

N4VIA02 Modbus RTU V2

Function code

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16 (2)
	03 Read			
	06/16 Write			

Read-only register,Read Function code Is 03				
Register address	Register contents	Number of bytes	Units	Remarks
0x0000-0x0001	CH0-CH1 Current value	2 Units 1mA		For example, the obtained data is 255, which is equal to 255mA. Ch2 cannot be read independently. To obtain the value of ch 2, read ch 1 and 2 simultaneously. Command: 01 03 00 00 00 02 C4 0B
0X0020-0X0021 (32-33)	CH0-CH1 Voltage value	2 Units 0.01V		For example, the obtained data is 1000, which is equal to 10.00V Ch2 cannot be read independently. To obtain the value of ch 2, read ch 1 and 2 simultaneously. Command: 01 03 00 20 00 02 C5 C1
Read / write register; Read function code is 03 ,Write function code is 06 and 16(0x10)				
0X0040-0X0041 (64-65)	CH0-CH1 Current correction value This register is only writable Read data: 0XFFFF	2 Units 1mA		If the current is deviated, please input the correct current value for correction Ch2 cannot be read independently. To obtain the value of ch 2, read ch 1 and 2 simultaneously. Command: 01 03 00 40 00 02 C5 DF
0X0060-0X0061 (96-97)	CH0-CH1 Voltage correction value This register is only writable Read data: 0XFFFF	2 Units 0.01V		If there is a deviation in the voltage value, please input the correct voltage value for correction Ch2 cannot be read independently. To obtain the value of ch 2, read ch 1 and 2 simultaneously. Command:

			01 03 00 60 00 02 C4 15	
0x00FA (250)	Current and voltage automatic reporting function The current of 2 channels is reported at the same time, and the voltage of 2 channels is reported at the same time	2	Second	0: Query function (default) 1-255: Automatically report, the unit is second. 1: Report every 1 second 2: Report every 2 seconds 10: Report every 10 seconds Maximum interval of 255 seconds
0x00FB (251)	Factory data reset	2	The address code is 0xFF, and at the current baud rate, enter the following commands to restore the factory settings: FF 06 00 FB 00 00 ED E5	
0x00FC (252)	Data return delay	2	0~1000	After receiving the command, return the data interval time (unit MS)
0x00FD (253)	RS485 address (Station address)	2		Read Address 0xFF Write Address 1-254
0x00FE (254)	Baud rate	2		0~8 0: 1200; 1: 2400; 2: 4800; 3: 9600; 4: 19200; 5: 38400; 6: 57600; 7: 115200 8: Factory reset
0x00FF (255)	Serial check digit	2		0: no parity 1: even parity 2: odd parity

The default baud rate is 9600, 8 data bits, one stop bit, and no parity bit.

Modbus RTU Communication protocol:

1. Read the current current

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	-----------------	-------------------------	-----------------	--------------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code 0x03

Register address: 0x0000~0x0001

Read number: 0x0002

The length of the returned current data is two bytes, and the high bit is at the front and the low bit at the back. Converting these two bytes into a decimal number is the current current value (unit: mA); when the highest bit is 1, it means a negative value. It is necessary to add 1 to this value, or directly subtract 65536 from this value, which is the current current value.

The following example illustrates:Read the current value of channel 0,

send data(RS485 address is 1): **01 03 00 00 00 01 84 0A**

Returns data: 01 03 02 03 E8 B8 FA

01 address code, 03 function code, 02 is data length, B8 FA is crc16 checksum

03E8 is the current value, the highest bit is 0, so the temperature is positive, convert it to decimal = 1000, that is, the current current size is 1000mA;

2. Read the voltage

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code 0x03

Register address: 0x0020~0x0021

Read number: 0x0002

The data length of the returned voltage value is two bytes. The high order is in

the front and the low order. Convert these two bytes into a decimal number and divide it by 100 to get the current resistance value (unit: V).

The following is an example: Read the voltage value of channel 0,

send data(RS485 address is 1): **01 03 00 20 00 01 85 C0**

Returns data: 01 03 02 05 07 FA D6

01 address code, 03 function code, 02 is data length, FA D6 is crc16 checksum

0507 is the resistance value, convert it to decimal = 1287, $1287 \div 100 = 12.87$ V is

the current channel 0 voltage;

3. Read RS485 address:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16 (2)
---	--------------	-------------------------	-----------------	-----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16 (2)
---	--------------	------------------------	----------	-----------

Broadcast address 0xff

Function code 0x03

Register address: 0x00FD

Read number: 0x0001

For example:

send data(RS485 address is 1): FF 03 00 FD 00 01 00 24

Returns data: FF 03 02 00 01 50 50

FF address code, 03 function code, 02 length, 01 current module address, 50 50
crc16 checksum.

Note: When using this command, only one module can be connected to the 485 bus, more than one will cause an error!

4. Write RS485 address:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16 (2)
---	--------------	-------------------------	-----------------	-----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16 (2)
---	--------------	------------------------	----------	-----------

Function code: 0x06/0x10

Register address: 0x00FD

Setting Content: 2Bytes(1-254)

For example, The current RS485 address is 1, We need to change the RS485 address to 3:

send data(RS485 address is 1):01 06 00 FD 00 03 58 3B

Returns data: 01 06 00 FD 00 03 58 3B

5. Read baud rate:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code 0x03

Register address: 0x00FE

Read number: 0x0001

For example:

send data(RS485 address is 1): 01 03 00 FE 00 01 E5 FA

Returns data: 01 03 02 00 03 F8 45

01 RS485 address, 03 Function, 02 length, F8 45 crc16

03 means the current baud rate is 9600bps

Baud rate corresponds to the number: 0: 1200; 1: 2400; 2: 4800; 3: 9600; 4: 19200;

5: 38400; 6: 57600; 7: 115200.

6. Write baud rate:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code: 0x06/0x10

Register address: 0x00FE

Setting Content: 2Bytes (0-4)

For example, Change the baud rate to 4800bps:

send data(RS485 address is 1):01 06 00 FE 00 02 69 FB

Returns data: 01 06 00 FE 00 02 69 FB

Baud rate corresponds to the number: 0: 1200; 1: 2400; 2: 4800; 3: 9600; 4: 19200; 5: 38400; 6: 57600; 7: 115200; 8: Factory reset

Note: 1 The baud rate will be updated when the module is powered up again!

2 The factory setting can be restored when the baud rate corresponding to the number is 8.

For example: 01 06 00 FE 00 08 E9 FC

7. Set the current correction:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Broadcast address 0x01~0xFE

Function code 0x06/0x10

Register address: 0x0040~0x0041

Setting Content: 2 Bytes, The highest bit represents the sign of positive and negative values, 0 represents positive, 1 represents negative, and the unit is 1mA. When the highest bit is 1, it represents a negative value. At this time, it is necessary to add 1 to this value, or directly subtract 65536 from this value, which is the current current value. 0x0040 represents the current correction value of channel 0, and 0x0041 represents the current correction value of channel 1.

For example: The current channel 0 current is 1000mA, and the read current is 990mA. Enter the correct current value to correct. 1000 is converted to hex

0x03E8

send data(RS485 address is 1): 01 06 00 40 03 E8 88 A0

Returns data: 01 06 00 40 03 E8 88 A0 The return frame is the same as the send frame.

8. Set the voltage correction value:

If the read voltage value deviates from the actual voltage value, please input the correct voltage value to this register.

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16 (2)
---	--------------	-------------------------	-----------------	-----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16 (2)
---	--------------	------------------------	----------	-----------

Broadcast address 0x01~0xFE

Function code 0x06/0x10

Register address: 0x0060~0x0061

0x0060 represents the voltage correction value of channel 0, and 0x0061 represents the voltage correction value of channel 2.

For example: the actual voltage of the current channel 0 is 12.00V, and the read value is 11.60V. Enter the correct voltage value to correct. $12.00 \times 100 = 1200$, converted to hexadecimal 0x04B0

send data(RS485 address is 1): 01 06 00 60 04 B0 8A A0

Returns data: 01 06 00 60 04 B0 8A A0 The return frame is the same as the send frame.

9. Read serial port check digit:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16 (2)
---	--------------	-------------------------	-----------------	-----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16 (2)
---	--------------	------------------------	----------	-----------

Broadcast address 0x01~0xFE

Function code 0x03

Register address: 0x00FF

Read number: 0x0001

For example:

send data(RS485 address is 1): 01 03 00 FF 00 01 B4 3A

Returns data: 01 03 02 00 00 B8 44

01 address code, 03 function code, 02 length, 0000 means no check, B8 44 crc16 check.

0: No parity; 1: Even parity; 2: Odd parity.

10. Set serial port check digit:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	-----------------	-------------------------	-----------------	--------------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	-----------------	------------------------	----------	--------------

Broadcast address 0x01~0xFE

Function code 0x06/0x10

Register address: 0x00FF

Read number: 0x0001

For example: Set the serial port parity to even parity,

send data(RS485 address is 1): 01 06 00 FF 00 01 78 3A

Returns data: 01 06 00 FF 00 01 78 3A The return frame is the same as the send frame.

01 address code, 06 function code, 00 FF register address, 00 01 means even parity, 78 3A crc16 parity.

0: No parity; 1: Even parity; 2: Odd parity.

11. Read return delay time:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	-----------------	-------------------------	-----------------	--------------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	-----------------	------------------------	----------	--------------

Broadcast address 0x01~0xFE

Function code 0x03

Register address: 0x00FC

Read number: 0x0001

For example:

send data(RS485 address is 1): 01 03 00 FC 00 01 44 3A

Returns data: 01 03 02 00 00 B8 44

01 address code, 03 function code, 02 length, 0000 means the return delay time is 0ms, B8 44 crc16 check.

12. Set the return delay time:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code: 0x06/0x10

Register address: 0x00FC(252)

Setting Content: 2 Bytes (0~1000)

For example, Set the data return delay to 200ms,

send data(RS485 address is 1): 01 06 00 FC 00 C8 48 6C

Returns data: 01 06 00 FC 00 C8 48 6C

13. Set current and voltage automatic reporting function:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2)
---	--------------	-------------------------	-----------------	----------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2)
---	--------------	------------------------	----------	----------

Function code: 0x06/0x10

Register address: 0x00FA

Setting Content: 2 Bytes (1-254)

For example, The current query function is to be changed to automatic reporting:
Automatically report in 1 second, send data(RS485 address is 1):01 06 00 FA 00 01 68 3B

Automatically report in 2 second, send data(RS485 address is 1):01 06 00 FA 00 02 28 3A

Automatically report in 3 second, send data(RS485 address is 1):01 06 00 FA 00 03 E9 FA

Automatically report in 4 second, send data(RS485 address is 1):01 06 00 FA 00 04 A8 38

Automatically report in 5 second, send data(RS485 address is 1):01 06 00 FA 00 05 69 F8

Automatically report in 10 second, send data(RS485 address is 1):01 06 00 FA 00 0A 29 FC

Disable reporting function: send data(RS485 address is 1):01 06 00 06 00 00 A9 FB
The return frame is the same as the send frame.

14. Restore factory settings:

Send data

RS485 address (Station address) (1)	Function (1)	Register address (2)	Read number (2)	CRC16(2))
---	--------------	-------------------------	-----------------	---------------

Returns data

RS485 address (Station address) (1)	Function (1)	Number of bytes (1)	data (n)	CRC16(2))
---	--------------	------------------------	----------	---------------

Function code: 0x06/0x10

Register address: 0x00FB(251)

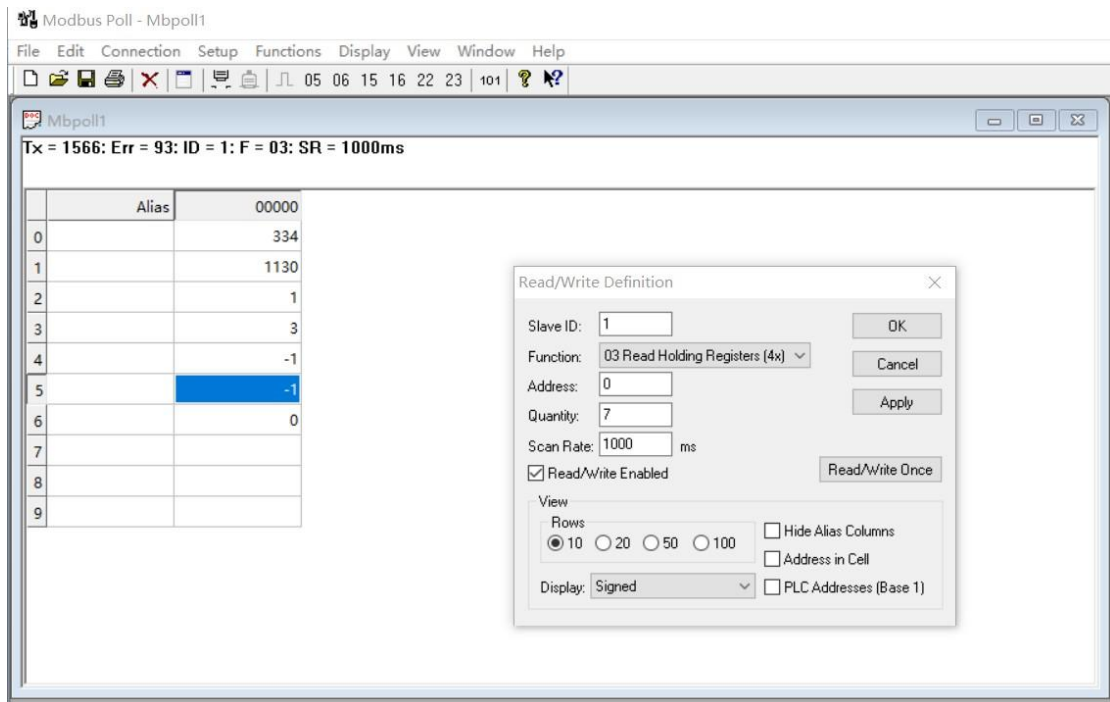
Setting Content: 2 Bytes (0)

send data(RS485 address is 1): FF 06 00 FB 00 00 ED E5

Returns data: FF 06 00 FB 00 00 ED E5

Hardware Reset: Short the RES jumper on the board for 5 seconds, then re-power the board.

MODBUS commands you can use "Modbus Poll" input, The value of 7 registers can be read as shown below (CRC check generated automatically)



You can also use HyperTerminal serial input, as shown below
(Manually add CRC check)



CRC check code(C51 MCU):

```
const unsigned char code auchCRCHi[256] = {
    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,
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 unsigned char code auchCRCLo[256] = {
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 int CRC_16(unsigned char *str, unsigned int usDataLen)
{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
    unsigned uIndex ; /* will index into CRC lookup table */
    while (usDataLen--){ /* pass through message buffer */
        {
            uIndex = uchCRCHi ^ *str++ ; /* calculate the CRC */
            uchCRCHi = uchCRCLo ^ uchCRCHi[uIndex];
            uchCRCLo = uchCRCLo[uIndex] ;
        }
    }
    return (uchCRCHi << 8 | uchCRCLo) ;
}

```