

东海电子收费专用短程通信
便携式 ETC 处理终端
开发手册



D O N G H A I

上海东海电脑股份有限公司

2008 年 11 月

目 录

1	概述.....	1
2	二次应用开发环境.....	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	二次应用开发指南.....	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	非接触IC卡读写库函数.....	11

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》
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 二次应用开发环境

2.1 二次应用开发环境需求

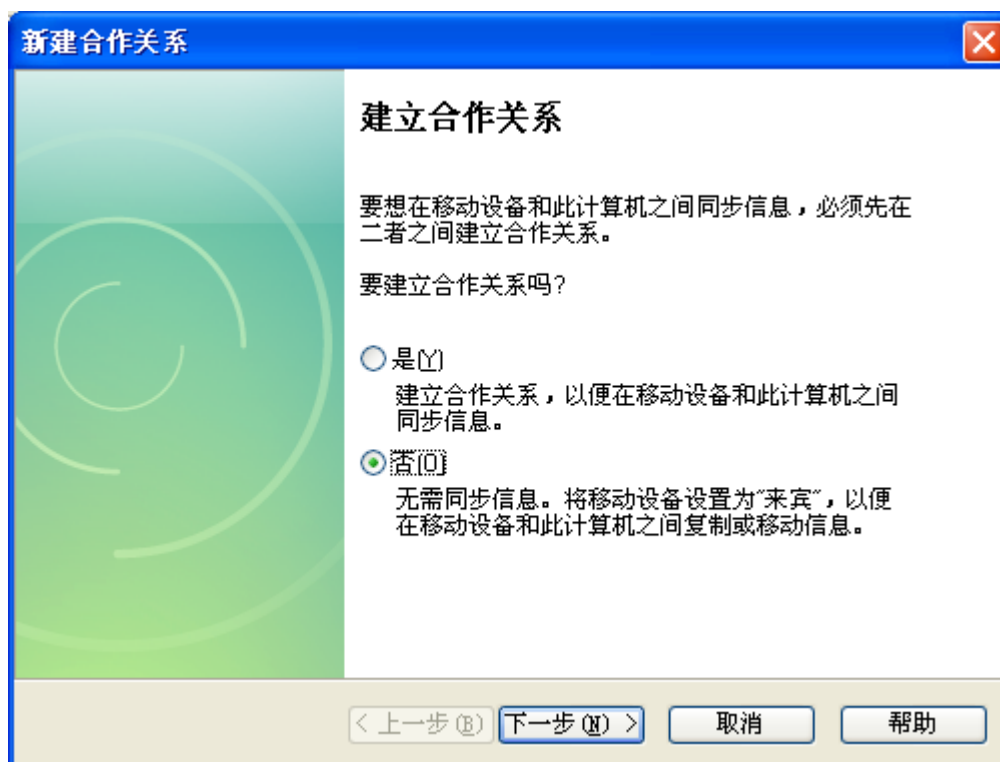
东海电子收费专用短程通信便携式 ETC 处理终端内嵌 Windows CE 5.0 操作系统，应用集成开发人员可在 PC 上使用嵌入式开发工具进行二次开发，并将程序下载至终端内存中。

在进行二次开发前，PC 上建议安装如下开发包：

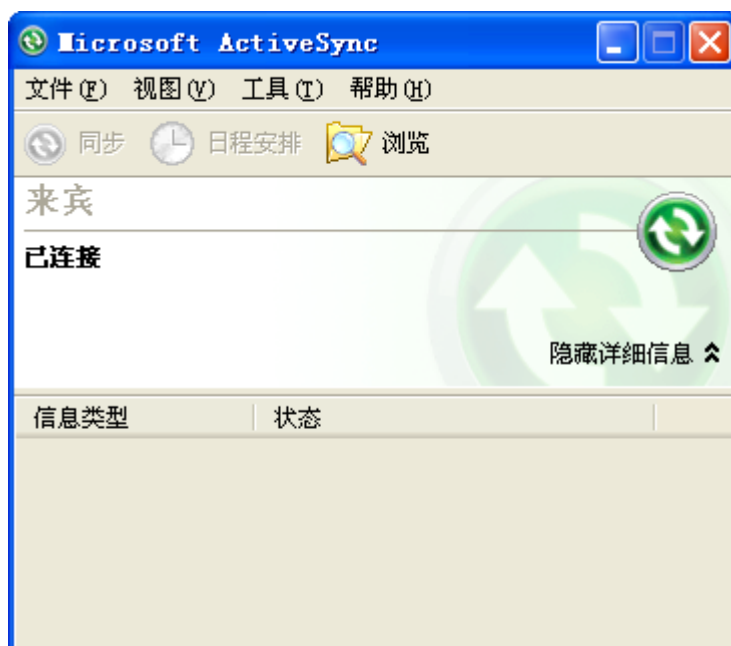
- eMbedded Visual C++ 4.0 (SP4)
- Standard SDK for Windows CE .net
- Microsoft ActiveSync 4.5
- 东海便携式 ETC 处理终端 SDK 开发包

2.2 建立终端与 PC 的连接

- PC 上安装 Microsoft 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：



■ 这时 ActiveSync 显示已连接，如下图。



■ 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。

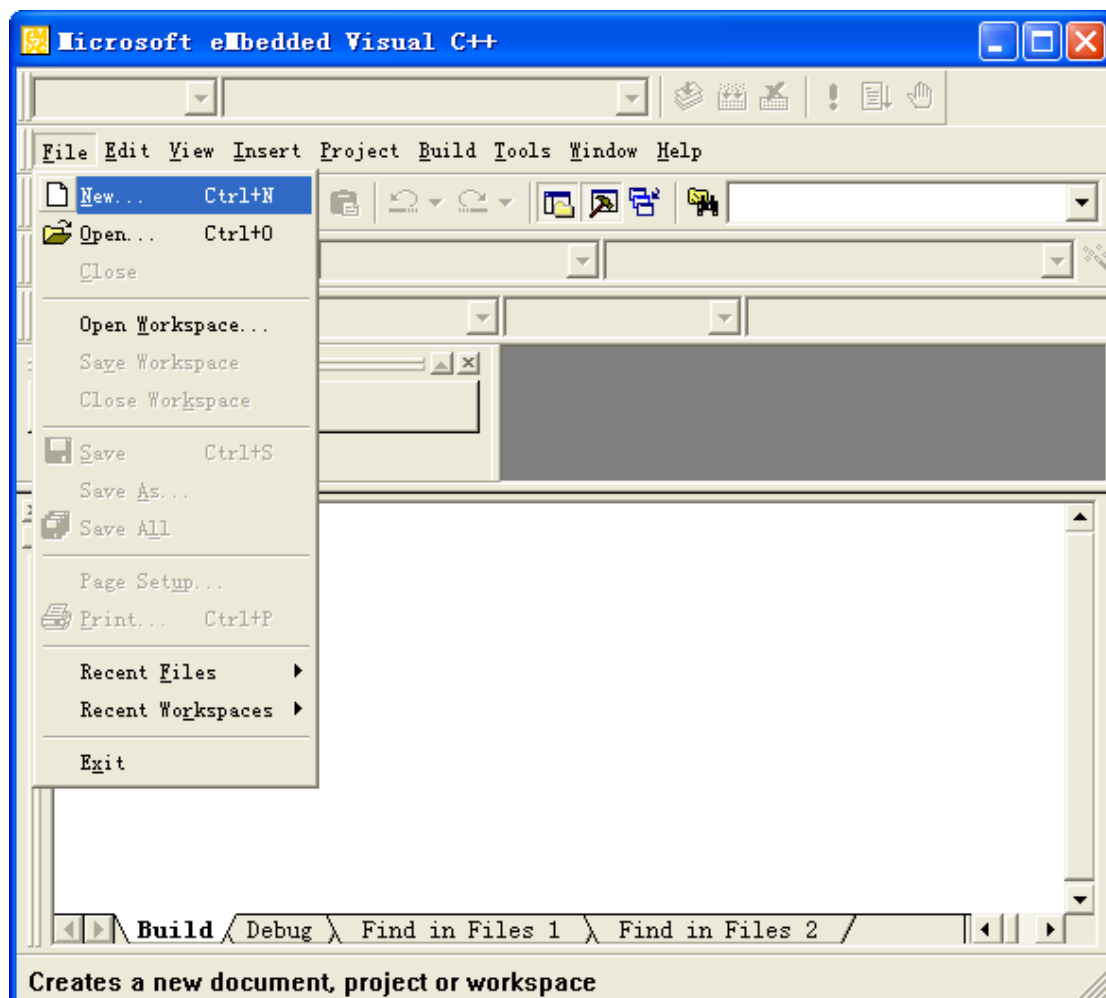
注意事项：

- 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。

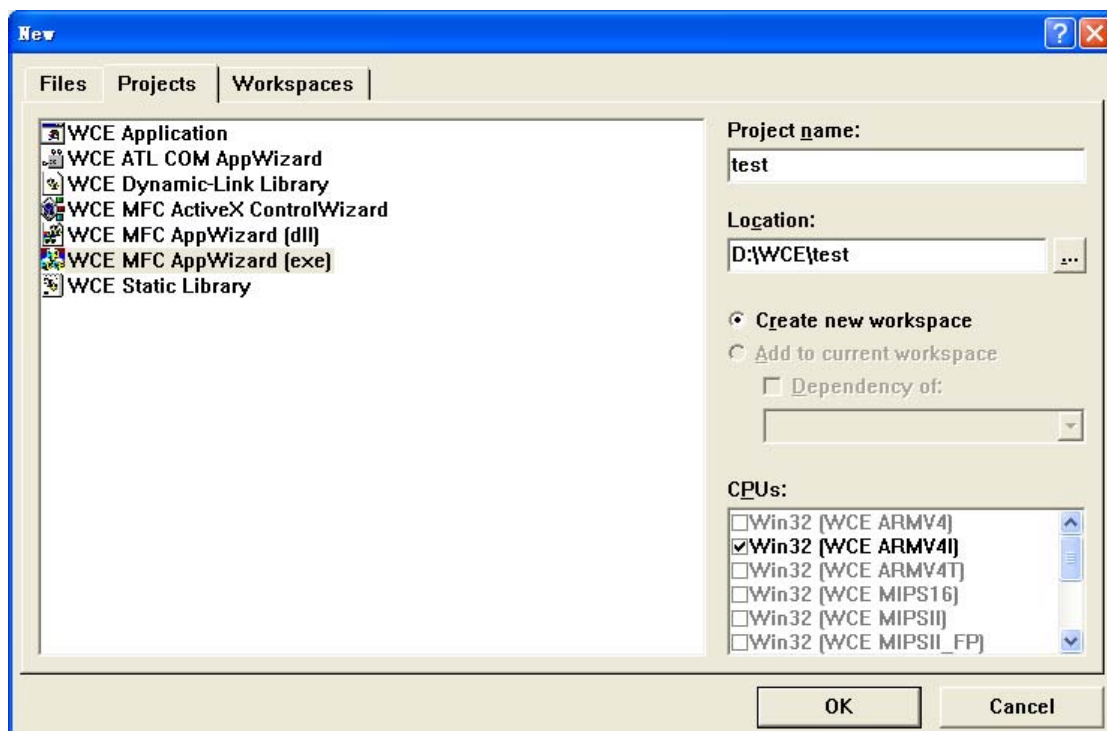
3 二次应用开发指南

3.1 运行 EVC 4.0

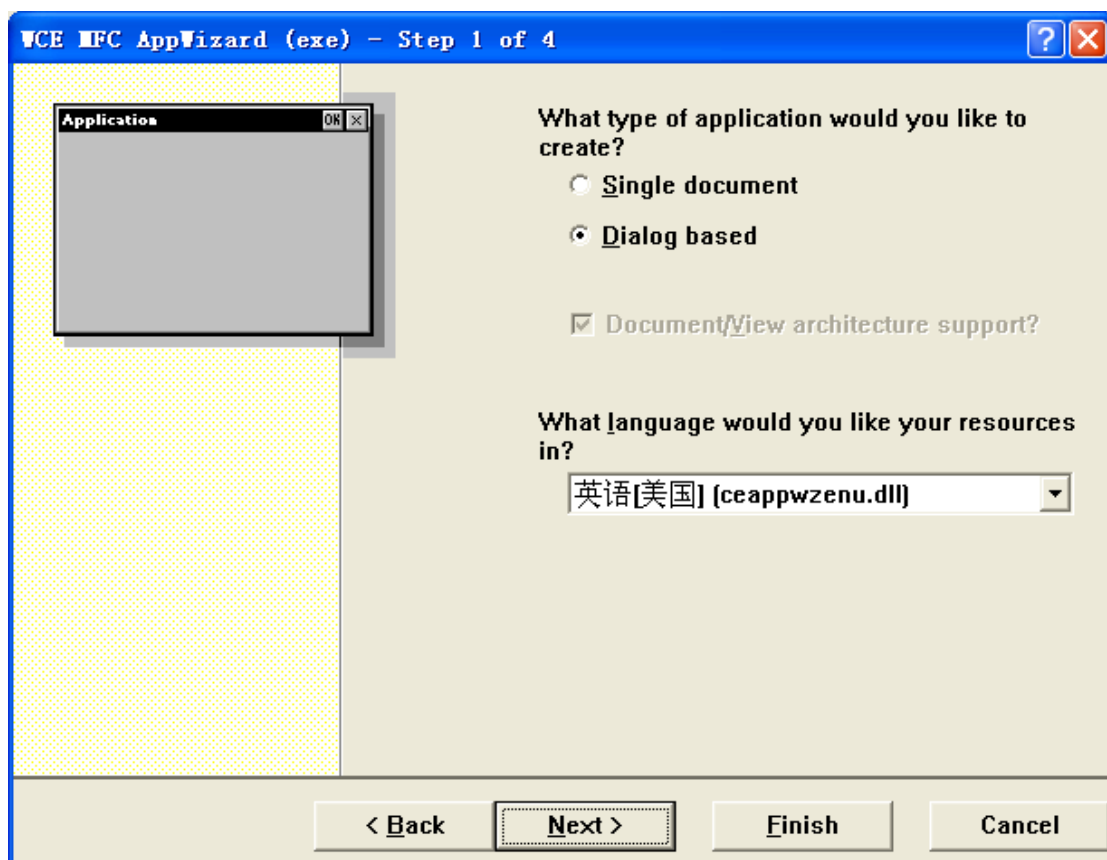
1. File 菜单下选择 New，如下图



2. 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：

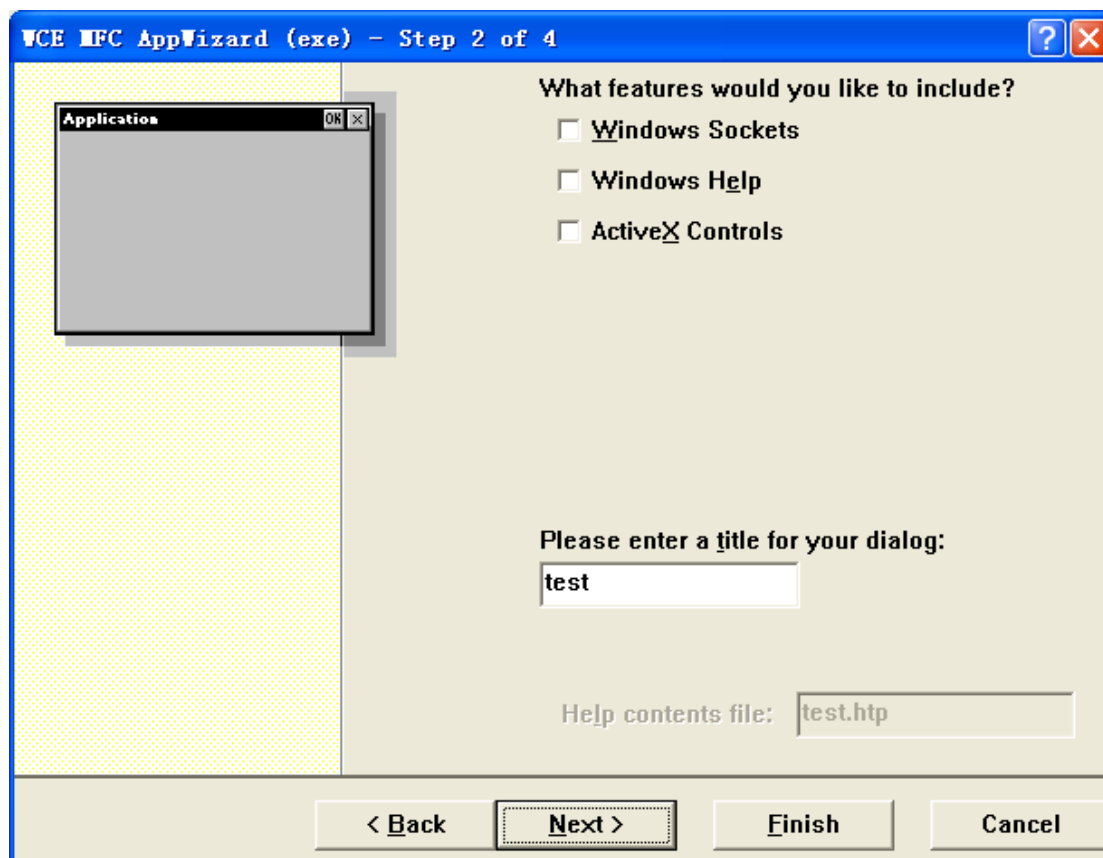


3. 在 WCE MFC AppWizard(exe)- Step 1 of 4 中, 将语言设置为“其他”, 保持默认值, 点击“Next”继续, 如下图:

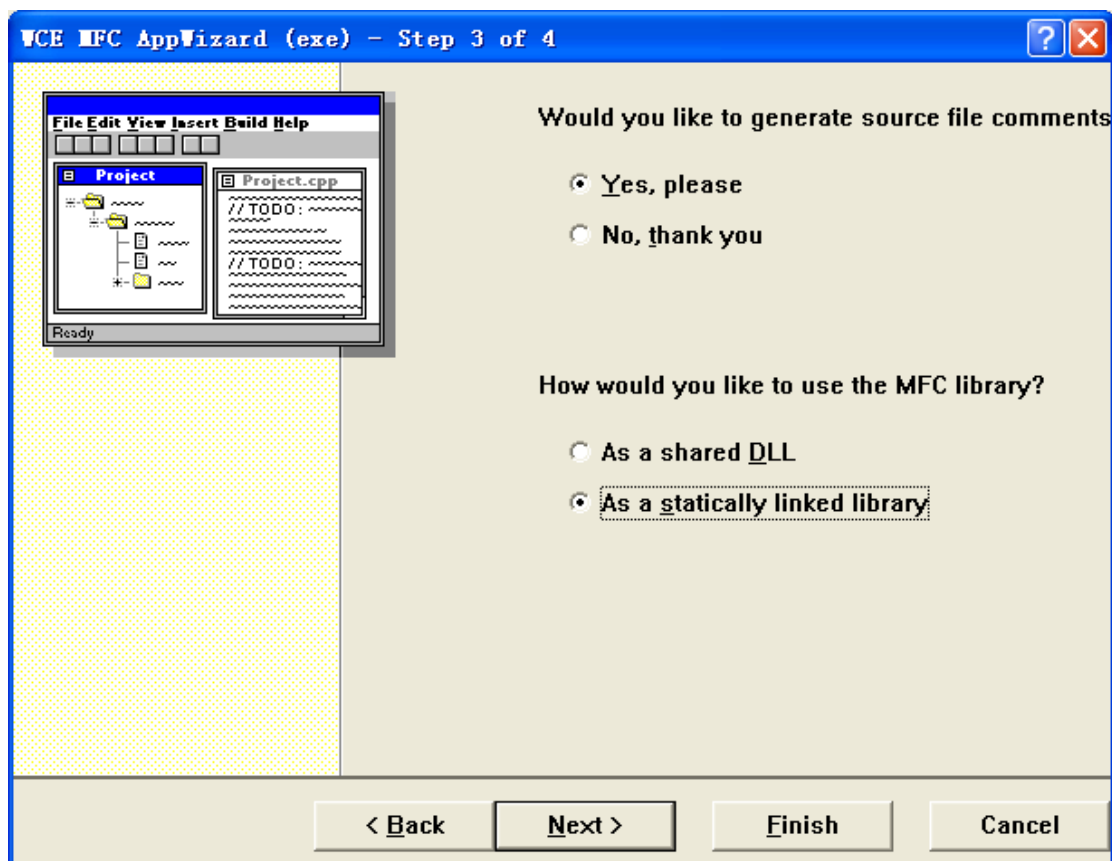


4. 在 WCE MFC AppWizard(exe)- Step 2 of 4 中, 点击“Next”继续, 如下

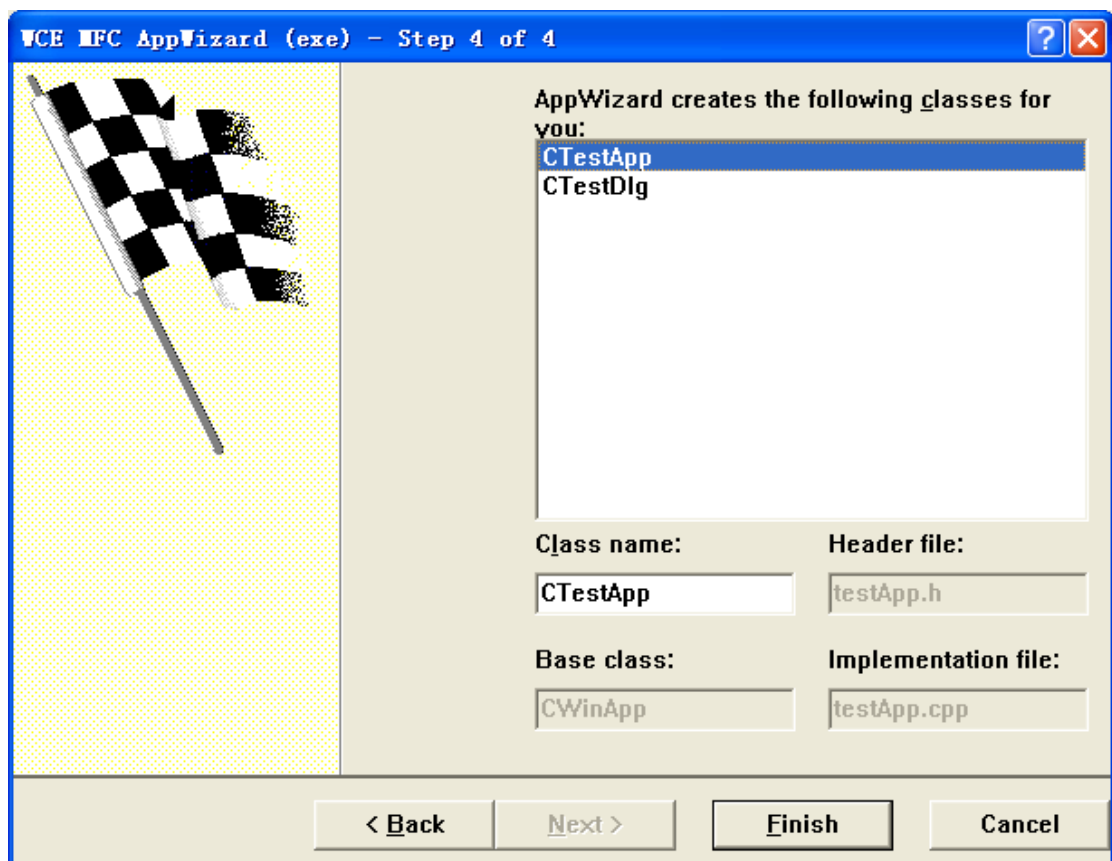
图：



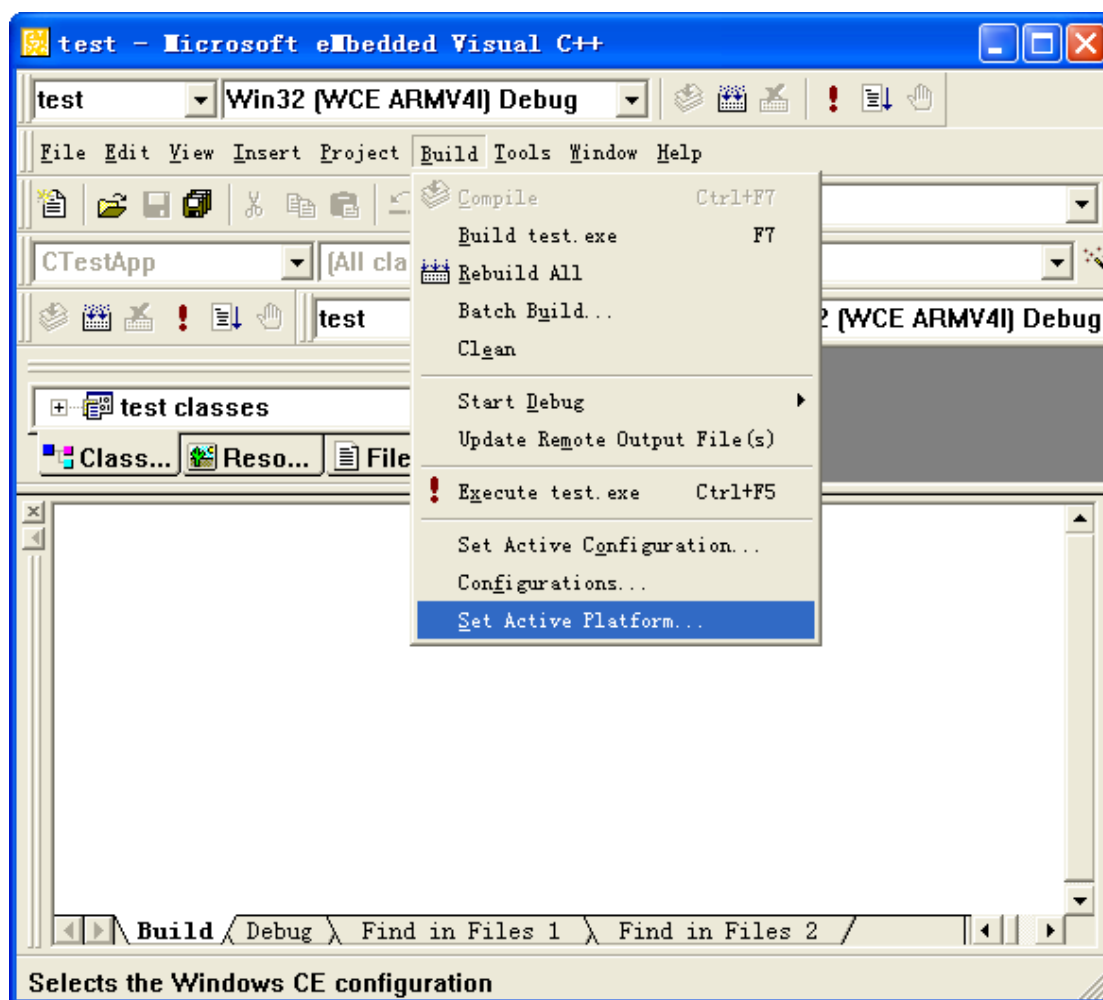
5. 在 WCE MFC AppWizard(exe)- Step 3 of 4 中，选择MFC 静态连接 ， 点击“Next”继续， 如下图：



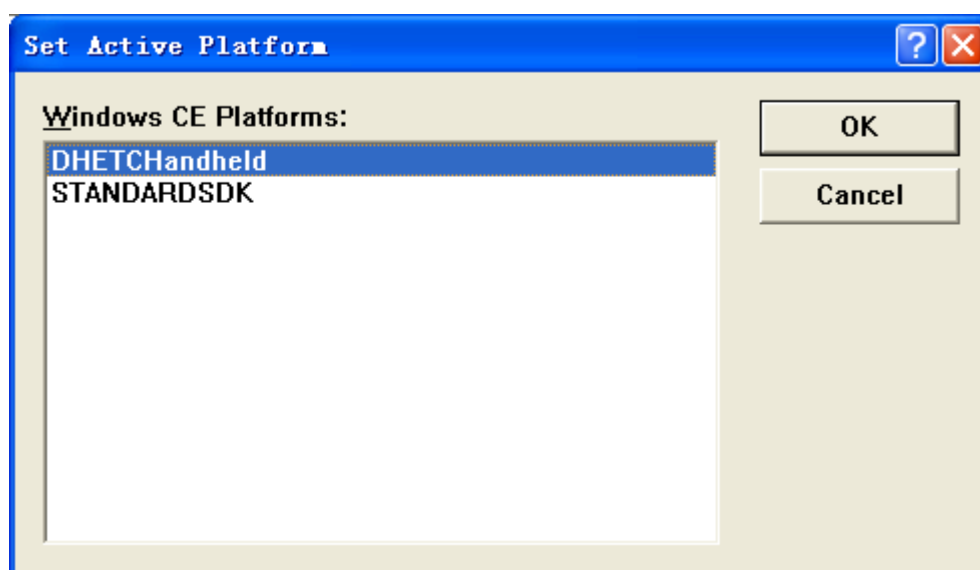
6. 在 WCE MFC AppWizard(exe)- Step 4 of 4 中, 点击“Finish”结束向导



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



8. 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:



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;

PLLChannelID—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—成功，非 0—失败

8. PSAM 通道复位

函数： DHETCResetPSAM()

参数： 无参数

返回： 0—成功，非 0—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 ~ 2

返回： 0—成功，非 0—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—成功，非 0—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：MIRquestEX2(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 一卡通 M1 非接触卡

0x12 国标 CPU 卡

0x13 一卡通 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: MI_Loadkey(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: MI_ReadBlock(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—成功, 非 0—失败

注意: 读的块必须在同一扇区

8. 写 M1 非接触卡块

函数: MI_WriteBlock(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—成功, 非 0—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: MI_Increment(unsigned char Block, long value)

参数: Block — 钱包块号

Value — 增加的值

返回: 0—成功, 非 0—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—成功, 非 0—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—成功, 非 0—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—成功, 非 0—失败