



**立宇泰电子**  
Liyutai Elec. CO., Ltd.

**专注于做最好的嵌入式计算机系统供应商**

## EMUC6Port 多串口板

# 使用说明

(适用于 NEW\_2440\_090209 V1.0)

**Rev 1.0**

**2009年4月28日**

**杭州立宇泰电子有限公司**

HangZhou LiYuTai Elec.Co.,Ltd

## 目 录

版本记录 .....	3
1 产品简介 .....	4
2 硬件规格 .....	4
2.1 核心板 .....	5
2.2 扩展板 .....	5
2.2.1 串行接口 .....	6
2.2.2 以太网接口 .....	7
2.2.3 LCD、触摸屏接口 .....	8
2.2.4 IO 口设备 .....	9
2.2.5 E2PROM 芯片 AT24C04 .....	11
2.2.6 DS3231 芯片,温度传感器和 RTC .....	11
2.2.7 DS18S20 芯片,温度传感器 .....	11
2.2.8 电源与复位 .....	11
3 软件支持 .....	12
3.1 文件说明 .....	12
3.2 设备与驱动说明 .....	12
3.2.1 串口设备 .....	12
3.2.2 IO 口设备 .....	13
3.2.3 IIC 设备 .....	13
3.2.4 DS18S20 设备 .....	14
4 如何开发应用程序 .....	15
4.1 基于 Embedded Visual C++开发环境 .....	15
4.1.1 安装 Embedded Visual C++ .....	15
4.1.2 安装 EVC4SP4 .....	16
4.1.3 安装 SDK .....	17
4.1.4 建立和编译应用程序 .....	21
4.1.5 利用 EVC 进行同步调试 .....	28
4.2 基于 Visual Studio 2005 开发环境 .....	29
4.2.1 安装 VS2005 .....	29
4.2.2 创建 Helloworld 实例 .....	29
4.2.3 建立连接 .....	31
4.2.4 编译和下载 .....	33

## 版本记录

版本号	描述	日期
V1.0	初始版本	2009-4-28

# 1 产品简介

EMUC6PORT 多串口板是一款采用 ARM9 高性能处理器的嵌入式多串口服务器应用型产品。在 WINCE5.0 中文操作系统支持下，用户可以使用多达 6 个独立控制的串行口，波特率最高可以支持 115200bps，并能够确保多个串口同时工作时的实时响应性能。

该产品采用具有工业级温度范围的器件，经过严格的测试可以在 -20~70℃ 环境下持续稳定地工作。同时该板还提供了 1 路 10M 以太网接口，TFT 液晶屏接口，1-Wire 温度传感器 DS18S20，按键和带有光电隔离的 RS485 接口、远红外/近红外接口，输入输出 IO，E2PROM，DS3231 等，资源丰富适用面广。目前 EMUC6PORT 多串口板已经应用在 zigbee 无线多功能抄表集中器产品中。

# 2 硬件规格

规格总表：

核心部分参数		
CPU	采用 ARM9 内核主频 400MHZ 的低功耗处理器，内置硬件看门狗。	
SDRAM	64M Bytes 存储空间，可以升级到 128M Bytes	
Nandflash	64M Bytes 存储空间，可以升级到 1G Bytes	
RTC 实时时钟	实时时钟，采用备份电池保持时间；	
串行口规格参数		
信号类型	TTL/RS232/RS485	
串口数	6 个用户可用串行口，1 路调试用串口。	
保护	RS485 接口提供光电隔离和防雷保护；RS232 接口提供 15KVESD 保护。	
通讯参数	校验	None, Even, Odd, Space, Mark;
	数据位	5, 6, 7, 8
	停止位	1, 2;
	速度	最高可达 115200bps。
FIFO	64 bytes/16bytes 硬件 FIFO	
工作状态	LED 指示灯指示数据发送和接收状态	
接口模式	自定义	
其他接口		
以太网接口	速度	支持 10Mbase-T
	接口方式	RJ45

	保护	2KV 浪涌保护
LCD 接口	可直接驱动 STN 以及 TFT 等液晶显示屏；其中 TFT 屏支持分辨率最高 800 x 600, 64k 色显示；	
USB 接口	1 路主 USB, 1 路从 USB 口, 符合 USB1.1 规范, 支持鼠标键盘, 支持 Activesync;	
电源	+5V 输入供电。整板平均工作功耗 1.5W。	
应用环境		
工作温度	-20-70℃	
存储温度	-40-85℃	
湿度	5%~90% (非凝结)	

## 2.1 核心板

核心系统采用 6 层板工艺, 具有良好的电气抗干扰性能, 采用 2×100 针 1.27mm 间距双排插座与扩展板连接。核心板的资源配置如下:

- **CPU:** 采用三星 S3C2440A 低功耗 32 位处理器, 主频 400MHz, 总线频率 100MHz;
- **SDRAM:** 共 64MB 存储空间, 其中用户可用 30.4MB 空间。
- **NANDFlash:** 共 64MB 存储空间, 其中用户可用 43.6MB 空间。

## 2.2 扩展板

扩展板采用 4 层板工艺。尺寸结构、引出接口完全按照客户定制要求设计。

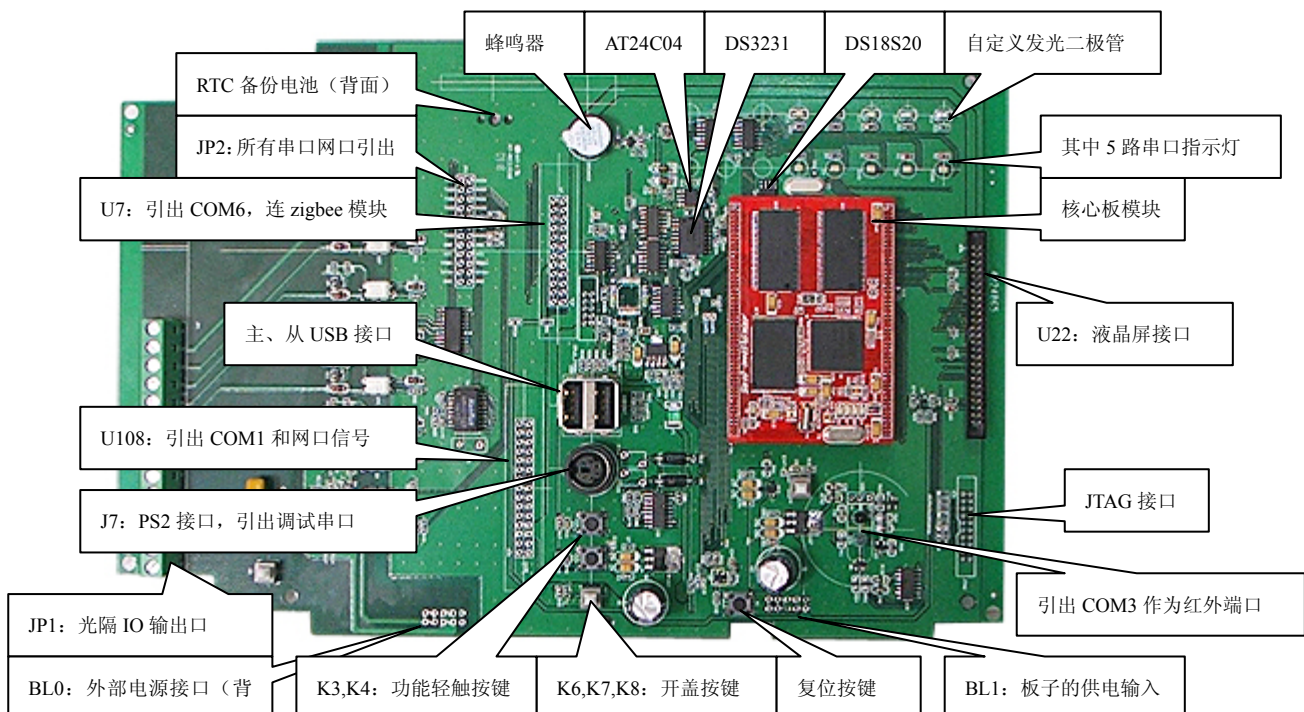
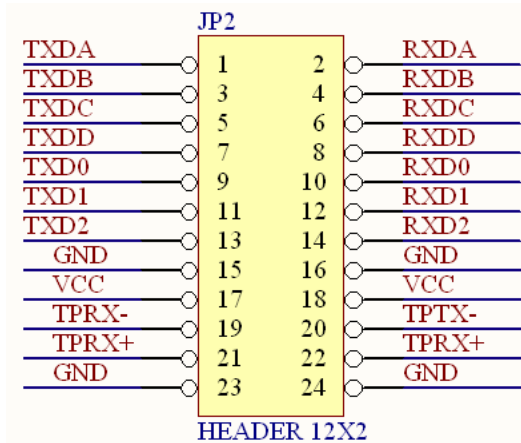


图 1: 扩展板上接口和电路布局

## 2.2.1 串行接口

EMUC6PORT 定制板一共提供了 6 路用户可用的串行口, 设备名为 COM1, COM3, COM6, COM7, COM8, COM9。它们从以下接口处引出:

- JP2: 上面引出了所有串口的发送和接收 TTL 信号对, 以及网络接口信号。对应关系为:



- COM1: TXD0,RXD0;
- COM2: TXD1,RXD1 (作为调试串口, 用户不可用);
- COM3: TXD2,RXD2;
- COM6: TXDA,RXDA;
- COM7: TXDB,RXDB;
- COM8: TXDC,RXDC;
- COM9: TXDD,RXDD。

- U7 上面引出了 COM6 的 TTL 信号, 相关引脚:  
Pin1, 2: TXDA;  
Pin3, 4: RXDA。
- U108 上面引出了 COM1 的 TTL 信号, 相关引脚:  
Pin1: RXD0;  
Pin3: TXD0。
- J7 引出了调试串口的 RS232 信号, 相关引脚:  
Pin3, 5: TXD;  
Pin4, 6: RXD。
- 红外部分电路使用了串口 COM3, 并工作在红外模式下 (符合 IrDA1.0 标准)。
- JP1 接口引出了 COM7 的 RS485 信号, 相关引脚:  
Pin13: RS485-A;  
Pin14: RS485-B。

下面来描述这 6 路串口的特性参数。

### 2.2.1.1 COM1~COM3 特性参数

它们都同时满足以下特性参数:

- 硬件: 处理器内置控制器;  
--I/O物理地址: 0x5000-0000~0x5000-40000~0x5000-80000;  
--FIFO: 64 byte 硬件 FIFO
- 波特率: 最高115200bps;
- 流向控制: 电路上支持无流控;
- 设备名: COM1~COM3;
- 信号线包括: TXD, RXD, GND;
- 传输模式: 全双工;

- 信号类型：LVTTL电平。
- 引出接口：JP2，U108，J7，红外部分电路。

### 2.2.1.2 COM6~COM9 特性参数

它们都同时满足以下特性参数：

- 硬件：采用串行口控制器16C554；
- I/O物理地址：

串口号	起始地址
A (COM6)	0x29000000
B (COM7)	0x28000000
C (COM8)	0x21000000
D (COM9)	0x20000000

- FIFO：16 byte 硬件FIFO；
- 波特率：最高**115200bps**；
- 数据位：5,6,7,8；
- 停止位：1,2；
- 奇偶校验位：even, odd, none, mark, space；

### 2.2.1.3 串口指示灯

板上提供5路串口指示灯，可以指示串口数据通讯情况。红灯亮表示有数据发送，绿灯亮表示有数据接收。指示灯和串口的对应关系：

串口号	指示灯标号
COM1	D100
COM2	D200
COM3	D3
COM6	D4
COM7	D6

TXD 和 RXD 线上如果有低电平，则指示灯被点亮。

## 2.2.2 以太网接口

### 2.2.2.1 技术参数

- 接口类型：10 Base-T
- 接口规程：符合IEEE-802.3
- 接口速率：10M

- 物理起始地址：0x18000000

### 2.2.2.2 接口定义

JP2和U108两个接口座上都引出了网口的差分信号，这些信号可以直接连接到RJ45座，接口定义如下：

JP2接口定义	
19	TPRX-
20	TPRX+
21	TPTX-
22	TPTX+
U108接口定义	
27	TPTX+
28	TPTX-
29	TPRX+
30	TPRX-

### 2.2.3 LCD、触摸屏接口

U22 接口上包含了 LCD 接口信号，触摸屏接口信号，下面分别介绍：

- 可以支持TFT型LCD屏：
  - 支持16-bpp, 64k色TFT屏；
  - 支持多种屏幕大小：  
典型的实际屏幕大小：800×600, 640×480, 320×240, 及其它规格
- 可以支持4线电阻式触摸屏。
- 引脚排布：在U22上提供了信号。引脚定义如下：

U22 (50pins)			
1	+5.0V	2	NC
3	+5.0V	4	GND
5	nRESET	6	VD0
7	VD1	8	VD2 (B0)
9	VD3(B1)	10	VD4(B2)
11	VD5(B3)	12	VD6(B4)
13	VD7(B5)	14	VD8
15	VD9	16	VD10(G0)

17	VD11(G1)	18	GND
19	VD12(G2)	20	VD13(G3)
21	VD14(G4)	22	VD15(G5)
23	VD16	24	VD17
25	VD18(R0)	26	VD19(R1)
27	VD20(R2)	28	VD21(R3)
29	VD22(R4)	30	VD23(R5)
31	GND	32	LCD_PWREN
33	GND	34	GND
35	GND	36	VM(Dateenb)
37	VFRAME (Vsync)	38	VLINE (Hsync)
39	VCLK	40	接 0 欧电阻到 LCD_VDD33
41	nDIS_OFF	42	接 0 欧电阻到 LCD_VDD33
43	TSYP	44	接 0 欧电阻到 LCD_VDD33
45	TSXP	46	接 0 欧电阻到 LCD_VDD33
47	TSYM	48	接 0 欧电阻到 LCD_VDD33
49	TSXM	50	GND

## 2.2.4 IO 口设备

属于 IO 口设备的列举如下，

- 1) 蜂鸣器 (单向输出);
- 2) 4 路 LED (单向输出);
- 3) 5 路 IO 口 (单向输出);
- 4) 5 路按键 (单向输入);
- 5) 电压检测 (单向输入);

Zigbee 和 GPRS 相关的 IO 口，提供地址由应用程序直接控制。

下面分别介绍。

### 2.2.4.1 蜂鸣器

电平控制的蜂鸣器，可以通过发送命令字给驱动程序，来控制它的响与不响。具体使用方法请参考 3.2 节。

### 2.2.4.2 四路 LED

D10,D11,D13,D15 是用户自定义的指示灯，可以通过发送命令字给驱动程序，来灵活控制这几盏灯的点亮和熄灭。具体使用方法请参考 3.2 节。

### 2.2.4.3 五路输出口

JP1 上提供了 5 路独立的 IO 口，方向固定为输出。电路上对这 5 路 IO 实施了光电隔离。可以通过发送命令字给驱动程序，来灵活控制这几盏灯的点亮和熄灭。具体使用方法请参考 3.2 节。

引脚号	定义
Pin1	光隔输出端电源
Pin2	光隔输出端地
Pin3	光隔 IO 输出 1
Pin4	光隔 IO 输出 2
Pin5	光隔 IO 输出 3
Pin6	光隔 IO 输出 4
Pin7	光隔 IO 输出 5

### 2.2.4.4 五路功能按键口

提供了 5 路功能按键，可以通过发送命令字给驱动程序，来读取按键的状态（按下或释放）。具体使用方法请参考 3.2 节。按键排序如下：

按键编号	对应按键
1	K4
2	K3
3	K8
4	K6
5	K7

### 2.2.4.5 低电压检测

提供 1 路低电压检测。可以通过发送命令字给驱动程序，来读取低电压检测的输出状态（输出 0，电压过低；输出 1，电压正常）。具体使用方法请参考 3.2 节。

### 2.2.4.6 其他 IO 口（应用层直接操作）

这些 IO 口的网络标号为 CAMDATA0~CAMDATA7，它们对应的 IO 口标号为：GPJ0~GPJ7，相关寄存器的物理地址为：

GPJCON 0x560000d0 端口配置寄存器，复位值为：0x0  
GPJDAT 0x560000d4 端口数据寄存器，复位值为：Undef.  
GPJUP 0x560000d8 上拉寄存器，复位值为：0x0000

## 2.2.5 E2PROM 芯片 AT24C04

板上提供了 1 路 E2PROM 设备：AT24C04，这是一片 512 字节大小的 E2PROM。可以通过发送命令字给驱动程序，来读取或写入任意一个存储空间的数据。具体使用方法请参考 3.2 节。

## 2.2.6 DS3231 芯片,温度传感器和 RTC

可以通过发送命令字给驱动程序，来读取或写入该芯片上的任意一个存储空间的数据。具体使用方法请参考 3.2 节。DS3231 通过板上的纽扣电池来保持 RTC 数据。

## 2.2.7 DS18S20 芯片,温度传感器

可以通过发送命令字给驱动程序，来实现温度数据的读取。具体使用方法请参考 3.2 节。

## 2.2.8 电源与复位

### 2.2.8.1 电源接口

**供电要求：**+5VDC 输入；建议采用+5V/1A 输出能力的电源供电。

**电源指示灯：**D16 为电源指示灯，当 BL1 上的+5V 电源接入，该指示灯点亮。

**整板功耗：**静态工作电流 300mA (@5V)，启动峰值电流 500mA (@5V)；

## 2.2.8.2 复位按键-RESET

采用专用复位集成电路设计，确保系统上电复位的可靠性，低电压自动复位，同时支持手动复位。如图1中的RESET即为手动复位键。

# 3 软件支持

## 3.1 文件说明

相关的软件工具文件，位于光盘目录下：

目录	文件/目录	描述
应用程序实例源码	SerialPort.rar	串口通讯实例
	Ds18s20.rar	DS18S20测试实例
	Iotest.rar	IO口设备测试实例, 包括蜂鸣器,LED,按键,低压检测
	IICtest.rar	RTC和E2PROM的测试实例
SDK	EMUC6PORT.m si	应用程序开发使用
USB驱动和同步软件	包括USB同步所需要的USB驱动和ActiveSync软件	
应用程序开发工具	EVC4.0	交叉编译工具

## 3.2 设备与驱动说明

### 3.2.1 串口设备

EMUC6PORT 提供 6 路用户可用的标准串行口，对应设备名分别如下：

对应硬件接口	设备名	描述
UART0	COM1:	普通串口
UART1	COM2:	调试口(目前调试程序用的)
UART2	COM3:	红外模式
16C554A	COM6:	普通串口
16C554B	COM7:	普通串口
16C554C	COM8:	普通串口
16C554D	COM9:	普通串口

应用程注意事项：

使用标准串口 API 编程即可

#### 测试程序使用方法说明：

自发自收测试：

示例：用跳线帽接上 TXDA 和 RXDA，打开测试软件，在设置项里选择 COM6，然后打开端口并点击接收；输入要发送的数据，点击发送，然后观察接收区是否正确收到。

### 3.2.2 IO 口设备

对应硬件设备	设备名	描述
蜂鸣器	LED1:	驱动支持蜂鸣器的响与不响控制
4 路 LED	LED1:	驱动支持对 4 盏灯状态的写入控制
5 路 IO 口	LED1:	驱动支持对 5 路 IO 的写入控制
5 路按键	LED1:	驱动支持对 5 路按键的状态读取
电压检测	LED1:	驱动支持对电压检测脚状态的读取

#### 驱动使用说明：

打开 IO 设备示例：

```
HANDLE dev =
```

```
  CreateFile(_T("LED1:"), GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0, NULL);
```

采用 DeviceIOControl 函数实现对 IO 口设备控制：

命令代码：

```
#define IOCTL_BUZZER_ON  0x300 //响蜂鸣器
#define IOCTL_BUZZER_OFF 0x301 //关蜂鸣器
#define IOCTL_SET_LED    0x302 //设置LED灯 0表示点亮，1表示熄灭
#define IOCTL_SET_IO     0x303 //设置5路IO口：GPG7，GPB1，GPB9，GPB10，GPB6
#define IOCTL_READ_KEY   0x304 //读取五个按键状态
#define IOCTL_READ_PWRD  0x305 //读取电压
```

示例：DeviceIoControl(dev, IOCTL\_BUZZER\_ON, NULL, NULL, NULL, NULL, NULL, NULL);

#### 测试程序使用方法说明：

蜂鸣器测试中用一个按钮控制蜂鸣器响和不响；IO测试，输入值，点击设置，设置成功则弹出提示对话框；LED测试，点击设置，默认0值会把四盏灯都点亮。后面两项，点击读取，编辑框里会显示状态。

### 3.2.3 IIC 设备

对应硬件设备	设备名	描述
AT24C04	RAE1:	E2PROM
DS3231	RAE1:	RTC

#### 驱动使用说明

定义了一个结构用来写入或读出数据:

```
typedef struct
{
    WORD   Addr; //地址: RTC 0~18, E2PROM 0~511
    BYTE   Data; //读写或写入的单字节数据
}UNIT;
命令代码:
#define IOCTL_RTC_READ 0x0
#define IOCTL_RTC_WRITE 0x1
#define IOCTL_E2PROM_READ 0x2
#define IOCTL_E2PROM_WRITE 0x3
```

**测试程序使用方法说明:**

点击读出, 芯片存储空间里的所有数据会从调试口打印出来。点击写入, 会向 RTC 写入 0x0 到 0x12 数据; 会向 E2PROM 写入 0x0 到 0xff,0xff 再到 0x0 的数据。清零则把芯片存储空间里的数据清空。

### 3.2.4 DS18S20 设备

对应硬件设备	设备名	描述
DS18S20	DTM1:	温度计

**驱动使用说明:**

定义了一个结构用来读出数据:

```
typedef struct _TmpStruct
{
    UINT8   subzero; // 正或负
    UINT8   cel; //温度的整数部分
    UINT8   cel_frac_bits; //温度的小数部分
} TmpStruct, *PTmpStruct;
```

命令代码:

```
#define IOCTL_CHIP_EXIST 0x401 //检测 DS18S20 是否已正常开启
#define IOCTL_CONVERT_T 0x402 //读取温度值
```

示例:

```
TmpStruct temperature_st;
DeviceIoControl(dev,IOCTL_CONVERT_T,NULL,NULL,&temperature_st,sizeof(TmpStruct),&readbytes,NULL);
```

**注意事项**

- 1) DS18S20 转换一次温度需要 750mS, 因此不可过快读取温度;
- 2) 如果读取错误 (返回错误号), 请重试。

**测试程序使用方法说明:**

点击读取, 若读取成功, 会显示在上面的编辑框, 也可通过调试口, 察看打印消息。

## 4 如何开发应用程序

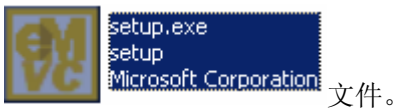
开发应用程序的过程一般分两步，第一步在PC机开发环境下设计和编译应用程序，第二步将它下载到目标系统，也就是我们的EMUC6PORT板子上运行、调试。实现第二步的前提是您已经按照第四章的要求，建立好了板子和PC机之间的同步连接。

WinCE5.0应用程序主要有两种常用开发环境，一种是Embedded Visual C++ (EVC)，另一种是用Visual Studio.net 2005。本章我们主要讲述用EVC开发的一般方法。下一章节讲述采用VS2005进行开发的方法。

### 4.1 基于 Embedded Visual C++开发环境

#### 4.1.1 安装 Embedded Visual C++

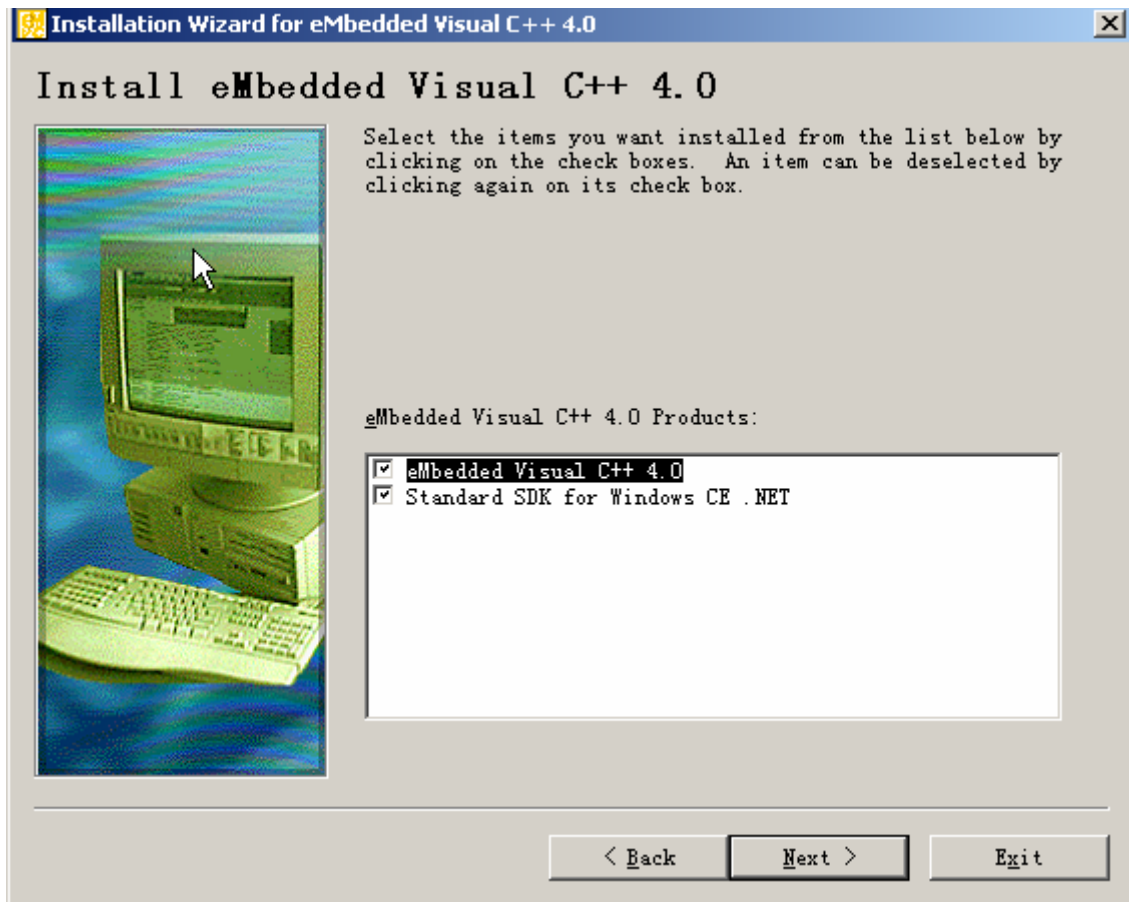
进入光盘“工具软件”目录，点击 EVC4.EXE 解压到硬盘中安装。在解压的文件夹中双击



文件。



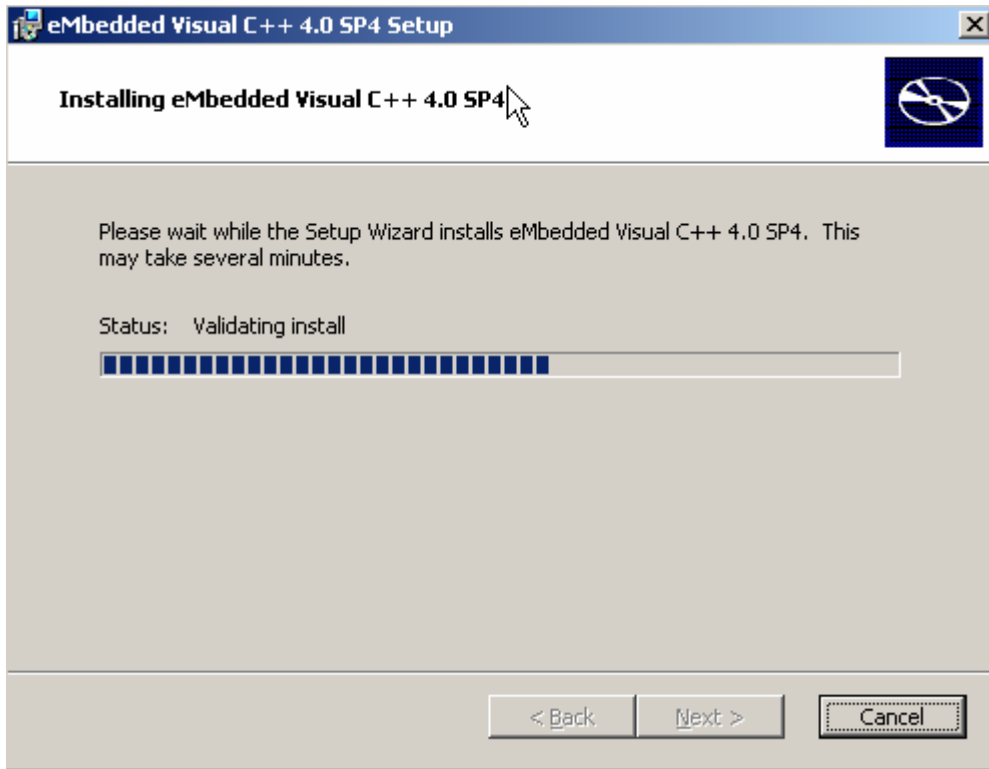
在 Installation Wizard for eMbedded Visual C++ 4.0 的对话框中选中 Embedded Visual C++ 4.0 和 Standard SDK for Windows CE.NET，如下图所示：



再安装提示默认安装即可完成。注意在安装EVC4后，还必须要安装EVC4SP4。

## 4.1.2 安装 EVC4SP4

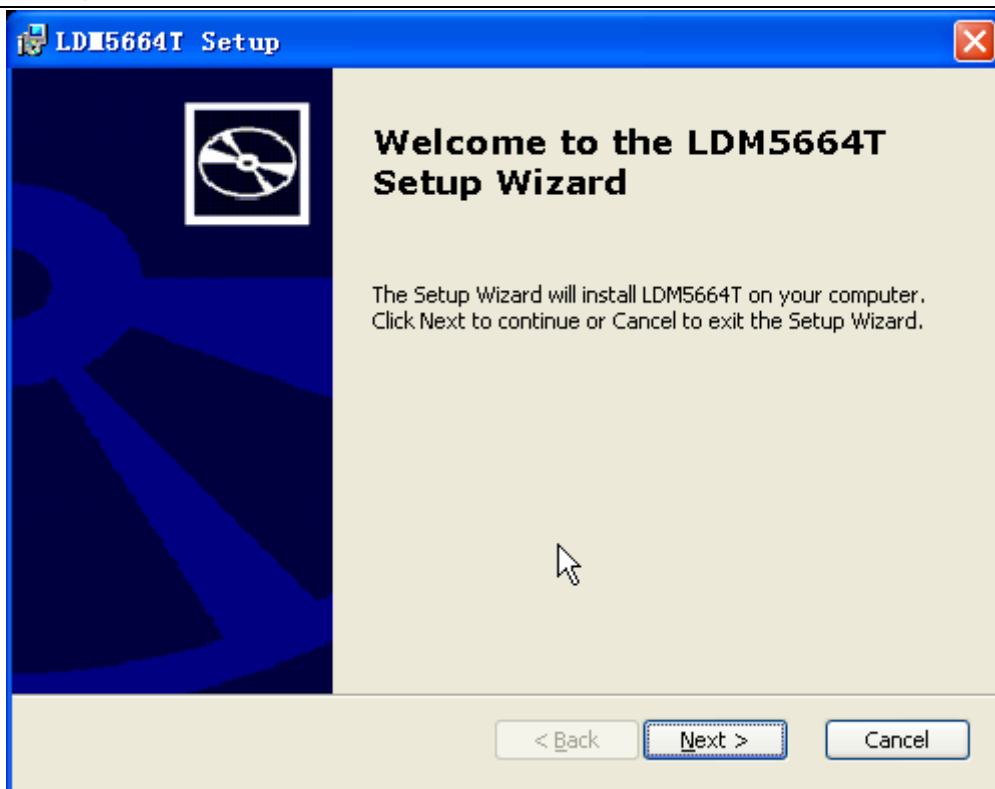
进入光盘“工具软件”目录，点击 EVC4SP4.EXE 解压到硬盘中安装。在解压的文件夹中双击 Setup.exe 文件。按照默认提示安装即可完成。



### 4.1.3 安装 SDK

EMUC6PORT 板子的 SDK 位于光盘的 SDK 目录下。

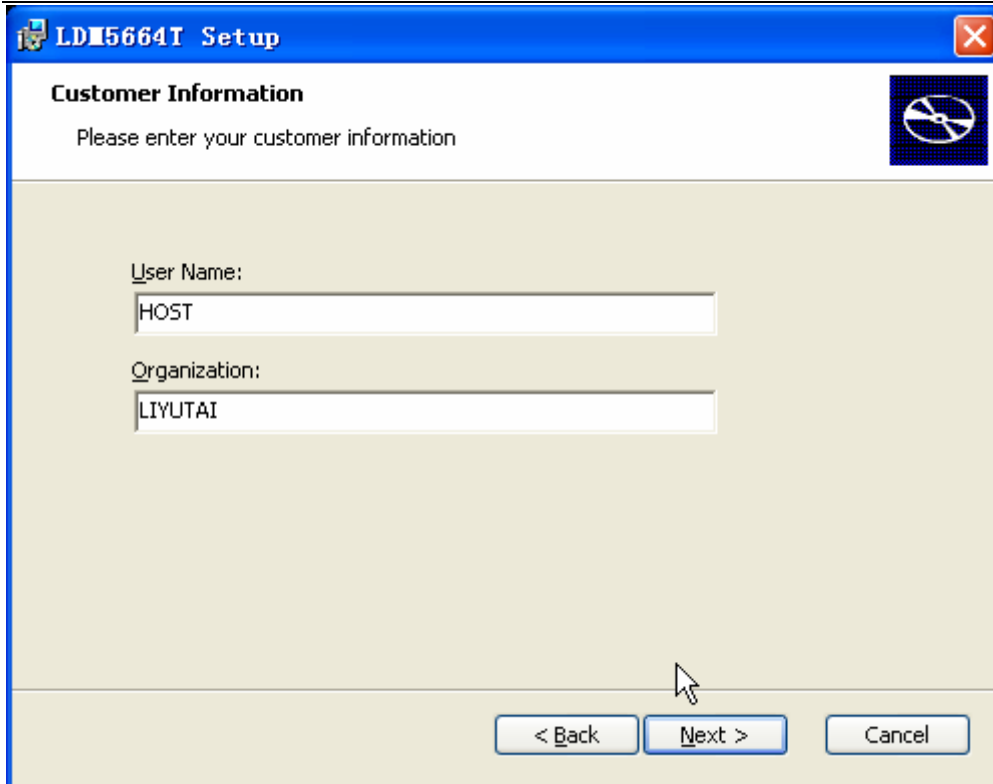
(1) 下面来安装这个SDK。双击EMUC6PORT.msi，出现以下对话框：



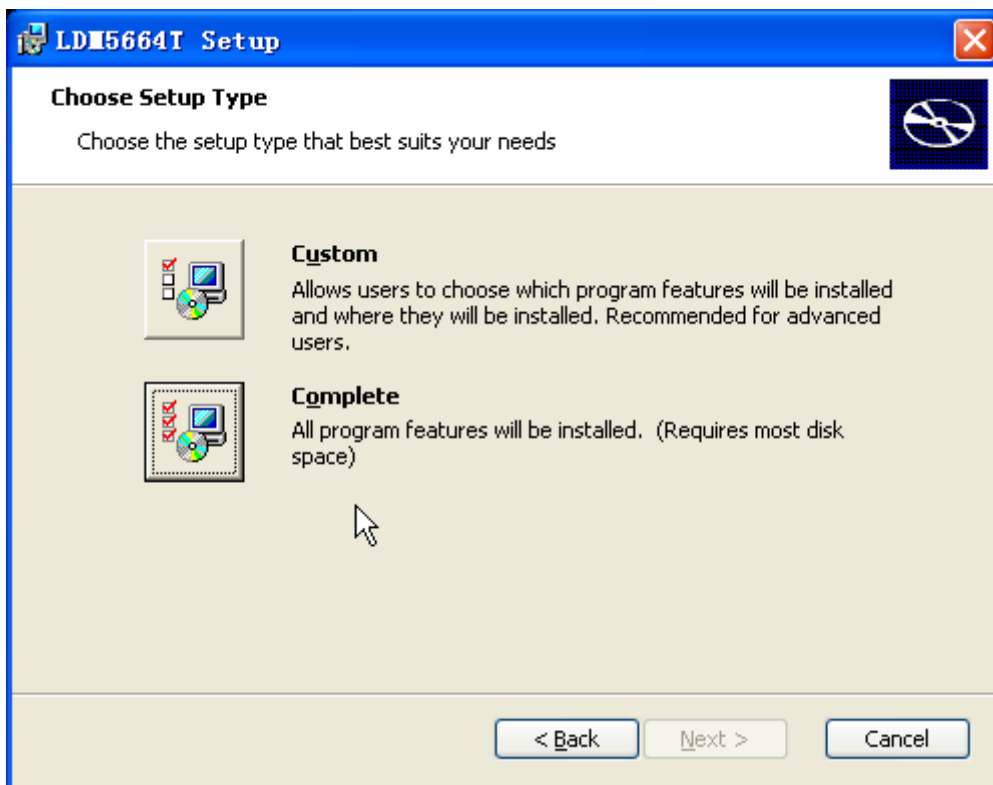
(2) 点击“Next>”，选中“ accept”， 点击“Next>”继续：



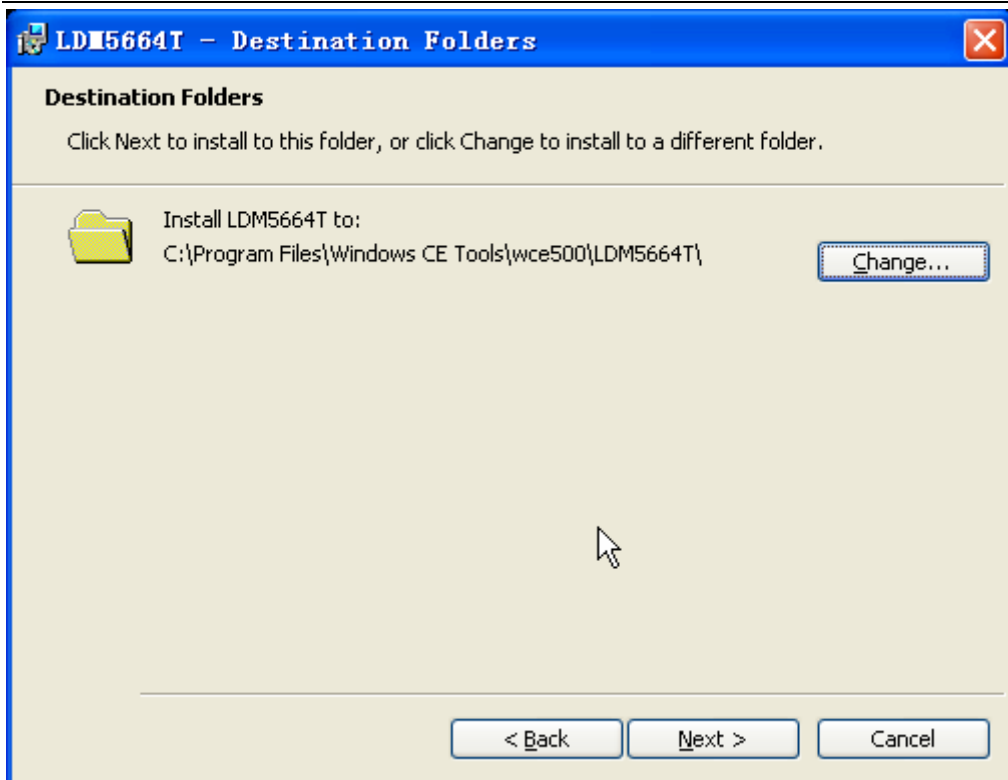
(3) 在Customer Information中，填入你的信息，点击“Next>”继续。



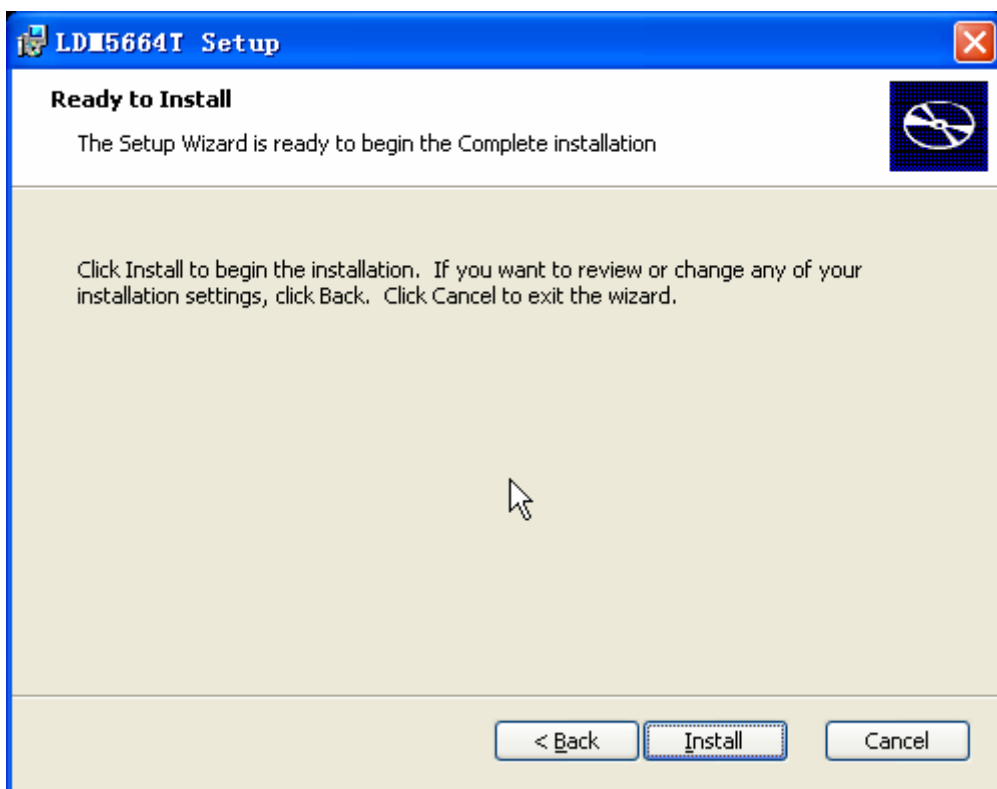
(4) 在Choose Setup type对话框中，直接点击“Complete”进行安装。



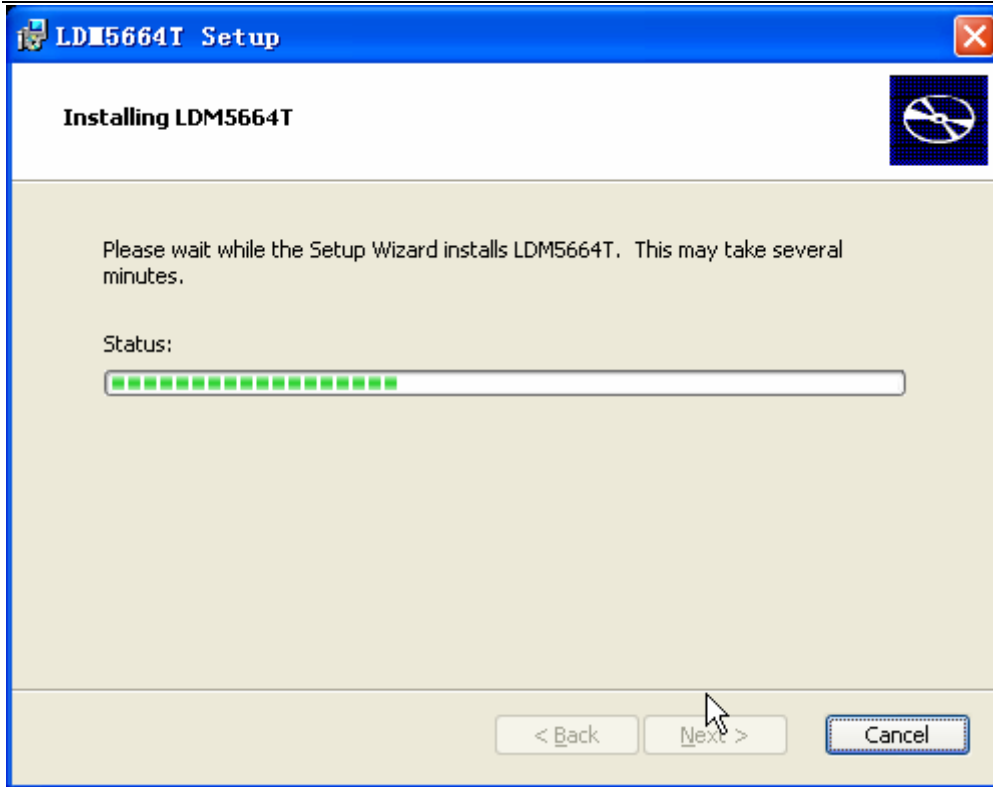
(5) 选取你要安装的目录，点击“Next>”进行安装：



(6) 点击“Install”：



(7) 开始安装EMUC6PORT SDK，如下图所示：

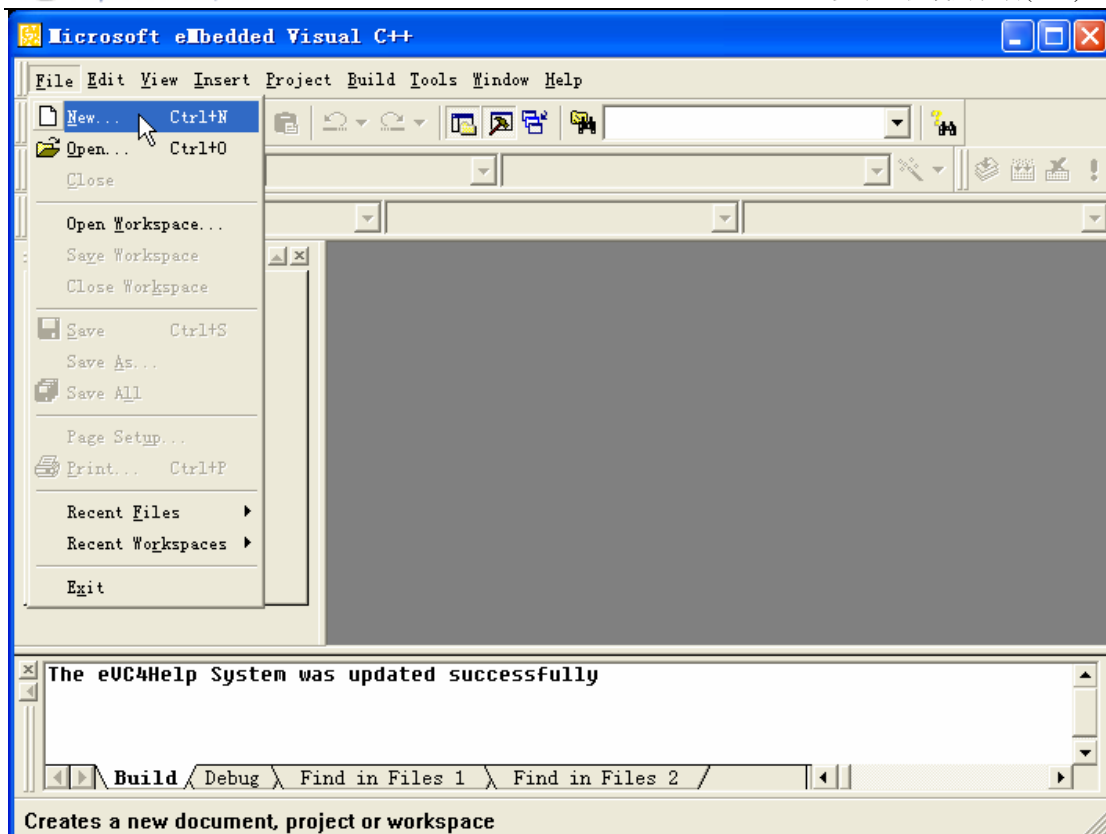


(8) 点击“Finish”安装完成。

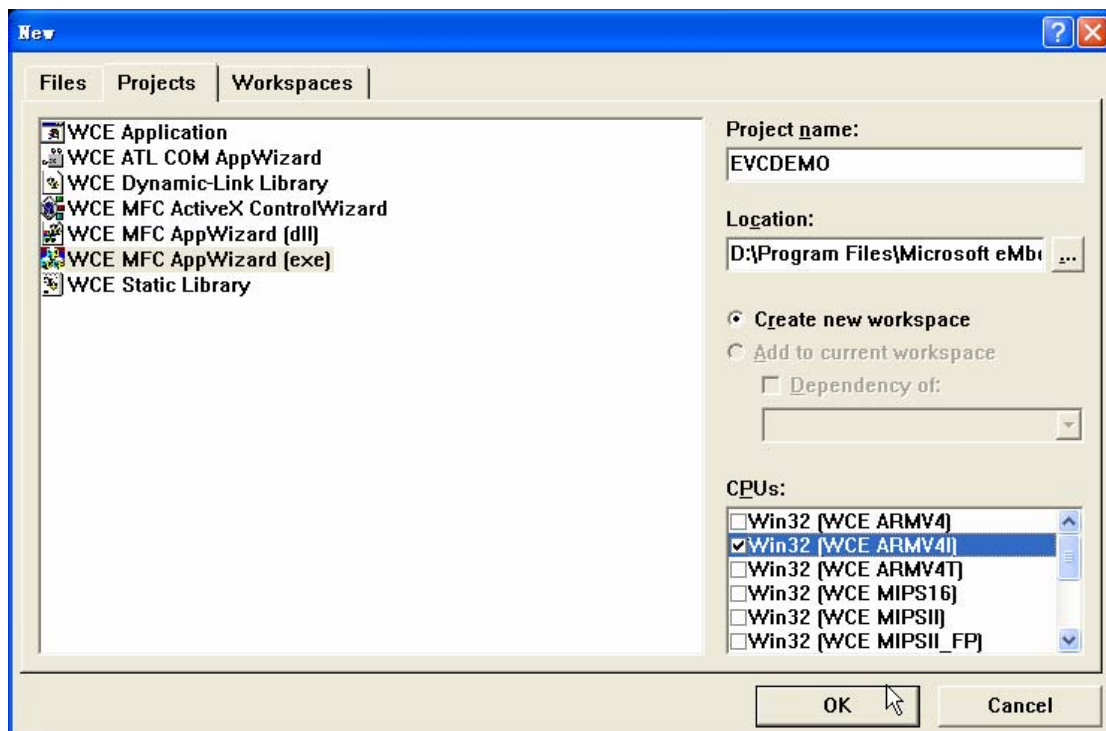
## 4.1.4 建立和编译应用程序

### 4.1.4.1 新建工程

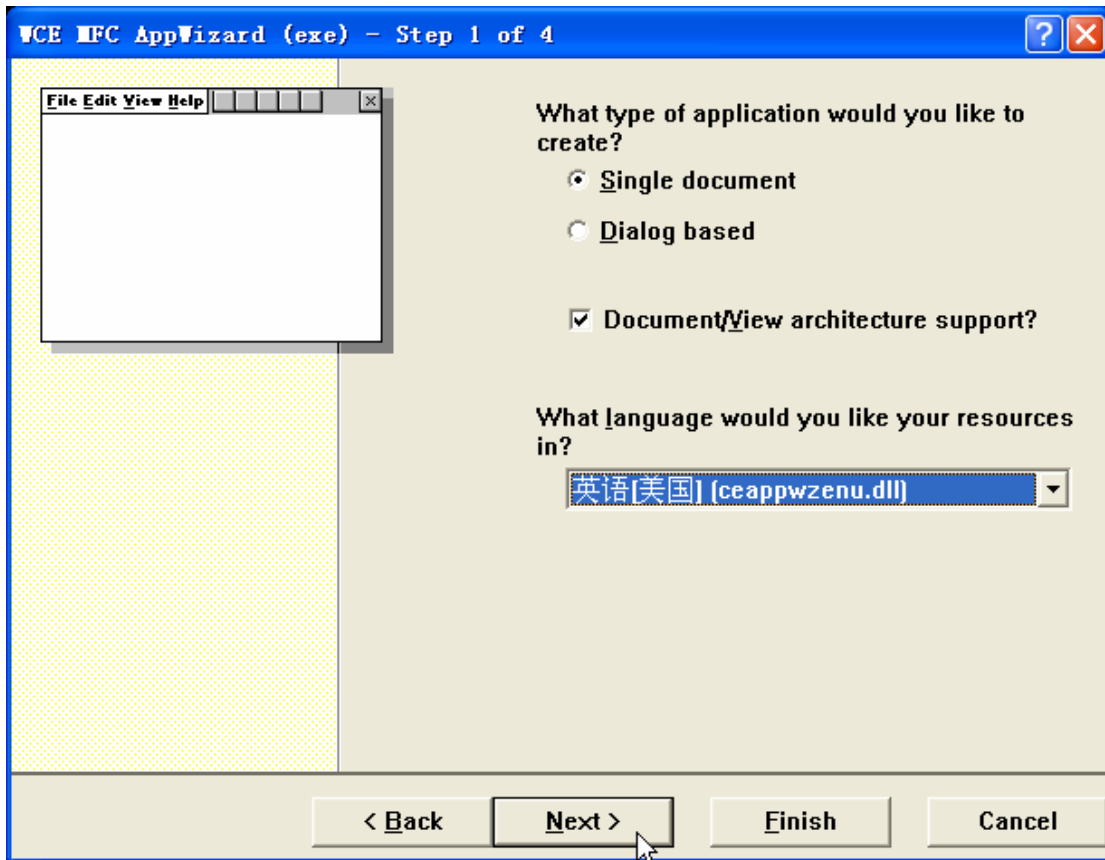
(1) 运行 Embedded Visual C++4.0 集成开发环境，建立一个新的工程。点击“File”菜单下的“New”，如下图所示：



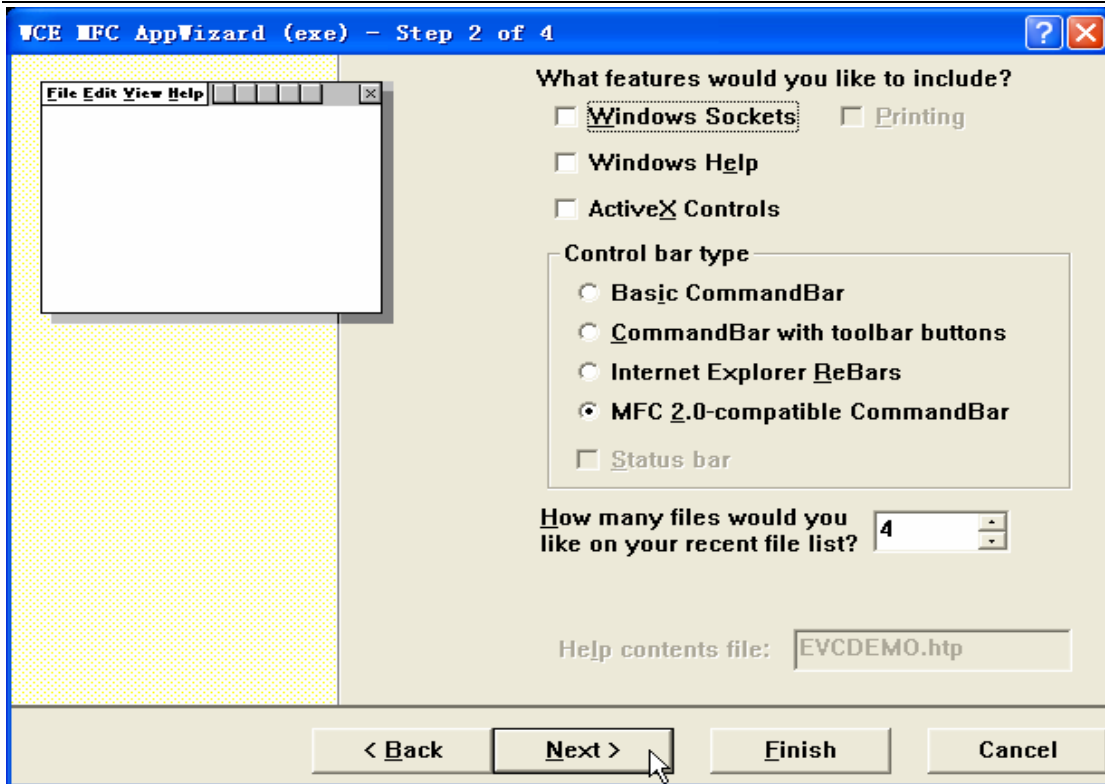
(2) 新建对话框中选择“Project”标签页，选中“WCE MFC AppWizard (exe)”项目，并在 Project name: 后输入工程名称，在 Location: 中设置工程所在路径。在 CPUs 栏目中选择“Win32 (WCE ARMV4I)” 点击 OK 按钮继续。



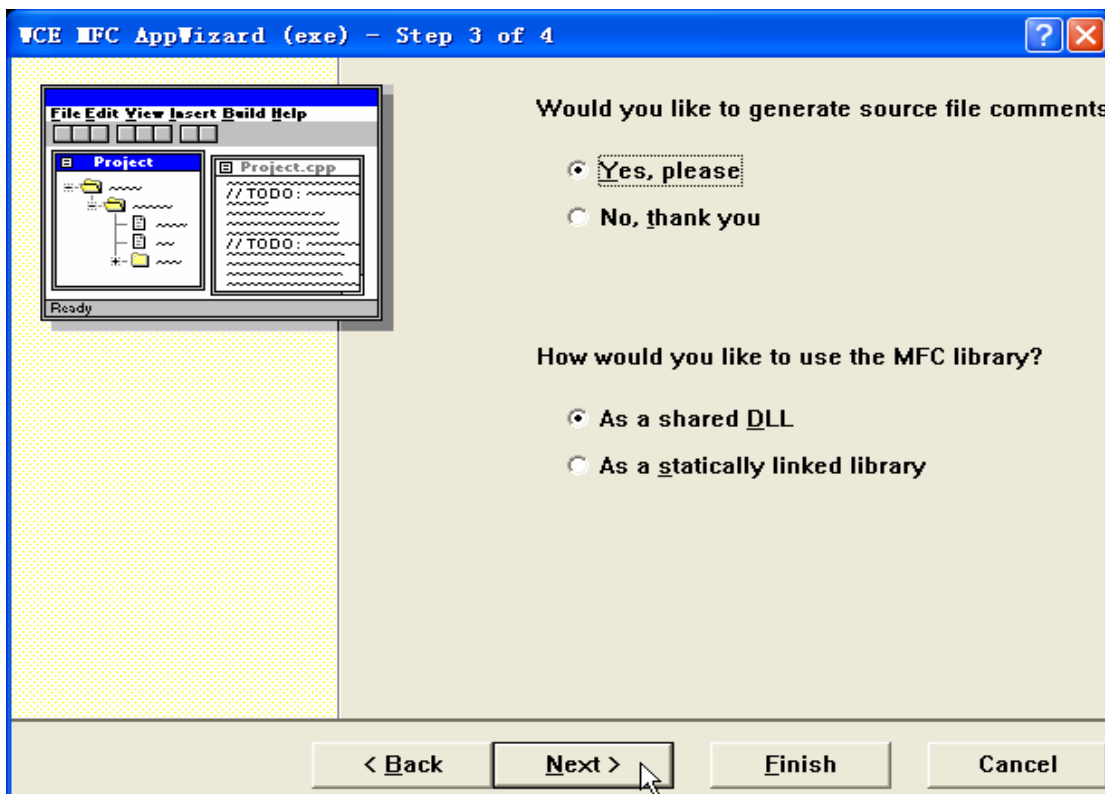
(3) 语言设置为“英语”，其他保持默认。点“Next>”继续。



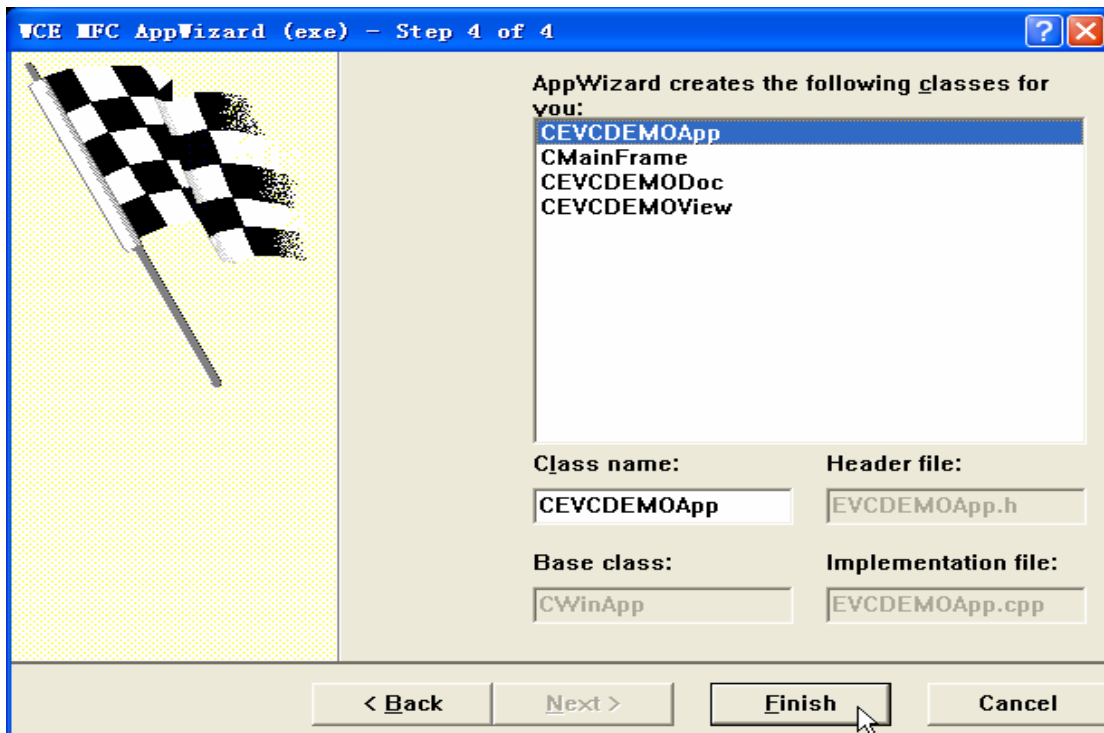
(4) 保持默认值并点击“Next>”继续：



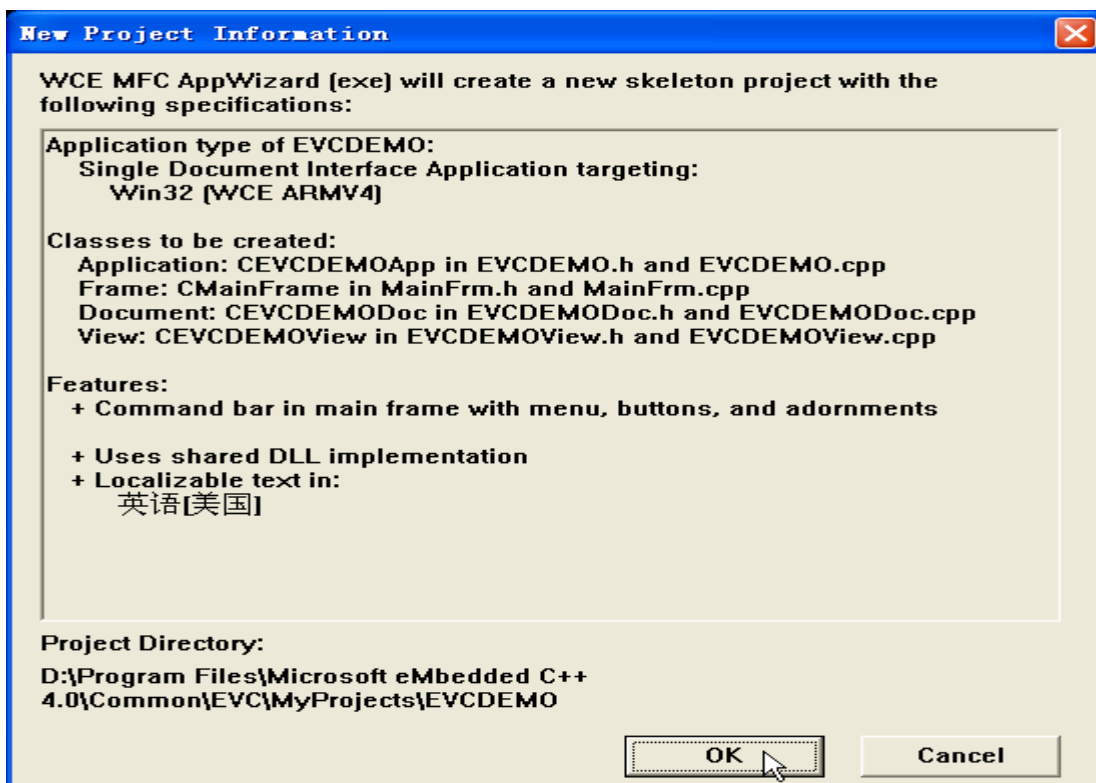
(5) 保持默认值并点击“Next>”继续:



(6) 继续保持默认值并点击“Finish”继续:



结果如下图所示:



#### 4.1.4.2 与 SDK 关联

(1) 选择“Build”菜单中的“Set Active Platform...”:

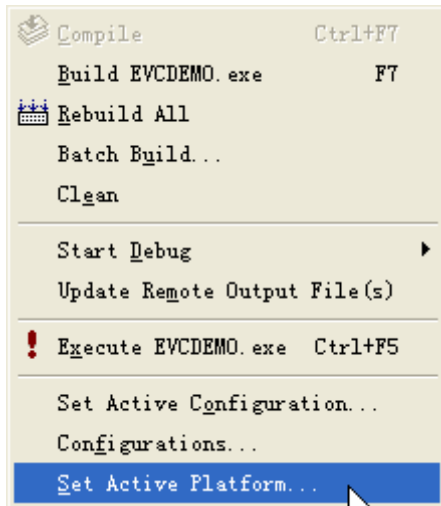
杭州立字泰电子有限公司  
56763526

网址: [www.hzlitai.com.cn](http://www.hzlitai.com.cn)  
[www.armsystem.com.cn](http://www.armsystem.com.cn)

第 25 页 共 34 页

电话: 0571-56763523

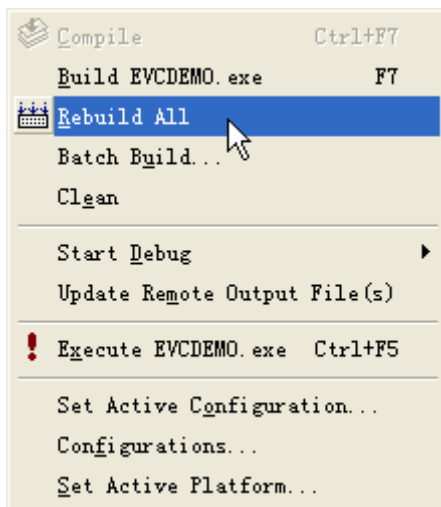
传真: 0571-56763523/25-808



(2) 选择 EMUC6PORT 作为激活的 Platform，并点击 OK。

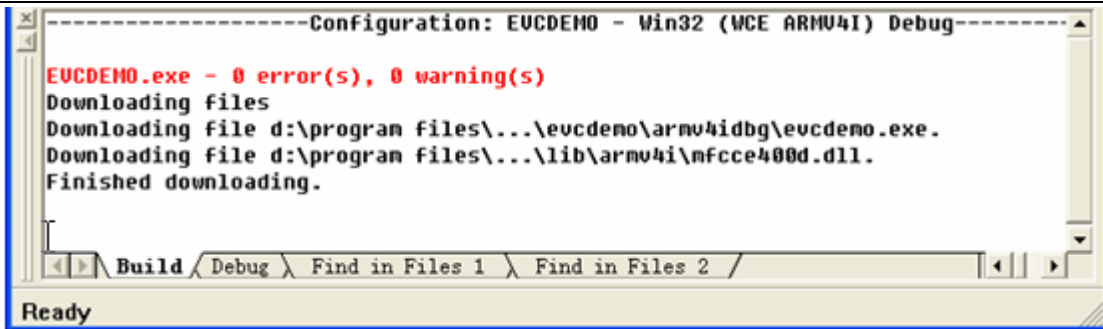
#### 4.1.4.3 编译并下载

(1) 选择“Build”菜单下的“Rebuild All”，如下图所示：

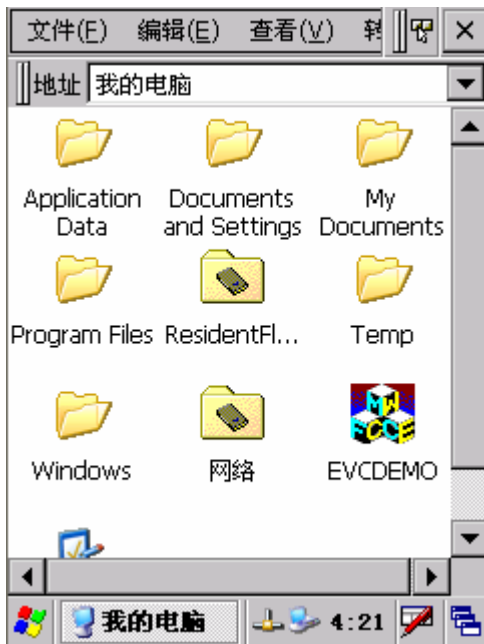


这时系统将会交叉编译您的工程。

(2) 如果编译成功，会有一个自动下载的过程，要求此时你连接好了 **USB 线**、同时激活了 **EMUC6PORT** 与 **PC 机上 Microsoft ActiveSync** 软件之间的同步连接。如果一切顺利，可以看到下载成功的显示：

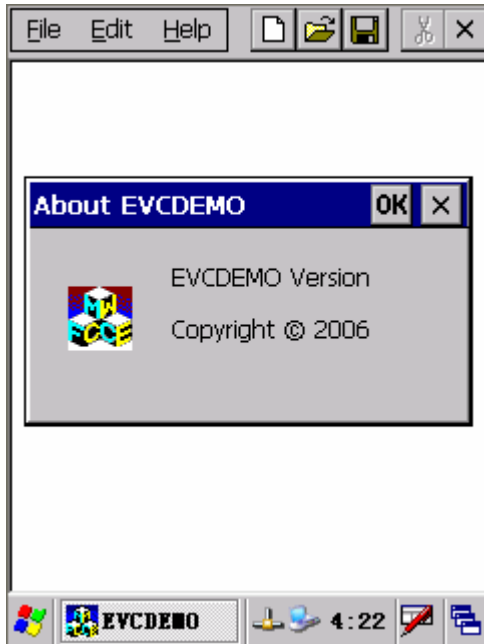


(3)“Finished downloading”表示此时应用程序已经下载到 EMUC6PORT 板子，打开 WinCE 中“我的设备”，可以看到我们建立的应用程序“EVCDEMO”：



#### 4.1.4.4 运行

(1) 双击 EVCDEMO 图标，即可观察到程序运行状况：

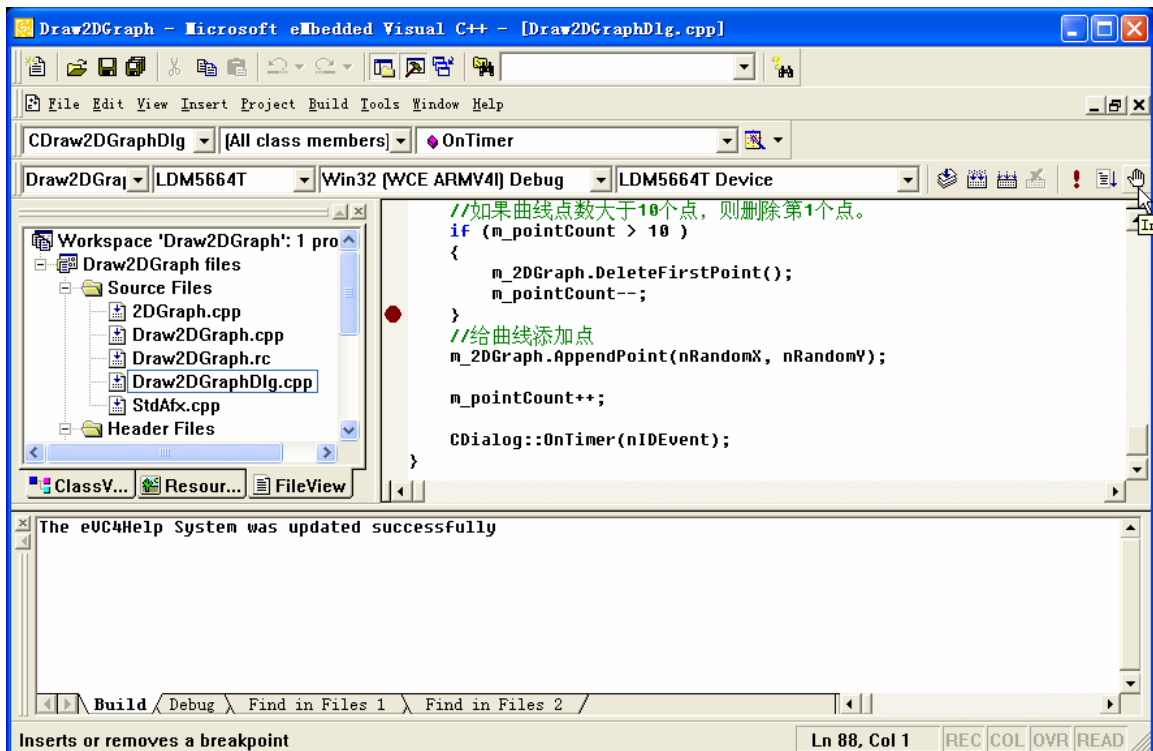


操作结束。

## 4.1.5 利用 EVC 进行同步调试

能够成功地将程序下载到 EMUC6PORT 上之后，我们就可以进一步利用 EVC 环境调试 (Debug) 这个程序。

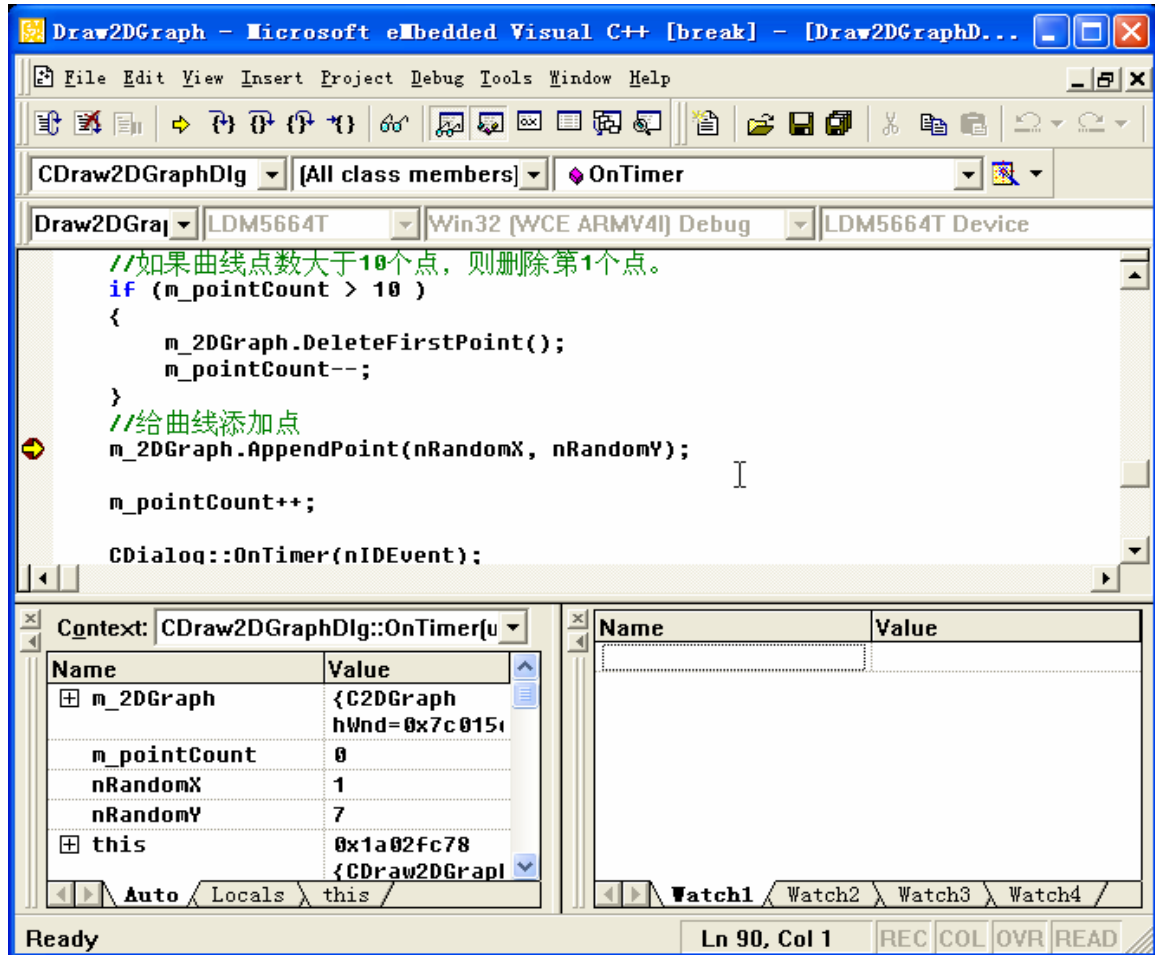
如下图所示，在程序中设置一个断点，如图：



使用“go”命令：



进行同步调试，如图：



## 4.2 基于 Visual Studio 2005 开发环境

### 4.2.1 安装 VS2005

VS2005 全称Microsoft Visual Studio 2005，您可以到微软公司网页阅读关于它的详细信息：<http://www.microsoft.com/taiwan/msdn/vs2005/default.msp>，本文中就不详细介绍了。

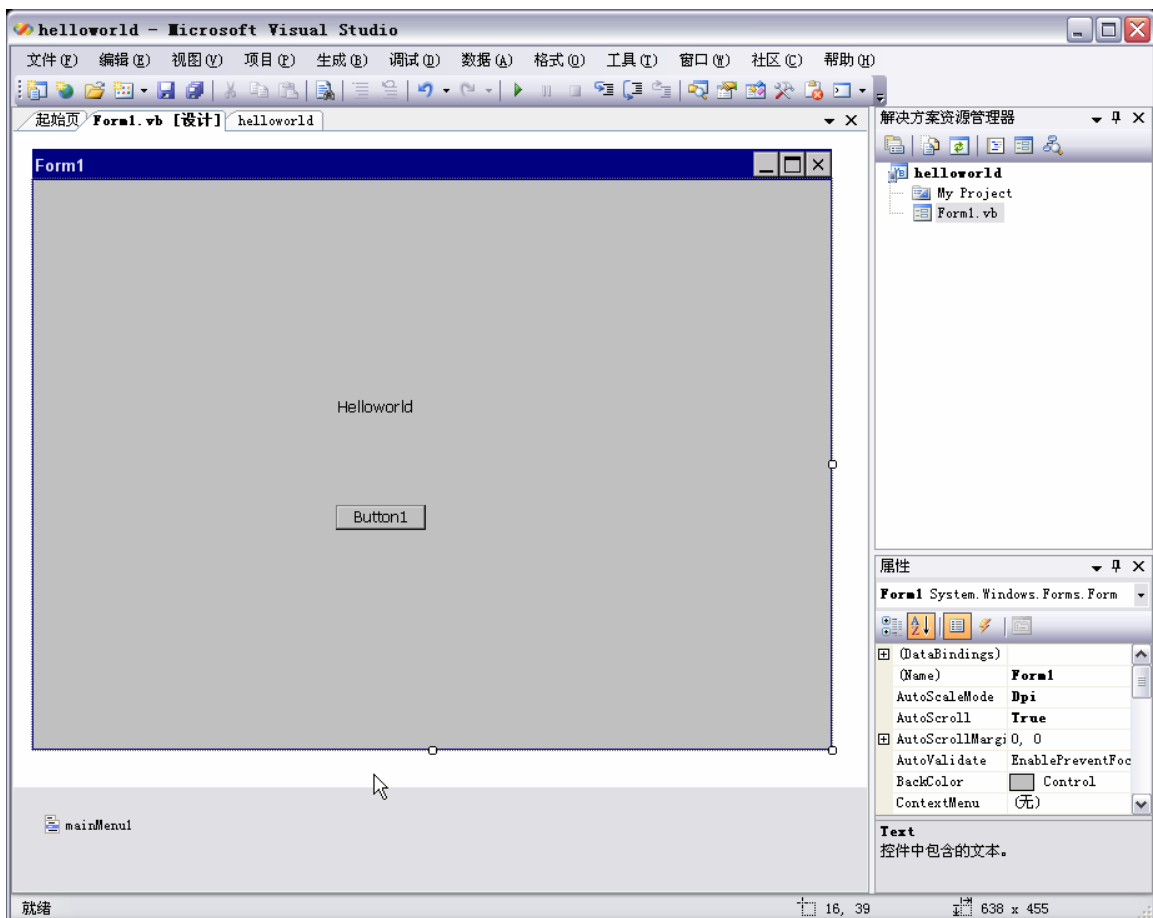
### 4.2.2 创建 Helloworld 实例

在 VS2005 新建一个项目，命名为 HelloWorld。选择“智能设备”——“WinCE5.0”，再

选择“设备应用程序”模块：



点击“确定”退出。这里我们仅以一个简单的程序为例，在窗体中加入一些控件，如下图所示：



### 4.2.3 建立连接

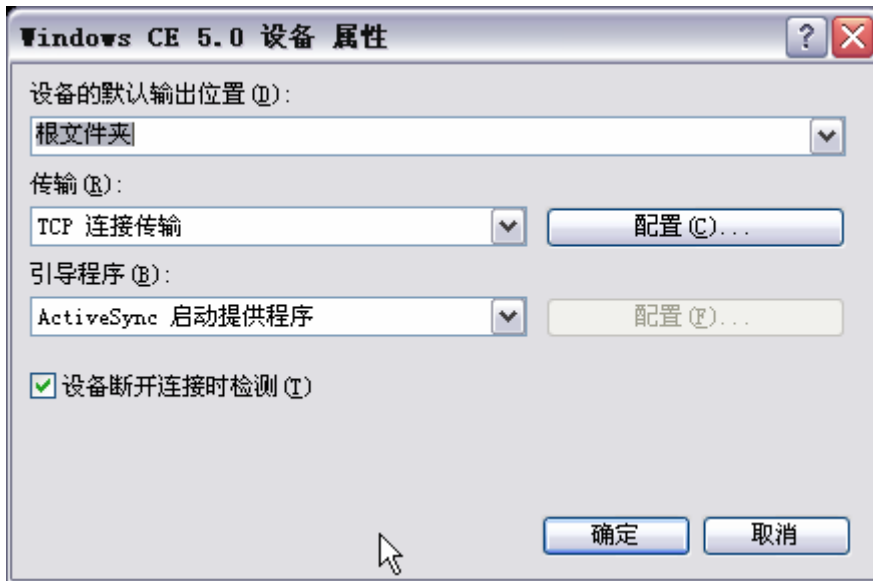
要求此时您已经连接好了 USB 线、同时激活了板子与 PC 机上 Microsoft ActiveSync 软件之间的同步连接。设定好将要连接的平台，单击“工具”菜单下的“选项...”：



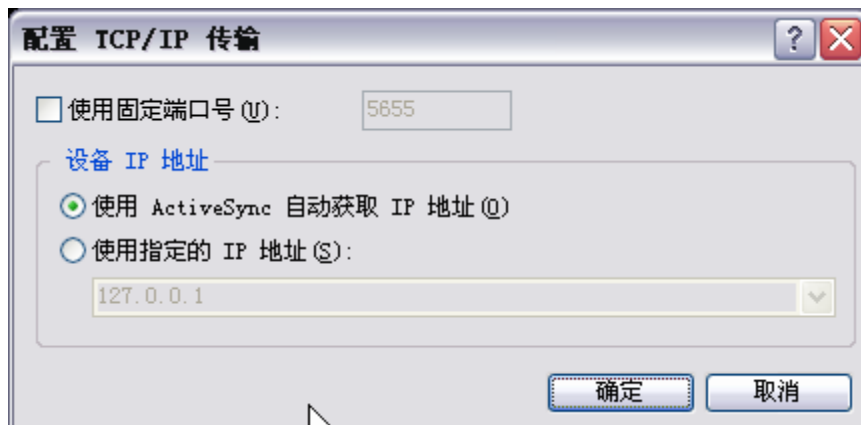
在弹出的对话框中选择“设备工具”->“设备”，在右侧选择“Windows CE 5.0”，默认设备中也选择“Windows CE 5.0 设备”：



单击上图中的“属性(P)...”，请按照下图所示来设置：

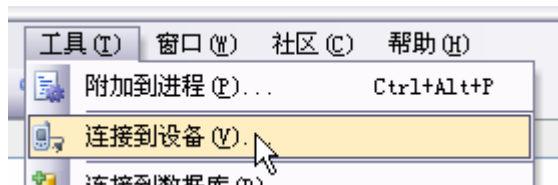


单击上图中“TCP 连接传输”右边的“配置...”按钮，显示如下：



这里选中“使用 ActiveSync 自动获取 IP 地址”即可。单击“确定”，返回主窗体。

单击菜单“工具”下的“连接到设备”项目：



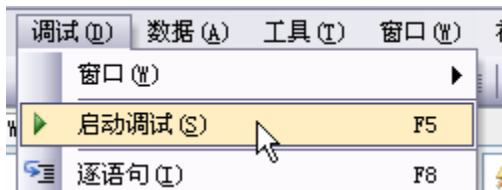
这时将弹出一个连接窗口，VS2005 开始探测与板子之间的连接，如果此时您已经连接好了 **USB 线**、同时激活了板子与 PC 机上 **Microsoft ActiveSync** 软件之间的同步连接，过一会儿，就会出现连接成功的提示：



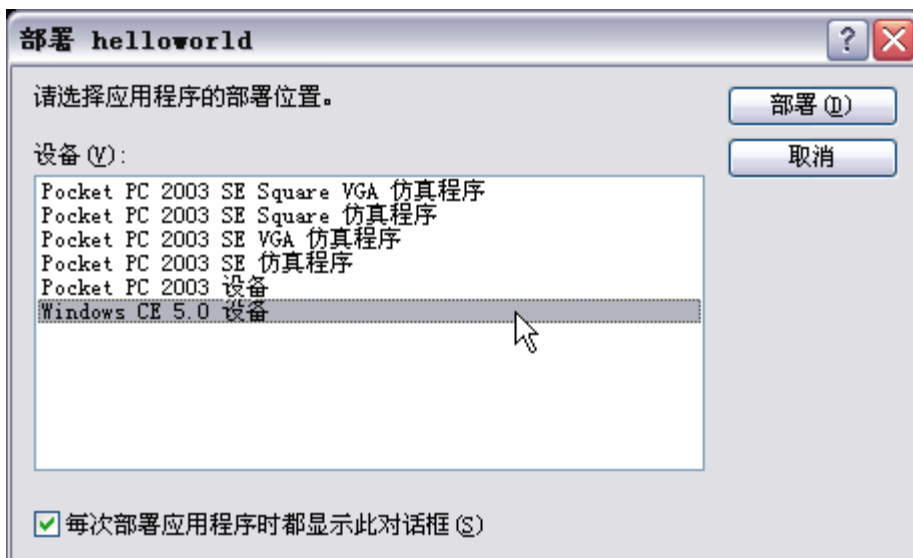
这样，我们就成功地建立了两者的连接。

## 4.2.4 编译和下载

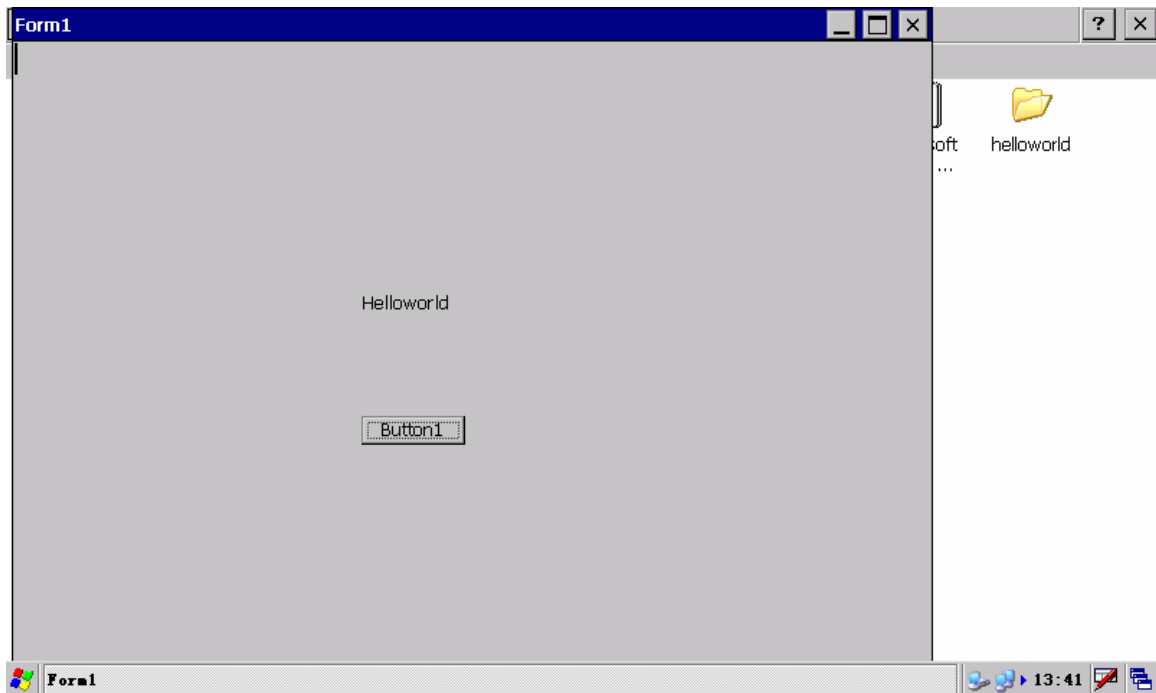
若上一步连接成功，点击菜单“调试”下面的“启动调试”项目：



弹出“部署 HelloWorld”的对话框，选中其中的“Windows CE 5.0 设备”，然后点击“部署”按钮：



如果程序没有问题，VS2005左下角的状态栏中会显示“部署已成功”，等待几秒钟以后，在板子的界面上会显示出应用程序界面，如下图所示：



用户可以观察程序运行情况，或者设置断点进行调试。