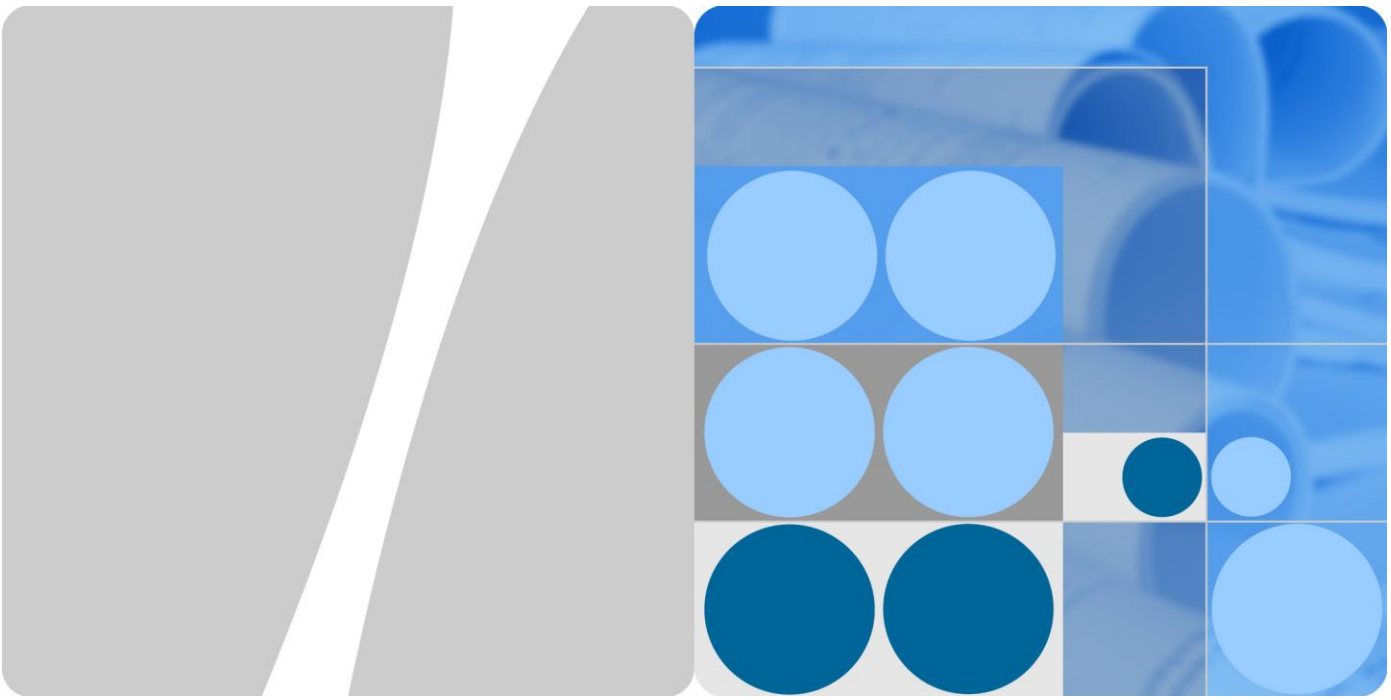


UPS2000-(1 kVA-3 kVA)

Modbus Protocol Development Guide

Issue 02

Date 2016-06-20

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://e.huawei.com

About This Document

Purpose

This document describes the Modbus protocol used between the RMS-MODBUS01B card, an optional component of the UPS2000-(1 kVA-3 kVA), or the UPS serial port (USB/RS232), and its host and provides examples of communication establishment processes.

Intended Audience

This document is intended for:

- Technical support engineers
- Maintenance engineers
- Product users

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol Conventions

Symbol	Description
DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Calls attention to important information, best practices

Symbol	Description
	and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all updates made in previous issues.

Issue 02 (2016-06-20)

This issue updated the collection and configuration table.

Issue 01 (2015-11-19)

This issue is the first official release.

Contents

1 Communication Protocol

1.1 Protocol Description

This section describes the Modbus protocol for command control and data exchange between the RMS-MODBUS01B cardor the UPS serial port (USB/RS232) and its host.

The functions stipulated in the protocol include:

1. The host obtains the UPS information by sending the read device identifier command.
2. The host obtains the specified UPS data by sending a read command.
3. The host configures relevant parameters and controls actions by sending a write command.

The host is the master node in the communication process. The information exchange is done by a question-and-answer method. The information and parameters of slave nodes use the target registers as storage addresses. The master node executes the read/write commands by accessing the registers. This protocol supports one master node networked with multiple slave nodes. The slave nodes are distinguished by address.

When the RMS-MODBUS01B card is used for communication, the address range is between 1 and 247. Different slave nodes correspond to different addresses. The slave nodes connecting to the same communication bus must not have the same address.

When the UPS serial port (USB/RS232) is used for communication, the communication protocol supports only one master node networked with one slave node. The slave node address is fixed to 1.

1.2 Terms

Master node: runs on the dedicated back-end server and is responsible for communication with slave nodes. It is the host.

Slave node: In this guide, it is the RMS-MODBUS01B card or the UPS serial port (USB/RS232). It collects information from the UPS power modules, and is used for responding to the master node.

RS485: indicates a serial communications standard which supports the half duplex serial short-range communication.

Read device identifier command: The command is sent from the master node to the slave nodes, and the slave nodes return the relevant UPS information.

Read command: The command is sent from the master node to the slave nodes, and the slave nodes return the relevant register content.

Write command: The relevant parameters are packed at the master node and sent to the slave nodes for configuration.

Register address: Every signal or parameter of the slave nodes corresponds to an address. The host obtains the relevant information or configures the relevant parameters. These processes are completed by accessing these register addresses.

Slave node address: This address is the RMS-MODBUS01B card address, set by the dual in-line package (DIP) switch. The range is between 1 and 247. The address of the UPS serial port (USB/RS232) is fixed to 1.

1.3 Physical Ports

1.3.1 Serial Communication Electrical Standards

The slave nodes communicate with the master node through the RS485 or RS232 ports.

Information transmission method:

1. Use the RTU transmission mode of the Modbus protocol.
2. The transmission mode for the character information is asynchronous mode using a frame format of one start bit, eight data bits (information byte), and one stop bit (10 bits in total).

1.3.2 Data Transmission Speed

The data transmission baud rate of the RMS-MODBUS01B card is 19200 bit/s by default and can be adjusted based on SW2.

NOTE
When connecting multiple UPSs to the network management system (NMS), use the RS485/232 converter with the isolation function. Do not hot swap the RS485/232 converter.

The data transmission baud rate of the UPS serial port (USB/RS232) is fixed to 9600 bit/s.

1.3.3 Cable Connection

1.3.3.1 RMS-MODBUS01B Card Communication Mode

Step 1 Set a hardware address for the Modbus card.

The hardware address for each card can be set using the SW1 DIP switch containing eight toggle switches, as shown in Figure 1-1. When you push toggle switch 1 downwards, the digit represents 1. If you push it upwards, the digit represents 0. The setting method is the same for the other seven toggle switches. You can set the DIP switch by using the binary method, as described in Table 1-1.

Figure 1-1 Address DIP switch

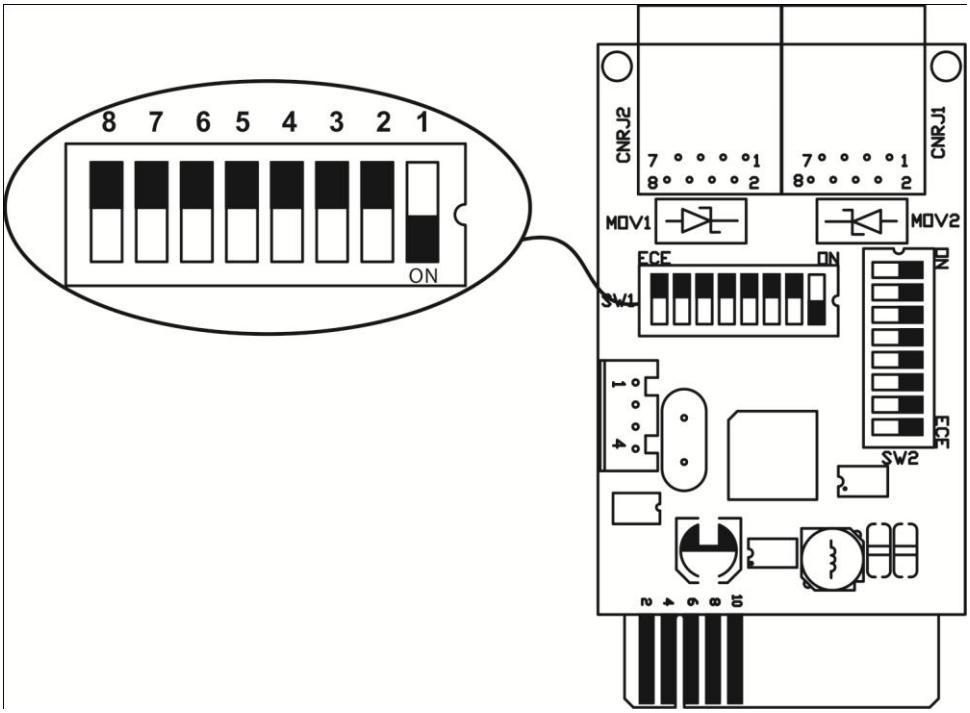


Table 1-1 Mapping between DIP switch settings and addresses (SW1)

ID	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	√	x	x	x	x	x	x	x
2	x	√	x	x	x	x	x	x
3	√	√	x	x	x	x	x	x
4	x	x	√	x	x	x	x	x
5	√	x	√	x	x	x	x	x
6	x	√	√	x	x	x	x	x
7	√	√	√	x	x	x	x	x
8	x	x	x	√	x	x	x	x
9	√	x	x	√	x	x	x	x
10	x	√	x	√	x	x	x	x
11	√	√	x	√	x	x	x	x
12	x	x	√	√	x	x	x	x
13	√	x	√	√	x	x	x	x
14	x	√	√	√	x	x	x	x
15	√	√	√	√	x	x	x	x

ID	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
16	x	x	x	x	√	x	x	x
17	√	x	x	x	√	x	x	x
18	x	√	x	x	√	x	x	x
19	√	√	x	x	√	x	x	x
20	x	x	√	x	√	x	x	x
21	√	x	√	x	√	x	x	x
22	x	√	√	x	√	x	x	x
23	√	√	√	x	√	x	x	x
24	x	x	x	√	√	x	x	x
25	√	x	x	√	√	x	x	x
26	x	√	x	√	√	x	x	x
27	√	√	x	√	√	x	x	x
28	x	x	√	√	√	x	x	x
29	√	x	√	√	√	x	x	x
30	x	√	√	√	√	x	x	x
31	√	√	√	√	√	x	x	x
32	x	x	x	x	x	√	x	x

NOTE

- √ means ON, x means OFF.
- Table 1-1provides the methods for setting addresses from 1 to 32. For details about how to set addresses from 33 to 247, see the binary DIP switch setting rules.

Step 2 Set a communications mode for the Modbus card over the SW2 DIP switch. The first four toggle switches from top to bottom are used to set the communications mode. Figure 1-2 shows the mapping between the toggle switches and the communications mode.

Figure 1-2 Communications DIP switch

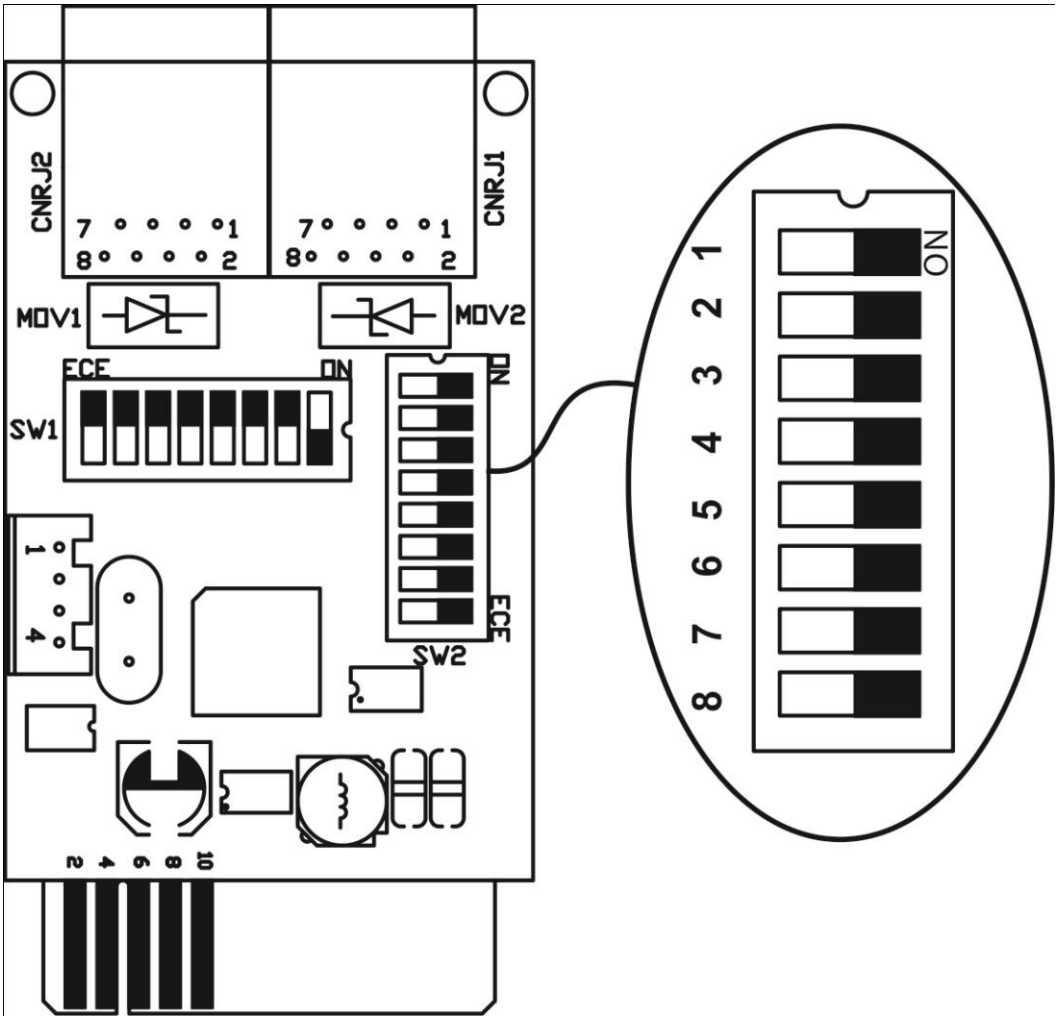


Table 1-2 Mapping between DIP switch settings and the communications mode

Function	Setting		Meaning
Baud rate	# 2	# 1	N/A
	OFF	OFF	2400 bit/s
	OFF	ON	4800 bit/s
	ON	OFF	9600 bit/s
	ON	ON	19200 bit/s (by default)
Parity check	# 4	# 3	N/A
	OFF	OFF	Even number parity
	ON	OFF	Odd number parity

Function	Setting		Meaning
	OFF	ON	No odd or even parity bit, stop bit 1 (by default)
	ON	ON	No odd or even parity bit, stop bit 2

Step 3 Set the RS-485 resistor type.

The RS-485 resistor type for the Modbus card is set by using the SW2 DIP switch. The sixth and seventh toggle switches from top to bottom are used to set the resistor type. Figure 1-3 lists the mapping between DIP switch settings and the resistor type.

Figure 1-3 Resistance DIP switch

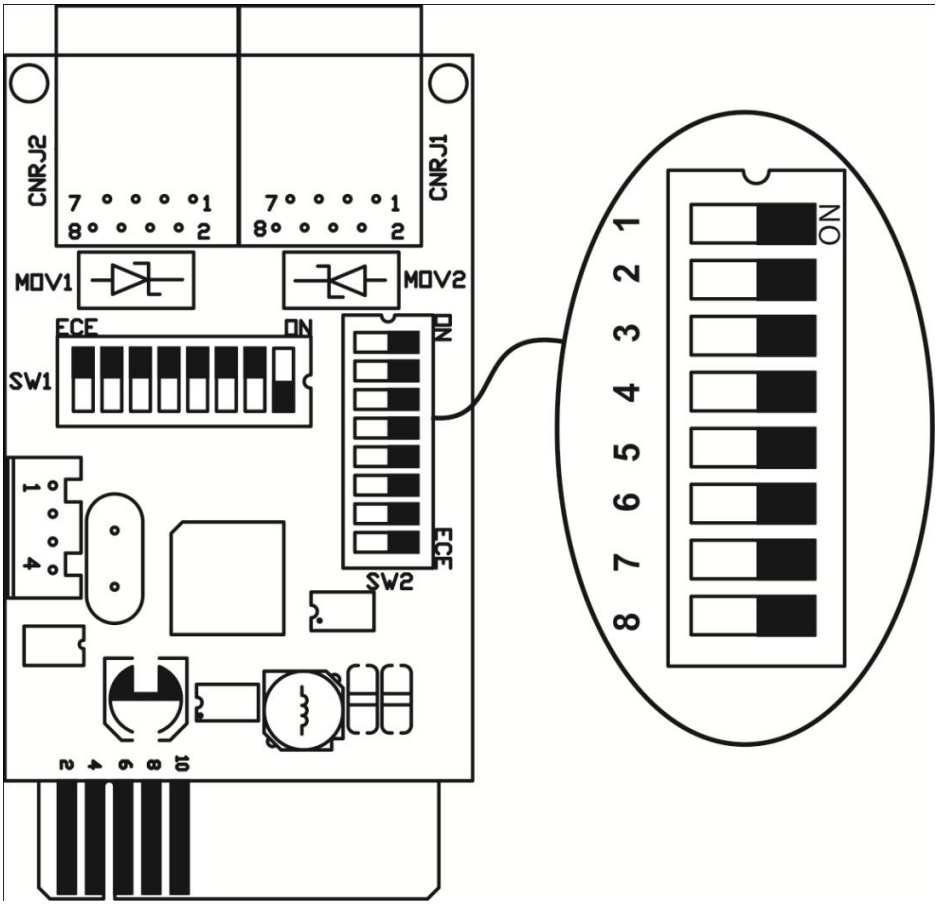


Table 1-3 RS-485 resistor

Function	Character #	Set to	Meaning
Pull-up resistor	# 6	ON	Enabled (by default)

Function	Character #	Set to	Meaning
		OFF	Disabled
Pull-down resistor	# 7	ON	Enabled (by default)
		OFF	Disabled
Terminal resistor	# 8	ON	Enabled (by default)
		OFF	Disabled

Step 4 Insert the Modbus card into the UPS smart slot and secure the screws.

Step 5 Connect the Modbus card to a computer using an RJ45 cable. You do not need to shut down the UPS.

Figure 1-4 RS-485 port pins

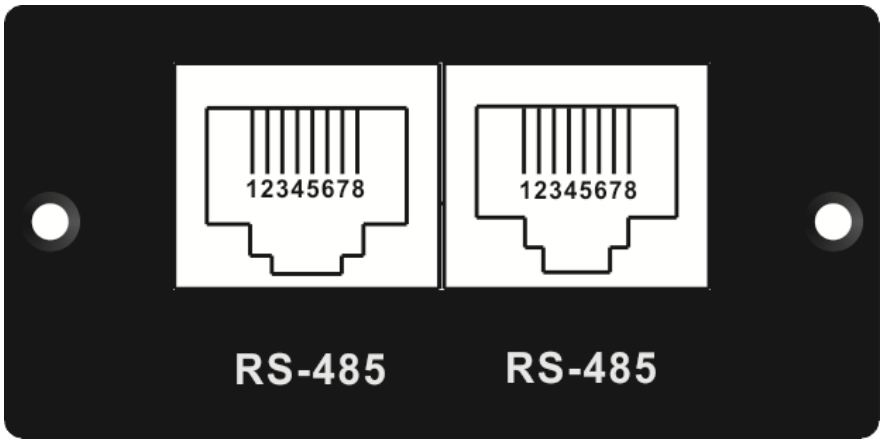
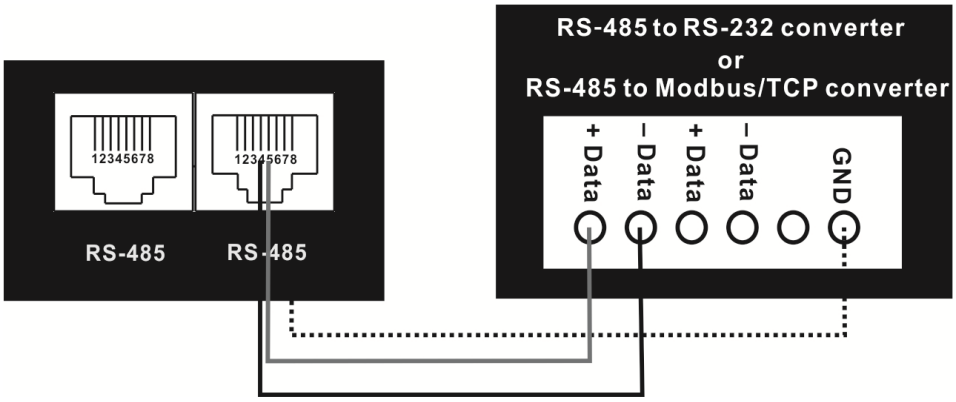


Table 1-4 RS-485 pin description

Pin	Function
4	RS-485 - B
5	RS-485 - A
8	Connects to the ground cable (GND)

Use an RS-485 to RS-232 converter or RS-485 to Modbus/TCP converter as a conversion media between the computer and Modbus card, and connect the conversion media to the RJ45 port based on the following wiring diagram.

Figure 1-5 Wiring diagram for connecting the Modbus card to the converter



Connect the Modbus card to the computer based on the following wiring diagrams.

Figure 1-6 Wiring diagram for connecting the Modbus card to the computer (Using an RS-485 to RS-232 converter)

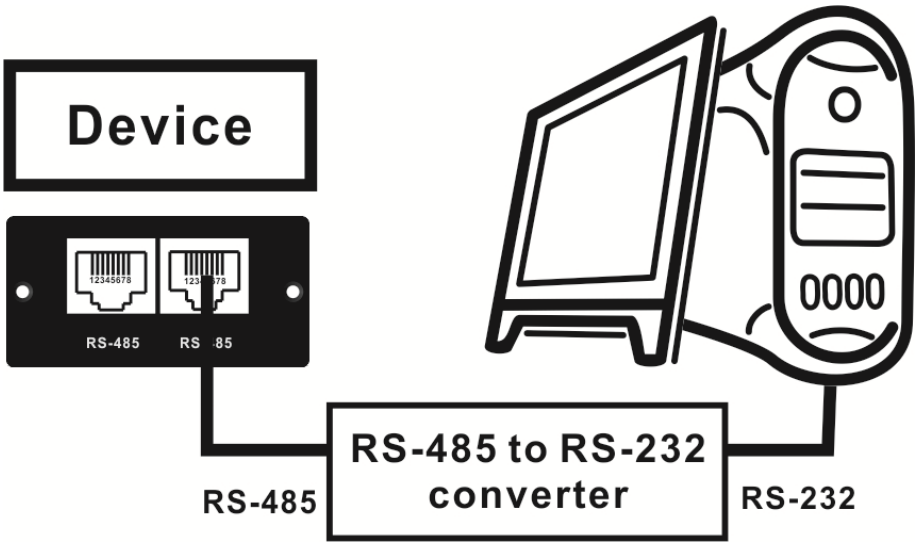
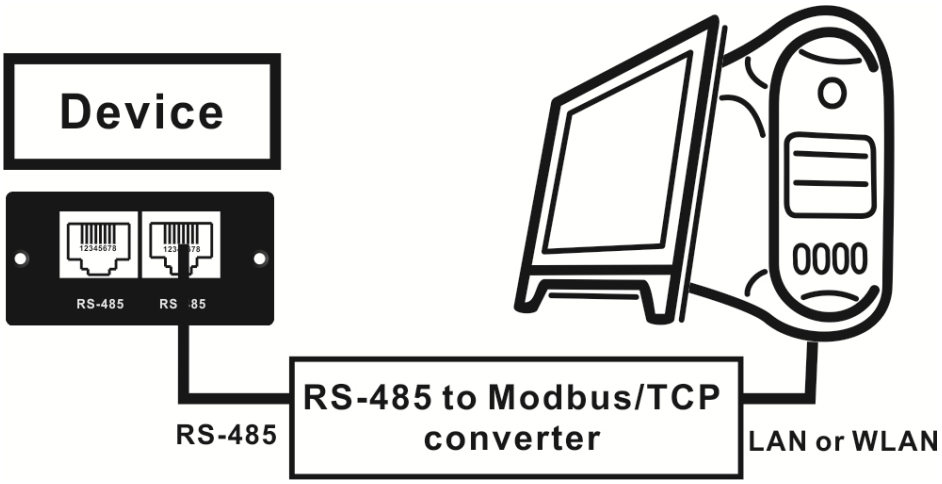


Figure 1-7 Wiring diagram for connecting the Modbus card to the computer (Using an RS-485 to Modbus/TCP converter)



----End

1.3.3.2 UPS Serial Port (USB/RS232) Communication Mode

A PC and a UPS can communicate through the serial port (USB or RS232). The UPS address is fixed to 1. You can use only either USB or RS232. If you use the USB connection mode, install the device driver software on the PC by referring to 1.3.3.3 Installing the Driver Software.

Figure 1-8 shows how to connect the USB port to a PC.

Figure 1-8 USB connection mode

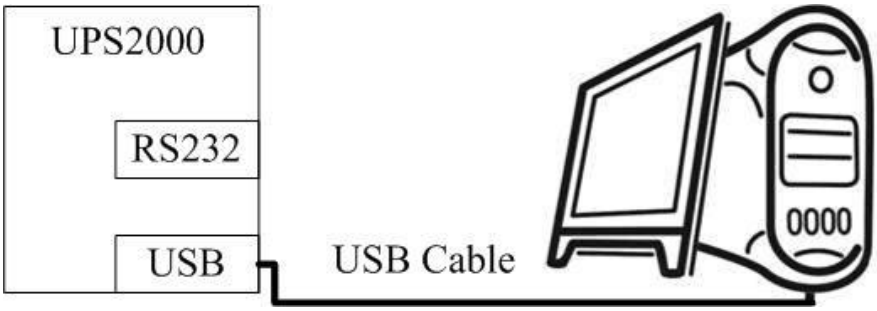
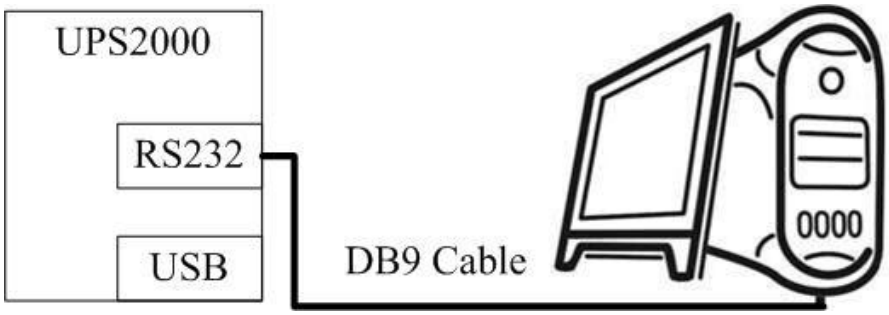


Figure 1-9 shows how to connect the RS232 port to a PC.

Figure 1-9 RS232 connection mode



1.3.3.3 Installing the Driver Software

If the serial port is used for the upgrade, no driver needs to be installed. In this case, skip this section. If the USB port is used for the upgrade, install the USB driver by performing the following steps**.**

After powering on the UPS, connect the UPS and a PC using a USB cable (see section "1.3.3.2 UPS Serial Port (USB/RS232) Communication Mode" for details). If the PC connects to the UPS for the first time, the PC identifies new hardware and automatically searches the device driver. After the search, the message "Device driver software was not successfully installed" is displayed. Click **Close**. Then, install the driver software in the device manager through the control panel.

Install the driver software in the same manner as that for common hardware. The following uses Windows 7 as an example to show how to install the driver software. Install the driver software in Windows XP in the same manner as that for Windows 7. The following shows the detailed procedure:

Step 1 Click **Close** in the **Driver Software Installation** dialog box. In the **Device Manager** pane, right-click **Unknown device**, and choose **Upgrade Driver Software**.

Figure 1-10 Message indicating driver software installation failure

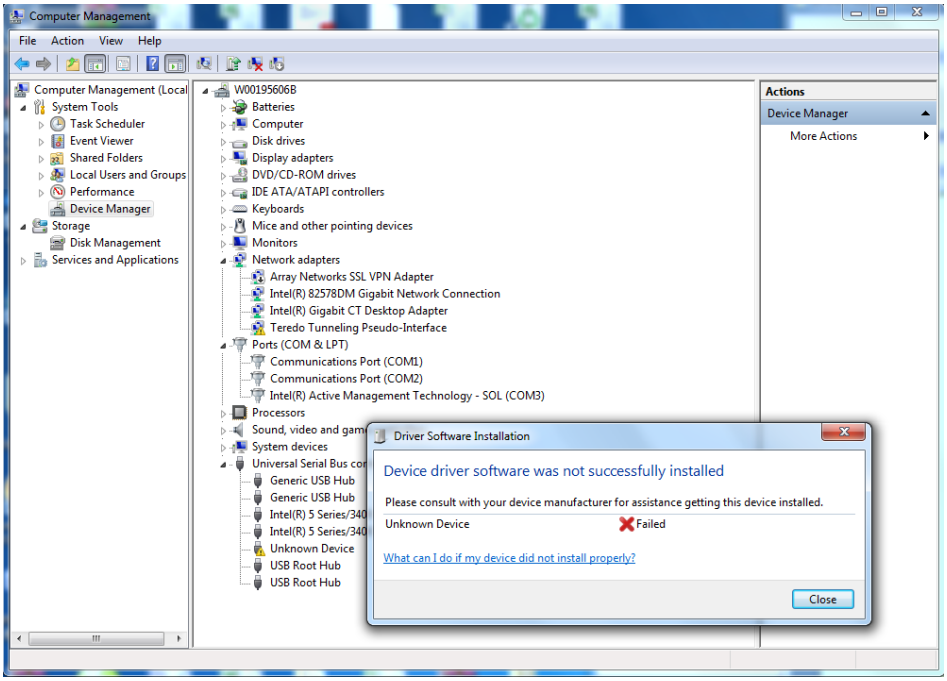
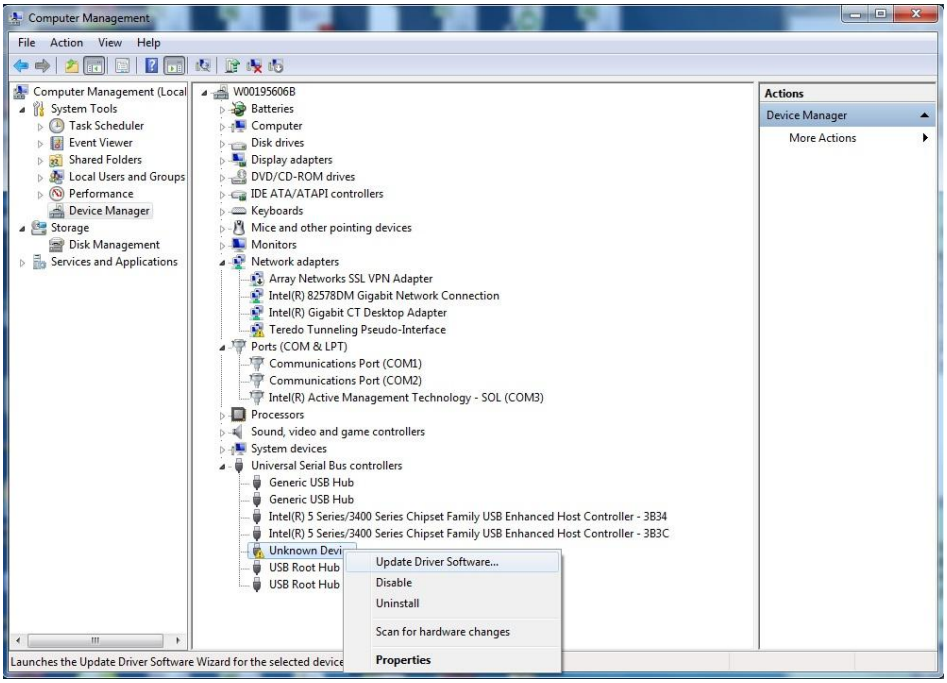
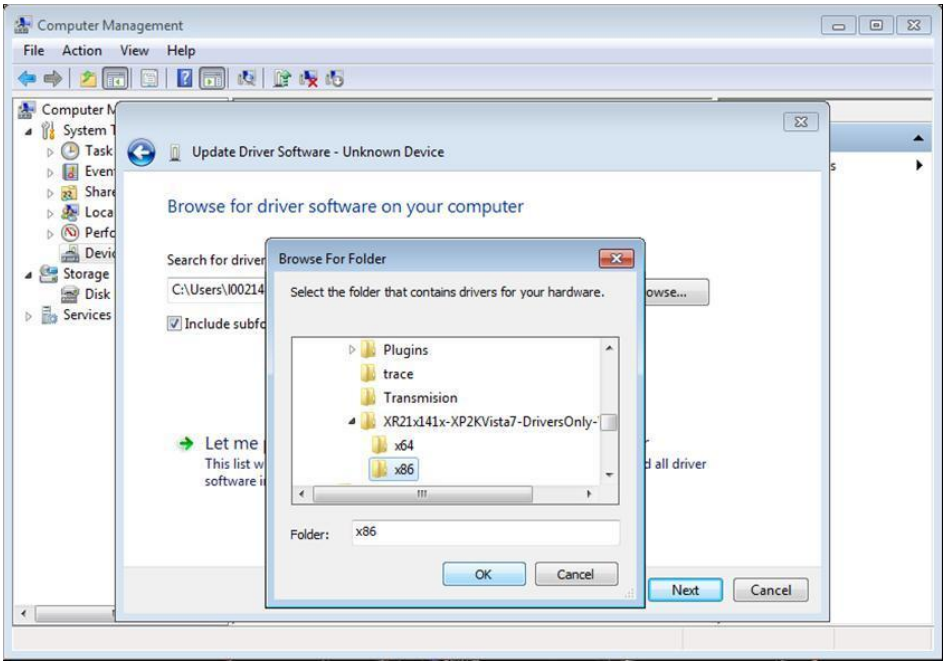


Figure 1-11 Installing the driver software



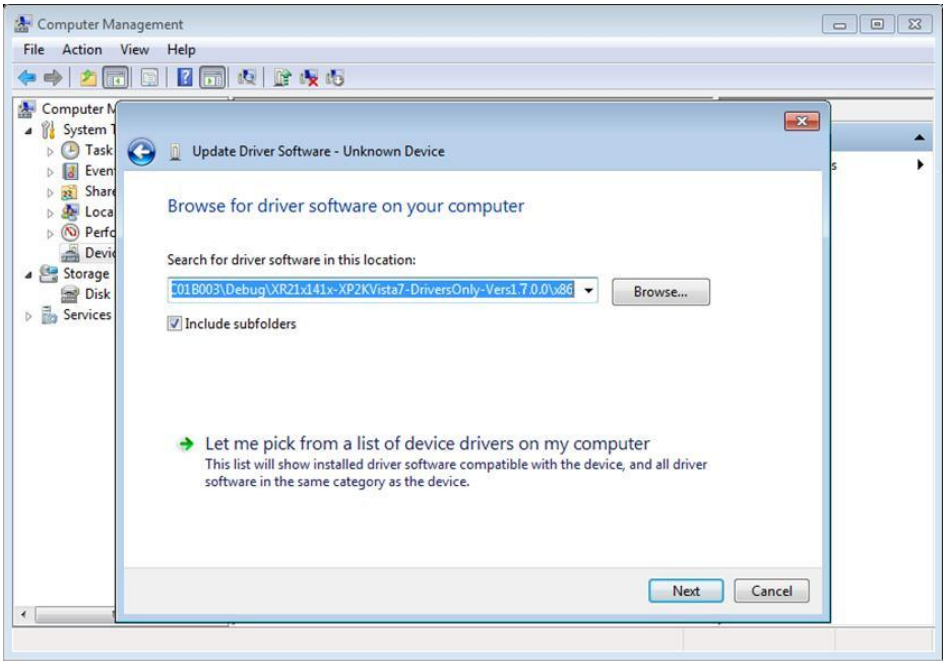
Step 2 In the displayed dialog box, select **Browse for driver software on your computer**, and click **Browse**. In the displayed dialog box, expand XR21x141x-XP2KVista7-DriversOnly-Vers1.7.0.0 (decompress the file in the folder), select **x86** (for a 32-bit system) or **x64** (for a 64-bit system) based on PC configurations, and then click **OK**.

Figure 1-12 Selecting driver software



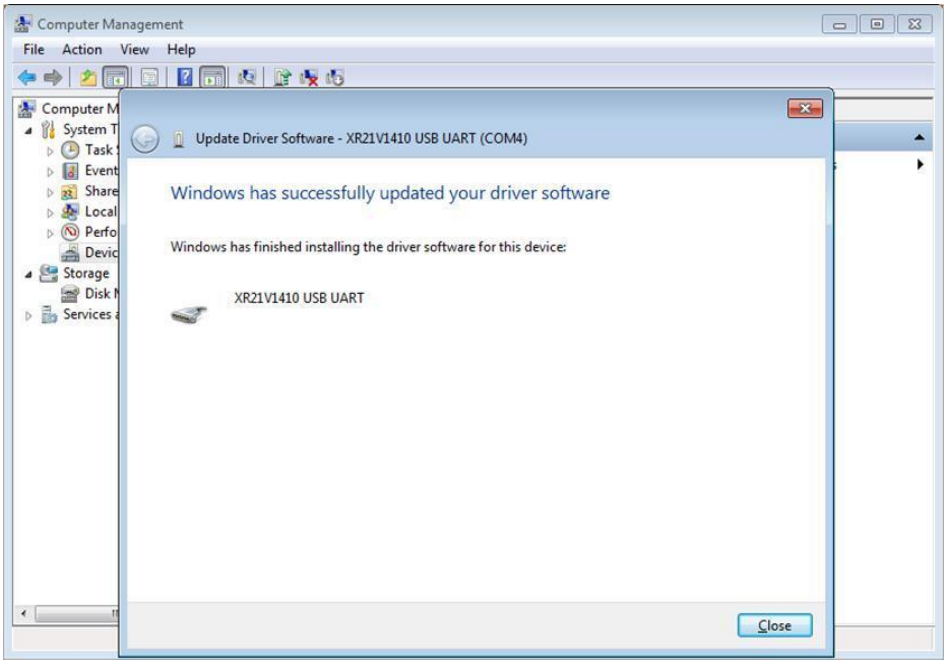
Step 3 Click **Next** to install the driver software.

Figure 1-13 Installing driver software



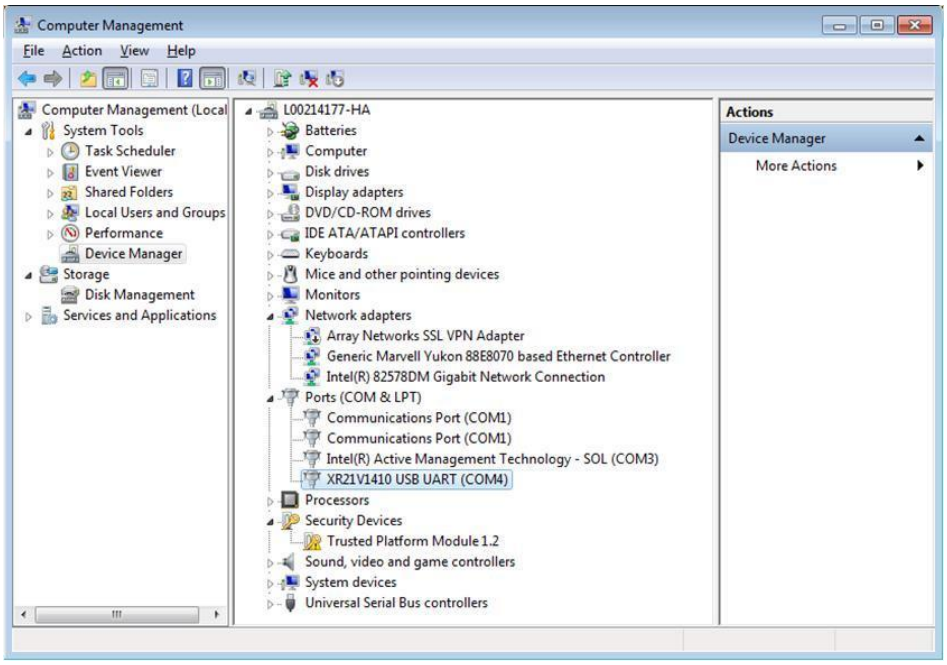
Step 4 After the software is installed, a dialog box is displayed, as shown in the following figure.

Figure 1-14 Driver software installation completed



Step 5 Click **Close**. **XR21V140 USB UART** is displayed under **Ports (COM & LPT)** of the device manager.

Figure 1-15 Driver software installation succeeded



----End

1.4 Communication Mode at the Physical Layer

After the slave nodes are powered on or reset and run steadily, the slave nodes respond to the read or write commands. Upon receiving a command, the slave nodes return the requested information to the master node under normal conditions. The slave nodes return the specific error codes corresponding to the error types under abnormal conditions.

The error code definitions are listed in Table 1-5.

Table 1-5 Error code list

Code	Name	Description
0x01	Illegal function code	Slave nodes received unsupported or incorrect function codes.
0x02	Illegal parameter	Slave nodes received invalid function code parameters.
0x03	Illegal data	Slave nodes received invalid data such as CRC checking error.
0x04	Slave node execution failed	Slave nodes failed to respond to a command.
0x05	Reload the start command	Slave nodes are busy.

1.5 Command Types and Formats at the Application Layer

1.5.1 Function Code List

Table 1-6 Function code list

Function Code	Meaning	Remarks
0x03	Read register command	Continuously read a single register or multiple registers.
0x06	Write single register command	Supports writing into a single register.
0x2B	Read device identifier command	N/A

1.5.2 CRC Checking Algorithm

```
unsigned short count CRC(unsigned char *addr, int num)
{
```

```
   unsigned short CRC = 0xFFFF;
   int i;
   while (num--)
   {
     CRC ^= *addr++;
       for (i = 0; i < 8; i++)
       {
           if (CRC & 1)
           {
              CRC >>= 1;
              CRC ^= 0xA001;
           }
           else
           {
              CRC >>= 1;
         }
       }
   }
   return CRC;
}
```

1.5.3 Definition and Format of the Read Device Identifier Command

Frame Format

Command frames:

Table 1-7 Frame format of the read device identifier command

Byte	0	1	2	3	4	5	6
Field	ADDR	CMD	MEI	ReadDevID	Target ID	LSB	MSB
Description	Controller address	Function code	MEI type	Read device ID	Object ID	CRC checking	

Response frames:

The response frames are not fixed and depend on the object ID.

Command Definition

This command code allows reading identifiers and added packets that are relevant to the physical and functional description of the remote devices.

Simulate the port of the read device identifier as an address space. This address space consists of a set of addressable data elements. The data elements are objects to be read, and the object IDs determine these data elements.

A data element consists of three objects:

- Basic device identifier: All objects of this type are mandatory, such as the manufacturer name, product code, and revision version.
- Normal device identifier: Except the basic data objects, the device provides additional and optional identifiers and data object description. Normal device identifiers define all types of objects based on standard definitions, but the execution of this type of objects is optional.
- Extended device identifier: Except the normal data objects, the device provides additional and optional identifiers and special data object description. All these data objects are related to the device.

Table 1-8 Command definition

Object ID	Object Name or Description	Type	Mandatory or Optional (M/O)	Category
0x00	Manufacturer name	ASCII character string	M	Basic
0x01	Product code	ASCII character string	M	
0x02	Main revision	ASCII character string	M	
0x03–0x7F	N/A	N/A	N/A	Normal
0x80–0xFF	N/A	N/A	N/A	Extended

Table 1-9 Request PDU

Function Code	1 byte	0x2B
MEI Type	1 byte	0x0E
ReadDevID Code	1 byte	0x01/0x02/0x03/0x04
Object ID	1 byte	0x00 to 0xFF

Request parameter description:

- Function code: 43 (decimal), 0x2B (hexadecimal)
- MEI type: the MEI type of 14 (0x0E) assigned to the device identifier port
- The ReadDevID has four access types:
  - 0x01: request to obtain the basic device identifier (stream access)
  - 0x02: request to obtain the normal device identifier (stream access)
  - 0x03: request to obtain the extended device identifier (stream access)
  - 0x04: request to obtain the special device identifier (special access)

If the identifier data is not suitable for individual responses, you can require several request and response transactions. The object ID byte provides the first identifier code obtained. For

the first transaction, the client must set the object ID to 0 in order to obtain the start bit of the device identifier data. For the subsequent transactions, the client server must set the object ID to the value returned by the previous responded server.

If the object ID does not match any known object, the server points to the object numbered 0 (from the start).

For single access, the ReadDevID code is 04. The object ID in the requesting process provides the obtained object identifier code.

If the object ID does not match any known object, the server returns an abnormal code numbered 02 (illegal data address).

Table 1-10 Response PDU

Function Code	1 byte	0x2B
MEI Type	1 byte	0x0E
ReadDevID Code	1 byte	0x01/0x02/0x03/0x04
Consistency Level	1 byte	N/A
More	1 byte	0x00/0xFF
Next Object ID	1 byte	Object ID
Object Number	1 byte	N/A
Object ID List	1 byte	N/A
Object Length	1 byte	N/A
Object Value	1 byte	N/A

Response parameter description:

- Function code: 43 (decimal), 0x2B (hexadecimal)
- MEI type: the MEI type of 14 (0x0E) assigned to the device identifier port
- ReadDevID code: same as the request ReadDevID code, that is 01, 02, 03, or 04
- Consistency level: the consistency level for the device identifiers and types of supported accesses

  01: basic identifier code (stream access only)

  02: normal identifier code (stream access only)

  03: extended identifier code (stream access only)

  81: basic identifier code (stream access and single access)

  82: normal identifier code (stream access and single access)

  83: extended identifier code (stream access and single access)
- More: If the ReadDevID code is 01, 02, or 03 (stream access) and the identifier data does not match the single response, you can require several request and response transactions.

  00: The object is no longer usable

  FF: Other identifier objects are usable, and more Modbus transactions are required

If the ReadDevID code is 04 (single access), this field must be set to 00.

- Next object ID: If more is FF, require the next object identifier code. If more is 00, set to 00 (unused).
- Object number: returned object identifier number in the response process.

  For single access, the object number = 1.
- Object 0.ID: the first object identifier code (stream access) or requested object identifier code (single access) returned by the PDU.
- Object 0.length: the byte length of the first object.
- Object 0.value: the value of the first object (the byte of the object 0.length).

  …
- Object N.ID: the identifier code of the last object in the response process.
- Object N.length: the byte length of the last object.
- Object N.value: the value of the last object (the byte of the object N.length).

Table 1-11 Abnormal response PDU

Function Code	1 byte	0xAB
Exception Code	2 bytes	N/A

Frame Format Examples

The following are some request examples of the read device identifiers of the basic device identifiers:

NOTE
In this example, all objects are returned to one response PDU.

Table 1-12 Example read device identifier requests

Request		**Response**	
Domain Name	Value	Domain Name	Value
Function	0x2B	Function	0x2B
MEI Type	0x0E	MEI Type	0x0E
ReadDevID Code	0x01	ReadDevID Code	0x01
Object ID	0x00	Consistency Level	0x01
		More	0x00
		Next Object ID	0x00
		Object Number	0x03
		Object ID	0x00
		Object Length	0x16
		Object Value	"Company

Request		Response	
			identifier"
		Object ID	0x01
		Object Length	0x0A
		Object Value	"Product code"
		Object ID	0x02
		Object Length	0x05
		Object Value	"V2.11"

If one device needs several transactions to process responses, execute the following transactions.

Table 1-13 First transaction

Request		Response	
Domain Name	Value	Domain Name	Value
Function	0x2B	Function	0x2B
MEI Type	0x0E	MEI Type	0x0E
ReadDevID Code	0x01	ReadDevID Code	0x01
Object ID	0x00	Consistency Level	0x01
		More	0xFF
		Next Object ID	0x02
		Object Number	0x03
		Object ID	0x00
		Object Length	0x16
		Object Value	"Company identifier"
		Object ID	0x01
		Object Length	0x1A
		Object Value	"Product code"

Table 1-14 Second transaction

Request		Response	
Domain Name	Value	Domain Name	Value

Request		Response	
Function	0x2B	Function	0x2B
MEI Type	0x0E	MEI Type	0x0E
ReadDevID Code	0x01	ReadDevID Code	0x01
Object ID	0x02	Consistency Level	0x01
		More	0x00
		Next Object ID	0x00
		Object Number	0x03
		Object ID	0x02
		Object Length	0x05
		Object Value	"V2.11"

Command for Querying Device Identifiers

- Request Frame Format

Table 1-15 Frame format of the query device identifier request

Function Code	1 byte	0x2B
MEI Type	1 byte	0x0E
ReadDevID Code	1 byte	0x01
Object ID	1 byte	0x00

- Response Frame Format

Table 1-16 Frame format of the response to the query device identifier request

Function Code			1 byte	0x2B
MEI Type			1 byte	0x0E
ReadDevID Code			1 byte	0x01
Consistency Level			1 byte	0x01
More			1 byte	N/A
Next Object ID			1 byte	N/A
Number of Objects			1 byte	N/A
Object List	First Object	Object ID	1 byte	0x00

		Object Length	1 byte	N
		Object Value	N bytes	N/A
	…	…	…	…

Table 1-17 Object IDs mapping to the query device identifier command

Object ID	Object Name or Description	Description	Category
0x00	Manufacturer name	"HUAWEI"	Basic
0x01	Product code	"UPS5000" "UPS2000" "UPS2000A" "SUN2000"	
0x02	Main version	ASCII character string, software version	

Command for Querying a Device List

- Request Frame Format

Table 1-18 Frame format of the query device list request

Function Code	1 byte	0x2B
MEI Type	1 byte	0x0E
ReadDevID Code	1 byte	0x03
Object ID	1 byte	0x87

- Response Frame Format

Table 1-19 Frame format of the response to the query device list request

Function Code	1 byte	0x2B
MEI Type	1 byte	0x0E
ReadDevID Code	1 byte	0x03
Consistency Level	1 byte	0x03
More	1 byte	N/A
Next Object ID	1 byte	N/A

Number of Objects			1 byte	N/A
Object List	First Object	Object ID	1 byte	0x87
		Object Length	1 byte	N
		Object Value	N byte	N/A
	…	…	…	…

This command corresponds to the object IDs in the following table.

Table 1-20 Object IDs mapping to the query device list command

Object ID	Object Name	Type	Description
0x80–0x86	Reserved	N/A	Null object with a returned object length 0.
0x87	Number of devices	int	Number of devices connected to this RS485 address.
0x88	First device information	ASCII character string	For the network element type that supports only one device at one RS485 address, only the first device description is returned.
0x89	Second device information	N/A	N/A
…	…	…	…
0xFF	120th device information	N/A	N/A

The definitions of the device description are:

Each device description consists of all "attribute = value" strings.

Attribute label = %s; attribute label = %s; …; attribute label = %s

For example: 1 = UPS5000, 2 = V100R001C01SPC120, 3 = P1.02-D1.0, 4 = 123232323, 5 = 2, 6 = 1.

Table 1-21 Attribute definitions

Attribute Label	Attribute Name	Type	Description
1	Device model	ASCII character	UPS5000

Attribute Label	Attribute Name	Type	Description
		string	UPS2000 UPS2000A SUN2000
2	Software version	ASCII character string	N/A
3	Version of the interface protocol	ASCII character string	N/A
4	ESN	ASCII character string	N/A
5	Device number	int	0, 1, 2, 3, … (assigned by network element)
6	Parallel networking number	int	0, 1, 2, 3, … (assigned by network element) -1: invalid value, indicates that a device does not belong to any parallel system If not applicable, this attribute is not returned.

1.5.4 Definition and Format of the Read Command

Read Command Definition

This command can be used to continuously read a single register or multiple registers.

Read Command Format

- Request Frame Format

Table 1-22 Frame format of the read request

Byte	0	1	2	3	4	5	6	7
Field	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Description	Controller address	Command type	Register starting address		Number of registers n		CRC checking	

NOTE

For example, the request frame is 01 03 2A F8 00 01 0D E3.

01 is the slave node address. 03 is the read command. 2AF8 is the register address. 0001 indicates the number of registers to be read is 1. 0DE3 is the CRC checking value.

- Response Frame Format

Table 1-23 Frame format of the response to the read request

Byte	0	1	2	3	4	5	6	…	L+1	L+2	L+3	L+4
Field	ADDR	CMD	Length	MSB	LSB	MSB	LSB	…	MSB	LSB	LSB	MSB
Description	Controller address	Command type	Data length L = n x 2	First register value		Second register value		…	Last register value		CRC checking	

NOTE

For example, the response frame is 01 03 02 00 01 79 84.

01 is the slave node address. 03 is the read command. 02 is the length of data to be read. 0001 indicates the first register value to be read. 7984 is the CRC checking value.

1.5.5 Definition and Format of the Write Single Register Command

Definition of the Write Single Register Command

This command supports writing into a single register.

Format of the Write Single Register Command

- Request Frame Format

Table 1-24 Frame format of the write single register request

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Field	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Description	Controller address	Command type	Register address		Data		CRC checking	

NOTE

For example, the request frame is 01 06 2B 15 00 01 50 2A.

01 is the slave node address. 06 is the write single register command. 2B15 is the register address. 0001 is the setting value. 502A is the CRC checking value.

- Response Frame Format

Table 1-25 Frame format of the response to the write single register request

Byte	0	1	2	3	4	5	6	7
Field	ADDR	CMD	MSB	LSB	MSB	LSB	LSB	MSB
Description	Controller address	Command type	Register address		Data		CRC checking	

NOTE

The formats of the request and response frames are the same for the write single register command.

For example, the response frame is 01 06 2B 15 00 01 50 2A.

01 is the slave node address. 06 is the write single register command. 2B15 is the register address. 0001 is the setting value. 502A is the CRC checking value.

2 Signal List

2.1 Collection and Configuration

2.1.1 Signal List Description

The signal names are the meanings of the signals.

The parameter types are the internal storage states of the signals in the RMS-MODBUS01B card.

The gains are used to ensure that all data is transmitted in the format of unsigned short integers during Modbus communication, thereby avoiding transmission of complex data in Modbus frames.

For the R/W attributes, R is read only, W is write only, and RW is read and write.

For the register addresses, N is the UPS number fixed at 1.

NOTE

For example, the register address of the UPS input voltage is 11000, which is 0x2AF8 in hexadecimal format.

2.1.2 Collection Signal List

Table 2-1 Collection signal list

Name	Parameter Type	Gain	Unit or Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
Input voltage	Floating point	10	V	N1000	1	R
Input frequency	Floating point	10	Hz	N1003	1	R
Bypass voltage	Floating point	10	V	N1004	1	R

Name	Parameter Type	Gain	Unit or Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
Bypass frequency	Floating point	10	Hz	N1007	1	R
Output voltage	Floating point	10	V	N1008	1	R
Output current	Floating point	10	A	N1011	1	R
Output frequency	Floating point	10	Hz	N1014	1	R
Output active power	Floating point	10	kW	N1015	1	R
Output apparent power	Floating point	10	kVA	N1018	1	R
Load ratio	Floating point	10	%	N1021	1	R
Power supply mode	Unsigned short integer (16 bits)	1	0: no power supplied 1: bypass mode 2: mains mode 3: battery mode 5: mains ECO	N1024	1	R
Input system	Unsigned short integer (16 bits)	1	0: single-phase	N1025	1	R
Output system	Unsigned short integer (16 bits)	1	0: single-phase	N1026	1	R
Temperature inside the UPS	Floating point	10	℃	N1027	1	R
UPS status	Unsigned short integer	1	Bit definition Other bits: reserved (filled with 0) 7: mains abnormal (0: normal, 1:	N1043	1	R

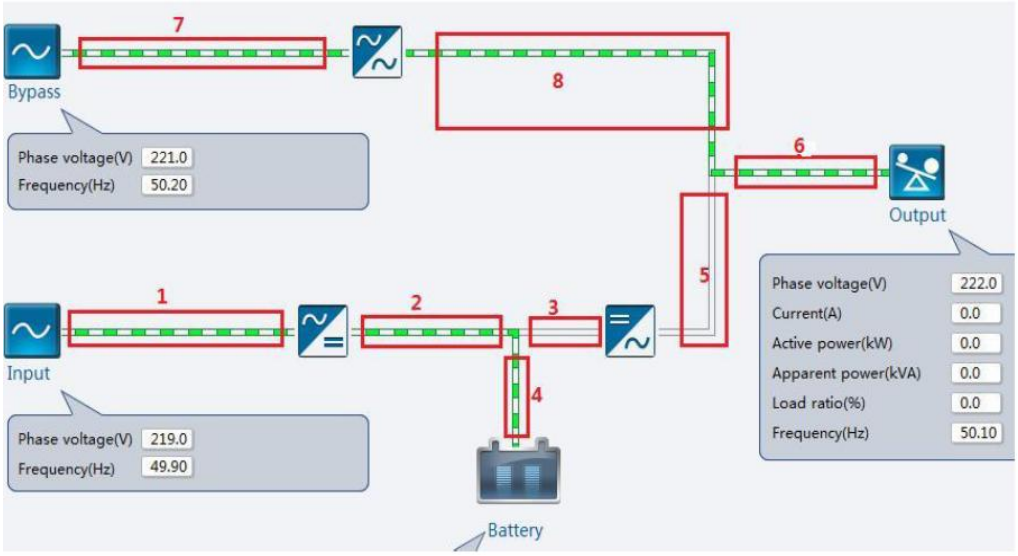
Name	Parameter Type	Gain	Unit or Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
			abnormal) 6: low battery voltage (0: normal, 1: low voltage) 5: reserved 4: UPS faulty (0: normal, 1: faulty) 3: UPS type (0: offline, 1: online) 2: battery self-check status (0: not in self-check, 1: self-checking) 1: reserved 0: reserved			
Battery voltage	Floating point	10	V	N2000	1	R
Battery status	Unsigned short integer (16 bits)	1	2: hibernating 3: float charging 4: equalized charging 5: discharging	N2002	1	R
Remaining battery capacity	Unsigned short integer (16 bits)	1	%	N2003	1	R
Battery backup time	Unsigned integer (32 bits)	1	s	N2004	2	R
Number of batteries	Unsigned short integer (16 bits)	1	[1, 20]	N2007	1	R
Battery capacity	Unsigned short integer (16 bits)	1	AH	N2033	1	R
Energy flow diagram	Unsigned short integer (16	1	00: hollow 01: solid	N9006 (bit 1,	1	R

Name	Parameter Type	Gain	Unit or Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
segment 1	bits)		10: flow to the right	0)		
Energy flow diagram segment 2	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 3, 2)	1	R
Energy flow diagram segment 3	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 5, 4)	1	R
Energy flow diagram segment 4	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow downwards 11: flow upwards	N9006 (bit 7, 6)	1	R
Energy flow diagram segment 5	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 9, 8)	1	R
Energy flow diagram segment 6	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 11, 10)	1	R
Energy flow diagram segment 7	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 13, 12)	1	R
Energy flow diagram segment 8	Unsigned short integer (16 bits)	1	00: hollow 01: solid 10: flow to the right	N9006 (bit 15, 14)	1	R
UPS model	Float	10	KVA	N9009	1	R

NOTE

The relationship between the serial numbers of state registers and the energy flow diagram is shown in the following figure.

Figure 2-1 Energy flow diagram



NOTE

- The user can use 0 as the UPS number. That is, the user can directly use the basic register address to query the corresponding value.
- Some registers may return the following invalid values:

0x7FFF: invalid value of the floating point type returned by one register

0xFFFF: invalid value of a type other than the floating point type returned by one register

0xFFFFFFFF: invalid value returned by two registers

2.1.3 Configuration and Control Signal List

Table 2-2 Configuration and control signal list

Signal Name	Parameter Type	Gain	Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
Startup state	Unsigned short integer (16 bits)	1	00: shutdown (you can start) 10: startup failed (you can start) 11: startup completed (you can shut down)	N1028	1	R
Startup	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N1029	1	W

Signal Name	Parameter Type	Gain	Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
Shutdown	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N1030	1	W
Single UPS ECO	Unsigned short integer (16 bits)	1	0: disabled (default) 1: enabled	N1031	1	R
Auto start	Unsigned short integer (16 bits)	1	0: disabled (default) 1: enabled	N1044	1	W/R
Bypass output	Unsigned short integer (16 bits)	1	0: disabled (default) 1: enabled	N1045	1	W/R
Buzzer off	Unsigned short integer (16 bits)	1	0: disabled (default) 1: enabled	N1046	1	W/R
Delay shutdown and restart	Floating point	10	0.1–99.0 min Step: 0.1 min	N1047	1	W/R
	Unsigned short integer (16 bits)	1	1–9999 min Step: 1 min	N1048	1	W/R
Delay shutdown	Floating point	10	0.1–99.0 min Step: 0.1 min	N1049	1	W/R
Cancel delayed shutdown	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N1050	1	W
Low battery voltage detected during battery self-check	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N2021	1	W

Signal Name	Parameter Type	Gain	Range	Register Address	Register Address Length (Number of Bits)	R/W Attribute
Stop battery self-check	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N2023	1	W
Battery short-time self-check	Unsigned short integer (16 bits)	1	The register value is fixed at 1.	N2028	1	W

2.2 Alarms

2.2.1 Signal List Description

The definitions of the UPS alarm register addresses comply with the following rules:

Register address: basic register address + UPS number x 1024

The UPS number is fixed at 1.

NOTE

- For example, the register address of the battery overvoltage alarm is 40163 + 1024 = 41187 = 0xA0E3 (hexadecimal).
- The user can use 0 as the UPS number. That is, the user can directly use the basic register address to query the corresponding value.

If the query is across registers, and a register is not defined in this table, the returned register value is 0.

2.2.2 Alarm Signal List

Table 2-3 Alarm signal list

Alarm ID	Alarm Cause ID	Alarm	Basic Register Address	Bit Value
0030	1	UPS internal overtemperature	40156	3
0010	1	Abnormal bypass voltage	40161	1
0010	2	Abnormal bypass voltage	40161	2

Alarm ID	Alarm Cause ID	Alarm	Basic Register Address	Bit Value
0025	1	Battery overvoltage	40163	3
0029	1	Battery maintenance notification	40164	1
0026	1	Battery undervoltage	40164	3
0022	1	Battery disconnected	40170	4
0066	1	Output overload	40173	5
0014	1	Startup timeout	40174	0
0066	2	Output overload	40174	3
0042	15	Rectifier fault/internal fault	40179	14
0042	17	Rectifier fault/internal fault	40179	15
0042	18	Rectifier fault/internal fault	40180	1
0042	24	Inverter fault/internal fault	40180	5
0042	27	Inverter fault/internal fault	40180	6
0042	28	Inverter fault/internal fault	40180	7
0042	31	Inverter fault/internal fault	40180	10
0042	32	Inverter fault/internal fault	40180	11
0042	36	Charger alarm/internal fault	40180	13
0042	42	Charger	40182	4

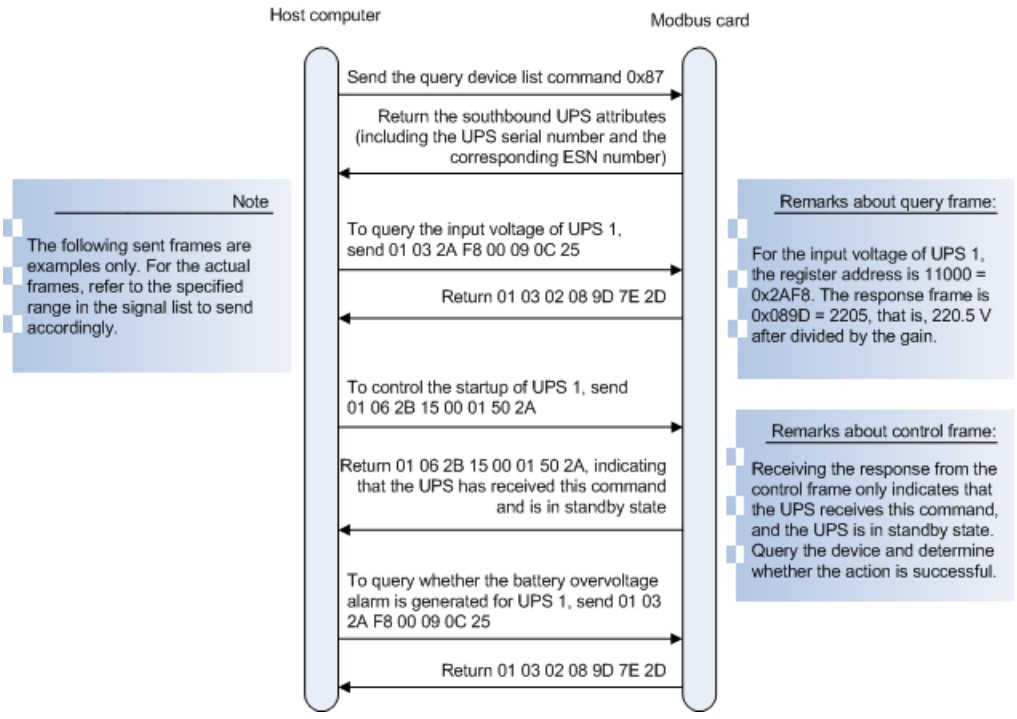
Alarm ID	Alarm Cause ID	Alarm	Basic Register Address	Bit Value
		alarm/internal fault		
0066	3	Output overload	40182	13
0066	4	Output overload	40182	14

3 Examples of the Host Communication Process

Based on the command types of the communication protocols, you can identify the register addresses of every UPS semaphore, and the UPS serial number offset is needed. The serial number of the southbound UPS connected to the RMS-MODBUS01B card is fixed at 1.

The following diagram shows a communication process for reference.

Figure 3-1 Communication process of the host computer and the Modbus card



A. Send a query device list command:

Query frame: 01 2B 0E 03 87 31 75

Response frame (for the convenience of viewing, the response frame is shown in segments)

01 2B 0E 03 03 00 00 02

87 04 00 00 00 01

88 44 31 3D 55 50 53 32 30 30 30 41 3B 32 3D 56 31 30 30 42 31 31 32 44 30 30 3B 33 3D 50 31 2E 30 2D 44 31 2E 30 3B 34 3D 32 31 30 32 32 39 30 37 31 30 48 47 46 39 30 30 30 30 31 33 3B 35 3D 31 3B 36 3D 2D 31

D2 28

According to section 1.5.3 Definition and Format of the Read Device Identifier Command, analyze this response frame.

The analysis results show that one UPS whose number is 1 is connected to the Modbus card.

Table 3-1 Analyzing the response frame

UPS Number	ESN Number	Parallel Networking Number
1	2102290710HGF9000013	1

Next, based on the UPS number, you can query and set the corresponding semaphores.

B. Query the input voltage of UPS 1, for example:

Query frame: 01 03 2A F8 00 01 0D E3

Response frame: 01 03 02 08 9D 7E 2D

According to section 2.1.2 Collection Signal List, the register address of the UPS 1 input voltage is 11000, which is 0x2AF8 in hexadecimal format. The response data is 0x089D, which is 2205 in decimal format. Divided by the gain 10, the value is 220.5, which indicates the input voltage of UPS 1 is 220.5 V.

C. Control UPS 1 to start.

Set frame: 01 06 2B 15 00 01 50 2A

Response frame: 01 06 2B 15 00 01 50 2A

D. Query UPS 1 battery undervoltage alarm state:

Query frame: 01 03 A0 E3 00 01 E6 CD

Response frame: 01 03 02 00 08 B9 82

The basic register address of the battery undervoltage alarm is 40163. Then, add the UPS 1 offset address:

40163 + 1 x 1024 = 41187 = 0xA0E3

The response data is 0x0008, and the bit 3 value is 1, indicating the battery undervoltage alarm exists.

A Acronyms and Abbreviations

C

CRC	Cyclic Redundancy Check

E

ECO	Economic Control Operation

U

UPS	Uninterruptible Power System