一、前言

1、首先谈谈本人基础。我熟悉 AVR 单片机，使用过 ATmega16 单片机大部分功能，如 4X4 键盘、UART、SPI、I2C、Timer、PWM 等等，接触过 DSP 芯片 TMS320C54X，懂得 VHDL 语言，简单使用过 LPC2131，并在其上移植过 uC/OS-II，学习过有关操作系统的基本知识。用过的相关软件有：ICCAVR、ADS1.2、CCS、uVision 等。

2、学习不要着急，如果你没有操作系统的基础，可能一时接受不了操作系统这个概念，而是拿着自已天天用的PC机的Windows操作系统的界面钻牛角尖。不要总感觉这资料是垃圾，那本书是骗钱的，之所以你看不懂那是因为你还没达到看懂的程度！当然现在骗钱的书很多，著书者很少从读者的角度去写书！

3、我所使用的开发环境：PC XP+GEC2440+WinCE5.0+VS2008+ActiveSync4.5+DNW
GEC2440 是广嵌科技术的开发板，不喜欢广嵌，售后技术支持做的不好，而且技术论坛也没人回帖，不建议使用本实验板！其实板子做的还行，只是相关服务有待加强！！！

二、入门准备

1、什么是操作系统？
这个问题是困扰了我很久的问题。由于我们大家对于 Windows XP 等 PC 机操作系统过于熟悉，而此操作系统又过于傻瓜式，以致于使我们感觉不到操作系统的本质，仅感觉操作系统就是一个窗口，反正在我接触嵌入式操作系统前是这样认为的。那么什么是操作系统呢？从我使用 uC/OS-II 的体验来说，简单点儿说操作系统就是一个调度器，从我现在使用 WinCE 的体验来说，操作系统就是管家婆。总的来说吧，操作系统就是用算法实现的管理系统所有资源一个后台。可能这样说还是有些抽象，别急，慢慢来！

2、操作系统安装在哪儿呢？界面呢?
这个问题绝对是 Windows 操作系统使用后遗症，总感觉操作要像我们 PC 机装系统一样进行安装，要有像 XP 等 OS 一样的操作界面。其实界面仅仅是人机交互的一种方式而已，不是操作系统必备的元素，而是操作系统的一种趋势，因为现在对友好的人机交互界面要求越来越高，而且是傻瓜程度过高越好！
以 uC/OS-II 为例，它的核代码就是几个 C 源文件，使用它时将其像其它程序一样加入你所建立的工程即可，当然在 uC/OS-II 与你所写的普通代码之间要有一个桥梁来进行链接，这个桥梁就是我们在移植操作系统时所要写的文件，它根本没有界面一说，为什么说它是操作系统呢？因为 uC/OS-II 有操作系统的一切特征！操作系统都有什么特征呢？自己网上查一吧！
那么移值 uC/OS-II 在 CPU 上有什么好处呢？个人认为，操作系统的核心好处在于多作务管理与调度。任务较少时，也许感觉不到它的好处，但任务多了，操作系统的好处就明显了，比如说吧，你的实验板上有八个 LED，要求你实现这八个 LED 以八种不同频率进行闪烁，你该怎么写呢？如果有了 uC/OS-II 操作系统，这就太简单了，将每个 LED 闪烁按 uC/OS-II 要求形式写成任务，然后将八个任务交由 uC/OS-II 调度即可！
说了一大堆 uC/OS-II 的相关内容，下面进入正题，谈 WinCE！

3、相关术语

0）PC 机（Personal Computer）就是指你的电脑，OS（Operating System）是指操作系统！
1）BSP（Board Support Package，板级支持包），介于硬件平台和操作系统之间的一层，属于操作系统；不同的操作系统对应于不同定义形式的 BSP。
2）Bootloader 与 BIOS：Bootloader 是引导程序，就是对实验平台进行初始化，设定一些相关参数等等。就我现在使用的 WinCE5.0 来说，根据个人理解，Bootloader 与 BIOS 是一个东西，就是在 ADS1.2 下的一个工程而已，里面含有 start.s 及其它的一些相关代码。这个现在我还有些模糊，仅谈到此。
3）OEM：Original Equipment Manufacturer 原始设备制造商
4）OAL：OEM Abstraction Layer
5）DLL：Dynamic Link Library,动态链接库
6）MFC：Microsoft Foudation Class,微软基础类
7）API：Win32 Application Programming Interface,Win32 应用程序编程接口
8）SDK：Software Development Kit，使用 WinCE 时必须安装 SDK，你不必在网上找 SDK 的安装包，对于每个具体实验板都对于一个 SDK，这个可以用 PB 生成。
9）PB、VS、EVC：PB 是 Platform Builder 的简称，VS 是 Visual Studio 的简称，EVC 是 Embedded Visual C++的简称。
10）Nand flash 与 Nor flash：与非 flash 和或非 flash，前者价格便宜，后者较贵。

4、所需开发软件
1）Platform Builder 5.0：此软件用来定制操作系统，生成内核，生成 SDK，编译驱动程序等。
2）Visual Studio 2008：我用的是此版本，当然也许不许这么高版本，网上用的多的是 VS2005，还有使用 EVC++的。此软件用来编写应用程序。
3）ActiveSync 4.5：此软用来同步 PC 机与实验板进行同步。此软件可以在微软中国官方网站下载。
4）DNW：串口调试工具，在上电时用来显示 BIOS 发往串口的相关信息，也可以输入相关参数进行设置的。
5）USB 同步驱动：必备!!!
VS2008 的安装按提示按装即可，PB5.0 的安装可以参考天嵌科技的手册进行安装，讲的很好很详细，至于 PB5.0 补丁的安装只需安装 Net2.0 与 Net3.5 的那一个即可，其实安装与否我原由我也不清楚，好好参考天嵌科技术的手册吧。另外要参考天嵌科技术的手册添加 BSP，生成相应的 SDK，然后安装 SDK，这些都安装好后还有一个工作就是安装 USB 驱动，这一关一定要过，安不上的话就等着安上再说！

5、相关参考书目：
何宗键 编著.Windows CE 嵌入式系统.北京：北京航空航天大学出版社
还有就是各个开发板厂商的使用手册都是很好的资料，必备！
主要有天嵌科技 TQ2440、朗成电子 AT2440EVB、友善之臂 mini2440 和 QQ2440、广嵌科技 GEC2440、飞凌等等，只有飞凌的资料是不公开的，其它的都可以在相关网站下载到。
有关 Visual Studio 的书可以参考一下机械工业出版社刘冰等编著的《C++程序设计教程——基于 Visual Studio 2008》，这本书我也没细看过，扫了一眼，讲的还抽合吧。

三、打开 WinCE 的大门

0、我的学习方法

我学习的理念是首先把整套开发环境搭建起来，然后运行一个最简单的程序入门再说，至于更深一层次的内容慢慢研究！

1、开发平台的建立

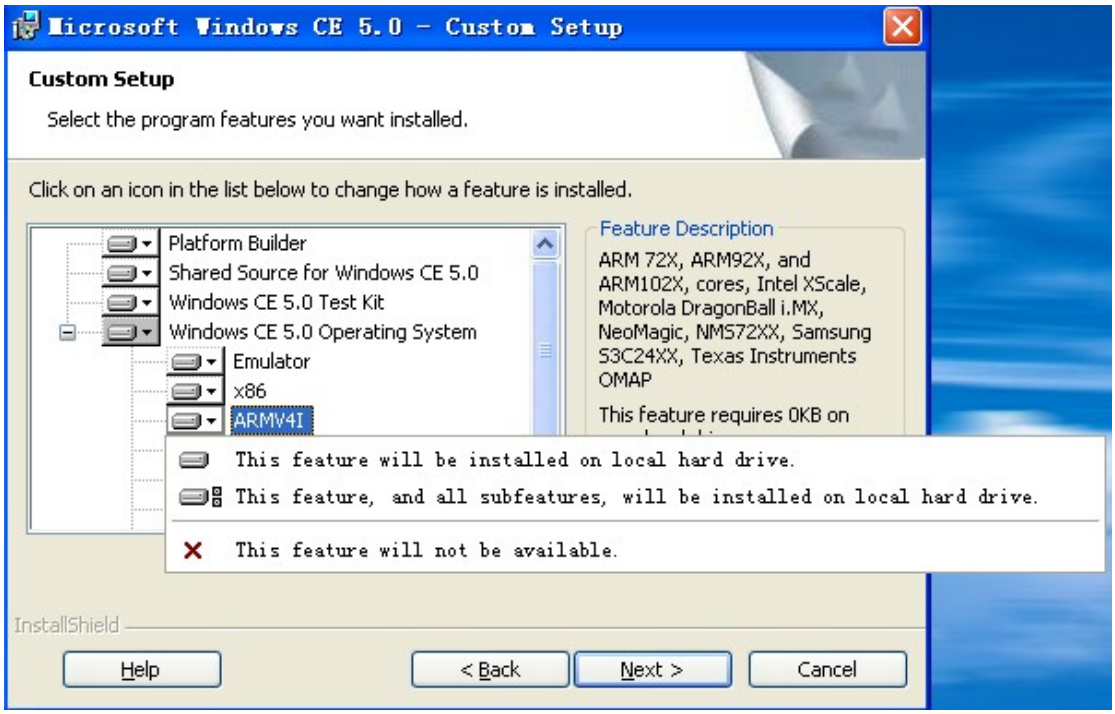
个人认为，学习单片机也好，ARM 也罢，首先要解决的事情就是将整套开发环境搭建好，成功跑一个最简单的程序，就算入门了，下面我说说 WinCE 开发环境的建立。

1）安装 VS2008，根据提示安装即可，没什么注意事项，唯一的提示就是要留有足够的硬盘空间，仅 VS 就得留 2G 空间吧，如果安装 MSDN 帮助文档的话，再留 3G 吧！

安装好 VS 后初次打开会让你选择默认开发环境，有 1、Visual Basic 开发设置；2、Visual C# 开发设置；3、Visual C++开发设置；4、Web 开发设置；5、常规开发设置。个人认为选哪一个都无所谓，差别仅在于新建项目时项目类型一栏的排列顺序有所差别而已。这个设置可以按如下更改：工具---》导入和导出设置（I）…---》重置所有设置---》否，仅重置设置，从而覆盖我的当前设置

2）安装 Microsoft_DotNetFXCHS1.1.exe，这是安装 WinCE 的前提，如果你电脑上装过 VC++ 等软件，应该就不用安装了，因为 Visual Studio 系列软件都需要这个的。

3）安装 Platform Builder5.0，安装 PB 其实就是所谓的安装 WinCE，这个参考天嵌的手册 1.1 节内容安装即可，有一点天嵌手册中没提到的是：



安装选择处理器到这一步时如果 ARMV4I 是一个“×”，则应单击向下的小黑三角，选择第一个或第二个选项，其它的我也不是很懂，反正第一次入门多安装了总比不能用好，以后精通了再选择最优方法吧。

4）安装 WinCEPB50-081231-Product-Update-Rollup-Armv4I.msi，这是 PB5.0 的一个补丁，网上相关文章千篇一率，写了一大堆补丁，个人认为安装这个就足够了，其它的等你用到时再安吧，我用了这么些天也没感觉到补丁有什么用。

5）安装 ActiveSync4.5，这个很简单！安装完后打开“我的电脑”就会有一个“移动设备”图标，这个以后有用！

6）DNW 不用安装，是一个绿色版的软件，累似于串口调试助手，但功能强一些吧！

7）安装 USB 同步用驱动，用 USB 线将实验板与 PC 机连接起来会提示安装驱动，这个实

验板厂商应该会提供的，也可以从网上下载！USB 驱动安装不当会导致 PC 机蓝屏，所以这一步必须过关，否则原地待命！
8）添加 BSP：打开 PB5.0，添加 BSP 包，BSP 是由厂商提供的，别告诉我实验板是你自己做的，一上手就写 BSP 有点不现实吧。参考天嵌手册 2.1.1 节内容，很轻松搞定！
9）安装 SDK：参考天嵌手册 2.4 节内容很轻松搞定，然后安装即可！
到现在为止环境基本搞定，其它相关细节参考开嵌手册即可！

2、什么叫定制操作系统？为什么要安装 WinCE 到 PC 机上？
由于受 uC/OS-II 操作系统的影响，总想着 WinCE 的源代码在哪儿？那些 API 函数的原型在哪儿头文件中？须知 uC/OS-II 的源代码是开源的，而 WinCE 的代码则相反！uC/OS-II 的呈现在我们面前的是几个 C 语言源文件，WinCE 呢？就我现在的理解，将 PB5.0 装到 PC 机上意味着将 WinCE 的全部功能放到了 PC 机上，然后我们需要做的是通过 PB 从 WinCE 所有功能中挑选出自己需要的功能，这些功能经自己通过 PB 组合后生成一个 NK.bin 和 NK.nb0 文件，关于这两个的区别参考天嵌手册的第 92 页，具体我也不太懂。但我个人的理解是，生成的 NK.nb0 就好像是我们在学单片机时生成的 hex 文件，我们将.nb0 烧到实验板上也就是将我们裁剪（即所谓的定制!!!）好的 WinCE“安装”到实验板上了，这个过程可以近似理解为将 hex 文件通过 ISP 或其它方式烧到单片机里一样的。
综上所述，我们安装 PB 在 PC 机上是将 WinCE 所有功能“暂时放到”PC 机上，然后通过 PB“挑出”你所需要的功能后并将其组合（即.nb0 文件）“放到”实验板上即可！
打个比方说吧，WinCE 各种功能就好像一块块积木（即安装 PB），然后我们用需要的积木搭出我们需要图形（即我们用 PB 定制的操作系统）。
WinCE 代码是不开源的，不要天天去想 API 函数在哪儿呢？真想看的话去找找比尔盖茨看看能不能让你看看！

3、Bootloader(BIOS)、WinCE 核（即 nb0 文件）及我们用 VS 编写的应用程序怎么烧到实验板上？都烧在哪里了呢？
1）Bootloaer 与 BIOS 的区别我还没搞懂，就现在的理解还是将它们合二为一，这个有待解决。它们可以用 sjf2440.exe 进行烧写；烧到哪儿了呢？这个由自己设定，我也没有烧过，只有一个粗略的理解：S3C2440 外扩了 Nand flash 和 Nor flash，还有 SDRAM，芯片内部也有（应该有吧？），这里我们不用管内部与外部，这个只要按要求进行外扩后用寻址到什么地址 CPU 会自动找相关存储空间的。我们可以将所有的 flash 看成一个整体，它是用来固化程序的，把所有 RAM 看成一个整体，它是用来运行程序的。我们将 flash 分成不同的分区，上电时 CPU 要从根据设定的起始地址的代码开始执行，个人认为 Bootloader 应该就烧在这里吧！这里的理解个人还有待加强，仅供参考！
2）WinCE 核可以通过过 USB 等方式下载到实验板上，比如用 DNW 就可以进行下载，之所以下载到哪儿了？下载到 flash 其中的一个分区，具体由 Bootloader 引导程序和你自己设定的参数！CPU 再聪明还得听人的指挥！
3）VS 编的应用程序烧在哪儿了？这个我还没弄明白，我是通过 ActiveSync 将 VS 生成的.exe 文件发送到实验板上或直接复制到实验板上的。怎么粘呢？安装好 ActiveSync 后 PC 机的“我的电脑”会多一个移动设备图标，如果你的 USB 同步驱动安装好的，双击“移动设备”其实就是相当于在远程控制着实验板，将 VS 的应用程序复制过来然后在实验板端相关目录下找到应用程序运行即可！不过这样运行应用程序是在 RAM 中运行，没有进行固化，至于怎么固化，我再好好研究！
4）这里一直提到 flash 分区的问题，至于究竟是怎么分区的目前我也不是很懂，个人认为是

通过 bootloader 进行相关设计的吧，在以前接触 LPC2131 时似乎看到过相关内容。

四、走进 WinCE

以下默认开发平台所有软件已安装完毕：

1、烧写 Bootloader（没烧过，只是这样认为要首先烧写 BIOS）
2、烧写 WinCE 核
由于本人拿到实验板时以上两步已做好，即我拿到的实验板上就有烧好的 WinCE5.0 操作系统，因此以上两步是我猜测的！
3、打开 VS，按照天嵌手册 4.2 节内容进行操作，如果成功运行，则 OK！本步骤成功后则说明平台搭建成功，但 HELLO 程序不涉及实验板上的任何具体硬件，个人认为这只能算是入门了一半，还有一个坎没迈过去，那就是流驱动的开发，因为开发第一个流驱动的工作量并不比搭建平台省事，尤其是全靠自己琢磨，旁边无人指导、无人探讨！
4、运行成功 HELLO 程序后你可能会想，如果想点亮实验板上的 LED 灯该怎么用啊？基于 VS 开发环境也没法控制 CPU 的 IO 寄存器啊？这就需要流驱动了！
大部分实验板的手册都有详细的 HELLO 程序过程，但很少有手册详细讲解流驱动的开发过程，下面以我的经历详细的写一写：
流驱动的基本介绍可以看看天嵌手册的第五章，脑子里起码要先有个基本概念！
1）打开以下目录：E:\WINCE500\PLATFORM\smdk2440\DRIVERS，并在该目录下新建一个文件夹，命名为 GPIOdriver，并用记事本打开该目录下的 dirs 文件，按其格式添加 \GPIOdriver，dirs 没有扩展名，打开看看就懂，没什么特别的！
注：其中 E 盘是我的安装目录，smdk2440 是 GEC2440 实验板提供的 BSP，按要求拷贝到了 E:\WINCE500\PLATFORM\ 目录。
2）打开刚刚新建的 GPIOdriver 文件夹，新建 txt 记事本文件，命名为 makefile，打开加入以下内容：!INCLUDE $(_MAKEENVROOT)\makefile.def，然后将.txt 扩展名去掉，使其变为无扩展名的文件。
3）仍然在 GPIOdriver 文件夹内，仍然新建 txt 文件，命名为 GPIOdriver，加入以下内容后将其扩展名更改为.def 文件：

```
LIBRARY GPIOdriver
EXPORTS
    GIO_Close
    GIO_Deinit
    GIO_Init
    GIO_IOControl
    GIO_Open
    GIO_PowerDown
    GIO_PowerUp
    GIO_Read
    GIO_Seek
    GIO_Write
```

4）仍然在 GPIOdriver 文件夹内，仍然新建 txt 文件，命名为 sources，加入以下内容后将其扩展名删除，使其成为无扩展名文件：

```
RELEASETYPE=PLATFORM
```

```
TARGETNAME=GPIODriver
TARGETTYPE=DYNLINK
DLLENTRY=DllEntry

TARGETLIBS= \
    $(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\coredll.lib \

MSC_WARNING_LEVEL = $(MSC_WARNING_LEVEL) /W3 /WX

INCLUDES= \
    $(_TARGETPLATROOT)\src\inc; \
    $(_COMMONOAKROOT)\inc; \

$(_PUBLICROOT)\common\oak\inc;$(_PUBLICROOT)\common\sdk\inc;$(_PUBLICROOT)\c
ommon\ddk\inc;    \
    ..\..\inc

SOURCES= \
    GPIODriver.cpp
```

5）仍然在 GPIOdriver 文件夹内，建立 GPIOdriver.cpp 和 GPIOdriver.h，这个大家找相关例子程序进行参考，我是参考的天嵌的例子，不过由与 BSP 不同，编译时总是出错，错误原因就是一些头文件不一样，有一点编程序经验应该都能查出来，这时最好的解决办法是把需要包含的内容直接写到 GPIOdriver.h 中去，这样就不会有编译错误了。至于如何编译，别急，请看后文！先这么做好，排查错误要慢慢来，先写好一个再说！

6）以上出现了多次 GPIOdriver，这些地方要命名一致，在写 def 文件时，GIO 只能是三个字母，这里注意！

7）以上工作完成后，GPIOdriver 文件夹下的工作就完成了，然后打开 E:\WINCE500\PLATFORM\smdk2440\FILES 目录，以记事本打开 platform.reg 文件，添加如下代码：

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\LEDdriver]
"Prefix"="GIO"
"Dll"="GPIOdriver.dll"
```

注意 GIO 与 GPIOdriver.dll 要与前面的命名一致！另外要注意，.reg 文件中有很多 IF 与 END 的配套使用，添加代码时不要放在这些中间了！至于 IF 与 END 的作用，初学者先跳过！

8）同样在 FILES 目录下打开 platform.bib 文件，加入以下代码：

```
GPIOdriver.dll $(_FLATRELEASEDIR)\GPIOdriver.dll NK SH
```

同样须注意命名与 IF、END 的配对！

9）以上工作完成后基本工作就差不多了，这里再说明一点，关于目录各个实验板厂商的 BSP 包各不相同，比如说天嵌的目录应为 TQ2440\Src\Drivers 和 TQ2440\Files，这个要灵活应变！

10）打开实验板商家所带的 PB 平台文件.pbxml 文件，在 PB 的左侧选择 FileView 选项（我这里默认就是这个选项），这里有树形结构的文件夹目录，打开 PLATFORM，smdk2440，drivers，然后会看到你刚刚建立的 GPIOdriver 文件，右击，选择 Build Current Project，就会对刚刚建立的流驱动进行编译，如果有错误会有提示，修改后再编译，直到没有错误为止！

11）点击菜单栏 Build OS---Build and Sysgen，对.pbxml 文件进行编译，这个大概要花一个

多小时，这里你可以去看会儿电影，编译完成后会有一些警告，不用管，只要没错误就行！编译前要选中 Build OS 菜单下的 Clean Before Building，Copy files to Release Directory afer build，Make Run-time Image After Build，至于为什么我也不是很明白，都是一些过来人提供的经验，这个等熟练后慢慢琢磨吧。
还有一种说法是添加一个驱动不用选择 Build and Sysgen，这样耗时太长！究竟是怎么弄来着，我忘了，慢慢研究吧。
12）编译成功后找开与.pbxml 文件同目录的 RelDir\smdk2440_ARMV4I_Release 目录，很发现有好多文件，可以找到 NK.nb0 和 NK.bin 文件。找这个文件的快速方法是将这些文件以“详细信息”方式显示，然后单击一下“大小”，会将所有文件以大小排序，最大的那个就是了，这样找起来会快些。到这里驱动编写完毕，NK.nb0 暂时留用！
13）打开 VS2008，新建项目，可以参考天嵌手册的第六章内容，至于添加的代码可天嵌的例程序，这个很简单的。这种小问题自己琢磨！
14）将已生成的 NK.nb0 通过 DNW 发送到实验板上，然后运行刚刚写好的 VS 应用程序就 OK 了！到现在第一个流驱动编写完毕！
15）以上的代码中好多不用问为什么，初学者直接复制即可，始果想弄明天一些，可以在网上搜索“WinCE 注册表”、“WinCE 流驱动”等关键字。

五、补充说明

WinCE开发有三种：API方法、MFC方法和.NET方法，曾经看过西安电子出版社张勇编写的《Windows CE 应用程序设计》，这本书是基于API方法讲解的，里面以例程序为主体，通过对例程序的讲解，循序渐进，讲的还可以吧，比较适合我，但这本书的平台是基于智能手机或WM6 仿真器的，与我们基于ARM9 开发板还是有很大区别的，比如说使用WM6 仿真器不用安装PB，不用安装ActiveSync，其实只需安装VS2008 和WM6 的SDK即可！因此受这本书的影响，当我基于ARM9 入门WinCE时有许多事不能理解，突然感觉头绪太多太乱了，比如说突然要安装PB、编写流驱动等等。另外，基于API编程很麻烦，因为所有界面都是用代码实现的，当然这种方法的运行效率是最高的。.NET没用过，不知何物，但API与MFC之间就好比使用HTML语言开发网页和使用Dreamweaver编辑网页一样，前者也是全部用代码实现，而后者则直接绘制界面。