HDLC-TCMS

Datasheet

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

yacer 亚册 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-TCMS protocol converter provides two 100M Ethernet interfaces, two RS-485 synchronous & asynchronous serial ports and two expansion interfaces to implement conversion between synchronous HDLC, asynchronous UART, CAN bus and Ethernet.

Industrial wide temperature, complete isolation and protection, compact size, suitable for train communication network.

On-board application CPU is used for secondary development of on-board programming by user, and seamlessly integrates data processing a communication.

1.2 Applications

- Protocol conversion between Synchronous and Asynchronous Serial interface；
- Protocol conversion between Serial and Ethernet interface;
- Protocol conversion between CAN and Serial interface;
- Protocol conversion between CAN and Ethernet interface;
- Train Communication Network (TCN);
- Train Control and Management System (TCMS);
- Embedded development and application.

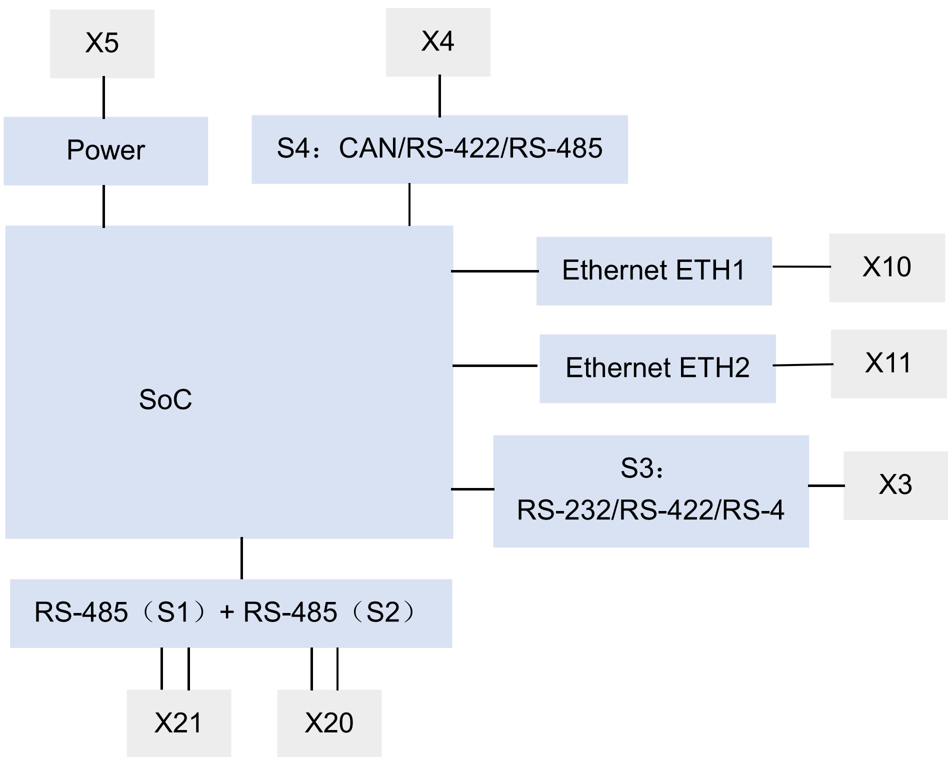
1.3 Features

- Two 10/100M Ethernet interfaces, supporting Ethernet switching and dual-IP;
- S1, S2: Two RS-485 synchronous and asynchronous serial ports with isolation;
- S3 expansion interface: Optional RS-232, RS-422 or RS-485 serial port;
- S4 expansion interface: Optional RS-422, RS-485 serial port or CAN bus;
- All serial ports support synchronous HDLC and asynchronous UART;
- Support NRZ, NRZI, DBPL, Manchester and differential Manchester coding formats;
- On-board application CPU for secondary programming;
- Perfect isolation protection;
- Industrial wide temperature.

1.4 Basic Function Blocks

The basic function blocks are shown in the following figure:

- S1, S2 are two RS-485 interfaces, which are connected to X20, X21;
- S3 expansion interface: Optional RS-232, RS-422 or RS-485, is connected to X3;
- S4 expansion interface: Optional CAN, RS-422 or RS-485, is connected to X4;
- ETH1, ETH2 are 100M Ethernet interfaces, which are connected X10, X11;
- X5 is the power connector.



1.5 Technical Specifications

Item	Parameters	Details
S1,S2 RS-485 Interface	Connector	1 x male D-Sub 9 (X20) + 1 x female D-Sub 9 (X21)
	Interface type	RS-485 half-duplex isolated serial
	Working mode	Synchronous HDLC, Asynchronous UART
	Coding formats	NRZ, NRZI, DBPL (Differential Bi-Phase Level), Manchester, Differential Manchester
	Baud rate	Synchronous: ≤ 6 Mbps Asynchronous: ≤ 3 Mbps
	Isolation	2.5 kVrms
S3 Expansion Interface	Connector	1 x male D-Sub 9 (X3)
	Optional type	RS-232 full-duplex serial RS-422 full-duplex isolated serial RS-485 half-duplex isolated serial
	Working mode	Synchronous HDLC, Asynchronous UART
	Coding formats	NRZ, NRZI, DBPL (Differential Bi-Phase Level), Manchester, Differential Manchester
	Baud rate	Synchronous ≤ 6 Mbps Asynchronous ≤ 3 Mbps
	Isolation	2.5 kVrms
S4 Expansion Interface	Connector	1 x male D-Sub 9 (X4)
	Optional type	CAN bus isolation interface (CAN2.0A, CAN2.0B, ISO 11898) RS-422 full-duplex isolated serial RS-485 half-duplex isolated serial
	Working mode	Synchronous HDLC, Asynchronous UART
	Coding formats	NRZI, DBPL (Differential Bi-Phase Level), Manchester, Differential Manchester
	Baud rate	Synchronous ≤ 6 Mbps Asynchronous ≤ 3 Mbps CAN: 50 Kbps ~ 1 Mbps
	Isolation	2.5 kVrms
ETH1, ETH2	Connector	2 x M12 with D-coding

Item	Parameters	Details
Ethernet Interface	Function	Ethernet switching, dual-IP function
	Rate	10/100 Mbps, supporting MDI / MDIX adaptation
	Protocol	TCP/IP
	Programming interface	UDP Server, UDP Client Unicast/Multicast/Broadcast
	Isolation	1.5 kVrms
Secondary development On-board programming	Application CPU	ARM Cortex-A9, main frequency 250 MHz
	Memory	DDR3, 128MB
	Flash	6 MB version space, 1 MB configuration space
	Data interface	Interacting with communication CPU based on shared memory
Configuration Management	Configuration tool	yacer-DMS configuration management software
	Console interface	Ethernet Interface or Serial
Power Requirements	Power Supply	LV: Nominal 24V, tolerance 9 ~ 36VDC MV: Nominal 36V, 48V, tolerance 18 ~ 75VDC HV: Nominal 72V, 96V, 110V, tolerance 40 ~ 160VDC
	Isolation	1.5 kVrms
	Protection	Anti-reverse protection
	Power consumption	< 3 W
	Connector	3 pin connector with 5mm pitch (Phoenix Contact MSTB 2,5 / 3-GF or equivalent)
Mechanical Characteristics	Dimensions	H x W x D: 124 mm x 36 mm x 104 mm
	Weight	500g
Operating Environment	Operating temperature	-40 ~ +70℃
	Storage temperature	-40 ~ +85℃
	Operating humidity	5 ~ 95% RH (no condensation)

1.6 Order Information

HDLC-TCMS - 4	3	5	- LV
400 series products: Two 100M Ethernet interfaces Two RS-485 serial ports			
S3 expansion interface:			
● None	0		
● Full-duplex RS-232	3		
● Full-duplex RS-422	4		
● Half-duplex RS-485	5		
S4 expansion interface:			
● None		0	
● Full-duplex RS-422		4	
● Half-duplex RS-485		5	
● CAN bus interface		6	
Supply voltage range:			
● Nominal 24V, tolerance 9 ~ 36VDC			LV
● Nominal 36V, 48V, tolerance 18 ~ 75VDC			MV
● Nominal 72V, 96V, 110V, tolerance 40 ~ 160VDC			HV

2 Hardware and Physical Interface

2.1 Appearance

2.2 LED Indicators

LED	Description
RUN	Running indicator, flashing during normal operation
ALARM	Alarm indicator, on when the device is not ready or fails, and constantly off during normal operation
S1	Serial port S1 indicator, blinking once after data is received or transmitted
S2	Serial port S2 indicator, blinking once after data is received or transmitted
S3	Expansion interface S3 indicator, blinking once after data is received or transmitted
S4	Expansion interface S4 indicator, blinking once after data is received or transmitted
LINK/ACT	Link/ACT indication of the Ethernet interfaces

2.3 Ethernet Interfaces ETH1, ETH2 (X10, X11)

2.3.1 Function Description

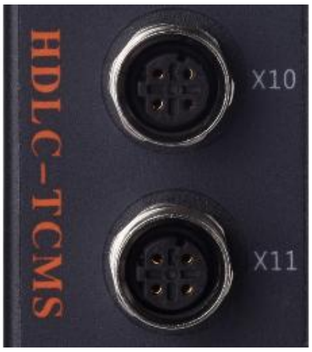
ETH1, ETH2 are two 10/100M Ethernet interfaces with connectors X10 and X11 as M12 (D-coding).

There are two working modes for the Ethernet interfaces:

- Ethernet switching mode: Enable the built-in Ethernet switching function;
- Dual IP mode: Each Ethernet interface has an independent IP address.

2.3.2 Pin Definition

X10, X11 Pin	Description
1	TD +
2	RD +
3	TD -
4	RD -

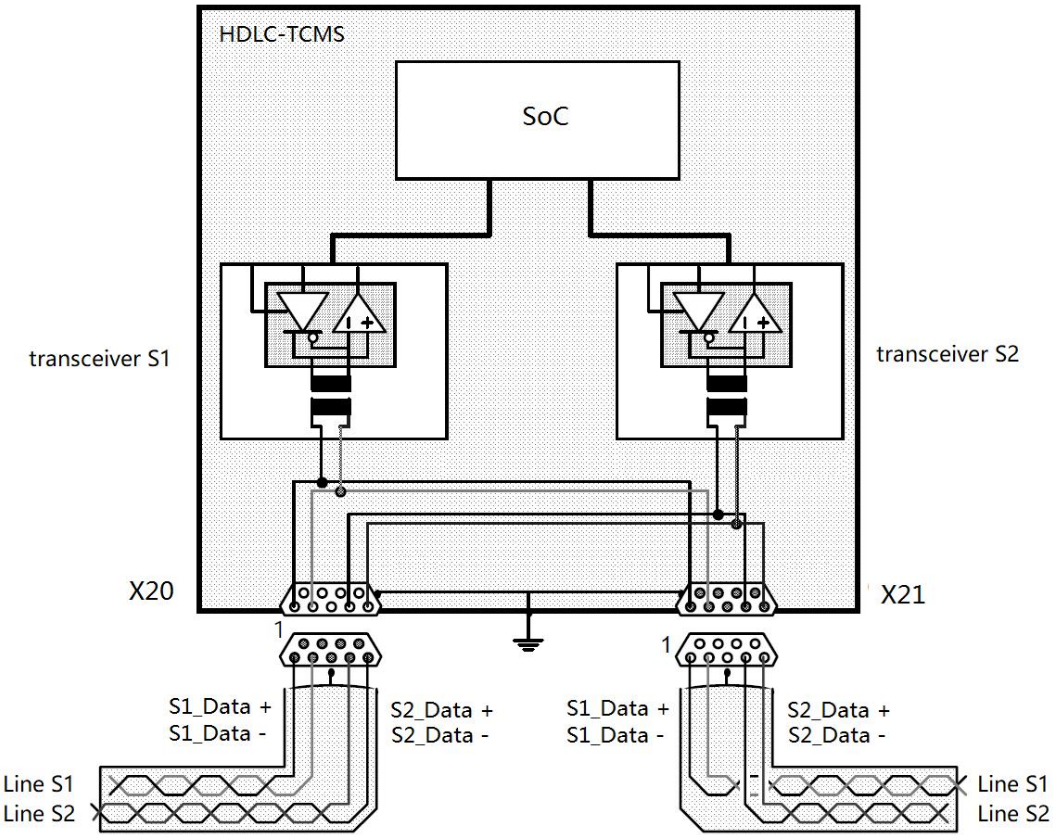


2.4 Dual RS-485 Interface S1, S2 (X20, X21)

2.4.1 Function Description

S1 and S2 are two band-isolated half-duplex RS-485 serial ports, supporting synchronous HDLC protocol and asynchronous UART working mode, encoding format supports NRZI, Manchester, differential Manchester, DBPL, etc.

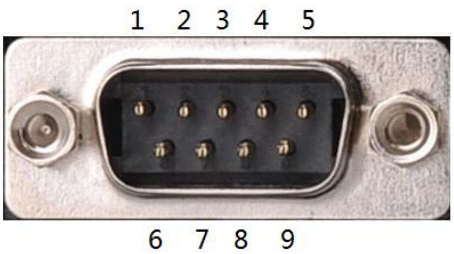
For easy connection and terminal matching of RS-485 bus, S1 and S2 are connected to connectors X20 (male) and X21 (female) together.



2.4.2 Pin Definition

Interfaces S1 and S2 are connected to both X20 and X21 connectors.

Connector models are shown in the following table:

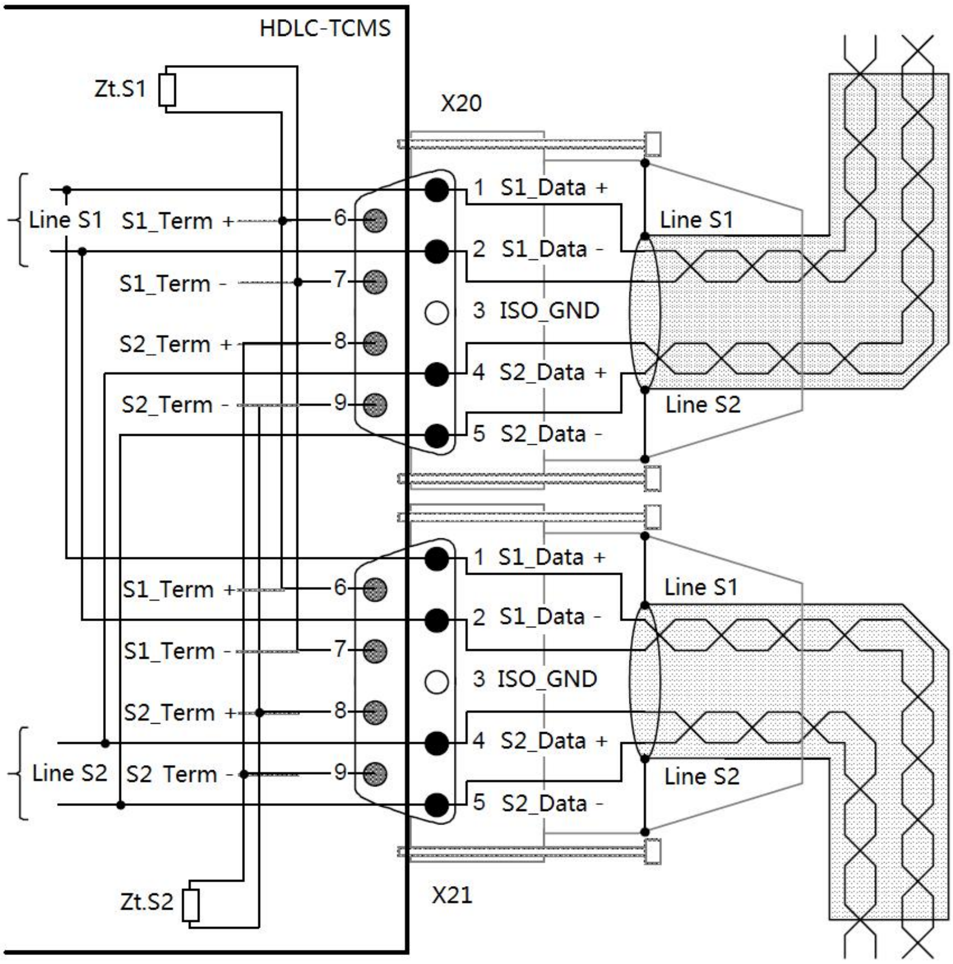
X20 (Male D-Sub 9)	X21 (Female D-Sub 9)
	

Connector pins are defined as follows:

Pin	X20 (Male)	X21 (Female)	Description
1	S1_Data +	S1_Data +	Serial port S1 RS-485 Data +
2	S1_Data -	S1_Data -	Serial port S1 RS-485 Data -
3	ISO_GND	ISO_GND	Isolated GND
4	S2_Data +	S2_Data +	Serial port S2 RS-485 Data +
5	S2_Data -	S2_Data -	Serial port S2 RS-485 Data -
6	S1_Term +	S1_Term +	Short connections 1-6, 2-7 enable terminal

Pin	X20 (Male)	X21 (Female)	Description
7	S1_Term -	S1_Term -	matching for serial port S1
8	S2_Term +	S2_Term +	Short connections 4-8, 5-9 enable terminal matching for serial port S2
9	S2_Term -	S2_Term -	

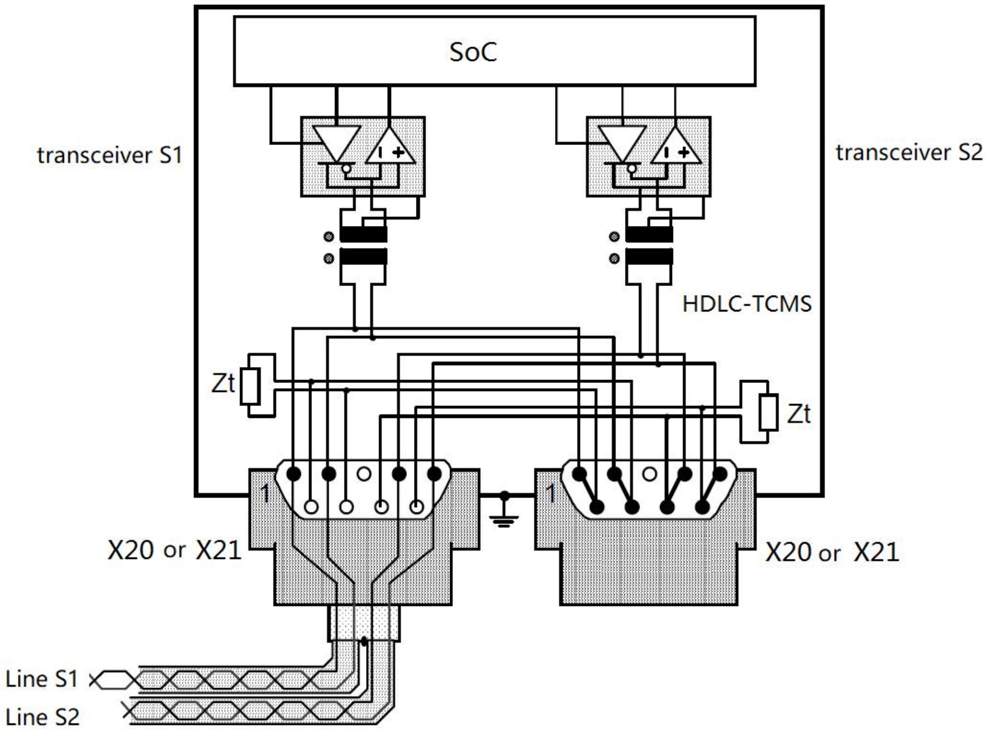
The signal definition diagram is as follows:



2.4.3 Terminators RS-485

At the end of the bus, the terminals are shortened as follows: 1-6, 2-7, 4-8, 5-9.

The resistance value of the matching resistance defaults to 120 ohms and needs to be customized when the product is manufactured if additional resistance values are required.



2.4.4 Clock Mode Pin Definition

S1 and S2 can be merged into a serial port to support NRZ coding format, and the pin definition is as follows :

Pin	X20 (Male)	X21 (Female)	Description
1	S1_Data +	S1_Data +	Serial port S1 RS-485 Data +
2	S1_Data -	S1_Data -	Serial port S1 RS-485 Data -
3	ISO_GND	ISO_GND	Isolated GND
4	S1_Clock +	S1_Clock +	Serial port S1 RS-485 Clock +
5	S1_Clock -	S1_Clock -	Serial port S1 RS-485 Clock -
6	S1_Data_Term +	S1_Data_Term +	Short connections 1-6, 2-7 enable terminal matching for S1 data
7	S1_Data_Term -	S1_Data_Term -	
8	S1_Clock_Term +	S1_Clock_Term +	Short connections 4-8, 5-9 enable terminal matching for S1 clock
9	S1_Clock_Term -	S1_Clock_Term -	

2.5 RS232/422/485 Interface S3 (X3)

2.5.1 Function Description

S3 is an expansion serial port which supports synchronous HDLC protocol and asynchronous UART working mode. The encoding format supports NRZI, Manchester, Differential Manchester, DBPL, etc.

Users can choose one of the following types when ordering:

- RS-232 full-duplex;
- RS-422 full-duplex with isolation;
- RS-485 half-duplex with isolation.

2.5.2 Pin Definition

S3 connector X3 uses male D-Sub 9 male connector, pin defined as follows:

Pin	RS-232 Full Duplex	RS-422 Full Duplex	RS-485 Half Duplex
1			
2	RXD		
3	TXD	ISO_GND	ISO_GND
4		TXD +	Data +
5	GND	TXD -	Data -
6			
7			
8		RXD +	Term +
9		RXD -	Term +

2.5.3 Terminator RS-485

In the RS-485 mode, 8-9 short connection enables the terminal matching.

The resistance value of the matching resistance defaults to 120 ohms and needs to be customized when the product is manufactured if additional resistance values are required.

2.6 CAN/422/485 Interface S4 (X4)

2.6.1 Function Description

S4 is an expansion interface, users can choose serial port or CAN bus interface when ordering. When the serial port is configured, one of the following configurations can be selected from the factory:

- RS-422: full-duplex with isolation;
- RS-485: half-duplex with isolation.

The serial port supports synchronous HDLC protocol and asynchronous UART mode,

The coding format supports NRZI, Manchester, differential Manchester, DBPL, etc.

2.6.2 Pin Definition

S4 connector X4 uses male D-Sub 9 male connector, pin defined as follows:

Pin	**CAN**	**RS-422 Full Duplex**	**RS-485 Half Duplex**
1	Term +		
2	CAN_L		
3	ISO_GND	ISO_GND	ISO_GND
4		TXD +	Data +
5		TXD -	Data -
6	Term -		
7	CAN_H		
8		RXD +	Term +
9		RXD -	Term -

2.6.3 Terminator CAN bus

In the CAN working mode, 1-6 short connection enables the terminal matching of CAN bus.

The resistance value of the matching resistance defaults to 120 ohms and needs to be customized when the product is manufactured if additional resistance values are required.

2.6.4 Terminator RS-485

In the RS-485 mode, 8-9 short connection enables the terminal matching.

The resistance value of the matching resistance defaults to 120 ohms and needs to be customized when the product is manufactured if additional resistance values are required.

2.6.5 Clock Mode Pin Definition

S3 and S4 can be merged into a serial port to support NRZ coding format, the number is S3.

At this point, the definition of X3 pin is unchanged, X4 as the clock signal, its definition is as follows :

Pin	RS-422 Full Duplex	RS-485 Half Duplex
1		
2		
3	ISO_GND	ISO_GND
4	TXC +	Clock +
5	TXC -	Clock -
6		
7		
8	RXC +	Term +
9	RXC -	Term -

2.7 Power Interface (X5)

2.7.1 Function Description

HDLC-TCMS is powered by DC power supply, supports isolation protection and surge protection, and provides anti-inverse protection.

Depending on the factory configuration, the power input range is as follows:

Model	Nominal Value	Minimum	Maximum
LV	24V	9V	36V
MV	36V、48V	18V	75V
HV	72V、96V、110V	40V	160V

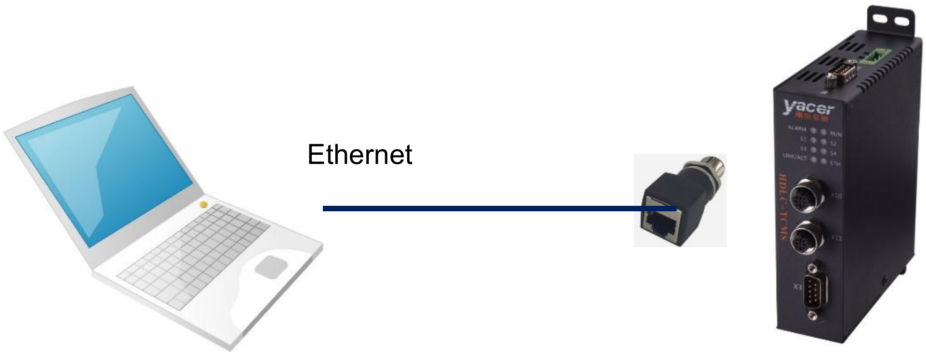
2.7.2 Pin Definition

The X5 uses a 3-pos 5mm terminal connector (Phoenix Contact MSTB 2,5 / 3-GF compatible).

Pin	Name	Description
1	V+	Voltage +
2	CASE	Protection Earth
3	V-	Voltage -

3 Building Configuration Environment

Connect the management computer with any Ethernet interface port of HDLC-TCMS through network cable, and run yacer-DMS configuration management software on the computer to configure the parameters and monitor running status of HDLC-TCMS.



4 Configuration management software

4.1 Get configuration management software yacer-DMS

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

- “Softwares” directory of HDLC-TCMS accompanied U-Disk;
- Official website of Yacer (http://www.yacer.com.cn) Software channel.

4.2 Run yacer-DMS software

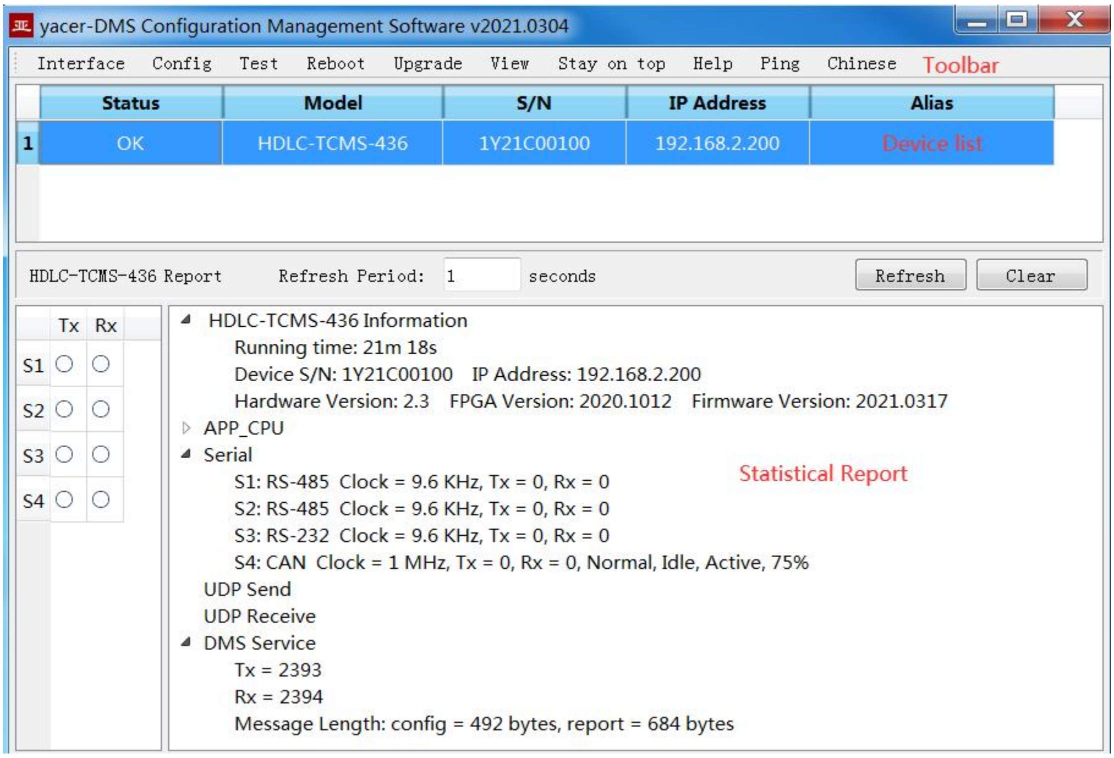
The yacer-DMS is an installation free application software, unzip yacer-DMS.zip, enter the working directory and double click the file yacer-DMS.exe to run.

4.3 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.



4.4 Statistical Report

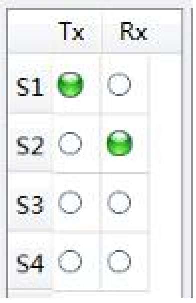
The statistical report has three panels: control panel, receive/transmit indication panel and information panel.

4.4.1 Control Panel

Control Widget	Function
Refresh Period: 1 seconds	Statistical report refresh cycle
Refresh	Manual refresh operation
Clear	Clear the statistical report

4.4.2 Receive/Transmit Indication Panel

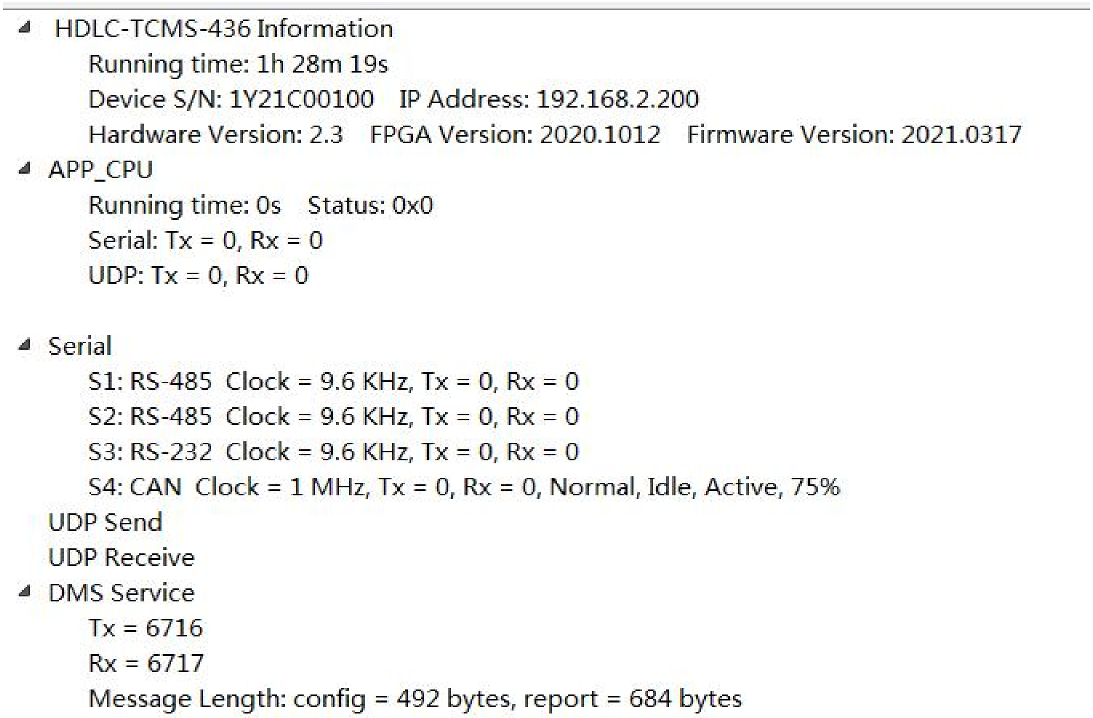
- Tx: The interface sends a frame of data, corresponding Tx indicator blinks once;
- Rx: The interface receives a frame of data, corresponding Rx indicator blinks once.



4.4.3 Information Panel

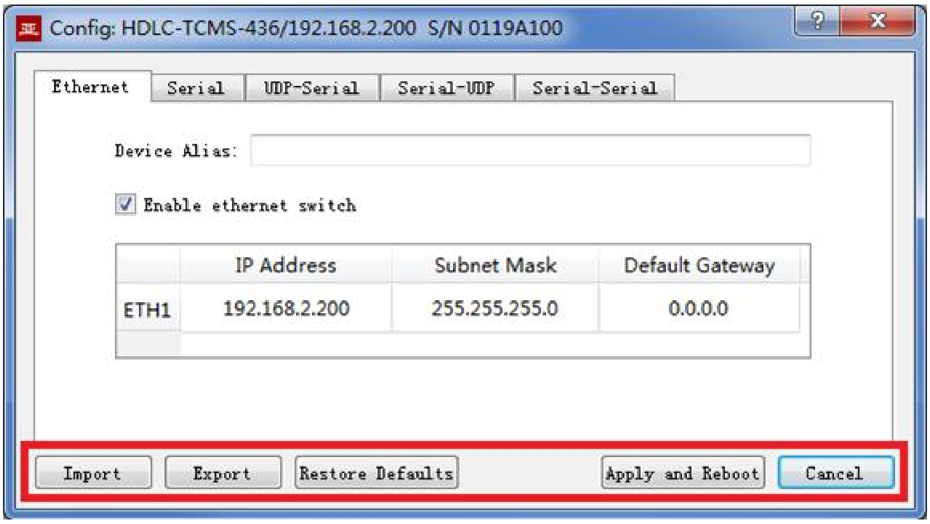
The right side of the statistical report is the information panel, which can display the following contents:

- Device information: Running time, S/N, IP address and Version number;
- APP_CPU: Application CPU operation information;
- 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.



4.5 Configure Device

Click the 'Config' button on the toolbar or double-click the selected device in the device list, 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:

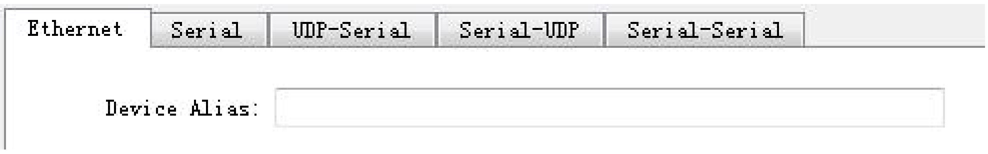
Button	Function
Import	Open the configuration file, read the configuration parameters refresh the configuration dialog
Export	Export configuration parameters from the configuration dialog to a file for saving
Restore Defaults	Refresh the configuration dialog with the factory paramters
Apply and Reboot	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

5 Function and Configuration

5.1 Ethernet Interface（ETH1、ETH2）

5.1.1 Device alias

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



5.1.2 IP configuration

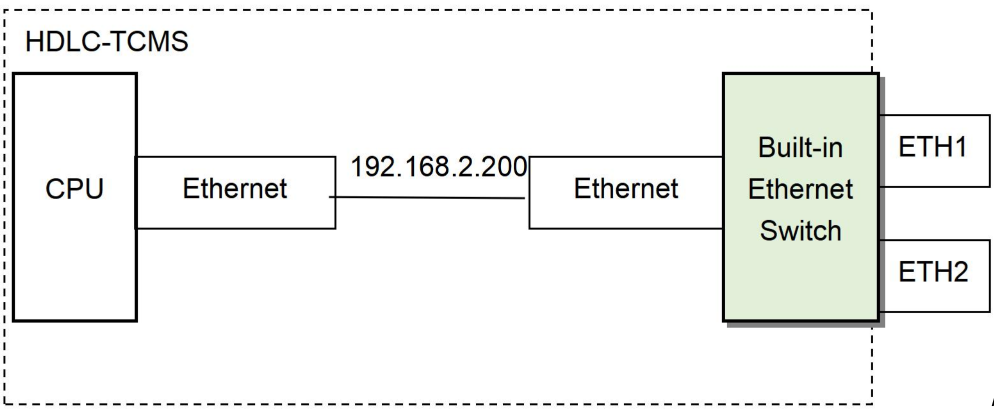
5.1.2.1 Enable Ethernet switch

By default, check the “Enable ethernet switch” checkbox to enable the built-in Ethernet switch, to provide Ethernet switching function between ETH1 and ETH2.

☑ Enable ethernet switch

	IP Address	Subnet Mask	Default Gateway
ETH1	192.168.2.200	255.255.255.0	0.0.0.0

After enabling the Ethernet switching function, HDLC-TCMS only has one IP address. The network functions are as follows:



/p;[]

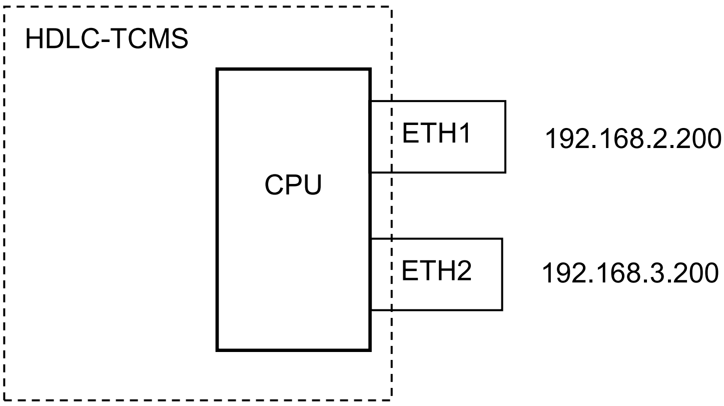
5.1.3 Dual IP Configuration

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

☐ Enable ethernet switch

	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-TCMS is equivalent to a PC equipped with two network cards.

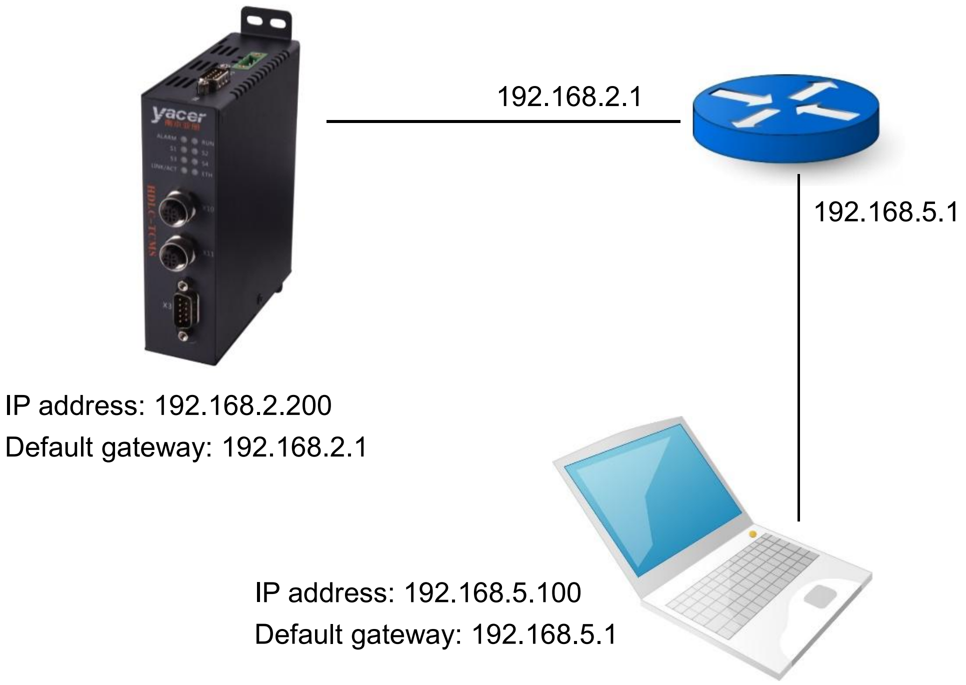


5.1.4 Default Gateway

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

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



5.2 Serial Port（S1 ~ S4）

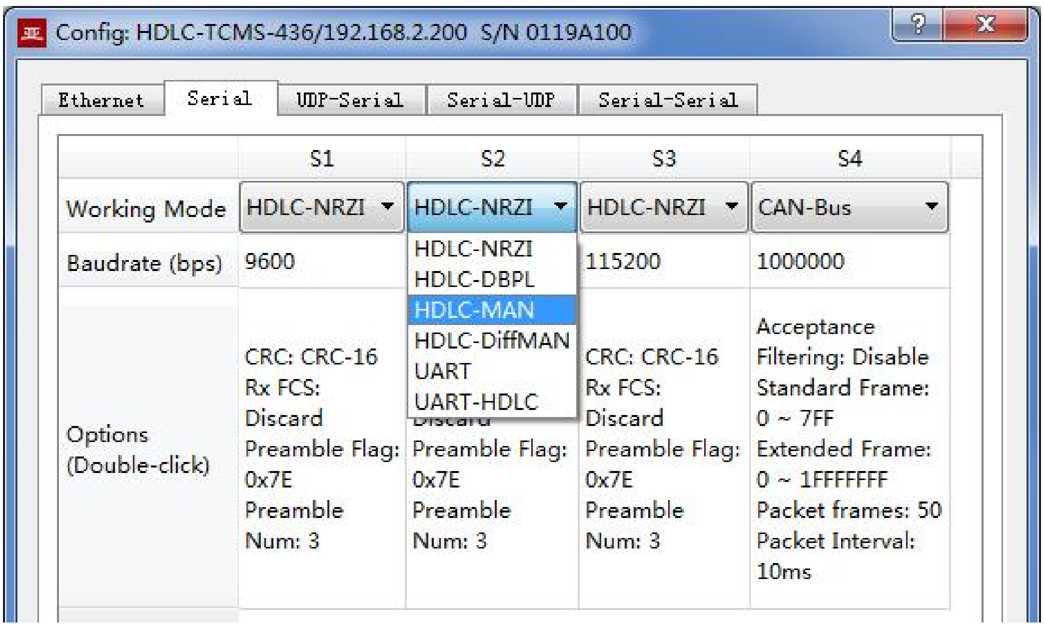
5.2.1 No Clock Working Mode

S1 and S2 are RS-485 half-duplex serial ports, S3 can be configured as RS-232/422/485 serial port when leaving the factory, S4 can be configured as RS-422/485 serial port when leaving the factory, supporting the synchronous and asynchronous working modes described in the table below.

Working Mode		Description
Synchronous	HDLC-NRZI	Synchronous HDLC protocol based on the NRZI encoding
	HDLC-DBPL	Synchronous HDLC protocol based on the DBPL (Differential Bi-Phase Level) encoding
	HDLC-MAN	Synchronous HDLC protocol based on the Manchester encoding
	HDLC-DiffMAN	Synchronous HDLC protocol based on the differential Manchester encoding
Asynchronous	Asynchronous UART	Universal asynchronous serial mode, similar to serial port on a universal computer
	Asynchronous HDLC	UART-based HDLC-like communication protocol

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.



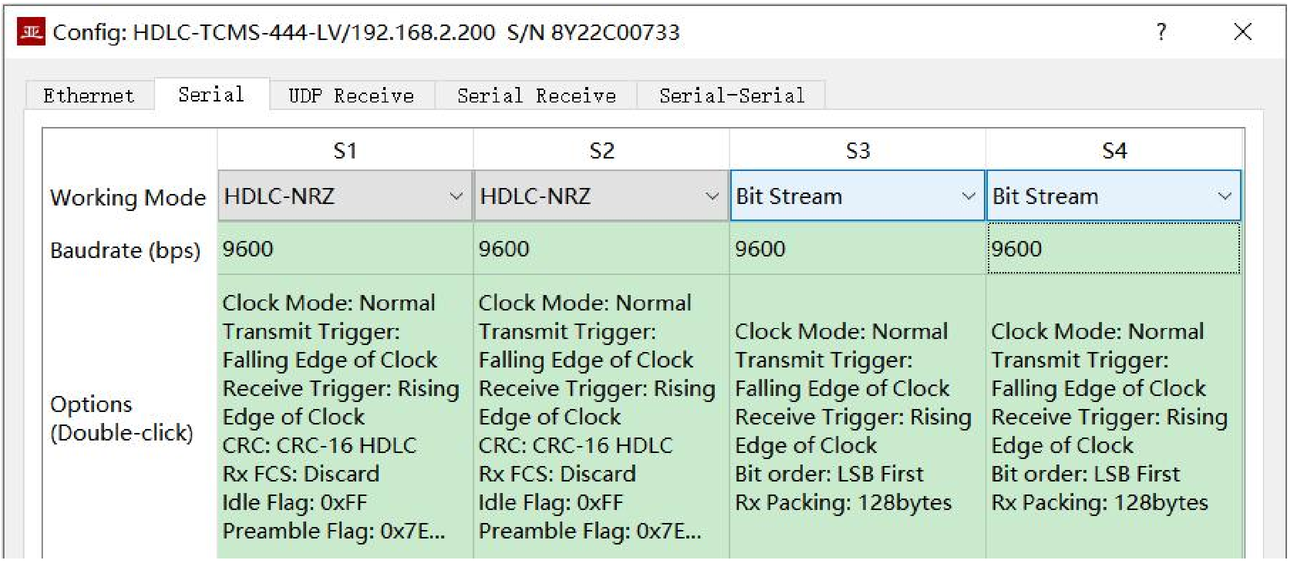
5.2.2 Clock Working Mode

S1 and S2 can be merged into a RS-485 half-duplex synchronous serial port with clock, and the serial number is S1.

If X3 and X4 are configured to RS-422 or RS-485 at the same time, the interface S3 and S4 can be merged into a synchronous serial port with clock, and the serial number is S3.

Clock working mode support :

Working Mode		Description
Synchronous	HDLC-NRZ	Synchronous HDLC protocol based on the NRZ encoding
	Bit Stream	Transmit or sampling serial bit data based on receiving clock

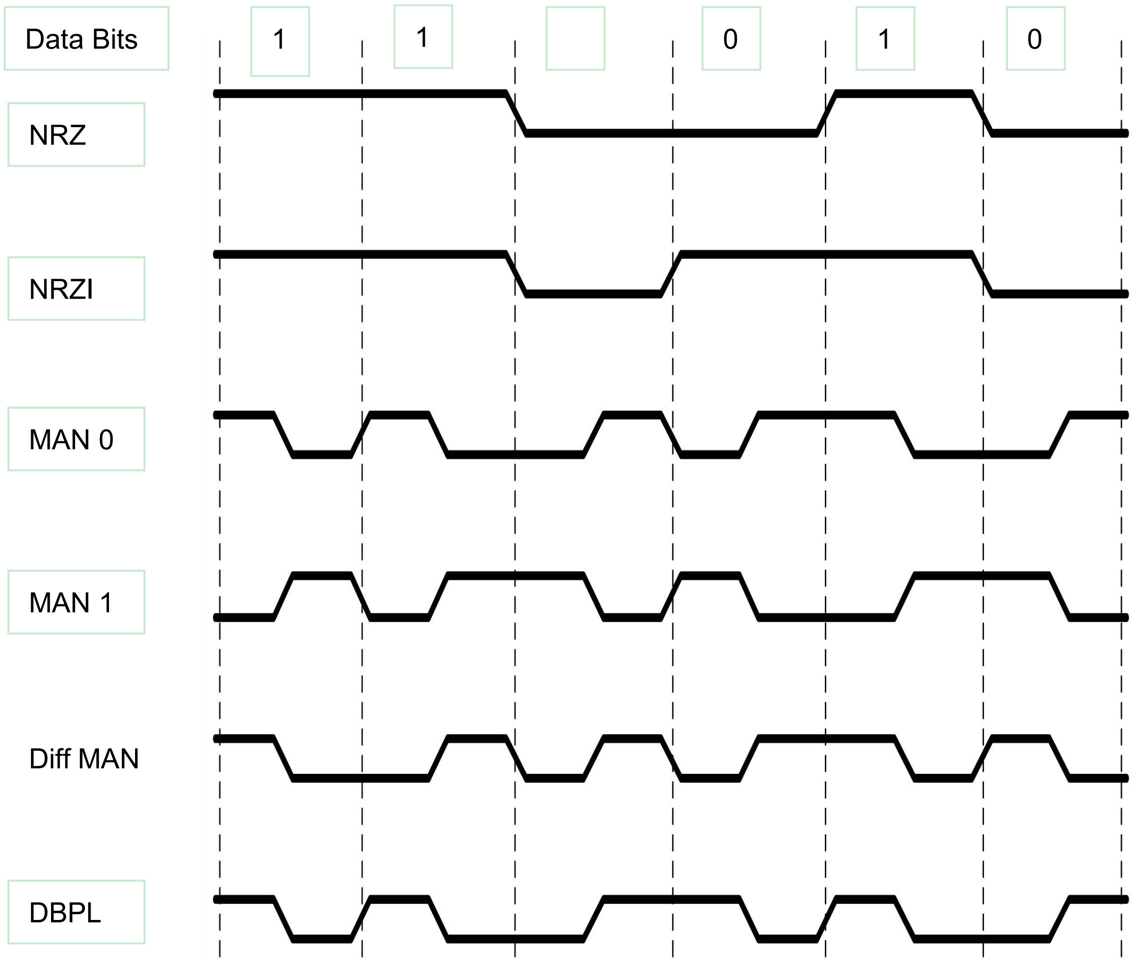


5.2.3 Baud rate

No matter which mode the serial port works in, the baud rate of both sides of the communication must be the same to ensure reliable and stable data communication.

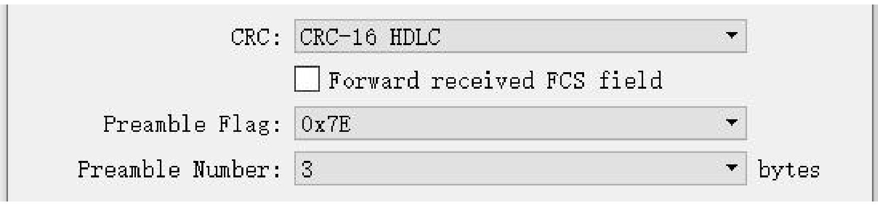
5.2.4 Encoding format of the synchronous serial port

For 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:



5.2.5 HDLC-NRZI Options

The option dialog of the HDLC-NRZI working mode is shown as follows:



5.2.5.1 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 disable: ● Send: No CRC calculation, no FCS field for HDLC frame ● Receive: No CRC checking
CRC-16 HDLC	16-bit ISO HDLC CRC verification
CRC-16 SDLC	16-bit IBM SDLC CRC verification

5.2.5.2 Forward received FCS field

This configuration will only take effect if CRC is enabled.

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

Open Flag	Address Field	Control Field	Information Field	FCS Field	Close Flag
0x7E	1 Byte	1 Byte	Variable Length	2 Bytes	0x7E
0x7E	User Data			CRC	0x7E

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

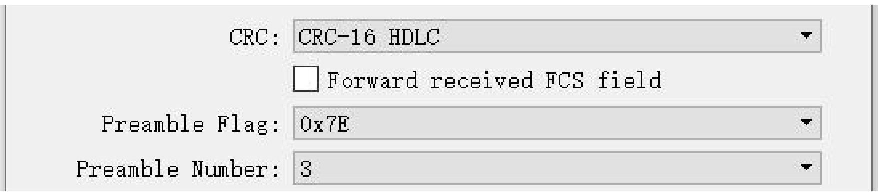
If this check box is not checked, HDLC-TCMS, after receiving HDLC frames and performing CRC checks, discards the FCS field of the last 2 bytes of data and forwards only user data.

5.2.5.3 Preamble Flag & Number

In half-duplex communication, it is often necessary to add preamble flags in front of the frame for receiver synchronization. The most common is to add 2-5 0x7E flag.

5.2.6 HDLC-DBPL Options

HDLC-DBPL uses the Differential Bi-Phase Level encoding format, and its Options dialog box is shown below:

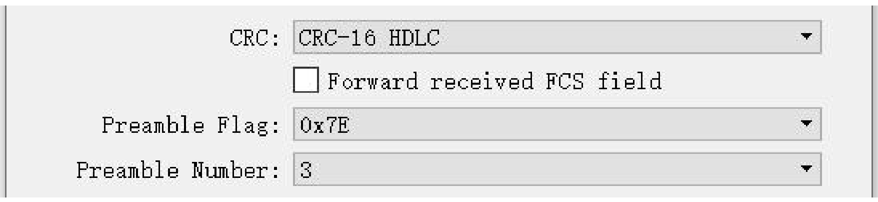


The parameters of HDLC-DBPL have the same meaning as HDLC-NRZI.

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.

5.2.7 HDLC-DiffMAN (differential Manchester) Options

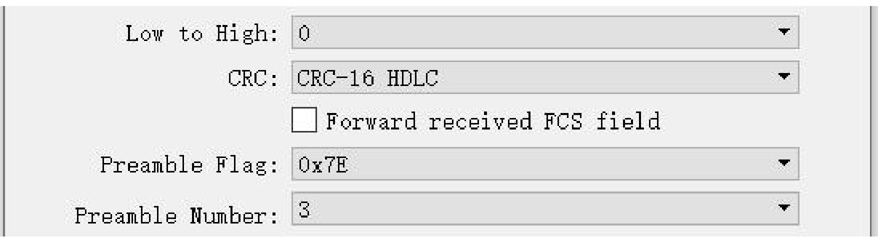
The Differential Manchester Options dialog box is shown below:



The parameters of HDLC-DiffMAN have the same meaning as HDLC-NRZI.

5.2.8 HDLC-MAN (Manchester) Options

The Manchester Options dialog box is shown below:



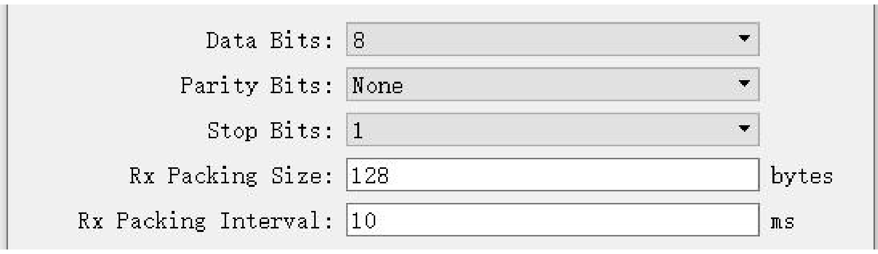
In addition to the same configuration parameters as NRZI, the Manchester encoding format has parameters with the meaning of edges with low to high waveforms for data lines:

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

5.2.9 UART Options

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-TCMS 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.

5.2.9.1 Packing Size

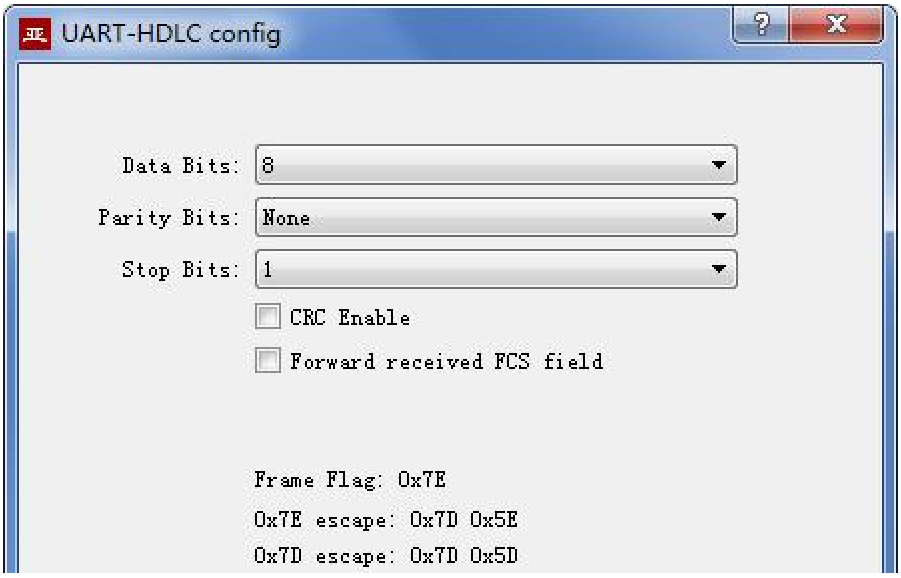
For example, if the Packing Size is set to 128 bytes, then when UART receives 128 bytes, a packet will be formed for forwarding.

5.2.9.2 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.

5.2.10 UART-HDLC Options

The UART-HDLC working mode is a customized 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.



The UART-HDLC 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:

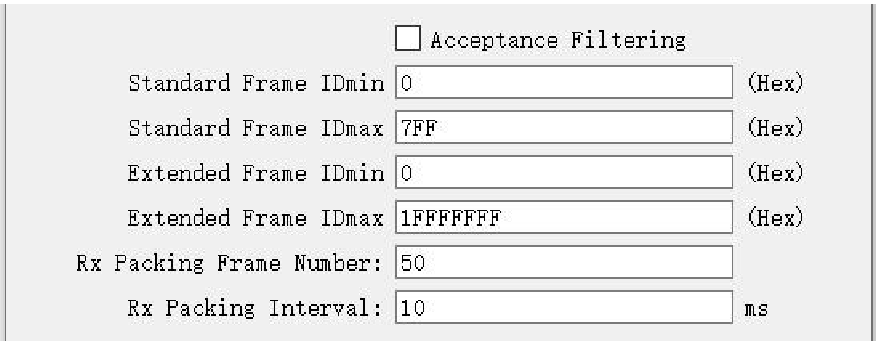
Original Data	Actual Transmit Data
0x7E	0x7D 0x5E
0x7D	0x7D 0x5D
Others	No change

The escape operation of data receive is as follows:

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

5.2.11 CAN Bus Options

S4 can be selected as CAN bus interface in the factory, and the configuration is as follows:



Because the CAN frame is very short, if each CAN frame 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-TCMS buffers the received can frames, and then forwards the buffered CAN frames into a packet, which is called packing.

Packing is controlled by two parameters, which are called Packing Frame Number & Packing interval.

5.2.11.1 Packing Frame Number

The maximum Packing Frame Number is 50.

As shown in the figure above, the Packing Frame Number is 50. When HDLC-TCMS receives 50 CAN frames, it is assembled into a data packet for forwarding.

5.2.11.2 Packing Interval

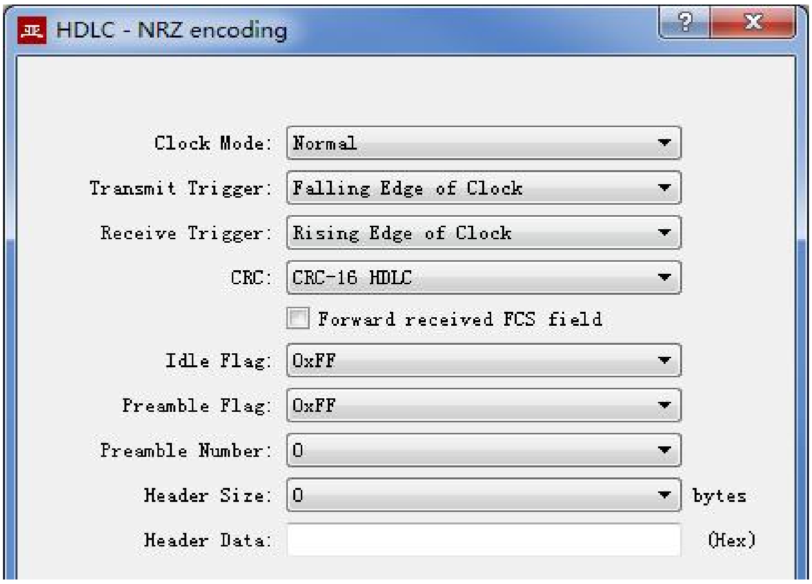
As the example above sets the packet interval to 10 ms, if no new CAN frame is received for more than 10 ms, the CAN frames of the buffer will be combined into a single packet to forward regardless of whether they are full or not.

5.2.11.3 Acceptance filtering

The acceptance filtering allows users to set the frame ID range that is allowed to be received.

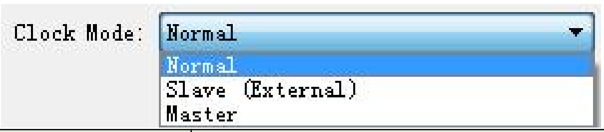
5.2.12 HDLC-NRZ Options

HDLC-NRZ is the common synchronous working mode, which is relies on receiving and receiving clock signals to achieve data bit synchronization, so the configuration of clock parameters is particularly important.



5.2.12.1 Clock Mode

There are 3 clock modes for the synchronous serial port, normal, slave clock & master clock.



Clock Mode	Transmit Clock	Receive Clock
Normal	Generation from the local device, output through pin TxC.	Generation from the remote terminal device, input through pin RxC.
Slave Clock (External)	Generation from the peer device, input through pin RxC. TxC output synchronizes with RxC automatically.	Generation from the remote terminal device, Input through pin RxC.
Master Clock	Generation from the local device, output through pin TxC.	Generation from the local device, ignoring the clock of pin RxC.

The slave clock mode is also called as the external clock working mode. When the remote terminal device is the DCE (Data Communication Equipment), HDLC-TCMS 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.

5.2.12.2 Transmit Trigger

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

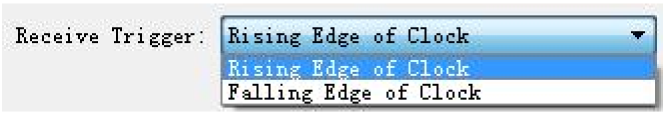
- Falling edge of clock: A new data bit is generated on the falling edge of clock
- Rising edge of clock: A new data bit is generated on the rising edge of clock

For communications that follow the HDLC protocol specification, the clock drop edge should be selected to trigger new data transmission. There are also some special applications where users use non-standard communication and use rising edge to trigger new data transmission.

5.2.12.3 Receive Trigger

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



In accordance with HDLC protocol specification for communication, since the falling edge is used to trigger new data, considering the stable time of new data, in order to ensure the correct reading of data, the receiving trigger must be configured as the clock rising edge.

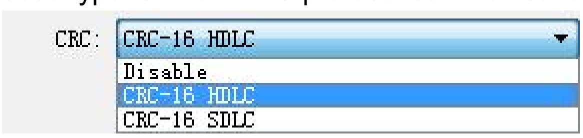
The local receive trigger configuration is determined according to the transmit trigger of the remote terminal device:

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

5.2.12.4 CRC

In order to verify the correctness of data communication, CRC function should be enabled.

By default, configure the protocol CRC check type with CRC-16-HDLC as the most commonly used type for the HDLC protocol communication.



CRC	Description
Disable	● CRC disable: ● No CRC calculation for data transmission or FCS

	field for HDLC frame ● No CRC check for data receiving
CRC-16 HDLC	Adopt the 16-bit IBM HDLC CRC check method
CRC-16 SDLC	Adopt the 16-bit IBM SDLC CRC check method

5.2.12.5 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.

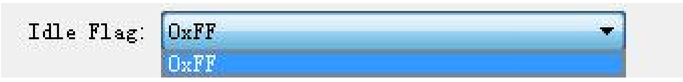
Opening Flag	Address Field	Control Field	Information Field	FCS Field	Closing Flag
0x7E	1 byte	1 byte	Variable length	CRC 2/4 bytes	0x7E
0x7E	User data			CRC 2/4 bytes	0x7E

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

If this option is not checked, HDLC-TCMS will discard the 2/4-byte FCS field at the end of data and only forward the user data after the receive HDLC frame check is passed.

5.2.12.6 Idel Flag

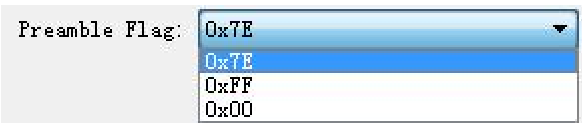
The definition of HDLC inter frame filling content, the default should be 0xFF.

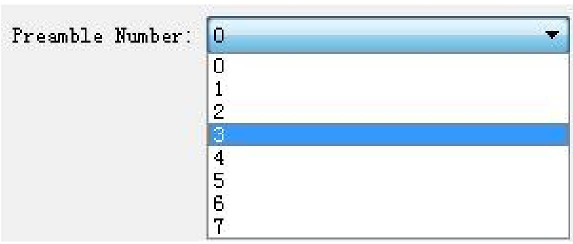


5.2.12.7 Preamble Flag and Number

During the half-duplex communication, a preamble flag is often required in front of the frame for receiving party synchronization, and the most commonly used method is to add 2~5 0x7E.

For full duplex applications, the Preamble number is often unrequired, set it to 0(no preamble).





5.2.12.8 Header Size and Data

As shown above, the header size is defined as 2, and the header data is defined as FF 03.

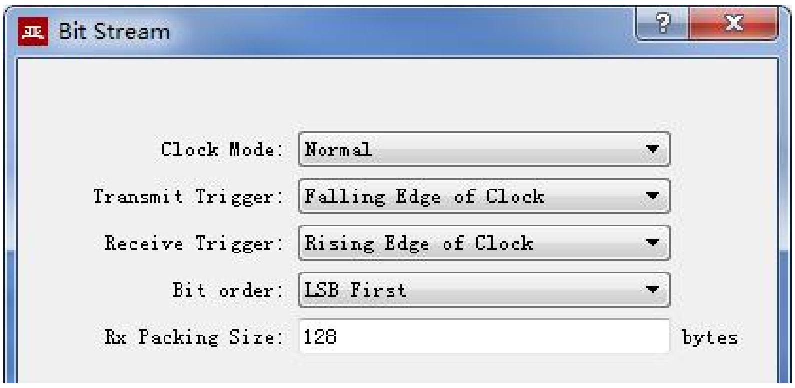
While transmitting HDLC, the FF 03 is added before the user data, and HDLC frame data is composed with user data.

When receiving HDLC, HDLC-TCMS discards the first 2 bytes of HDLC frame data as the frame header, and only forwards the subsequent data to the user.

Opening Flag	Frame Header	User Data	FCS Field	Closing Flag
0x7E	0xFF 0x03	Variable length	CRC 2/4byte	0x7E

5.2.13 Bit Stream Options

The rising or falling edge of each clock cycle samples the 1bit data on the data line, which forms a UDP message and transmits 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 and receive trigger.

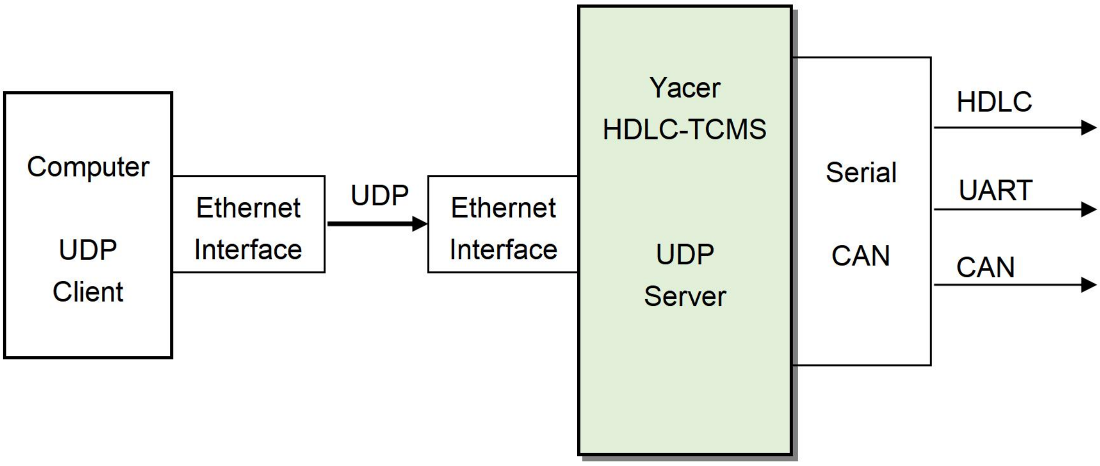
The online bit stream is stored in the computer or system memory in the form of byte. The receive/transmit sequence determines the conversion mode of byte and bit.

5.3 UDP to Serial

5.3.1 Function Description

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

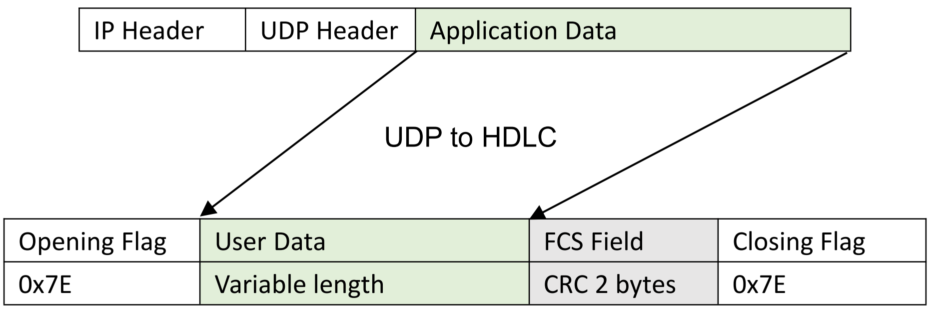
The typical application is shown as follows. PC sends a UDP message over the Ethernet interface as the UDP Client, and HDLC-TCMS sends it out from the synchronous serial port after converting the received UDP message into the HDLC frame.



5.3.2 Protocol Conversion

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

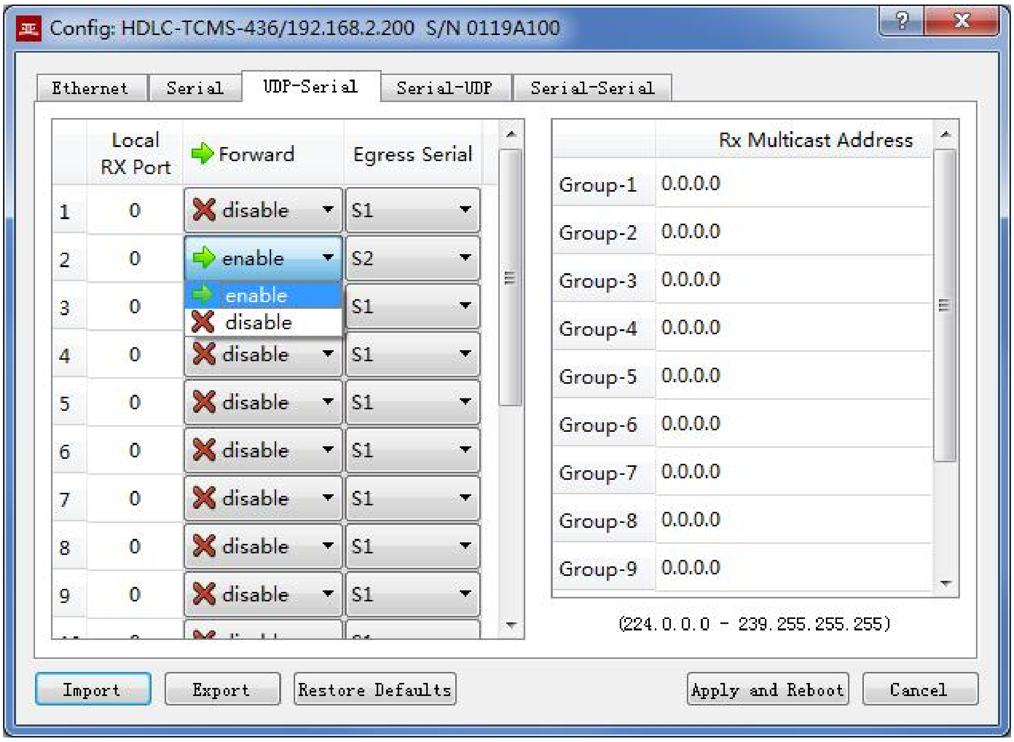
In order to reduce the computational load of the computer and the complexity of user programming, generally, the UDP message does not contain the FCS field of HDLC, which is calculated and filled by HDLC-TCMS.



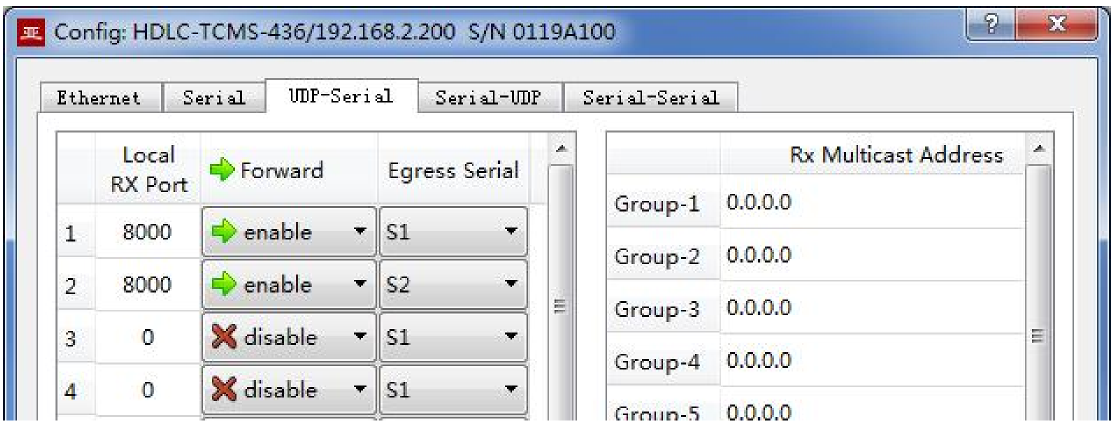
5.3.3 Forward Configuration

Set UDP to serial ports, each line represents a UDP port to serial forwarding entry, select "enable" to take effect. Three forwarding strategies can be implemented

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



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



5.3.4 Receive UDP Multicast

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

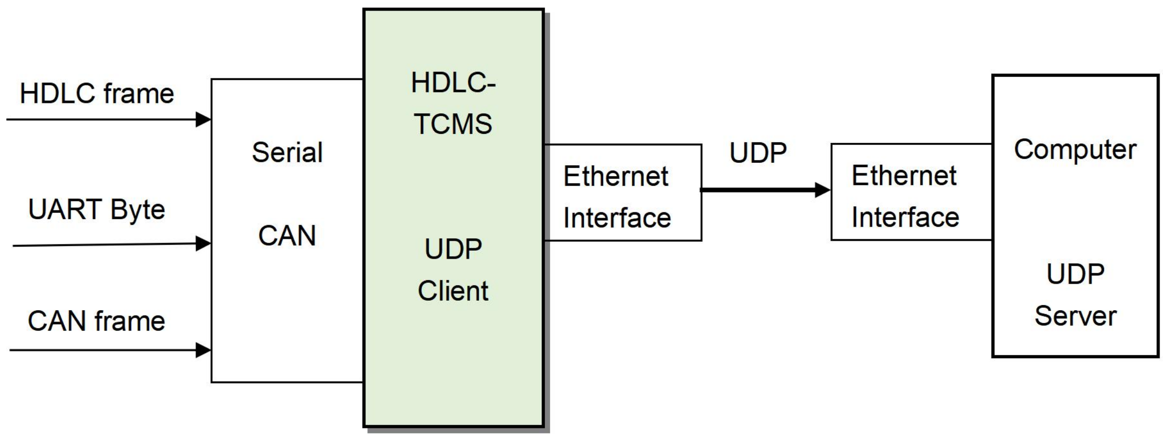
Range of the multicast address is 224.0.0.0 ~ 239.255.255.255, 224.8.8.8 is the configuration management address for HDLC-TCMS and cannot be used by users.

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

5.4 Serial to UDP

5.4.1 Function description

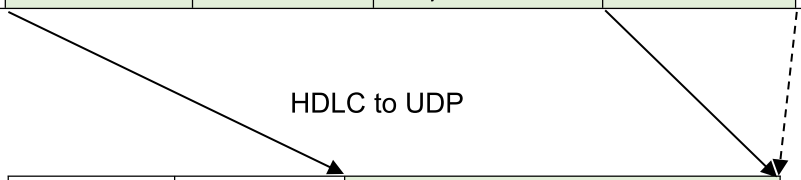
HDLC-TCMS receives HDLC frames through synchronous serial ports, receives UART strings through asynchronous serial ports and subpackages them, receives CAN frames through CAN ports and packages them together, converts the above data frames or packets into UDP messages, and sends the messages to computers or servers through Ethernet interface according to configuration.



5.4.2 Protocol Conversion

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

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

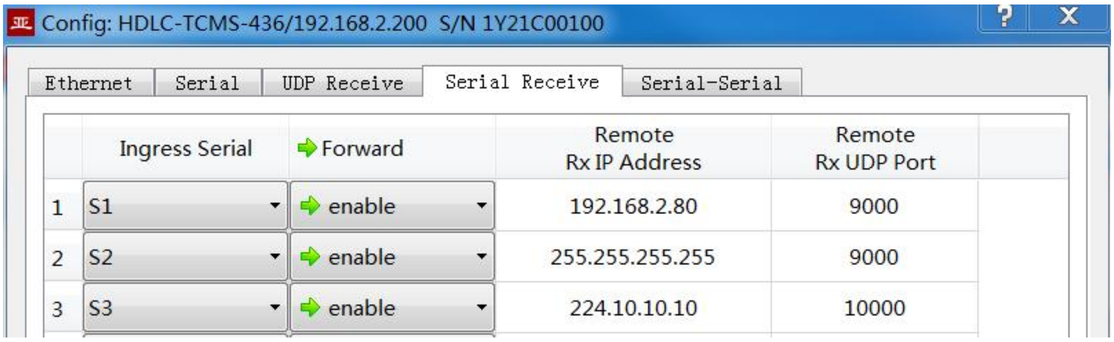


IP Header	UDP Header	Application Data

5.4.3 Forward 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:

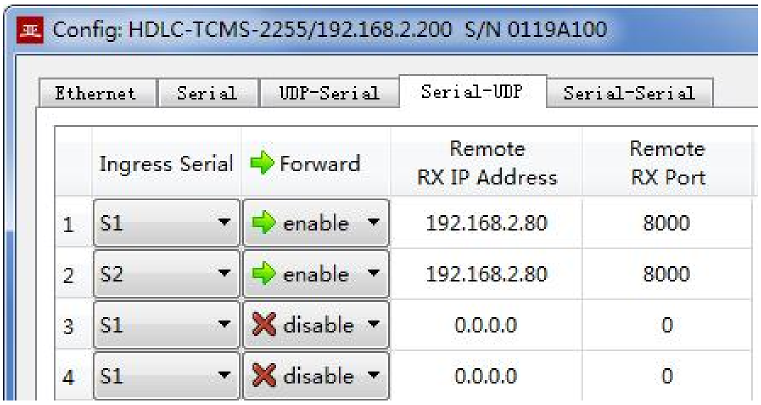
- Serial port S1 to UDP unicast, destination IP address 192.168.2.80 and destination UDP port 9000;
- Serial port S2 to UDP broadcast, all hosts in the network can receive data from S2 at port 9000;
- Serial port S3 to UDP multicast, only computers joined 224.10.10.10 groups in the network can receive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

5.4.4 How UDP Server identifies Source Serial ports

In converged 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.

5.4.4.1 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.

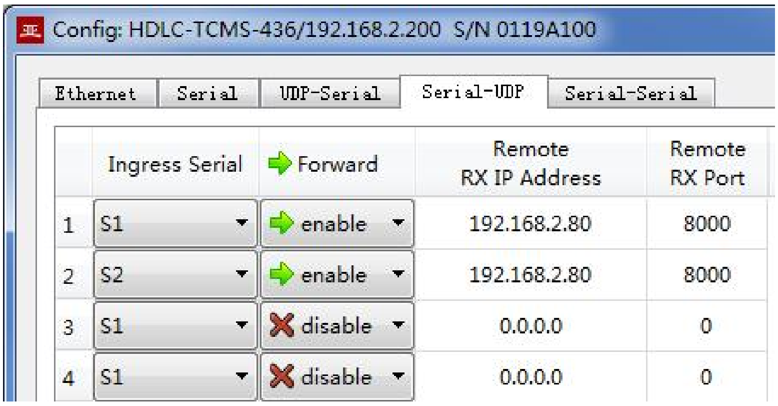


5.4.4.2 Identify source serial port based on source UDP port

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

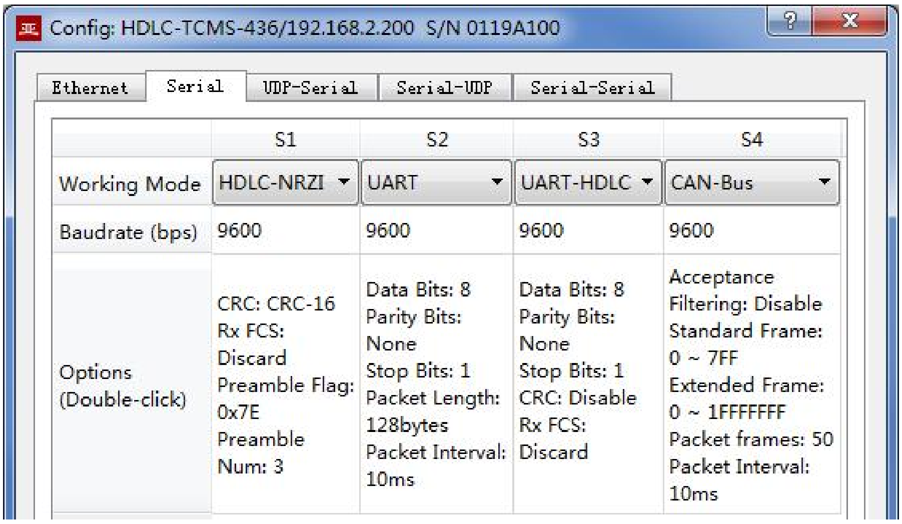


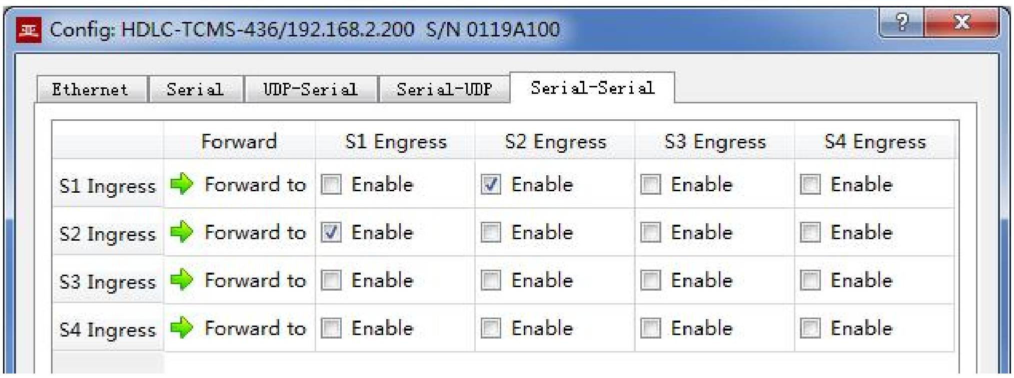
5.5 Serial to Serial

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;
- Conversion between Serial Port and CAN Bus.

Serial S1 works in synchronous HDLC mode and S2 works in asynchronous UART mode as shown below. The “serial-serial” is configured to forwarding each other between S1 and S2 to realize data conversion between synchronous and asynchronous serial ports.



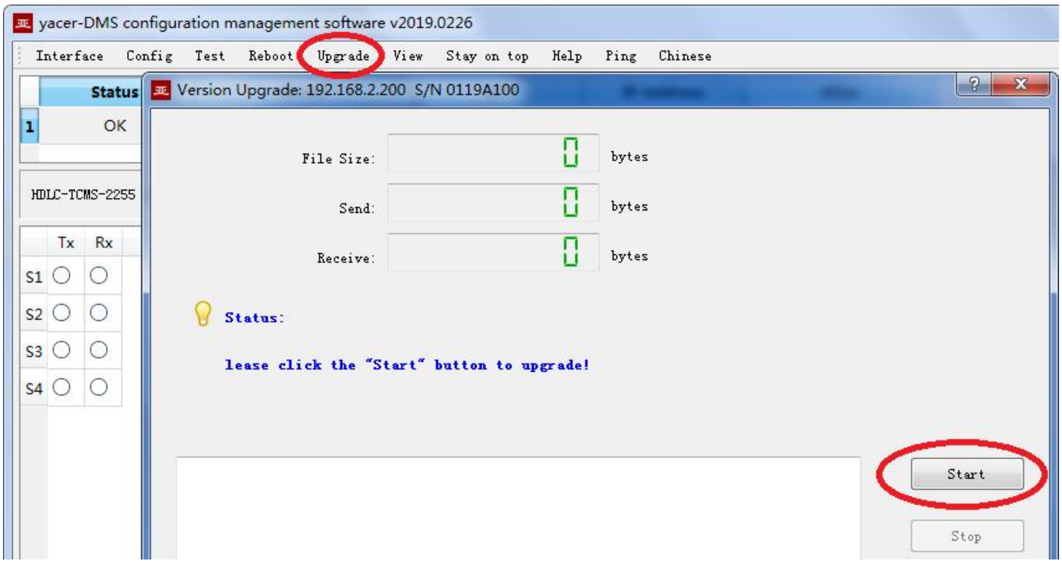


6 System Maintenance

6.1 Firmware Version Upgrade

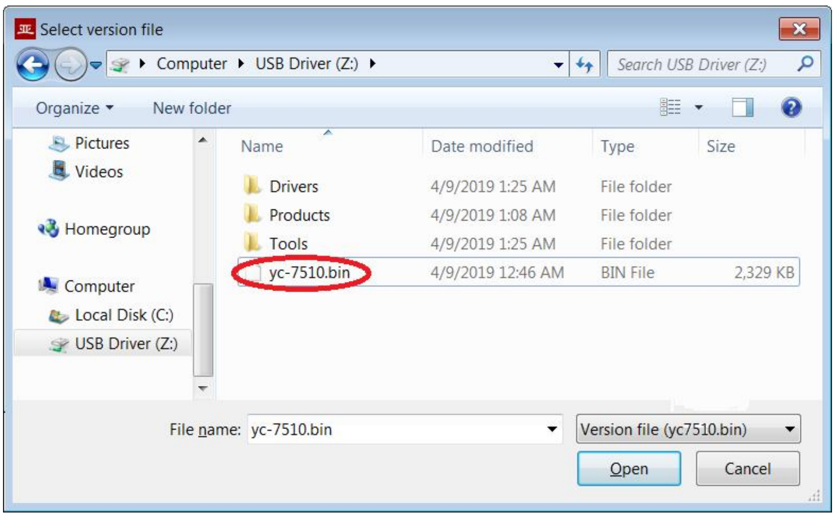
6.1.1 Start Upgrade

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



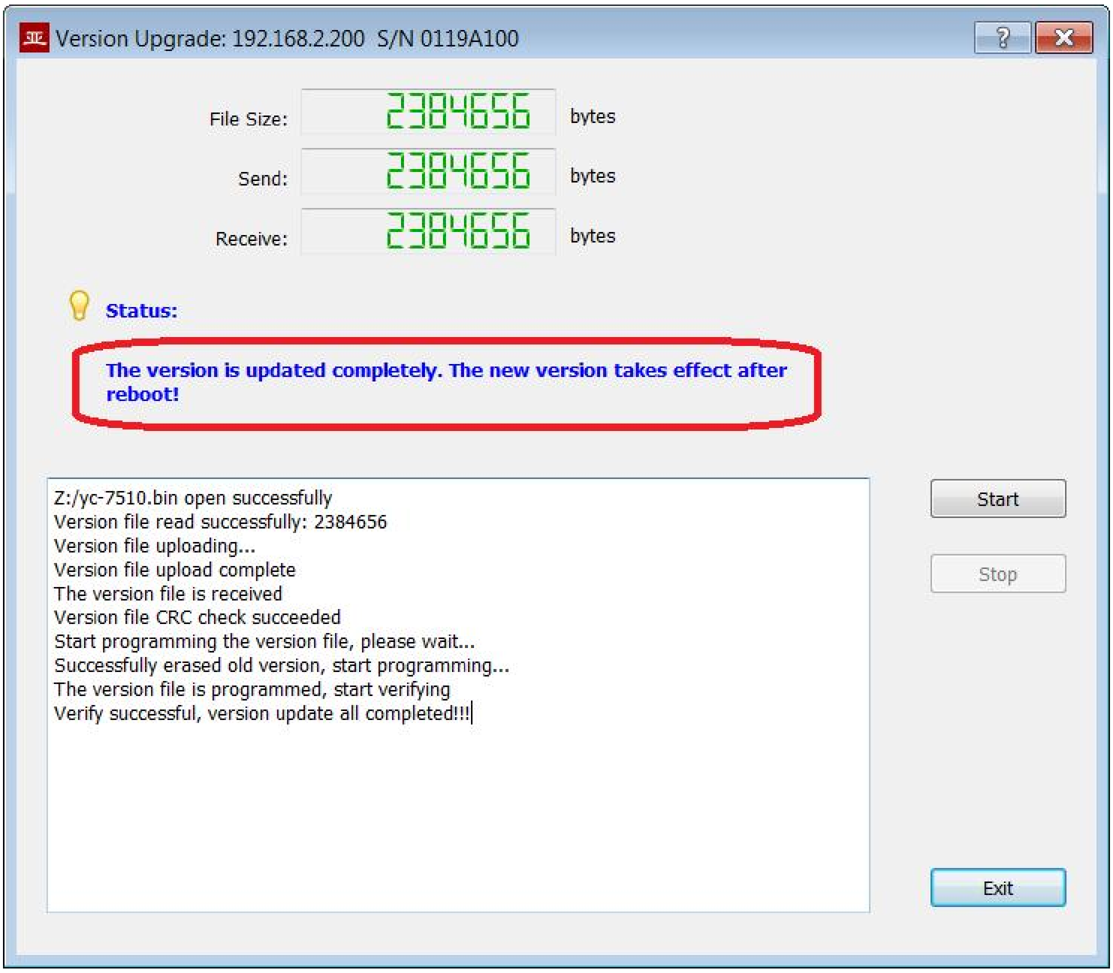
6.1.2 Select Version File

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.



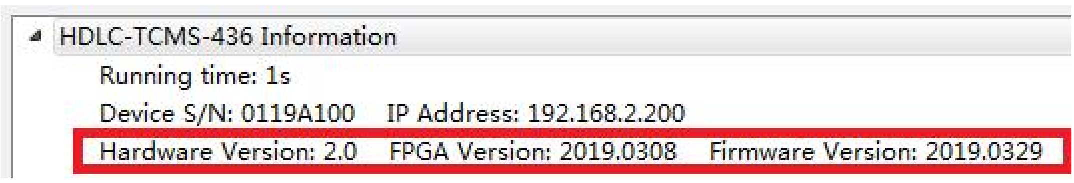
6.1.3 Complete Upgrade

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



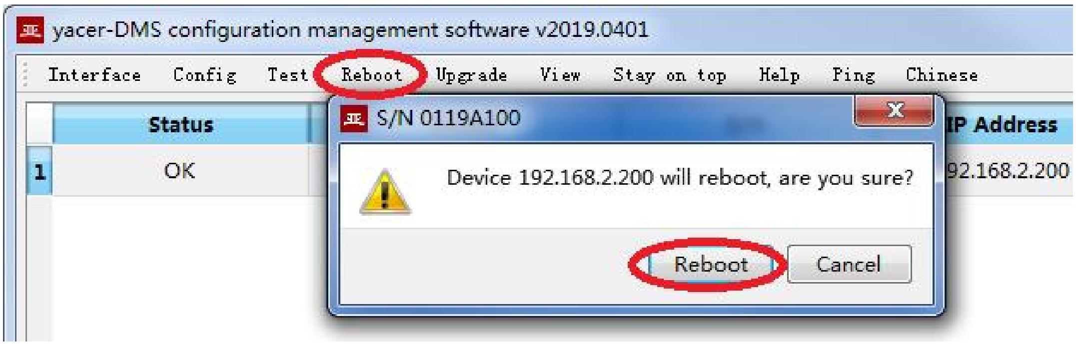
6.1.4 Confirm Upgrade

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.



6.2 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.



7 Mechanical Data

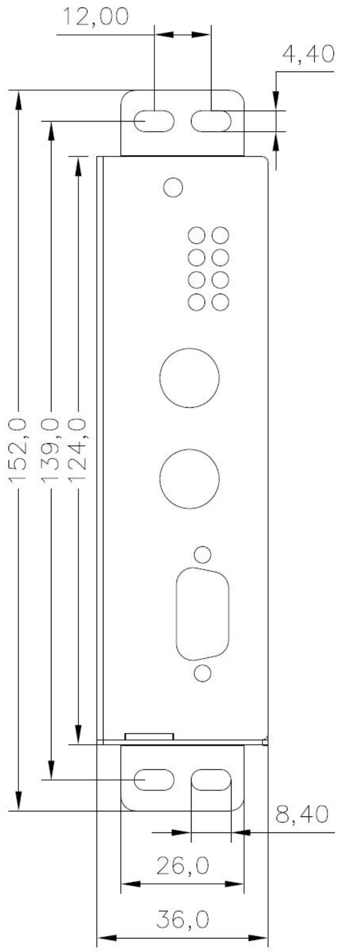
7.1 Dimensions & Mounting

The material of the housing shell is stainless steel. The thickness of the sheet metal is 1mm.

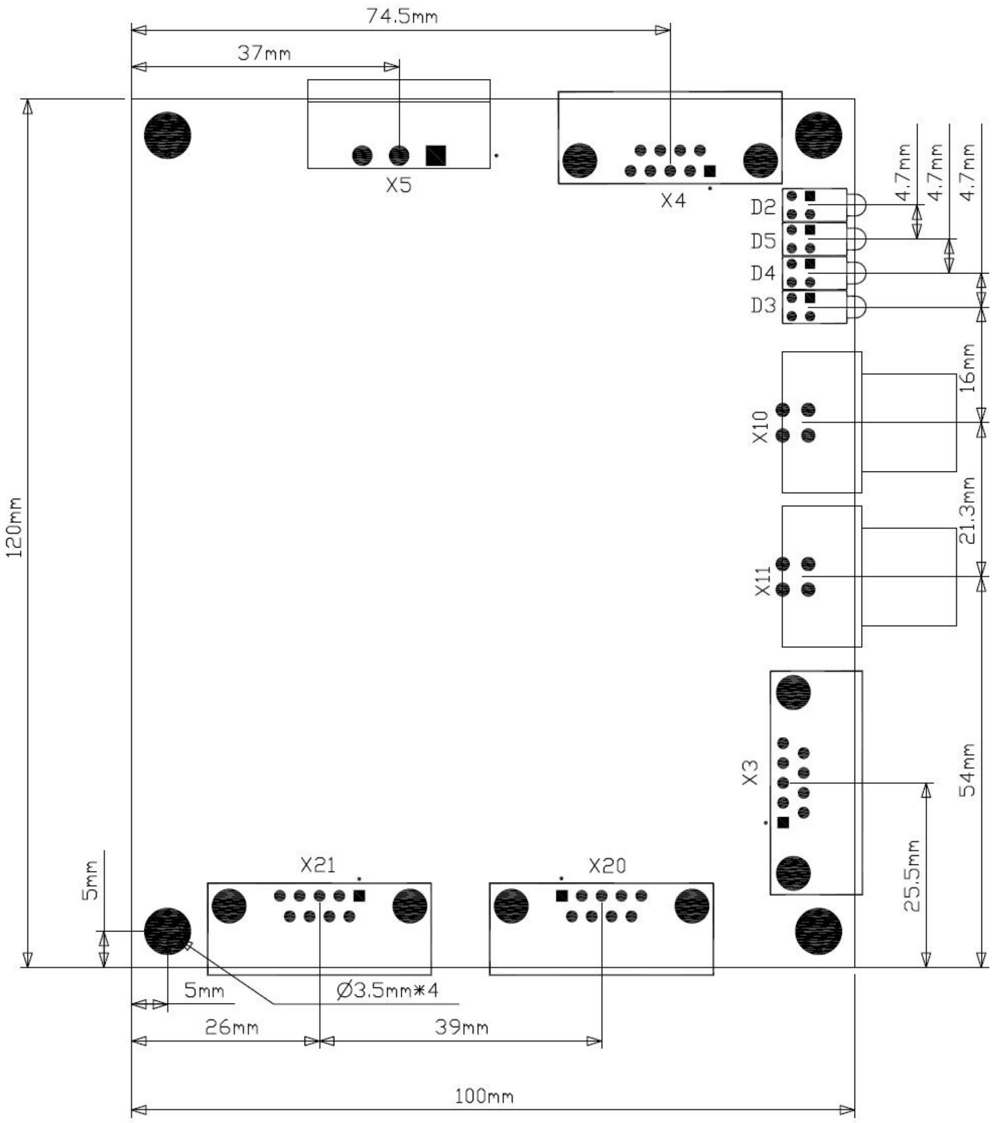
Dimensions (Height x Width x depth) = 124 x 36 x 104mm.

HDLC-TCMS can be mounted directly using four M4 Screws.

Mounting hole diameter = 4.4mm.



7.2 Board Size



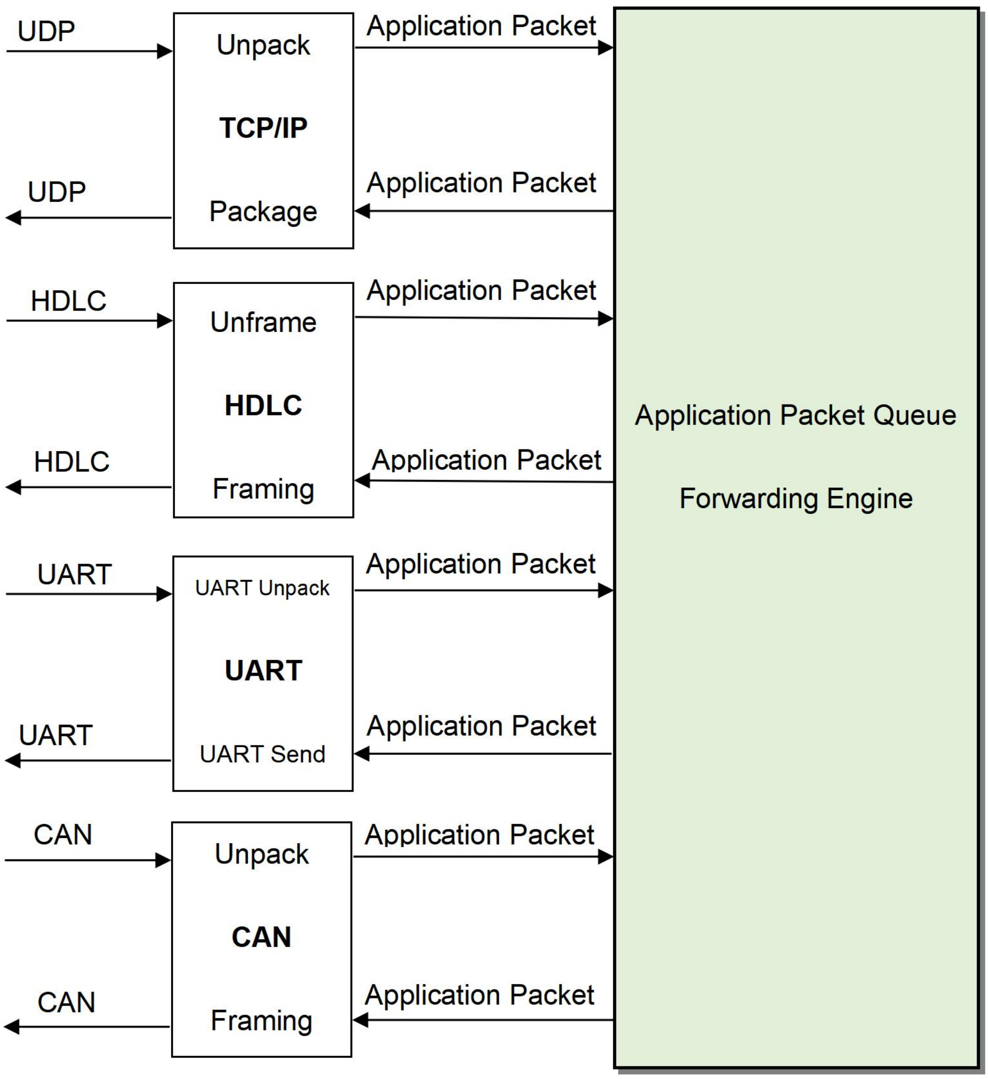
8 Application of Protocol Conversion

8.1 Serial Port Data Conversion

8.1.1 Application Packet and Conversion Model

Serial data conversion includes:

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

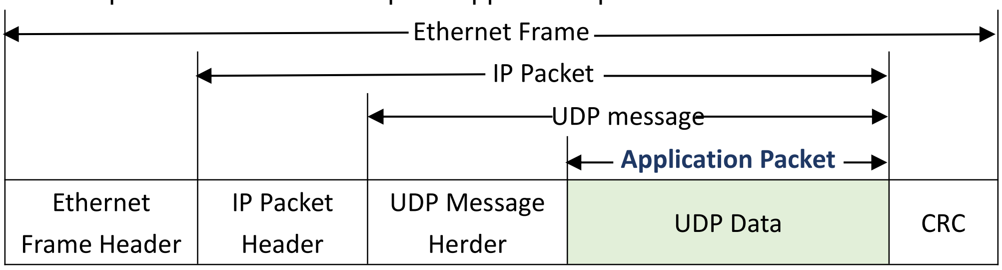


At the time of receiving, the receiving processing modules of different types of interfaces perform unpacking or decoding operations on the data, extract the application data packets, and send them to the queue of the system.

The forwarding engine of HDLC-TCMS 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.

8.1.2 UDP message format

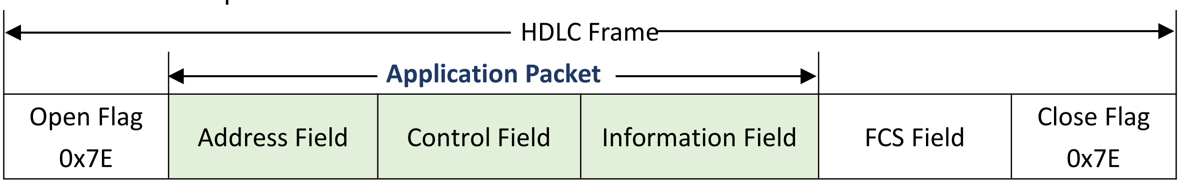
In the UDP protocol, the application packet is packaged in the data area of the UDP message. Each UDP packet contains a complete application packet.



8.1.3 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.

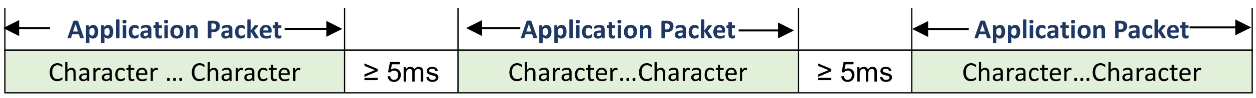
For HDLC-TCMS, instead of distinguishing between address field, control field, and information field, they are uniformly presented as application packets to the upper application to fill in and process the UART packet format



8.1.4 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-TCMS 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.

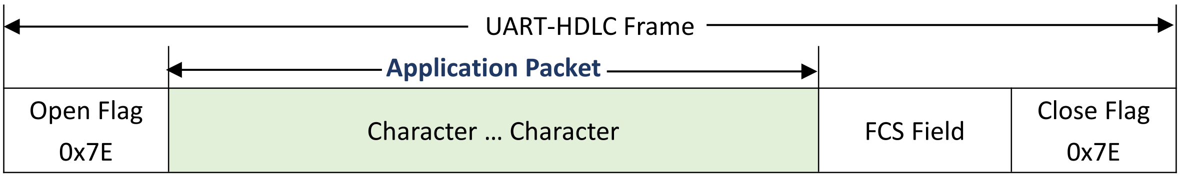


In the actual application, data transmission is not allowed during the packet interval; otherwise, it may result in a waste of communication bandwidth, and the higher the baud rate is, the more serious the waste is.

8.1.5 UART-HDLC frame format

The UART-HDLC 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-HDLC 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

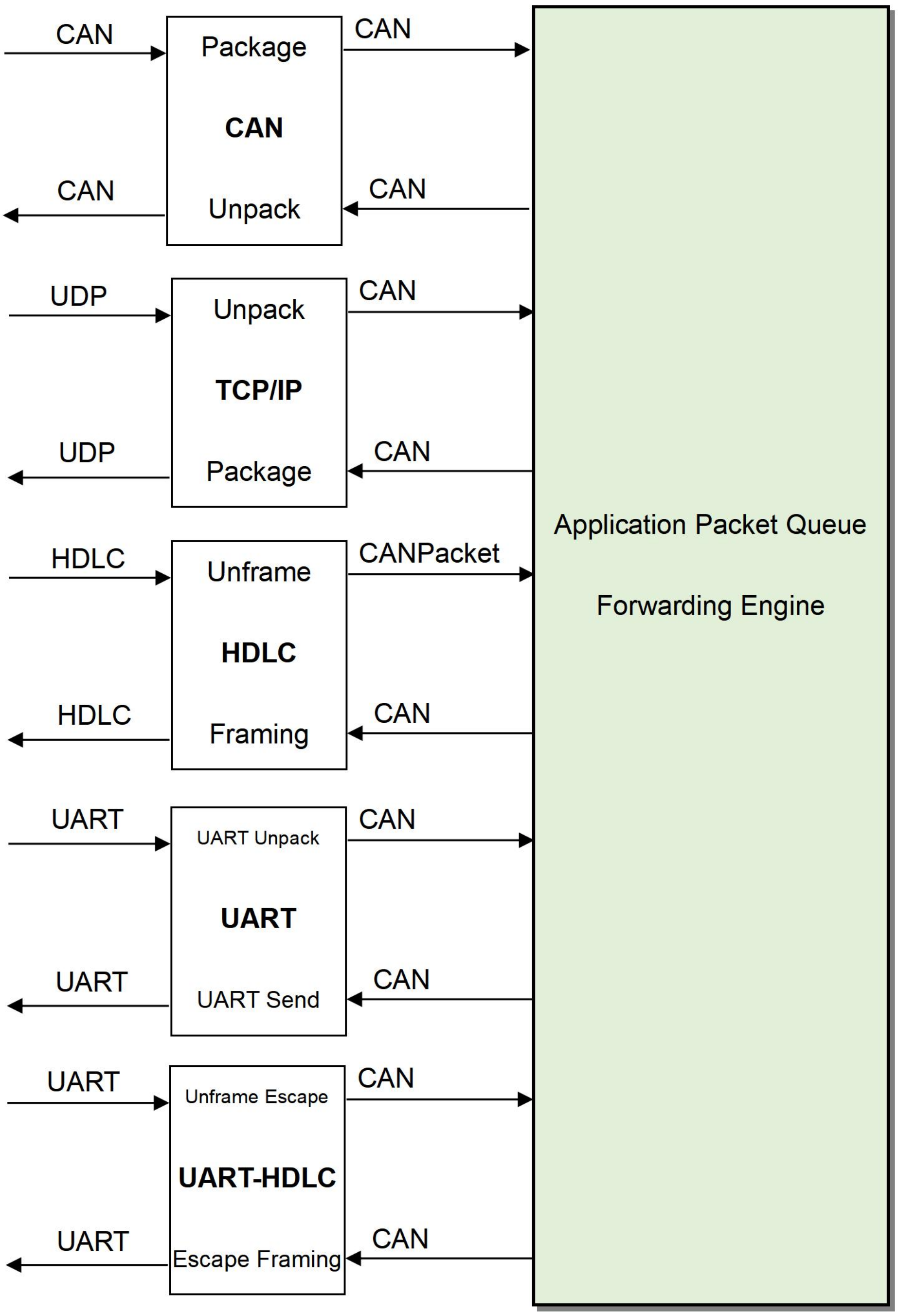
The escape operation of data receive is as follows:

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

8.2 CAN frame data conversion

8.2.1 CAN packet and conversion model

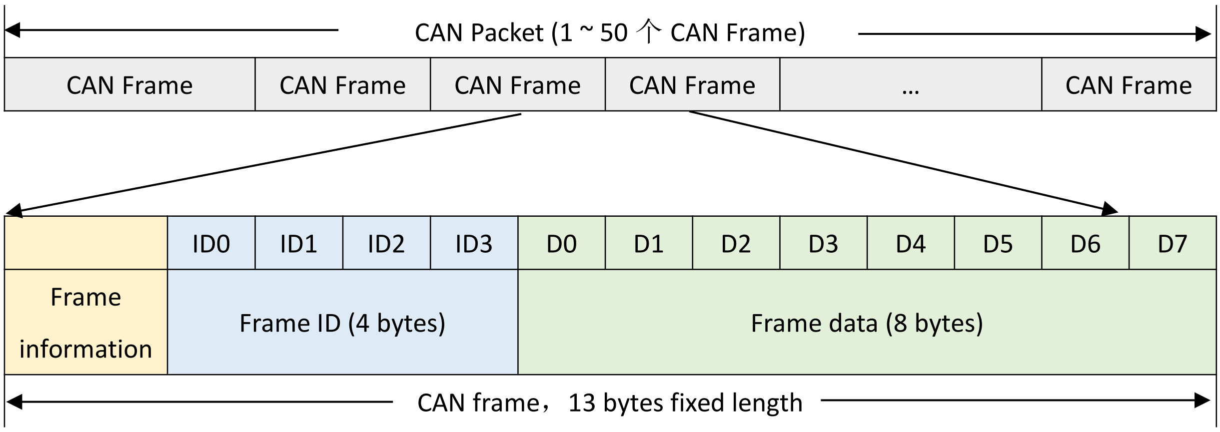
Since the CAN frame is very short, multiple CAN data frames are formed into a CAN packet to improve the forwarding efficiency, with each CAN packet corresponding to an application packet of the other interface.



8.2.2 CAN packet format

8.2.2.1 Packet format

CAN packet consists of 1 ~ 50 CAN frames with the fixed length of each CAN frame as 13 bytes.



8.2.2.2 Frame information

The frame information is 1-byte long, with the format defined as follows:

- FF: Identification of the standard frame and the extended frame, 1 for the extended frame, 0 for the standard frame;
- RTR: Identification of the remote frame and the data frame, 1 for the remote frame, 0 for the data frame;
- DLC: Length of the CAN actual data.

Frame information	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	FF	RTR	Reserved	Reserved	DLC.3	DLC.2	DLC.1	DLC.0

8.2.2.3 Frame ID

The frame ID occupies 4 bytes, but the ID bit number of the standard and extended frames differs.

Standard frame ID: 11 bits for the standard frame ID, with the value range of 0x000 ~ 0x7FF, valid fill range of ID.10 ~ ID.0.

Standard Frame ID 4 Bytes		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	ID0								
	ID1								
	ID2						ID.10	ID.9	ID.8
	ID3	ID.7	ID.6	ID.5	ID.4	ID.3	ID.2	ID.1	ID.0

Extended frame ID: 29 bits for the Extended frame ID, with the value range of 0x00000000 ~ 0x1FFFFFFF, valid fill range of ID.28 ~ ID.0.

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Standard Frame ID 4 Bytes	ID0				ID.28	ID.27	ID.26	ID.25	ID.24
	ID1	ID.23	ID.22	ID.21	ID.20	ID.19	ID.18	ID.17	ID.16
	ID2	ID.15	ID.14	ID.13	ID.12	ID.11	ID.10	ID.9	ID.8
	ID3	ID.7	ID.6	ID.5	ID.4	ID.3	ID.2	ID.1	ID.0

8.2.2.4 Frame data

The frame data occupies 8-byte space with the effective data length of 0 ~ 8 bytes; the first byte is the starting byte of the valid data, and the effective length is determined by the DLC value in the frame information.

Frame Data 8 Bytes	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

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.