

HDLC-USB

便携式协议转换器

Rev.2024.0914

# 用户手册

**yacer** 亚册  
让通信如搭积木般简单

## 目录

1 概述 .....	1
1.1 简介 .....	1
1.2 特点 .....	1
1.3 应用 .....	1
1.4 订购选型 .....	1
1.5 技术规格 .....	2
1.6 机械尺寸 .....	3
2 硬件接口 .....	4
2.1 外观 .....	4
2.2 LED 指示灯 .....	4
2.3 串口 .....	5
2.4 以太网口 .....	5
2.5 电源接口 .....	6
2.5.1 USB 电源接口 .....	6
2.5.2 DC: 电源接口 .....	6
3 搭建配置环境 .....	6
4 yacer-DMS 配置管理软件 .....	7
4.1 获取配置管理软件 yacer-DMS .....	7
4.2 运行 yacer-DMS 软件 .....	7
4.3 软件主界面 .....	7
4.4 统计报告 .....	8
4.4.1 控制面板 .....	8
4.4.2 收发指示面板 .....	8
4.4.3 信息显示面板 .....	8
4.5 配置设备 .....	9
5 功能及配置 .....	10
5.1 以太网接口配置 .....	10
5.1.1 设备别名 .....	10
5.1.2 IP 地址及默认网关 .....	10
5.2 串口配置 .....	11
5.2.1 选择串口工作模式 .....	11
5.2.2 波特率 .....	12
5.2.3 同步串口编码格式 .....	12
5.2.4 HDLC-NRZ 参数配置 .....	13
5.2.5 HDLC-NRZI 工作模式参数配置 .....	16
5.2.6 异步 UART 参数配置 .....	16
5.2.7 异步 HDLC 参数配置 .....	17
5.3 UDP 转串口 .....	18
5.3.1 应用模型 .....	18
5.3.2 协议转换 .....	18

5.3.3 参数配置 .....	19
5.3.4 接收组播 .....	19
5.4 串口转 UDP .....	20
5.4.1 应用模型 .....	20
5.4.2 协议转换 .....	20
5.4.3 参数配置 .....	21
5.4.4 UDP Server 如何识别源串口 .....	22
5.5 串口转串口 .....	23
6 系统维护 .....	24
6.1 固件版本更新 .....	24
6.1.1 开始更新 .....	24
6.1.2 找到固件版本 .....	24
6.1.3 更新完成 .....	25
6.1.4 更新确定 .....	25
6.2 复位设备 .....	26
6.3 Ping .....	26
7 协议转换应用 .....	27
7.1 应用数据包与转换模型 .....	27
7.2 UDP 报文格式 .....	28
7.3 HDLC 帧格式 .....	28
7.4 UART 数据包格式 .....	28
7.5 UART-HDLC 帧格式 .....	29

## 1 概述

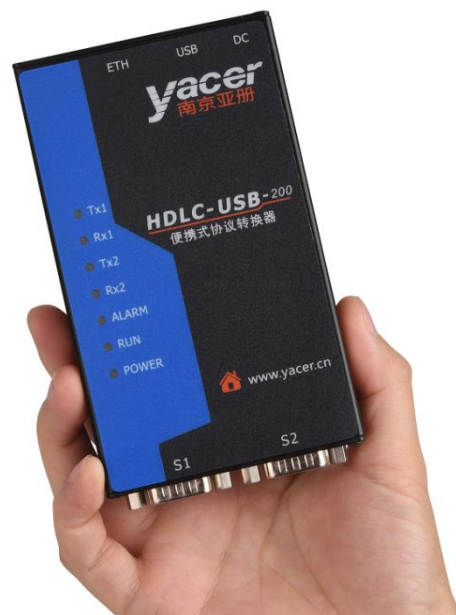
### 1.1 简介

亚册 HDLC-USB 协议转换器，提供 2 路多协议同异步串口、1 路 100M 以太网口，支持同步、异步串口互转，实现串口、以太网口之间的协议转换。

超小、超轻、超薄，支持 USB 供电。尤其适用于便携、嵌入式应用。

### 1.2 特点

- 1 路 100M 以太网接口
- 2 路多协议同异步串口
- 支持 RS-232、RS-422/485 全双工、RS-485 半双工
- 支持 HDLC、UART 等工作模式
- 支持 USB 供电
- 工业级宽温
- 轻薄小巧，便于携带



### 1.3 应用

- 同步串口、异步串口互相转换
- 串口、以太网之间的协议转换
- 列车机车通信网络（TCN）、TCMS 系统
- 空管航管监视系统，ADS-B、二次雷达（SSR）信号引接与输出
- 空管自动化系统（ATC）、空中交通管理（ATM）
- 空管雷达数据记录与回放
- 测控数据采集
- 便携式应用

### 1.4 订购选型

产品型号	串口	以太网接口
HDLC-USB-200	2 x RS-232/422/485 同异步串口	1 x 100M

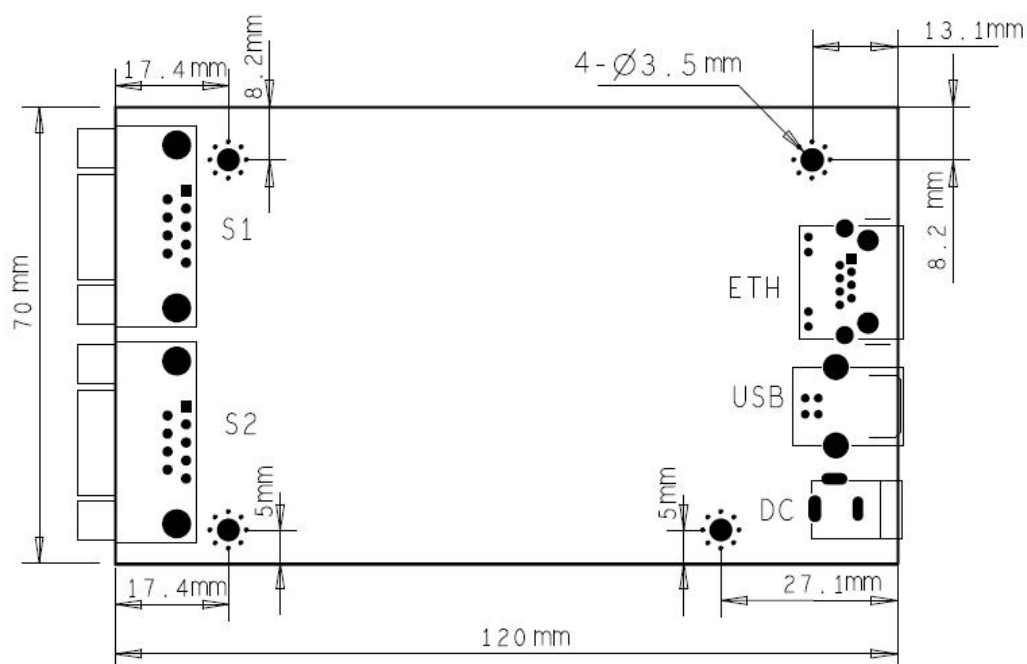
## 1.5 技术规格

<b>串口</b>	
数量	2 x DB9 (针型)
工作模式	同步 HDLC、TCMS-HDLC、异步 UART
接口类型	RS-232、RS-422、RS-485, 软件可设置
编码格式	NRZ、NRZI
波特率	NRZ ≤ 250 Kbps NRZI ≤ 64 Kbps
HDLC 帧长	≤ 1470 字节
同步时钟	常规、主时钟、从时钟 (外时钟)
ESD 保护	± 15 KV
<b>以太网口</b>	
数量	1 x RJ45
速率	100 Mbps, 支持 MDI/MDIX 自适应
协议	TCP/IP 协议
编程接口	UDP Server、UDP Client, 支持单播/组播/广播
<b>USB 接口</b>	
接口	1 x B 型 USB 接口 (母口), 仅供电
<b>配置管理</b>	
配置工具	yacer-DMS 配置管理软件
配置接口	以太网口
<b>电源需求</b>	
输入电压	+5V DC
功耗	< 2W
电源接口	USB 接口供电 DC 5.5x2.1mm 接口
<b>机械特性</b>	
尺寸	高 x 宽 x 深: 21.5 mm x 73 mm x 123 mm
重量	200g
外壳	便携式铝合金全封闭外壳
<b>工作环境</b>	
工作温度	-40 ~ +70°C
存储温度	-40 ~ +85°C
工作湿度	5 ~ 95% RH (无凝结)

## 1.6 机械尺寸



单板尺寸如下：



## 2 硬件接口

### 2.1 外观

产品一端为 S1、S2 两个串口，另一端为以太网接口（ETH）、USB 接口和 DC 电源接口。正面为丝印及 LED 指示。



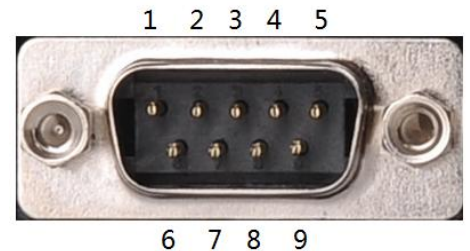
### 2.2 LED 指示灯

LED	描述
Tx1	串口 S1 发送指示灯，成功发送一帧数据时闪烁
Rx1	串口 S1 接收指示灯，成功接收一帧数据时闪烁
Tx2	串口 S2 发送指示灯，成功发送一帧数据时闪烁
Rx2	串口 S2 接收指示灯，成功接收一帧数据时闪烁
ALARM	告警指示灯，设备启动未就绪或故障时灯亮，正常运行时常灭
RUN	运行指示灯，正常运行时绿灯闪烁
POWER	电源指示灯，上电后常亮

## 2.3 串口

HDLC-USB 提供 2 路 DB9 针式串行接口，支持 RS-232/422/485 协议，支持同步、异步工作模式。

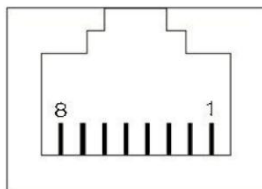
DB-9 针型	RS-232	RS -422 RS-485 全双工	RS-485 半双工
1	RxDat	RxDat+	
2	RxClock	RxClock+	
3	GND	GND	GND
4	TxClock	TxClock+	Clock+
5	TxDat	TxDat+	Data+
6		RxDat-	
7		RxClock-	
8		TxClock-	Clock-
9		TxDat-	Data-



## 2.4 以太网口

RJ-45 接口，100M 以太网，支持 MDI/MDIX 自适应。

RJ-45	信号
1	Tx+
2	Tx-
3	Rx+
6	Rx-





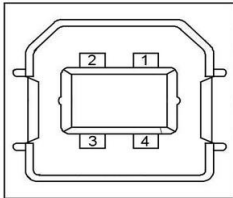
## 2.5 电源接口

+5VDC，可以取自 USB 电源接口或 DC 电源接口，二者不能同时供电。

### 2.5.1 USB 电源接口

B 型 USB 接口，仅供电源使用。

引脚序号	信号	描述
1	VBUS	电源+5V
2	D-	数据-, 未用
3	D+	数据+, 未用
4	GND	
Shell	Shield	



### 2.5.2 DC: 电源接口

HDLC-USB 使用+5V 直流电源供电。

接口采用通用的 DC 5.5-2.1mm 电源插座。



## 3 搭建配置环境

通过网线连接管理计算机与 HDLC-USB 的网口，在计算机上运行 yacer-DMS 配置管理软件，即可对 HDLC-USB 进行运行状态监控和参数配置。



## 4 yacer-DMS 配置管理软件

### 4.1 获取配置管理软件 yacer-DMS

用户可通过以下方式获取配置管理软件的压缩包 yacer-DMS.zip:

- HDLC-USB 随机 U 盘的软件工具目录;
- 亚册官网 <http://www.yacer.cn> 的软件频道。

### 4.2 运行 yacer-DMS 软件

yacer-DMS 为免安装应用软件, 对 yacer-DMS.zip 进行解压缩, 进入工作目录双击 yacer-DMS.exe 即可运行。

### 4.3 软件主界面

下图即为配置管理软件的主界面, 该界面可以分为三个部分。

- 工具条: 功能操作按钮;
- 设备列表: 显示在线设备基本信息及运行状态;
- 统计报告: 显示指定设备的收发指示、设备详细信息、收发统计。



## 4.4 统计报告

统计报告由三个面板组成：控制面板、收发指示面板、信息显示面板。

### 4.4.1 控制面板

HDLC-USB-200 统计报告    报告刷新周期:  秒    刷新    清零

控制部件	功能
报告刷新周期: <input style="width: 40px;" type="text" value="1"/> 秒	设置报告刷新周期
<span style="border: 1px solid #ccc; padding: 2px 10px;">刷新</span>	手动刷新统计报告
<span style="border: 1px solid #ccc; padding: 2px 10px;">清零</span>	统计报告清零

### 4.4.2 收发指示面板

- 发：对应串口每发一帧数据发指示灯闪烁一次。
- 收：对应串口每收一帧数据收指示灯闪烁一次。



### 4.4.3 信息显示面板

显示以下内容：

- 设备信息：运行时间、序列号、IP 地址、版本号。
- 串口：所有串口的工作时钟频率、以及数据收发统计。
- UDP 发送：对每个使能的串口转 UDP 条目，显示相关的 UDP Client 发送包数。
- UDP 接收：对每个使能的 UDP 转串口条目，显示相关的 UDP Server 接收包数。
- DMS 服务：显示设备与管理计算机之间的配置管理消息收发统计。

## 4.5 配置设备

点击工具条上的“设备配置”按钮，或双击设备列表中的选中设备，yacer-DMS 弹出配置对话框。

对话框按照接口和功能，把配置项分为多个配置页面。



在对话框最下端包括以下操作按钮：

按钮	功能
导入	打开配置文件，读取配置参数刷新配置对话框
导出	将配置对话框中配置参数导出到文件中进行保存
获取默认配置	用设备出厂的默认配置刷新对话框内容
应用配置并重启设备	将对话框中的配置参数写入设备，并重新设备使配置生效
取消	取消当前配置操作

## 5 功能及配置

### 5.1 以太网接口配置



#### 5.1.1 设备别名

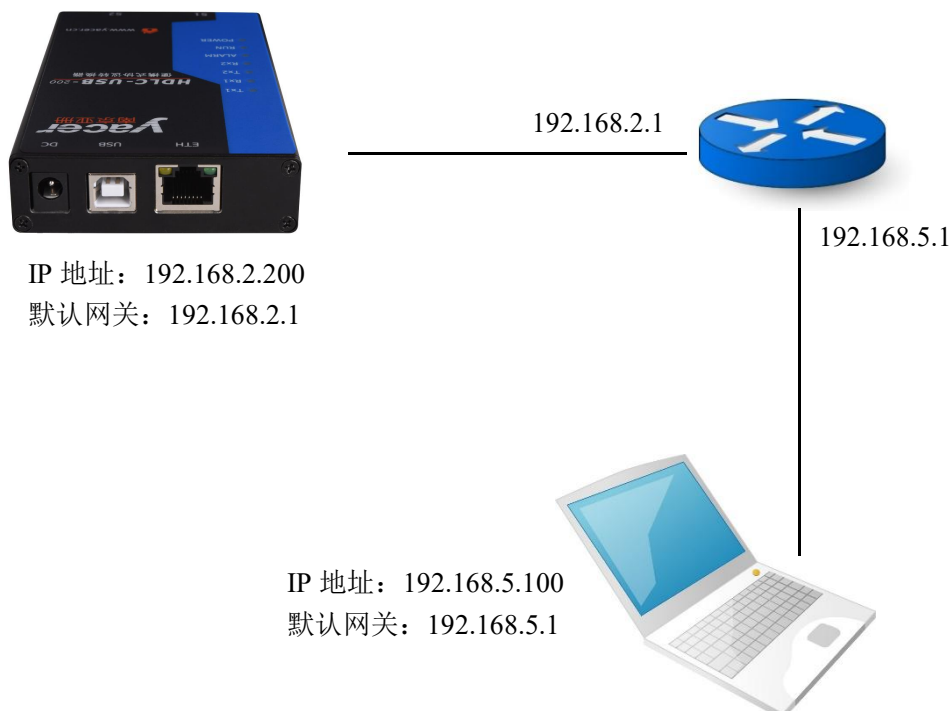
允许用户为 HDLC-USB 设置别名，从而为设备增加描述或助记标识。

#### 5.1.2 IP 地址及默认网关

IP 地址及子网掩码配置如上图所示，默认网关缺省为 0.0.0.0，表示无网关配置。

如果 HDLC-USB 需要与其他网段的主机通信，必须要借助于外部路由器才行。此时 HDLC-USB 的 IP 地址，必须和相联路由器端口的 IP 地址在同一网段，同时把该路由器 IP 地址设置为默认网关。

如下图所示，HDLC-USB 的 IP 地址为 192.168.2.200，远端计算机的 IP 地址为 192.168.5.100，由于不属于同一网段，二者必须借助于路由器才能够相互通信。HDLC-USB、计算机都需要把相联路由器端口的 IP 地址，设置为本设备的默认网关。



## 5.2 串口配置

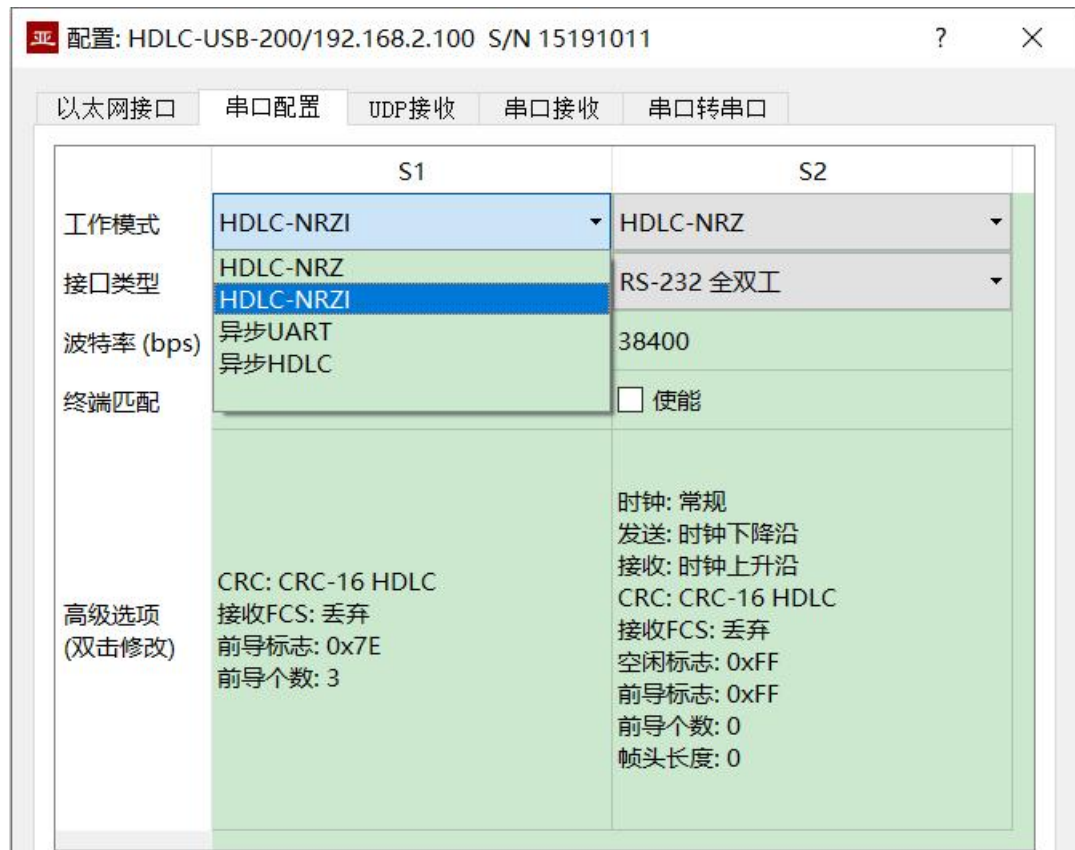
### 5.2.1 选择串口工作模式

串口 S1、S2 为同异步串口，支持同步、异步工作模式。

工作模式		描述
同步	HDLC-NRZ	基于 NRZ 编码的同步 HDLC 协议
	HDLC-NRZI	基于 NRZI 编码的同步 HDLC 协议
异步	异步 UART	通用异步串口工作模式，等同于普通计算机上的串口
	异步 HDLC	基于 UART 的类 HDLC 通信协议

用户可以通过“工作模式”下拉列表选择所需工作模式。由于每种工作模式的参数配置不同，确定工作模式后，“高级选项”单元格的内容自动根据工作模式进行调整。

如果需要进一步配置所选工作模式的工作参数，鼠标双击“高级选项”所在单元格，即可弹出参数配置对话框。

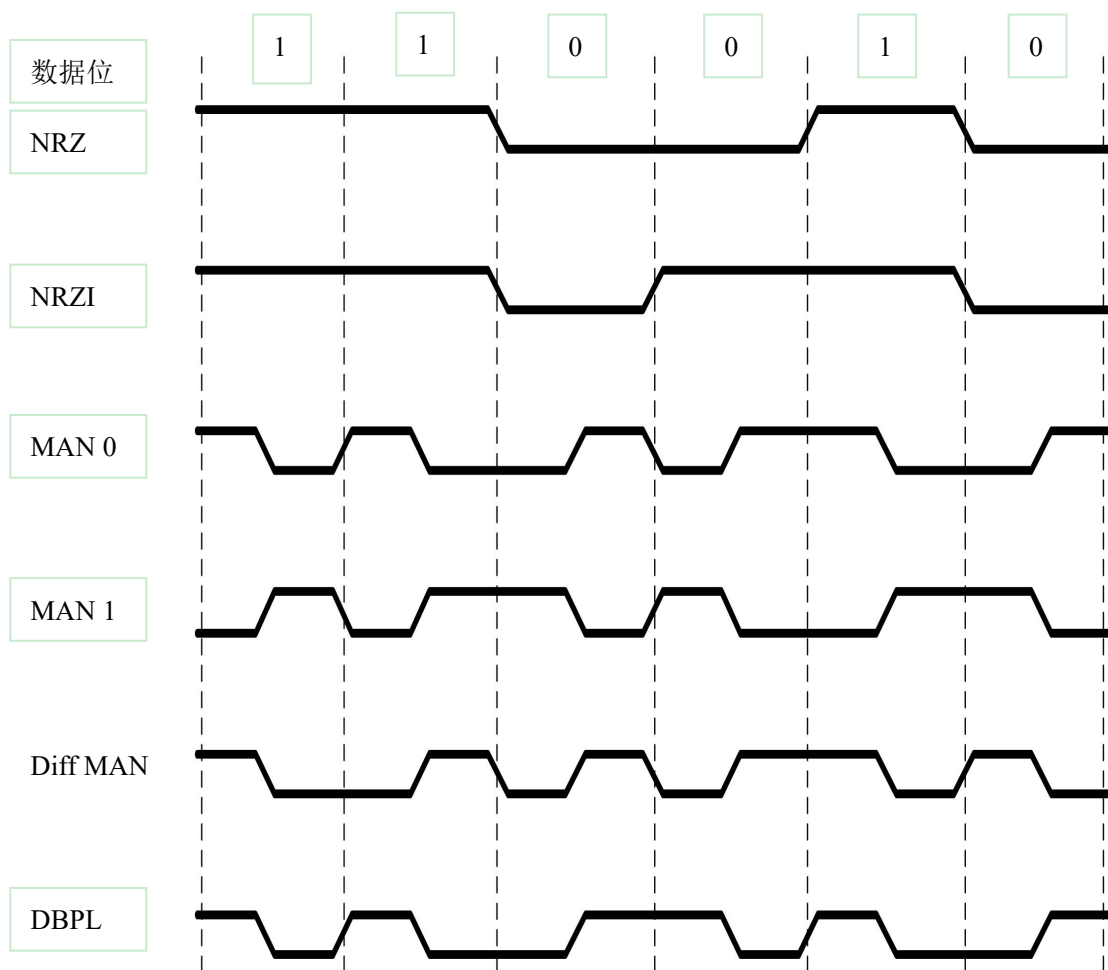


### 5.2.2 波特率

“波特率”配置了串口的通信速率，对于 HDLC-NRZI、HDLC-DBPL、HDLC-MAN、HDLC-DiffMAN 等同步工作模式，以及所有的异步工作模式来说，通信双方的波特率必须相同才能够保证数据的正确传输。

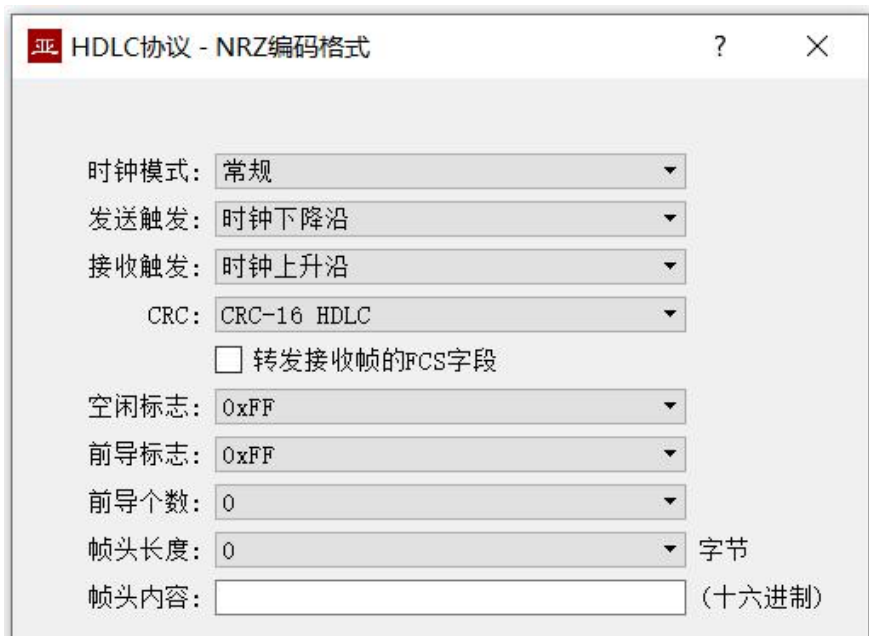
### 5.2.3 同步串口编码格式

HDLC-NRZ、HDLC-NRZI、HDLC-DBPL、HDLC-MAN、HDLC-DiffMAN 等同步工作方式，链路层采用 HDLC 协议，编码格式区别如下图所示：



### 5.2.4 HDLC-NRZ 参数配置

HDLC-NRZ 是常用的同步工作模式，多用于空管、航管领域的二次雷达、ADS-B 数据通信。



#### 5.2.4.1 时钟模式



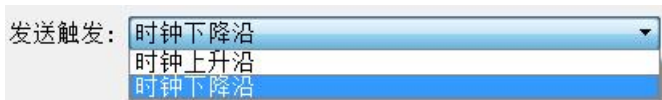
同步串口的时钟模式包括三种，常规、从时钟、主时钟。

时钟模式	发送时钟	接收时钟
常规	本设备产生，TxC 输出	对端设备产生，取自 RxC
从时钟（外时钟）	对端设备产生，取自 RxC TxC 输出自动与 RxC 同步	对端设备产生，取自 RxC
主时钟	本设备产生，TxC 输出	本设备产生，忽略 RxC 时钟

从时钟模式也称为外时钟工作模式。当对端为传输设备时，HDLC-USB 常配置为从时钟模式，用传输设备提供的时钟发送数据，确保数据在全网的传输基于一个时钟，避免了时钟不同源造成的丢包隐患。



### 5.2.4.2 发送触发

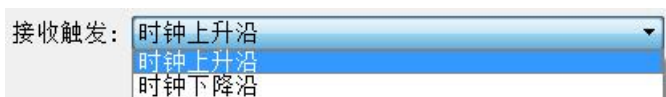


发送触发定义了新数据 bit 的产生时刻：

- 时钟下降沿：在时钟的下降沿，产生新的数据位
- 时钟上升沿：在时钟的上升沿，产生新的数据位

空管通信中，一般选择时钟下降沿触发新数据发送。在一些自主研发的应用场合，用户有时会采用上升沿触发新数据发送。

### 5.2.4.3 接收触发



接收触发定义了串口接收数据的采样时刻：

- 时钟上升沿：在 RxC 信号的上升沿，读取 RxD 线上的数据
- 时钟下降沿：在 RxC 信号的下降沿，读取 RxD 线上的数据

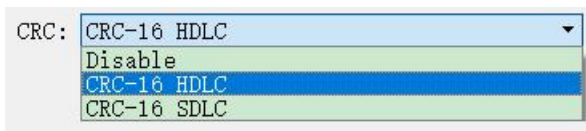
空管通信中，由于采用下降沿触发新数据，由于新数据稳定需要时间，为了保证数据的正确读取，接收触发必须配置为时钟上升沿。

接收触发配置要根据对端设备的发送触发确定：

对端发送触发	本端接收触发
时钟下降沿	时钟上升沿
时钟上升沿	时钟下降沿

### 5.2.4.4 CRC 校验

空管或列车通信中，一般选择 CRC-16 HDLC 模式，此模式基于 ISO HDLC 协议标准进行 CRC 校验。



### 5.2.4.5 转发接收帧的 FCS 字段

转发接收帧的FCS字段

该配置仅在 CRC 使能时有效。

HDLC 帧结构如下表所示，其中 FCS 为帧校验序列字段。

默认情况下，该选项不勾选，HDLC-USB 在接收 HDLC 帧校验通过后，抛弃数据末尾 2 个字节的 FCS 字段，仅转发用户数据。

如果该选项勾选，则转发用户数据+FCS 字段。

开始标志 Opening Flag	地址字段 Address Field	控制字段 Control Field	信息字段 Information Field	帧校验序列 FCS Field	结束标志 Closing Flag
0x7E	1 字节	1 字节	可变长	CRC 2 字节	0x7E
0x7E	用户数据			CRC 2 字节	0x7E

#### 5.2.4.6 空闲标志

基于 HDLC 协议，HDLC-USB 采用连续的 1 作为帧间空闲标志，默认选择为 0xFF。

空闲标志: 0xFF

#### 5.2.4.7 前导标志与前导个数

前导标志: 0x7E

前导个数: 0  
1  
2  
3  
4  
5  
6  
7

在半双工应用中，常常在帧头增加 2~5 个 0x7E 前导用于接收方同步准备。对于全双工应用，往往不需要前导，把前导个数设置为 0 即可（无前导）。

#### 5.2.4.8 帧头长度与帧头内容

帧头长度: 2 字节  
0  
1  
2  
3  
4

帧头内容: FF 03 (十六进制)

以上图为例，帧头长度定义为 2，帧头内容定义为十六进制的 FF 03：

- 在 HDLC 发送时，在用户数据前增加 FF 03，同用户数据一起组成 HDLC 帧数据。
- 在 HDLC 接收时，HDLC-USB 把 HDLC 帧数据的前 2 个字节作为帧头抛弃，仅转发后续数据给用户。

开始标志 Opening Flag	帧头	用户数据	帧校验序列 FCS Field	结束标志 Closing Flag
0x7E	0xFF 0x03	可变长	CRC 2/4 字节	0x7E

### 5.2.5 HDLC-NRZI 工作模式参数配置

与 NRZ 编码格式不同，NRZI 编码格式的数据中包含时钟信息，只需要通信双方的波特率相同，不再需要时钟模式、发送触发、接收触发等参数。

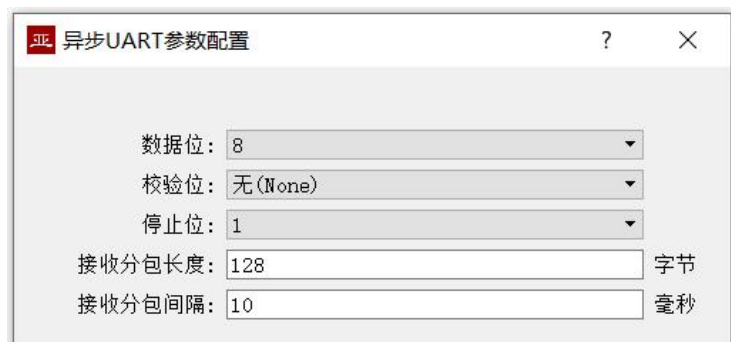
HDLC-NRZI 的配置参数见下图，参数作用及配置方式同 HDLC-NRZ 工作模式。



### 5.2.6 异步 UART 参数配置

UART 是一种字符流的通信方式，数据位、校验位、停止位定义了异步串口的基本工作参数，该参数必须和对端设备配置完全相同。

一般来说数据位定义为 8 位，即 1 个字节，这样 UART 相当于字节流的通信。



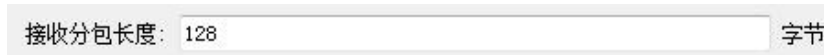
当把 UART 的字节流转换为 UDP 报文或 HDLC 帧时，如果每个字节都转换为一个 UDP 报文进行传输，开销太大、效率太低。

为了提高效率，HDLC-USB 将收到的字节流进行缓冲，然后把缓冲下来的若干字节组成一个 UDP 报文发出，这个过程成为分包。

分包由两个参数进行控制，称为分包长度、分包间隔。

#### 5.2.6.1 接收分包长度

举例来说，如果分包长度设置为 128 字节，那么当 UART 收满 128 字节后，组成一个数据包进行转发。



#### 5.2.6.2 接收分包间隔

如上例设置了分包间隔为 10ms，如果 UART 超过 10ms 未收到新的数据，则不论是否收满 128 字节，都会把缓冲区的数据组成一个数据包进行转发。

接收分包间隔: 10 毫秒

### 5.2.7 异步 HDLC 参数配置

异步 HDLC 工作模式是亚册自定义的一种通信方式，在普通 UART 通信之上对字节流进行封装，构成异步 HDLC 帧。这样异步串口就可以和 UDP 报文、同步 HDLC 帧进行基于数据包的通信。



异步 HDLC 帧格式就是把数据包前后增加 0x7E 作为起止标记，帧结构如下：

开始标志	信息字段	帧校验字段	结束标志
0x7E	2 – 1470 字节数据	2 字节 CRC 校验	0x7E

由于信息字段、帧校验字段有可能出现 0x7E，因此在传输前对信息字段、帧校验字段进行字符转义，转义规则如下：

- 0x7E：转义为 0x7D 0x5E 两个字符
- 0x7D：转义为 0x7D 0x5D 两个字符
- 其他字符：不转义

数据发送转义操作如下：

原始数据	实际发送数据
0x7E	0x7D 0x5E
0x7D	0x7D 0x5D
其他	无变化

数据接收转义操作如下：

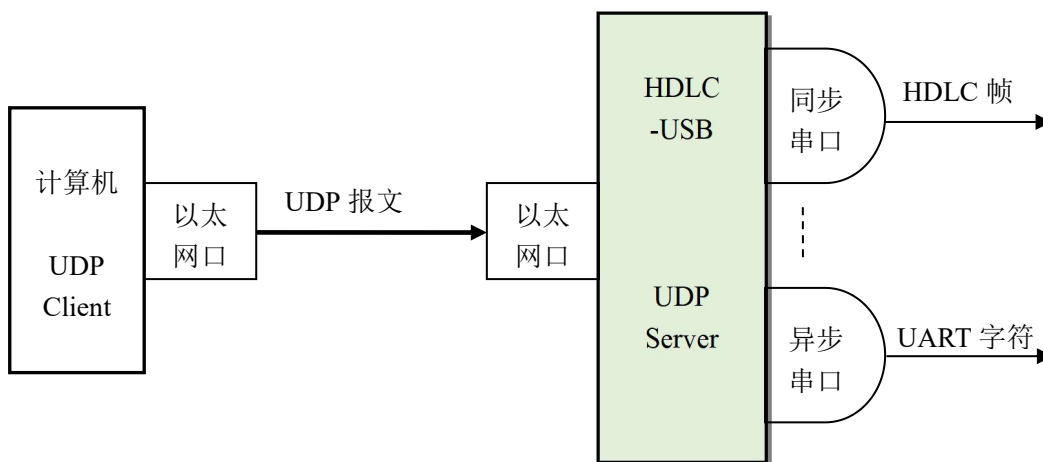
实际接收数据	数据
0x7D 0x5E	0x7E
0x7D 0x5D	0x7D
其他	无变化

### 5.3 UDP 转串口

#### 5.3.1 应用模型

借助 HDLC-USB，计算机或服务器能够实现同步 HDLC 串口发送功能。

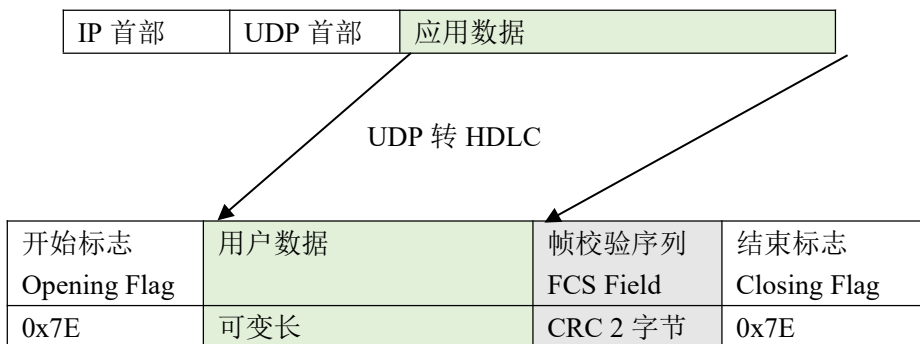
典型的应用如下图所示。计算机作为 UDP Client 通过以太网接口发送 UDP 报文，HDLC-USB 把收到的 UDP 报文进行处理，根据配置转换为 HDLC 帧、或 UART 数据，然后从串口发送出去。



#### 5.3.2 协议转换

最典型的 UDP 转 HDLC 应用如下图所示，HDLC-USB 把 UDP 的应用数据装入 HDLC 帧的用户数据区，然后计算 CRC 并填充 FCS 字段，形成一个完整的 HDLC 帧进行发送。

为了减轻计算机的计算负荷、降低用户编程的复杂性，一般情况下，UDP 报文中不包含 HDLC 的 FCS 字段，该字段由 HDLC-USB 计算填充。



### 5.3.3 参数配置

对 UDP 转串口进行设置，每一行表示一个 UDP 端口到串口的转发条目，可以实现三种转发策略：

- 转发：指定的 UDP 端口接收的数据可以转发到指定的串口。
- 汇聚：多个不同的 UDP 端口接收的数据可以转发到同一个串口。
- 分路：从同一个 UDP 端口接收的数据可以转发到不同的串口。



下面的配置实现了从一个 UDP 端口收到的数据，同时分发到 2 个串口输出的应用：



### 5.3.4 接收组播

如果需要接收组播 UDP 报文，在右侧“组播接收地址”列表添加需要加入的组播地址。

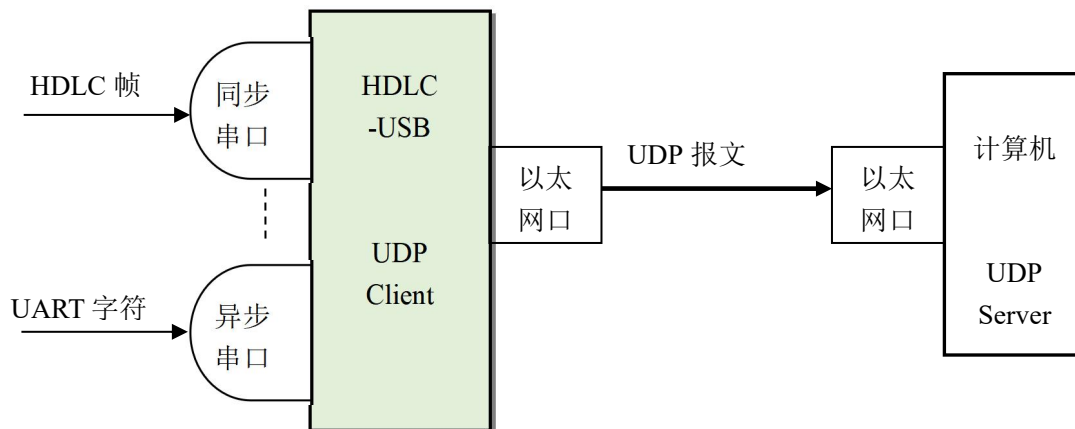
组播地址的范围为 224.0.0.0 ~ 239.255.255.255，其中 224.8.8.8 为 HDLC-USB 配置管理地址，用户不能使用该地址。

组播地址配置为 0.0.0.0 表示该条目未生效。

## 5.4 串口转 UDP

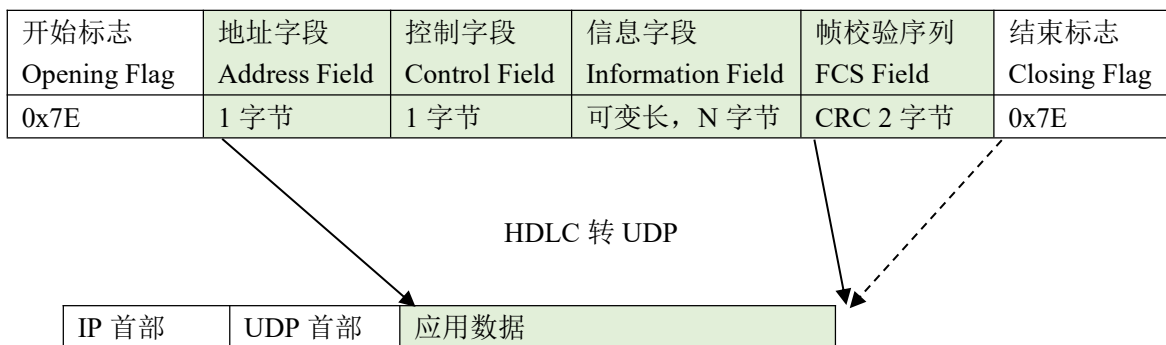
### 5.4.1 应用模型

串口转 UDP 功能示意图如下，HDLC-USB 通过串行接口，接收来自其他设备的 HDLC 帧或 UART 数据，将其转换为 UDP 报文，通过以太网发送给计算机或服务器。



### 5.4.2 协议转换

为了确保用户数据的完整性，HDLC-USB 把完整的 HDLC 帧置于 UDP 应用数据中，转发给 UDP Server。





### 5.4.3 参数配置

对串口转 UDP 进行设置，每一行表示一个串口到 UDP 的转发条目，可以实现三种转发策略：

- 转发：从特定串口接收的数据可以转发到特定的目的 IP + UDP 端口；
- 汇聚：多个不同的串口接收的数据可以转发到同一个目的 IP + UDP 端口；
- 分路：从同一个串口接收的数据可以转发到不同的目的 IP 或 UDP 端口。



如上图所示，配置了三个串口转 UDP 条目，分别实现：

- 串口 S1 转 UDP 单播，目的 IP 地址为 192.168.2.80，目的 UDP 端口为 8000
- 串口 S1 转 UDP 广播，网络中所有主机均能在 9000 端口收到来自 S2 的数据
- 串口 S2 转 UDP 组播，网络中只有加入了 224.10.10.10 组的计算机，才能够收到来自 S2 的数据。



#### 5.4.4 UDP Server 如何识别源串口

在很多应用中，例如空管自动化（ATC）应用，需要把源自多个不同串口的 HDLC 帧，转发给一台服务器或计算机，进行统一处理。

在这种情况下，需要有一种策略，让计算机能够知道所收的 UDP 报文，源自哪个串口。

##### 5.4.4.1 根据目的 UDP 端口区分源串口

如下图所示，为每个串口设置不同的转发目的 UDP 端口。UDP Server 计算机在不同的 UDP 端口接收数据，在 8001 端口所收报文来自串口 S1，8002 端口所收报文来自串口 S2。



##### 5.4.4.2 根据源 UDP 端口区分源串口

当采用目的 UDP 端口识别源串口方案的时候，UDP Server 需要在多个 UDP 端口上监听和接收数据。当串口数量很多的时候，不仅 UDP Server 的端口资源占用过多，配置及编程的复杂度也提高不少。

为了简化 UDP Server 侧的实现，我们可以采用下例的配置，把每个转换都转发至 UDP Server 的同一端口。**亚册 HDLC-USB 在转发时，会自动根据源串口调整 UDP 报文的源端口号，其中串口 S1 转发的 UDP 报文源端口为 8001、串口 S2 的为 8002。**

这样，UDP Server 仅需在一个端口（下例为 8000）监听和接收数据，然后根据源 UDP 端口区分源串口。如果存在多台 HDLC-USB，UDP Server 可以通过源 IP 区分源设备。



## 5.5 串口转串口

串口转串口能够把指定串口的输入数据，转发到其他串口输出，主要用于同步、异步串口之间的转换。

如下面所示配置，串口 S1 工作在同步 HDLC 模式，S2 工作在异步 UART 模式。其串口转串口配置为 S1、S2 之间的互相转发，则可以实现同、异步串口之间的数据转换。



## 6 系统维护

### 6.1 固件版本更新

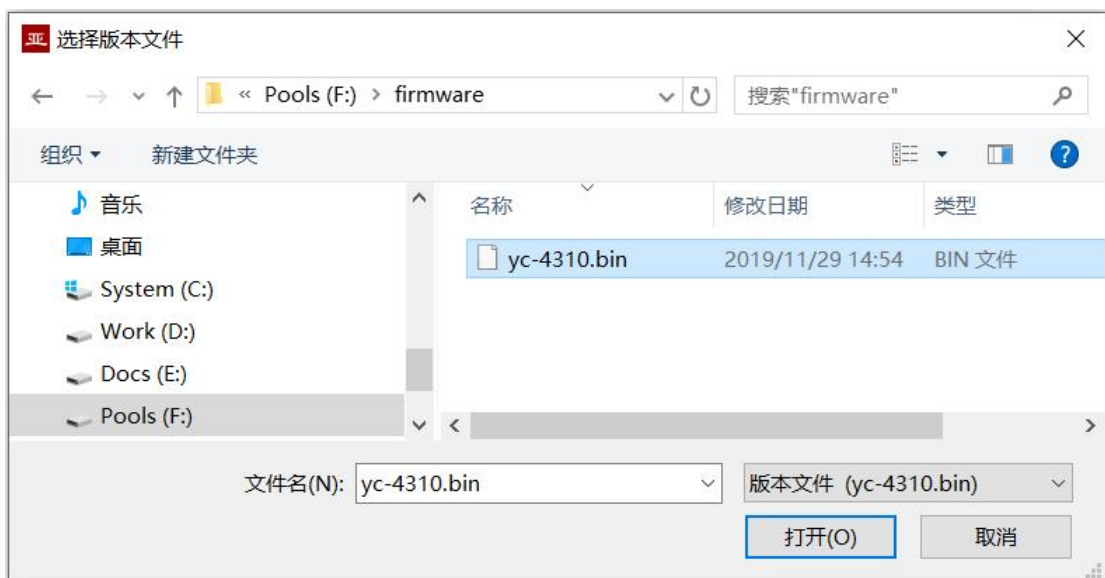
#### 6.1.1 开始更新

点击工具条上的 **版本更新** 按钮，弹出版本更新对话框，点击 **开始更新** 按钮。



#### 6.1.2 找到固件版本

弹出“选择版本文件”对话框，找到存放最新固件版本所在的文件夹，选中并点击“打开”，即开始更新。



### 6.1.3 更新完成

更新完成以后页面状态显示“版本更新完成”即表示版本更新完成。



### 6.1.4 更新确定

更新完成后，将设备重新加电，观察统计报告中的版本信息，通过版本日期确定新版本是否成功更新。



## 6.2 复位设备

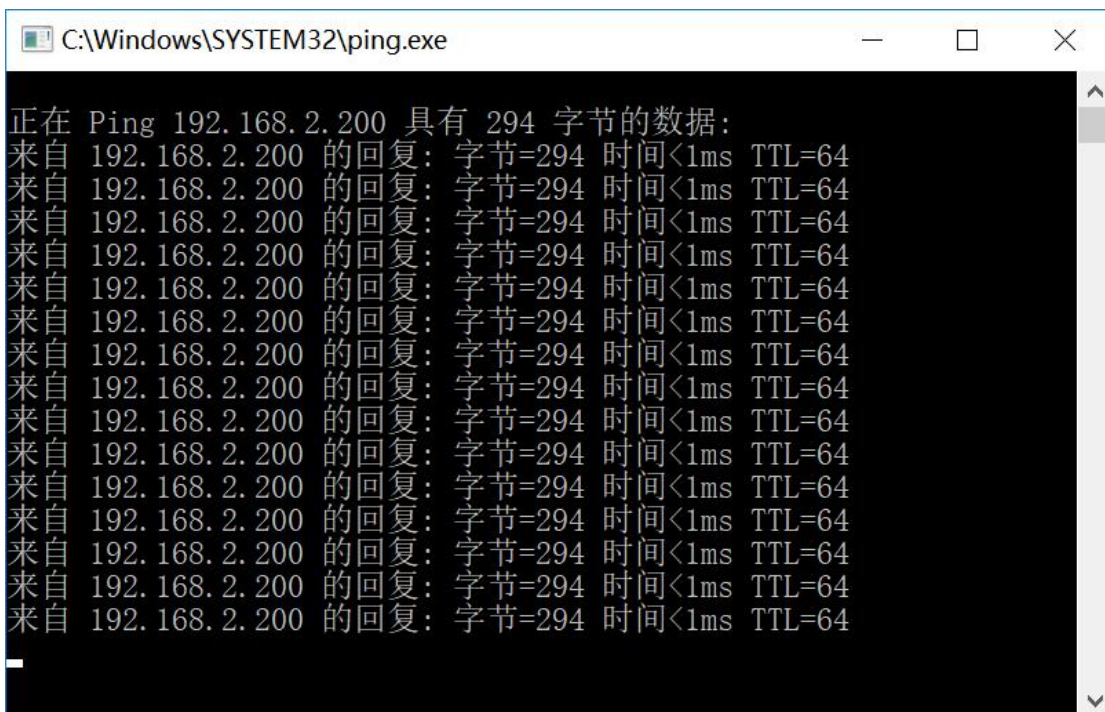
点击工具条上的“复位设备”按钮，弹出设备重启对话框，点击“重启”按钮，设备将进行重新启动。



## 6.3 Ping

点击工具条上的“Ping”按钮，DMS 自动对所选设备启动 ping 命令，以检查配置管理计算机和 HDLC-USB 之间的网络连接是否正常。

执行 Ping 命令前，请首先确保计算机和 HDLC-USB 的 IP 地址在同一网段。





## 7 协议转换应用

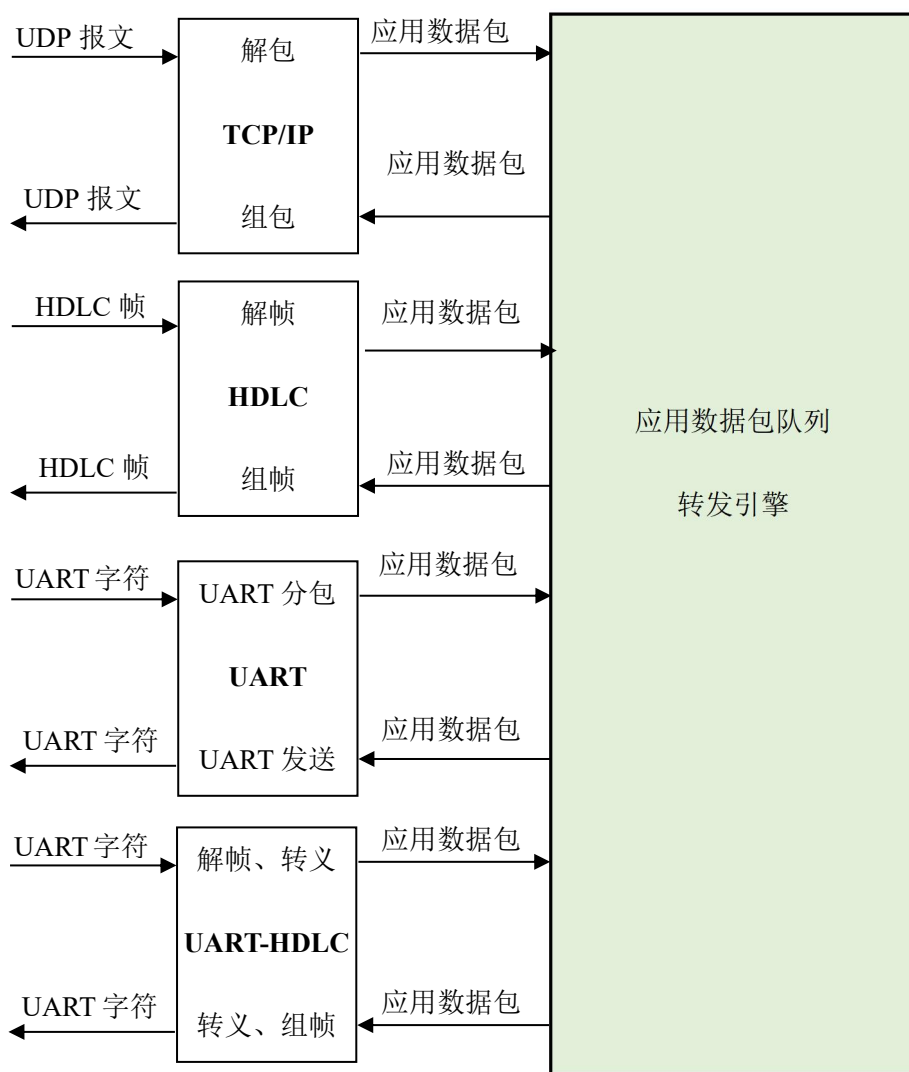
### 7.1 应用数据包与转换模型

串口数据转换包括：

- 串口与 UDP 之间的协议转换
- 同步串口、异步串口之间的数据转换

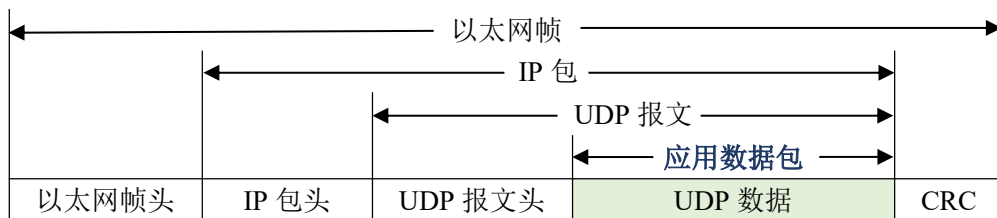
在接收时，不同类型接口的接收处理模块，对数据进行解包或解帧操作，提取应用数据包，送入系统的队列。

HDLC-USB 的转发引擎读取应用数据包队列，根据转发配置送至各接口的发送模块。发送模块负责应用数据包的组帧或组包操作，以产生不同类型的协议包或数据帧，通过物理接口发送出去。



## 7.2 UDP 报文格式

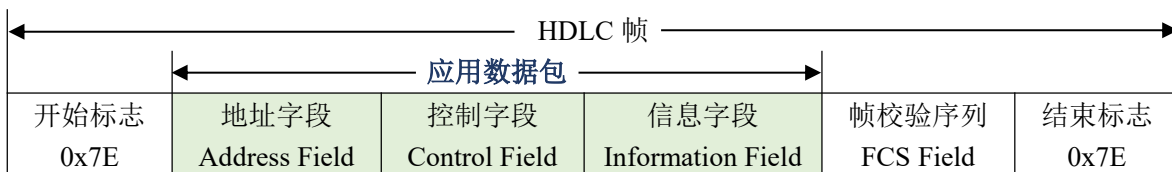
在 UDP 协议中，应用数据包封装在 UDP 报文的数据区中，每个 UDP 报文包含一个完整的应用数据包。



## 7.3 HDLC 帧格式

一个完整的 HDLC 帧由位于开始标志和结束标志之间的若干字段组成，包括地址字段、控制字段、信息字段及用于 CRC 校验的 FCS 字段。

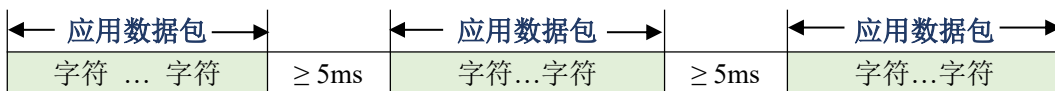
对于 HDLC-USB 来说，并不区分地址字段、控制字段和信息字段，而是把他们统一作为应用数据包交给上层应用填写和处理。



## 7.4 UART 数据包格式

当串口工作在异步 UART 模式时，从串口收到的是无头无尾的字符流，其中并没有用来进行分包或分帧的信息。

HDLC-USB 采用时间信息来分包，允许用户定义 UART 的分包时间。举例来说假如分包时间为 5ms，那么当超过 5ms 没有收到新的字符时，则认为一个数据包接收完毕。

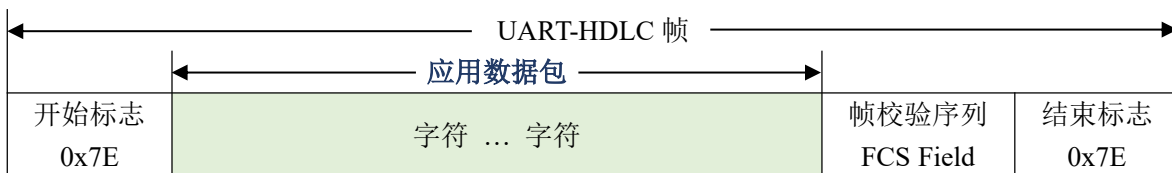


在实际应用中，包间隔期间期间不能有数据传输，导致通信带宽存在浪费，而且波特率越高浪费约严重。

## 7.5 UART-HDLC 帧格式

UART-HDLC 工作模式采用另一种策略为 UART 提供分包能力。如下图所示，数据发送方计算应用数据包的 CRC，并在头尾增加 0x7E 作为开始和结束标志，构建一个 UART-HDLC 帧。

这种策略不需要额外增加分包时间，可以充分利用通信带宽，但增加了通信双方处理的复杂度。



由于应用数据包和帧校验字段有可能出现 0x7E，因此收发双方需要对应用数据包和帧校验字段进行字符转义，转义规则如下：

- 0x7E：转义为 0x7D 0x5E 两个字符
- 0x7D：转义为 0x7D 0x5D 两个字符
- 其他字符：不转义

数据发送转义操作如下：

原始数据	实际发送数据
0x7E	0x7D 0x5E
0x7D	0x7D 0x5D
其他	无变化

数据接收转义操作如下：

实际接收数据	数据
0x7D 0x5E	0x7E
0x7D 0x5D	0x7D
其他	无变化