HDLC-USB

Portable Protocol Converter

Rev 2024.0914

HDLC-USB

Datasheet

Email: yacer@yacer.cn
Website: www.yacer.com.cn

yacer 亚册
Building Blocks of Communication

1 Overview

1.1 Introduction

The Yacer HDLC-USB portable protocol converter provides dual multi-protocol synchronous & asynchronous serial ports and single 10/100M Ethernet interface to achieve the protocol conversion between the serial port and the Ethernet interface.

With the USB-powered support, it is ultra-light and ultra-thin with vibration resistance in a very small form factor. Especially suitable for the portable, embedded application.

1.2 Features

- Single 10/100M Ethernet interface
- Dual multi-protocol synchronous/asynchronous serial port
- RS-232, RS-422, Rs-485
- Full-duplex, Half-duplex
- HDLC, TCMS, UART, Bit stream
- USB-powered support
- Industrial wide temperature
- Thin, smart, portable

1.3 Applications

- Train Communication Network (TCN)
- Train Control and Monitoring System (TCMS)
- ADS-B, Secondary Surveillance Radar (SSR)
- Air Traffic Control (ATC), Air Traffic Management (ATM)
- Synchronous and asynchronous serial data communication and conversion
- Measurement and control data acquisition
- Portable application

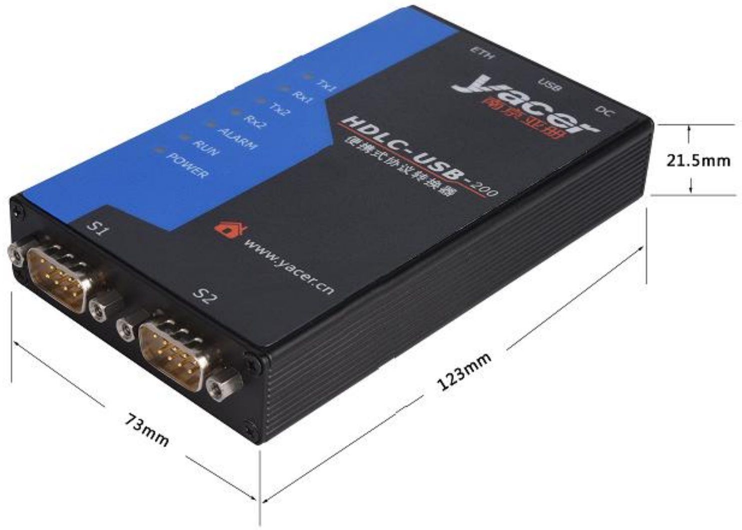
1.4 Technical Specifications

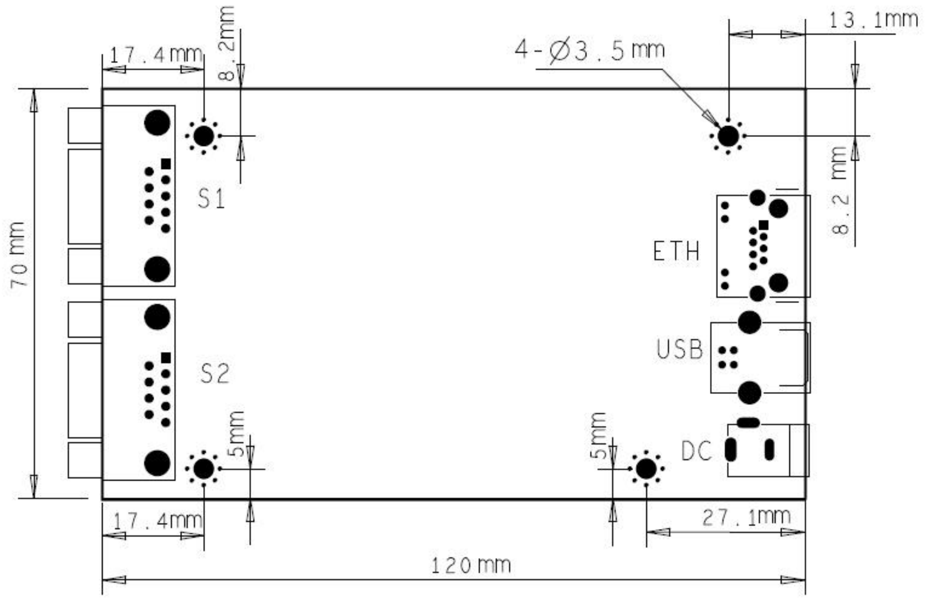
Serial Port	
Quantity	2 x D-sub 9 (Male)
Working mode	Synchronous HDLC, TCMS-HDLC, asynchronous UART, synchronous Bit Stream
Interface type	RS-232, RS-422, RS-485 Full-duplex, Half-duplex
Encoding format	NRZ, NRZI
Baud rate	NRZ ≤ 250 Kbps NRZI ≤ 64 Kbps
HDLC frame length	≤ 1470 bytes
Synchronous clock	Normal, slave and master clock mode
ESD protection	± 15 KV
Ethernet Interface	
Quantity	1 x RJ45
Rate	10/100 Mbps, Auto MDI/MDI-X
Protocol	TCP/IP
Programming interface	UDP Server, UDP Client, Unicast/Multicast/Broadcast
USB Interface	
Interface	1 x type-B USB interface (female), only for the power supply
Configuration Management	
Configuration tool	yacer-DMS configuration management software
Configuration interface	Ethernet interface
Power Requirements	
Input voltage	+5V DC
Power consumption	< 2W
Power interface	USB interface DC-5.5-2.1mm interface
Mechanical Characteristics	
Dimensions	HxWxD: 21.5 mm x 73 mm x 123 mm
Weight	200g
Enclosure	Portable aluminum alloy fully-enclosed enclosure
Operating Environment	
Operating temperature	-40 ~ +70℃
Storage temperature	-40 ~ +85℃
Operating humidity	5 ~ 95% RH (no condensation)

1.5 Order Information

Product Model	Serial Port	Ethernet Interface
HDLC-USB-200	2 x RS-232/422/485 synchronous & asynchronous serial ports	1 x 10/100M

1.6 Mechanical Dimensions



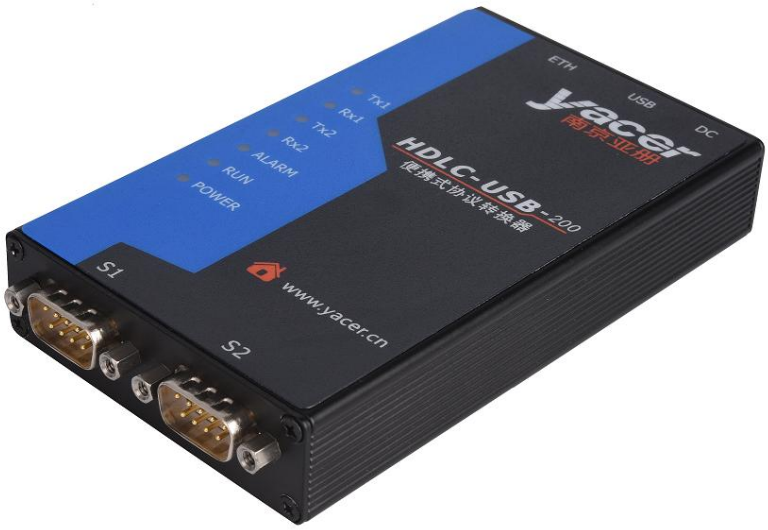


2 Hardware Interface

2.1 Appearance

S1 and S2 serial ports are located on one end of the product while Ethernet interface (ETH), USB interface and DC power interface are located on the other end.

Silk screen and LED indicators are located on the front.



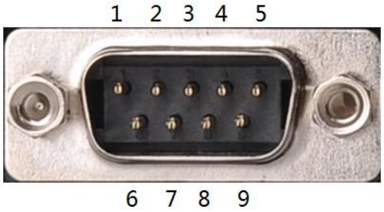
2.2 LED indicators

LED	Description
TX1	Sending indicator of the serial port S1, flashing after one frame data is transmitted successfully
RX1	Receiving indicator of the serial port S1, flashing after one frame data is received successfully
TX2	Sending indicator of the serial port S2, flashing after one frame data is transmitted successfully
RX2	Receiving indicator of the serial port S2, flashing after one frame data is received successfully
ALARM	Alarm indicator, on when the device is not ready to start or in case of failure, and constantly off during normal operation
RUN	Running indicator, flashing during normal operation
POWER	Power indicator, constantly on after powered

2.3 Serial Port

HDLC-USB provides dual DB9 male serial interface with support for the RS-232/422/485 protocol as well as the synchronous & asynchronous working modes.

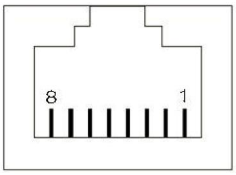
DB-9 male	RS-232	RS -422 RS-485 full-duplex	RS-485 half-duplex
1	RxData	RxData+	
2	RxClock	RxClock+	
3	GND	GND	GND
4	TxClock	TxClock+	CLK+
5	TxDatat	TxDdata+	Data+
6		RxData-	
7		RxClock-	
8		TxClock-	CLK-
9		TxData-	Data-



2.4 Ethernet Interface

RJ-45 interface, 10/100M Ethernet, support for auto MDI/MDI-X crossover.

RJ-45	Signal
1	Tx+
2	Tx-
3	Rx+
6	Rx-



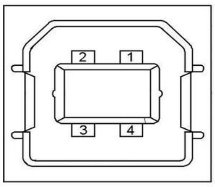
2.5 Power Interface

+5VDC, from the USB or DC power interface.

2.5.1 USB power interface

Type-B USB interface, only for the power supply.

Pin Number	Signal	Description
1	VBUS	Power supply +5V
2	D-	Data-, unused
3	D+	Data+, unused
4	GND	
Shell	Shield	



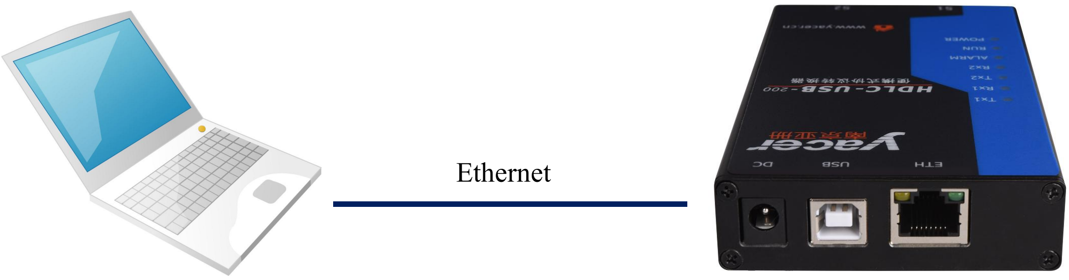
2.5.2 DC: power interface

HDLC-USB operates with the +5V DC power supply.

The interface adopts the universal DC 5.5-2.5mm power socket.

3 Building Configuration Environment

Connect the management computer and HDLC-USB over the network cable, as shown below.

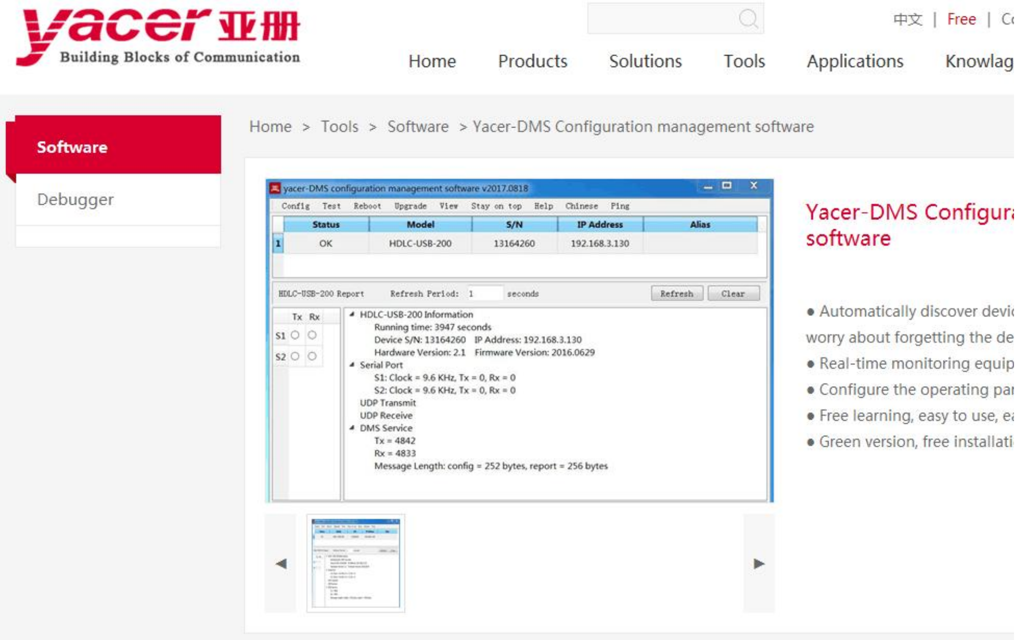


4 yacer-DMS Configuration Management Software

4.1 Get the Configuration Management Software yacer-DMS

4.1.1 Official website of yacer

Visit the official website of yacer http://en.yacer.cn to enter the “Tools” channel, and open the yacer-DMS software page to download the latest version of the software.


yacer 亚册
Building Blocks of Communication
中文 | Free | Co
Home
Products
Solutions
Tools
Applications
Knowlag
Software
Debugger
Home > Tools > Software > Yacer-DMS Configuration management software
yacer-DMS configuration management software v2017.0818
Config Test Reboot Upgrade View Stay on top Help Chinese Ping
HDLC-USB-200 Report Refresh Period: 1 seconds
Refresh
Clear
Tx Rx
S1 ○ ○
S2 ○ ○

	Status	Model	S/N	IP Address	Alias
1	OK	HDLC-USB-200	13164260	192.168.3.130	

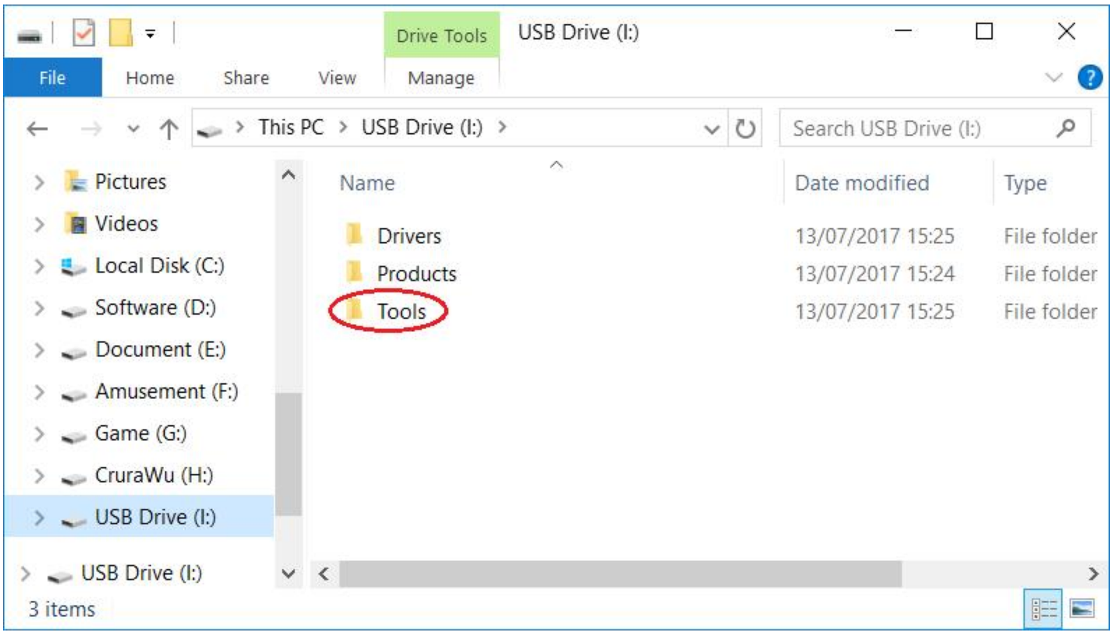
```
◢ HDLC-USB-200 Information
    Running time: 3947 seconds
    Device S/N: 13164260   IP Address: 192.168.3.130
    Hardware Version: 2.1   Firmware Version: 2016.0629
◢ Serial Port
    S1: Clock = 9.6 KHz, Tx = 0, Rx = 0
    S2: Clock = 9.6 KHz, Tx = 0, Rx = 0
  UDP Transmit
  UDP Receive
◢ DMS Service
    Tx = 4842
    Rx = 4833
    Message Length: config = 252 bytes, report = 256 bytes
```

Yacer-DMS Configura
software

- Automatically discover devi
worry about forgetting the de
- Real-time monitoring equip
- Configure the operating pa
- Free learning, easy to use, e
- Green version, free installati

4.1.2 Accompanied USB disk

Insert the accompanied USB disk to the PC, open it and double-click to enter the “Tools” folder.

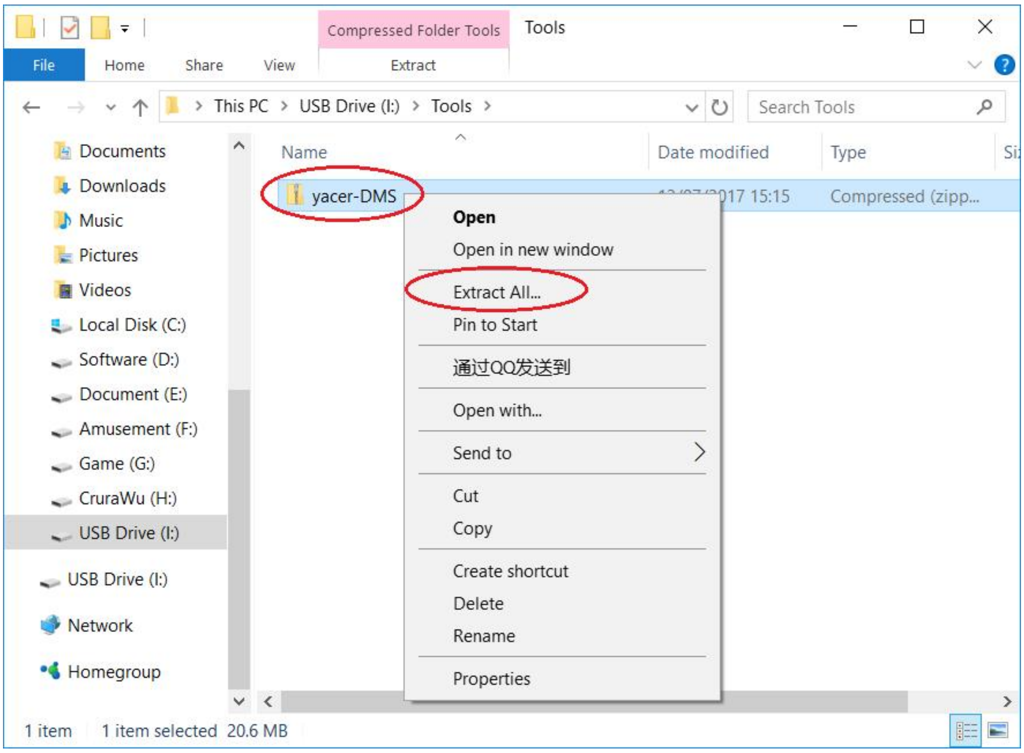


Name	Date modified	Type
Drivers	13/07/2017 15:25	File folder
Products	13/07/2017 15:24	File folder
Tools	13/07/2017 15:25	File folder

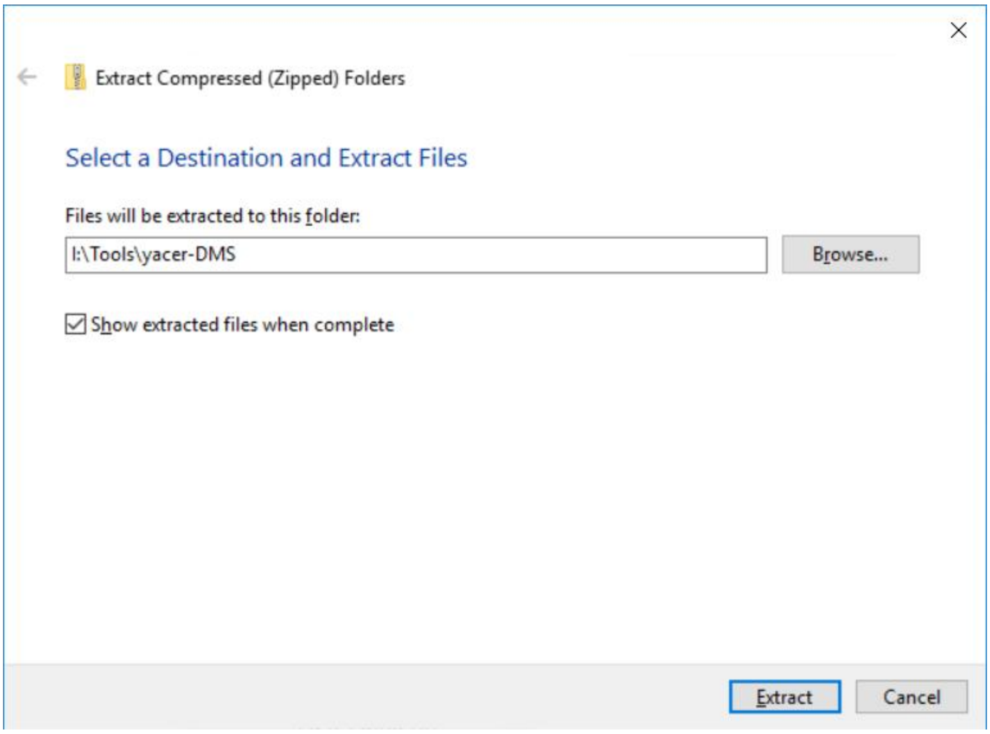
4.2 Run yacer-DMS

4.2.1 Unzip the file

- After getting the yacer-DMS.zip compressed file, first unzip it.



- With the Extract Compressed Folder dialog box popping up, set the path for saving extracted files in the box and click on the “Extract” button once set up.


Extract Compressed (Zipped) Folders

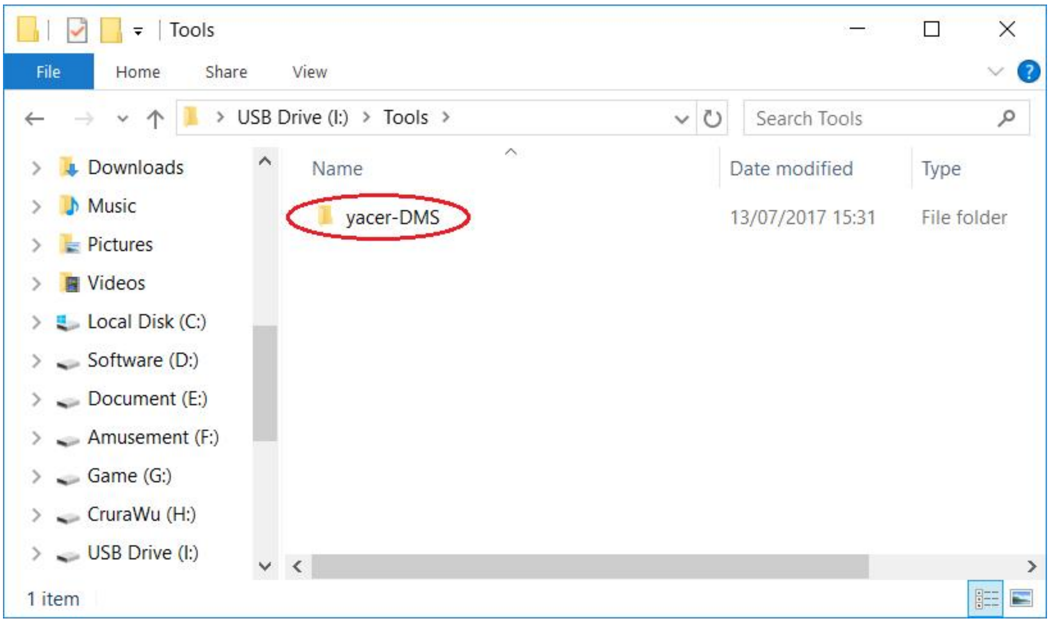
Select a Destination and Extract Files

Files will be extracted to this folder:
I:\Tools\yacer-DMS

☑ Show extracted files when complete

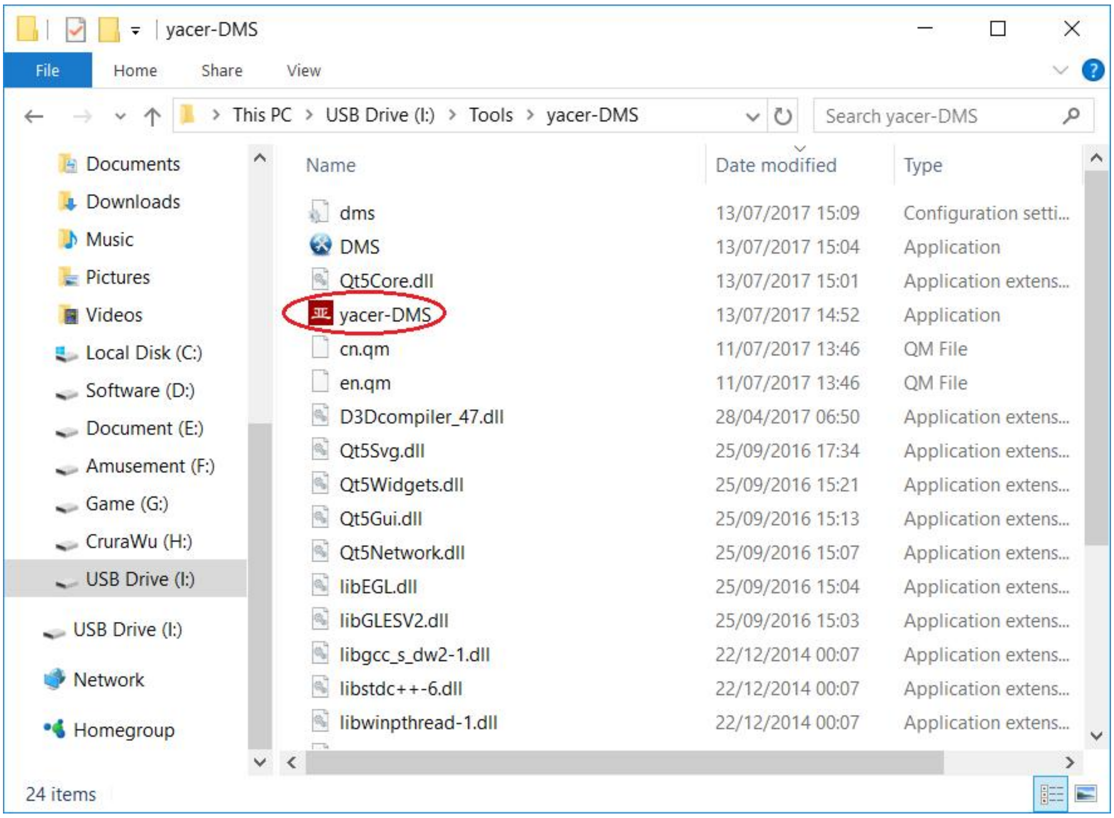
4.2.2 Run yacer-DMS

- Locate the decompressed folder according to the path for extracting the compressed files and click on it.



Name	Date modified	Type
yacer-DMS	13/07/2017 15:31	File folder

- Locate the yacer-DMS.exe file under directory and double-click on this file to run the configuration software.

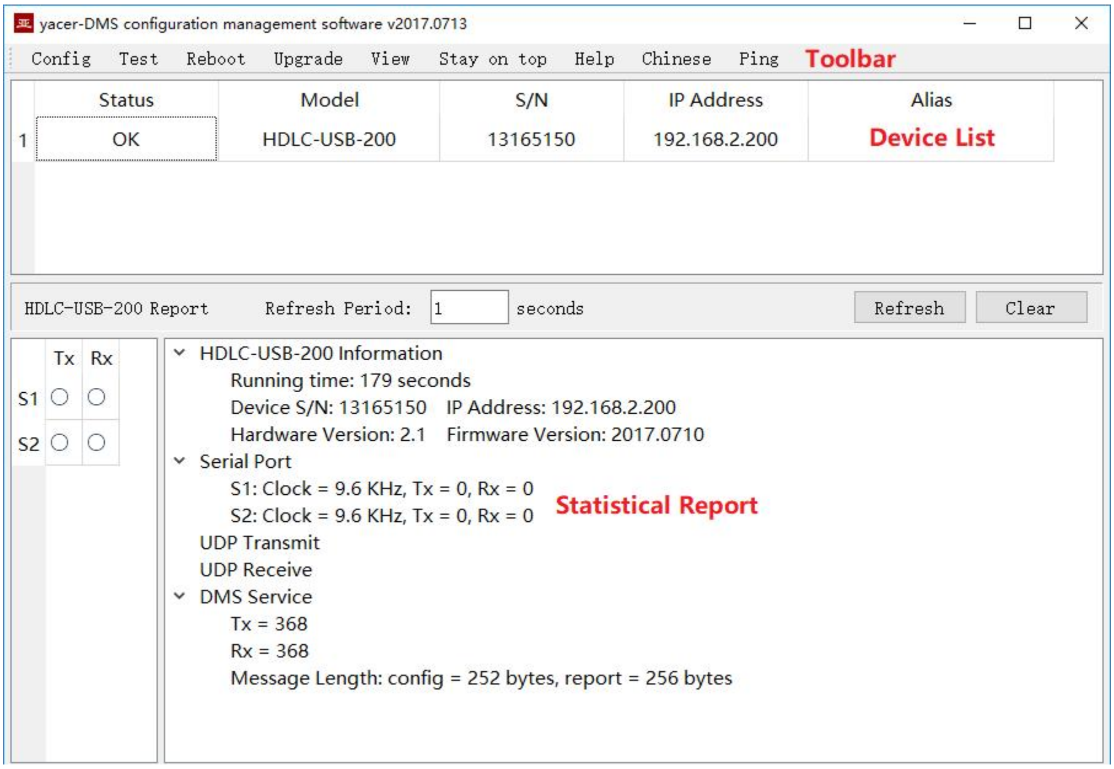


Name	Date modified	Type
dms	13/07/2017 15:09	Configuration setti...
DMS	13/07/2017 15:04	Application
Qt5Core.dll	13/07/2017 15:01	Application extens...
yacer-DMS	13/07/2017 14:52	Application
cn.qm	11/07/2017 13:46	QM File
en.qm	11/07/2017 13:46	QM File
D3Dcompiler_47.dll	28/04/2017 06:50	Application extens...
Qt5Svg.dll	25/09/2016 17:34	Application extens...
Qt5Widgets.dll	25/09/2016 15:21	Application extens...
Qt5Gui.dll	25/09/2016 15:13	Application extens...
Qt5Network.dll	25/09/2016 15:07	Application extens...
libEGL.dll	25/09/2016 15:04	Application extens...
libGLESV2.dll	25/09/2016 15:03	Application extens...
libgcc_s_dw2-1.dll	22/12/2014 00:07	Application extens...
libstdc++-6.dll	22/12/2014 00:07	Application extens...
libwinpthread-1.dll	22/12/2014 00:07	Application extens...

4.3 Main Window of DMS

Below is the main window of the configuration management software yacer-DMS, including three parts:

- Toolbar: Function operation buttons
- Device List: Displaying the basic information and running status of the on-line device
- Statistical Report: Displaying the receive/transmit indication & statistics, and device details of the specified device


yacer-DMS configuration management software v2017.0713
Config Test Reboot Upgrade View Stay on top Help Chinese Ping
Toolbar
Device List
HDLC-USB-200 Report
Refresh Period:
seconds
Refresh
Clear
Statistical Report

	Status	Model	S/N	IP Address	Alias
1	OK	HDLC-USB-200	13165150	192.168.2.200	

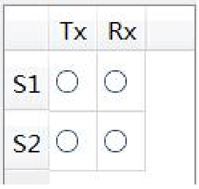
Refresh Period: 1 seconds

	Tx	Rx
S1	○	○
S2	○	○

```
HDLC-USB-200 Information
    Running time: 179 seconds
    Device S/N: 13165150   IP Address: 192.168.2.200
    Hardware Version: 2.1   Firmware Version: 2017.0710
Serial Port
    S1: Clock = 9.6 KHz, Tx = 0, Rx = 0
    S2: Clock = 9.6 KHz, Tx = 0, Rx = 0
UDP Transmit
UDP Receive
DMS Service
    Tx = 368
    Rx = 368
    Message Length: config = 252 bytes, report = 256 bytes
```

4.4 Statistical Report

4.4.1 Receive/transmit indication



	Tx	Rx
S1	○	○
S2	○	○

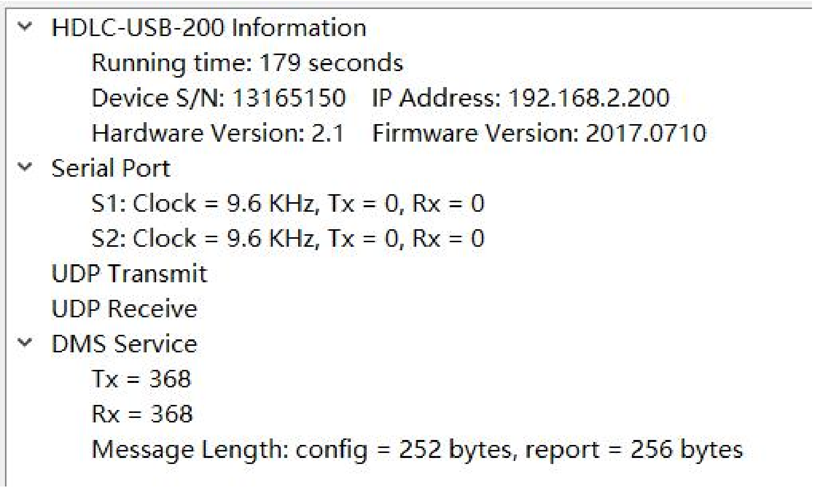
Transmit: Each time the corresponding serial port transmits one frame data, the transmit indicator flashes once.

Receive: Each time the corresponding serial port receives one frame data, the receive indicator flashes once.

4.4.2 Information display

The information display area is located on the right side of the statistical report, showing the following contents:

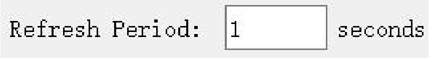
- Device Information: Running time, serial number, IP address, version number
- Serial Port: Receive/transmit statistics of all serial ports
- UDP Transmit: Displaying the relevant transmitted packets of the UDP Client for each enabled serial port to UDP entry
- UDP Receive: Displaying the relevant received packets of the UDP Server for each enabled UDP to serial port entry
- DMS Service: Displaying the information receive/transmit statistics of the configuration management between the device and the configured management computer



```
HDLC-USB-200 Information
    Running time: 179 seconds
    Device S/N: 13165150   IP Address: 192.168.2.200
    Hardware Version: 2.1   Firmware Version: 2017.0710
Serial Port
    S1: Clock = 9.6 KHz, Tx = 0, Rx = 0
    S2: Clock = 9.6 KHz, Tx = 0, Rx = 0
UDP Transmit
UDP Receive
DMS Service
    Tx = 368
    Rx = 368
    Message Length: config = 252 bytes, report = 256 bytes
```

4.4.3 Control Widget

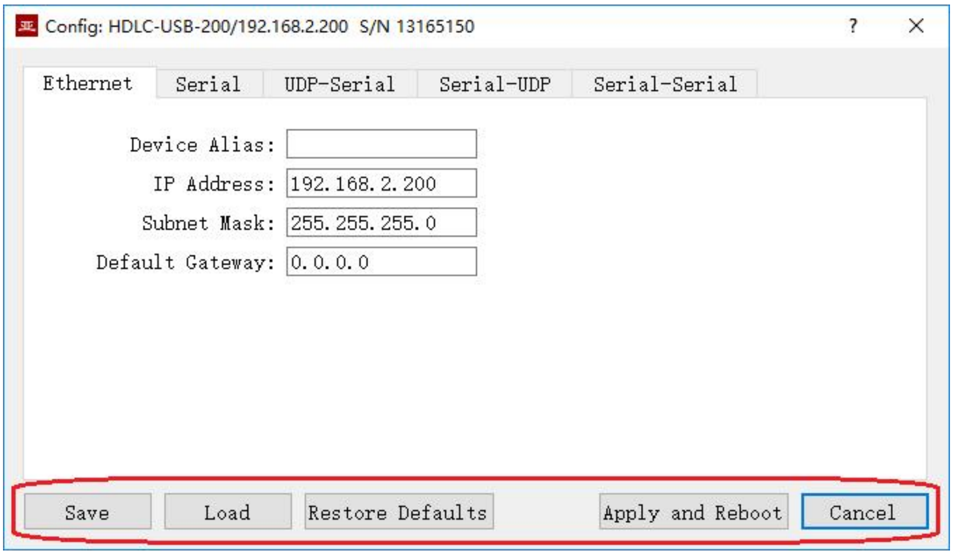
The statistical report includes the following control widgets:

Control Widget	Function
 Refresh Period: 1 seconds	Set the refresh period of report

Refresh	Manually refresh the statistical report
Clear	Clear the statistical report

4.5 Configure Device

Click on the Config button on the toolbar or double-click on the specified device in the device list; yacer-DMS pops up the configuration dialog. According to the interface and function, the dialog divides the configuration item into several configuration pages.


Config: HDLC-USB-200/192.168.2.200 S/N 13165150

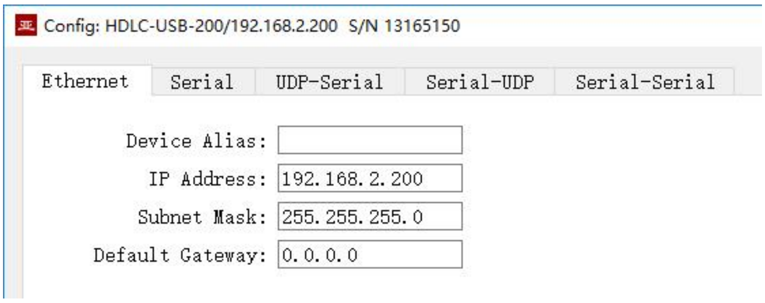
Device Alias:
IP Address: 192.168.2.200
Subnet Mask: 255.255.255.0
Default Gateway: 0.0.0.0

The following operation buttons are located at the bottom of the dialog:

Button	Function
Save	Save the configuration content of the current dialog to the file
Load	Open the configuration file and read the refreshed configuration dialog content
Restore Defaults	Refresh the dialog content with the device's default factory configuration
Apply and Reboot	Write the configuration content of the dialog into the device and restart the device to bring the configuration into effect
Cancel	Cancel the current configuration operation

5 Function and Configuration

5.1 Ethernet Interface Configuration



Config: HDLC-USB-200/192.168.2.200 S/N 13165150

Device Alias:

IP Address: 192.168.2.200

Subnet Mask: 255.255.255.0

Default Gateway: 0.0.0.0

5.1.1 Device alias

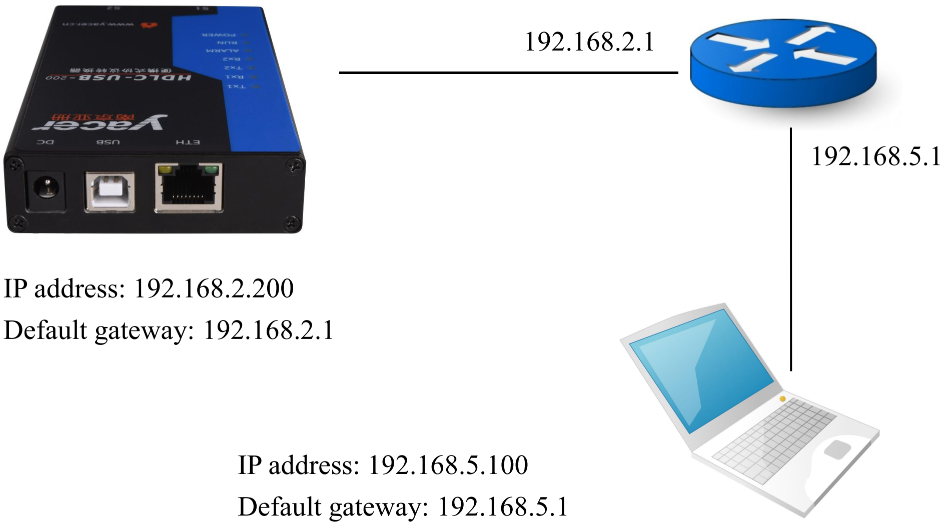
It allows users to set an alias for the HDLC-USB, thus adding description to the device or helping to remember the identification.

5.1.2 IP address and default gateway

Configuration of the IP address and subnet mask is shown above; the default gateway defaults to be 0.0.0.0, representing that there is no gateway configuration.

If HDLC-USB needs to communicate with the host on other subnet, it must rely on an external router. At this time, the HDLC-USB'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-USB 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-USB and PC need to set the IP address of the connected router port to the default gateway of this device.



192.168.2.1
192.168.5.1
IP address: 192.168.2.200
Default gateway: 192.168.2.1
IP address: 192.168.5.100
Default gateway: 192.168.5.1

5.2 Serial Port Configuration

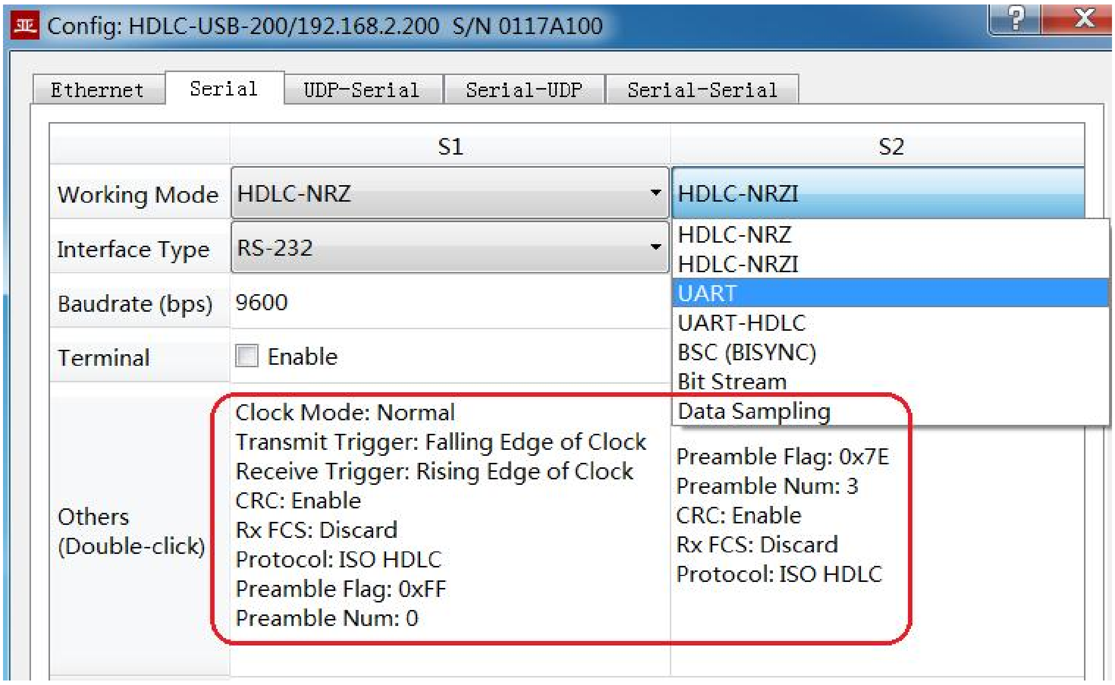
5.2.1 Select the working mode of the serial port

Serial S1 and S2 are synchronous and asynchronous serial ports, with support for the synchronous and asynchronous working modes.

Working Mode		Description
Asynchronous	UART	universal asynchronous serial, equivalent to the serial port on the common computer
	UART- HDLC	UART-based similar HDLC communication protocol
Synchronous	HDLC-NRZ	Synchronous HDLC protocol based on the NRZ encoding
	HDLC-NRZI	Synchronous HDLC protocol based on the NRZI encoding
	Bit Stream	Serial Bit data based on the receive clock sampling
	Data sampling	Original sampling of the received data based on the 8x baud rate

Users can select the desired working mode from the “working mode” combobox. Due to different parameter configuration of each working mode, contents of the “Others” cell will be adjusted automatically according to the determined working mode.

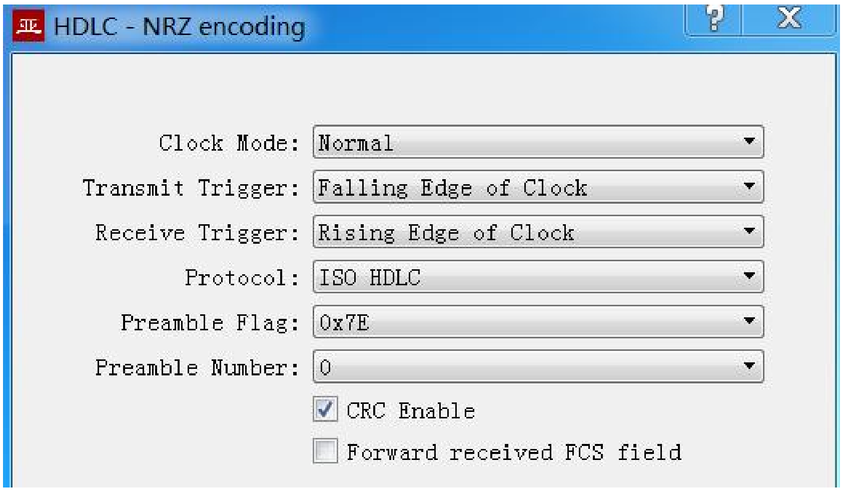
If further configuration of working parameters of the selected working mode is required, mouse double-click on the “Others” cell to pop up the parameter configuration dialog.


Config: HDLC-USB-200/192.168.2.200 S/N 0117A100

	S1	S2
Working Mode	HDLC-NRZ	HDLC-NRZI
Interface Type	RS-232	
Baudrate (bps)	9600	
Terminal	□ Enable	
Others (Double-click)	Clock Mode: Normal Transmit Trigger: Falling Edge of Clock Receive Trigger: Rising Edge of Clock CRC: Enable Rx FCS: Discard Protocol: ISO HDLC Preamble Flag: 0xFF Preamble Num: 0	Preamble Flag: 0x7E Preamble Num: 3 CRC: Enable Rx FCS: Discard Protocol: ISO HDLC

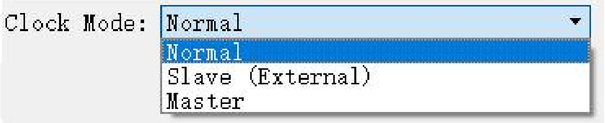
5.2.2 HDLC-NRZ parameter configuration

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



Clock Mode: Normal
Transmit Trigger: Falling Edge of Clock
Receive Trigger: Rising Edge of Clock
Protocol: ISO HDLC
Preamble Flag: 0x7E
Preamble Number: 0
☑ CRC Enable
□ Forward received FCS field

5.2.2.1 Clock mode

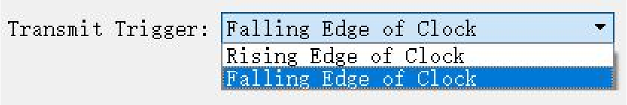

Clock Mode: Normal

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

Clock Mode	Transmit Clock	Receive Clock
Normal	Generation from the local device, TxC output	Generation from the opposite device, RxC input
Slave Clock (External)	Generation from the opposite device, RxC input TxC output synchronizes with RxC automatically	Generation of the opposite device, RxC input
Master Clock	Generation of the local device, TxC output	Generation of the local device, ignoring the RxC clock

The slave clock mode is also called as the external clock working mode. When the opposite device is the DCE, HDLC-USB is often configured as the slave clock mode and transmits data with the clock provided by the DCE, ensuring the data transmission across the whole network based a clock and avoiding packet loss concerns caused due to different clock sources.

5.2.2.2 Transmit trigger

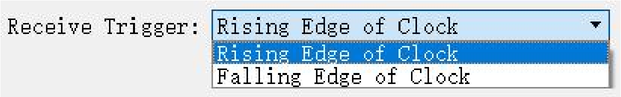

Transmit Trigger: Falling Edge of Clock

Transmit trigger defines the generation clock edge of the new data bit:

- Falling edge of clock: A new data bit is generated on the falling edge of clock
- Rising edge of clock: A new data bit is generated on the rising edge of clock

During the ATC communication, falling edge of clock is generally chosen to trigger the new data transmission. In some custom development applications, sometimes users will adopt the rising edge to trigger the new data transmission.

5.2.2.3 Receive trigger


Receive Trigger: Rising Edge of Clock

Receive trigger defines the sampling clock edge of the serial port receive data:

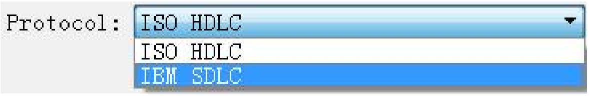
- Rising edge of clock: Data on the RxD line is read on the rising edge of the RxC signal
- Falling edge of clock: Data on the RxD line is read on the falling edge of the RxC signal

During the ATC communication, receive trigger must be configured as the rising edge of clock to ensure correct reading of data as it takes time to stabilize the new data that is triggered with the falling edge.

The local receive trigger configuration is determined according to the transmit trigger of the remote terminal device:

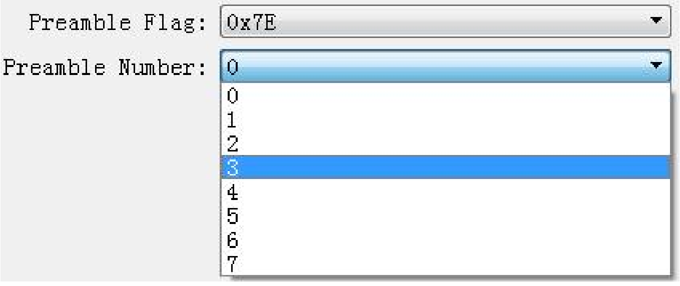
Remote Transmit Trigger	Local Receive Trigger
Falling edge of clock	Rising edge of clock
Rising edge of clock	Falling edge of clock

5.2.2.4 Protocol


Protocol: ISO HDLC

- ISO HDLC: ISO HDLC protocol standard
- IBM SDLC: SDLC protocol standard

5.2.2.5 Leading Sign and Number



Preamble Flag: 0x7E

Preamble Number: 0

For the full-duplex applications, leading is always not required with the number of leadings set to be zero (no leading).

For the full-duplex applications, leading is always not required with the number of leadings set to be zero (no leading).

5.2.2.6 CRC enable

☑ CRC Enable

By default, the HDLC protocol communication should enable the CRC function.

CRC Enable	Data Transmit	Data Receive
☑ CRC Enable	HDLC-USB automatically calculates the CRC data, and adds a 2-byte FCS field at the end of data	Check CRC of the receive frame and discard the data frame with the check failed
□ CRC Enable	No CRC calculation or FCS field adding	No CRC check and receive all frames

5.2.2.7 Forward received FCS field

□ Forward received FCS field

This configuration is only effective with CRC enable.

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

By default, this option is not checked; HDLC-USB will discard the 2-byte FCS field at the end of data and only forward the user data after the receive HDLC frame check is passed.

If this option is checked, then forward the user data and FCS field.

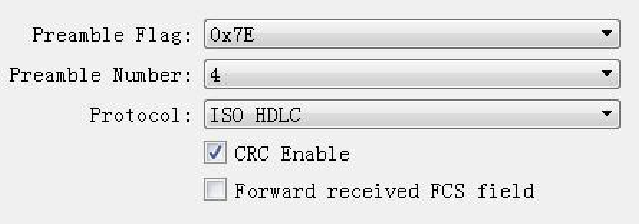
Opening Flag	Address Field	Control Field	Information Field	FCS Field	Closing Flag
0x7E	1 byte	1 byte	Variable length	CRC 2 bytes	0x7E
0x7E	User data			CRC 2 bytes	0x7E

5.2.3 HDLC-NRZI parameter configuration

HDLC-NRZI is mainly used for the train communication.

Unlike the NRZ encoding format, the NRZI encoding format data contains clock information, which only requires that the baud rate of the both communication sides should be the same, instead of the clock mode, transmit trigger, receive trigger and other parameters.

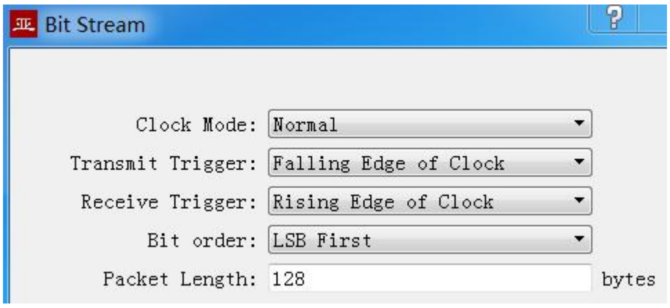
The below figure describes the HDLC-NRZI parameter configuration and the parameter function and configure same with HDLC-NRZ mode.



Preamble Flag: 0x7E
Preamble Number: 4
Protocol: ISO HDLC
☑ CRC Enable
□ Forward received FCS field

5.2.4 Bit Stream parameters

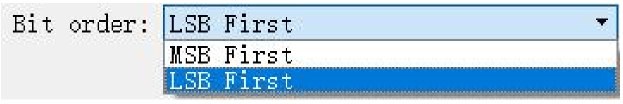
The rising or falling edge of each clock cycle samples the 1bit data on the data line, which forms a UDP message and sends to the destination IP after receiving a byte with the packet length by forming a byte with each 8bit.



Clock Mode: Normal
Transmit Trigger: Falling Edge of Clock
Receive Trigger: Rising Edge of Clock
Bit order: LSB First
Packet Length: 128 bytes

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.



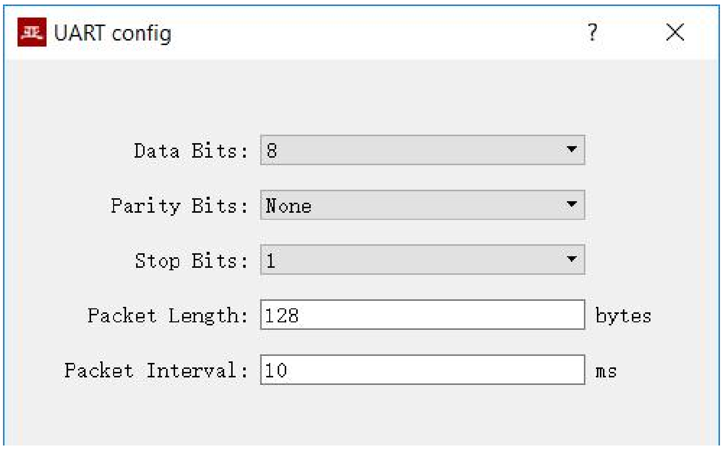
Bit order: LSB First

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

5.2.5 UART parameter configuration

Asynchronous UART is a character stream communication; data bits, parity bits and stop bits define the basic working parameters of the asynchronous serial port, which must be configured identically to the opposite terminal device.

Generally, data bits are defined as 8, i.e. one byte, and UART corresponds to the byte stream communication.


UART config

Data Bits: 8

Parity Bits: None

Stop Bits: 1

Packet Length: 128 bytes

Packet Interval: 10 ms

When the UART byte stream is converted into a UDP message or HDLC frame, it is too costly and inefficient if each byte is converted to a UDP message for transmission.

To improve efficiency, the protocol converter buffers the received byte stream and forms a number of buffered bytes into a UDP message to send, of which this process is called as packet.

Packets are controlled with two parameters, namely the packet length and the packet interval.

5.2.5.1 Packet length

For example, if the packet length is set to 128 bytes, then it will form a UDP message to send after UART receives the full 128 bytes.

5.2.5.2 Packet interval

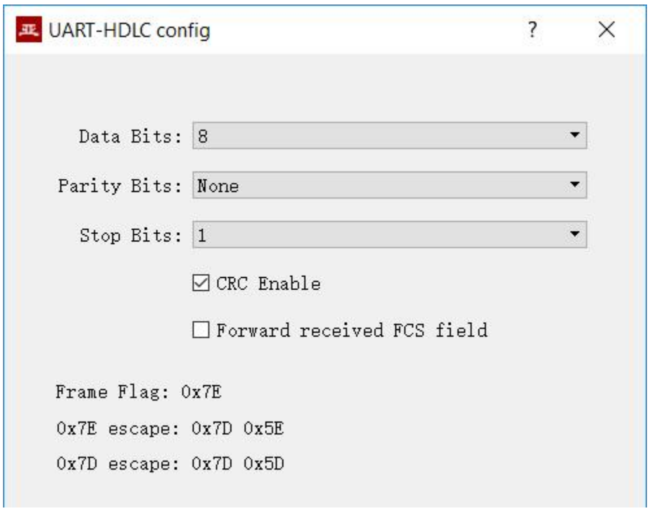
Packet improves the UDP transmission efficiency, but also introduces new problems.

As shown in the above example again, if the transmitter sends 100 bytes and stops sending, the receiver will always be waiting as the packet length is less than 128 bytes.

In order to solve the problem of waiting, users need to configure the packet interval parameter, such as set to 10ms; then data received by UART will wait for a maximum of 10ms. In case of 10ms timeout, it will convert the buffer data into a UDP message to send no matter whether it has received the full 128 bytes.

5.2.6 UART-HDLC parameter configuration

The UART-HDLC working mode is a custom protocol by yacer which form the asynchronous HDLC frame on the basis of the normal UART communication by packaging the byte stream. Therefore, the asynchronous serial port can perform the packet-based communication with the UDP message and synchronous HDLC frame.



Data Bits: 8

Parity Bits: None

Stop Bits: 1

☑ CRC Enable

□ Forward received FCS field

Frame Flag: 0x7E

0x7E escape: 0x7D 0x5E

0x7D escape: 0x7D 0x5D

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

opening Flag	Information Field	FCS Field	closing Flag
0x7E	2-1470 bytes of data	2-byte CRC data	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 transmit 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:

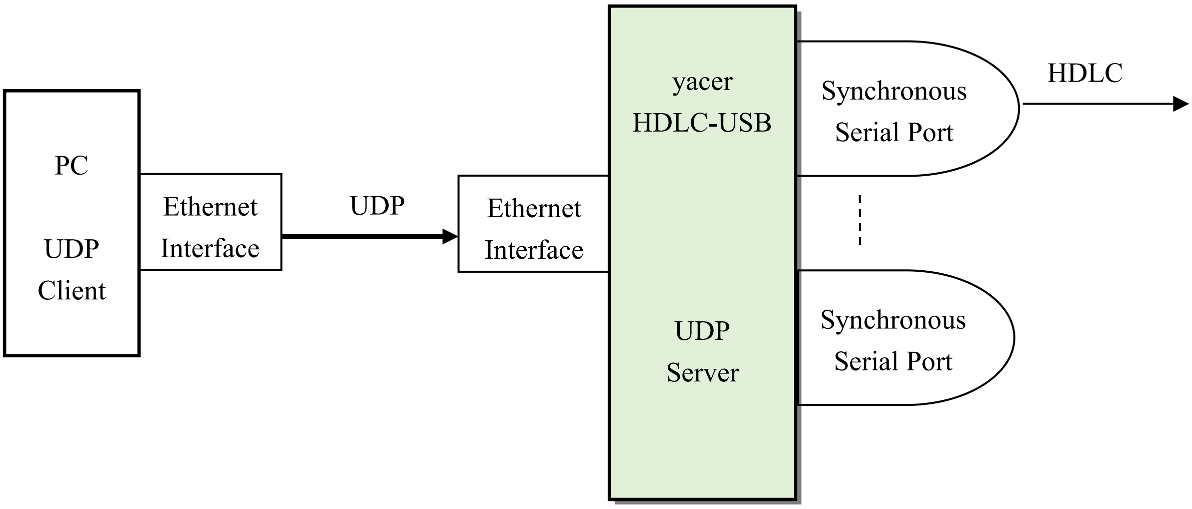
Actual Receive Data	Data
0x7D 0x5E	0x7E
0x7D 0x5D	0x7D
Others	No change

5.3 UDP to Serial Conversion

5.3.1 Application model

With HDLC-USB, PC or server can realize the transmit function of the synchronous HDLC serial port.

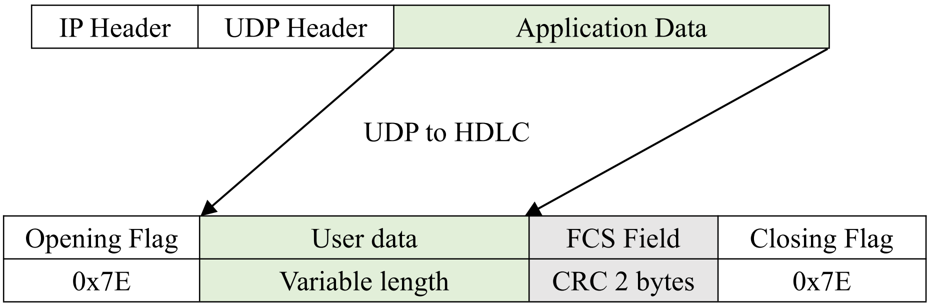
The typical application is shown as follows. PC sends a UDP message over the Ethernet interface as the UDP Client, and HDLC-USB sends it out from the synchronous serial port after converting the received UDP message into the HDLC frame.



5.3.2 Protocol Conversion

The most typical UDP-to-HDLC application is shown below. HDLC-USB loads the UDP application data into the user data area of the HDLC frame, and then calculates CRC and populates the FCS field to form a complete HDLC frame to send.

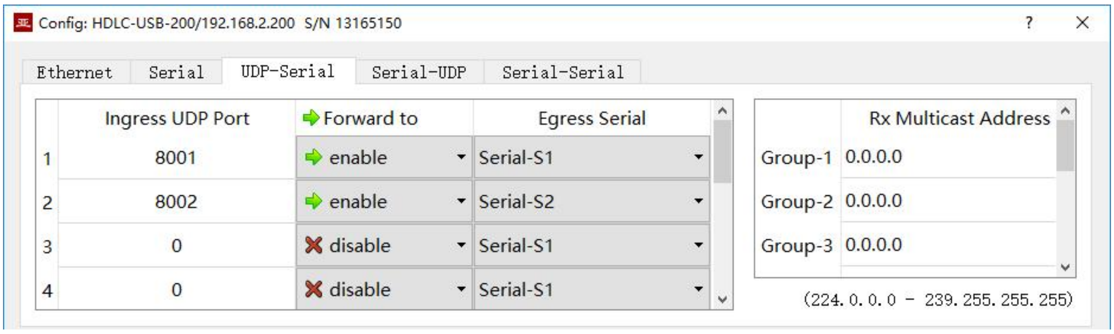
In order to reduce the calculation load of PC and the user programming complexity, normally, the FCS field of HDLC is not included in the UDP message, which is populated through HDLC-USB calculation.



5.3.3 Parameter configuration

Set the UDP to serial port. Each row represents the forwarding entry from a UDP port to the serial port with three forwarding strategies to be achieved:

- Forwarding: Data received by the specified UDP port can be forwarded to the specified serial port.
- Multiplexer: Data received by several different UDP ports can be forwarded to the same serial port.
- Demultiplexer: Data received from the same UDP port can be forwarded to the different serial ports.

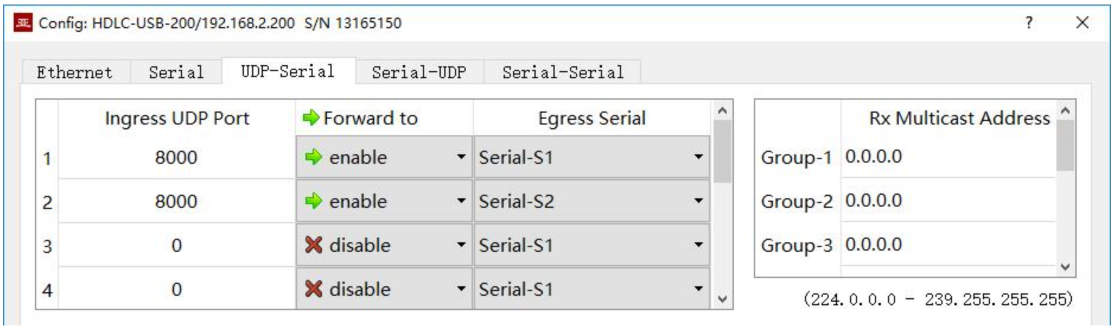


	Ingress UDP Port	Forward to	Egress Serial
1	8001	enable	Serial-S1
2	8002	enable	Serial-S2
3	0	disable	Serial-S1
4	0	disable	Serial-S1

	Rx Multicast Address
Group-1	0.0.0.0
Group-2	0.0.0.0
Group-3	0.0.0.0

(224. 0. 0. 0 – 239. 255. 255. 255)

The following configuration realizes the application, where data received from a UDP and distributed to 2 serial ports:



	Ingress UDP Port	Forward to	Egress Serial
1	8000	enable	Serial-S1
2	8000	enable	Serial-S2
3	0	disable	Serial-S1
4	0	disable	Serial-S1

	Rx Multicast Address
Group-1	0.0.0.0
Group-2	0.0.0.0
Group-3	0.0.0.0

(224. 0. 0. 0 – 239. 255. 255. 255)

5.3.4 Receive multicast

If users need to receive the multicast UDP message, add the required multicast address from the right “Rx Multicast Address” list.

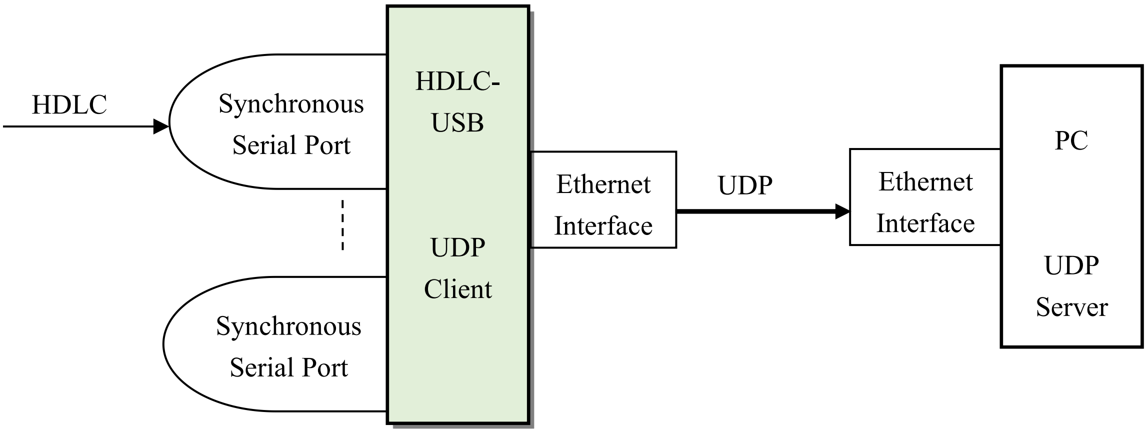
Range of the multicast address is 224.0.0.0 ~ 239.255.255.255, 224.8.8.8 is the configuration management address of the HDLC-USB and users can’t use this address.

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

5.4 Serial to UDP Conversion

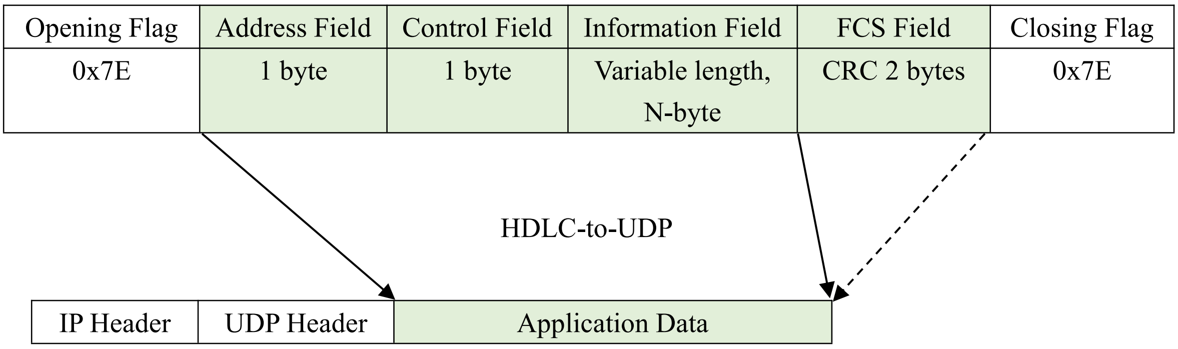
5.4.1 Application model

The HDLC-to-UDP function figure is shown below. HDLC-USB receives the HDLC frame from other devices over the synchronous serial interface, converts it into the UDP message and sends to PC or server over Ethernet.



5.4.2 Protocol Conversion

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



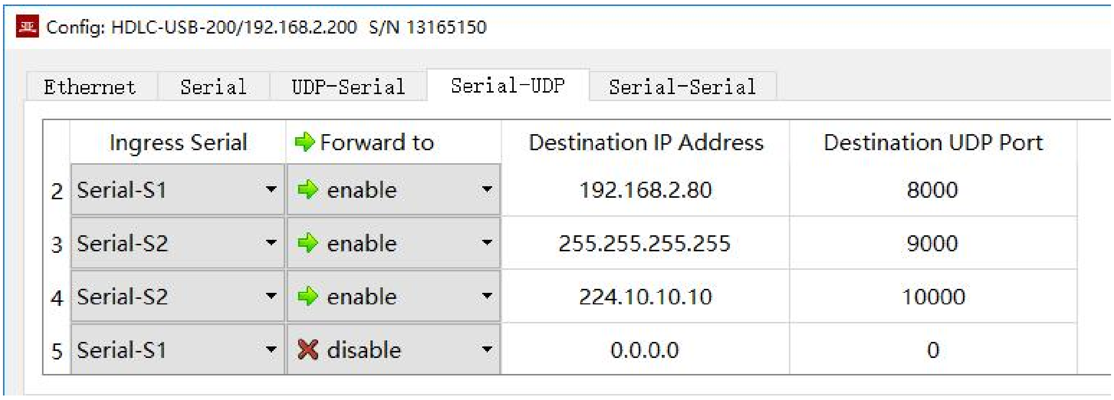
Opening Flag	Address Field	Control Field	Information Field	FCS Field	Closing Flag
0x7E	1 byte	1 byte	Variable length, N-byte	CRC 2 bytes	0x7E

IP Header	UDP Header	Application Data

5.4.3 Parameter configuration

Set the serial port to UDP. Each row represents the forwarding entry from a serial port to the UDP port with three forwarding strategies to be achieved:

- Forwarding: Data received from the specified serial port can be forwarded to the specified destination UDP port.
- Multiplexer: Data received from several different serial ports can be forwarded to the same destination UDP port.
- Demultiplexer: Data received from the same serial port can be forwarded to the different destination UDP ports.



Config: HDLC-USB-200/192.168.2.200 S/N 13165150

	Ingress Serial	Forward to	Destination IP Address	Destination UDP Port
2	Serial-S1	enable	192.168.2.80	8000
3	Serial-S2	enable	255.255.255.255	9000
4	Serial-S2	enable	224.10.10.10	10000
5	Serial-S1	disable	0.0.0.0	0

As shown above, three serial port to UDP entries are configured for achieving the following items:

- Serial port S1 to UDP unicast, with the destination IP address as 192.168.2.80 and destination UDP port as 8000
- Serial port S2 to UDP broadcast, all hosts on the subnet can receive data from S2 at the 9000 port
- Serial port S3 to UDP multicast, only the PC joining Group 224.10.10.10 on the network can receive data from S3.

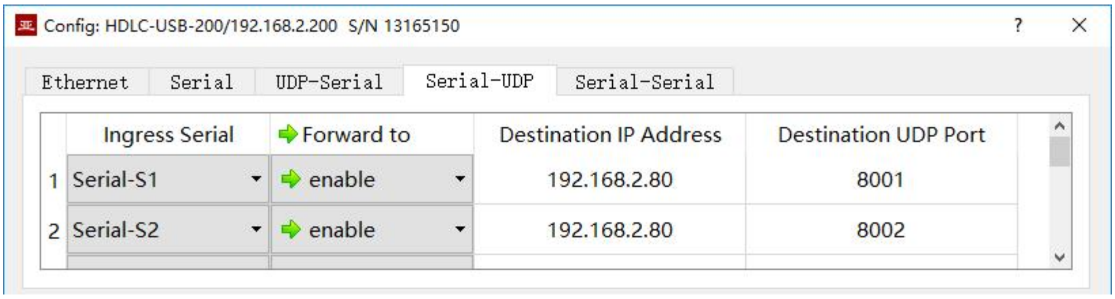
5.4.4 How does the UDP server identify the source serial

In many applications, such as the ATC, the HDLC frame from several different serial ports needs to be forwarded to a server or PC.

In this case, a strategy enables PC to know which serial port the received UDP message data is from.

5.4.4.1 Distinguish the source serial port according to the destination UDP port

As shown below, set different forwarding destination UDP ports for each serial port. The UDP Server PC receives data at the different UDP ports. Message received at port 8001 is from the serial port S1 while message received at port 8002 is from the serial port S2.



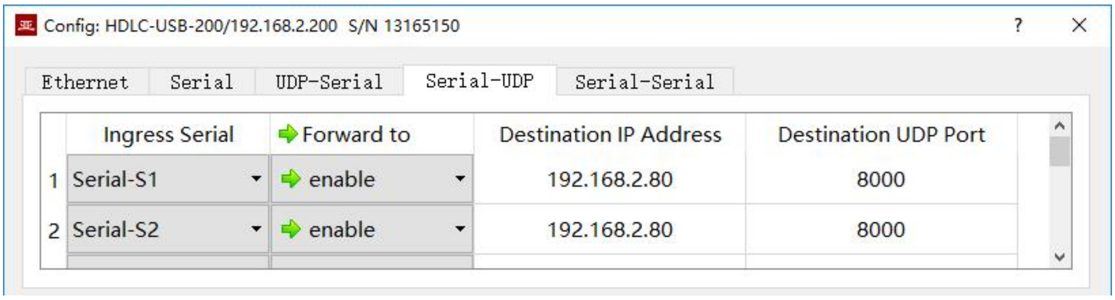
	Ingress Serial	Forward to	Destination IP Address	Destination UDP Port
1	Serial-S1	enable	192.168.2.80	8001
2	Serial-S2	enable	192.168.2.80	8002

5.4.4.2 Distinguish the source serial port according to the source UDP port

When the source serial port is identified with the destination UDP port, UDP Server needs to listen and receive data on a plurality of UDP ports. In case there are many serial ports, not only the UDP Server port occupies too many resources, the configuration and programming complexity also increases significantly.

In order to simplify implementation of the UDP Server side, we can use the configuration example below, forwarding each conversion to the same port of the UDP Server. During yacer's HDLC-USB forwarding, it will automatically adjust the source port number of the UDP message according to the source serial port. The source ports of the UDP message forwarded by the serial ports S1 and S2 are 8001 and 8002 respectively; the following is gradually increasing.

Thus, UDP Server only needs to listen and receive data at a port (8000 in the example below) and distinguishes the source serial port according to the source UDP port. If several HDLC-USBs are provided, UDP Server can distinguish the source device via the source IP.



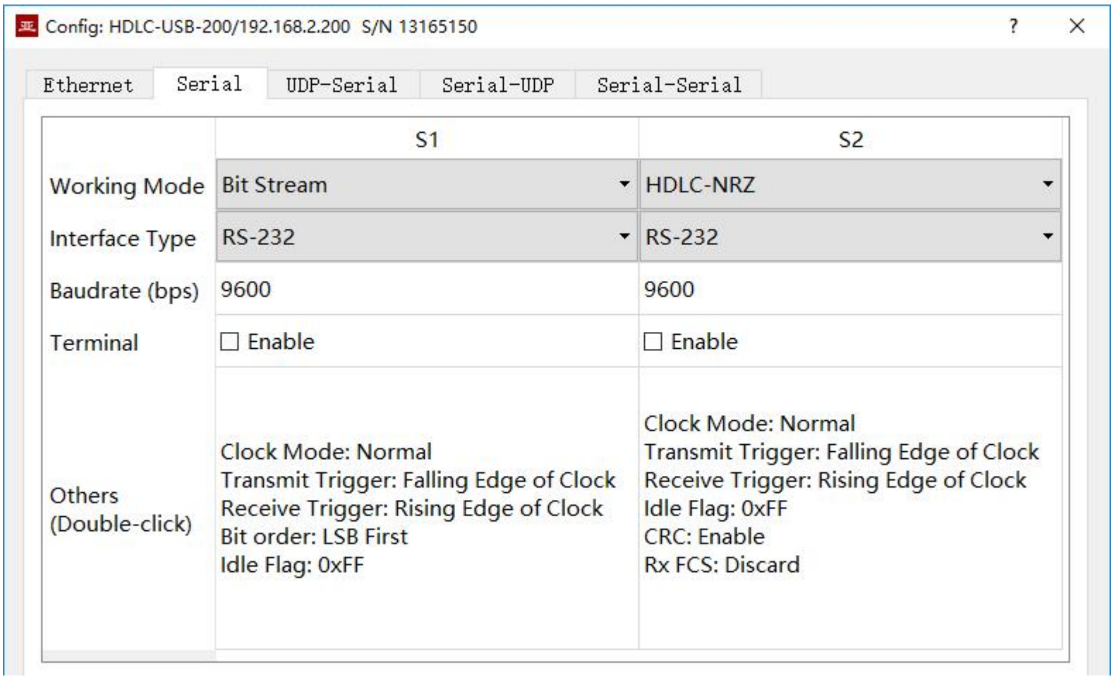
	Ingress Serial	Forward to	Destination IP Address	Destination UDP Port
1	Serial-S1	enable	192.168.2.80	8000
2	Serial-S2	enable	192.168.2.80	8000

5.5 Serial to Serial Conversion

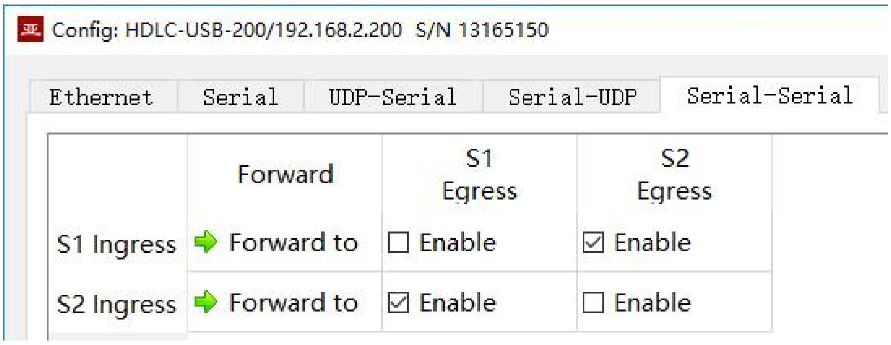
Serial-serial can forward the input data of the specified serial port to other serial port outputs, which is mainly used for:

- Conversion between synchronous and asynchronous serial ports
- Serial port Demultiplexer: Divide the single serial port data into multi-channel data. Unlike the common demultiplexer, demultiplexing is possible with HDLC-USB, the different baud rates and clock modes can be set for each-channel serial port, avoiding the packet loss caused by clock inconsistency.

For the configuration as shown below, the serial port S1 operates in the synchronous HDLC mode while S2 in the asynchronous UART mode. The serial to serial conversion is the mutual forwarding between S1 and S2 to realize the data conversion between synchronous and asynchronous serial ports.



	S1	S2
Working Mode	Bit Stream	HDLC-NRZ
Interface Type	RS-232	RS-232
Baudrate (bps)	9600	9600
Terminal	□ Enable	□ Enable
Others (Double-click)	Clock Mode: Normal Transmit Trigger: Falling Edge of Clock Receive Trigger: Rising Edge of Clock Bit order: LSB First Idle Flag: 0xFF	Clock Mode: Normal Transmit Trigger: Falling Edge of Clock Receive Trigger: Rising Edge of Clock Idle Flag: 0xFF CRC: Enable Rx FCS: Discard

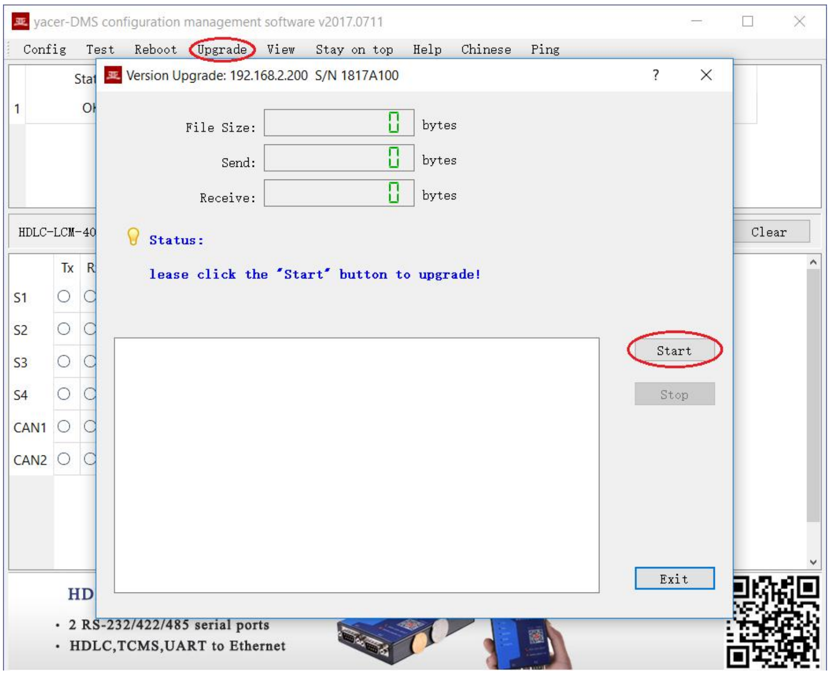


	Forward	S1 Egress	S2 Egress
S1 Ingress	➡ Forward to	□ Enable	☑ Enable
S2 Ingress	➡ Forward to	☑ Enable	□ Enable

6 Firmware Version Upgrade

6.1 Start Upgrade

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


yacer-DMS configuration management software v2017.0711
Config
Test
Reboot
Upgrade
View
Stay on top
Help
Chinese
Ping
Start
Stop
Exit
Clear
HDLC-LCM-40
Tx
R
S1
S2
S3
S4
CAN1
CAN2
HD
• 2 RS-232/422/485 serial ports
• HDLC,TCMS,UART to Ethernet

Version Upgrade: 192.168.2.200 S/N 1817A100

File Size: 0 bytes

Send: 0 bytes

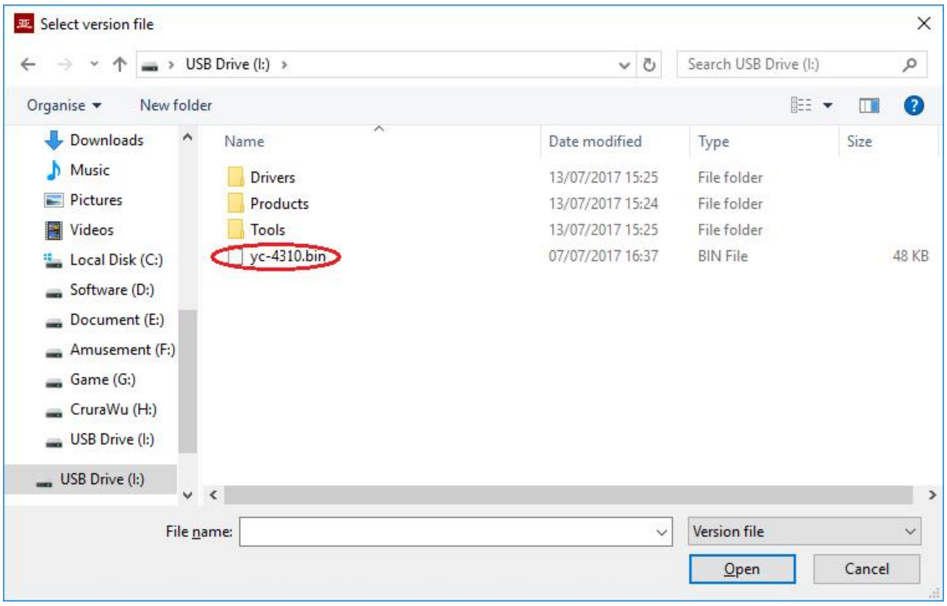
Receive: 0 bytes

Status:

lease click the 'Start' button to upgrade!

6.2 Locate Firmware Version

The “Selection Version File” dialog pops up. Locate the folder for storing the latest firmware version, select and click “Open” to start updating.



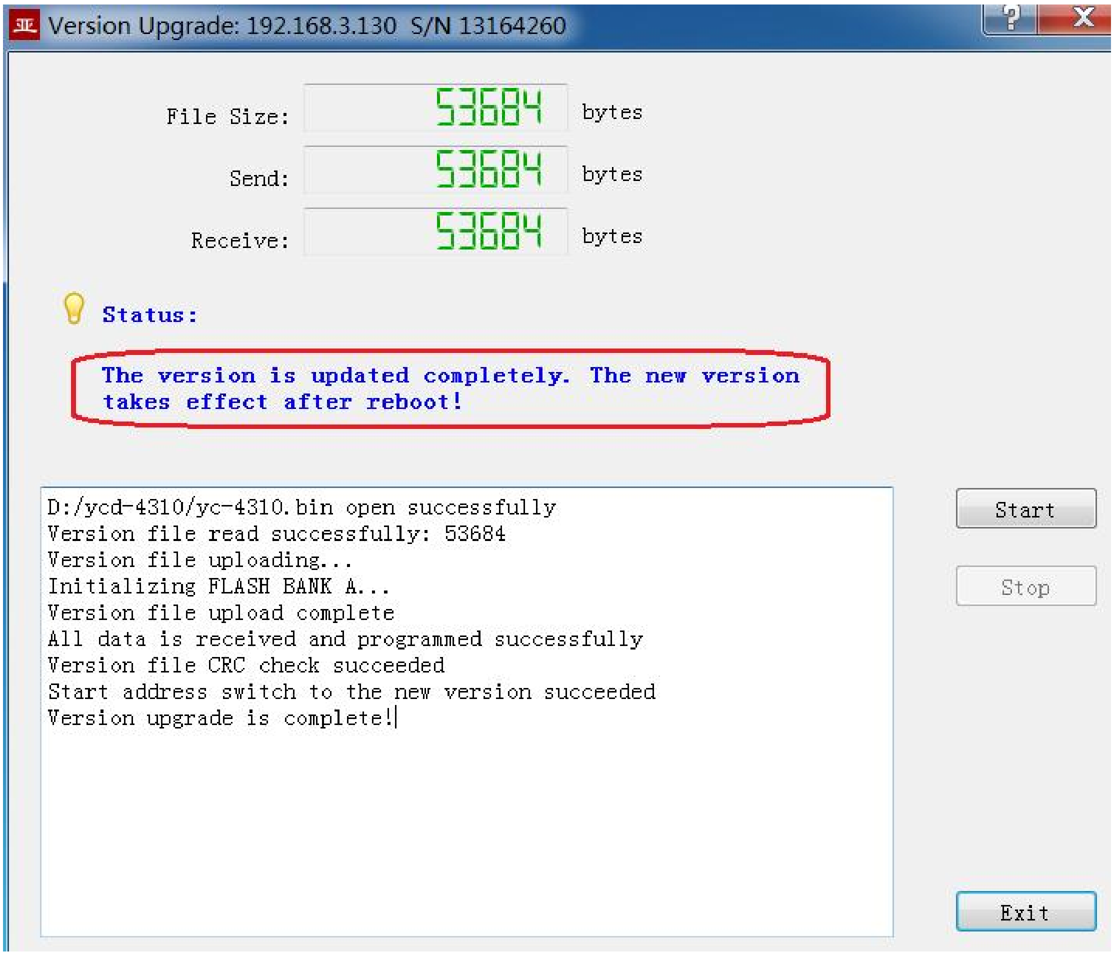
Name	Date modified	Type	Size
Drivers	13/07/2017 15:25	File folder	
Products	13/07/2017 15:24	File folder	
Tools	13/07/2017 15:25	File folder	
yc-4310.bin	07/07/2017 16:37	BIN File	48 KB

File name:

Version file

6.3 Upgrade Completed

After completion of upgrade, "Version Update Completed" displaying on the page indicates that the version update is completed.



Version Upgrade: 192.168.3.130 S/N 13164260

File Size: 53684 bytes

Send: 53684 bytes

Receive: 53684 bytes

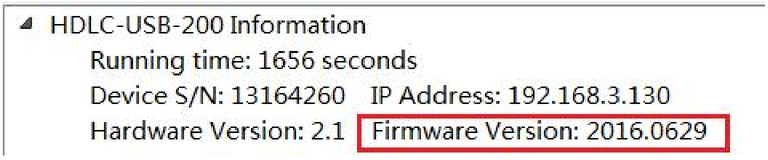
Status:

The version is updated completely. The new version takes effect after reboot!

```
D:/ycd-4310/yc-4310.bin open successfully
Version file read successfully: 53684
Version file uploading...
Initializing FLASH BANK A...
Version file upload complete
All data is received and programmed successfully
Version file CRC check succeeded
Start address switch to the new version succeeded
Version upgrade is complete!
```

6.4 Upgrade Confirmation

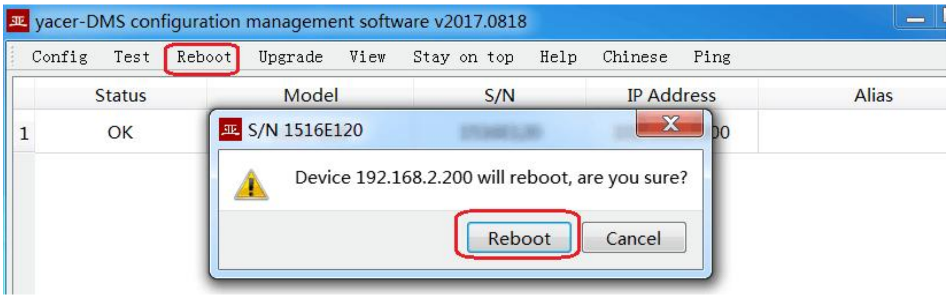
After completion of update, re power-on the device, observe the version information in the statistical report and determine whether the new version is updated successfully via the version date.



HDLC-USB-200 Information
Running time: 1656 seconds
Device S/N: 13164260 IP Address: 192.168.3.130
Hardware Version: 2.1 Firmware Version: 2016.0629

7 Device Reboot

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

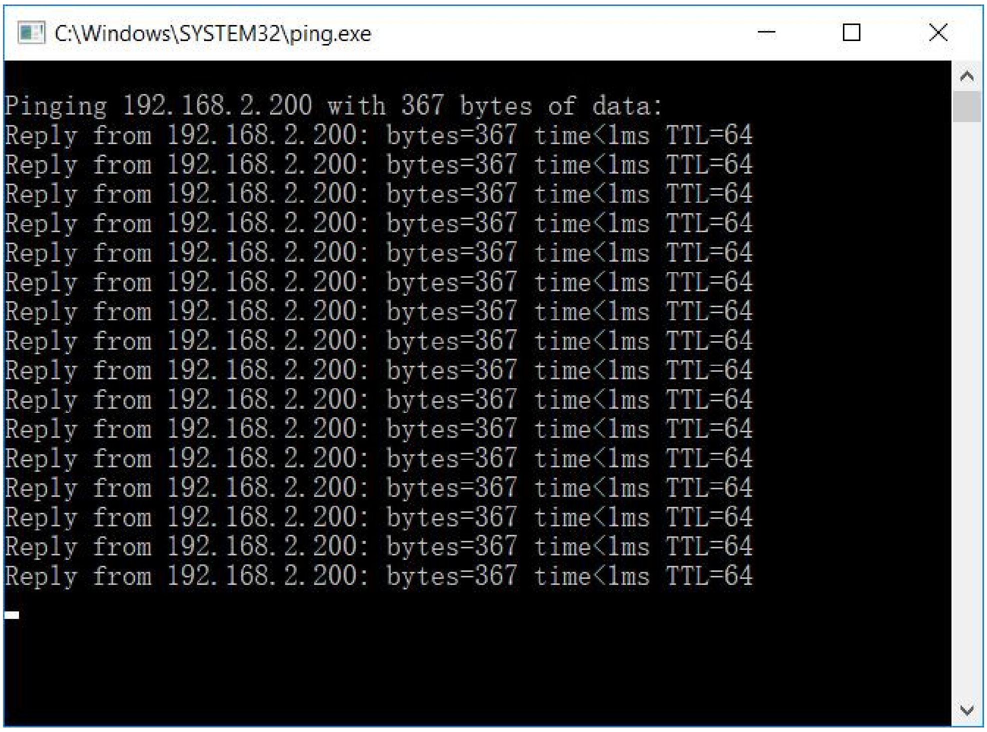


Device 192.168.2.200 will reboot, are you sure?

8 PING

By clicking on the Ping button on the toolbar, DMS will start the ping command automatically for the selected device so as to check whether the network connection between the configuration management computer and HDLC-USB is normal.

Before performing the Ping command, first ensure that the IP address of PC and HDLC-USB is on the same subnet.



```
Pinging 192.168.2.200 with 367 bytes of data:
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
Reply from 192.168.2.200: bytes=367 time<1ms TTL=64
```