HDLC-LCM

Datasheet

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

Building Blocks of Communication

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.

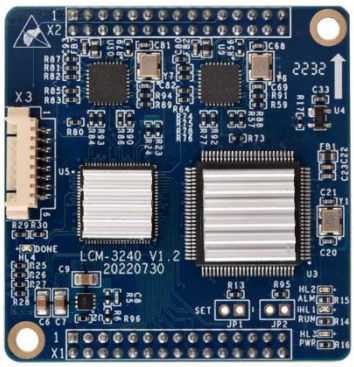
Table of Contents

1 Overview

1.1 Introduction

The Yacer HDLC-LCM embedded communication module provides four synchronous / asynchronous serial ports and two 10/100M Ethernet interfaces to realize protocol conversion between HDLC, UART and Ethernet.

46.5 x 48 mm tiny size, 2.0mm pin interface. +3.3V power supply, industrial grade wide temperature, suitable for embedded applications.



1.2 Features

- 2 10/100M Ethernet PHY interface;
- 4 serial port, support synchronous HDLC protocol, asynchronous UART working mode;
- Encoding format support NRZ, NRZI, DBPL, Manchester, differential Manchester;
- Optional 1-way CAN bus interface;
- Open and flexible configuration management;
- +3.3V power supply, low power consumption, industrial grade wide temperature;
- 46.5 x 48 mm tiny size, 2.0 mm pin connector

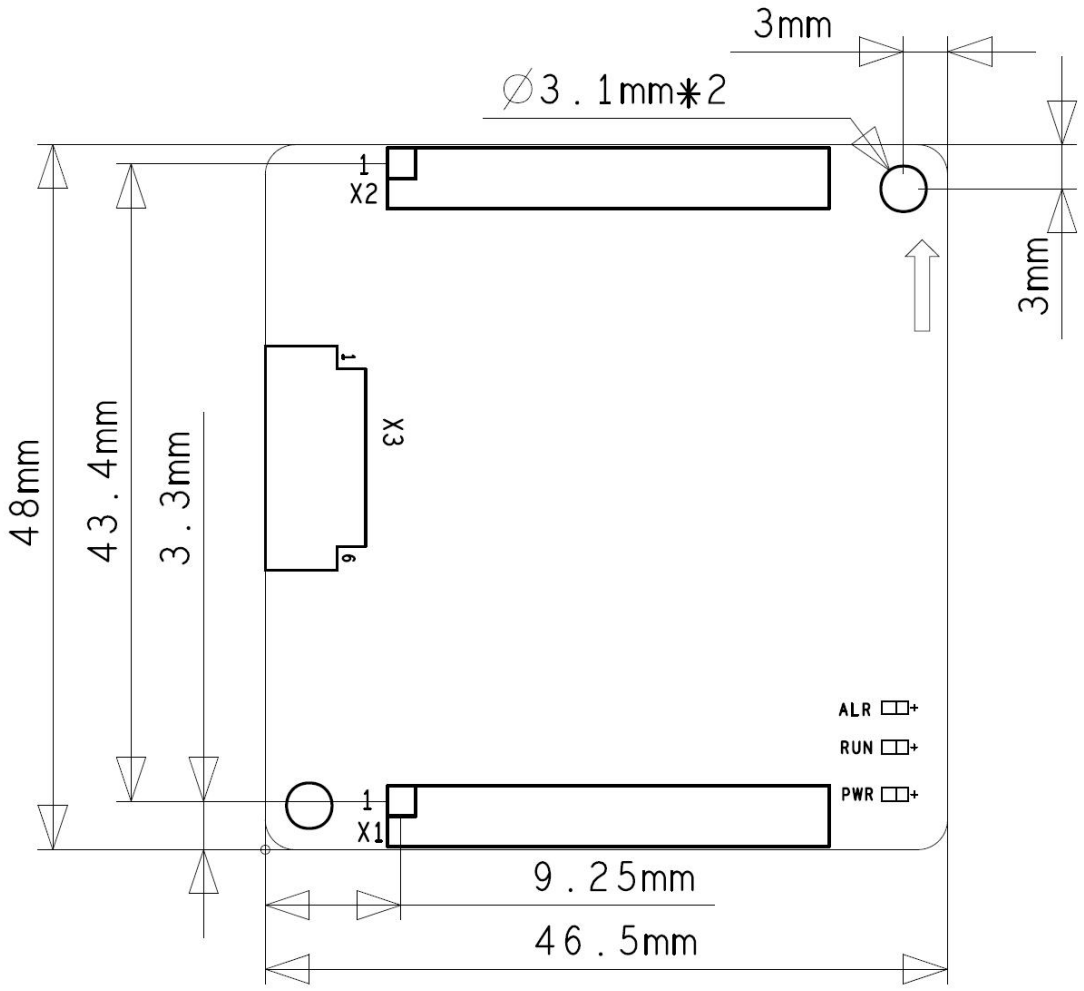
1.3 Applications

- Serial to Ethernet
- Ethernet to Serial
- Synchronous serial port, asynchronous serial port conversion
- HDLC, UDP protocol conversion
- HDLC, UART conversion
- Aviation, aerospace data communication
- Train Control and Management System (TCMS)
- Train Communication Network (TCN)
- Air Traffic Control Automation System (ATC), Air Traffic Management (ATM)
- Embedded Application and Development

1.4 Technical Specifications

Item	Parameters	Details
Synchronous / Asynchronous Serial Port	Quantity	4
	Level standard	3.3V LVCMOS
	Working mode	Synchronous HDLC, Asynchronous UART
	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
	Synchronous clock	General, Master, Slave(external clock)
Ethernet Interface	Quantity	2 x 10/100M PHY
	Rate	10/100 Mbps, supporting MDI / MDIX adaptation
	Protocol	TCP/IP
	Programming interface	UDP Server, UDP Client Support unicast/multicast/broadcast
Configuration Management	Configuration interface	Dedicated DMS-UART interface (with the help of yacer DMS-UART-8P configuration cable) Ethernet port, serial port
	Configuration tool	yacer-DMS configuration management software
Power Requirements	Input voltage	+3.3 VDC
	Power consumption	< 2 W
Mechanical Characteristics	Connector	2x 30 PIN double row pin connectors (2*15) with 2.0mm pitch
	Dimensions	46.5 x 48 mm
	Weight	25 g
Operating Environment	Operating temperature	-40 ~ +85℃
	Storage temperature	-40 ~ +85℃
	Operating humidity	5 ~ 95% RH (no condensation)

1.5 Mechanical Data



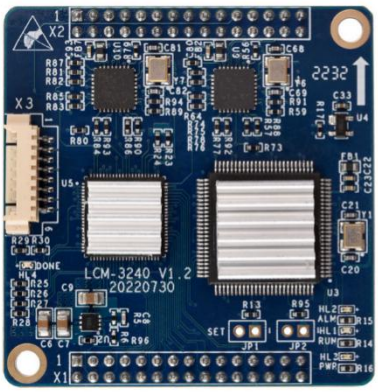
1.6 Order Information

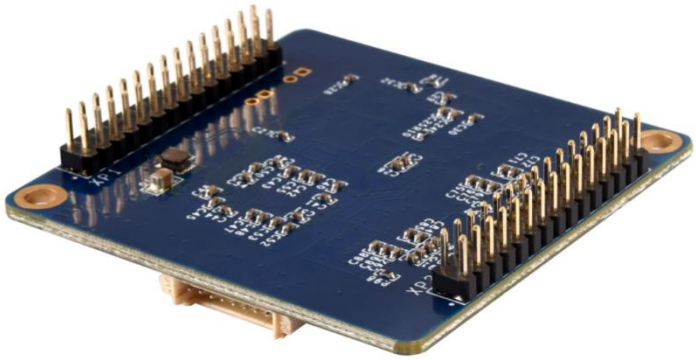
Product Model	Serial Port				CAN	Ethernet
	S1	S2	S3	S4		
HDLC-LCM-200	Sync	Sync	Async	Async	0	2 x 10/100 PHY
HDLC-LCM-400	Sync	Sync	Sync/Async	Sync/Async	0	2 x 10/100 PHY
HDLC-LCM-200C	Sync	Sync	Async	Async	1	2 x 10/100 PHY
HDLC-LCM-400C	Sync	Sync	Sync/Async	Sync/Async	1	2 x 10/100 PHY

2 Hardware and Physical Interface

2.1 Appearance

The top and bottom view of HDLC-LCM are as follows:





2.2 LED Indicators

Item	Description
RUN	Running indicator, flashing during normal operation
ALM	Alarm indicator, on when the device is not ready or fails, and off during normal operation
PWR	Power indicator, always on after power on

2.3 Pin Definition

2.3.1 X1: 2x15 2.0mm pitch connector

Pin	Signal	Type	Description
1	GND		Ground
2	GND		Ground
3	S1_TXD	O	Serial S1 data transmit
4	S1_RXD	I	Serial S1 data receive
5	S1_TXC	O	Serial S1 transmit clock
6	S1_RXC	I	Serial S1 receive clock
7	S1_TX_EN	O	Serial S1 transmit enable control, enable level is high
8	S1_LED	O	Serial S1 transmit/receive indication, drive LED

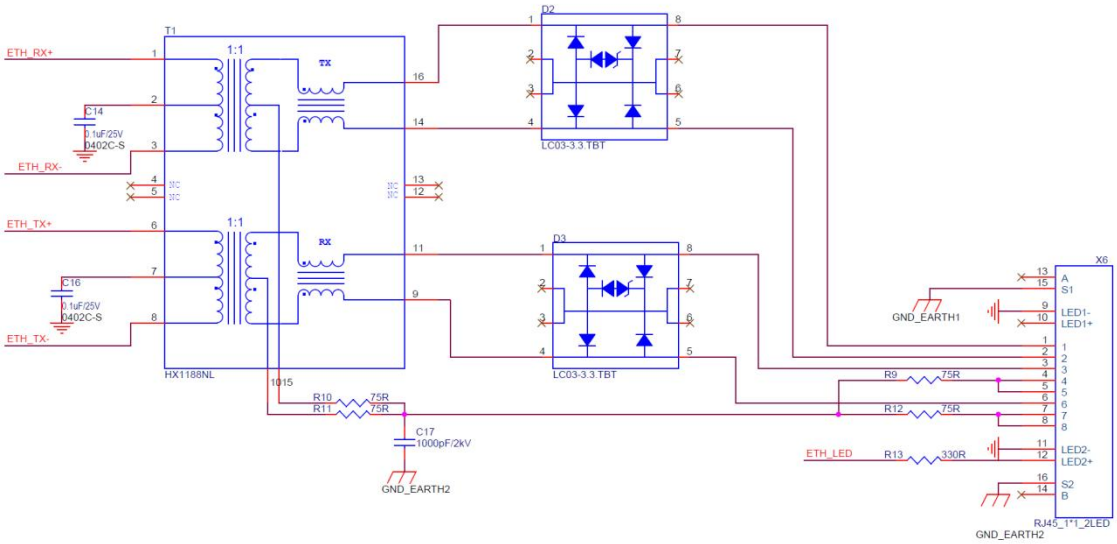
Pin	Signal	Type	Description
			negative
9	S2_TXD	O	Serial S2 data transmit
10	S2_RXD	I	Serial S2 data receive
11	S2_TXC	O	Serial S2 transmit clock
12	S2_RXC	I	Serial S2 receive clock
13	S2_TX_EN	O	Serial S2 transmit enable control, enable level is high
14	S2_LED	O	Serial S2 transmit/receive indication, drive LED negative
15	GND		Ground
16	GND		Ground
17	S3_TXD	O	Serial S3 data transmit
18	S3_RXD	I	Serial S3 data receive
19	S3_TXC	O	Serial S3 transmit clock
20	S3_RXC	I	Serial S3 receive clock
21	S3_TX_EN	O	Serial S3 transmit enable control, enable level is high
22	S3_LED	O	Serial S3 transmit/receive indication, drive LED negative
23	S4_TXD	O	Serial S4 data transmit
24	S4_RXD	I	Serial S4 data receive
25	S4_TXC	O	Serial S4 transmit clock
26	S4_RXC	I	Serial S4 receive clock
27	S4_TX_EN	O	Serial S4 transmit enable control, enable level is high
28	S4_LED	O	Serial S4 transmit/receive indication, drive LED negative
29	GND		Ground
30	GND		Ground

2.3.2 X2: 2x15 2.0mm pitch connector

Pin	Signal	Type	Description
1	GND		Ground
2	GND		Ground
3	VCC3V3	I	Power input, +3.3 VDC
4	VCC3V3	I	Power input, +3.3 VDC
5	NC		Standby, this pin must be left floating
6	NC		Standby, this pin must be left floating
7	RESET_IN	I	Module reset, active low. Power-On Reset supported, Pin can be hang.

Pin	Signal	Type	Description
8	NC		Standby, this pin must be left floating
9	NC		Standby, this pin must be left floating
10	NC		Standby, this pin must be left floating
11	LED_RUN	O	System operation indication, drive LED negative
12	LED_ALARM	O	System alarm indication, drive LED negative
13	CAN_TX	O	CAN interface data transmit
14	CAN_RX	I	CAN interface data receive
15	GND		Ground
16	GND		Ground
17	ETH1_TX+		Tx+ for Ethernet PHY interface 1, external network transformer required
18	ETH1_TX-		Tx- for Ethernet PHY interface 1, external network transformer required
19	ETH1_RX+		Rx+ for Ethernet PHY interface 1, external network transformer required
20	ETH1_RX-		Rx- for Ethernet PHY interface 1, external network transformer required
21	LED_ETH1		Link/Act indication for Ethernet 1, drive LED positive
22	NC		Standby, this pin must be left floating
23	ETH2_TX+		Tx+ for Ethernet PHY interface 2, external network transformer required
24	ETH2_TX-		Tx- for Ethernet PHY interface 2, external network transformer required
25	ETH2_RX+		Rx+ for Ethernet PHY interface 2, external network transformer required
26	ETH2_RX-		Rx- for Ethernet PHY interface 2, external network transformer required
27	LED_ETH2		Link/Act indication for Ethernet 2, drive LED positive
28	NC		Standby, this pin must be left floating
29	GND		Ground
30	GND		Ground

2.4 Ethernet Reference Circuit



2.5 LED Reference Circuit

3 Building Configuration Environment

3.1 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-LCM accompanied U-Disk;
- Official website of Yacer (http://www.yacer.com.cn) Software channel.

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.

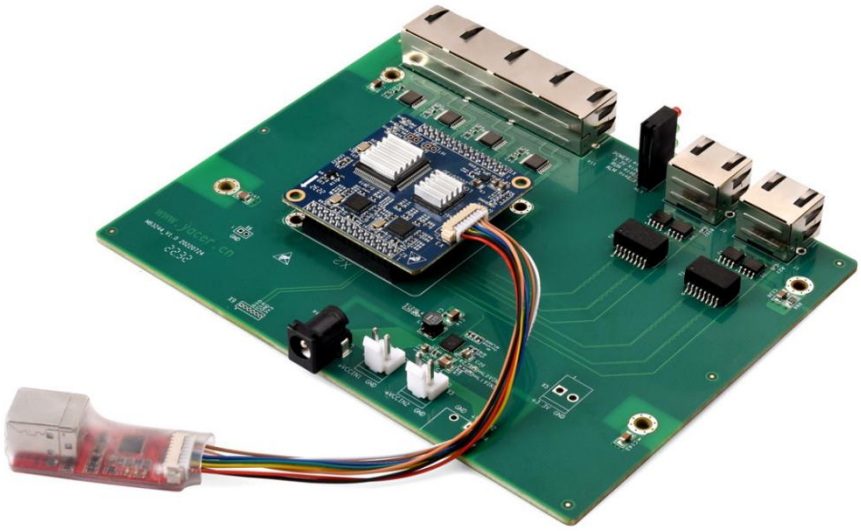
3.2 Building Configuration Environment

HDLC-LCM provides a variety of configuration management methods to meet different application scenarios.

After the HDLC-LCM is configured, the configuration parameters are saved in FLASH on the HDLC-LCM board, and will be automatically loaded to work every time HDLC-LCM is powered on or restarted in the future.

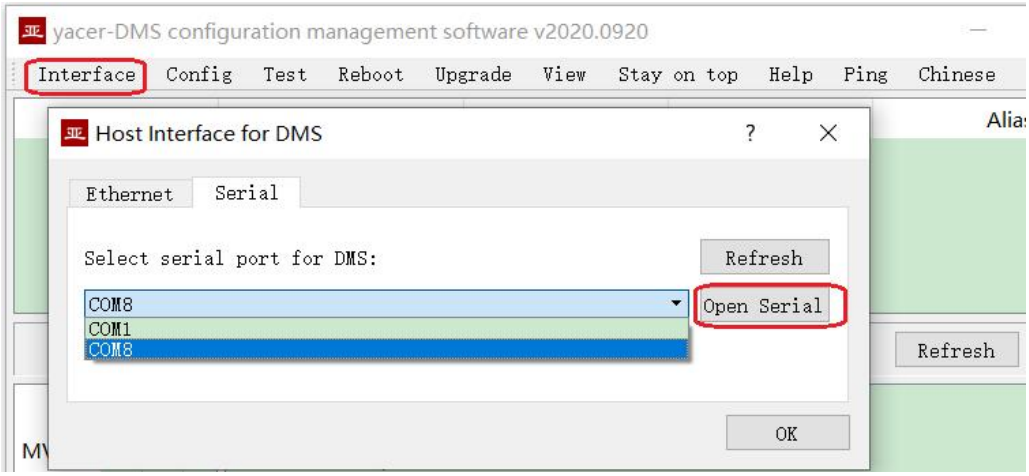
3.2.1 Configure with dedicated DMS-UART interface

Connect the dedicated DMS-UART interface (X3) of HDLC-LCM to the USB interface of the computer with the DMS-UART-8P configuration cable.

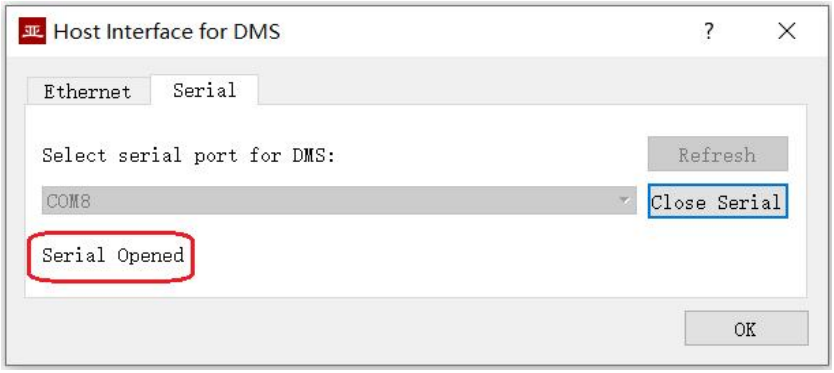


When DMS-UART-8P configuration cable is connected to the management computer USB interface, the computer will add a USB simulation serial port.

Click the “Interface” button on the toolbar to pop up the “Host Interface for DMS” configuration dialog. Enter the “Serial” page, select the serial port of the computer connected to HDLC-LCM from the drop-down list, and click “Open Serial” button.



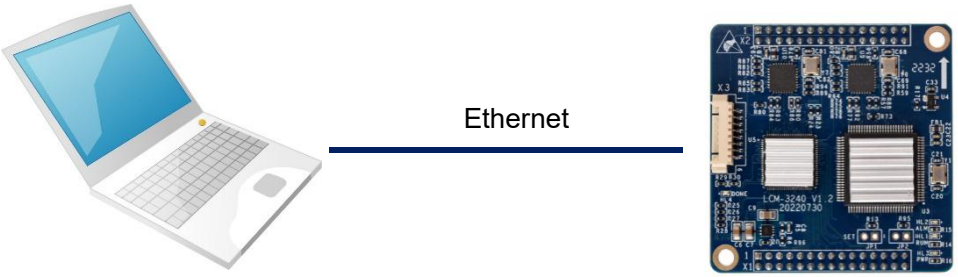
If the serial port is successfully opened, the status is as follows:



3.2.2 Configure with extended DMS-UART interface

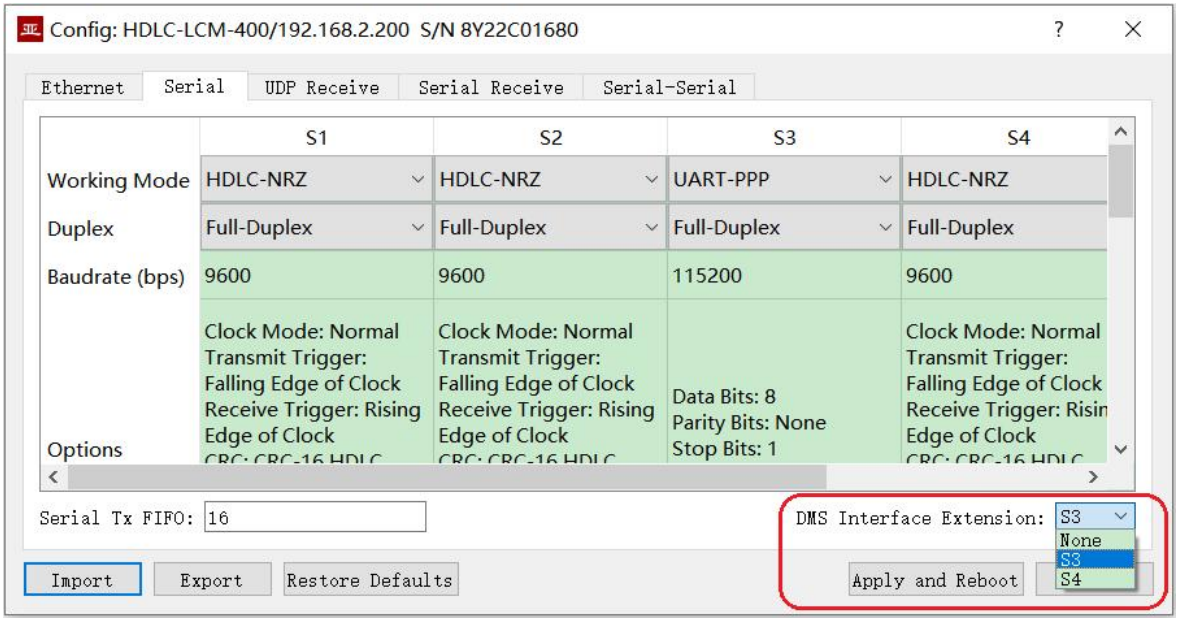
Users can connect the HDLC-LCM and the management computer via Ethernet, and run the yacer-DMS configuration management software on the computer to manage the configuration of the HDLC-LCM.

Considering network security, users can turn off Ethernet port support for DMS through yacer-DMS.



3.2.3 Configuration through serial port

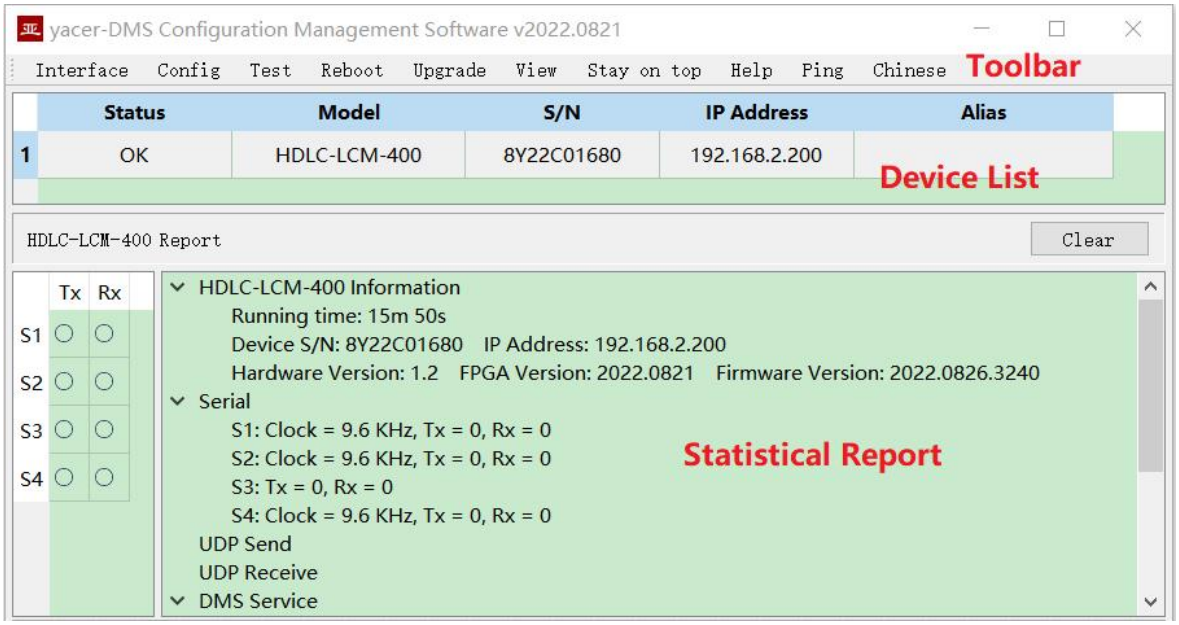
Users can choose serial port S3 or S4 to support DMS and realize serial port to configure HDLC-LCM.



3.3 Main Window of yacer-DMS

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

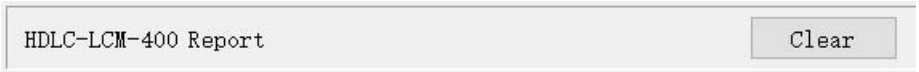
- Toolbar: Functional operation buttons;
- Device List: Displaying the basic information and operation status of online devices;
- Statistical Report: Displaying the receive/transmit indication & statistics, and device details.



3.4 Statistical Report

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

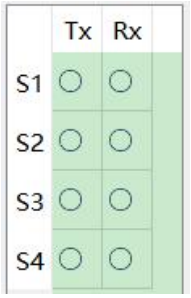
3.4.1 Control Panel



Control Widget	Descriptions
Clear	Clear the statistical report

3.4.2 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.



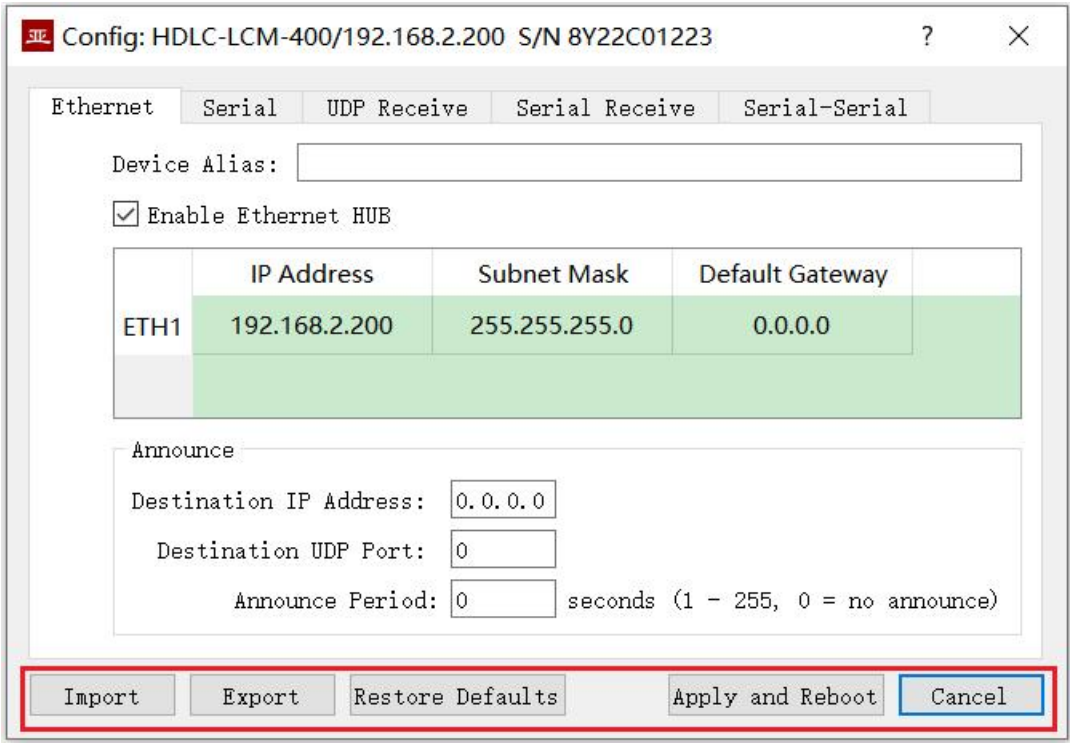
3.4.3 Information Panel

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.

3.5 Configure Device

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



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

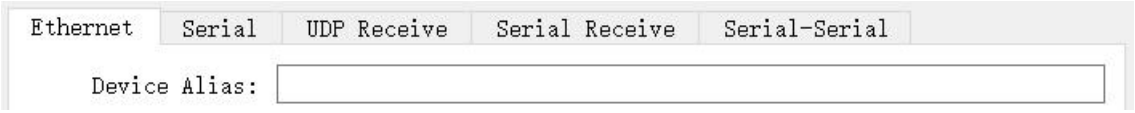
Button	Function
Import	Open the configuration file, read the configuration parameters refresh the configuration dialog
Export	Export configuration parameters from the configuration dialog to a file for saving
Restore Defaults	Refresh the configuration dialog with the factory paramters
Apply and Reboot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

4 Function and Configuration

4.1 Ethernet Interface

4.1.1 Device alias

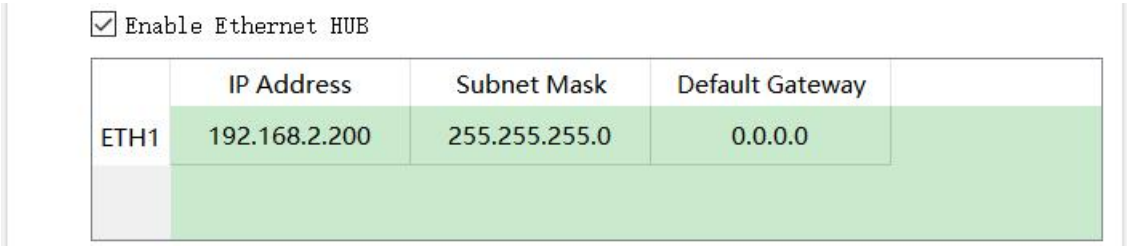
Allow users to set aliases for HDLC-LCM to add descriptions or mnemonic identities to the device.



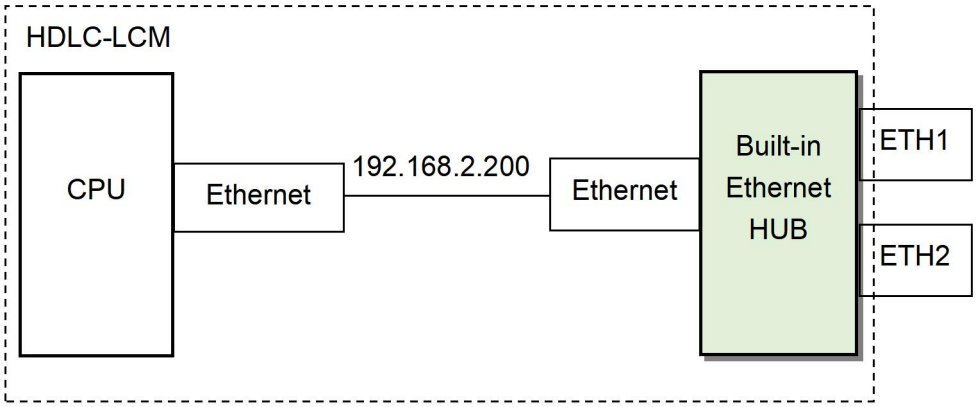
4.1.2 IP configuration

4.1.2.1 Enable Ethernet HUB

By default, check the “Enable Ethernet HUB” checkbox to enable the built-in Ethernet HUB.



After enabling the Ethernet HUB function, HDLC-LCM only has one IP address for external use. The network functions are as follows:



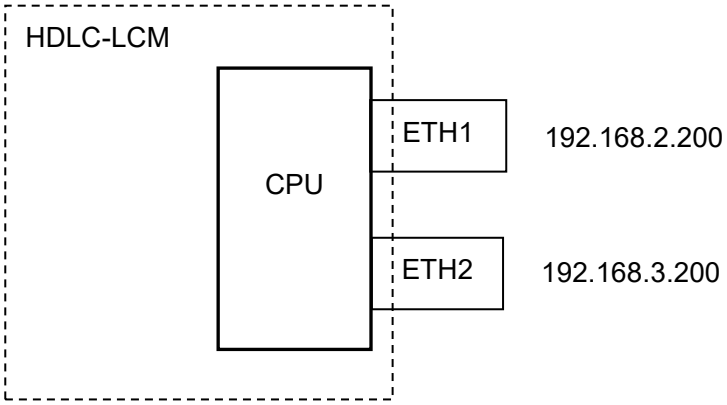
4.1.2.2 Dual IP Configuration

When the “Enable Ethernet HUB” checkbox is unchecked, ensure ETH1 and ETH2 are not on the same subnet for configuration as they have an independent IP address.

☐ Enable Ethernet HUB

	IP Address	Subnet Mask	Default Gateway
ETH1	192.168.2.200	255.255.255.0	0.0.0.0
ETH2	192.168.3.200	255.255.255.0	0.0.0.0

With the dual-IP function figure as follows, the HDLC-LCM is equivalent to a PC equipped with two network cards.

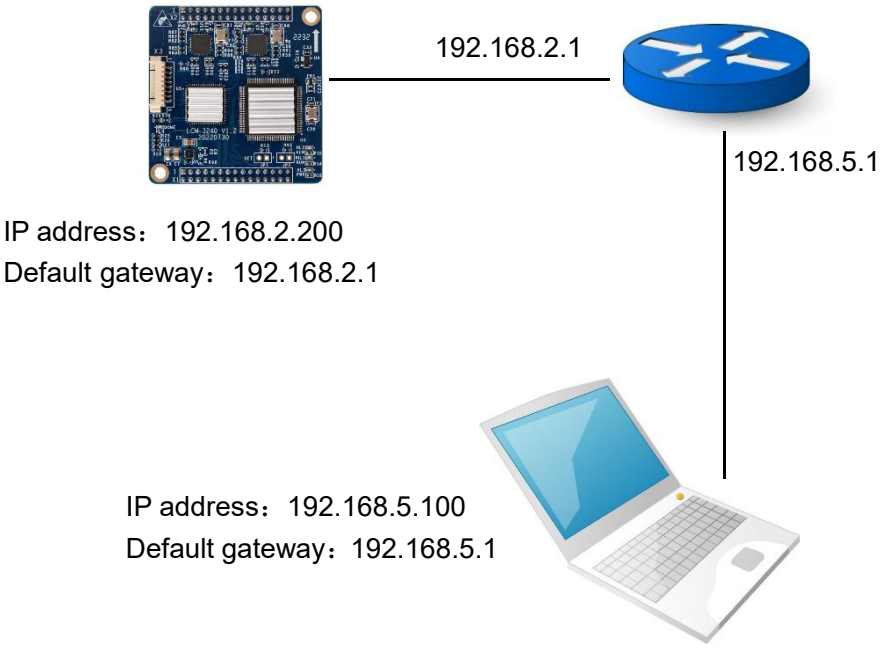


4.1.3 Default Gateway

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

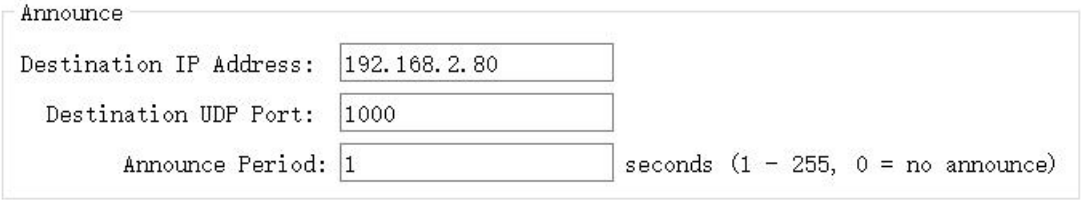
If HDLC-LCM needs to communicate with the host on other subnet, it must rely on an external router. At this time, the HDLC-LCM’s IP address must be on the same subnet with the IP address of the connected router port. Meanwhile, the IP address of router is set to the default gateway.

As shown below, the IP address of HDLC-LCM 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-LCM and PC need to set the IP address of the connected router port to the default gateway of this device.



4.1.4 Notification Configuration

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



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.
Notification Period	Sending period of device operation information notification, 1 ~ 255 seconds

4.2 Serial Port

4.2.1 Working mode of the serial port

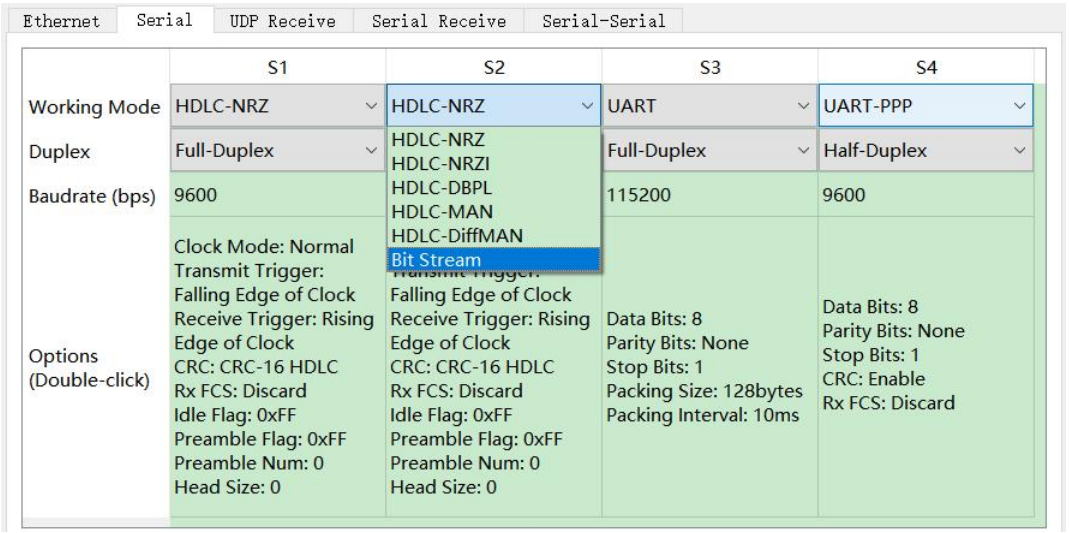
Serial ports S1 and S2 are synchronous and asynchronous serial ports, which support synchronous and asynchronous working modes.

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.



4.2.2 Duplex mode

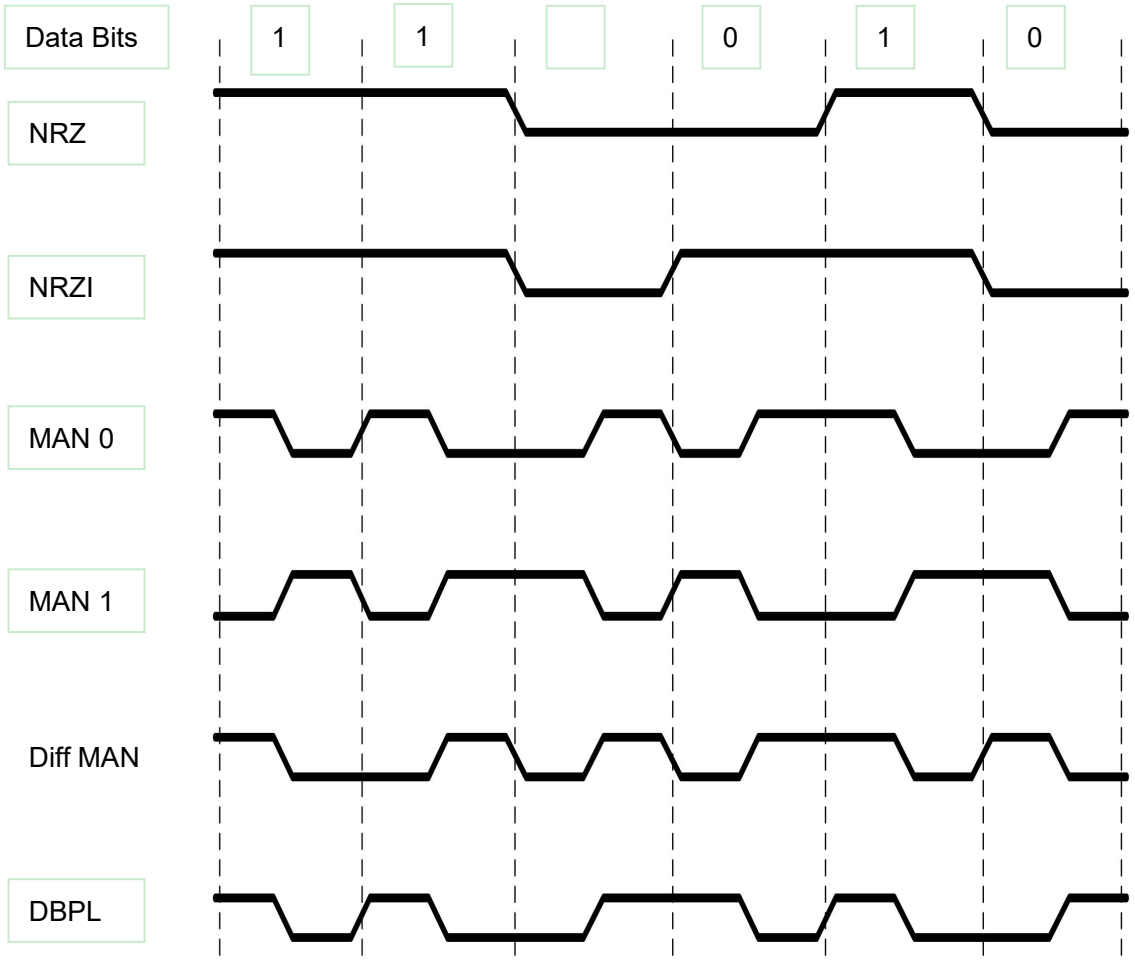
Full duplex or half duplex mode can be selected as required.

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.

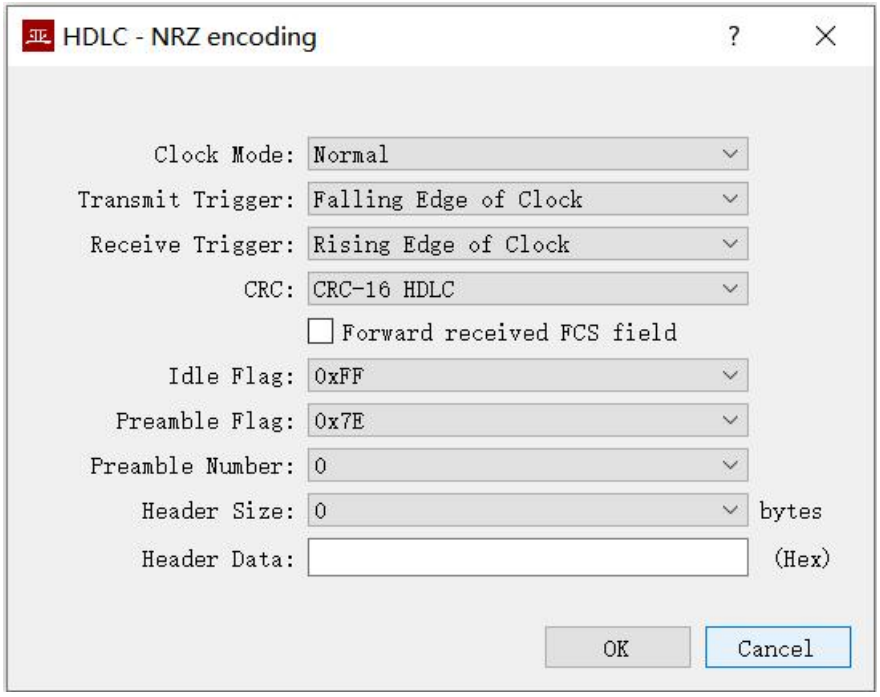
4.2.4 Encoding format of the synchronous serial port

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:

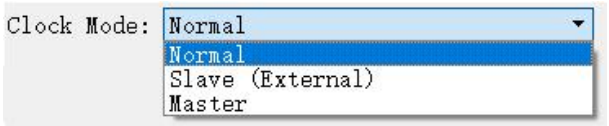


4.2.5 HDLC-NRZ Parameter Configuration

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.



4.2.5.1 Clock Mode

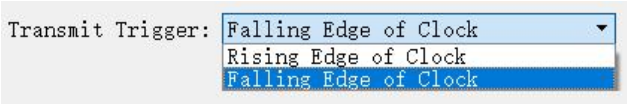


There are three clock modes for the synchronous serial port, normal, slave clock and master clock.

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-LCM 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

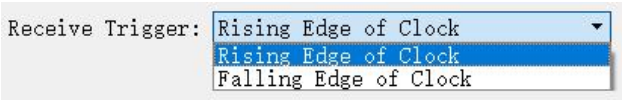


Transmit trigger defines the generation time of new data bit:

- 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.

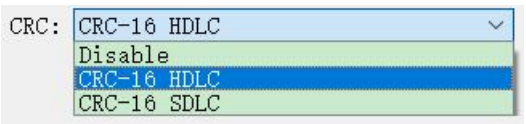
The receive trigger configuration should be determined according to the transmit trigger of the opposite device:

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.

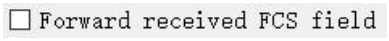
By default, the CRC-16-HDLC check mode should be selected for HDLC communication.



CRC Type	Description
Disable	CRC disabled: ● Send: No CRC calculation, no FCS field for HDLC frame

CRC Type	Description
	• 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



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-LCM 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.

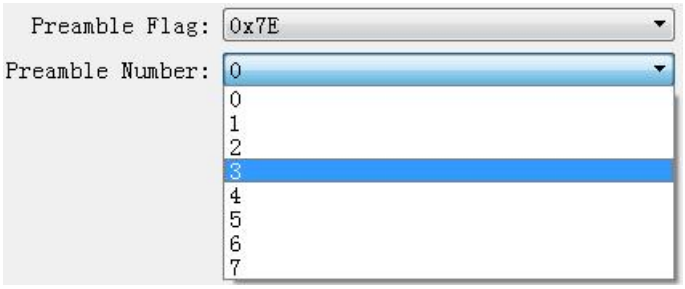
4.2.5.6 Idle Flag

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

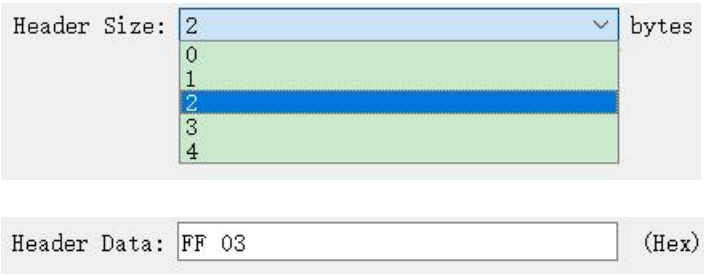
4.2.5.7 Preamble Flag and Number

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



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:

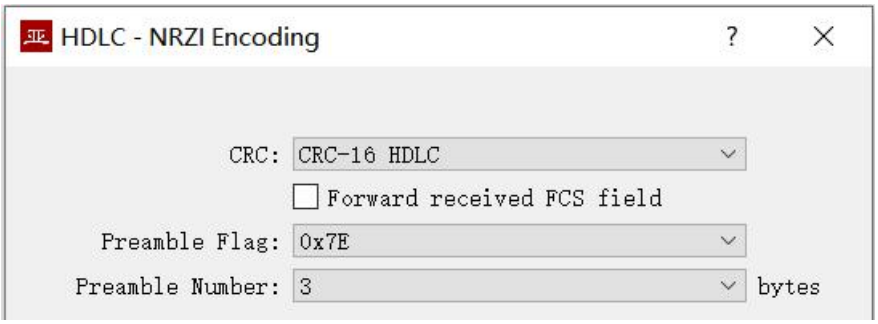
- HDLC send: add FF 03 before user data to make up HDLC frame data with user data;
- HDLC Receive: HDLC-LCM discards the first two bytes of HDLC frame data as a header and forwards only subsequent data to the user.

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

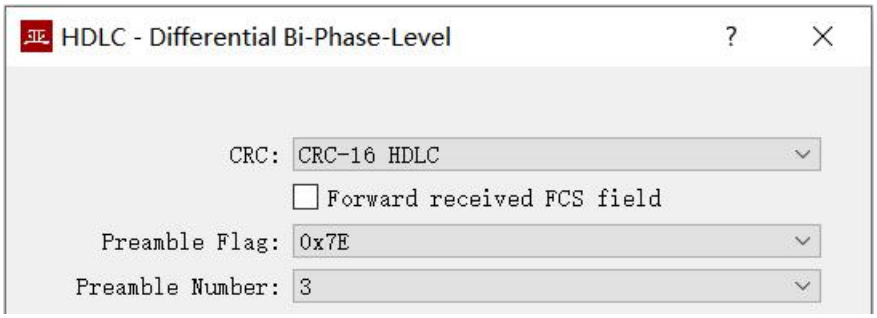
Unlike NRZ encoding format, NRZI encoding format contains clock information in the data, which only requires the same baud rate for both sides of communication, and no longer requires parameters such as clock mode, transmit trigger, and receive trigger.

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

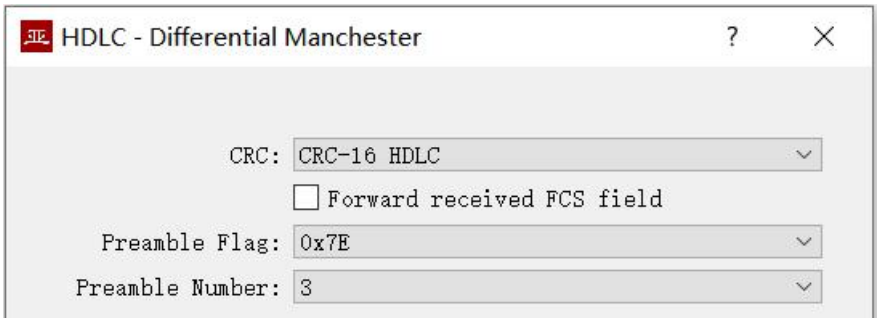
HDLC-DBPL adopts the Differential Bi-Phase Level coding format with the same parameter meaning as HDLC-NRZI.



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.

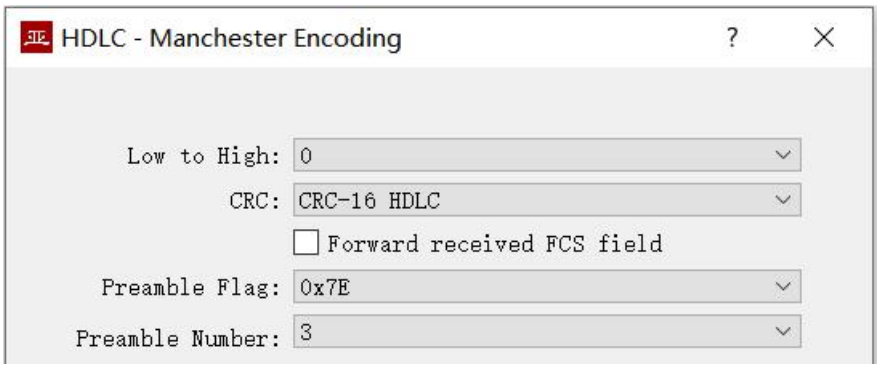
4.2.8 HDLC-DiffMAN (differential Manchester) Configuration

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



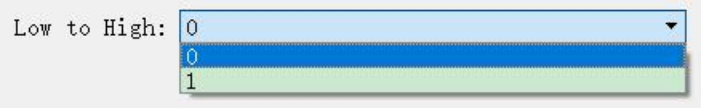
4.2.9 HDLC-MAN (Manchester) Configuration

The Manchester Options dialog box is shown below:



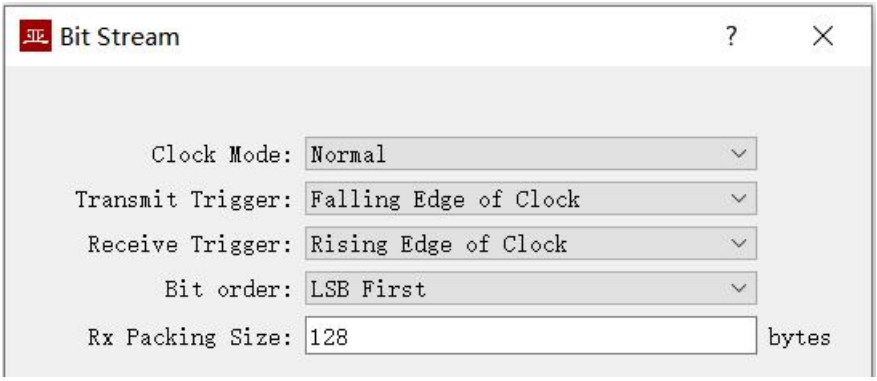
In addition to the same configuration parameters as NRZI, the Manchester encoding format has parameters with the meaning of edges with low to high waveforms for data lines:

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



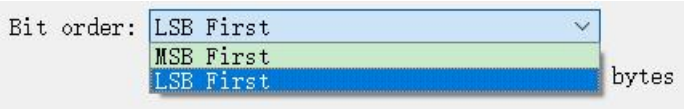
4.2.10 Bit Stream Parameter Configuration

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.

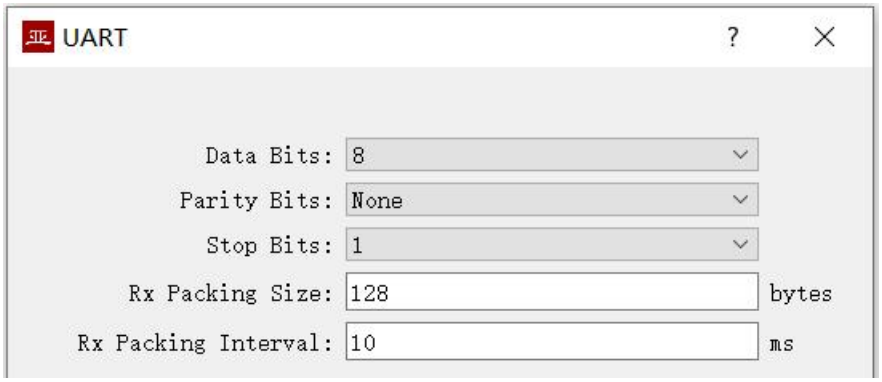


Receive/Transmit Sequence	Transmit Operation	Receive Operation
MSB First	Ttransmit the high-bit byte first	Data received first is placed in the high-bit byte
LSB First	Transmit the low-bit byte first	Data received first is placed in the low-bit byte

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.



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-LCM 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

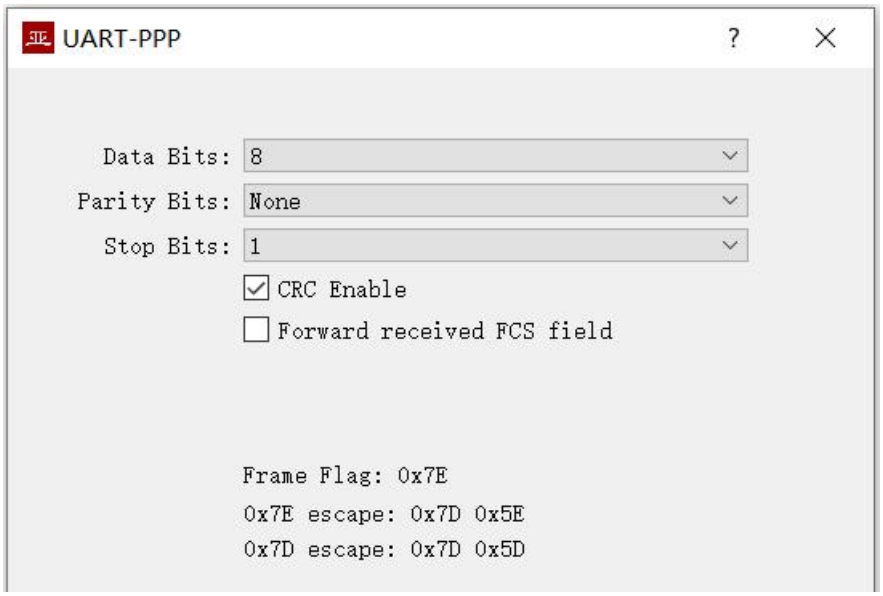
For example, if the Packing Size is set to 128 bytes, then when UART receives 128 bytes, a packet will be formed for forwarding.

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.

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:

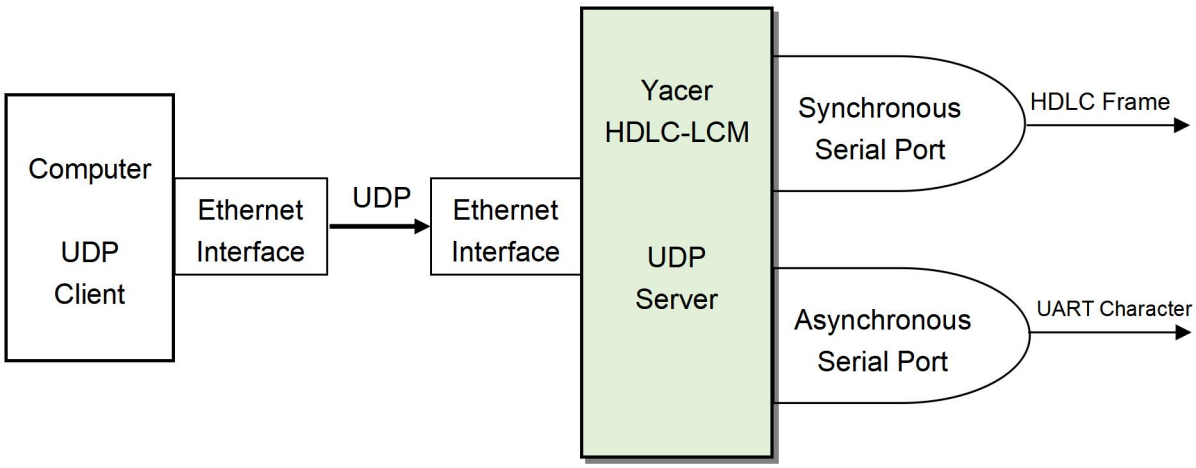
Original Data	Actual Transmit Data
0x7D 0x5E	0x7E
0x7D 0x5D	0x7D
Others	No change

4.3 UDP to Serial

4.3.1 Function Description

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

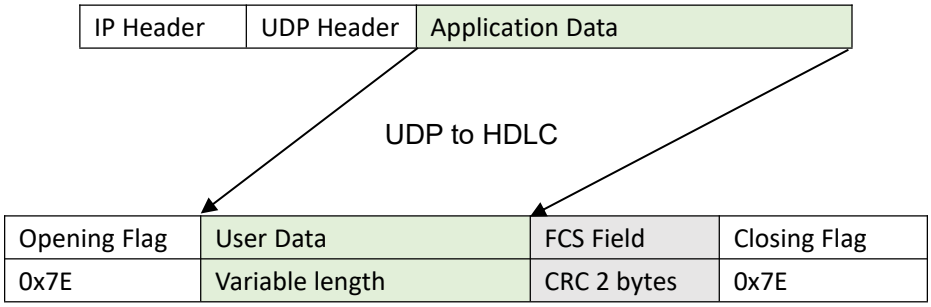
A typical application is shown in the figure below. Computer as UDP Client sends UDP messages through Ethernet interface, HDLC-LCM processes the received UDP messages, converts them into HDLC frames or UART data according to the configuration, and then sends them out from the serial port



4.3.2 Protocol Conversion

The most typical application of UDP to HDLC is shown in the figure below. HDLC-LCM 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.

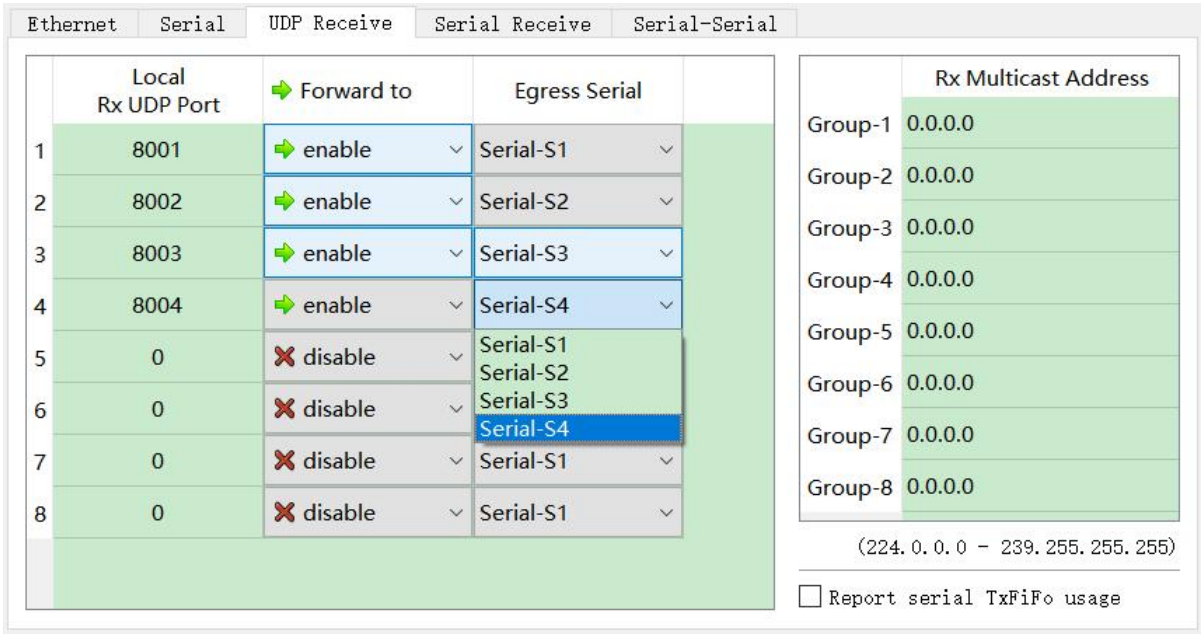
In order to reduce the computational load of the computer and the complexity of user programming, generally, the UDP message does not contain the FCS field of HDLC, which is calculated and filled by HDLC-LCM.



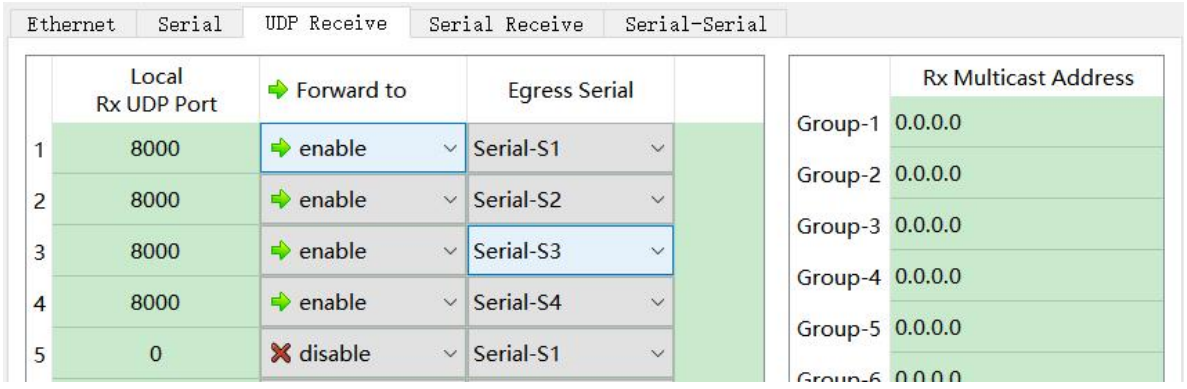
4.3.3 Forwarding Configuration

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.



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



4.3.4 Receive Multicast

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

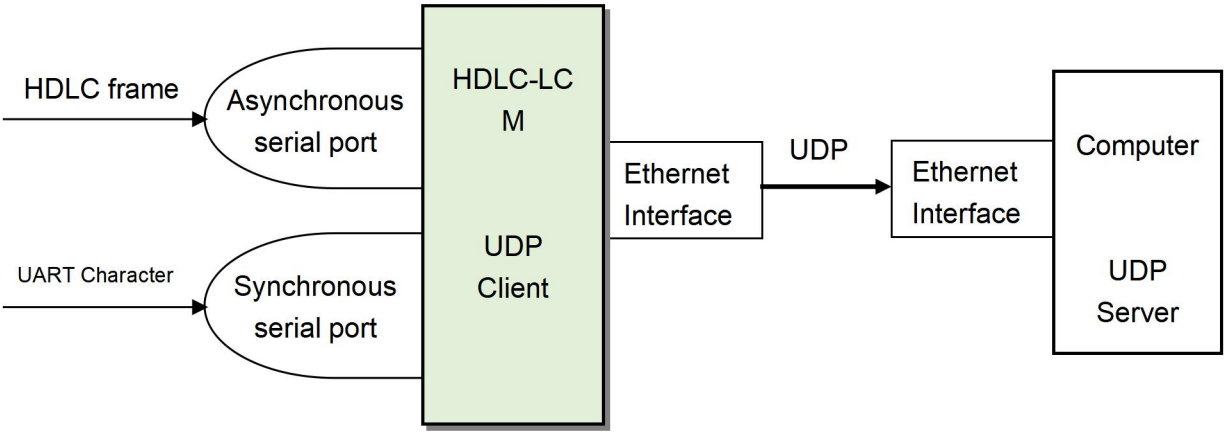
Range of the multicast address is 224.0.0.0 ~ 239.255.255.255, 224.8.8.8 is the configuration management address for HDLC-LCM and cannot be used by users.

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

4.4 Serial to UDP

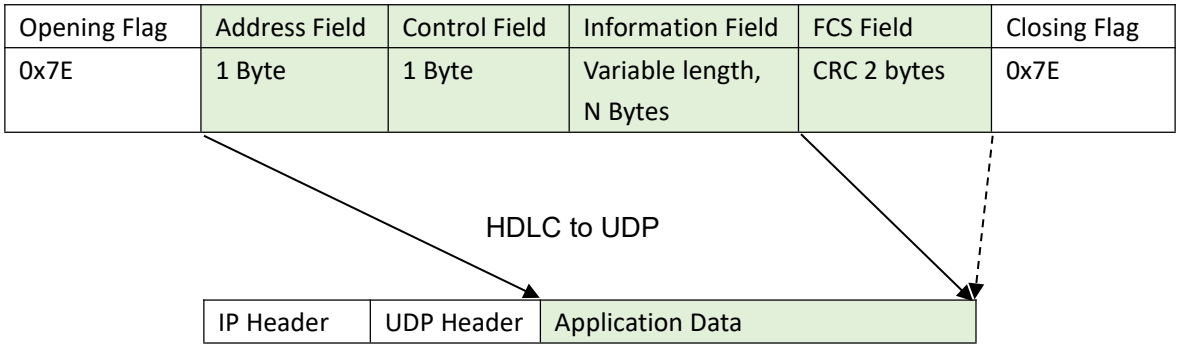
4.4.1 Function Description

The schematic diagram of serial port to UDP function is as follows. HDLC-LCM 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

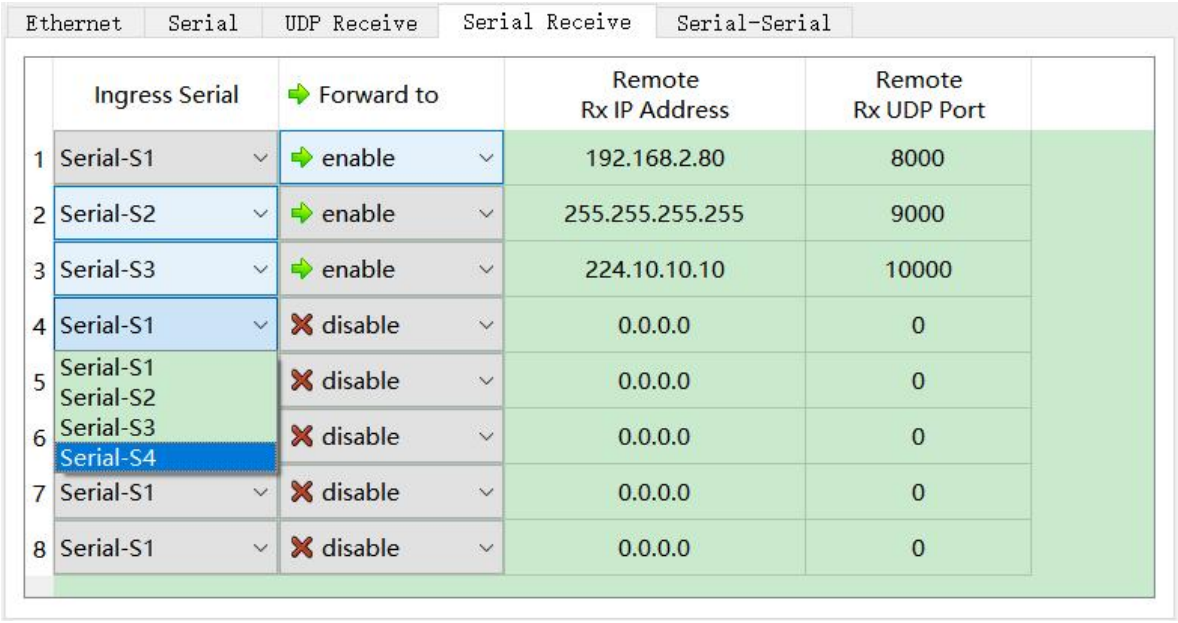
To ensure the integrity of user data, HDLC-LCM places complete HDLC frames in UDP application data and forwards them to UDP Server.



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.



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

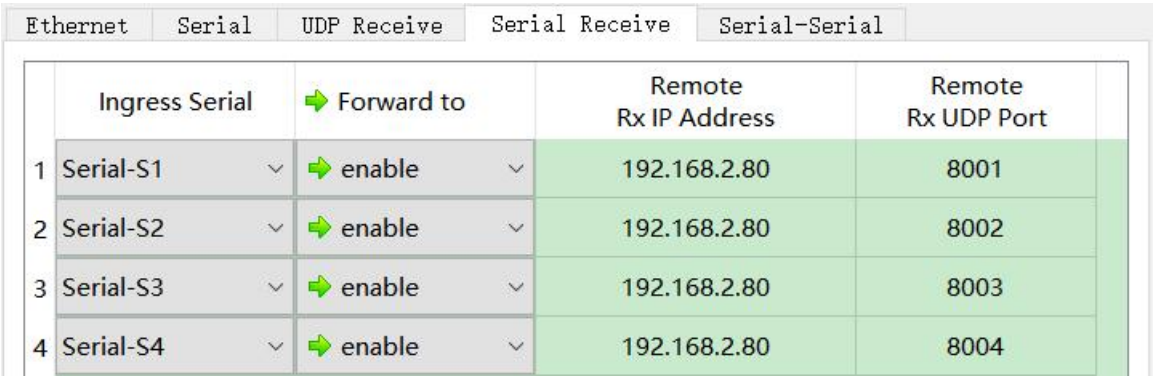
- Serial port S1 to UDP unicast, destination IP address 192.168.2.80 and destination UDP port 8000;
- Serial port S2 to UDP broadcast, all hosts in the network can receive data from S2 at port 9000;
- Serial port S3 to UDP multicast, only computers joined 224.10.10.10 groups in the network can receive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

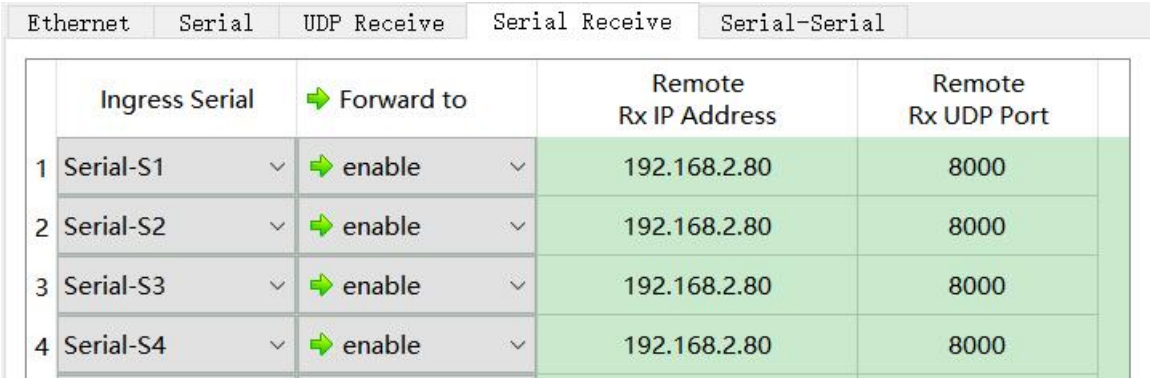


4.4.4.2 Identify Source Serial Port Based on Source UDP Port

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-LCM 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-LCM, UDP Server can distinguish the source devices through the source IP.

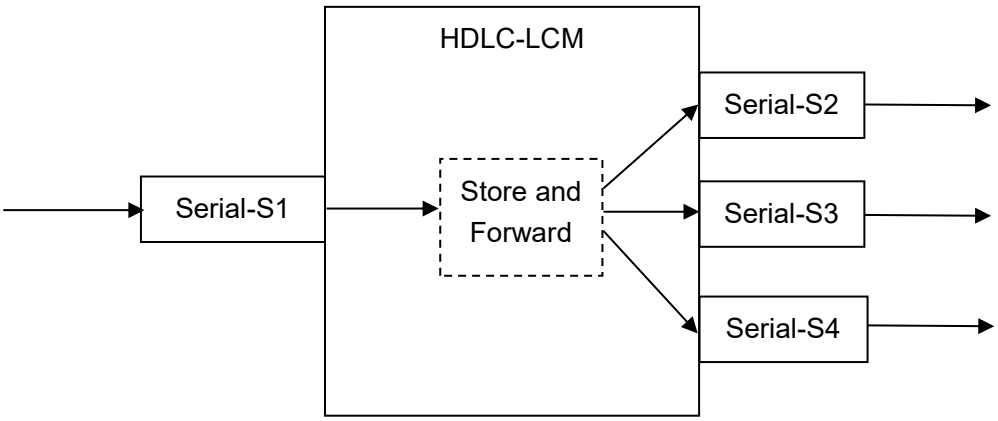


4.5 Serial to Serial

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-LCM to implement demultiplexer, each serial port can set different baud rate and clock mode, avoiding packet loss caused by clock inconsistency.

As shown in the above figure, the input of serial port S1 is split to the output of S2, S3 and S4. HDLC-LCM 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.

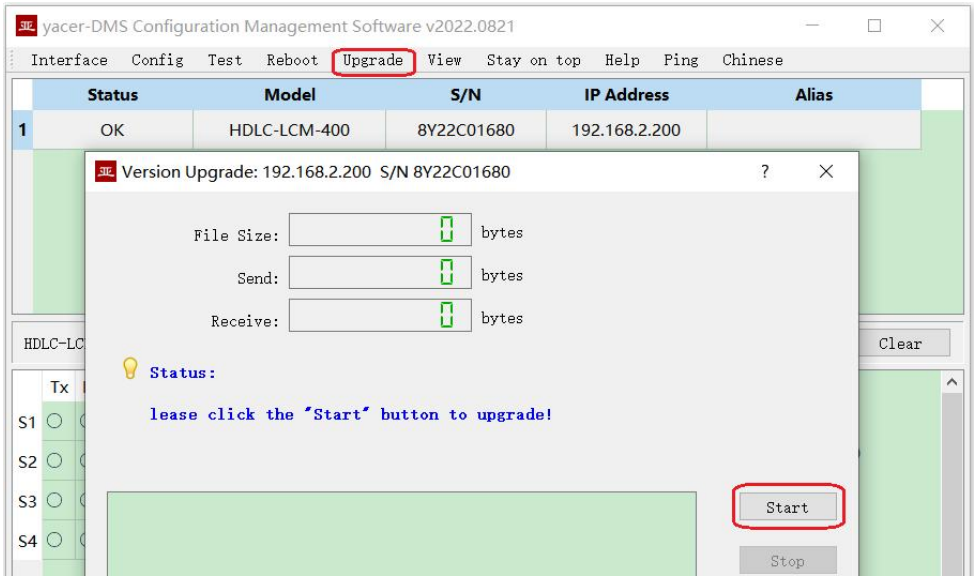


5 System Maintenance

5.1 Firmware Version Upgrade

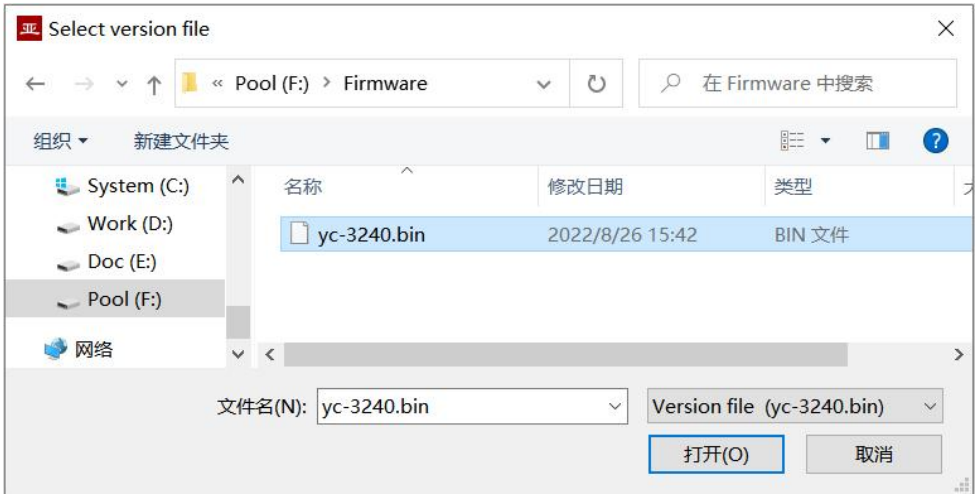
5.1.1 Start Upgrade

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



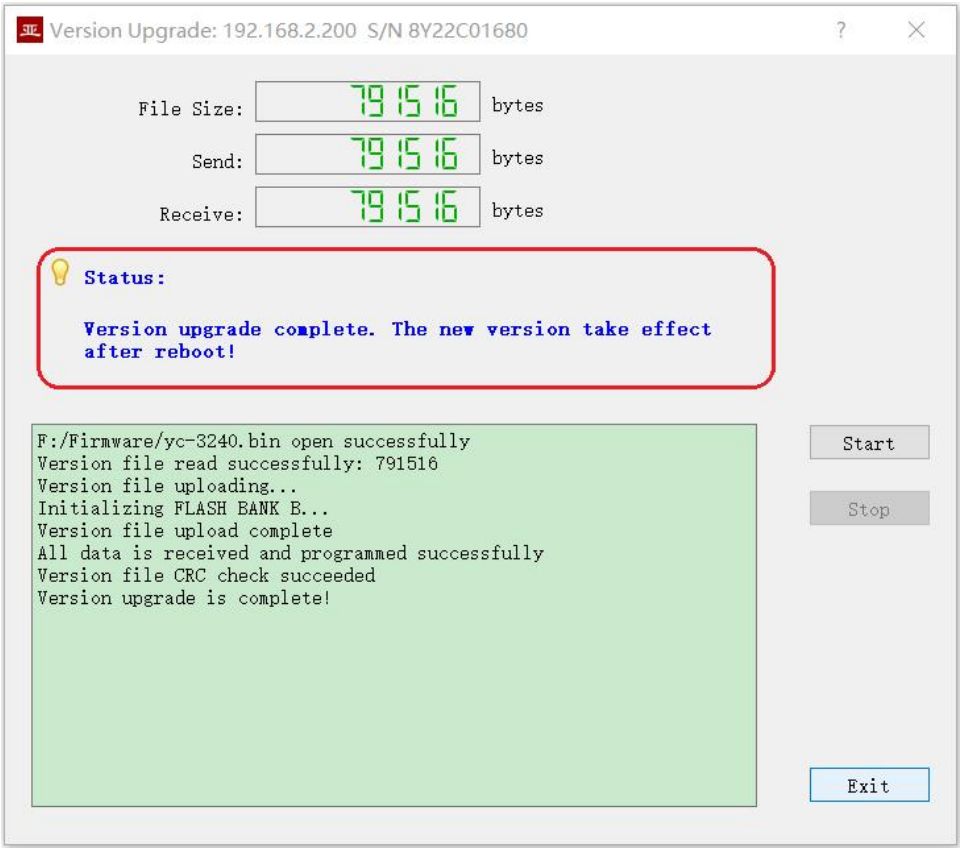
5.1.2 Select Version File

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.



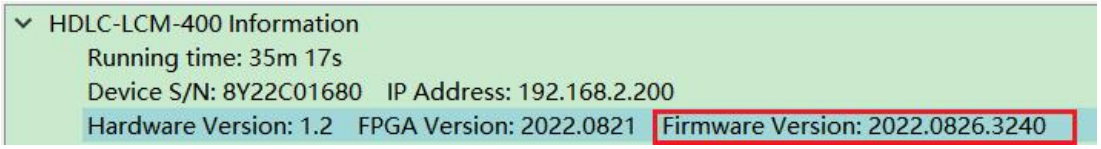
5.1.4 Re-powering takes effect

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.

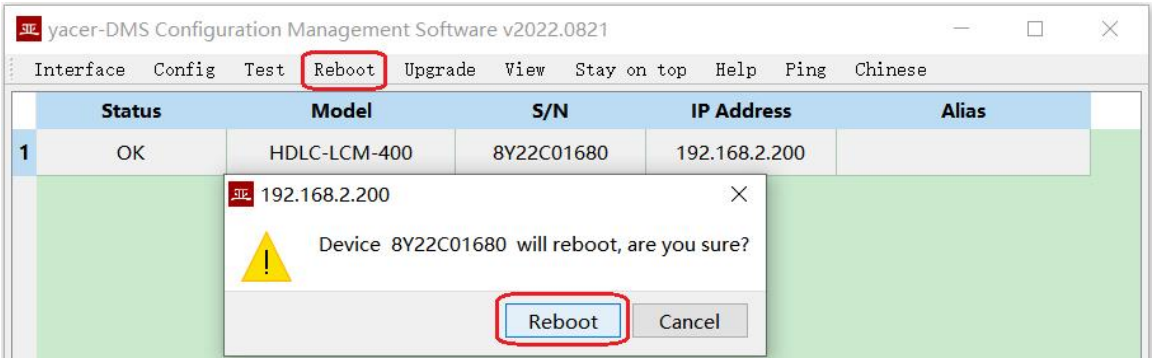
5.1.5 Confirm Upgrade

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.



6 Forwarding Function and Data Format

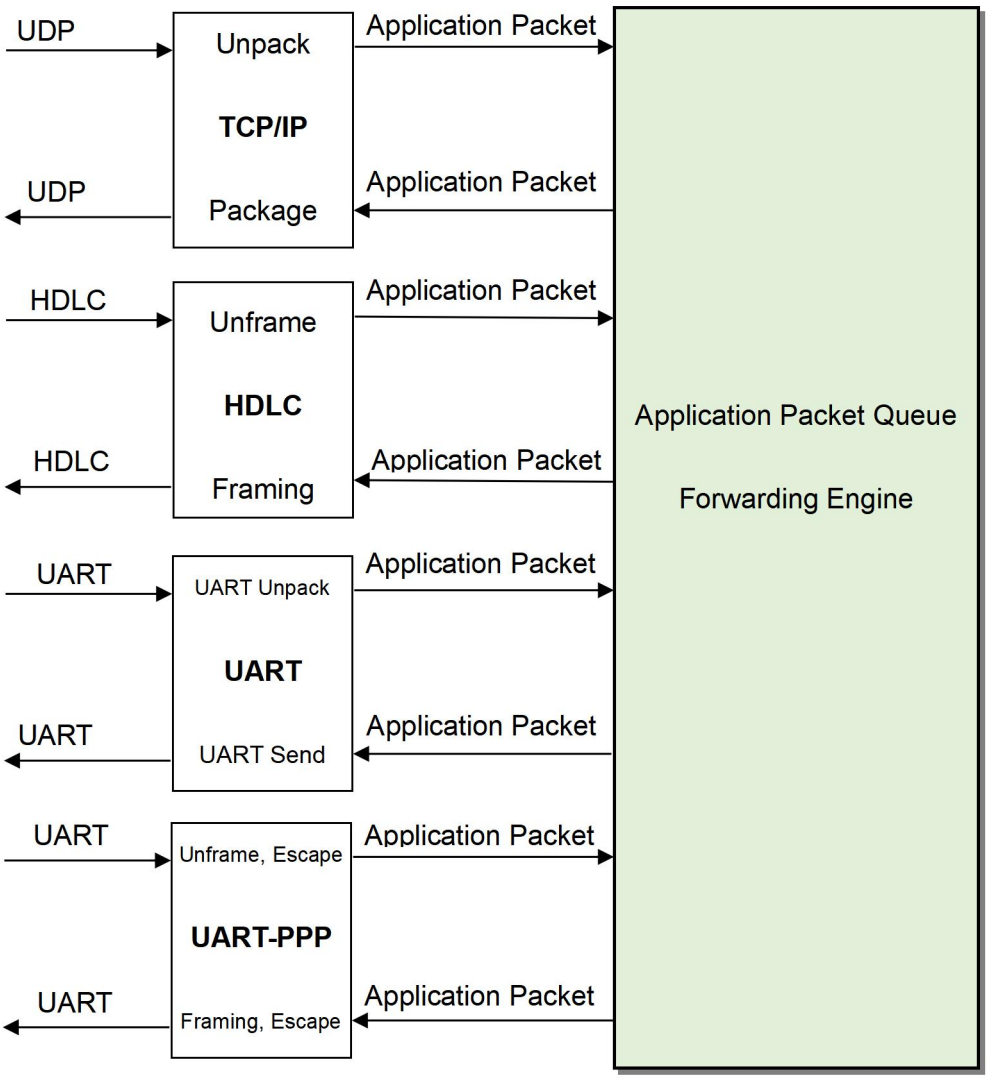
6.1 Application Packet and Conversion Model

Serial data conversion includes:

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

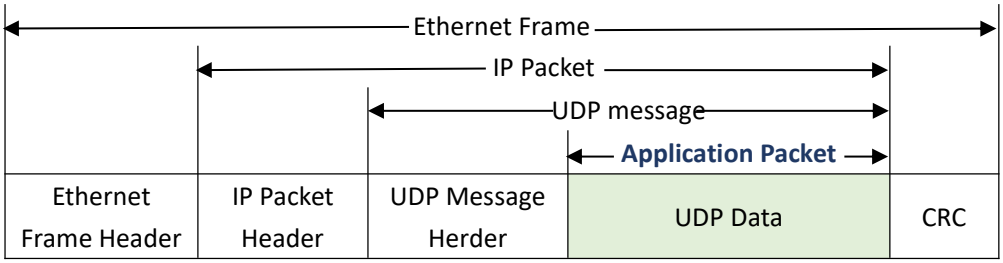
At the time of receiving, the receiving processing modules of different types of interfaces perform unpacking or decoding operations on the data, extract the application data packets, and send them to the queue of the system.

The forwarding engine of HDLC-LCM 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

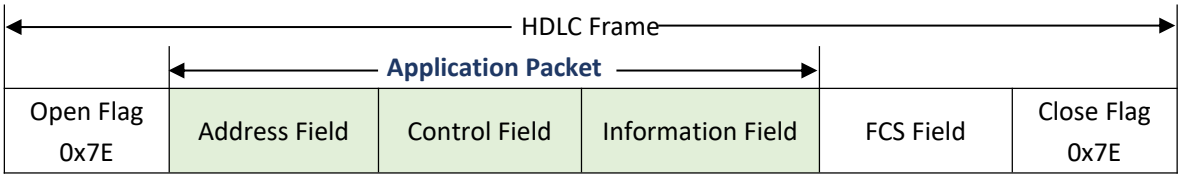
In the UDP protocol, the application packet is packaged in the data area of the UDP message. Each UDP packet contains a complete application packet.



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.

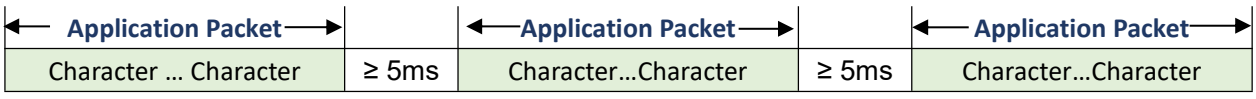
For HDLC-LCM, instead of distinguishing between address field, control field, and information field, they are uniformly presented as application packets to the upper application to fill in and process the UART packet format



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-LCM 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.

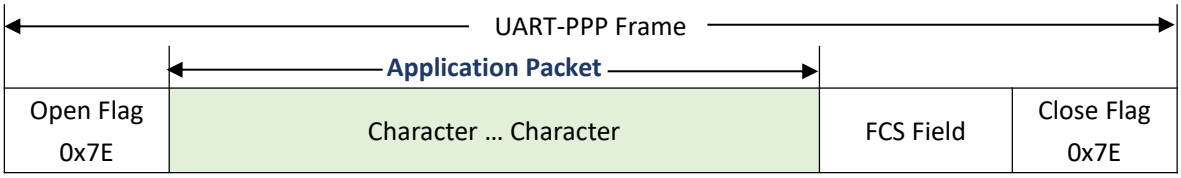


In the actual application, data transmission is not allowed during the packet interval; otherwise, it may result in a waste of communication bandwidth, and the higher the baud rate is, the more serious the waste is.

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.