

东海电子收费专用短程通信 桌面发行设备（HIU） 开发手册



上海东海电脑股份有限公司

2008年11月

目 录

1	概述.....	1
2	应用接口说明.....	2
2.1	总体架构.....	2
2.2	数据接口的应用模式.....	2
2.3	数据通讯接口形式.....	3
3	透明传输应用接口定义.....	4
3.1	接口通讯规范.....	4
3.2	通讯数据帧格式.....	4
3.3	命令和数据域定义.....	5
4	API应用接口定义.....	8
4.1	API函数集列表.....	8
4.2	API函数使用说明.....	9

1 概述

东海电子收费专用短程通信桌面发行设备（TQXS-WB-HIU）是按照中国电子收费专用短程通信标准GB/T 20851-2007进行设计开发。本产品可用于公路电子收费系统领域车载单元的初始化以及个性化，可以和符合中国电子收费专用短程通信标准的车载电子标签设备通讯。

本开发手册阐述了东海电子收费专用短程通信桌面发行设备（HIU）与PC之间的应用接口，供应用集成开发人员进行应用开发参考。

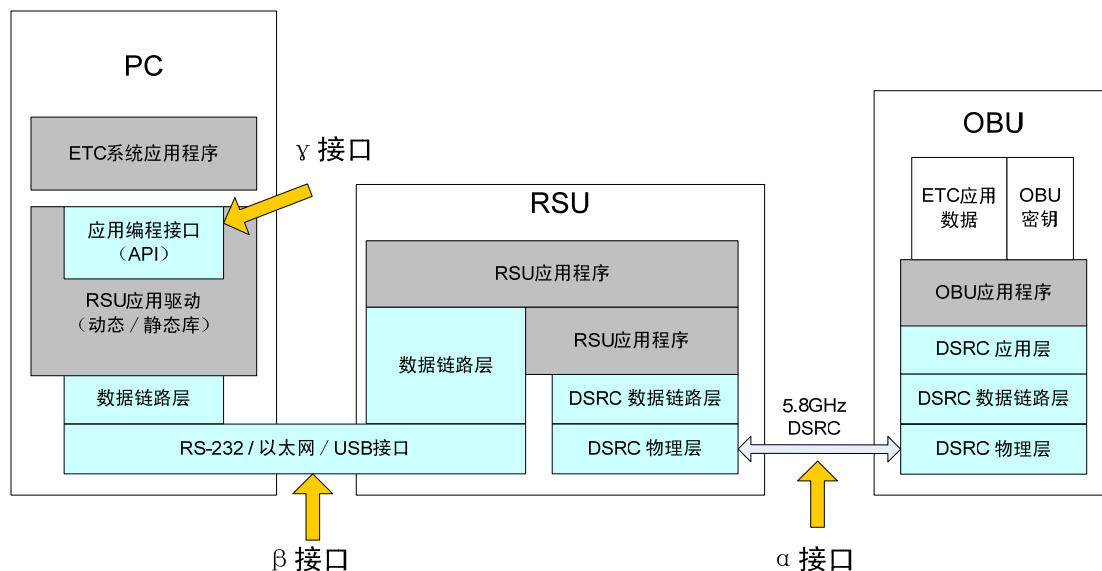
东海电子收费专用短程通信桌面发行设备（HIU）应用开发接口定义参照了以下标准、规范和技术要求：

1. GB/T20851.3-2007《电子收费 专用短程通信 第3部分：应用层》
2. GB/T20851.4-2007《电子收费 专用短程通信 第4部分：设备应用》
3. “区域联网电子不停车收费系统应用示范 技术规范 第10部分 OBE-RSE 间ETC 应用交易流程规范”，交通部公路科学研究院，2008年4月
4. “区域联网电子不停车收费系统应用示范 技术规范 第12部分电子收费RSU与车道控制器的接口”，交通部公路科学研究院，2008年4月
5. “DSRC 设备协议一致性检测送检指南”，交通部公路科学研究院，2008年4月
6. “长三角（苏、浙、沪、皖、赣）高速公路联网电子不停车收费(ETC)系统应用技术要求”，2008年9月
7. “上海市高速公路电子不停车收费(ETC)系统应用技术要求”，上海市市政工程管理局，2008年9月

2 应用接口说明

2.1 总体架构

东海电子收费专用短程通信桌面发行设备（HIU）应用系统总体架构见下图：



为了便于描述，本开发手册中定义上述系统架构中涉及到的数据接口为：

- α 接口：OBU 与 HIU 之间的 DSRC 接口。
- β 接口：HIU 与 PC 之间的原语传输接口。
- γ 接口：HIU 设备驱动程序的应用编程接口。

其中， α 接口满足 GB/T 20851 系列国家标准及《OBE-RSE 的 ETC 应用交易流程规范》等标准的要求。

本开发手册对上述总体架构中的 β 接口和 γ 接口进行定义。

2.2 数据接口的应用模式

东海电子收费专用短程通信桌面发行设备 HIU 与 PC 间通信可采用如下两种应用模式：

- β 接口方式：PC 采用向通信端口直接发送数据帧的方式与 HIU 进行通信（即上图中 β 接口），即透明传输原语指令接口方式；

- γ 接口方式：PC采用调用封装过的应用编程接口（API）函数方式与HIU进行通信（即上图中 γ 接口），API以Windows动态 / 静态库的形式提供，文件名为DHRSUDLL.dll，相应的头文件以DHRSUDLL.h命名。

2.3 数据通讯接口形式

东海电子收费专用短程通信桌面发行设备HIU与PC之间的接口物理形式为USB接口，通过USB口模拟成标准串行接口进行通讯，采用半双工的异步串行通讯方式，协议格式为“115200,N,8,1”，即波特率115200bps，无奇偶校验，8位数据，1个停止位。

3 透明传输应用接口定义

3.1 接口通讯规范

1. HIU 为服务器端，PC 为客户端；
2. 当 PC 与 HIU 建立连接后，由 PC 发 BST 消息到 HIU，HIU 收到后按照一定时间间隔（间隔可设置）连续发送 BST，直到收到 VST ；
3. 各数据包统一采用报文头+报文构成，其中报文包括指令代码和指令数据。指令是透明传输命令，即指令数据就是除去物理层的通信原语，HIU 收到后增加上物理层信息，直接发送至 OBU，并将 OBU 返回的原语除去物理层后不解析直接返回 PC。

3.2 通讯数据帧格式

HIU 和 PC 通讯的数据帧格式如下：

STX	RSCTL	LEN	CMD	DATA	BCC
-----	-------	-----	-----	------	-----

说明如下：

字段	描述
STX	帧开始标志，2 字节，取值为 55AAH，字符串方式
RSCTL	帧序列号，1 个字节 (1) 帧序列号的低半字节为 0~7，高半字节保留； (2) 帧序列号每次加一，用于标识每一次的通信。
LEN	DATA 域的长度，2 字节
CMD	命令头，1 字节
DATA	发送的数据
BCC	异或校验值，从 RSCTL 到 DATA 所有字节的异或值，1 字节

3.3 命令和数据域定义

3.3.1 CMD 命令头定义

1. PC 发往 HIU 的指令如下表：

指令名称	代码	功能说明
初始化	F0	对 HIU 关键参数进行初始化/设置，建立与 HIU 的连接
BST	F1	判断 PC 是否与 HIU 处于连接状态
透明传输	F2	数据域为通信原语，HIU 收到后直接转发给 OBU
PSAM 通道指令	F9	对 HIU 中 PSAM 卡的通道操作

2. HIU 发往 PC 的信息帧如下表：

帧名称	代码	功能说明
初始化返回	E0	返回 HIU 设备编号等信息
BST 返回	E1	返回 BST 应答数据
透明传输	E2	数据域为通信原语，HIU 收到 OBU 应答后直接转发给 PC
PSAM 通道指令	E9	对 HIU 中 PSAM 卡的通道操作响应

3.3.2 PC 指令 Data 域结构

1. 初始化指令—F0

位置	字节数	数据元	数据内容
0	4	Datetime	UNIX 当前时间，高位在前
4	1	BST Interval	BST 间隔，单位毫秒，范围 1-10ms
5	1	RetryInterval	交易重试间隔，单位毫秒，范围 1-10ms
6	1	TxPower	功率级数，范围 0-31； 0：关闭天线，31：最大功率
7	1	PLLChannelID	信道号，范围 0-1

2. BST 指令—F1

位置	字节数	数据元	数据内容
----	-----	-----	------

0	n	BST	BST 原语, hex 格式
---	---	-----	----------------

3. 透明传输指令—F2

位置	字节数	数据元	数据内容
0	1	ConfirmMode	1: 需等待 OBU 应答, 0: 无需等待应答
1	n	原语	通信原语

4. PSAM 通道指令—F9

位置	字节数	数据元	数据内容
0	1	PSAM 卡槽号	0x00~0x05 表示卡槽号
1	1	APDU 指令数	
2	1	指令 1 长度	
3	n	指令 1 内容	
.....	

3.3.3 HIU 信息帧数据结构

1. 初始化返回 - E0

位置	字节数	数据元	数据内容
0	1	FrameType	数据帧类型标识, 此处取值 E0
1	1	RSUStatus	RSU 主状态参数; 0x00 表示正常, 否则表示异常;
2	1	RSUManulID	RSU 厂商代码
3	3	RSUIndividualID	RSU 编号
6	2	RSUVersion	RSU 软件版本号
8	10	Reserved	保留字节。

2. BST 返回—E1

位置	字节数	数据元	数据内容
0	1	CMDType	指令代码, 此处取值 E1;

1	1	Status	0x00 表示收到 VST, 0x01 表示未收到 VST
2	n	VST	Status 为 0x00 时, 有此域; 0x01 时此域长度为 0

3. 透明传输返回—E2

位置	字节数	数据元	数据内容
0	1	Status	0x00 表示收到信息, 0x01 表示超时
1	n	原语	Status 为 0x00 时, 有此域; 0x01 时此域长度为 0

4. PSAM 通道指令返回—E9

位置	字节数	数据元	数据内容
0	1	Status	0x00 表示正常返回, 0x01 表示无响应
1	1	响应指令数	
2	1	数据 1 长度	
3	n	数据 1 内容	
.....	

4 API 应用接口定义

本接口参照《区域联网电子不停车收费系统应用示范 技术规范 第12部分 电子收费 RSU与车道控制器的接口》中关于原语模式的API函数定义，将必须用到的DSRC服务原语分别定义函数，适用于车道控制器、OBU发行设备调用。

4.1 API 函数集列表

1. HIU 设备指令

指令名称	指令	功能说明
打开设备端口	DHRSUOpen	打开与 HIU 连接的串口
关闭设备端口	DHRSUClose	关闭与 HIU 的连接
HIU 初始化	RSU_Init_rq	对 HIU 关键参数进行初始化/设置，建立与 HIU 的连接
HIU 初始化返回	RSU_Init_rs	返回 HIU 设备编号等信息
PSAM 通道复位	PSAM_Channel_Reset	对 HIU 的 PSAM 通道复位,所有 PSAM Slot 中的 PSAM 需重新上电。建立与 HIU 的重新连接后必须执行该操作。
PSAM 上电	PSAM_INIT_rq	PSAM 上电
PSAM 上电返回	PSAM_INIT_rs	PSAM 上电返回
PSAM 通道指令	PSAM_Channel_rq	对 HIU 中 PSAM 卡的通道操作
PSAM 通道指令返回	PSAM_Channel_rs	对 PSAM 卡通道操作响应

2. INITIALISATION 原语指令

指令名称	指令	功能说明
BST	INITIALISATION_rq	HIU 广播 BST 给 OBU
VST	INITIALISATION_rs	HIU 接收 OBU 返回的 VST

3. ACTION 原语指令

指令名称	指令	功能说明
GetSecure_Request	GetSecure_rq	对 OBU 文件进行安全读取
GetSecure_Response	GetSecure_rs	HIU 接收 OBU 返回的系统文件信息
TransferChannel_Request	TransferChannel_rq	HIU 通过透明通道对 OBU 进行操作
TransferChannel_Response	TransferChannel_rs	透明通道操作的响应
SetMMI_Request	SetMMI_rq	对 OBU 进行界面提示
SetMMI_Response	SetMMI_rs	对 OBU 进行界面提示的响应

4. EVENT_REPORT 原语指令

指令名称	指令	功能说明
EVENT_REPORT_Request	Event_Report_rq	释放 OBU

4.2 API 函数使用说明

4.2.1 HIU 设备指令

1. 打开设备端口

函数：int RSU_Open(int com)

参数：com—串口号

返回：0—成功，非 0—失败

2. 关闭设备端口

函数：int RSU_Close()

参数：无

返回：0—成功，非 0—失败

3. HIU 初始化

函数：int RSU_INIT_rq(char *UNIXTime, int BSTInterval, int RetryInterval ,
int TxPower, int PLLChannelID)

参数： UNIXTime—UNIX 时间，4 字节；

BSTInterval—BST 间隔，单位毫秒，范围 1-10ms；

RetryInterval—交易重试间隔，单位毫秒，范围 1-10ms；

TxPower—RSU 发射功率设置，取值范围为 0~31，0：关闭天线，31：
最大功率；

PLLChannelID—信道号，范围 0-1

返回： 0—成功，非 0—失败

4. HIU 初始化返回

函数： int RSU_INIT_rs(int *RSUStatus, int *rlen, char *RSUinfo)

参数： RSUStatus —RSU 主状态参数；0x00 表示正常，否则表示异常；

rlen —信息长度；

RSUinfo—RSU 设备信息，包括 RSU 厂商代码（1 字节）、RSU 编号（3
字节）、RSU 软件版本号（2 字节）、保留字节（10 字节）

返回： 0—成功，非 0—失败

5. PSAM 通道复位

函数： int PSAM_CHANNEL_Reset()

参数： 无

返回： 0—成功，非 0—失败

6. PSAM 上电

函数： int PSAM_INIT_rq(int com,int pos, int baud)

参数： com—串口号；

pos—PSAM 卡槽号 0x00~0x05；

baud—PSAM 通讯速率， 9600， 38400， 115200；

返回： 0—成功，非 0—失败

7. PSAM 上电返回

函数： int PSAM_INIT_rs (char *PSAMNO)

参数: PSAMNO —PSAM 终端编号;

返回: 0—成功, 非 0—失败

8. PSAM 通道指令

函数: int PSAM_CHANNEL_rq(int pos, int APDULIST, char *APDU)

参数: pos—PSAM 卡槽号 0x00~0x05;

APDUList—PSAM 指令数;

APDU—PSAM 指令, 按顺序为指令 1 长度 (1 字节)、指令 1、指令 2 长度 (1 字节)、指令 2、...

返回: 0—成功, 非 0—失败

9. PSAM 通道指令返回

函数: int PSAM_CHANNEL_rs(int *APDULIST, char *Data)

参数: APDUList—PSAM 指令数;

Data—PSAM 指令返回, 按顺序为数据 1 长度 (1 字节)、数据 1、数据 2 长度 (1 字节)、数据 2、...

返回: 0—成功, 非 0—失败

4.2.2 INITIALISATION 原语指令

1. BST

函数: int INITIALISATION_rq (char * BeaconID , char *Time, int Profile, int MandApplicationlist, int MandApplicationlen, char *MandApplication, int Profilelist)

参数: BeaconID—由 1 字节 manufacturerID 和 3 字节 individualID 组成;

Time—UNIX 时间, 4 字节;

Profile—配置选项, 长度为 1 字节, 配置值如下:

00H—配置 0(A 类)的信道 1 01H—配置 0(A 类)的信道 2

10H—配置 1(B 类)的信道 1 11H—配置 1(B 类)的信道 2

MandApplicationlist—应用列表数

MandApplicationlen—应用列表长度

MandApplication—应用列表，定义见下表

1	Option indicator	0	Dsrc-did 不显示
	Option indicator	1	Parameter 存在
	DsrcApplicationEntityID	00 0001	无扩展, AID = 1 (ETC)
1	Option indicator icctransmode	1 xxxxxx x	Container 存在 RSU 支持的卡片交易模式
1	PretreatPara	0010 1001	Container Type=0x29 携带文件: 预处理操作参数
1	Option indicator Option indicator Option indicator Option indicator fill fill fill fill	0 0 0 0 0 0 0 0	不读取 0002 文件 不读取 00012 文件 不读取 00015 文件 不读取 00019 文件 填充 填充 填充 填充
1	sysInfoFileMode	xxxxxxxx	系统信息文件读取长度
2	Length0002	xxxxxxxx xxxxxxxx	预读 0002 偏移量和长度, 由 OP 确定是否存在
2	Length0012	xxxxxxxx xxxxxxxx	预读 0012 偏移量和 0012 长度, 由 OP 确定是否存在
2	Length0015	xxxxxxxx xxxxxxxx	预读 0015 偏移量和 0015 长度, 由 OP 确定是否存在
2	Length0019	xxxxxxxx xxxxxxxx	预读 0019 偏移量和 0019 长度, 由 OP 确定是否存在

Profilelist—配置文件号

返回: 0—成功, 非 0—失败

2. VST

函数: int INITIALISATION_rs (int *ReturnStatus, int *Profile, int *Applicationlist, char *Application, char *ObuConfiguration)

参数: ReturnStatus—0 表示收到 VST; 1 表示未收到 VST, 后面参数值均可忽略

profile—OBU 返回的配置, 长度为 1 字节, 配置值如下:

00H—配置 0(A 类)的信道 1 01H—配置 0(A 类)的信道 2

10H—配置 1(B 类)的信道 1 11H—配置 1(B 类)的信道 2

applicationlist—应用列表数;

application—应用列表，按顺序为：

AID、DID、Option indicator、
Container1、Container1_Data、
Container2、Container2_Data ...

应用列表的具体定义参见 GB/T 20851.3-2007、GB/T 20851.4-2007
及《OBE-RSE 的 ETC 应用交易流程规范》中的定义；

ObuConfiguration—OBU 配置信息内容，长度为 7 字节，包括以下内容：

4 字节 OBU 的 MAC 地址，1 字节设备硬件状态，2 字节 OBU 状态；

返回：0—成功，非 0—失败

4.2.3 ACTION 原语指令

1. GetSecure_Request

函数：int GetSecure_rq (int accessCredentialsOp, int mode, int DID, char *

AccessCredentials, int keyIdForEncryptOp, int FID, int offset, int length, char * RandRSU,
int KeyIdForAuthen, int KeyIdForEncrypt)

参数：accessCredentialsOp—是否带认证码，0 表示不带有认证码，1 表示带有；

mode—确认模式，1：需应答，0：无需应答；

DID—要读取的 OBU 应用号，ETC 应用为 1，标识站应用为 2；

AccessCredentials—OBU 认证码，8 字节；根据 accessCredentialsOp
决定 RSU 发出的数据帧中是否有该域，如无，则填 8 字节 0x00；

keyIdForEncryptOp—是否存在 keyIdForEncrypt 域，0 不存在，1 存在；

FID—要读取的 OBU 文件号；

offset—要读取的 OBU 文件偏移地址；

length—要读取的 OBU 文件长度；

RandRSU—RSU 发出的 8 字节随机数；

KeyIdForAuthen—OBU 计算认证码的密钥索引号；

KeyIdForEncrypt—OBU 加密密钥索引号。

返回：0—成功，非 0—失败

2. GetSecure_Response

函数: GetSecure_rs(int *DID, int *FID, int *length, char * File, char *authenticator, int *ReturnStatus)

参数: DID—读取的 OBU 应用号;

FID—读取的 OBU 文件号;

length—读取的 OBU 文件长度;

File—读取的 OBU 文件内容;

Authenticator—OBU 认证码, 采用 ESAM 情况下为 8 字节 0x00;

ReturnStatus—OBU 处理状态;

返回: 0—成功, 非 0—失败;

3. TransferChannel_Request

函数: int TransferChannel_rq(int mode, int DID, int ChannelID, int APDULIST, char *APDU)

参数: mode—确认模式, 1: 需应答, 0: 无需应答;

DID—DSRC-DID, 目录号;

actionType—Action 服务类型, 此处填 0x03, 表示 Transfer_channel;

ChannelID—通道 ID 号, 0- OBU, 1 - ICC , 2 - ESAM;

APDULIST—对通道操作的 APDU 命令数;

APDU—对通道操作的 APDU 命令, 按顺序为指令 1 长度 (1 字节)、指令 1、指令 2 长度 (1 字节)、指令 2、...

返回: 0—成功, 非 0—失败。

4. TransferChannel_Response

函数: int TransferChannel_rs (int *DID, int *ChannelID, int *APDULIST, char *Data)

参数: DID—DSRC-DID, 目录号;

ChannelID—通道 ID 号;

APDULIST—对通道操作的 APDU 命令数;

Data—对通道操作的 APDU 命令返回数据，按顺序为数据 1 长度（1 字节）、数据 1、数据 2 长度（1 字节）、数据 2、...

返回：0—成功，非 0—失败。

5. SetMMI _Request

函数：int SetMMI_rq(int mode, int DID, int SetMMIPara)

参数：mode—确认模式，1：需应答，0：无需应答；

DID—DSRC-DID,目录号；

SetMMIPara—人机界面参数，

0—交易正常，1—交易异常，2—联系运营商；

返回：0—成功，非 0—失败

6. SetMMI _Response

函数：int SetMMI_rs (int *DID,int *ReturnStatus)

参数：DID—DSRC-DID,目录号；

ReturnStatus—OBU 处理状态；

返回：0—成功，非 0—失败。

4.2.4 EVENT_REPORT 原语指令

1. EVENT_REPORT_Request

函数：int EVENT_REPORT_rs (int mode, int DID, int EventType)

参数：mode—确认模式，1：需应答，0：无需应答；

DID—DSRC-DID,目录号；

EventType—Release=0

返回：0—成功，非 0—失败