**Modbus \_ RTU Communication Description**

Version: Version 1.0, December 2023

**Modbus \_ RTU Communication Description**

Communication data format frame format

Device Register Address Fault Code Table

Warning code description

Description of power flow signs

CRC check algorithm

# Communication data format

During communication, the data is sent back in the form of word (WORD — 2 bytes). In each word sent back, the high byte is in the front and the low byte is in the back.If two words are sent back consecutively (such as long shaping), the high word is in the front and the low word is in the back.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data type** | **Number of registers** | **Number of bytes** | **Explain** |
| Character type | 1 | 1 | Send back two characters at a time, and use 0 to supplement less than two characters. |
| Integer | 1 | 2 | Send back once, high byte first, low byte last |
| Long | 2 | 4 | Send it back in two words, with the high word in the front and the low word in the back. |

**Frame format**

**Register content query (function code 03H)**

The start address and end address of the query must be the start address and end address of a complete data block,Otherwise, the data returned is incorrect.For example, if the start address of the register of the device serial number is 186 and the length is 12, the start address cannot be between 186 and 198 during the query.Similarly, the end address (start address + number of registers read) cannot fall within this range.

**Frame format sent by upper computer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte order** | **Code** | **Examples** | **Explain** |
| 0 | Device address | 01H | Device address (1 ~ 247) |
| 1 | 03H | 03H | Function Code |
| 2 | Start Register Address High Byte | 00H | Upper 8 bits of register address |
| 3 | Start register address low byte | 10H | Lower 8 bits of register address |
| 4 | Number of registers high byte | 00H | Upper 8 bits of register number |
| 5 | Low byte of register number | 02H | Number of registers Lower 8 bits |
| 6 | CRC16 check high byte | C0H | High 8 bits of CRC16 check |
| 7 | CRC16 check low byte | CBH | CRC16 check lower 8 bits |

**The lower computer parses the returned frame format successfully.**

|  |  |  |
| --- | --- | --- |
| **Byte order** | **Code** | **Explain** |
| 0 | Device address | Device address (1 ~ 247) |
| 1 | 03H | Function Code |
| 2 | Number of bytes of data sent back | N = number of registers requested \* 2 |
| 3 | 1st register data high byte |  |
| 4 | 1st register data low byte |  |
| ...... | ...... |  |
| ...... | ...... |  |
|  | Nth register data high byte |  |
|  | Nth register data low byte |  |
| N+3 | CRC16 check high byte | The bytes before the CRC are checked |
| N+4 | CRC16 check low byte |  |

**Lower computer analysis error return frame format**

|  |  |  |
| --- | --- | --- |
| **Byte order** | **Code** | **Explain** |
| 0 | Device address | Device address (1 ~ 247) |
| 1 | 03H | Function Code |
| 2 | Number of bytes of data sent back | N = number of registers requested \* 2 |
| 3 | 1st 0 | A total of N zeros are looped back |
| 4 | 2nd 0 |  |
| ...... | ...... |  |
| ...... | ...... |  |
|  | Nth-1st 0 |  |
|  | Nth 0 |  |
| N+3 | CRC16 check high byte | The bytes before the CRC are checked |
| N+4 | CRC16 check low byte |  |

**Register reading example:**

Read the data from the RMS value of the mains voltage (start register 202) to the average value of the mains power, where the mains voltage returns 220.0 V,Mains frequency returns 50.0 Hz and average mains power returns 1200 w

Host computer: 01 03 00 CA 00 03 25 F5

Lower computer: 01 03 06 08 FC 13 88 04 B0 F7 F3

**Register content setting (function code 10 H)**

**Frame format sent by upper computer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte order** | **Code** | **Examples** | **Explain** |
| 0 | Device address | 01H | Device address (1 ~ 247) |
| 1 | 10H | 10H | Function Code |
| 2 | Start Register Address High Byte | 01H | Upper 8 bits of register address |
| 3 | Start register address low byte | 10H | Lower 8 bits of register address |
| 4 | Number of registers high byte | 00H | Upper 8 bits of the number of registers (constantly equal to 0) |
| 5 | Low byte of register number | 02H | Number of registers Lower 8 bits |
| 6 | Number of bytes to be written |  | N = Number of registers \* 2 |
| 7 | 1st register data high byte |  |  |
| 8 | 1st register data low byte |  |  |
| ...... | ...... |  |  |
| ...... | ...... |  |  |
|  | Nth register data high byte |  |  |
|  | Nth register data low byte |  |  |
| N+7 | CRC16 check high byte |  | High 8 bits of CRC16 check |
| N+8 | CRC16 check low byte |  | CRC16 check lower 8 bits |

**The lower computer parses the returned frame format successfully.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Byte order** | **Code** | **Examples** | **Explain** |
| 0 | Device address | 01H | Device address (1 ~ 247) |
| 1 | 10H | 10H | Function Code |
| 2 | Start Register Address High Byte | 01H | Upper 8 bits of register address |
| 3 | Start register address low byte | 10H | Lower 8 bits of register address |
| 4 | Number of registers high byte | 00H | Upper 8 bits of the number of registers (constantly equal to 0) |
| 5 | Low byte of register number | 02H | Number of registers Lower 8 bits |
| 6 | CRC16 check high byte | 41H | High 8 bits of CRC16 check |
| 7 | CRC16 check low byte | F1H | CRC16 check lower 8 bits |

**Lower computer analysis error return frame format**

|  |  |  |
| --- | --- | --- |
| **Byte order** | **Code** | **Explain** |
| 0 | Device address | Device address (1 ~ 247) |
| 1 | 90H | Function Code |
| 2 | Error code | Error code |
| 3 | CRC16 check high byte | High 8 bits of CRC16 check |
| 4 | CRC16 check low byte | CRC16 check lower 8 bits |

**Error code description**

|  |  |
| --- | --- |
| **Code** | **Explain** |
| 01H | Read-only register operated on |
| 03H | Write data that exceeds the acceptable range |
| 07H | Registers that are not allowed to be modified in the current operating mode |

**Register write example:**

Set the output voltage (start register 320) to 220V

Host computer: 01 10 01 40 00 01 02 08 98 BE 3A

Lower computer: 01 10 01 40 00 01 01 E1

**Device register address**

R: Indicates that the 03 H command is supported if it is readable. W: means writable, that is, supports 10 H commands. Int: Integer; Long: Long;UInt: unsigned integer; ULong: unsigned long integer; ASC: ASCII code Max: maximum value; Min: minimum value

Addresses in the following table are in decimal notation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Equipment fault code |  | ULong | 100 | 2 | R | 32-bit fault code. Each bit corresponds to a fault code. See the fault code table for details. Fault code 1 corresponds to bit1, fault code 2 corresponds to bit2, and so on. |
| Reserved |  |  | 102 | 2 |  | Reserved address |
| Obtain the warning code for unmasked processing |  |  | 104 | 2 |  | The 32-bit warning code is described in the warning code description |
| Reserved |  |  | 106 | 2 |  | Reserved address |
| Obtain the warning code after shield processing |  | ULong | 108 | 2 | R/W | The 32-bit warning code is described in the warning code description |
| Reserved |  |  | 110 | 61 |  | Reserved address |
| Device type |  | UInt | 171 | 1 | R |  |
| Device name |  | ASC | 172 | 12 | R/W | Device name, written or read in ASCII |
| Invalid data |  | UInt | 184 | 1 | R | Agreement number, return 1 for this agreement |
| Reserved |  |  | 185 | 1 |  | Reserved address |
| Device serial number |  | ASC | 186 | 12 | R |  |
| Reserved |  |  | 198 | 2 |  | Reserved address |
| Invalid data |  | UInt | 200 | 1 |  | Internal command |
| Working mode |  | UInt | 201 | 1 | R | 0: Power on mode  1: Standby mode  2: Mains mode  3: Off-grid mode  4: Bypass mode  5: Charging mode  6: Failure Mode |
| Mains voltage effective value | 0.1v | Int | 202 | 1 | R |  |
| Mains frequency | 0.01Hz | Int | 203 | 1 | R |  |
| Average mains power | 1w | Int | 204 | 1 | R |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Effective value of inverter voltage | 0.1v | Int | 205 | 1 | R |  |
| Effective value of inverter current | 0.1A | Int | 206 | 1 | R |  |
| Inverter frequency | 0.01Hz | Int | 207 | 1 | R |  |
| Inverter power average | 1w | Int | 208 | 1 | R | Positive indicates inverter output and negative indicates inverter input |
| Inverter charging power | 1w | Int | 209 | 1 | R |  |
| Effective value of output voltage | 0.1v | Int | 210 | 1 | R |  |
| Effective value of output current | 0.1A | Int | 211 | 1 | R |  |
| Output frequency | 0.01Hz | Int | 212 | 1 | R |  |
| Output active power | 1w | Int | 213 | 1 | R |  |
| Output apparent power | 1VA | Int | 214 | 1 | R |  |
| Average battery voltage | 0.1v | Int | 215 | 1 | R |  |
| Average battery current | 0.1A | Int | 216 | 1 | R |  |
| Average battery power | 1w | Int | 217 | 1 | R |  |
| Invalid data |  |  | 218 | 1 |  | Internal command |
| Average PV voltage | 0.1v | Int | 219 | 1 | R |  |
| Average PV current | 0.1A | Int | 220 | 1 | R |  |
| Reserved |  |  | 221 | 2 |  | Reserved address |
| Average PV power | 1w | Int | 223 | 1 | R |  |
| Average PV charging power | 1w | Int | 224 | 1 | R |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Percent of load | 1% | Int | 225 | 1 | R |  |
| DCDC temperature | 1℃ | Int | 226 | 1 | R |  |
| Inverter temperature | 1℃ | Int | 227 | 1 | R |  |
| Reserved |  |  | 228 | 1 |  | Reserved address |
| Battery percentage | 1% | UInt | 229 | 1 | R |  |
| Invalid data |  |  | 230 | 1 |  | Internal command |
| Power flow status |  | UInt | 231 | 1 | R | See the description of power flow flag bit for details. |
| Battery current filter average | 0.1A | Int | 232 | 1 | R | A positive number indicates charging and a negative number indicates discharging. |
| Average value of inverter charging current | 0.1A | Int | 233 | 1 | R |  |
| Average PV charging current | 0.1A | Int | 234 | 1 | R |  |
| Invalid data |  |  | 235 | 1 |  | Internal command |
| Invalid data |  |  | 236 | 1 |  | Internal command |
| Reserved |  |  | 237 | 63 |  | Reserved address |
| Output mode |  | Uint | 300 | 1 | R/W | 0: single machine;  1: parallel;  2: Three-phase combination-P1  3: Three-phase combination-P2  4: Three-phase combination-P3 |
| Output priority |  | Uint | 301 | 1 | R/W | 0: Main-PV-Battery (UTI)  1: PV-mains-battery (SOL) [priority inverter]  2: PV-battery-mains (SBU)  3: PV-Mains-Battery (SUB) [Priority Mains] |
| Input voltage range |  | Uint | 302 | 1 | R/W | 0：APL；  1：UPS； |
| Buzzer mode |  | Uint | 303 | 1 | R/W | 0: mute in all cases;  1: Sound when the input source changes or there is a specific warning or fault; 2: Sound when there is a specific warning or fault;  3: Sound in case of fault; |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Reserved |  |  | 304 | 1 | R/W | Reserved address |
| LCD backlight |  | Uint | 305 | 1 | R/W | 0: Timed closing;  1: Always on; |
| LCD automatically returns to the home page |  | Uint | 306 | 1 | R/W | 0: Do not return automatically;  1: Automatic return after 1 minute; |
| Energy saving mode switch |  | Uint | 307 | 1 | R/W | 0: Energy saving mode off;  1: Energy-saving mode on; |
| Overload automatic restart |  | Uint | 308 | 1 | R/W | 0: Overload failure does not restart;  1: Automatic restart in case of overload; |
| Over-temperature automatic restart |  | Uint | 309 | 1 | R/W | 0: No restart for over-temperature fault;  1: Automatic restart for over-temperature fault; |
| Overload Bypass Enable |  | Uint | 310 | 1 | R/W | 0: forbidden;  1: Enable; |
| Reserved |  |  | 311 | 2 |  | Reserved address |
| Battery Eq Mode Enable |  | Uint | 313 | 1 | R/W | 0: forbidden;  1: Enable; |
| Warning Mask [I] |  | ULong | 314 | 2 | R/W | The warning corresponding to 1 is normally displayed, and the warning corresponding to 0 is shielded. |
| Dry contact |  | Uint | 316 | 1 | R/W | 0: normal mode;  1: Grounding box mode; |
| Reserved |  |  | 317 | 3 |  | Reserved address |
| Output voltage | 0.1v | Uint | 320 | 1 | R/W | 2200: 220V output;  2300: 230v output;  2400: 240v output; |
| Output frequency | 0.01Hz | Uint | 321 | 1 | R/W | 5000: 50Hz output;  6000: 60Hz output; |
| Battery type |  | Uint | 322 | 1 | R/W | 0：AGM；  1：FLD；  2：USER；  3：Li1  4：Li2  5：Li3  6：Li4 |
| Battery overvoltage protection point [A] | 0.1v | Uint | 323 | 1 | R/W | Range: (B + 1V \* J) ~ 16.5v \* J |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Maximum charge voltage [B] | 0.1v | Uint | 324 | 1 | R/W | Range: C ~ (A-1v) |
| Floating charge voltage  [C] | 0.1v | Uint | 325 | 1 | R/W | Range: (12v \* J) ~ B |
| Mains mode battery discharge recovery point [D] | 0.1v | Uint | 326 | 1 | R/W | Range: (B-0.5V \* J) ~ Max (12V \* J, E)  Set to 0 to indicate a full recovery |
| Battery low voltage protection point in mains mode [E] | 0.1v | Uint | 327 | 1 | R/W | Range: Min (14.3v \* J, D) ~ Max (11v \* J, F) |
| Reserved |  |  | 328 | 1 |  | Reserved address |
| Off-grid mode battery low voltage protection point [F] | 0.1v | Uint | 329 | 1 | R/W | Range: (10v \* J) ~ Min (13.5v \* J, E) |
| Waiting time from constant voltage to floating charge | min | Uint | 330 | 1 | R/W | Range: 1 ~ 900 min  Set to 0 to default to 10 min |
| Battery charging priority |  | Uint | 331 | 1 | R/W | 0: mains supply is preferred;  1: PV priority;  2: PV is at the same level as mains supply;  3: PV charging only allowed |
| Maximum charge current [G] | 0.1A | Uint | 332 | 1 | R/W | Range: Max (10 A, H) ~ 80 A |
| Maximum mains charging current [H] | 0.1A | Uint | 333 | 1 | R/W | Range: 2A ~ G |
| The charging voltage of Eq | 0.1v | Uint | 334 | 1 | R/W | Range: C ~ (A-0.5v \* J)<http://c/> |
| bat\_eq\_time | min | Uint | 335 | 1 | R/W | Range: 0 ~ 900 |
| Eq timed out | min | Uint | 336 | 1 | R/W | Range: 0 ~ 900 |
| Two-time Eq charge interval | day | Uint | 337 | 1 | R/W | Range: 1 ~ 90 |
| Automatic Mains Output Enable |  | Uint | 338 | 1 | R/W | 0: No mains power output without pressing the power button 1: Automatic mains power output without pressing the power button |
| Reserved |  |  | 339 | 2 |  | Reserved address |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Mains mode battery discharge SOC protection value [K] | 1% | Uint | 341 | 1 | R/W | Range: 20% ~ 50% |
| Mains mode battery discharge SOC recovery value | 1% | Uint | 342 | 1 | R/W | Range: 60% ~ 100% |
| Battery discharge SOC protection value in off-grid mode | 1% | Uint | 343 | 1 | R/W | Range: 3% ~ Min (K, 30%) |
| Reserved |  |  | 344 | 7 |  |  |
| Maximum discharge current protection | 1A | Uint | 351 | 1 | R/W | Maximum discharge current protection value in stand-alone mode |
| Reserved |  |  | 352 | 54 |  |  |
| Boot mode |  | Uint | 406 | 1 | R/W | 0: Can be powered on locally or remotely  1: Only local boot  2: Remote boot only |
| Reserved |  |  | 407 | 13 |  | Reserved address |
| Remote switch |  | Uint | 420 | 1 | R/W | 0: Remote shutdown  1: Remote power-on |
| Invalid data |  |  | 421 | 1 |  | Internal command |
| Reserved |  |  | 422 | 3 |  | Reserved address |
| Forcing the charge of Eq |  | Uint | 425 |  | W | 1: Manually force Eq to charge once |
| Exits the fail-locked state |  | Uint | 426 |  | W | 1: Exit the fault lock state (it will take effect only when the machine enters the fault mode) |
| Invalid data |  |  | 427 | 1 |  | Internal command |
| Reserved |  |  | 428 | 22 |  | Reserved address |
| Invalid data |  |  | 450 | 7 |  | Internal command |
| Reserved |  |  | 457 | 3 |  | Reserved address |
| Clear the record |  |  | 460 | 1 | W | 0 xAA: clear the operation record and fault record (effective in non-off-grid mode) |
| Reset user parameters |  |  | 461 | 1 | W | 0 xAA: User parameters are restored to default values (works in non-off-grid mode) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data name** | **Unit** | **Data format** | **Start address** | **Number of registers** | **Read and write** | **Remark** |
| Invalid data |  |  | 462 | 6 |  | Internal command |
| Reserved |  |  | 468 | 32 |  | Reserved address |
| Invalid data |  |  | 500 | 34 |  | Internal command |
| Reserved |  |  | 534 | 66 |  | Reserved address |
| Invalid data |  |  | 600 | 34 |  | Internal command |
| Program version |  | ASC | 626 | 8 | R |  |
| Reserved |  |  | 634 | 7 |  | Reserved address |
| Rated power | w | UInt | 643 | 1 | R |  |
| Rated number of cells [J] | PCS | UInt | 644 | 1 | R |  |
| Reserved |  |  | 645 | 55 |  | Reserved address |
| Fault record storage information [K] |  | ULong | 700 | 2 | R | Upper 16 bits: the position of the latest record;  Lower 16 bits: total number of existing fault messages |
| Fault Information Query Index |  | Uint | 702 | 1 | R/W | Set the fault information index to be queried, range: 0 ~ total number of existing fault information |
| Fault Record [M] |  |  | 703 | 26 | R | See the fault record format for details. |
| Run the log |  |  | 729 | 16 | R | See the operation log description for details |
| Reserved |  |  | 745 | 5 |  | Reserved address |

**Fault code table**

|  |  |
| --- | --- |
| **Fault code** | **Explain** |
| bit1 | Inverter overtemperature |
| bit2 | DCDC overtemperature |
| bit3 | Battery overvoltage |
| bit4 | PV overtemperature |
| bit5 | The output is shorted |
| bit6 | Inverter overvoltage |
| bit7 | Output overload |
| bit8 | Busbar overvoltage |
| bit9 | Bus soft start timeout |
| bit10 | PV overcurrent |
| bit11 | PV overpressure |
| bit12 | Battery overcurrent |
| bit13 | Inverter overcurrent |
| bit14 | Busbar low voltage |
| bit15 | Reserved |
| bit16 | Inverter DC component is too high |
| bit17 | Reserved |
| bit18 | Output current zero bias is too large |
| bit19 | Inverter current zero bias is too large |
| bit20 | Excessive zero bias of battery current |
| bit21 | PV current zero bias is too large |
| bit22 | Inverter low voltage |
| bit23 | Inverter negative power protection |
| bit24 | Loss of parallel system host |
| bit25 | Abnormal synchronization signal of parallel system |
| bit26 | Incompatible battery type |
| bit27 | Incompatible parallel version |

**Warning code description**

A system warning is a 32-bit unsigned long integer, one for each bit,Each bit can also be masked by a warning mask I. After masking, the corresponding warning will not be read on the LCD or by a command.

**Warning code table**

|  |  |
| --- | --- |
| **Warning Code** | **Explain** |
| bit 0 | Mains supply zero-crossing loss |
| bit 1 | Mains waveform is abnormal |
| bit 2 | Mains overvoltage |
| bit 3 | Mains undervoltage |
| bit 4 | The mains supply is too frequent |
| bit 5 | Mains underfrequency |
| bit 6 | PV undervoltage |
| bit 7 | Overtemperature |
| bit 8 | Low battery voltage |
| bit 9 | Battery is not connected |
| bit 10 | Overload |
| bit 11 | Charge the battery Eq |
| bit 12 | The battery is discharged to a low voltage and has not been charged back to the recovery point |
| bit 13 | Output power derating |
| bit 14 | The fan is blocked |
| bit 15 | PV energy too low to use |
| bit 16 | Parallel communication is interrupted |
| bit 17 | Inconsistent single parallel output mode |
| bit 18 | Excessive voltage difference of parallel battery |
| bit19 | Abnormal lithium battery communication |
| bit20 | Battery discharge current exceeds the set value |
| bit 21~31 | Reserved |

**Description of power flow signs**

The power flow has a 16-bit unsigned integer representation, and the meaning of each bit is as follows:

**Power flow level table**

|  |  |
| --- | --- |
| **Bit** | **Explain** |
| bit0~1 | 0: PV is not connected to the system  1: PV is connected to the system |
| bit2~3 | 0: The mains supply is not connected to the system  1: The mains supply has been connected to the system |
| bit4~5 | 0: The battery will not be charged or discharged  1: Battery charging  2: Battery discharge |
| bit6~7 | 0: The system does not output on-load  1: System output with load |
| bit8 | 0: Mains power is not charged  1: Mains charging |
| bit9 | 0: PV not charged  1: PV charging |
| bit10 | 0: Battery icon is on  1: Battery icon off |
| bit11 | 0: PV icon is on  1: PV icon off |
| bit12 | 0: The electric map of the city lights up  1: Mains icon off |
| bit13 | 0: The load icon is on  1: Load icon off |

**CRC check algorithm**

Parametric model : CRC - 16 / MODBUS X16 + X15 + X2 + 1

C Language code

const char auchCRCHi [] = {

0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,

0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,

0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 ,

0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 ,

0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,

0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 ,

0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,

0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 ,

0x00 , 0xC1 , 0x81 , 0x40 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 ,

0x80 , 0x41 , 0x00 , 0xC1 , 0x81 , 0x40 , 0x00 , 0xC1 , 0x81 , 0x40 ,

0x01 , 0xC0 , 0x80 , 0x41 , 0x01 , 0xC0 , 0x80 , 0x41 , 0x00 , 0xC1 ,

0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,

0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,

0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,

0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40

} ;

const char auchCRCLo[] = {

0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,

0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,

0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,

0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,

0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,

0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,

0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,

0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,

0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,

0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,

0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,

0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,

0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,

0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,

0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,

0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,

0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,

0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,

0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,

0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,

0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,

0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,

0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,

0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,

0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,

0x43, 0x83, 0x41, 0x81, 0x80, 0x40

} ;

unsigned short sModbusCrc16(INT8U \*chMsg, INT16U dataLen)

{

unsigned char ubCRCHi = 0xFF; unsigned char ubCRCLo = 0xFF; unsigned char duwIndex; while (dataLen --)

{

duwIndex = 0xff&(ubCRCHi ^ \*chMsg++);

ubCRCHi = 0xff&(ubCRCLo ^ auchCRCHi[duwIndex]); ubCRCLo = auchCRCLo[duwIndex];

}

return (ubCRCHi << 8 | ubCRCLo);

}