 0001H **to** 007DH |
| Check code | WORD | CRC checksum of all the above bytes |

**Exception response:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Error code | BYTE | 90H |
| Exception code | BYTE | N (N=1, 2, 3, 4) |
| Check code | WORD | CRC checksum of all the above bytes |

**1.4.4 Reset to factory defaults**

**Request:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 78H |
| Complementary data | WORD | 0000H |
| Complementary data | WORD | 0001H |
| Check code | WORD | CRC checksum of all the above bytes |

**Normal response**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 78H |
| Complementary data | WORD | 0000H |
| Complementary data | WORD | 0001H |
| Check code | WORD | CRC checksum of all the above bytes |

**Exception response:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Error code | BYTE | F8H |
| Exception code | BYTE | N (N=1, 2, 3, 4) |
| Check code | WORD | CRC checksum of all the above bytes |

**1.4.4 Clear history**

**Request:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 79H |
| Complementary data | WORD | 0000H |
| Complementary data | WORD | 0001H |
| Check code | WORD | CRC checksum of all the above bytes |

**Normal response**:

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Function code | BYTE | 79H |
| Complementary data | WORD | 0000H |
| Complementary data | WORD | 0001H |
| Check code | WORD | CRC checksum of all the above bytes |

**Exception response:**

|  |  |  |
| --- | --- | --- |
| **Description** | **No. of bytes** | **Command** |
| Device address | BYTE | 01H **to** F7H |
| Error code | BYTE | F9H |
| Exception code | BYTE | N (N=1, 2, 3, 4) |
| Check code | WORD | CRC checksum of all the above bytes |

**2. PDU Address Allocation Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Table 1: (Data below suffixed with an "H" are hexadecimal, and the others are decimal)** | | | | | | | | | | | | | |  | |  |
| **Description** | | | | **PDU address** | **Bytes** | | | **R/ W** | | **Description** | **Data (range)** | | **Meaning** | | **Unit** | | **Remarks** |
|  | |  | | 0000H  to  0009H | 20 | | | - | | Reserved |  | |  | | - | |  |
| **System Information** | |  | | 000AH | 2 | | | R | | 8 higher bits:  max. voltage supported by the system | 0CH (decimal 12) | | 12V | | - | |  |
| 18H (decimal 24) | | 24V | |
| 24H (decimal 36) | | 36V | |
| 30H (decimal 48) | | 48V | |
| 60H (decimal 96) | | 96V | |
| FFH (decimal 255) | | Automatic recognition of system voltage | |
| 8 lower bits:  rated charging current | 0AH (decimal 10) | | 10A | |
| 14H (decimal 20) | | 20A | |
| 1EH (decimal 30) | | 30A | |
| 2DH (decimal 45) | | 45A | |
| 3CH (decimal 60) | | 60A | |
| 000BH | 2 | | | R | | 8 higher bits:  rated discharging current | 0AH (decimal 10) | | 10A | |  | |  |
| 14H (decimal 20) | | 20A | |  | |  |
| 1EH (decimal 30) | | 30A | |  | |  |
| 2DH (decimal 45) | | 45A | |  | |  |
| 3CH (decimal 60) | | 60A | |  | |  |
| 8 lower bits:  product type | 00 (controller)  01 (inverter)  ... | |  | |  | |  |
| 000CH  to  0013H | 16 | | | R | | Product model |  | |  | | - | |  |
| 0014H 0015H | 4 | | | R | | Software version |  | |  | | - | |  |
| 0016H 0017H | 4 | | | R | | Hardware version |  | |  | | - | |  |
| 0018H 0019H | 4 | | | R | | Product serial number |  | |  | | - | |  |
| 001AH | 2 | | | R/W | | Controller, device address | 1 to 247 | |  | | - | | 8 lower bits |
| RAM information | | | | | | | | | | | | | | | |
|  | | 0100H | | 2 | R | | Battery capacity SOC | | 0 to 100 | | Current battery capacity value | % | |  | |
|  | |  | | 0101H | | 2 | R | | Battery voltage | |  | | Battery voltage \* 0.1 | V | |  | |
| **Dynamic controller information** | |  | | 0102H | | 2 | R | | Charging current (to battery) | |  | | Charging current \* 0.01 | A | |  | |
|  | | 0103H | | 2 | R | | Controller temperature | |  | | Actual temperature value  (b7: sign bit; b0-b6: temperature value) | °C | |  | |
|  | | Battery temperature | |  | |
|  | | 0104H | | 2 | R | | Street light (load) voltage | |  | | Street light voltage \* 0.1 | V | |  | |
|  | | 0105H | | 2 | R | | Street light (load) current | |  | | Street light current \* 0.01 | A | |  | |
|  | | 0106H | | 2 | R | | Street light (load) power | |  | | Actual value | W | |  | |
| **Solar panel information** | | 0107H | | 2 | R | | Solar panel voltage | |  | | Solar panel voltage \* 0.1 | V | |  | |
| 0108H | | 2 | R | | Solar panel current (to controller) | |  | | Solar panel current \* 0.01 | A | |  | |
| 0109H | | 2 | R | | Charging power | |  | | Actual value | W | |  | |
|  | | 010AH | | 2 | W | | Light On/ Off command | | 0 or 1 | | 1: to turn on street light, 0: to turn off street light | - | |  | |
|  | | 010BH | | 2 | R | | Battery's min. voltage of the current day | |  | | Battery's min. voltage of the current day \* 0.1 | V | |  | |
| 010CH | | 2 | R | | Battery's max. voltage of the current day | |  | | Battery's max. voltage of the current day \* 0.1 | V | |  | |
| 010DH | | 2 | R | | Max. charging current of the current day | |  | | Max. charging current of the current day \* 0.01 | A | |  | |
| 010EH | | 2 | R | | Max. Discharging current of the current day | |  | | Max. discharging current of the current day \* 0.01 | A | |  | |
| 010FH | | 2 | R | | Max. charging power of the current day | |  | | Actual value | W | |  | |
| 0110H | | 2 | R | | Max. discharging power of the current day | |  | | Actual value | W | |  | |
| 0111H | | 2 | R | | Charging amp-hrs of the current day | |  | | Actual value | AH | |  | |
| 0112H | | 2 | R | | Discharging amp-hrs of the current day | |  | | Actual value | AH | |  | |
| 0113H | | 2 | R | | Power generation of the current day | |  | | Actual value | kilowatt hour/ 10000 | |  | |
| 0114H | | 2 | R | | Power consumption of the day | |  | | Actual value | kilowatt hour/ 10000 | |  | |
| **Historical data information** | | 0115H | | 2 | R | | Total number of operating days | |  | |  | days | |  | |
| 0116H | | 2 | R | | Total number of battery over-discharges | |  | |  | - | |  | |
| 0117H | | 2 | R | | Total number of battery full-charges | |  | |  | - | |  | |
| 0118H 0119H | | 4 | R | | Total charging amp-hrs of the battery | |  | | Actual value | AH | |  | |
| 011AH 011BH | | 4 | R | | Total discharging amp-hrs of the battery | |  | | Actual value | AH | |  | |
| 011CH 011DH | | 4 | R | | Cumulative power generation | |  | | Actual value | kilowatt hour/ 10000 | |  | |
| 011EH 011FH | | 4 | R | | Cumulative power consumption | |  | | Actual value | kilowatt hour/ 10000 | |  | |
|  | | 0120H | | 2 | R | | Street light status | | 0 or 1 | 8 higher bits | b7: 0 indicates the street light is off,  1 indicates the street light is on | - | |  | |
|  | | Street light brightness | | 00 to 64H | b0 to b6: brightness value | % | |  | |
| Charging state | |  | 8 lower bits | 00H: charging deactivated | - | |  | |
| 01H: charging activated |
| 02H: mppt charging mode |
| 03H: equalizing charging mode |
| 04H: boost charging mode |
| 05H: floating charging mode |
| 06H: current limiting (overpower) |
| **Controller**  **fault**  **information** | | 0121H 0122H | | 4 | R | | Controller fault and warning information | |  |  | b31 reserved  b30: circuit, charge MOS short circuit  b29: anti-reverse MOS short |  | | E.g.:  A certain bit being 1 indicates some fault occurs to the corresponding item, while a certain bit being 0 indicates the corresponding item is free from faults. When all items function normally, the bits return to 00000000H. | |
| B28: solar panel reversely connected |
| B27: solar panel working point over-voltage |
| B26: solar panel counter-current |
| B25: photovoltaic input side over-voltage |
| B24: photovoltaic input side short circuit |
| B23: photovoltaic input overpower |
| B22: ambient temperature too high |
| B21: controller temperature too high |
| B20: load overpower  or load over-current |
| B19: load short circuit |
| B18: battery under-voltage warning  16 High bit |
| B17: battery over-voltage |
| B16: battery over-discharge |
| **B0**-B15 reserved |
|  | | EEPROM | | | | | | | | | | | | | | | |
|  | |  | E001H | | | 2 | R/W | | Dimming command | | 0000H to 0064H (decimal 0 to 100) | | To set street light brightness value | % | |  | |
| **Controller parameter settings** | | **Battery parameter settings** | E002H | | | 2 | R | | Nominal battery capacity | |  | |  | AH | |  | |
| E003H | | | 2 | R/W | | 8 higher bits: system voltage setting  8 lower bits: recognized voltage | |  | | 12: 12V;  24: 24V;  36: 36V;  48: 48V;  FF: automatic recognition  Others: automatic recognition | - | |  | |
| E004H | | | 2 | R/W | | Battery type | |  | | Open, sealed, gel, lithium,  self-customized | - | |  | |
| E005H | | | 2 | R/W | | Over-voltage threshold | | 70 to 170 | |  | V | | Setting range:  (7 to 17) V  E.g.:  when the over-voltage threshold needs to be set to 17.0 V and one decimal place is to be kept, first multiply the figure by 10, i.e. 17.0V \* 10 = 170V, then convert it to a hexadecimal value 00AAH, and next write the value into 0103H. | |
| E006H | | | 2 | R/W | | Charging voltage limit | | 70 to 170 | |  | V | |
| E007H | | | 2 | R/W | | Equalizing charging voltage | | 70 to 170 | |  | V | |  | |
| E008H | | | 2 | R/W | | Boost charging voltage/ overcharge voltage (lithium batteries) | | 70 to 170 | |  | V | |
| E009H | | | 2 | R/W | | Floating charging voltage/ overcharge recovery voltage (lithium batteries) | | 70 to 170 | |  | V | |
| E00AH | | | 2 | R/W | | Boost charging recovery voltage | | 70 to 170 | |  | V | |
| E00BH | | | 2 | R/W | | Over-discharge recovery voltage | | 70 to 170 | |  | V | |
| E00CH | | | 2 | R/W | | Under-voltage warning level | | 70 to 170 | |  | V | |
| E00DH | | | 2 | R/W | | Over-discharge voltage | | 70 to 170 | |  | V | |
| E00EH | | | 2 | R/W | | Discharging limit voltage | | 70 to 170 | |  | V | |  | |
| E00FH | | | 2 | R/W | | 8 higher bits: end-of-charge SOC  8 lower bits: end-of-discharge SOC | |  | |  | - | |  | |
| E010H | | | 2 | R/W | | Over-discharge time delay | | 0 to 120 | |  | S | |  | |
| E011H | | | 2 | R/W | | Equalizing charging time | | 0 to 300 | | Step length + 10 | Min | |  | |
| E012H | | | 2 | R/W | | Boost charging time | | 10 to 300 | | Step length + 10 | Min | |  | |
| E013H | | | 2 | R/W | | Equalizing charging interval | | 0 to 255 | | 0: closed, step length + 5 | day | |  | |
| E014H | | | 2 | R/W | | Temperature compensation factor | | 0 to 5 | | 0: not compensated, step length + 1 | mV/ °C/ 2V | |  | |
| **Load operating duration and power settings** | E015H | | | 2 | R/W | | 1st-stage operating duration | | 00H to 15H | |  | H | |  | |
| E016H | | | 2 | R/W | | 1st-stage operating power | | 0 to 100 | |  | % | |  | |
| E017H | | | 2 | R/W | | 2nd-stage operating duration | | 00H to 15H | |  | H | |  | |
| E018H | | | 2 | R/W | | 2nd-stage operating power | | 0 to 100 | |  | % | |  | |
| E019H | | | 2 | R/W | | 3rd-stage operating duration | | 00H to 15H | |  | H | |  | |
| E01AH | | | 2 | R/W | | 3rd-stage operating power | | 0 to 100 | |  | % | |  | |
| E01BH | | | 2 | R/W | | Morning on operating duration | | 00H to 15H | |  | H | |  | |
| E01CH | | | 2 | R/W | | Morning on operating power | | 0 to 100 | |  | % | |  | |
|  |  | | |  |  | |  | |  | |  |  | |  | |
| **Mode setting** | E01DH | | | 2 | R/W | | Load working modes | | 00H | | Sole light control, light control over on/ off of load | - | |  | |
| 01H | | Load is turned on by light control, and goes off after a time delay of 1 hour |
| 02H | | Load is turned on by light control, and goes off after a time delay of 2 hours |
| 03H | | Load is turned on by light control, and goes off after a time delay of 3 hours |
| 04H | | Load is turned on by light control, and goes off after a time delay of 4 hours |
| 05H | | Load is turned on by light control, and goes off after a time delay of 5 hours |
| 06H | | Load is turned on by light control, and goes off after a time delay of 6 hours |
| 07H | | Load is turned on by light control, and goes off after a time delay of 7 hours |
| 08H | | Load is turned on by light control, and goes off after a time delay of 8 hours |
| 09H | | Load is turned on by light control, and goes off after a time delay of 9 hours |
| 0AH (decimal 10) | | Load is turned on by light control, and goes off after a time delay of 10 hours |
| 0BH (decimal 11) | | Load is turned on by light control, and goes off after a time delay of 11 hours |
| 0CH (decimal 12) | | Load is turned on by light control, and goes off after a time delay of 12 hours |
| 0DH (decimal 13) | | Load is turned on by light control, and goes off after a time delay of 13 hours |
| 0EH (decimal 14) | | Load is turned on by light control, and goes off after a time delay of 14 hours |
| 0FH (decimal 15) | | Manual mode |
| 10H (decimal 16) | | Debugging mode |
| 11H (decimal 17) | | Normal on mode |
| **Light control setting** | E01EH | | | 2 | R/W | | Light control delay | | 0 to 60 | |  | Min | |  | |
| E01FH | | | 2 | R/W | | Light control voltage | | 1 to 40 | |  | V | |  | |
|  | E020H | | | 2 | R/W | | LED load current setting | | N | |  | 10mA | | (N \* 10) mA | |
|  | E021H | | | 2 | R/W | | Special power control | | 8 higher bits | | b3 to b7 not used | - | |  | |
|  | b2: 1—charging mode controlled by voltage  0—charging mode controlled by SOC |
|  | b1: 1—special power control function enabled   0—special power control function disabled |
|  | b0: 1—each night on function enabled   0—each night on function disabled |
|  | 8 lower bits | | b3 to b7 not used |
| b2: no charging below 0 °C (1: enabled, 0: disabled) |
|  |  | | b0 to b1: charging method (00: direct charging, 01: PWM charging) |
|  | | **MES**  **Load operating duration and power settings** | E022H | | | 2 | R/W | | Working hours determined by automatic sensing 1 | | 0 to 15 | | Step length + +1 | H | |  | |
| E023H | | | 2 | R/W | | Power with people sensed 1 | | 0 to 100 | | Step length + 10 | % | |  | |
| E024H | | | 2 | R/W | | Power with no people sensed 1 | | 0 to 100 | | Step length + 10 | % | |  | |
| E025H | | | 2 | R/W | | Working hours determined by automatic sensing 2 | | 0 to 15 | | Step length + +1 | H | |  | |
| E026H | | | 2 | R/W | | Power with people sensed 2 | | 0 to 100 | | Step length + 10 | % | |  | |
| E027H | | | 2 | R/W | | Power with no people sensed 2 | | 0 to 100 | | Step length + 10 | % | |  | |
| E028H | | | 2 | R/W | | Working hours determined by automatic sensing 3 | | 0 to 15 | | Step length + +1 | H | |  | |
| E029H | | | 2 | R/W | | Power with people sensed 3 | | 0 to 100 | | Step length + 10 | % | |  | |
| E02AH | | | 2 | R/W | | Power with no people sensed 3 | | 0 to 100 | | Step length + 10 | % | |  | |
| E02BH | | | 2 | R/W | | Sensing time delay | | 0 to 250 | | Step length + 10 | S | |  | |
| E02CH | | | 2 | R/W | | LED load current | | N | |  | 10mA | | (N \* 10) mA | |
| E02DH | | | 2 | R/W | | Special power control | | 8 higher bits | | b7 to b2: not used | - | |  | |
| b1: intelligent power |
| b0: each night on |
| 8 lower bits | | b7 to b4: battery type (00: lead-acid battery,  01: lithium battery) |
| b3: charging method (0: PWM charging,  1: direct charging) |
| b2: no charging below 0 °C (0: disabled,  1: enabled) |
| b1 to b0: system voltage (00:12V battery,  01: 24V battery) |
|  | | | Historical data record (FLASH) | | | | | | | | | | | | | | |
| **Historical data** | | | 0xF000 | | | 2 | R | | Historical data of the current day | |  | |  |  | | The returned data is a block of data of the day(s) to be read, and the size of the block is 20 bytes | |
| 0xF001 | | | 2 | R | | Data before the current day | |  | |  |  | |

1. **Command Parsing and Example: (controller address 01H is taken for example, and hereinafter the actual PDU address is not taken into consideration)**

3.1 To read controller's system voltage and system current

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PDU address | Bytes | R/ W | Data | | Meaning |
| 000AH | 2 | R | 8 higher bits: system voltage | 0CH (decimal 12) | 12V |
| 18H (decimal 24) | 24V |
| 24H (decimal 36) | 36V |
| 30H (decimal 48) | 48V |
| 60H (decimal 96) | 96V |
| FFH (decimal 255) | Automatic recognition of system voltage |
| 8 lower bits: system current | 0AH (decimal 10) | 10A |
| 14H (decimal 20) | 20A |
| 1EH (decimal 30) | 30A |
| 2DH (decimal 45) | 45A |
| 3CH (decimal 60) | 60A |

According to "Table 1", the PDU address is known to be 000AH. Read 1 word (2 bytes)

To send: 01 03 000A 0001 A408

To receive: 01 03 02 181E 324C

Parsing: high byte 18H indicates the controller's system voltage is 24V, and low byte 1EH indicates the system current is 30A

3.2 To read the controller's model and the PDU addresses are known to be 000CH to 0013H in sequence and occupy a total of 16 bytes. Assume these addresses store the following data (ASCII) in sequence:

’’, ' ', ' ', ' ', 'M', 'T', '4', '8', '3' , '0' , ' ', ' ', ' ', ' ', ' ', ' ' To send: 01 03 000C 0008 840F

To receive: 01 03 10 2020 2020 4D54 3438 3330 2020 2020 2020 EE98

Parsing: this controller's model is MT4830 (the ASCII corresponding to 20H is ' ', and space can be neglected)

3.3 To read the controller's software version and hardware version, and the PDU addresses are known to be 0014H, 0015H, 0016H and 0017H in sequence

To send: 01 03 0014 0004 040D

To receive: 01 03 08 0003 0201 0001 0203 8A54

Parsing: (the highest byte OOH is not used) 030201H indicates the controller's software version is V03.02.01

(the highest byte OOH is not used) 010203H indicates the controller's hardware version is V01.02.03

3.4 To read the controller's product serial number and the PDU addresses are 0018H and 0019H in sequence as shown in "Table 1"

To send: 01 03 0018 0002 740F

To receive: 01 03 04 1501 FFFF AE4F

Parsing: 1501FFFFH is the product serial number, indicating it's the 65535th (hexadecimal FFFFH) unit produced in Jan. of 2015

3.5 To read battery capacity SOC, and the PDU address is known to be 0100H

To send: 01 03 0100 0002 C5F7

To receive: 01 03 02 0064 B9AF

Parsing: (the highest byte OOH is not used) the battery capacity SOC is 64H% （decimal 100%)

3.6 To read battery voltage:

Multiply the battery voltage reading by 0.1

The PDU address is known to be 0101H

To send: 01 03 0101 0001 D436

To receive: 01 03 02 007B F867

Parsing: *formula (* ***=*** *battery voltage \* 0.1)*

Battery voltage: (007BH, decimal 123), 007BH \* 0.1 = 12.3V

3.7To read the battery's surface temperature and controller temperature, and the PDU addresses are known to be 0102H and 0103 in sequence

To send: 01 03 0102 0002 6437

To receive: 01 03 02 0020 0028 73E7

Parsing: 0020H indicates the battery's surface temperature is 30 °C, and if the figure turns out to be 800AH, then it indicates the battery's surface temperature is -10 °C

0028H indicates the controller's temperature is 40 °C, and if the figure turns out to be 800BH, then it indicates the controller's temperature is -11 °C

3.8 To read street light voltage, (discharging) current and power, and the PDU addresses are known to be 0104H, 0105H and 0106H in sequence

To send: 01 03 0104 0003 45F6

To receive: 01 03 06 0078 00C8 00F0 00C5

Parsing:

Formula: street light voltage = street light voltage reading \* 0.1

0078H is the street light voltage reading, so the actual street light voltage is: 0078H \* 0.1 = 120 \* 0.1 = 12.0V

Formula: street light current = street light current reading \* 0.01

00C8H is the street light current reading, so the actual street light current is: 00C8H \* 0.01 = 200 \* 0.01 = 2.00A

00F0H is the street light power (decimal 240W) which can also be calculated via formula: street light voltage \* street light current

3.9To read solar panel voltage, charging current and charging power, and the PDU addresses are known to be 0107H, 0108H and 0109H in sequence

To send: 01 03 0107 0003 B5F6

To receive: 0090 0096 00D8 011E

Parsing:

Formula: solar panel voltage = solar panel voltage reading \* 0.1

00AAH is the solar panel voltage reading, so the actual solar panel voltage is: 0090H \* 0.1 = 144 \* 0.1 = 14.4V

Formula: solar panel charging current = solar panel charging current reading \* 0.01

0096H is solar panel charging current reading, so the actual solar panel charging current is: 0096H \* 0.01= 150 \* 0.01 = 1.50A

00D8H is solar panel charging power (decimal 216 W) which can also be calculated via formula: solar panel voltage \* solar panel charging current

3.10 To read the current day's min. battery voltage, max. battery voltage, max. charging current, max. discharging current, max. charging power, max. discharging power, charging amp-hrs, discharging amp-hrs, power generation, power consumption, and the PDU addresses are 010BH to 0114H in sequence as shown in "Table 1"

To send: 01 03 010B 0003 75F5

To receive: 01 03 06 0070 0084 00D8 20CD

Parsing: in the returned command

The 4th and 5th bytes 0070H indicate the current day's min. battery voltage: 0070H \* 0.1 = 112 \* 0.1 = 11.2V

The 6th and 7th bytes 0084H indicate the current day's max. battery voltage: 0084H \* 0.1 = 132 \* 0.1 = 13.2V

The 8th and 9th bytes 00D8H indicate the current day's max. charging current: 00D8H \* 0.01 = 216 \* 0.01 = 2.16V

E.g.: to read the controller's charging amp-hrs and discharging amp-hrs on the current day, and the PDU addresses are known to be 0111H and 0112H respectively

To send: 01 03 00111 0002 31D4

To receive: 01 03 04 0608 0810 7D75

Parsing: the 4th and 5th bytes 0608H are the current day's charging amp-hrs (decimal 1544AH);

Parsing: the 6th and 7th bytes 0810H are the current day's discharging amp-hrs (decimal 2064AH)

3.11 To read the number of operating days, over-discharges and full-charges, and the PDU addresses are 0115H, 0116H and 0117H respectively

To send: 01 03 0115 0003 15F3

To receive: 01 03 06 0008 0001 0006 1176

Parsing:

The 4th and 5th bytes 0008H are the number of operating days, indicating the system has operated for 8 days

The 6th and 7th bytes 0001H are the number of over-discharges, indicating th battery has been over-discharged once

The 8th and 9th bytes 0006H are the number of full-charges, indicating the battery has been fully charged for 6 times

3.12To read the battery's total charging amp-hrs and discharging amp-hrs, and the PDU addresses are known to be 0118H, 0119H, 011AH and 011BH in sequence

To send: 01 03 0118 0004 C5F2

To receive: 01 03 08 0001 0203 0000 0108 C0A3

Parsing: the 4th to 7th bytes 00010203H are the battery's total charging amp-hrs (decimal 66051AH = 66.051KAH)

The 8th to 11th bytes 00000108H are the battery's total discharging amp-hrs (decimal 264AH = 0.264KAH)

3.13 To read the controller's cumulative power generation and cumulative power consumption, and the PDU addresses are known to be 011CH to 011FH in sequence and occupy a total of 8 bytes.

To send: 01 03 011C 0004 840F

To receive: 01 03 08 0000 07D0 0000 03E8 550C

Parsing: 000007D0H are the controller's cumulative power generation (decimal 2000 kilowatt-hours)

The 8th to 11th bytes 000003E8H are the cumulative power consumption (decimal 1000 kilowatt-hours)

3.14 To read street light status, brightness and battery status, and the PDU addresses are known to be 0120H

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PDU address | Bytes | R/ W | Item | Value | | Meaning |
| 0120H | 2 | R | Street light status | 0 or 1 | High byte | b7: 0 indicates the street light is off,  1 indicates the street light is on |
| Street light brightness | 00 to 64H | b0 to b6: brightness value |
| Battery status |  | Low byte | 00H: charging deactivated |
| 01H: charging activated |
| 02H: mppt charging mode |
| 03H: equalizing charging mode |
| 04H: boost charging mode |
| 05H: floating charging mode |
| 06H: constant current (overpower) |

To send: 01 03 0120 0001 843C

To receive: 01 03 02 E402 7285

Parsing: E4H is (80H | 64H)

The 4th byte b7 being 1 indicates the street light is on, otherwise it's off, and b0 **to** b6 being 64H indicates the street light's brightness is 100%

The 5th byte 02H indicates mppt charging mode is in operation (for parsing of other statuses, refer to "PDU Address Allocation Table")

3.15To read faults and warnings, and the PDU addresses are 0121H and 0122H respectively

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PDU address | Bytes | R/ W | Item | byte | Meaning |
| 0121H 0122H | 4 | R | Controller fault and warning information | 16 High bit | B31 reserved |
| B30：circuit, charge MOS short circuit |
| B29：Anti-reverse MOS short |
| B28: solar panel reversely connected |
| B27 solar panel working point over-voltage |
| B26: solar panel counter-current |
| B25: photovoltaic input side over-voltage |
| B24: photovoltaic input side short circuit |
| B23: photovoltaic input overpower |
| B22: ambient temperature too high |
| B21: controller temperature too high |
| B20: load overpower  or load over-current |
| B19: load short circuit |
| B18: battery over-voltage |
| B17: battery under-voltage |
| B16: battery over-discharge |

To send: 01 03 0121 0002 95FD

To receive: 01 03 04 0101 0000 AA0F

Parsing:

The first four or five bytes for the fault information of the high 16 bit B24, 0101H for 1, said the photovoltaic input side short circuit, B16 1 said the battery over discharge

(for parsing of other fault codes, refer to the "Meaning" column of the "PDU Address Allocation Table")

3.16 To turn on the load, and knowing the PDU address is 010AH, you need write on/ off command into this address (0001 to turn on the load, 0000 to turn off the load)

To turn on the load:

To send: 01 06 010A 0001 69F4

To receive: 01 06 0100 0001 49F6

To turn off the load:

To send: 01 06 010A 0000 A834

To receive: 01 06 0100 0000 8836

3.17 To set street light brightness, and the PDU address is known to be E001H

If street light brightness needs to be set to 100% (hexadecimal 64H%) (the setting range is 0 **to** 100%)

To send: 01 06 E001 0064 EE21

To receive: 01 06 0101 0064 D81D

3.18 To read street light brightness, and the PDU address is known to be 0120H

To send: 01 03 0120 0001 843C

To receive: 01 03 02 E400 F344

Parsing:

The highest bit is responsible for turning on the street light, and the 7 lower bits of the high byte are for adjusting the brightness value, E4H&7FH = 64H = 100%

3.19 To set over-voltage threshold, charging limit voltage, equalizing charging voltage, boost charging voltage, floating charging voltage, boost charging recovery voltage, over-discharge recovery voltage, over-discharge voltage, boost charging time, equalizing charging interval, temperature compensation factor

The addresses are known to be E005H to E014H in sequence, and occupy a total of 16 words or 32 bytes

(for each setting range, refer to the "Meaning" column of the "PDU Address Allocation Table")

E.g.: parameter settings need to be done according to the following table

|  |  |  |
| --- | --- | --- |
| Item to set | Data processing | Data to send |
| Over-voltage threshold 17.0V | Multiplied by 10 | 17.0 \* 10 = 170, hexadecimal 00AAH |
| Charging limit voltage 15.5V | Multiplied by 10 | 15.5 \* 10 = 155, hexadecimal 009BH |
| Equalizing charging voltage 14.6V | Multiplied by 10 | 14.6 \* 10 = 146, hexadecimal 0092H |
| Boost charging voltage 14.4V | Multiplied by 10 | 14.4 \* 10 = 144, hexadecimal 0090H |
| Floating charging voltage 13.8V | Multiplied by 10 | 13.8 \* 10 = 138, hexadecimal 008AH |
| Boost charging recovery voltage 13.2V | Multiplied by 10 | 13.2 \* 10 = 132, hexadecimal 0084H |
| Over-discharge recovery voltage 12.6V | Multiplied by 10 | 12.6 \* 10 = 126, hexadecimal 007EH |
| Under-voltage threshold 17.0 V | Multiplied by 10 | 12.0 \* 10 = 120, hexadecimal 0078H |
| Over-discharge voltage 11.0V | Multiplied by 10 | 11.0 \* 10 = 110, hexadecimal 006EH |
| Over-discharge limit voltage 10.5V | Multiplied by 10 | 10.5 \* 10 = 105, hexadecimal 0069H |
| End of charge and discharge capacity 100%|50% |  | 100<<8|50, hexadecimal 6432H |
| Over-discharge time delay 5S |  | Hexadecimal 0005H |
| Equalizing charging time 60min |  | 003CH |
| Boost charging time 60min |  | 003CH |
| Equalizing charging interval 30 days |  | 001EH |
| Temperature compensation factor 5 mV/ °C/ 2V |  | 0005H |

To send: 01 10 E005 0010 00AA 009B 0092 0090 008A 0084 007E 0078 006E 0069 6432 0005 003C 003C 001E 0005 C140

To receive: 01 10 E005 0010 E604

3.20To set the load's 1st, 2nd, 3rd and morning on stage operating durations and powers, and the PDU addresses are known to be E015H **to** E01CH and occupy a total of 8 words or 16 bytes

E.g.: parameter settings need to be done according to the following table

|  |  |  |
| --- | --- | --- |
| Item to set | Set value | Data to send |
| 1st-stage operating duration | 4 hours | 0004H |
| 1st-stage operating power | 100% | 0064H (decimal 100) |
| 2nd-stage operating duration | 0 hours | 0000H |
| 2nd-stage operating power | 75% | 004BH (decimal 75) |
| 3rd-stage operating duration | 4 hours | 0004H |
| 3rd-stage operating power | 50% | 0032H (decimal 50) |
| Morning on operating duration | 0 hours | 0000H |
| Morning on operating power | 25% | 0019H (decimal 25) |

To send: 01 10 E015 0008 10 0004 0064 0000 004B 0004 0032 0000 0019 957F

To receive: 01 10 E015 0008 E7CB

3.21 To set load working mode, and the PDU address is known to be E01DH

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PDU address | Bytes | R/ W | Item | Value | Meaning |
| E01DH | 2 | R/W | Load working modes | 00H | Sole light control, light control over on/ off of load |
| 01H | Load is turned on by light control, and goes off after a time delay of 1 hours |
| 02H | Load is turned on by light control, and goes off after a time delay of 2 hours |
| 03H | Load is turned on by light control, and goes off after a time delay of 3 hours |
| 04H | Load is turned on by light control, and goes off after a time delay of 4 hours |
| 05H | Load is turned on by light control, and goes off after a time delay of 5 hours |
| 06H | Load is turned on by light control, and goes off after a time delay of 6 hours |
| 07H | Load is turned on by light control, and goes off after a time delay of 7 hours |
| 08H | Load is turned on by light control, and goes off after a time delay of 8 hours |
| 09H | Load is turned on by light control, and goes off after a time delay of 9 hours |
| 0AH (decimal 10) | Load is turned on by light control, and goes off after a time delay of 10 hours |
| 0BH (decimal 11) | Load is turned on by light control, and goes off after a time delay of 11 hours |
| 0CH (decimal 12) | Load is turned on by light control, and goes off after a time delay of 12 hours |
| 0DH (decimal 13) | Load is turned on by light control, and goes off after a time delay of 13 hours |
| 0EH (decimal 14) | Load is turned on by light control, and goes off after a time delay of 14 hours |
| 0FH (decimal 15) | Manual mode |
| 10H (decimal 16) | Debugging mode |
| 11H (decimal 17) | Normal on mode |

According to the "PDU Address Allocation Table", if "load is turned on by light control, and goes off after a time delay of 8 hours" needs to be set to, send command 0008H

To send: 01 06 E01D 0008 2FCA

To receive: 01 06 E01D 0008 2FCA

3.22 Reset to factory defaults

To send: 01 78 0000 0001 6000

To receive: 01 78 0000 0001 6000

Parsing: 01 is the id number, 78 is the command to reset to factory defaults, and 6000 is for checking

3.23 Clear history

To send: 01 79 0000 0001 5DC0

To receive: 01 79 0000 0001 5DC0

Parsing: 01 is the id number, 79 is the command to clear history, and 5DC0 is for checking. Be careful to use this command, as execution of it will lead to loss of all historical data and to recover the data will be impossible.

To inquire about controller addresses

1. To know the address of a certain controller, you can use a read command (write commands are not recommended) to conduct address polling. When receiving data conforming to the sent command, the address contained in the command is the address of the controller (note: this method applies to separate controller connection)

2) To seek out multiple controllers connected via communication lines, also perform address polling, and the returned command conforming to related rules contains the address information of the controllers, so you know which controllers are connected to the server.