

HDLC-CPCI/PXIe

Datasheet

Email: yacer@yacer.cn

Web: www.yacer.com.cn

Foreword

Notational Conventions

The following categorized signal words with defined meaning might appear in the manual.







| Signal Words | Meaning |
|--|---|
|  DANGER | Indicates a high potential hazard which, if not avoided, will result in death or serious injury. |
|  CAUTION | Indicates a potential risk which, if not avoided, could result in property damage, data loss, lower performance, or unpredictable result. |
|  ANTISTATIC | Indicates static sensitive equipment. |
|  DANGER! ELECTRIC SHOCK | Indicates High voltage danger. |
|  TIPS | Provides methods to help you solve a problem or save you time. |
|  NOTE | Provides additional information as the emphasis and supplement to the text. |

Table of Contents

| | |
|--|-----------|
| Foreword | I |
| 1 Overview | 4 |
| 1.1 Introduction | 4 |
| 1.2 Features | 4 |
| 1.3 Applications | 4 |
| 1.4 Technical Specifications | 5 |
| 1.5 Order Information | 6 |
| 2 Hardware and Physical Interface | 7 |
| 2.1 Appearance | 7 |
| 2.2 LED Indicators | 7 |
| 2.3 Ethernet Interface | 8 |
| 2.4 Serial Port | 8 |
| 2.4.1 Function Description | 8 |
| 2.4.2 Duplex Mode Selection | 8 |
| 2.4.3 Pin Definition | 9 |
| 3 Building Configuration Environment | 11 |
| 3.1 Connecting the computer to the HDLC-CPCI | 11 |
| 3.2 Get configuration management software yacer-DMS | 11 |
| 3.3 Run yacer-DMS software | 11 |
| 3.4 Main Window of yacer-DMS | 11 |
| 3.5 Statistical Report | 12 |
| 3.5.1 Control Panel | 12 |
| 3.5.2 Receive/Transmit Indication Panel | 12 |
| 3.5.3 Information Panel | 13 |
| 3.6 Configure Device | 13 |
| 4 Function and Configuration | 14 |
| 4.1 Ethernet Interface Configuration | 14 |
| 4.1.1 IP address and Device alias | 14 |
| 4.1.2 Default Gateway | 14 |
| 4.1.3 Announce Configuration | 15 |
| 4.2 Serial Port Configuration | 16 |
| 4.2.1 Working mode of the serial port | 16 |
| 4.2.2 Duplex mode | 17 |
| 4.2.3 Baud rate | 17 |
| 4.2.4 Encoding format of the synchronous serial port | 17 |
| 4.2.5 HDLC-NRZ Parameter Configuration | 18 |
| 4.2.6 HDLC-NRZI Parameter Configuration | 21 |
| 4.2.7 HDLC-DBPL Parameter Configuration | 21 |
| 4.2.8 HDLC-DiffMAN (differential Manchester) Configuration | 22 |
| 4.2.9 HDLC-MAN (Manchester) Configuration | 22 |
| 4.2.10 Bit Stream Parameter Configuration | 23 |

| | |
|---|-----------|
| 4.2.11 UART Parameter Configuration | 23 |
| 4.2.12 UART-PPP Parameter Configuration | 24 |
| 4.3 UDP to Serial Configuration | 26 |
| 4.3.1 Function Description | 26 |
| 4.3.2 Protocol Conversion | 26 |
| 4.3.3 Forwarding Configuration | 27 |
| 4.3.4 Receive UDP Multicast | 27 |
| 4.4 Serial to UDP | 28 |
| 4.4.1 Function Description | 28 |
| 4.4.2 Protocol Conversion | 28 |
| 4.4.3 Forwarding Configuration | 29 |
| 4.4.4 How UDP Server identifies Source Serial ports | 29 |
| 4.5 Serial to Serial Configuration | 31 |
| 5 System Maintenance | 32 |
| 5.1 Firmware Version Upgrade | 32 |
| 5.1.1 Start Upgrade | 32 |
| 5.1.2 Select Version File | 32 |
| 5.1.3 Complete Upgrade | 33 |
| 5.1.4 Re-powering takes effect | 33 |
| 5.1.5 Confirm Upgrade | 34 |
| 5.2 Reboot Device | 34 |
| 5.3 Ping | 34 |
| 6 Forwarding Function and Data Format | 35 |
| 6.1 Application Packet and Conversion Model | 35 |
| 6.2 UDP Message Format | 36 |
| 6.3 HDLC Frame Format | 36 |
| 6.4 UART Data Packet | 36 |
| 6.5 UART-PPP Frame Format | 37 |

1 Overview

1.1 Introduction

The Yacer HDLC-CPCI/PXIe synchronous serial communication card provides 4x RS-422/485 serial ports and supports synchronous HDLC/SDLC protocol. 1x100M Ethernet interface to realize the protocol conversion between serial port and Ethernet.

Optional CPCI/PXI 3U or CPCIe/PXIe 3U bus specifications, drive-free.



1.2 Features

- 1x 100M Ethernet interface;
- 4x synchronous/asynchronous RS-422 / 485 serial port, rate up to 12 Mbps;
- Support synchronous HDLC, SDLC, asynchronous UART, Bit stream and other working modes;
- Encoding format support NRZ, NRZI, DBPL, Manchester, differential Manchester;
- Optional CPCI/PXI or CPCIe/PXIe 3U specifications;
- Driver-free, does not occupy host resources;
- Industrial grade wide temperature.

1.3 Applications

- Serial to Ethernet, Ethernet to Serial;
- Synchronous serial port, asynchronous serial port to convert each other;
- Serial port one way into multiple ways;
- High-speed synchronous serial data transmission and conversion;
- Telemetry, measurement and control data acquisition and transmission;
- Satellite, radio data transmission;
- Aviation and aerospace data communication;
- CPCI/CPCIe bus application, PXI/PXIe bus application.

1.4 Technical Specifications

| Item | Parameters | Details |
|----------------------------|-------------------------|--|
| Serial Port | Quantity | 4 |
| | Interface | 1 x Female DB-44 |
| | Working mode | Synchronous HDLC, Asynchronous UART, Synchronous Bitstream |
| | Interface type | RS-422 full duplex, RS-485 half duplex |
| | Encoding format | NRZ, NRZI, DBPL (Differential Bi-Phase Level), Manchester, Differential Manchester |
| | Baud rate | Synchronous NRZ: ≤ 12 Mbps, Synchronous Other: ≤ 6 Mbps, Asynchronous: ≤ 1 Mbps |
| | HDLC frame length | ≤ 1470 bytes |
| | Synchronous clock | General, Master, Slave(external clock) |
| | ESD protection | ± 15 KV |
| Ethernet Interface | Quantity | 1 x RJ-45 |
| | Rate | 10/100 Mbps, supporting MDI / MDIX adaptation |
| | Protocol | TCP/IP |
| | Programming interface | UDP Server, UDP Client Support unicast/multicast/broadcast |
| Computer Bus | CPCI/PXI | Compact PCI 3U, PXI 3U |
| | CPCIe/PXIe | CPCIe 3U, PXIe 3U |
| | Driver | Driver-free |
| Configuration Management | Configuration interface | Ethernet interface |
| | Configuration tool | yacer-DMS configuration management software |
| Power Requirements | Input voltage | 4.5 ~ 17 VDC |
| | Power consumption | < 3 W |
| | Power connector | CPCI/PXI or CPCIe/PXIe bus power supply |
| Mechanical Characteristics | Dimensions | 160 mm x 100 mm |
| | Weight | 450 g |
| Operating Environment | Operating temperature | -40 ~ +75°C |
| | Storage temperature | -40 ~ +85°C |
| | Operating humidity | 5 ~ 95% RH (no condensation) |

1.5 Order Information

| Product Model | RS-232 Serial Port | Ethernet Interface | Bus Specification |
|---------------|---|--------------------|-------------------|
| HDLC-CPCI-204 | 2x Synchronous serial ports + 2x Asynchronous serial ports | 1 x 10/100M | CPCI/PXI 3U |
| HDLC-CPCI-404 | 4x synchronous serial ports, 2 of which support asynchronous | 1 x 10/100M | CPCI/PXI 3U |
| HDLC-PXIe-204 | 2x Synchronous serial ports + 2x Asynchronous serial ports | 1 x 10/100M | CPCle/PXIe 3U |
| HDLC-PXIe-404 | 4x synchronous serial ports, 2 of which support asynchronous | 1 x 10/100M | CPCle/PXIe 3U |

2 Hardware and Physical Interface

2.1 Appearance

The CPCI panel leads to the serial interface (S1 ~ S4) and the 100M Ethernet interface, with the CPCI bus connector on the other end.



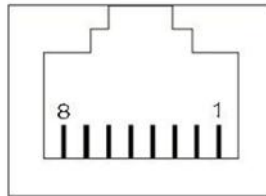
2.2 LED Indicators

| Item | Description |
|--------|--|
| ALARM | Alarm indicator, light is on when the device is not ready for startup or failure, long off when normal operation |
| RUN | Running indicator, green light flashing during normal operation |
| PWROUT | Power output indicator, long light indicates that the power supply circuit is working normally |
| PWRIN | Power input indicator, long light after correct power supply |
| S1_LED | Serial S1 send/receive indication, blinks once for each frame of data sent/received |
| S2_LED | Serial S2 send/receive indication, blinks once for each frame of data sent/received |
| S3_LED | Serial S3 send/receive indication, blinks once for each frame of data sent/received |
| S4_LED | Serial S4 send/receive indication, blinks once for each frame of data sent/received |

2.3 Ethernet Interface

One 10/100M Ethernet RJ-45 interface with MDI/MDIX adaptive support.

| RJ-45 Pin | Ethernet Signal |
|-----------|-----------------|
| 1 | Tx + |
| 2 | Tx - |
| 3 | Rx + |
| 6 | Rx - |



2.4 Serial Port

2.4.1 Function Description

Serial port supports synchronous HDLC/SDLC protocol and asynchronous UART working mode, and the encoding format supports NRZI, Manchester, differential Manchester, DBPL, etc.

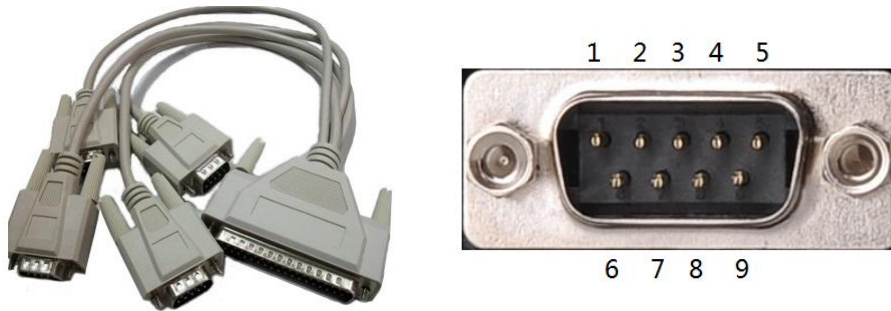
2.4.2 Duplex Mode Selection

Serial port supports both full-duplex RS-422 and half-duplex RS-485 physical layer standards, and the user can select the appropriate working mode by configuring the J6 jumper.

| Serial Port | Jumper Name | RS-422 full duplex | RS-485 half duplex |
|-------------|---------------|--------------------|--------------------|
| S1 | TXD1- / RXD1- | OFF | ON |
| | TXD1+ / RXD1- | OFF | ON |
| | TXC1- / RXC1- | OFF | ON |
| | TXC1+ / RXC1+ | OFF | ON |
| S2 | TXD2- / RXD2- | OFF | ON |
| | TXD2+ / RXD2- | OFF | ON |
| | TXC2- / RXC2- | OFF | ON |
| | TXC2+ / RXC2+ | OFF | ON |
| S3 | TXD3- / RXD3- | OFF | ON |
| | TXD3+ / RXD3- | OFF | ON |
| | TXC3- / RXC3- | OFF | ON |
| | TXC3+ / RXC3+ | OFF | ON |
| S4 | TXD4- / RXD4- | OFF | ON |
| | TXD4+ / RXD4- | OFF | ON |
| | TXC4- / RXC4- | OFF | ON |
| | TXC4+ / RXC4+ | OFF | ON |

2.4.3 Pin Definition

Serial ports S1 ~ S4 share a DB44 hole type connector, which can be converted to 4-way DB9 pin type connector via the standard YC9-44T one-in-four cable.



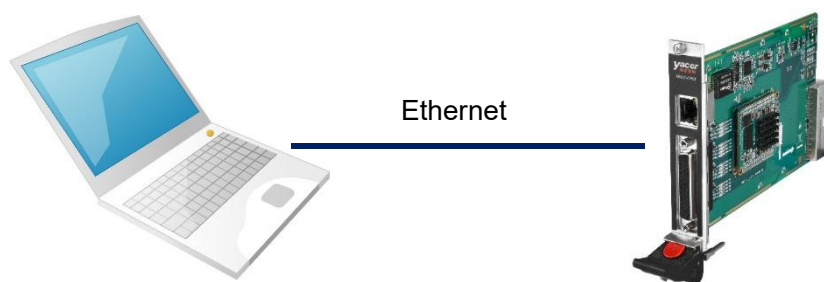
| DB44 hole type | | RS-422 full duplex | RS-485 half duplex | YC9-44T one-in-four cable DB9 pin type |
|----------------|-----|--------------------|--------------------|--|
| Serial | PIN | | | |
| S1 | 32 | TxData1 + | Data1 + | S1-5 |
| | 31 | TxData1 - | Data1 - | S1-9 |
| | 18 | TxClock1 + | Clock1 + | S1-4 |
| | 3 | TxClock1 - | Clock1 - | S1-8 |
| | 1 | RxData1 + | | S1-1 |
| | 16 | RxData1 - | | S1-6 |
| | 17 | RxClock1 + | | S1-2 |
| | 2 | RxClock1 - | | S1-7 |
| | 33 | GND | GND | S1-3 |
| S2 | 4 | TxData2 + | Data2 + | S2-5 |
| | 19 | TxD2ata - | Data2 - | S2-9 |
| | 22 | TxClock2 + | Clock2 + | S2-4 |
| | 7 | TxClock2 - | Clock2 - | S2-8 |
| | 5 | RxData2 + | | S2-1 |
| | 20 | RxData2 - | | S2-6 |
| | 21 | RxClock2 + | | S2-2 |
| | 6 | RxClock2 - | | S2-7 |
| | 35 | GND | GND | S2-3 |
| S3 | 8 | TxData3 + | Data3 + | S3-5 |
| | 23 | TxData3 - | Data3 - | S3-9 |
| | 26 | TxClock3 + | Clock3 + | S3-4 |
| | 11 | TxClock3 - | Clock3 - | S3-8 |
| | 9 | RxData3 + | | S3-1 |
| | 24 | RxData3 - | | S3-6 |
| | 25 | RxClock3 + | | S3-2 |
| | 10 | RxClock3 - | | S3-7 |
| | 39 | GND | GND | S3-3 |

| DB44 hole type | | RS-422 full duplex | RS-485 half duplex | YC9-44T one-in-four cable DB9 pin type |
|----------------|-----|--------------------|--------------------|--|
| Serial | PIN | | | |
| S4 | 12 | TxData4 + | Data4 + | S4-5 |
| | 27 | TxData4 - | Data4 - | S4-9 |
| | 30 | TxClock4 + | Clock4 + | S4-4 |
| | 15 | TxClock4 - | Clock4 - | S4-8 |
| | 13 | RxData4 + | | S4-1 |
| | 28 | RxData4 - | | S4-6 |
| | 29 | RxClock4 + | | S4-2 |
| | 14 | RxClock4 - | | S4-7 |
| | 43 | GND | GND | S4-3 |

3 Building Configuration Environment

3.1 Connecting the computer to the HDLC-CPCI

Connect the Ethernet port of the management computer and HDLC-CPCI through the network cable, and run the yacer-DMS configuration management software on the computer to monitor the operation status and configure the parameters of HDLC-CPCI.



3.2 Get configuration management software yacer-DMS

Users can obtain the compressed package yacer-DMS.zip of configuration management software through the following ways:

- Software & Tools directory of HDLC-CPCI accompanied U-Disk;
- Official website of Yacer (<http://www.yacer.com.cn>) Software channel.

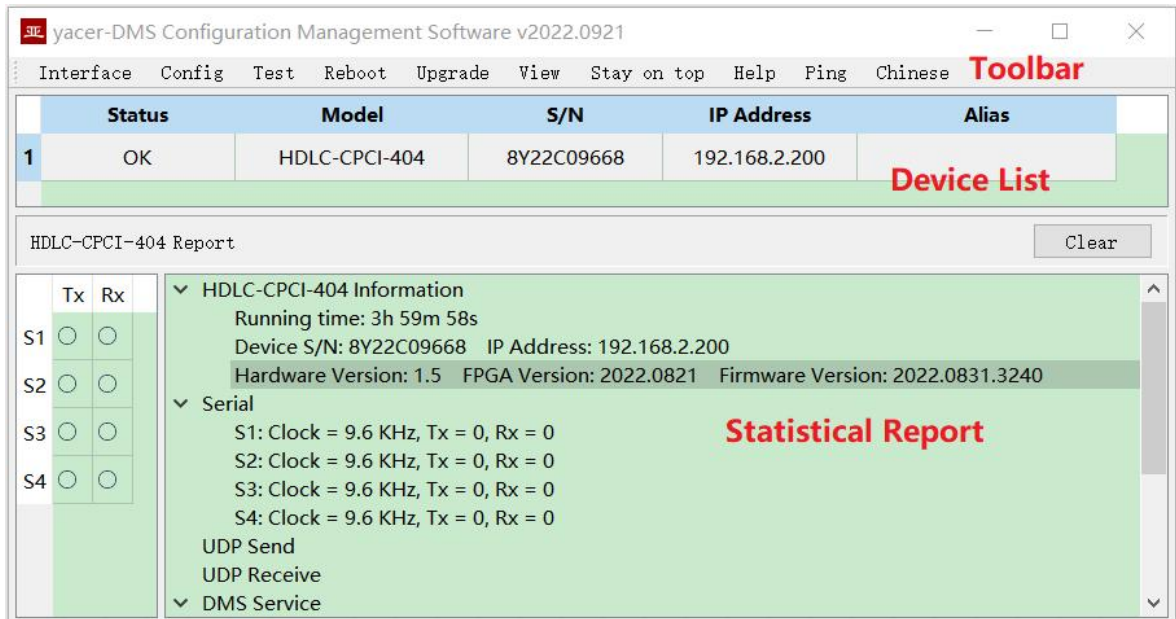
3.3 Run yacer-DMS software

The yacer-DMS is an installation free application software, unzip yacer-DMS.zip, enter the working directory and double click the file yacer-DMS.exe to run.

3.4 Main Window of yacer-DMS

The following figure is the main interface of the configuration management software, which can be divided into three parts:

- Toolbar: Functional operation buttons;
- Device List: Displaying the basic information and operation status of online devices;
- Statistical Report: Displaying the receive/transmit indication & statistics, and device details.



3.5 Statistical Report

The statistical report has three panels: control panel, receive/transmit indication panel and information panel.

3.5.1 Control Panel

| HDLC-CPCI-404 Report Clear | |
|---|------------------------------|
| Control Widget | Descriptions |
| <input type="button" value="Clear"/> | Clear the statistical report |

3.5.2 Receive/Transmit Indication Panel

- Tx: The interface sends a frame of data, corresponding Tx indicator blinks once;
- Rx: The interface receives a frame of data, corresponding Rx indicator blinks once.

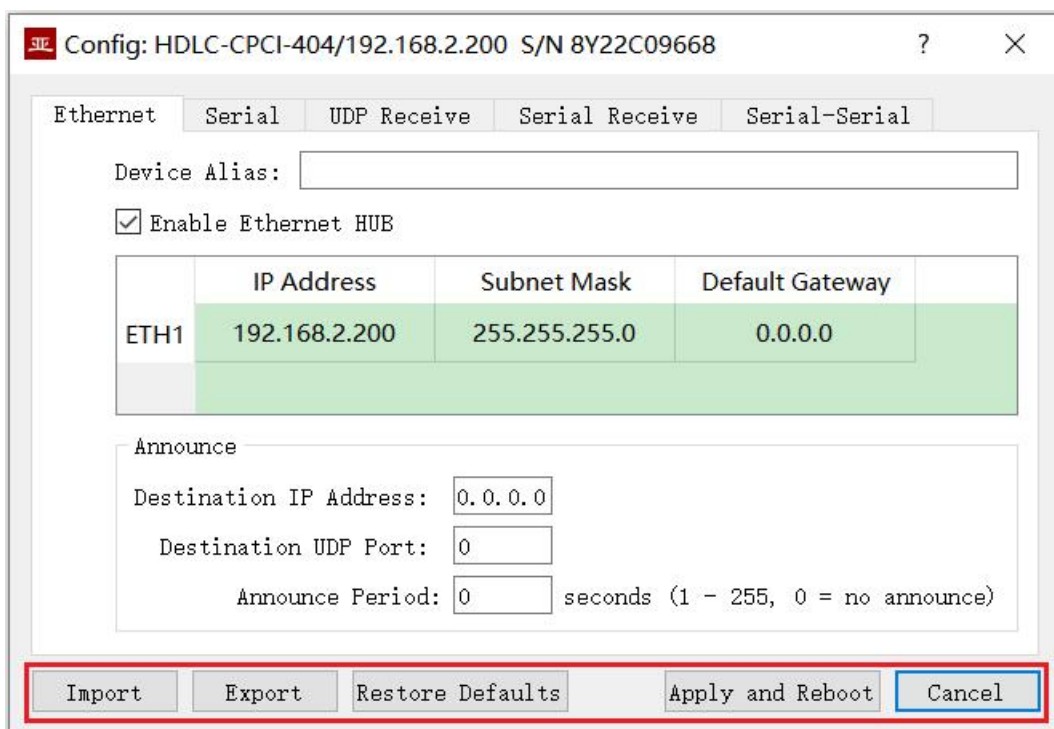
| | Tx | Rx |
|----|-----------------------|-----------------------|
| S1 | <input type="radio"/> | <input type="radio"/> |
| S2 | <input type="radio"/> | <input type="radio"/> |
| S3 | <input type="radio"/> | <input type="radio"/> |
| S4 | <input type="radio"/> | <input type="radio"/> |

3.5.3 Information Panel

The information display panel shows the following:

- Device information: Running time, S/N, IP address and Version number;
- Serial: Receive/transmit statistics of all serial ports;
- UDP Send: Send packets of the UDP Client for each enabled serial port to UDP entry;
- UDP Receive: Received packets of UDP server for each enabled UDP to serial port entries;
- DMS Service: DMS message receive/transmit statistics.

3.6 Configure Device



The bottom of the dialog box includes the following operation buttons:

| Button | Function |
|------------------|--|
| Import | Open the configuration file, read the configuration parameters refresh the configuration dialog |
| Export | Export configuration parameters from the configuration dialog to a file for saving |
| Restore Defaults | Refresh the configuration dialog with the factory parameters |
| Apply and Reboot | Write the configuration parameters in the dialog to the device, and restart the device to make the configuration take effect |
| Cancel | Cancel current configuration operation |

4 Function and Configuration

4.1 Ethernet Interface Configuration

4.1.1 IP address and Device alias

The user sets the Ethernet interface address for the HDLC-CPCI by configuring the ETH1 IP address, subnet mask and default gateway.

Device aliases can also be set, thus adding descriptions or helper identifiers to the device.

The screenshot shows a configuration window with tabs for Ethernet, Serial, UDP Receive, Serial Receive, and Serial-Serial. The Ethernet tab is active. Below the tabs, there is a 'Device Alias' text box, a checked checkbox for 'Enable Ethernet HUB', and a table for interface configuration.

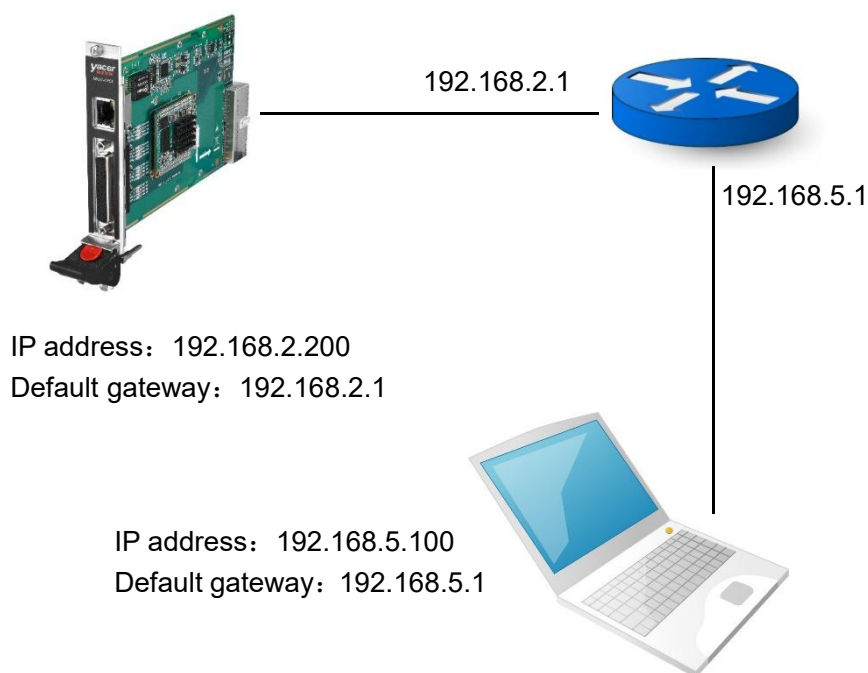
| | IP Address | Subnet Mask | Default Gateway |
|------|---------------|---------------|-----------------|
| ETH1 | 192.168.2.200 | 255.255.255.0 | 0.0.0.0 |

4.1.2 Default Gateway

By default, the default gateway is 0.0.0.0, representing that there is no gateway configuration.

If HDLC-CPCI needs to communicate with the host on other subnet, it must rely on an external router. At this time, the HDLC-CPCI's IP address must be on the same subnet with the IP address of the connected router port. Meanwhile, the IP address of router is set to the default gateway.

As shown below, the IP address of HDLC-CPCI and remote PC is 192.168.2.200 and 192.168.5.100 respectively. As they do not belong to the same subnet, they must rely on the router for communication. HDLC-CPCI and PC need to set the IP address of the connected router port to the default gateway of this device.



4.1.3 Announce Configuration

HDLC-CPCI can actively send announce data to the management computer to provide converter operation information, data communication transceiver statistical reports.

Announce

Destination IP Address:

Destination UDP Port:

Announce Period: seconds (1 - 255, 0 = no announce)

The notification settings include the following parameters:

| Parameter | Description |
|------------------------|---|
| Destination IP address | The notification message is sent to the destination IP, which can be unicast or multicast address |
| Destination UDP port | The notification is sent by UDP message. The destination UDP port number can be set. |
| Announce Period | Sending period of device operation information notification, 1 ~ 255 seconds |

4.2 Serial Port Configuration

4.2.1 Working mode of the serial port

Serial ports S1 and S2 are synchronous and asynchronous serial ports, which support synchronous and asynchronous working modes.

Other serial ports are synchronous serial ports, which only support synchronous working mode.

| Working Mode | | Description | Supported Port |
|--------------|--------------|--|----------------|
| Synchronous | HDLC-NRZ | Synchronous HDLC protocol based on NRZ encoding | All ports |
| | HDLC-NRZI | Synchronous HDLC protocol based on NRZI encoding | All ports |
| | HDLC-DBPL | Synchronous HDLC protocol based on the DBPL (Differential Bi-Phase Level) encoding | All ports |
| | HDLC-MAN | Synchronous HDLC protocol based on the Manchester encoding | All ports |
| | HDLC-DiffMAN | Synchronous HDLC protocol based on the differential Manchester encoding | All ports |
| | Bit Stream | Transmit or sampling serial bit data based on receiving clock | S1, S2 |
| Asynchronous | UART | Universal asynchronous serial mode, similar to serial port on a universal computer | S3, S4 |
| | UART-PPP | Frame transmission on UART interface using PPP protocol | S3, S4 |

Users can select the desired working mode from the “working mode” combo box. Due to different parameter configurations of each working mode, the contents of the “Options” cell will be adjusted automatically according to the determined working mode.

If you need to further more configuration of working parameters for the selected working mode, double-click on the “Options” cell to pop up the parameter configuration dialog.

| | S1 | S2 | S3 | S4 |
|------------------------|--|--|---|---|
| Working Mode | HDLC-NRZ | HDLC-NRZ | UART | UART-PPP |
| Duplex | Full-Duplex | Full-Duplex | Full-Duplex | Half-Duplex |
| Baudrate (bps) | 9600 | 9600 | 115200 | 9600 |
| Options (Double-click) | Clock Mode: Normal Transmit Trigger: Falling Edge of Clock Receive Trigger: Rising Edge of Clock CRC: CRC-16 HDLC Rx FCS: Discard Idle Flag: 0xFF Preamble Flag: 0xFF Preamble Num: 0 Head Size: 0 | Clock Mode: Normal Transmit Trigger: Falling Edge of Clock Receive Trigger: Rising Edge of Clock CRC: CRC-16 HDLC Rx FCS: Discard Idle Flag: 0xFF Preamble Flag: 0xFF Preamble Num: 0 Head Size: 0 | Data Bits: 8 Parity Bits: None Stop Bits: 1 Packing Size: 128bytes Packing Interval: 10ms | Data Bits: 8 Parity Bits: None Stop Bits: 1 CRC: Enable Rx FCS: Discard |

4.2.2 Duplex mode

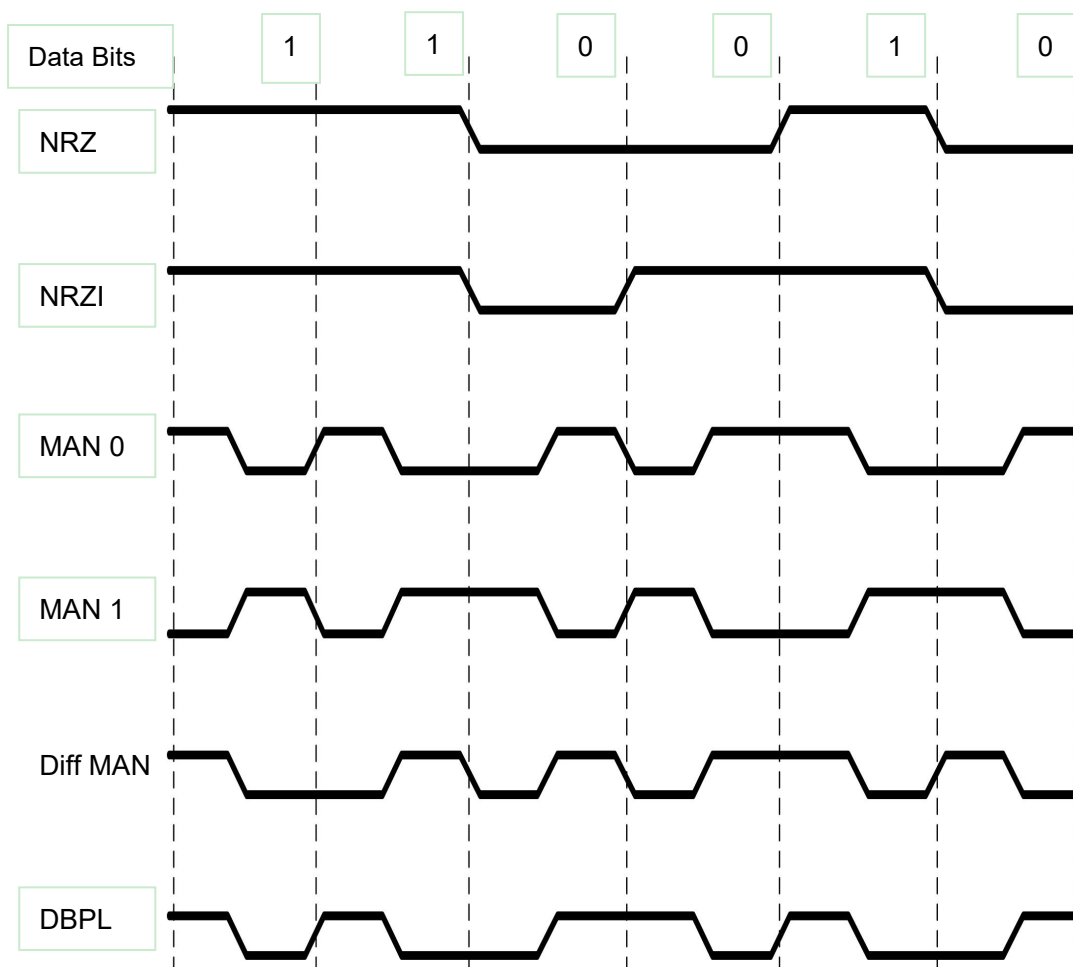
Must be consistent with the J6 configuration.

4.2.3 Baud rate

The "Baud Rate" configures the communication rate of the serial port. For HDLC-NRZI, HDLC-DBPL, HDLC-MAN, HDLC-DiffMAN and all asynchronous modes, the baud rate of both sides of communication must be the same to ensure proper data transmission.

4.2.4 Encoding format of the synchronous serial port

For HDLC-NRZ, HDLC-NRZI, HDLC-DBPL, HDLC-MAN, HDLC-DiffMAN and other synchronous working modes, the link layer adopts the HDLC protocol with the encoding format difference as follows:



4.2.5 HDLC-NRZ Parameter Configuration

HDLC-NRZ is the common synchronous working mode, which is mainly used for the secondary surveillance radar and ADS-B data communication in air traffic control (ATC) and air traffic management (ATM) fields.

HDLC - NRZ encoding

Clock Mode: Normal

Transmit Trigger: Falling Edge of Clock

Receive Trigger: Rising Edge of Clock

CRC: CRC-16 HDLC

Forward received FCS field

Idle Flag: 0xFF

Preamble Flag: 0x7E

Preamble Number: 0

Header Size: 0 bytes

Header Data: (Hex)

OK Cancel

4.2.5.1 Clock Mode

Clock Mode: Normal

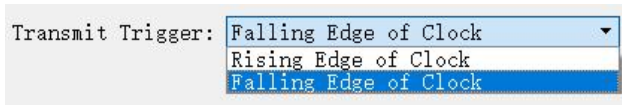
- Normal
- Slave (External)
- Master

There are three clock modes for the synchronous serial port, normal, slave clock and master clock.

| Clock Mode | Transmit Clock | Receive Clock |
|-----------------|---|--|
| Normal | Local device generated, output via pin TxC | Opposite device generated, input via pin RxC |
| Slave(External) | Opposite device generated from pin RxC. TxC output synchronizes with RxC automatically. | Opposite device generated, input via pin RxC |
| Master | Local device generated, output via pin TxC | Local device generated, ignore pin RxC input |

Slave clock mode is also called external clock mode. When the opposite side is the transmission device, HDLC-CPCI is often configured to slave clock mode, sending data with the clock provided by the transmission device to ensure that the data transmission in the whole network is based on the same clock, avoiding the hidden danger of packet loss caused by different clock sources.

4.2.5.2 Transmit Trigger

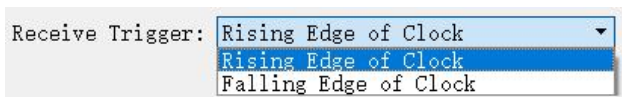


Transmit trigger defines the generation time of new data bit:

- Falling edge of clock: Generate new data bits at the falling edge of clock;
- Rising edge of clock: Generate new data bits at the rising edge of clock.

Communication that follows the HDLC protocol specification should choose the falling edge of the clock to trigger new data transmission. There are also some special applications where the user uses a non-standard communication method with rising edge triggered new data transmission.

4.2.5.3 Receive Trigger



Receive trigger defines the sampling time of serial receiving data:

- Rising edge of clock: Read data on the RxD line at the rising edge of RxC signal;
- Falling edge of colck: Read data on the RxD line at the falling edge of RxC signal.

Communication following the HDLC protocol specification, since the falling edge is used to trigger the new data, the receive trigger must be configured as a rising edge of the clock in order to ensure the correct reading of the data, considering the stabilization time of the new data.

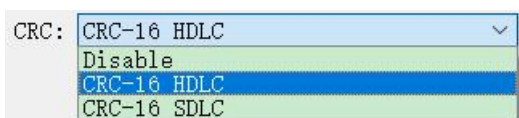
The receive trigger configuration should be determined according to the transmit trigger of the opposite device:

| Opposite Transmit Clock | Local Receive Clock |
|-------------------------|-----------------------|
| Falling edge of clock | Rising edge of clock |
| Rising edge of clock | Falling edge of clock |

4.2.5.4 CRC

To verify the correctness of data communication, CRC functionality should be enabled.

By default, the CRC-16-HDLC check mode should be selected for HDLC communication.



| CRC Type | Description |
|----------|---------------|
| Disable | CRC disabled: |

| CRC Type | Description |
|-------------|---|
| | <ul style="list-style-type: none"> Send: No CRC calculation, no FCS field for HDLC frame Receive: No CRC checking |
| CRC-16 HDLC | 16-bit ISO HDLC CRC verification |
| CRC-16 SDLC | 16-bit IBM SDLC CRC verification |

4.2.5.5 Forward Received FCS Field

Forward received FCS field

This configuration only takes effect when CRC enabled.

HDLC frame structure is shown in the following table, where FCS is the frame check sequence field.

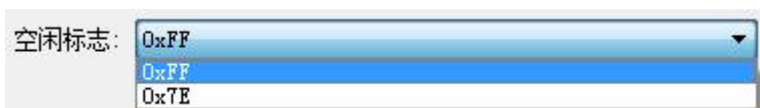
| Opening Flag | Address Field | Control Field | Information Field | FCS Field | Closing Flag |
|--------------|---------------|---------------|-------------------|-------------|--------------|
| 0x7E | 1 Byte | 1 Byte | Variable length | CRC 2 bytes | 0x7E |
| 0x7E | User data | | | CRC 2 bytes | 0x7E |

By default, this option is unchecked, HDLC-CPCI discards the FCS field of 2 bytes at the end of the data and forwards only user data after receiving the HDLC frame and passing the CRC check.

If the check box is checked, the user data + FCS field is forwarded.

4.2.5.6 Idle Flag

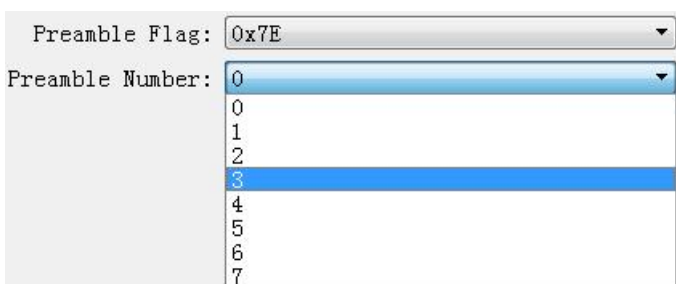
Define the fill content between HDLC frames, 0xFF should be selected by default



4.2.5.7 Preamble Flag and Number

For half-duplex applications, 2-5 0x7E preamble flag are often added to the frame header to prepare the receiver for synchronization.

For full-duplex applications, there is often no need for a preamble flag, just set the number of preamble to 0 (no preamble).



4.2.5.8 Frame Header Length and Content

Header Size: bytes

Header Data: (Hex)

The above figure is taken as an example. The length of frame header is defined as 2, and the content of frame header is defined as FF 03 in hexadecimal:

- HDLC send: add FF 03 before user data to make up HDLC frame data with user data;
- HDLC Receive: HDLC-CPCI discards the first two bytes of HDLC frame data as a header and forwards only subsequent data to the user.

| Opening Flag | Flame Header | User Data | FCS Field | Closing Flag |
|--------------|--------------|-----------------|-------------|--------------|
| 0x7E | 0xFF 0x03 | Variable length | CRC 2 bytes | 0x7E |

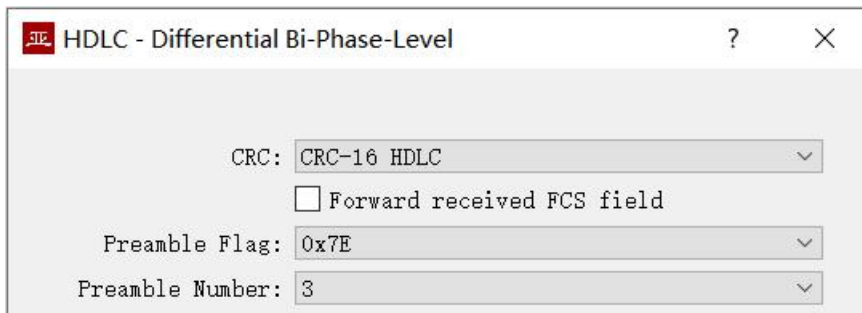
4.2.6 HDLC-NRZI Parameter Configuration

Unlike NRZ encoding format, NRZI encoding format contains clock information in the data, which only requires the same baud rate for both sides of communication, and no longer requires parameters such as clock mode, transmit trigger, and receive trigger.

The configuration parameters of HDLC-NRZI are shown in the figure below, and the function and configuration of the parameters are the same as HDLC-NRZ working mode.

4.2.7 HDLC-DBPL Parameter Configuration

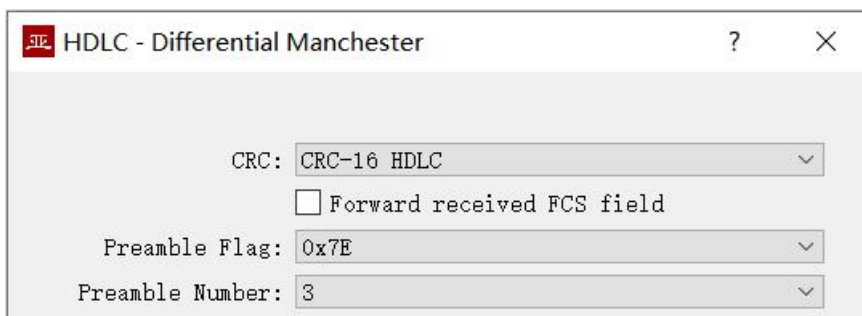
HDLC-DBPL adopts the Differential Bi-Phase Level coding format with the same parameter meaning as HDLC-NRZI.



It is important to note that many claims that DBPL encoding is in fact differential Manchester encoding, so users need to carefully refer to the definition of the Synchronous Serial Encoding Format chapter to choose the correct working mode.

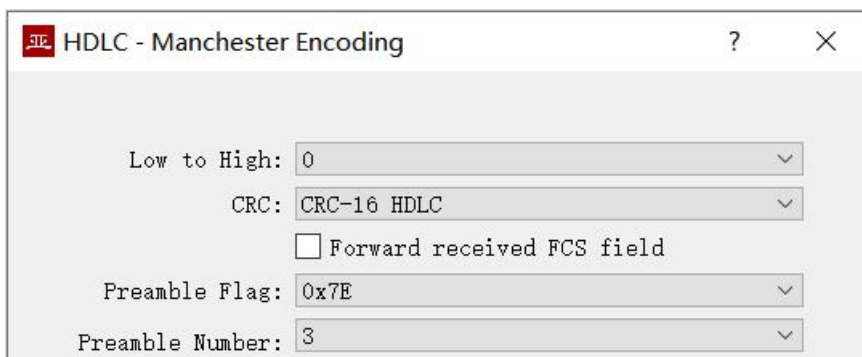
4.2.8 HDLC-DiffMAN (differential Manchester) Configuration

The Differential Manchester Options dialog box is shown below , with the same parameter meaning as HDLC-NRZI.



4.2.9 HDLC-MAN (Manchester) Configuration

The Manchester Options dialog box is shown below:



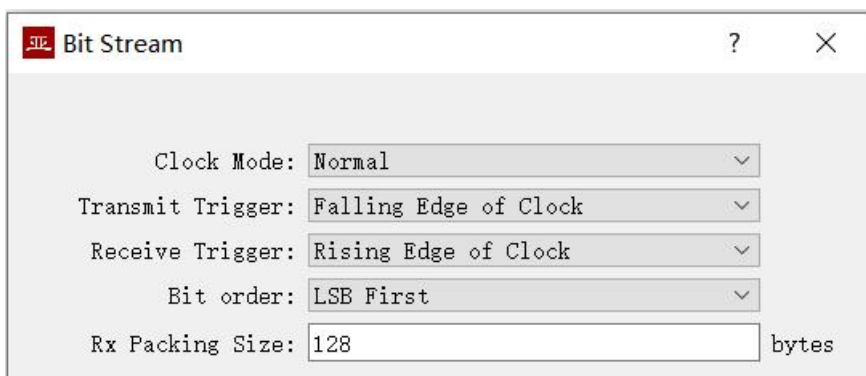
In addition to the same configuration parameters as NRZI, the Manchester encoding format has parameters with the meaning of edges with low to high waveforms for data lines:

- 0: Low to high edges represent logical 0;
- 1: Low to high edges represent logical 1.



4.2.10 Bit Stream Parameter Configuration

The rising or falling edge of each clock cycle samples the 1bit data on the data line, which forms a UDP message and sends to the destination IP after receiving a byte with the packet length by forming a byte with each 8bit.



Refer to the HDLC-NRZ parameter configuration for configuration of clock mode, transmit trigger, receive trigger and idle flag.

The online bit stream is stored in the computer or system memory in the form of byte. The receive/transmit sequence determines the conversion mode of byte and bit.

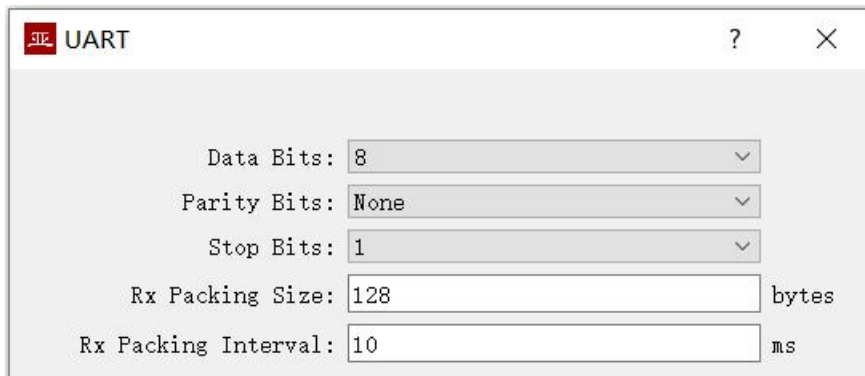


| Receive/Transmit Sequence | Transmit Operation | Receive Operation |
|---------------------------|----------------------------------|--|
| MSB First | Transmit the high-bit byte first | Data received first is placed in the high-bit byte |
| LSB First | Transmit the low-bit byte first | Data received first is placed in the low-bit byte |

4.2.11 UART Parameter Configuration

UART is a type of character stream communication. Data bits, parity bits and stop bits define the basic working parameters of the asynchronous serial port, which must be identical to the configuration of opposite device.

Generally, Data bits are defined as 8 bits (1 byte), so that UART corresponds to the communication of byte streams.



The screenshot shows a window titled "UART" with a close button and a help icon. It contains five configuration fields: "Data Bits" set to 8, "Parity Bits" set to None, "Stop Bits" set to 1, "Rx Packing Size" set to 128 bytes, and "Rx Packing Interval" set to 10 ms.

When converting the byte stream of UART into UDP message or HDLC frame, if every byte is converted into a UDP message for transmission, the overhead is too large and the efficiency is too low.

In order to improve the efficiency, HDLC-CPCI will buffer the received byte stream, and then send out a UDP message composed of several buffered bytes. This process is called packing. Packing is controlled by two parameters, which are called Packing Size & Packing interval.

4.2.11.1 Packing Size

For example, if the Packing Size is set to 128 bytes, then when UART receives 128 bytes, a packet will be formed for forwarding.



The screenshot shows a single configuration field: "Rx Packing Size: 128 bytes".

4.2.11.2 Packing Interval

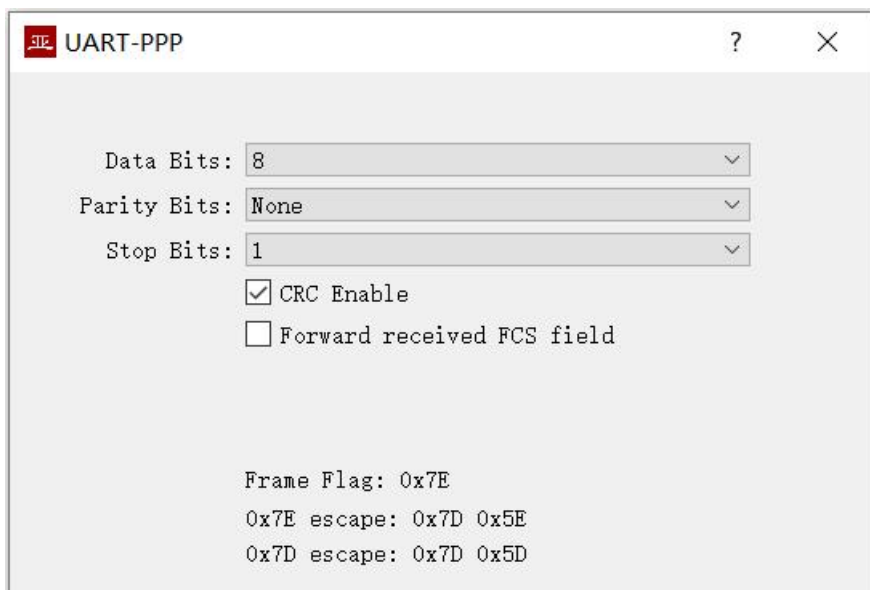
For example, the above example sets the Packing Interval to 10ms. If the UART does not receive new data after 10ms, the data in the buffer will be forwarded as a packet regardless of whether 128 bytes are received.



The screenshot shows a single configuration field: "Rx Packing Interval: 10 ms".

4.2.12 UART-PPP Parameter Configuration

Since the UART sends and receives a headerless character stream, in order to perform one-to-one conversion with UDP messages or HDLC frames, a UART-PPP frame is constructed by adding 0x7E as the start and end flags at its header and end and inserting a frame checksum sequence.



The UART-PPP frame format adds 0x7E before and after the packet as the opening flag closing flag with the frame structure as follows:

| Opening Flag | Information Field | FCS Field | Closing Flag |
|--------------|-------------------|-------------|--------------|
| 0x7E | 2-1470 bytes | 2 bytes CRC | 0x7E |

As the information field and FCS field may appear 0x7E, perform the character escape on such fields before transmission with the escape rules as follows:

- 0x7E: Escaped to two characters, 0x7D 0x5E;
- 0x7D: Escaped to two characters, 0x7D 0x5D;
- Other characters: No escape.

The escape operation of data send is as follows:

| Original Data | Actual Transmit Data |
|---------------|----------------------|
| 0x7E | 0x7D 0x5E |
| 0x7D | 0x7D 0x5D |
| Others | No change |

The escape operation of data receive is as follows:

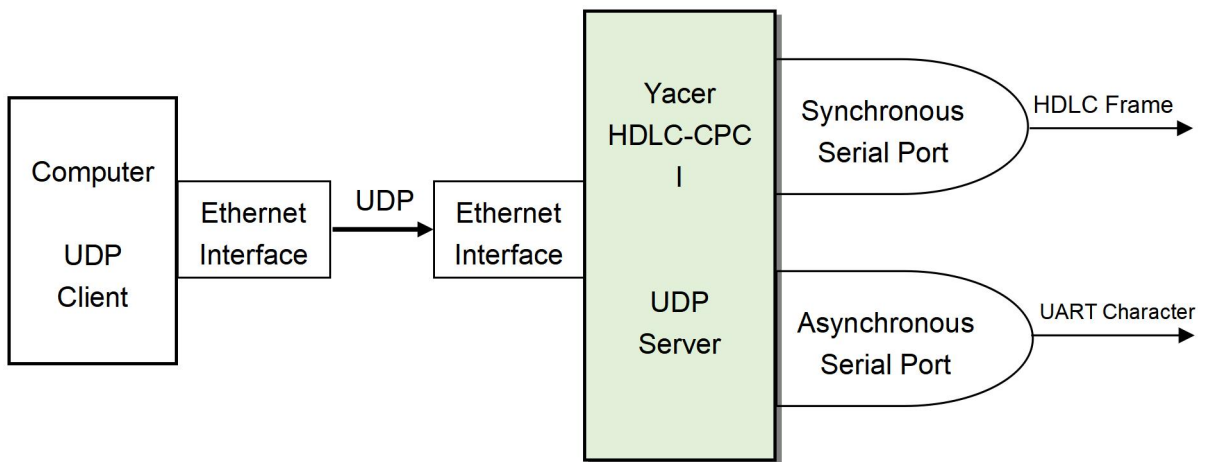
| Original Data | Actual Transmit Data |
|---------------|----------------------|
| 0x7D 0x5E | 0x7E |
| 0x7D 0x5D | 0x7D |
| Others | No change |

4.3 UDP to Serial Configuration

4.3.1 Function Description

With HDLC-CPCI, a PC or server can realize the data send function of the synchronous HDLC or UART.

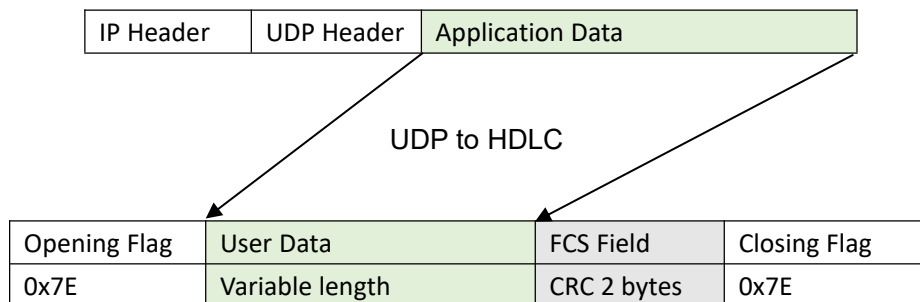
A typical application is shown in the figure below. Computer as UDP Client sends UDP messages through Ethernet interface, HDLC-CPCI processes the received UDP messages, converts them into HDLC frames or UART data according to the configuration, and then sends them out from the serial port



4.3.2 Protocol Conversion

The most typical application of UDP to HDLC is shown in the figure below. HDLC-CPCI loads UDP application data into the user data area of HDLC frame, then calculates CRC and fills FCS field to form a complete HDLC frame for sending.

In order to reduce the computational load of the computer and the complexity of user programming, generally, the UDP message does not contain the FCS field of HDLC, which is calculated and filled by HDLC-CPCI.



4.3.3 Forwarding Configuration

Set UDP to serial ports, each line represents a UDP port to serial forwarding entry. Three forwarding strategies can be implemented:

- Forwarding: Data received by the specified UDP port can be forwarded to the specified serial port;
- Convergence: Data received by multiple different UDP ports can be forwarded to the same serial port;
- Branch: Data received from the UDP port can be forwarded to multiple serial ports at the same time.

| Ethernet | | Serial | | UDP Receive | | Serial Receive | | Serial-Serial | |
|-------------------|------------|---------------|----------------------|---|---------|----------------|--|---------------|--|
| Local Rx UDP Port | Forward to | Egress Serial | Rx Multicast Address | | | | | | |
| 1 | 8001 | enable | Serial-S1 | Group-1 | 0.0.0.0 | | | | |
| 2 | 8002 | enable | Serial-S2 | Group-2 | 0.0.0.0 | | | | |
| 3 | 8003 | enable | Serial-S3 | Group-3 | 0.0.0.0 | | | | |
| 4 | 8004 | enable | Serial-S4 | Group-4 | 0.0.0.0 | | | | |
| 5 | 0 | disable | Serial-S1 | Group-5 | 0.0.0.0 | | | | |
| 6 | 0 | disable | Serial-S2 | Group-6 | 0.0.0.0 | | | | |
| 7 | 0 | disable | Serial-S3 | Group-7 | 0.0.0.0 | | | | |
| 8 | 0 | disable | Serial-S4 | Group-8 | 0.0.0.0 | | | | |
| | | | | (224.0.0.0 - 239.255.255.255) | | | | | |
| | | | | <input type="checkbox"/> Report serial TxFiFo usage | | | | | |

The following configuration implements an application that receives data from a UDP port and distributes it to four serial ports simultaneously:

| Ethernet | | Serial | | UDP Receive | | Serial Receive | | Serial-Serial | |
|-------------------|------------|---------------|----------------------|-------------|---------|----------------|--|---------------|--|
| Local Rx UDP Port | Forward to | Egress Serial | Rx Multicast Address | | | | | | |
| 1 | 8000 | enable | Serial-S1 | Group-1 | 0.0.0.0 | | | | |
| 2 | 8000 | enable | Serial-S2 | Group-2 | 0.0.0.0 | | | | |
| 3 | 8000 | enable | Serial-S3 | Group-3 | 0.0.0.0 | | | | |
| 4 | 8000 | enable | Serial-S4 | Group-4 | 0.0.0.0 | | | | |
| 5 | 0 | disable | Serial-S1 | Group-5 | 0.0.0.0 | | | | |
| | | | | Group-6 | 0.0.0.0 | | | | |

4.3.4 Receive UDP Multicast

If users need to receive a multicast UDP message, add the required multicast address to the "Rx Multicast Address" list on the right.

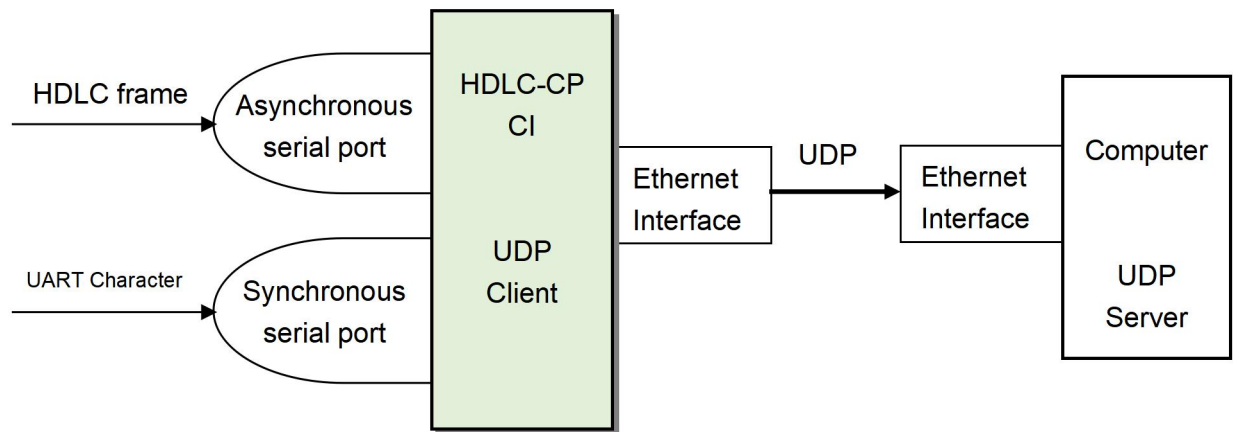
Range of the multicast address is 224.0.0.0 ~ 239.255.255.255, 224.8.8.8 is the configuration management address for HDLC-CPCI and cannot be used by users.

The multicast address configured as 0.0.0.0 indicates that the entry is not valid.

4.4 Serial to UDP

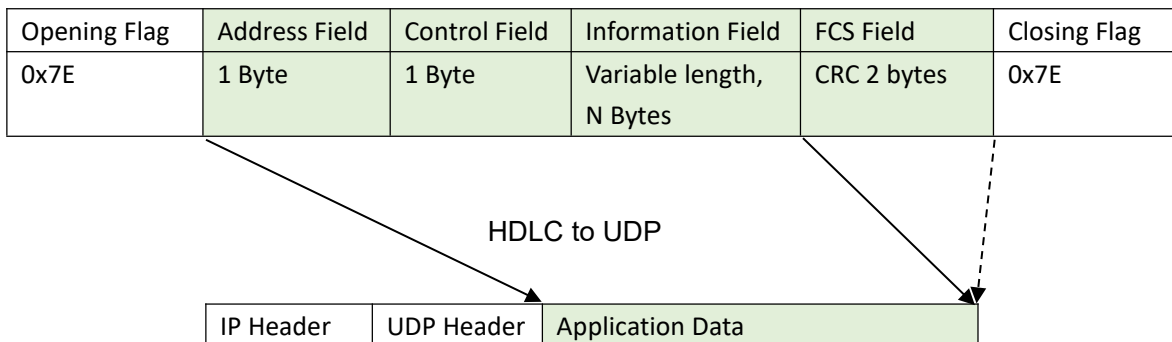
4.4.1 Function Description

The schematic diagram of serial port to UDP function is as follows. HDLC-CPCI receives HDLC frames or UART data from other device through serial interface, converts them into UDP messages, and sends the messages to computers or servers through Ethernet.



4.4.2 Protocol Conversion

To ensure the integrity of user data, HDLC-CPCI places complete HDLC frames in UDP application data and forwards them to UDP Server.



4.4.3 Forwarding Configuration

Set serial ports to UDP, each line represents a serial to UDP port forwarding entry. Three forwarding strategies can be implemented

- Forwarding: Data received from the specified serial port can be forwarded to the specified destination IP + UDP port;
- Convergence: Data received from multiple different serial ports can be forwarded to the same destination IP + UDP port;
- Branch: Data received from the serial port can be forwarded to multiple destination IP or UDP port at the same time.

| Ethernet | | Serial | UDP Receive | Serial Receive | Serial-Serial |
|----------|----------------|------------|----------------------|--------------------|---------------|
| | Ingress Serial | Forward to | Remote Rx IP Address | Remote Rx UDP Port | |
| 1 | Serial-S1 | enable | 192.168.2.80 | 8000 | |
| 2 | Serial-S2 | enable | 255.255.255.255 | 9000 | |
| 3 | Serial-S3 | enable | 224.10.10.10 | 10000 | |
| 4 | Serial-S1 | disable | 0.0.0.0 | 0 | |
| 5 | Serial-S1 | disable | 0.0.0.0 | 0 | |
| 6 | Serial-S2 | disable | 0.0.0.0 | 0 | |
| 6 | Serial-S3 | disable | 0.0.0.0 | 0 | |
| 6 | Serial-S4 | disable | 0.0.0.0 | 0 | |
| 7 | Serial-S1 | disable | 0.0.0.0 | 0 | |
| 8 | Serial-S1 | disable | 0.0.0.0 | 0 | |

As shown in the figure above, three Serial to UDP entries are configured to implement:

- Serial port S1 to UDP unicast, destination IP address 192.168.2.80 and destination UDP port 8000;
- Serial port S2 to UDP broadcast, all hosts in the network can receive data from S2 at port 9000;
- Serial port S3 to UDP multicast, only computers joined 224.10.10.10 groups in the network can receive data from S3.

4.4.4 How UDP Server identifies Source Serial ports

In many applications, such as air traffic control automation (ATC) applications, HDLC frames originating from multiple different serial ports need to be forwarded to a server or computer for uniform processing. In this case, a strategy is needed to let the computer know which serial port the UDP message is received from.

4.4.4.1 Identify Source Serial Port Based on Destination UDP Port

As shown in the figure below, set different forwarding destination UDP ports for each serial port. As a UDP Server, computer receives data on different UDP ports: the message received on port 8001 comes from serial port S1, and the message received on port 8002 comes from serial port S2.

| Ethernet | | Serial | | UDP Receive | | Serial Receive | | Serial-Serial | |
|----------|----------------|--------|--------------|-------------|----------------------|----------------|--------------------|---------------|--|
| | Ingress Serial | | ➔ Forward to | | Remote Rx IP Address | | Remote Rx UDP Port | | |
| 1 | Serial-S1 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8001 | | |
| 2 | Serial-S2 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8002 | | |
| 3 | Serial-S3 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8003 | | |
| 4 | Serial-S4 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8004 | | |

4.4.4.2 Identify Source Serial Port Based on Source UDP Port

UDP Server needs to listen and receive data on multiple UDP ports when using the destination UDP port to identify the source serial port scheme. When the number of serial ports is large, not only does UDP Server consume too much port resources, but also the complexity of configuration and programming increases a lot.

To simplify the implementation of the UDP Server side, we can forward each transformation to the same port of the UDP Server using the configuration shown below. *Yacer HDLC-CPCI automatically adjusts the source port number of UDP message according to the source serial port when forwarding, where serial port S1 forwards UDP message with source port 8001 and serial port S2 with 8002, increasing gradually below.*

In this way, UDP Server only needs to listen and receive data on one port (in the example below, 8000), and then differentiate the source serial ports based on the source UDP port. If there are multiple HDLC-CPCI, UDP Server can distinguish the source devices through the source IP.

| Ethernet | | Serial | | UDP Receive | | Serial Receive | | Serial-Serial | |
|----------|----------------|--------|--------------|-------------|----------------------|----------------|--------------------|---------------|--|
| | Ingress Serial | | ➔ Forward to | | Remote Rx IP Address | | Remote Rx UDP Port | | |
| 1 | Serial-S1 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8000 | | |
| 2 | Serial-S2 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8000 | | |
| 3 | Serial-S3 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8000 | | |
| 4 | Serial-S4 | ▼ | ➔ enable ▼ | ▼ | 192.168.2.80 | | 8000 | | |

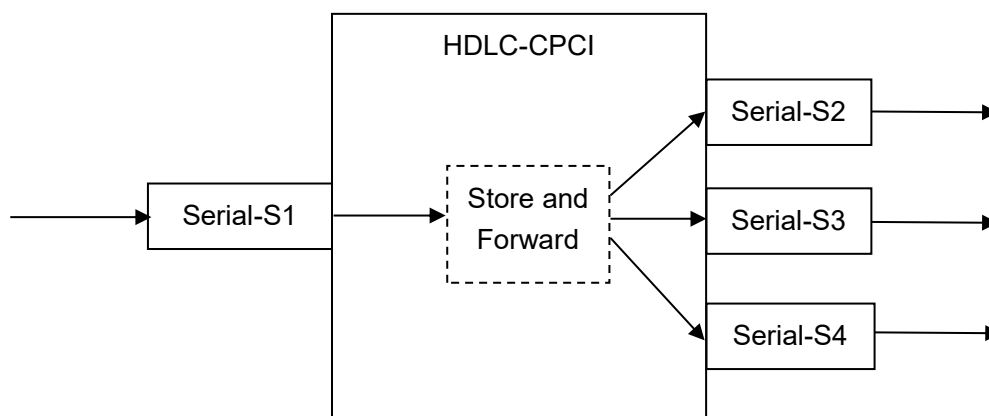
4.5 Serial to Serial Configuration

Serial to Serial can forward the input data of a specified serial port to the output of other serial ports. It is mainly used for:

- Conversion between synchronous and asynchronous serial ports;
- Serial port splitter: Divide the single serial port data into multiple channels. Unlike the common demultiplexer. Using HDLC-CPCI to implement demultiplexer, each serial port can set different baud rate and clock mode, avoiding packet loss caused by clock inconsistency.

| | Forward | S1 Egress | S2 Egress | S3 Egress | S4 Egress |
|------------|--------------|---------------------------------|--|--|--|
| S1 Ingress | ➔ Forward to | <input type="checkbox"/> Enable | <input checked="" type="checkbox"/> Enable | <input checked="" type="checkbox"/> Enable | <input checked="" type="checkbox"/> Enable |
| S2 Ingress | ➔ Forward to | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable |
| S3 Ingress | ➔ Forward to | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable |
| S4 Ingress | ➔ Forward to | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable | <input type="checkbox"/> Enable |

As shown in the above figure, the input of serial port S1 is split to the output of S2, S3 and S4. HDLC-CPCI stores and forwards the receiver, even if the baud rate and clock mode of S1, S2, S3 and S4 are different, it will not lose packets.

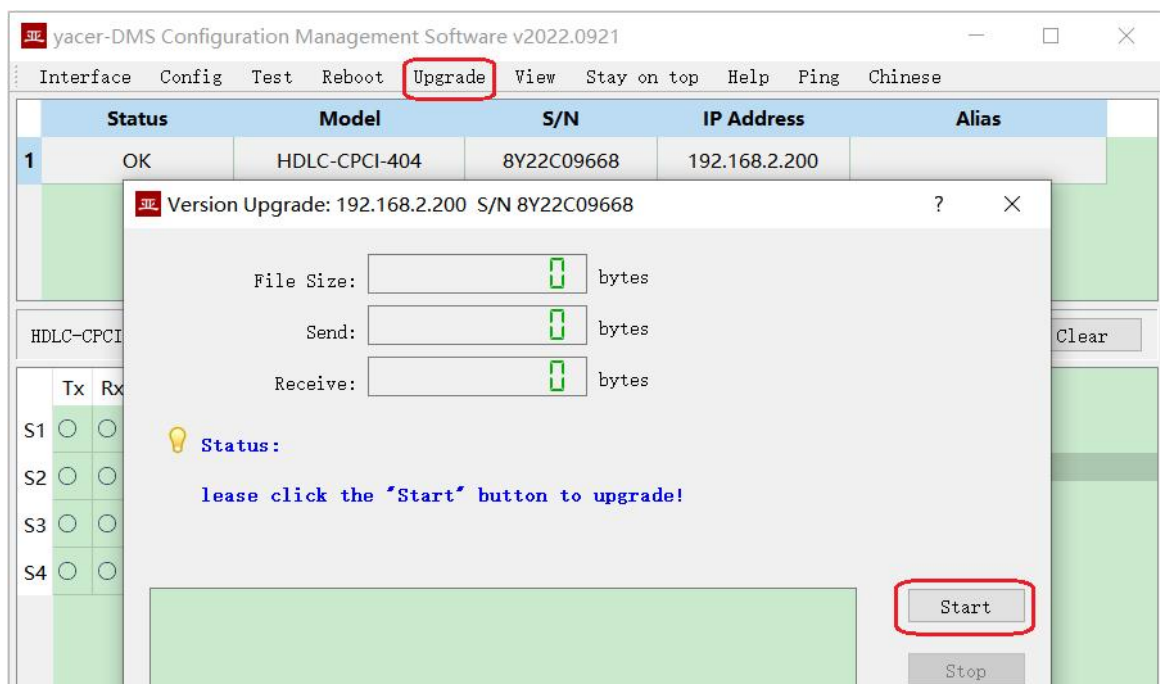


5 System Maintenance

5.1 Firmware Version Upgrade

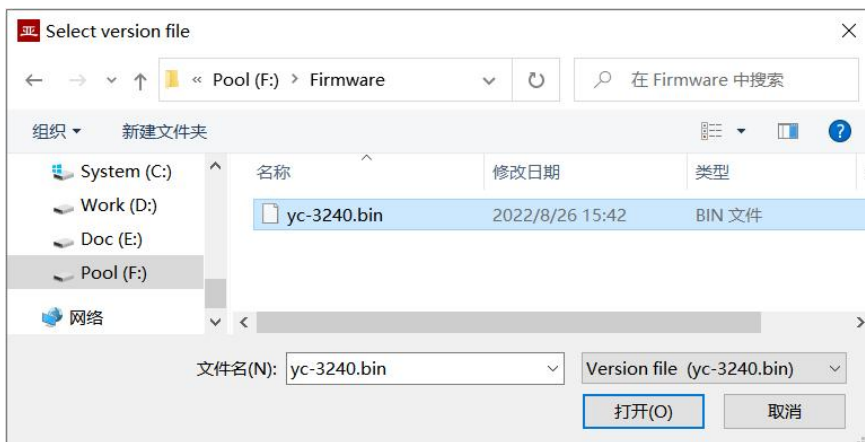
5.1.1 Start Upgrade

Click the “Upgrade” button on the toolbar to pop up the version upgrade dialog, and then click the “Start” button.



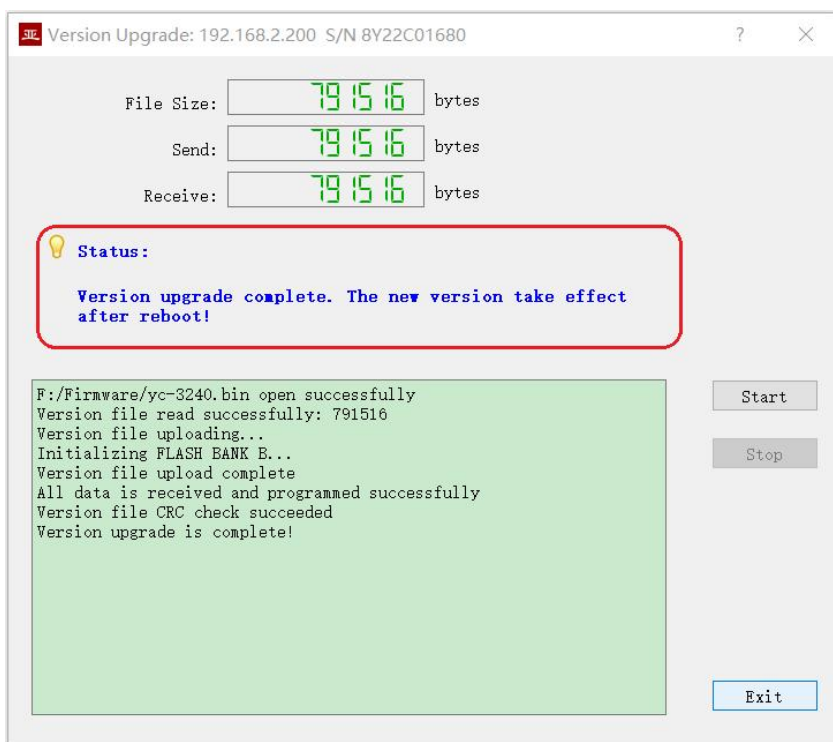
5.1.2 Select Version File

Pop up the “Select version file” dialog, and find the folder where the latest firmware version is stored, select the corresponding file, and click “Open” to start the update.



5.1.3 Complete Upgrade

When the page displays “Version upgrade complete” status, it indicates that the version upgrade is completed.



5.1.4 Re-powering takes effect

The device is re-powered. Wait for a minute or so for the new version to start taking effect.



CAUTION The device CANNOT be powered off during the waiting period.

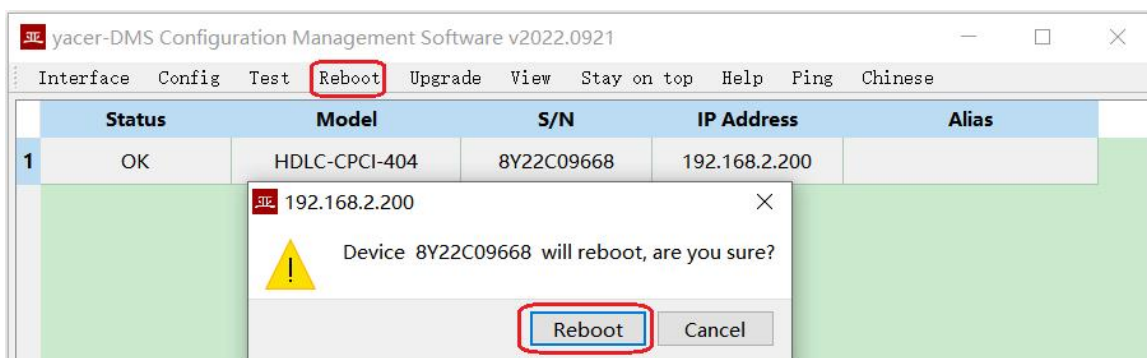
5.1.5 Confirm Upgrade

After the upgrade is completed, power up the device again, observe the version information in the statistical report, and determine whether the new version is successfully updated by the version date.



5.2 Reboot Device

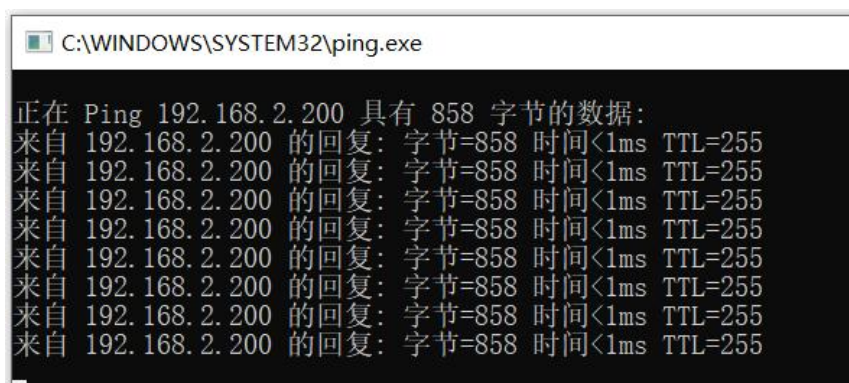
Click the "Reboot" button on the toolbar to pop up the device reboot dialog, and then click the "Reboot" button to reboot the device.



5.3 Ping

By clicking the "Ping" button on the toolbar, DMS automatically starts the ping command on the selected device to check whether the network connection between the configuration management computer and HDLC-CPCI is working properly.

Before executing the Ping command, first make sure that the IP addresses of the computer and HDLC-CPCI are in the same subnet.



6 Forwarding Function and Data Format

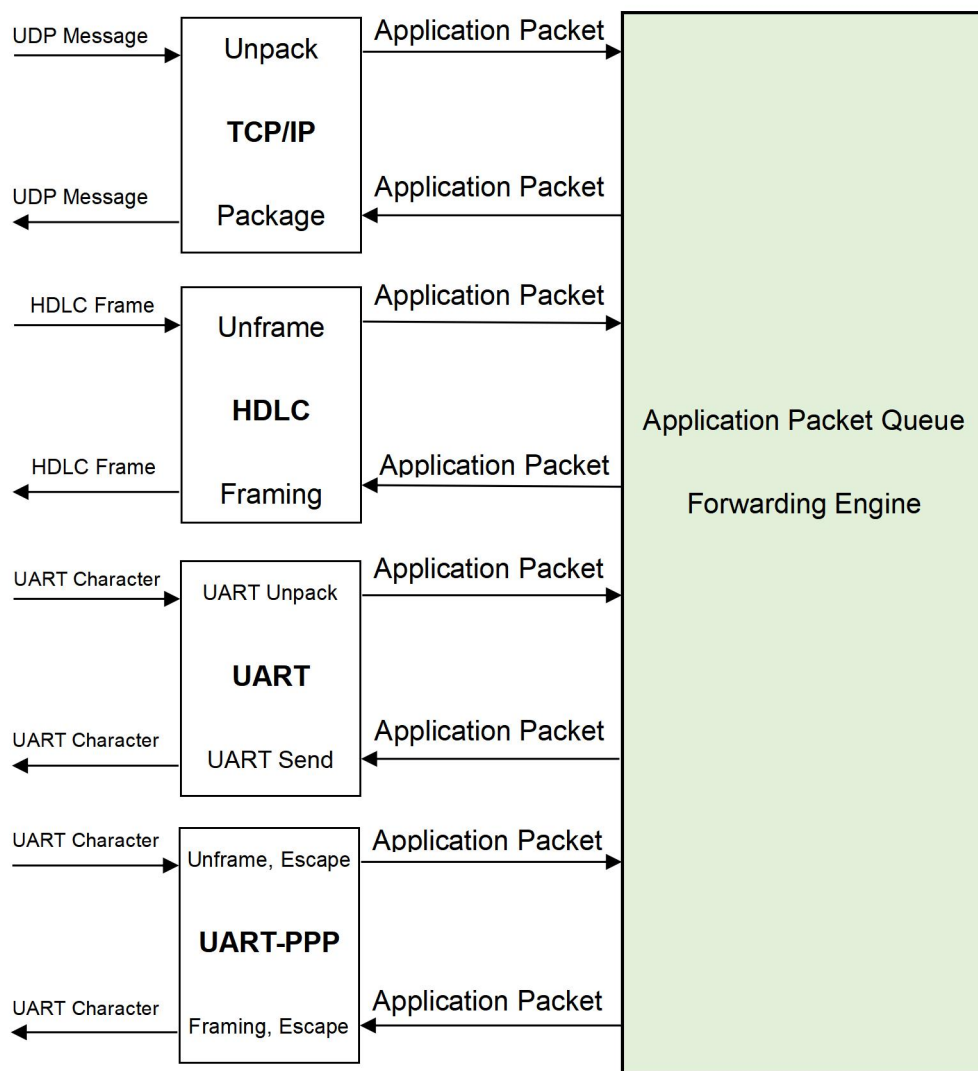
6.1 Application Packet and Conversion Model

Serial data conversion includes:

- Protocol conversion between serial port and UDP;
- Data conversion between synchronous and asynchronous serial ports.

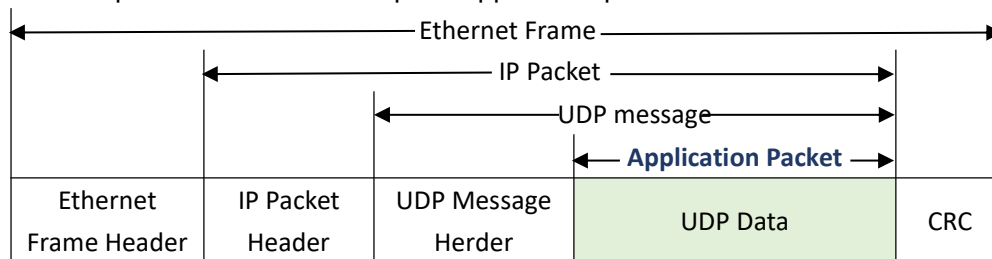
At the time of receiving, the receiving processing modules of different types of interfaces perform unpacking or decoding operations on the data, extract the application data packets, and send them to the queue of the system.

The forwarding engine of HDLC-CPCI reads the application packet queue and sends it to the sending module of each interface according to the forwarding configuration. The sending module is responsible for the framing or packing operation of the application package to generate different types of protocol packages or data frames and send them through the physical interface.



6.2 UDP Message Format

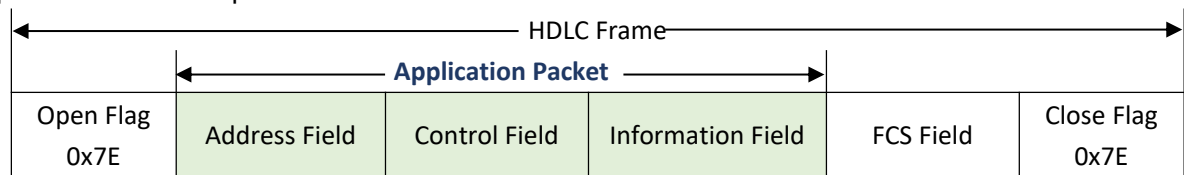
In the UDP protocol, the application packet is packaged in the data area of the UDP message. Each UDP packet contains a complete application packet.



6.3 HDLC Frame Format

A complete HDLC frame consists of several fields between the leading flag and the closing flag, including address field, control field, information field and FCS field for CRC check.

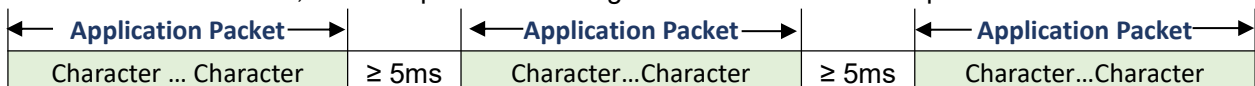
For HDLC-CPCI, instead of distinguishing between address field, control field, and information field, they are uniformly presented as application packets to the upper application to fill in and process the UART packet format



6.4 UART Data Packet

When the serial port is working in the asynchronous UART mode, there is a character stream without head or tail received from the serial port, where there is no information used to perform unpacking or deframing.

HDLC-CPCI adopts the time information for unpacking, allowing users to define the packet interval of UART. For example, if the packet interval is 5ms, when no new characters are received over 5ms, then the packet receiving is considered to be complete.

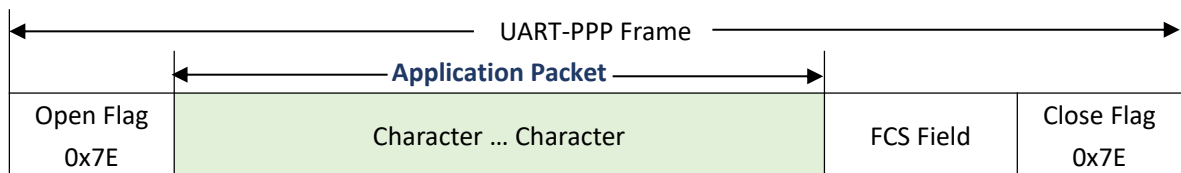


In the actual application, data transmission is not allowed during the packet interval; otherwise, it may result in a waste of communication bandwidth, and the higher the baud rate is, the more serious the waste is.

6.5 UART-PPP Frame Format

The UART-PPP working mode adopts another strategy to provide the unpacking capacity for UART. As shown in the following figure, the data sender calculates the application packet's CRC and adds the 0x7e to the head and tail as the leading and closing flags to form an UART-PPP frame.

This strategy does not require increasing the additional packet interval and can make full use of the communication bandwidth, but increases the processing complexity of both communication sides.



As the information field and FCS field may appear 0x7E, perform the character escape on such fields before transmission with the escape rules as follows:

- 0x7E: Escaped to two characters, 0x7D 0x5E;
- 0x7D: Escaped to two characters, 0x7D 0x5D;
- Other characters: No escape.

The escape operation of data send is as follows:

| Original Data | Actual Transmit Data |
|---------------|----------------------|
| 0x7E | 0x7D 0x5E |
| 0x7D | 0x7D 0x5D |
| Others | No change |

The escape operation of data receive is as follows:

| Original Data | Actual Transmit Data |
|---------------|----------------------|
| 0x7D 0x5E | 0x7E |
| 0x7D 0x5D | 0x7D |
| Others | No change |

About the Manual

- The manual is for reference only. If there is inconsistency between the manual and the actual product, the actual product shall prevail.
- We are not liable for any loss caused by the operations that do not comply with the manual.
- All the designs and software are subject to change without prior written notice. The product updates might cause some differences between the actual product and the manual. Please contact the customer service for the latest program and supplementary documentation.
- There still might be deviation in technical data, functions and operations description, or errors in print. If there is any doubt or dispute, we reserve the right of final explanation.
- Upgrade the reader software or try other mainstream reader software if the manual (in PDF format) cannot be opened.
- Please visit our website, contact the supplier or customer service if there is any problem occurring when using the device.
- If there is any uncertainty or controversy, we reserve the right of final explanation.