# 东海电子收费专用短程通信 便携式 ETC 处理终端 开发手册



## 上海东海电脑股份有限公司

2008年11月

目	录
_	

1	概这	<u>.</u>	1
2	二沙	x应用开发环境	2
	2.1	二次应用开发环境需求	2
	2.2	建立终端与PC的连接	2
3	二め	文应用开发指南	4
	3.1	运行EVC 4.0	4
	3.2	东海便携式ETC处理终端SDK开发包库函数说明	9
	3.2.	1 DSRC处理库函数	9
	3.2.2	2 非接触IC卡读写库函数1	1

# 1 概述

东海电子收费专用短程通信便携式 ETC 处理终端(TQXS-WB-HIU-2)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,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。本终端具有 IC 接口,可操作发行各类用户卡;具有 DSRC 专用短程通信接口,可对 OBU 进行初始化、个人化和查验 OBU 数据的功能。

本开发手册阐述了东海电子收费专用短程通信便携式ETC处理终端 (TQXS-WB-HIU-2)二次应用开发的指导说明,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便携式ETC处理终端(TQXS-WB-HIU-2)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》
- "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
- "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
- "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
- "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 二次应用开发环境

## 2.1 二次应用开发环境需求

东海电子收费专用短程通信便携式 ETC 处理终端内嵌 Windows CE 5.0 操作 系统,应用集成开发人员可在 PC 上使用嵌入式开发工具进行二次开发,并将程 序下载至终端内存中。

在进行二次开发前,PC上建议安装如下开发包:

- eMbedded Visual C++ 4.0 (SP4)
- Standard SDK for Windows CE .net
- Micorosoft ActiveSync 4.5
- 东海便携式 ETC 处理终端 SDK 开发包

## 2.2 建立终端与 PC 的连接

- PC 上安装 Micorosoft ActiveSync 4.5
- 用 USB 数据线连接 PC 机和手持机,在 PC 机端将会检测到 USB 设备,然 后指定路径安装驱动;
- 选择东海便携式 ETC 处理终端驱动目录下的驱动文件,安装好驱动后,将
   会弹出"建立合作关系"对话框,选择'否',点击下一步继续,如下图:

新建合作关系	2	
	建立合作关系	
	要想在移动设备和此计算机之间同步信息,必须先在 二者之间建立合作关系。 要建立合作关系吗? ② 是(Y) 建立合作关系,以便在移动设备和此计算机之间	
	同步信息。 (•) 蒼(0) 无需同步信息。将移动设备设置为"来宾",以便 在移动设备和此计算机之间复制或移动信息。	
	<上一步 (8) 下一步 (8) ▶ 取消 帮助	

■ 这时 ActiveSync 显示已连接,如下图。

🕲 Licro	soft ActiveSync	
文件(22) 礼	视图(V) 工具(T) 帮助(H)	
🔕 同步	🕒 日程安排 🔯 浏览	
来宾		
已连接		S
		隐藏详细信息 🗙
信息类型	状态	

■ 此时可以在我的电脑里打开移动设备, 查看 ETC 终端上的文件。 注意事项:

- ETC 终端上"NandFlash"文件夹是掉电以后还能够保存的,如果拷贝文件到其他文件夹掉电以后会丢失。
- 请等 ETC 终端 Windows CE 系统启动完毕后, PC 再与其进行连接。



# 3 二次应用开发指南

## 3.1运行 EVC 4.0

1. File 菜单下选择 New, 如下图

📴 Licrosoft eLbedde	d Visual C++	
	● 目 ! 目 ⊕	
<u>File E</u> dit <u>V</u> iew <u>I</u> nsert	<u>P</u> roject <u>B</u> uild <u>T</u> ools <u>W</u> indow <u>H</u> elp	
New Ctrl+N	💼 🗠 + 🗠 + 📴 🗖 📅 🐂	•
	-	$\overline{\mathbf{v}}$
Dren Workspace		
: Saye Workspace		
Close Wor <u>k</u> space		
Eave Ctrl+S		
Save <u>A</u> s		
Save All		-
Page Set <u>u</u> p		
Print Ctrl+P		
Recent <u>F</u> iles 🕨 🕨		
Recent Wo <u>r</u> kspaces 🕨		
E <u>w</u> it		
	•	
Build Debug	$\lambda$ Find in Files 1 $\lambda$ Find in Files 2 /	┥╽┝
Creates a new documer	it, project or workspace	

 选择 WCE MFC AppWixard(exe), CPUs 选择 Win32 (WCE ARM V4I),在 Project name 中输入工程名,在 Location 中确定工程所在的路径,点击 "OK"按钮继续,如下图:

3 WCE Application	Project <u>n</u> ame:
■ WCE ATL COM Appwizard	test
WCE Dynamic-Link Elbrary	Lo <u>c</u> ation:
WCE MFC AppWizard (exe)	D:\WCE\test
	© Create new workspace ○ Add to current workspace □ Dependency of: CPUs:
	□Win32 (WCE ARMV4)
	Win32 [WCE ARMV4I]  Win32 [WCE ARMV4I]
	□Win32 (WCE MIPS16)
	TWin32 IWCE MIPSID

 在 WCE MFC AppWizard(exe)- Step 1 of 4 中,将语言设置为"其他",保 持默认值,点击"Next"继续,如下图:

VCE NFC AppVizard (exe) - Step	plof4 🥐 🔀
Applicatios OX ×	What type of application would you like to create? ○ Single document ○ Dialog based ☑ Document/⊻iew architecture support?
	What language would you like your resources in? 英语[美国] (ceappwzenu.dll)
< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish Cancel

4. 在 WCE MFC AppWizard(exe)- Step 2 of 4 中, 点击"Next"继续, 如下



VCE IFC AppVizard (exe	) - Step 2 d	of 4 ?	×
Applicstion ()K	×	What features would you like to include? <u>W</u> indows Sockets Windows H <u>e</u> lp Active <u>X</u> Controls	
		Please enter a <u>title for your dialog:</u> test Help contents file: test.htp	
	< <u>B</u> ack	<u>Next&gt;</u> <u>F</u> inish Cancel	

图:

5. 在 WCE MFC AppWizard(exe)- Step 3 of 4 中,选择 MFC 静态连接,点击 "Next"继续,如下图:

VCE NFC AppVizard (ex	xe) - Step 3	of 4		? 🛛
File Edit Yiew Jesert Build Help         Project         Project         ************************************		Would you l Yes, p No, <u>th</u> How would y As a s As a <u>s</u>	like to generate sou lease ank you you like to use the l hared <u>D</u> LL tatically linked libra	MFC library?
	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel

6. 在 WCE MFC AppWizard(exe)- Step 4 of 4 中, 点击"Finish"结束向导

WCE MFC AppWizard (exe	) - Step 4 o	f 4	? 🛛
		AppWizard creates the you: CTestApp CTestDlg	following <u>c</u> lasses for
		C <u>l</u> ass name:	Header file:
		CTestApp	testApp.h
		Base class: CWinApp	Implementation file: testApp.cpp
	< <u>B</u> ack	<u>N</u> ext > <u>F</u> in	ish Cancel





7. 选择 "Build" 菜单中的 "Set Active Platform., 如下图:

 在 Set Active Platform 对话框中,选择新安装的 DHETCHandHeld,点击"OK" 按钮继续,如下图:

Set Active Platform	? 🛛
Windows CE Platforms:	ОК
STANDARDSDK	Cancel

9. 点击运行,工程编译链接完成后,会自动下载到 ETC 处理终端的 WINCE 根目



录上去,并运行。

注意:

如果 WCE 无法连接,请修改一下 ETC 处理终端的网卡设置为 DHCP 动态获取 IP。

### 3.2东海便携式 ETC 处理终端 SDK 开发包库函数说明

### 3.2.1 DSRC 处理库函数

- 1. 打开 DRSC 设备端口
  - 函数: DHETCOpenReader(); 参数: 无 返回: 0-成功, 非0-失败
- 2. 关闭 DRSC 设备端口
  - 函数: DHETCCloseReader () 参数: 无 返回: 0-成功, 非0-失败
- 3. DSRC 设备蜂鸣
  - 函数: DHETCReaderBeep(unsigned char \_10ms) 参数: \_10ms, 蜂鸣时间(取值范围1~30,30即300ms) 返回: 0-成功,非0-失败
- 4. 复位 DSRC 设备

函数: DHETCResetReader() 参数: 无 返回: 0-成功, 非0-失败

5. 初始化 DSRC 设备

函数: DHETCInitDSRC(unsigned char BSTInterval, unsigned char RetryInterval,



unsigned char TxPower, unsigned char PLLChannelID, unsigned char \*Respond,int \*rlen)

参数: 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; PLLChanneIID-信道号,范围 0-1 Respond—OBU 返回数据 rlen—OBU 返回数据长度 返回: 0-成功,非0-失败

6. 发送 BST

函数: DHETCDSRCbst(unsigned char \*Command, int len, unsigned char

\*Respond, int \*rlen, int NoRespond)

参数: Command - BST 数据串, 从段字头 0x91 开始,

例如"91 C0 05 D1 00 00 49 33 B5 88 00 01 01 00";

len - BST 数据串长度;
Respond - OBU 返回数据
rlen - OBU 返回数据长度
NoRespond - 0 需等待 OBU 应答 非0 无需等待 OBU 应答
返回: 0-成功,非0-失败

7. DSRC 交易

函数: DHETCDSRC(unsigned char \*Command,int len,unsigned char \*Respond,int

\*rlen,int NoRespond)

参数: Command - DSRC 命令数据串,从段字头 0x91 开始, 例如"91 04 01 04 1A 00";
len - 命令数据串长度;
Respond - OBU 返回数据
rlen - OBU 返回数据长度
NoRespond - 0 需等待 OBU 应答 非 0 无需等待 OBU 应答



返回: 0一成功,非0一失败

- 8. PSAM 通道复位
  - 函数: DHETCResetPSAM()
  - 参数: 无参数
  - 返回: 0一成功,非0一失败
- 9. 添加 PSAM 通道命令
  - 函数: DHETCAddPSAMCommand(unsigned char \*cmd,int cmdlen)

参数: cmd - PSAM APDU 命令,例如 "\x00\x84\x00\x00\x04"

- cmdlen PSAM APDU 命令长度
- 返回: 0-成功,非0-失败
- 10. PSAM 通道命令列表处理
  - 函数: DHETCPSAMChannel(int PSAMSlot)
  - 参数: PSAMSlot PSAM Slot 号, 0  $\sim$  2
  - 返回: 0一成功,非0一失败
- 11. 从第一条开始依次取出 PSAM 处理结果
  - 函数: DHETCPopPSAMRespond(unsigned char \*buf,int &rlen)
  - 参数: buf PSAM 返回的处理结果
    - rlen 处理结果长度
  - 返回: 0一成功,非0一失败
  - 注: 每发一次 DHETCPopPSAMRespond 指令, 按顺序返回对应的 DHETCAddPSAMCommand 发出的指令执行结果

## 3.2.2 非接触 IC 卡读写库函数

1. 打开非接触 IC 卡读写器

函数: DHOpenReader();

参数:无 返回:0-成功,非0-失败

2. 关闭非接触 IC 卡读写器

函数: DHCloseReader(); 参数: 无 返回: 0-成功, 非0-失败

3. 读卡器蜂鸣

函数: DHReaderBeep(unsigned char \_10ms) 参数: \_10ms, 蜂鸣时间(取值范围1 ~ 30, 30即 300ms) 返回: 0-成功, 非0-失败

- 4. 非接触 IC 卡读写器复位
  - 函数: DHResetReader() 参数: 无参数 返回: 0-成功,非0-失败
- 5. 侦测非接触卡

函数: MIRequestEX2(unsigned long \*CardSN, unsigned char \*type, unsigned char \*ATS, int &rlen )

参数: CardSN — 返回卡号 type — 卡类型 0x10 M1 非接触卡 0x11 — 卡通 M1 非接触卡 0x12 国标 CPU 卡 0x13 — 卡通 CPU 非接触卡 ATS — 返回 ATS rlen — ATS 长度 返回: 0—成功, 非 0—失败

#### 6. M1 非接触卡扇区密钥认证

函数: MILoadkey(unsigned char Block, unsigned char KEYAB, unsigned char \*Key)

参数: Block — 要认证的块号 KEYAB — 密钥类型, 0 keyA, 1 KeyB Key — 密钥 返回: 0-成功, 非 0-失败

#### 7. 读 M1 非接触卡块数据

函数: MIReadBlock(unsigned char Block, unsigned char Num, unsigned char \*Data)

- 参数: Block 要读的起始块号
  - Num 读取块数量
  - Data 返回数据
- 返回: 0一成功,非0一失败
- 注意: 读的块必须在同一扇区

#### 8. 写 M1 非接触卡块

函数: MIWriteBlock(unsigned char Block, unsigned char Num, unsigned char \*Data)

- 参数: Block 要写的起始块号
  - Num 写块数量
  - Data 返回数据
- 返回: 0一成功,非0一失败
- 注意: 写的块必须在同一扇区
- 9. M1 非接触卡钱包块加值

函数: MIIncrement(unsigned char Block, long value)

- 参数: Block 钱包块号
  - Value 增加的值
- 返回: 0一成功,非0一失败



#### 10. M1 非接触卡钱包块减值

函数: MIDecrement (unsigned char Block, long value)

- 参数: Block 钱包块号
  - Value 扣减的值
- 返回: 0一成功, 非0一失败

#### 11. 钱包恢复操作

- 函数: MIRestore(unsigned char SBlock, unsigned char DBlock)
- 参数: SBlock 备份钱包块号 DBlock — 需回复的钱包块号
- 返回: 0-成功,非0-失败

#### 12. 结束 M1 非接触卡操作

- 函数: MIHalt()
- 参数: 无参数
- 返回: 0一成功,非0一失败

#### 13. CPU卡 MIPRO 命令

#### 函数: MIPROCommand(unsigned char \*Command,int len,unsigned char

#### \*Respond, int &rlen)

参数: Command - CPU卡 APDU 指令;

len - APDU 指令数据长度;

Respond - CPU卡返回数据

```
rlen - CPU 返回数据长度
```

#### 返回: 0一成功,非0一失败