**智能物资柜主控板通信规约**

**拟 制： 2022年3月 24日**

**版本说明**

|  |  |
| --- | --- |
| 版本 | 说明 |
| V1.01 | 1、增加状态点表 |
| 2、增加配置点表 |
| 3、增加打开方式，人脸、指纹、条码 |
| V1.02 | 1、修改报文类型，增加盘点启动命令、盘点结束命令 |
| 2、修改物资存取过程中的通信流程 |
| V1.03 | 物资消息由原来一条物资消息传送一个EPC值改为一条物资消息传送多个EPC值 |
| V1.04 | 为方便分帧、提高报文解析效率，报头由0x68修改为0x68 0x68. |
| V1.05 |  |

**1、简介**

* 1. **目的**

本文档对主控板与工控机的数据通信接口进行了规范，主控板与工控机的通信设计需按该协议进行开发。

* 1. **定义**

RFID ：Radio Frequency Identification，射频识别

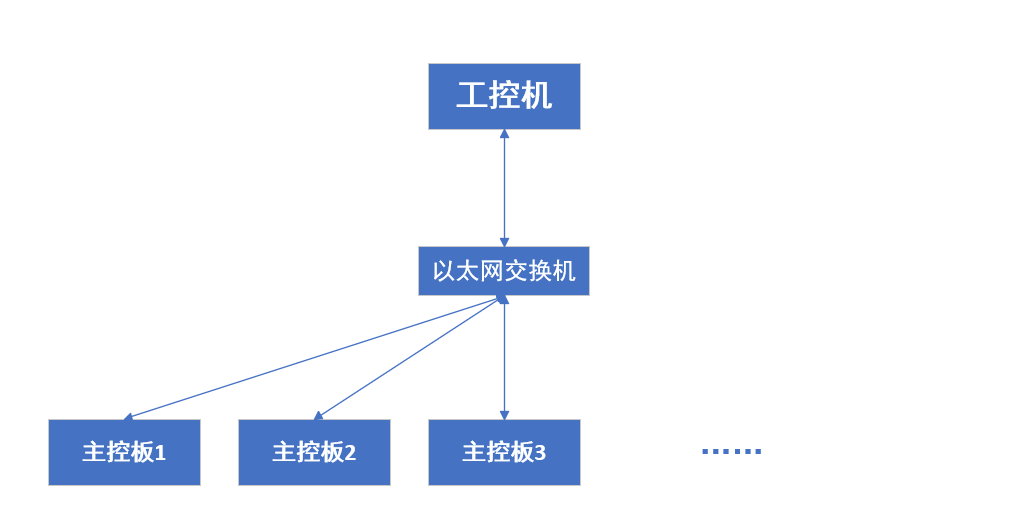
APDU：Application Protocol Data Unit，应用协议数据单元

SBAP：Smart Box Application Protocol，智能柜应用协议

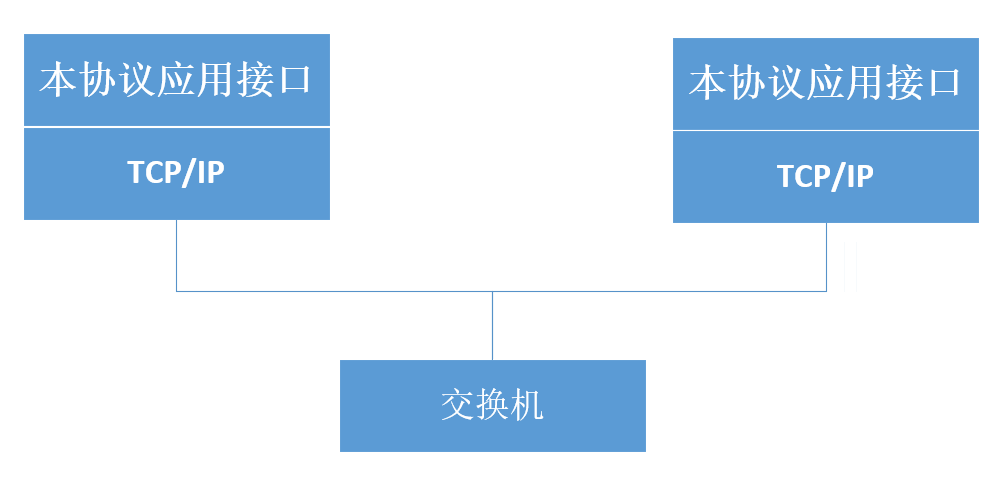
PDU ：Protocol Data Unit，协议数据单元

FID ：Fingerprint identification，指纹标识

* 1. **系统框架**



* 1. **协议的体系结构**



本协议是基于TCP/IP协议上的应用层协议， 其中主控板作为客户端、工控机作为服务端。

1. **应用报文与数据结构**

**2.1 本协议的APDU**

本协议传输接口（TCP到用户）是一个定向流接口，为了明确报文启停定义了SBAP报文头。APDU长度为PDU数据的长度，占两个字节，PDU为协议数据。



APDU格式如下：



PDU格式说明：

地址域 ：占1个字节，范围为0x01-0x3F，该地址为主控板拨码开关的硬件地址

报文类型：占1个字节，其内容如下：

|  |
| --- |
| < 1 > 握手命令  < 2 > 解出握手命令  < 3 > 授权消息  < 4 > 启动盘点  < 5 > 结束盘点  < 6 > 物资消息  < 7> 状态消息  < 8 > 读配置命令  < 9 > 写配置命令  < A > 开锁命令  < B > 单次盘点结束  < C > 物资校验  < D > 停止盘点  < 0，14~255 > 预留 |

控制位域：Bit1~Bit7为预留位；Bit0表报文流向，数据由主控板到工控机该值为0，数据由工控机到中空板该值为1。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 预留 | | | | | | | 方向 |

数据域：

数据采用高位在前的格式，具体如下：



校验位域：

该值为PDU的校验和，采用CRC16校验方法，具体校验算法参照附录1。

**2.2 报文帧结构**

**2.2.1 握手命令**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | 5（APDU数据长度） | 5（APDU数据长度） |
| 4 | 地址 | 地址 |
| 5 | 1（报文类型） | 1（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7 | 数据域 | 数据域 |
| 8~9 | CRC校验码 | CRC校验码 |

数据域为握手发起方，0表示发起方为主控板，1表示发起方为工控机

例：

上行 ： 68 68 00 06 01 01 00 00 18 50

下行 ： 68 68 00 06 01 01 01 00 88 51

**2.2.2 解除握手命令**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | 5（APDU数据长度） | 5（APDU数据长度） |
| 4 | 地址 | 地址 |
| 5 | 2（报文类型） | 2（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7 | 数据域 | 数据域 |
| 8~9 | CRC校验码 | CRC校验码 |

数据域为解除握手发起方，0表示发起方为主控板，1表示发起方为工控机

例：

上行 ： 68 68 00 06 01 02 00 00 18 A0

下行 ： 68 68 00 06 01 02 01 00 88 A1

**2.2.3 授权消息**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 3（报文类型） | 3（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

下行数据域仅包含授权类型

上行数据域格式：



格式说明：  
 身份类型：占1个字节，上传的授权消息为IC卡时该值为0，上传的授权消息为指纹时该值为1，上传的授权消息为条码该值为2。

授权消息长度：授权消息类型加上授权消息的数据长度。

授权消息 ：即用户身份消息（IC卡号、用户指纹FID、条码）。

示例：

指纹消息

指纹特征值为： AC

上行： AC E1 CC

下行： 68 68 00 06 01 03 01 01 88 31

IC卡消息

IC卡值为： 0x49A0192E

上行： 68 68 00 0A 01 03 00 00 49 A0 19 2E 35 2F

下行： 68 68 00 06 01 03 01 00 48 F0

条码消息

条码值： 12345

上行： 68 68 00 09 01 03 00 02 31 32 33 34 35 33 13

下行： 68 68 00 06 01 03 01 02 89 71

**2.2.4 开始盘点**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 4（报文类型） | 4（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

下行数据域格式：



天线端口：该区域在盘点指令为1、2的时候有效，占4个字节，bit0表示使能天线1、bit1表示使能天线2、bit2表示使能天线3......bit31表示使能天线32。

连续读/单次读： 1读写器连续读取，0读写器单次读取。

示例：

下行： 68 68 00 0B 01 04 01 00 00 00 01 00 46 C4

上行： 68 68 00 06 01 04 00 00 19 40

**2.2.5 结束盘点**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 5（报文类型） | 5（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 无数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

下行数据域格式：

****

执行结果：1个字节，0表示盘点成功，1表示盘点失败

物资数量：2个字节，表示柜内物资数量

示例：

下行：68 68 00 07 01 05 01 00 28 D2 48

上行：68 68 00 05 01 05 00 50 23

**2.2.6 物资消息**

|  |  |
| --- | --- |
| 字节位置 | 上行 |
| 0 ~ 1 | 0x68 0x68 |
| 2~3 | APDU数据长度 |
| 4 | 地址 |
| 5 | 6（报文类型） |
| 6 | 0（报文控制字） |
| 7~ | 数据域 |
|  | CRC校验码 |

该消息不用回复无下行数据域

上行数据域格式：

****

标签数量 ：1个字节，该值为1 ~ 50。

示例：

EPC值1：E2 80 68 94 00 00 50 15 E2 11 F1 82

EPC值2：E2 80 68 94 00 00 50 15 E2 12 9D 82

上行：68 68 00 1E 01 06 00 02 E2 80 68 94 00 00 50 15 E2 11 F1 82 E2 80 68 94 00 00 50 15 E2 12 9D 82 B9 94

**2.2.7 状态消息**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 7（报文类型） | 7（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 无数据域 |
|  | CRC校验码 | CRC校验码 |

上行数据域格式：



格式说明：

温度值 ： 占2个字节，该值为整型数，扩大十倍上送。

湿度值 ： 占2个字节，该值为整型数，扩大十倍上送。

温湿度传感器状态：占1个字节，0表正常，1表异常。

加热除湿机状态 ：占1个字节，根据加热除湿机上送状态值定义。

RFID读卡器状态 ：占1个字节，0表示空闲、1表示忙碌、2表示故障。

电子锁状态 ：占2个字节，共16bit，bit0表示锁编号1状态、bit2表示锁编号2的状态......bit9表示锁编号10的状态、bit11-bit15为0

门状态 ：占2个字节，高字节表主门状态，低字节表副门状态

示例：

温度值：25.2℃

湿度值：45.0%

门状态 ：副门打开、主门打开

加热除湿机状态：0

RFID读卡器状态：空闲 0

电子锁状态：正常 0

示例：

上行：68 68 00 0F 01 07 00 00 FC 01 C2 01 01 00 00 00 00 B4 BB

下行：68 68 00 05 01 07 01 F0 E3

**2.2.8 读配置命令**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 8（报文类型） | 8（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

配置点表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 内容 | 数据类型 | 备注 |
| 通信参数 | Local IP | U8\*4 | 默认192.168.6.2 |
| Local port | U16 | 默认2400 |
| Local mask | U8\*4 | 默认255.255.255.0 |
| Local gateway | U8\*4 | 默认192.169.5.1 |
| Remote IP | U8\*4 | 默认192.168.6.1 |
| Remote port | U16 | 默认2400 |
| 系统参数 | Temperature | U16 | 温度定值 |
| Humidity | U16 | 湿度定值 |
| Sensor select | U8 | 温湿度数据选择0：无，1：温湿度传感器，2：加热除湿机 |
| Smoke sensor | U8 | 烟感使能位 |

上行数据域仅包含配置地址，占1个字节，0表示通信参数，1表示系统参数

下行数据域格式：



格式说明：

配置地址：占一个字节，该值为0表示通讯参数配置，该值为1表示系统参数

配置数据：配置内容。

示例：

读通信参数

本地IP ：192.168.6.2 本地端口号：2400

下行：68 68 00 06 01 08 01 00 8A 81

上行：68 68 00 1A 01 08 00 00 C0 A8 06 02 09 60 FF FF FF 00 C0 A9 05 01 C0 A8 06 01 09 60 52 6A

读系统参数

温度定值：22.3℃

湿度定值：50%

温度传感器选择：温度传感器 1

烟感使能位：不使能 0

下行：68 00 06 01 08 01 01 4A 40

上行：68 00 0C 01 08 00 01 00 DF 01 F4 01 00 8F 82

**2.2.9 写配置命令**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | 9（报文类型） | 9（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

配置点表

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 内容 | 数据类型 | 备注 |
| 通信参数 | Local IP | U8\*4 | 默认192.168.6.2 |
| Local port | U16 | 默认2400 |
| Local mask | U8\*4 | 默认255.255.255.0 |
| Local gateway | U8\*4 | 默认192.169.5.1 |
| Remote IP | U8\*4 | 默认192.168.6.1 |
| Remote port | U16 | 默认2400 |
| 系统参数 | Temperature | U16 | 温度定值 |
| Humidity | U16 | 湿度定值 |
| Sensor select | U8 | 温湿度数据选择0：无，1：温湿度传感器，2：加热除湿机 |
| Smoke sensor | U8 | 烟感使能位 |

上行数据域格式：



下行数据域格式：



格式说明：

配置地址：占1个字节该值为0表示通讯参数配置，该值为0表示通讯参数配置，该值为1表示系统参数

配置结果：占1个字节，0表配置成功，1表配置失败

示例：

配置通信参数

下行：68 68 00 1A 01 09 01 00 C0 A8 06 02 09 60 FF FF FF 00 C0 A9 05 01 C0 A8 06 01 09 60 02 77

上行：68 68 00 06 01 09 00 00 DA D1

配置系统参数

下行：68 68 00 0C 01 09 01 01 00 DF 01 F4 01 00 D3 4E

上行：68 68 00 06 01 09 00 01 1A 10

**2.2.10 开锁命令**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | A（报文类型） | A（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

上行数据域仅包含锁编号。

下行数据域：



格式说明：

锁编号：占1个字节，一个主控板可控制8个锁。

开锁结果：0表开锁成功，1表开锁失败。

示例：

下行：68 68 00 06 01 0A 01 01 8A E1

上行：68 68 00 07 01 0A 00 01 00 88 1B

**2.2.11 单次读结束/物资校验结束**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | B（报文类型） | B（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

示例：

上行：68 68 00 05 01 0B 00 30 27

下行：68 68 00 05 01 0B 01 F0 E6

**2.2.12 物资校验**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | C（报文类型） | C（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

**下行数据域：**

****

标签数量：2

EPC1:E28068940000400F4F66D171

EPC2:E28068940000500F4F66D5FC

示例：

下行：68 68 00 1E 01 0C 01 02 E2 80 68 94 00 00 40 0F 4F 66 D1 71 E2 80 68 94 00 00 50 0F 4F 66 D5 FC BB 23

上行：68 68 00 05 01 0C 00 00 25

**2.13 停止盘点**

|  |  |  |
| --- | --- | --- |
| 字节位置 | 上行 | 下行 |
| 0 ~ 1 | 0x68 0x68 | 0x68 0x68 |
| 2~3 | APDU数据长度 | APDU数据长度 |
| 4 | 地址 | 地址 |
| 5 | D（报文类型） | D（报文类型） |
| 6 | 0（报文控制字） | 1（报文控制字） |
| 7~ | 数据域 | 数据域 |
|  | CRC校验码 | CRC校验码 |

示例：

下行：68 68 00 05 01 0D 01 50 E5

上行：68 68 00 05 01 0D 00 90 24

**3、报文传输的过程**

**3.1 正常传输过程**

**3.1.1 通信控制**

主控板和工控机的通信采用先握手再通信、链路变化重新握手原则，通信结束需解除握手，一般流程如下：

1. 握手，工控机或主控板发起握手，得到应答后建立握手。
2. 通信，双方进行信息交互。
3. 解除握手，通信结束需发送解除握手命令，该操作由主动发起握手一方操作。

其过程如下：



**3.1.2 存取物资的场景**

存取物资的过程如下：

1、主控板采集并上送用户IC卡、指纹或条码信息。

2、授权，对工控机的授权分为两种情况，一是工控机自己通过采集用户信息（人脸）获得开锁权限，二是用户通过主控板上送用户信息（IC卡、指纹）给主控板获得开锁权限。

3、开锁，工控机下发开锁指令进行开锁。

4、存取物资，打开柜门后用户可进行存取物资操作。

5、关门、盘点物资，存取物资后用户关门，工控机下发盘点命令开始盘点物资。

6、结束盘点，工控机下发盘点结束命令结束盘点。

具体流程如下：

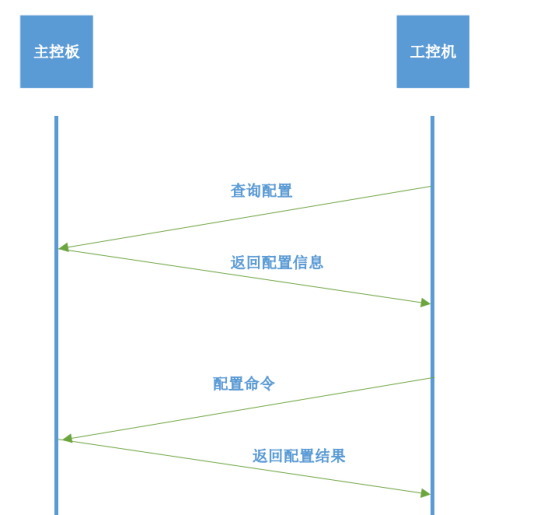


**3.1.3 状态消息循环传输场景**

状态消息每10min上送一次，其传输流程如下：



**3.1.4 配置主控板场景**



**3.2 异常传输场景**

**3.2.1 超时重传**

若2s内未接收到响应报文需重新传输消息或命令。

**3.2.2 存取物资过程中的异常**

用户长时间未关门，主控板需在5min时工控机发送门状态，由工控机做出响应。

附录1：

CRC16校验算法：

const unsigned char auchCRCHi[]=

{

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40

};

const unsigned char auchCRCLo[] =

{

0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,

0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,

0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,

0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,

0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,

0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,

0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,

0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,

0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,

0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,

0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,

0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,

0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,

0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,

0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,

0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,

0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,

0x40

};

unsigned short CRC16(unsigned char \*updata,unsigned short len)

{

unsigned char uchCRCHi=0xff;

unsigned char uchCRCLo=0xff;

short uindex;

while(len--)

{

uindex=uchCRCHi^\*updata++;

uchCRCHi=uchCRCLo^auchCRCHi[uindex];

uchCRCLo=auchCRCLo[uindex];

}

return (uchCRCHi|uchCRCLo<<8);//高字节在前，低字节在后

}

unsigned short H\_CRC16(unsigned char \*updata,unsigned short len)

{

unsigned char uchCRCHi=0xff;

unsigned char uchCRCLo=0xff;

short uindex;

while(len--)

{

uindex=uchCRCHi^\*updata++;

uchCRCHi=uchCRCLo^auchCRCHi[uindex];

uchCRCLo=auchCRCLo[uindex];

}

return (uchCRCHi<<8|uchCRCLo);

}