深圳市九鼎创展技术有限公司技术文档

文档名称：x210v3s WinCE 手册

深圳市九鼎创展科技有限公司

地址：深圳市宝安区兴业路宝安互联网产业基地 B 区 3003B 室

网址：http://www.9tripod.com

论坛：http://www.xboot.org

版权声明

敬告：

在售开发板的手册会经常更新，请在 http://www.9tripod.com 网站下载最新手册，不再另行通知。

版本说明

版本号	日期	作者	描述
Rev.01	2012/04/25	Terry	原始版本
Rev.02	2012/05/03	Terry	增加 LED 和 audio CODEC 驱动实例
Rev.03	2013/01/29	Terry	增加 inand 烧写方法

技术支持

一：论坛

在 BBS 论坛上发贴询问，是取得技术支持的有效途径，我司技术人员将在 24 小时内回贴，并尽可能的在 48 小时内解决所提出的问题。如遇节假日将顺延。

论坛网址：www.xboot.org

技术支持时间：周一到周五，9：00 到 18：00

二：邮件

在通过 BBS 论坛仍然无法解决的情况下，可以采用电子邮件的形式咨询。

邮箱地址：supports@9tripod.com

三：QQ 群

九鼎创展所有开发平台都对应有 QQ 交流群，QQ 群为广大客户提供一个共同交流讨论的地方，并不代表能够得到及时支持，技术人员并不能保证每时每刻在群里面盯着问题，很多技术问题并不能马上就能得到解决，请尽量在论坛上发贴提问，留给技术人员测试的时间，感谢您的支持。

网　　址：　www.9tripod.com

联系电话：　0755-61952306

E-mail：　supports@9tripod.com

销售与服务网络

公司：深圳市九鼎创展科技有限公司
地址：深圳市宝安区兴业路宝安互联网产业基地 B 区 3003B 室
邮编：518101
电话：4000033436　0755-33121205　0755-33133436
网址：http://www.9tripod.com
论坛：http://www.xboot.org
淘宝：http://armeasy.taobao.com
1688：http://armeasy.1688.com

QQ 群：
x4412/ibox4412一群：　【16073601】
x4412/ibox4412 二群：　【211126231】
x4418/ibox4418 论坛：　【199358213】
x6818/ibox6818 论坛：　【189920370】
x210/i210 一群：　【23831259】
x210/i210 二群：　【211127570】
x3288 技术论坛：　【159144256】

热烈欢迎广大同仁扫描右侧九鼎创展官方公众微信号，关注有礼，您将优先得知九鼎创展最新动态！

目录

第1章 VS2005 和 WINCE6.0 的安装

VS2005 和 WinCE6.0 的安装请参考指导手册:《VS2005_WinCE6.0 安装指导手册(v02)》

第2章 WinCE6.0 系统定制

定制系统是嵌入式开发非常重要的一步，根据项目功能需求，定制出一个稳定精简而又满足要求的系统是一个很有学问的工作。下面我们来动手定制一个功能比较全的系统，这些功能都是比较常用的功能。主要是为了供初学者了解整个定制流程，还有熟悉一些常用组件的功能。实际项目开发中，一般不需要这么多组件，因为这样的系统比较臃肿。

2.1 安装 BSP

BSP 全称是板级支持包，是介于硬件和操作系统之间的一层，对上提供操作系统认识的软件控制接口，对下实现硬件寄存器的实际控制操作，承上启下的作用。不同的操作系统有不同定义形式的 BSP，例如 WINCE 的 BSP 和 Linux 的 BSP 相对于某一 CPU 来说尽管实现的功能一样，可是写法和接口定义是完全不同的，因为不同系统有不同的接口定义，例如同是实现打印操作，不同的系统，实现的操作函数接口是不一样的。但原理基本相同。写 BSP 一定要按照该系统 BSP 的定义形式来写，所以 BSP 的编程过程大多数是在某一个成型的 BSP 模板上进行修改。

X210-II 开发板的 WINCE 系统 BSP 是基于三星官方发布的 SMDKV210 开发板 BSP 修改而来，大部分保留了三星原来的架构，针对 X210-II 开发板的硬件特点（例如使用了不同的音频芯片等）而作了修改和优化。此 BSP 在三星原 BSP 基础上还加入了很多我们特有的功能。

安装 BSP 步骤：

x210-II 开发板光盘目录 \x210\WINCE\BSP 下有两个文件夹

1） S5PV210_SEC_V1
2） SMDKV210

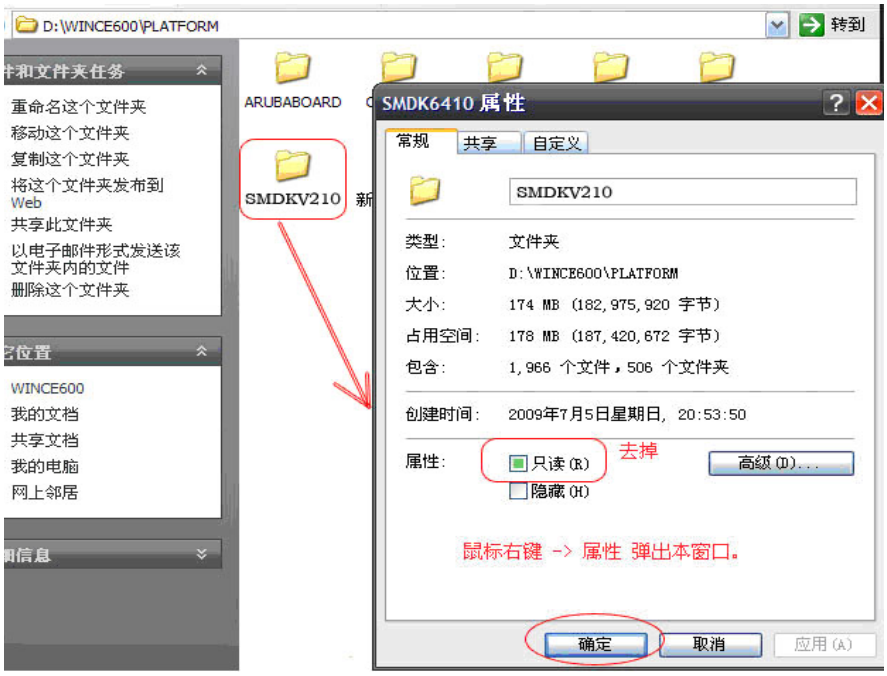
把 S5PV210_SEC_V1 文件夹拷贝到\WINCE600\PLATFORM\COMMON\SRC\SOC 目录下。假如你的 WINCE 是安装在 D 盘，则路径为：D:\WINCE600\PLATFORM\COMMON\SRC\SOC

把 SMDKV210 文件夹拷贝到 \WINCE600\PLATFORM 目录下，假如你的 WINCE 是安装在 D 盘，则路径为：D:\WINCE600\PLATFORM

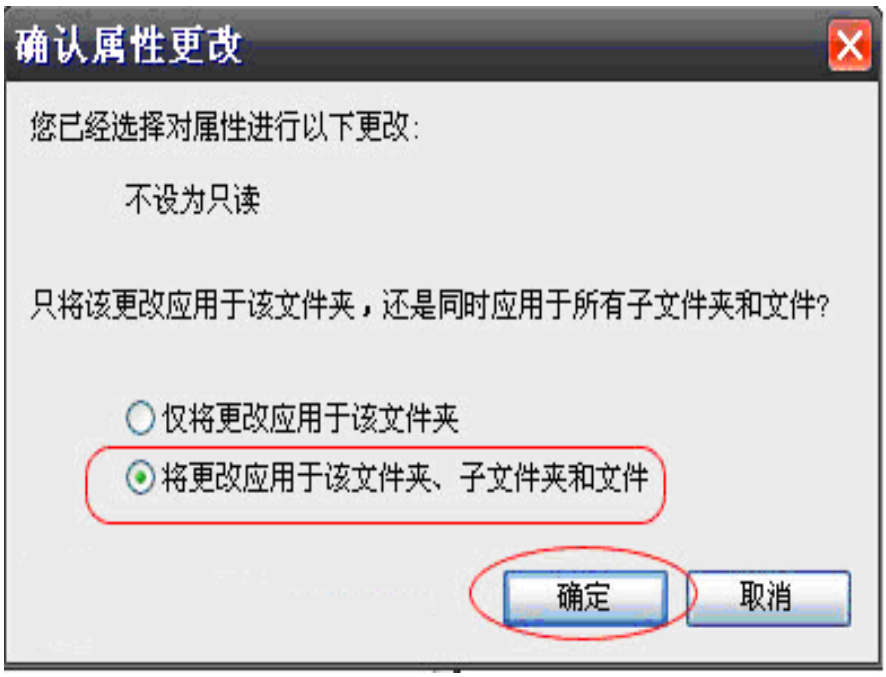
去掉 S5PV210_SEC_V11 和 SMDKV210 整个文件夹的只读属性。

方法：

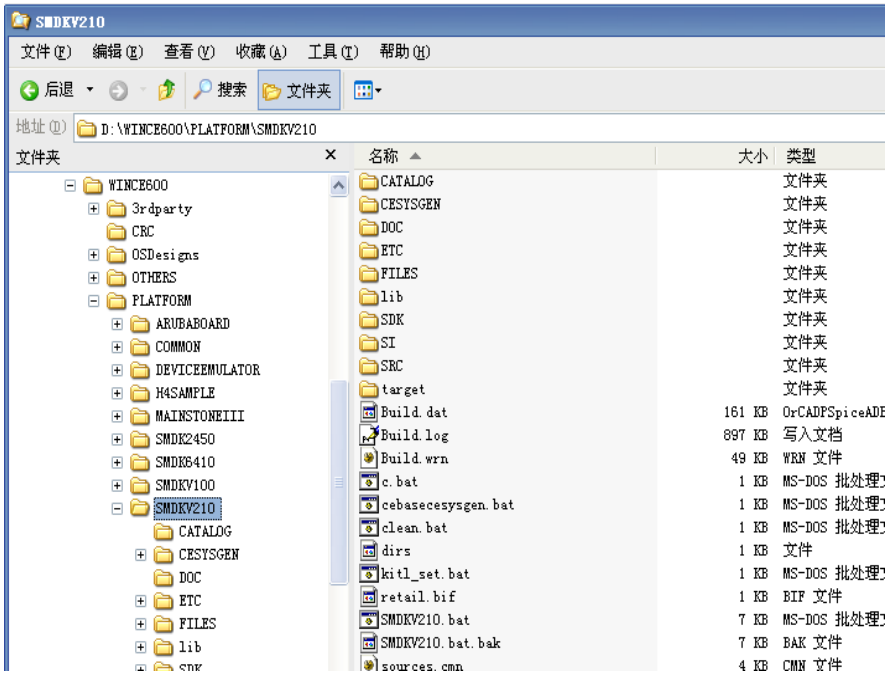
（1） 鼠标右键点文件夹，在右键菜单中选“属性”弹出如下窗口

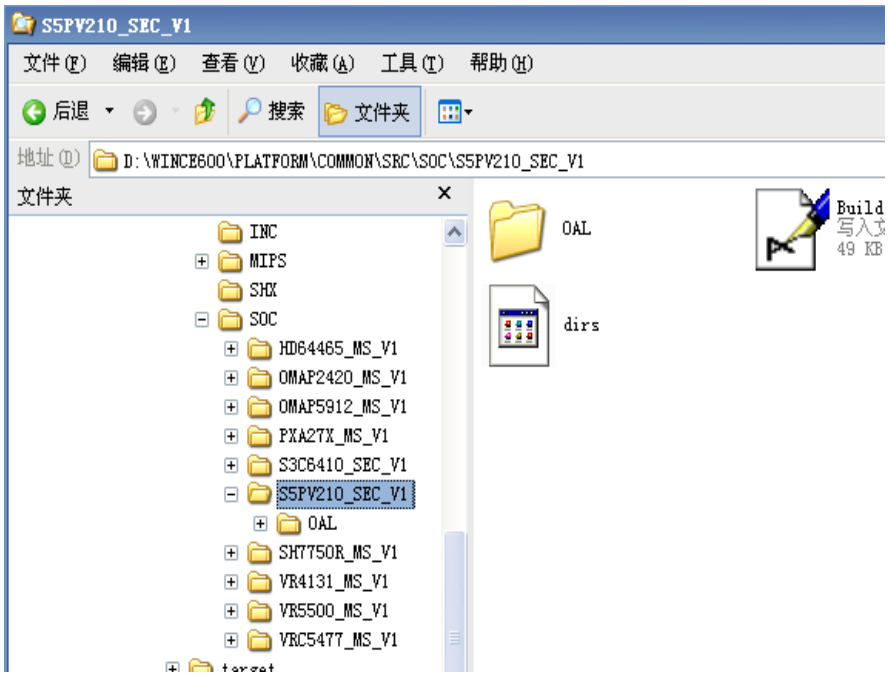


（2） 确定后，系统询问选择更改整个文件夹属性



拷贝后如下图



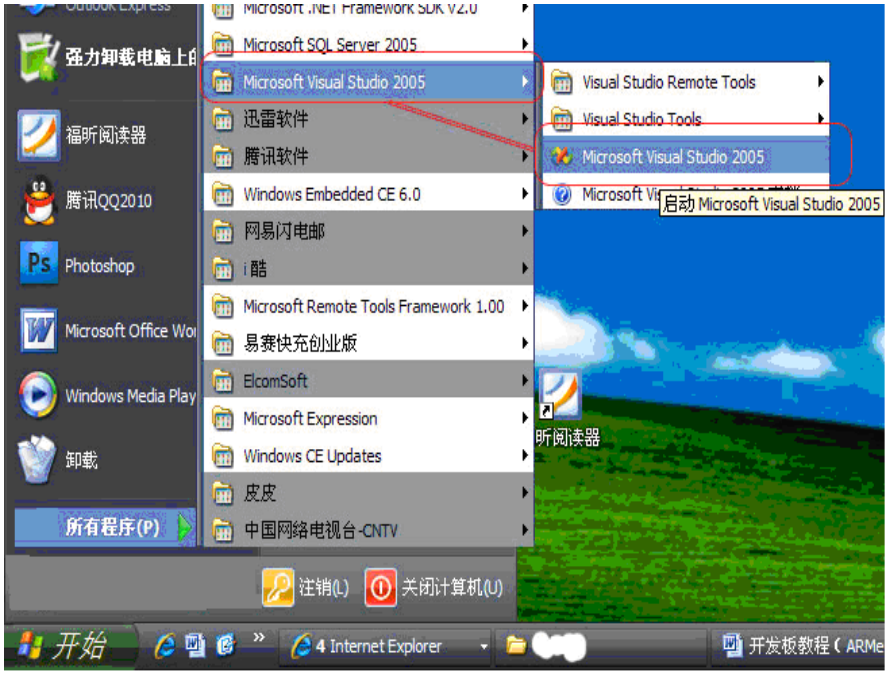


BSP 安装完毕！

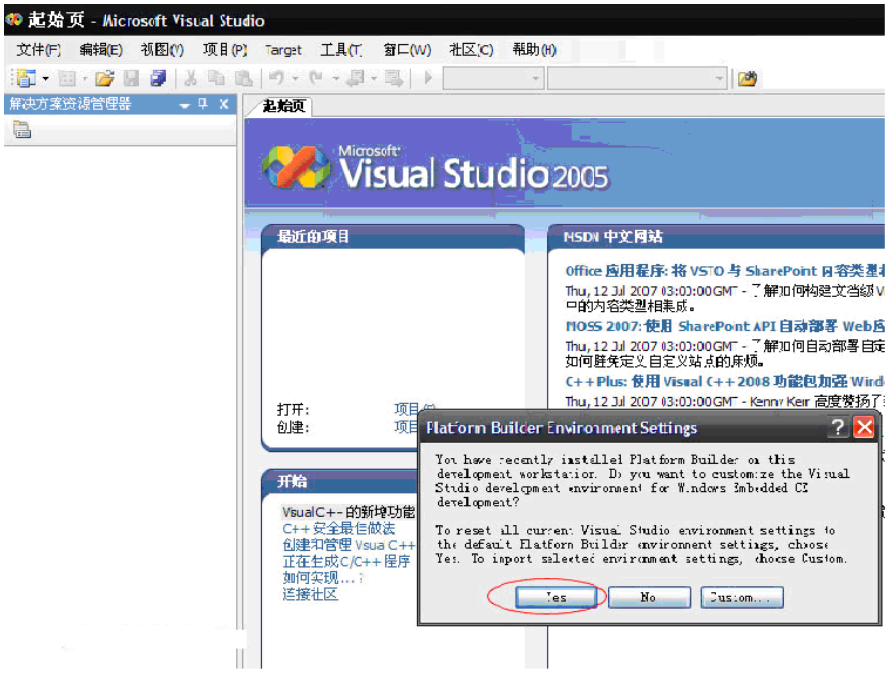
2.2 建立 WinCE 工程

2.2.1 启动 VS2005

开始菜单 > 所有程序 > Microsoft Visual Studio 2005



第一次打开，软件会提示新装了 Windows CE Platform Build 插件，是否要设置。



选择 Yes，则按默认值自动设置环境变量。更新进 VS2005。

2.2.2 新建工程

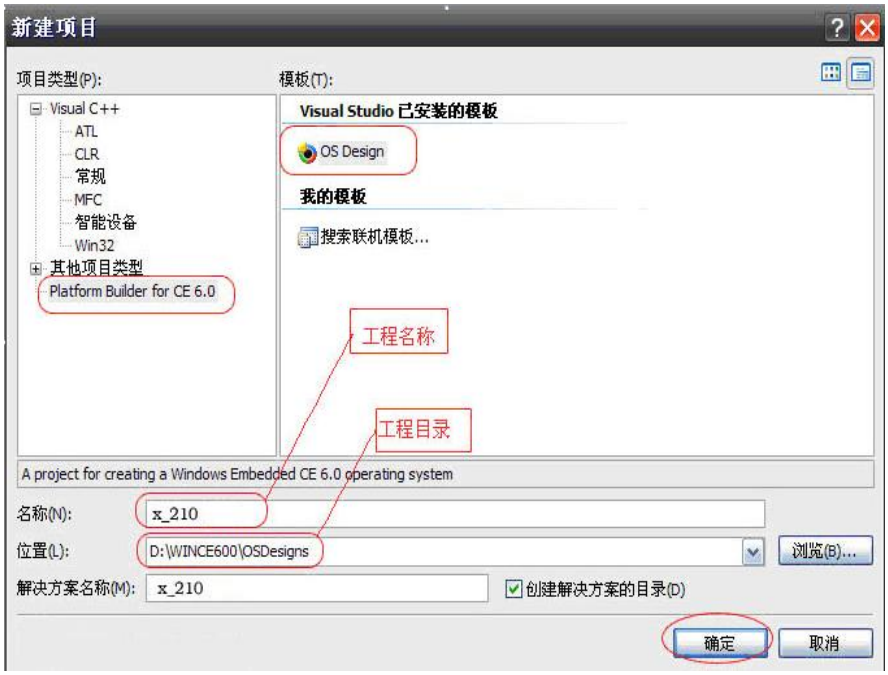
1. 文件 > 新建 > 项目 进入新建工程向导。

选择项目类型：Platform Builder for CE 6.0

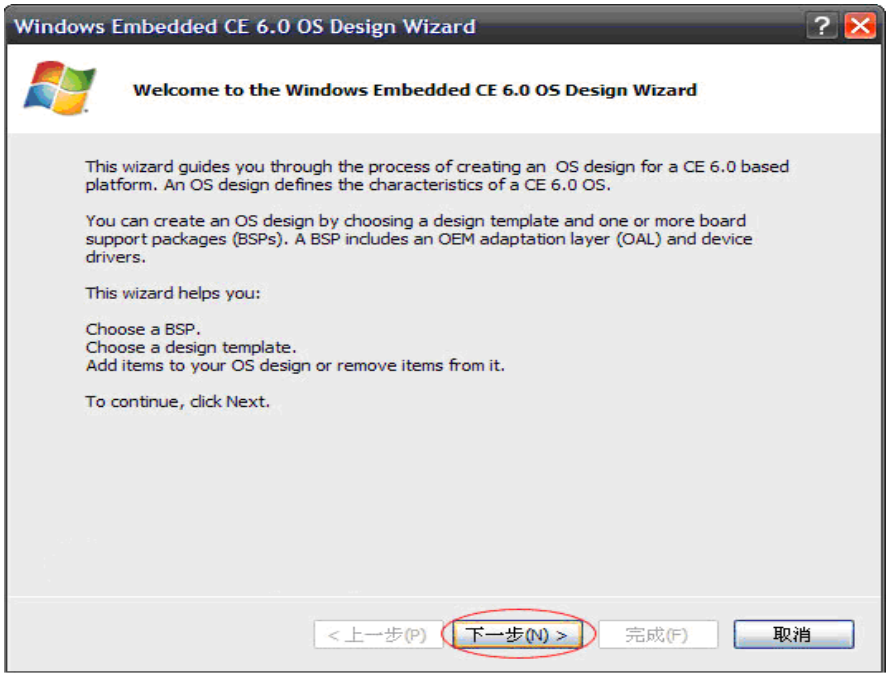
名称：工程名字

位置：工程存放目录

点“确定”

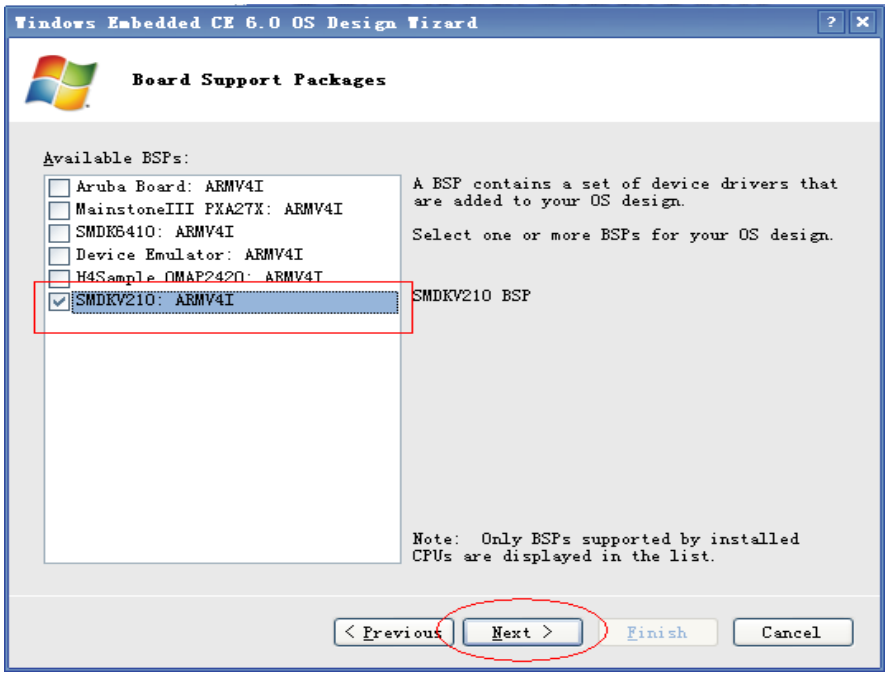


2. 点确定，进入欢迎界面，“下一步”



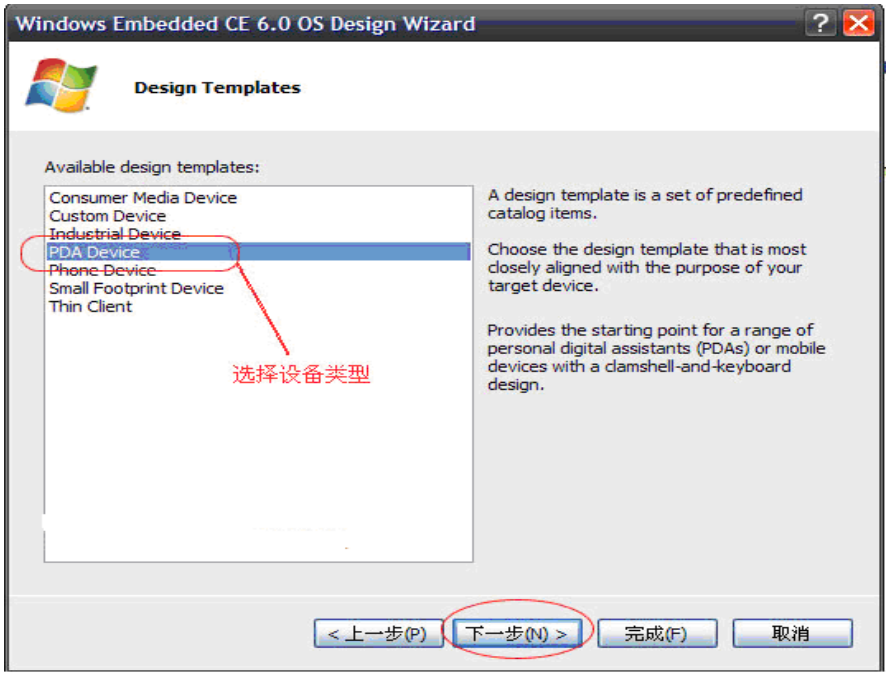
3. 选择 BSP，这里可以看到我们前面安装的 SMDKV210 BSP。

选择：SMDKV210:ARMV4I

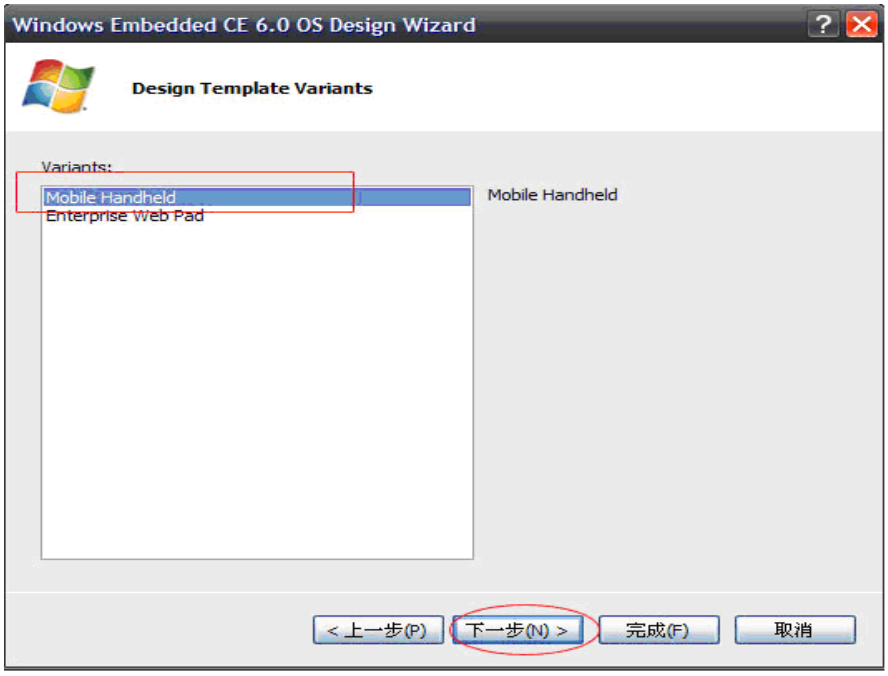


4. 选择设备类型

选择：PDA Device “下一步”



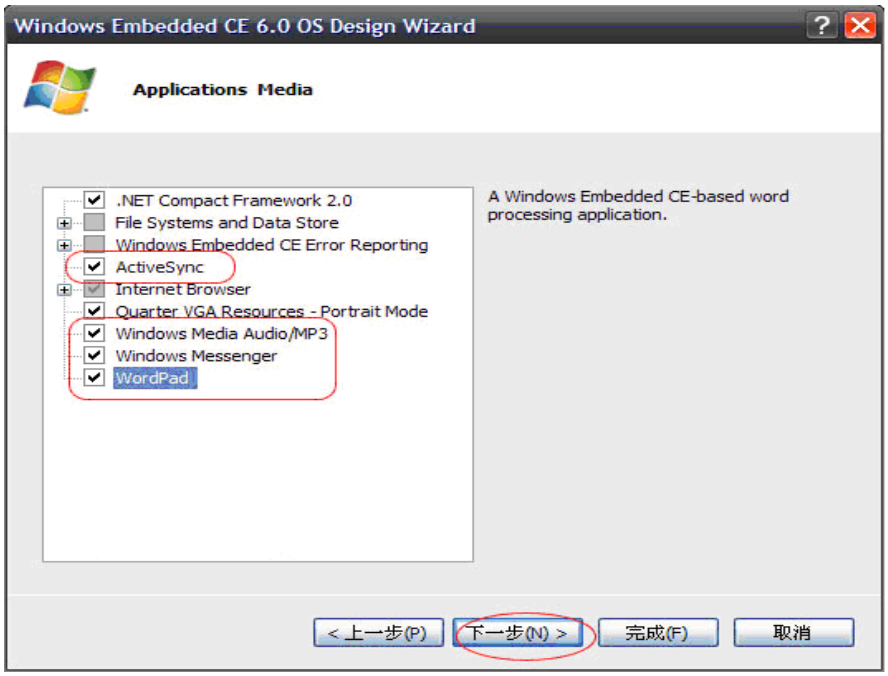
5. 选择 Mobile Handle ，“下一步”



6. 选择一些常用的应用

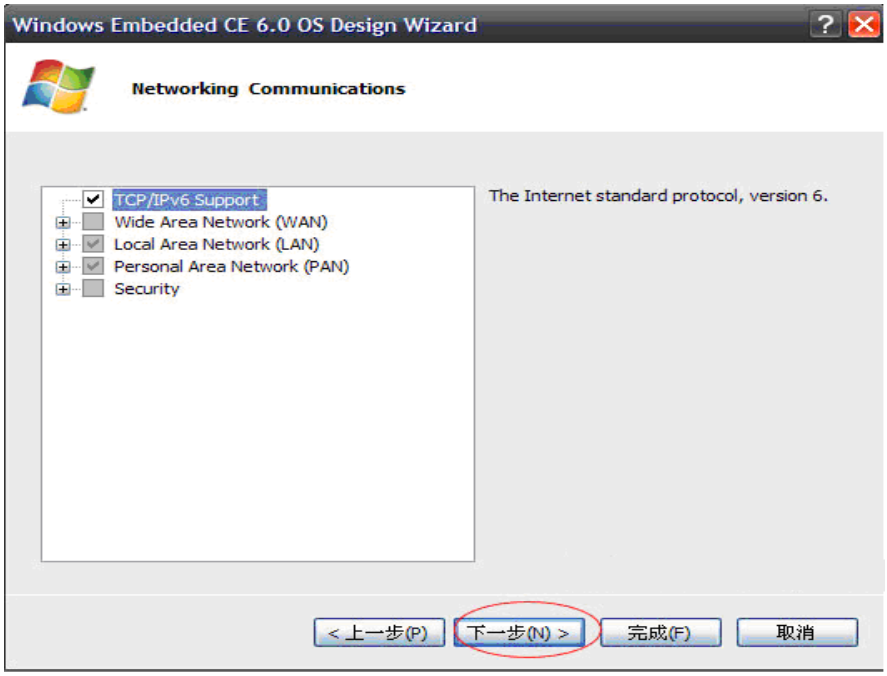
ActiveSync	用于与 PC 连接同步
Internet Explore 6.0	IE6 浏览器
Windows Media Play	微软音乐播放器
Word Pad	微软 Word

初学者建议全选上，以了解它们都有什么功能。“下一步”

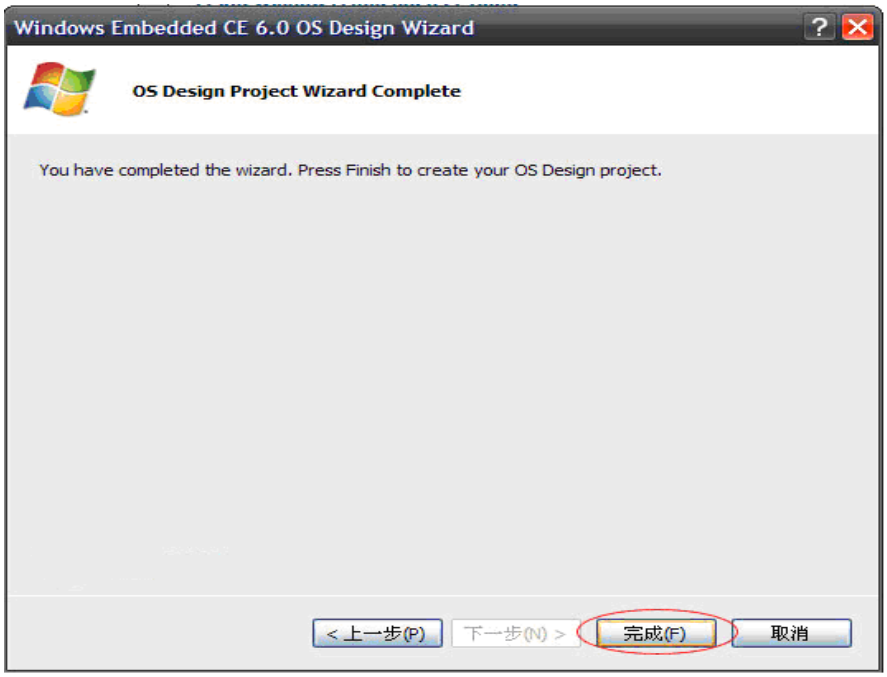


7. 选择 TCP/IP 组件

例如支持蓝牙网络，红外线等，我们选择默认，“下一步”



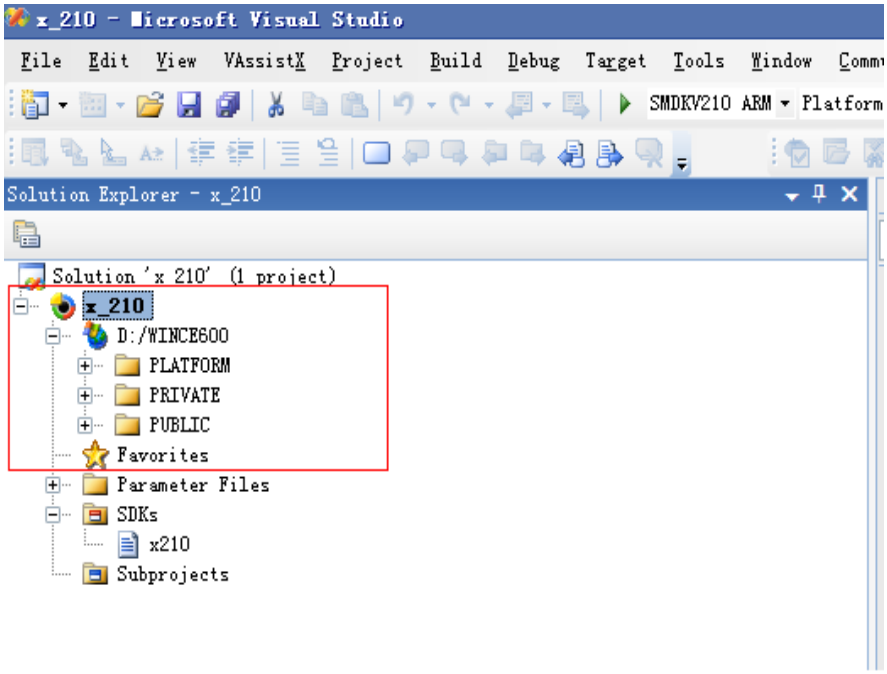
8. 配置完成，点 “完成”



9. 出现一个安全警告窗口，这是因为我们选择的网络协议里，默认使用到了 OBEX 服务，该服务有可能造成隐私内容泄漏，

不用理会它，点“Acknowledge”

10. 到这一步，完成了工程的建立。在 VS2005 里可以看到刚建立的 X210 工程。



2.3 工程属性设置

2.3.1 生成镜像类型选择

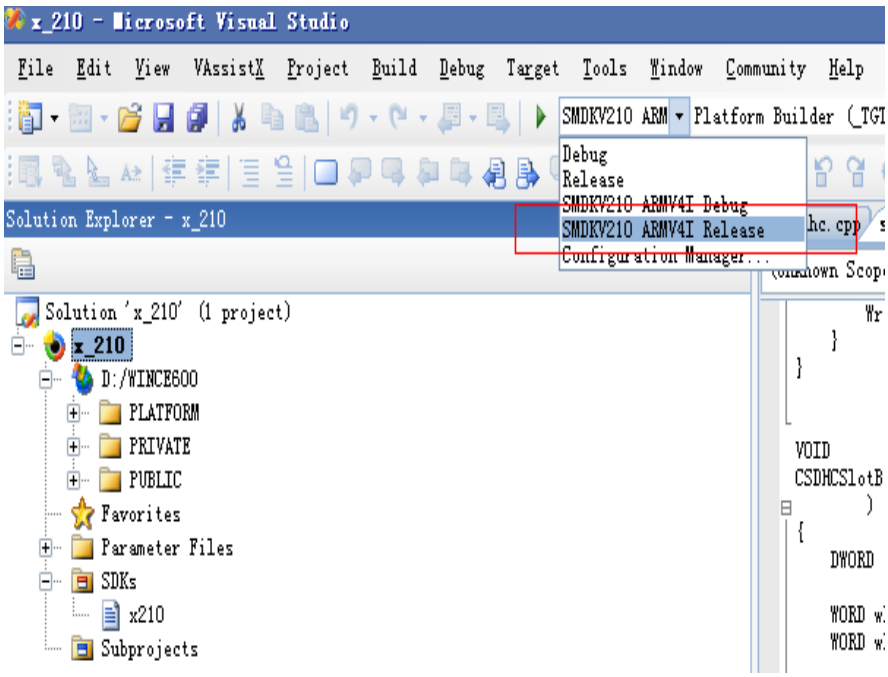
PlatformBuilder 可以生成两种镜像文件：

1. 用于发布给用户，称做 Release 版本，生成后的文件存放在：

D:\WINCE600\OSDesigns\x_210\x_210\RelDir\SMDKV210_ARMV4I_Release 其中“x_210”是我们创建的工程名。

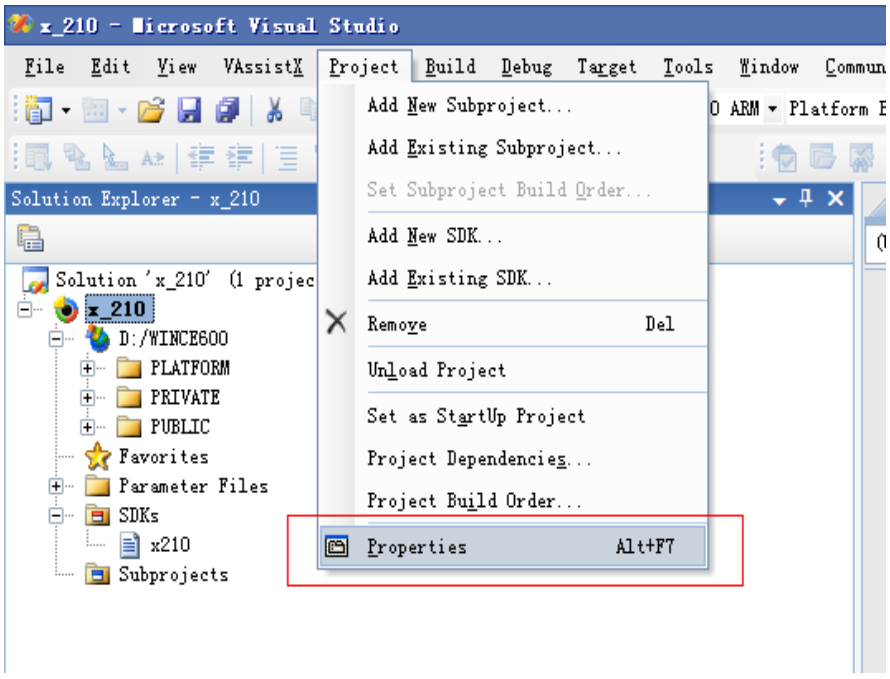
2. 用于调试，称做 Debug 版本，生成后的文件存放在 D:\WINCE600\OSDesigns\x_210\x_210\RelDir\ SMDKV210_ARMV4I _Debug 这个版本类型，会打出很多系统调试信息，当然，NK 也会很大。

3. 在这里我们选择 SMDKV210 ARMV4I Release 类型



2.3.2 属性页面

操作： 项目 > 属性 进入属性页面



1. 选择默认语言

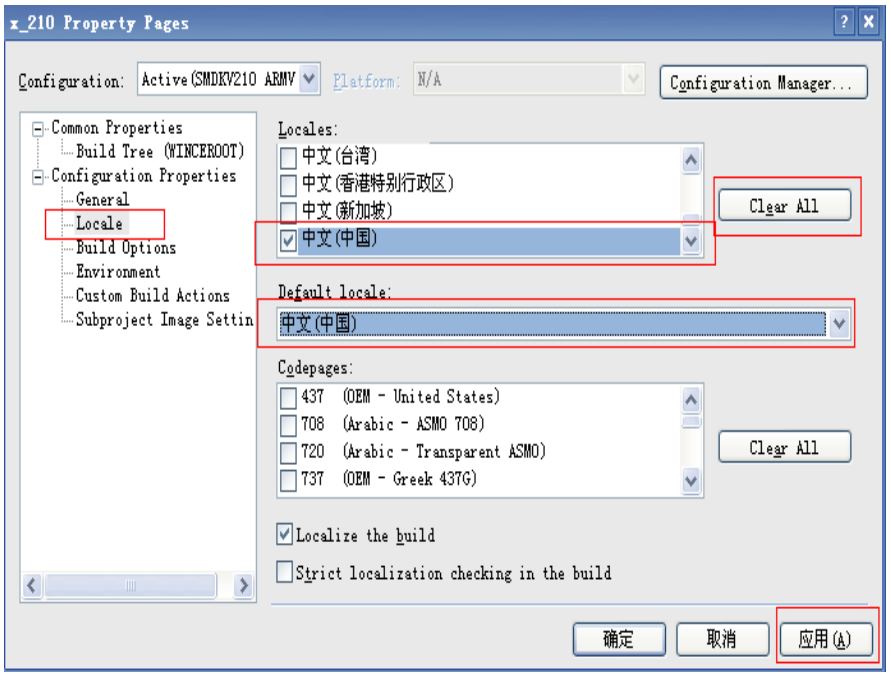
（1）左边树型菜单选“Locale” 打开右边的默认语言设置页。

（2）点“Clear All”按钮

（3）Locales: 选择 中文（中国）

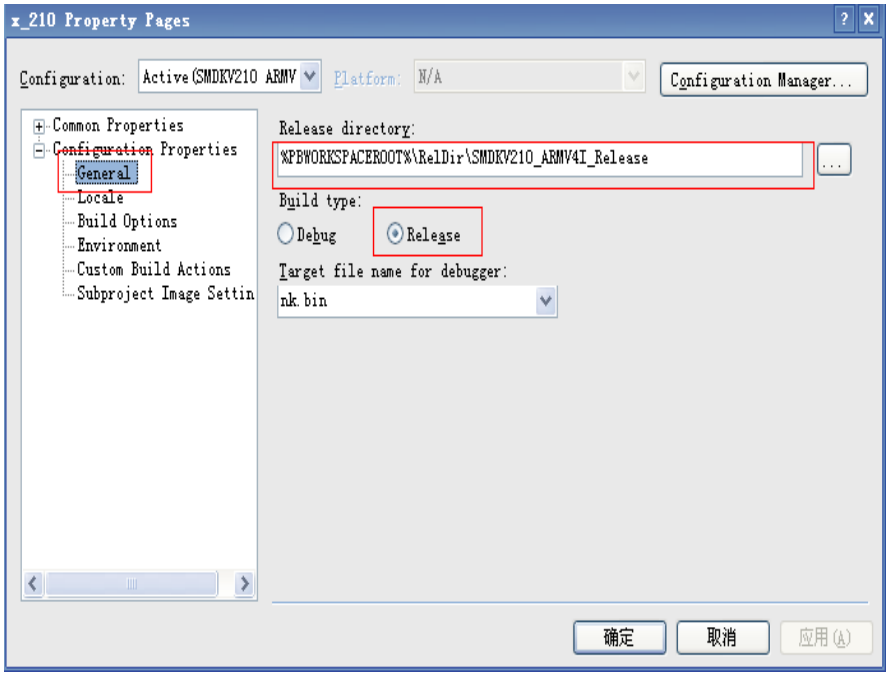
（4）Default locale: 选择 中文（中国）

其他如下图，点“应用”



2. 生成镜像设置

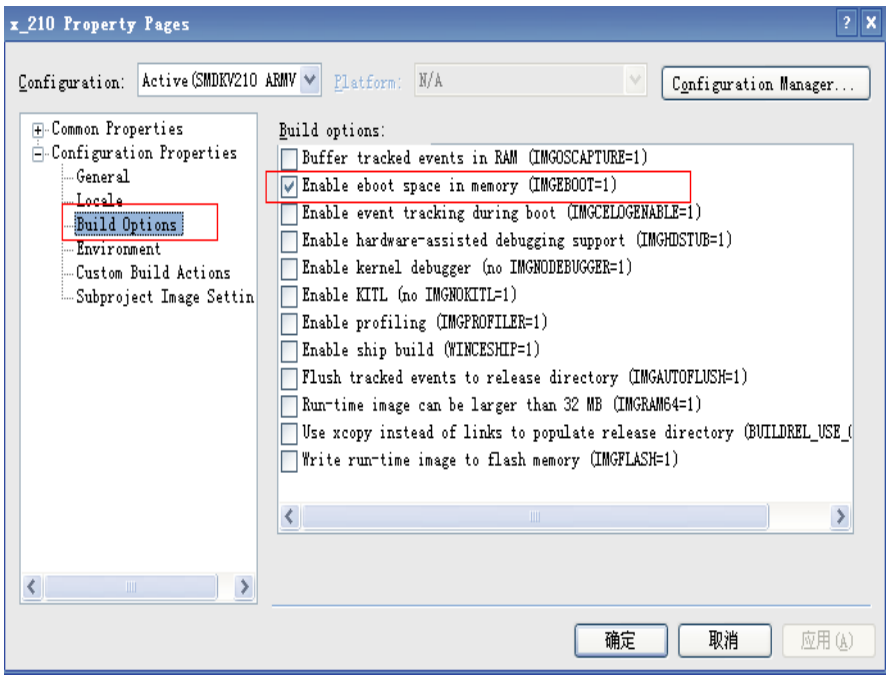
General 设置镜像的路径和类型，还有名字等，我们就按默认的，不用改它。



3. 编译设置

Build Options ,只选 Enable eboot space in memory(IMGEBOOT=1)，去掉其它的几个选项，

因为我们是用 EBOOT 来引导 WINCE 启动的，选这项的意思就是级 EBOOT 保留空间，KITL 是用于内核调试的，这里我们不选。

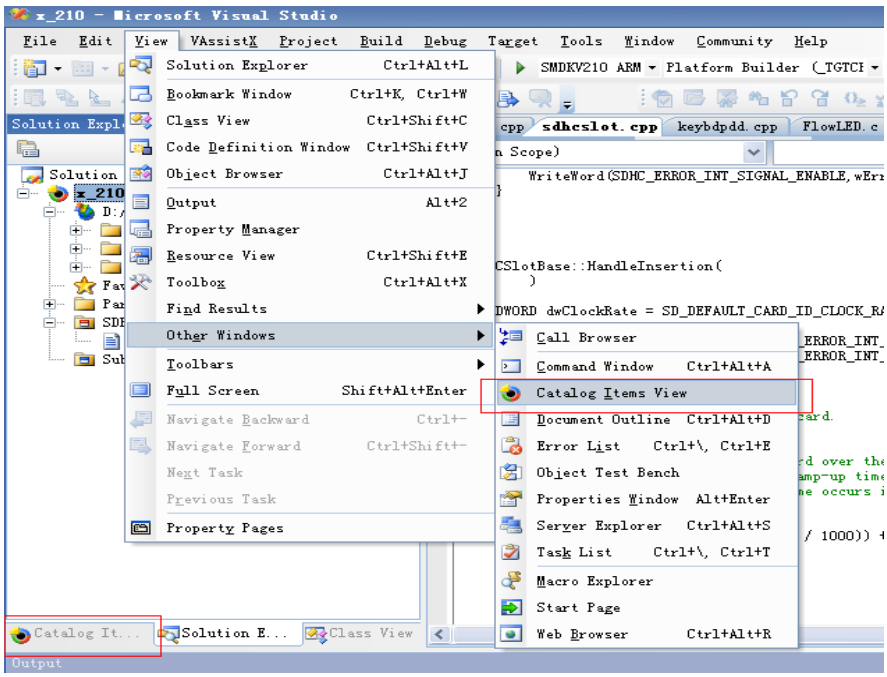


剩下的几项不用理它，“确定”保存并关闭设置页。

2.4 添加系统组件（裁剪系统）

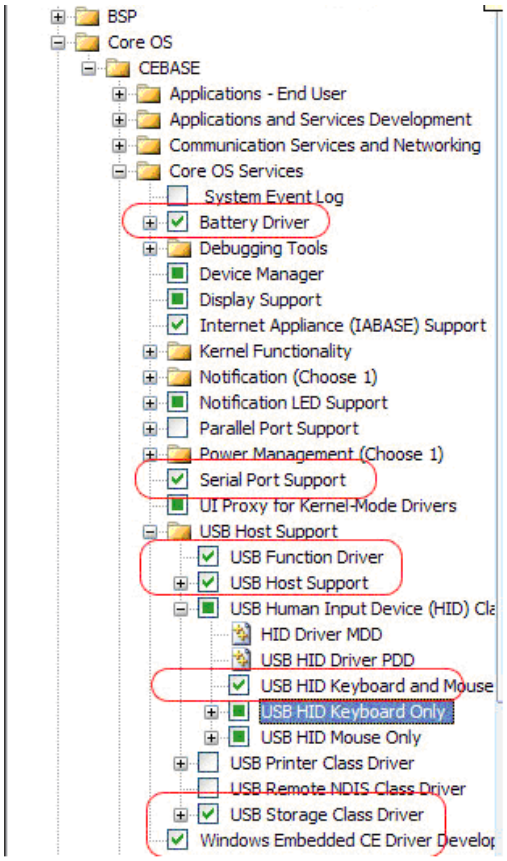
这是系统功能的裁剪，这一步是定制系统的核心内容，决定了我们建立的这个嵌入式系统的功能，通过组件选择，可以生成不同用途的系统为各种应用场合服务，这也是嵌入式系统的精髓所在。

VS2005 窗口右边选择 Catalog Items View，如果没看到这个项，可以在“视图→其他窗口→Catalog Items View”调出该窗口



2.4.1 操作系统核心组件

打开 Catalog Items View 树型目录：Core OS > CEBASE > Core OS Services



我们选择：

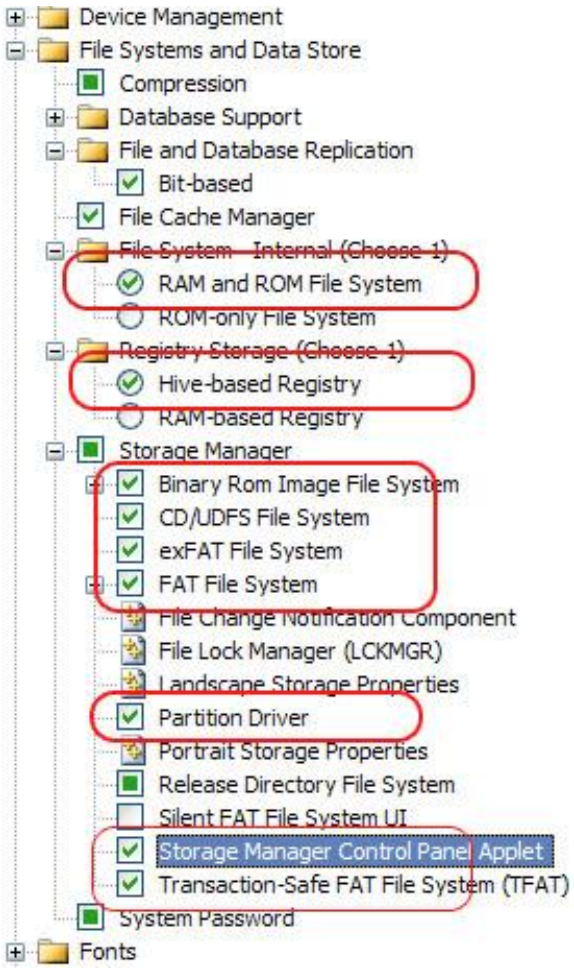
1. Battery Driver (电池驱动)
2. Serial Port Support (串口支持)
3. USB Host Support (USB Host 支持)
4. USB Human Ipnut Device (HID)

   USB HID Keyboard and Mouse

   (USB 鼠标键盘都支持)
5. USB Storage Class Driver

   (USB 大容量存储支持，U 盘支持)
6. Windows Embedded CE Driver Develop Kit support Library (Kit 支持)

其他的默认



2.4.2 文件系统组件

打开：Core OS > CEBASE > File Systems and Data Store

勾选：

1. 基于 RAM 和 ROM 的文件系统

File System->RAM and ROM File System

2. HIVE 注册表，（掉电不丢失）

Registry Storage->Hive-based Registry

3. 支持了 BINFS、CD/UDFS File System、ExFAT、FAT 等文件系统

Storage Manager->

-Binary Rom Image file System

-exFAT File System

-Storage Manager Control Panel Applet

-TFAT File System

-Partition Driver (分区驱动)

等等…

2.4.3 通信组件

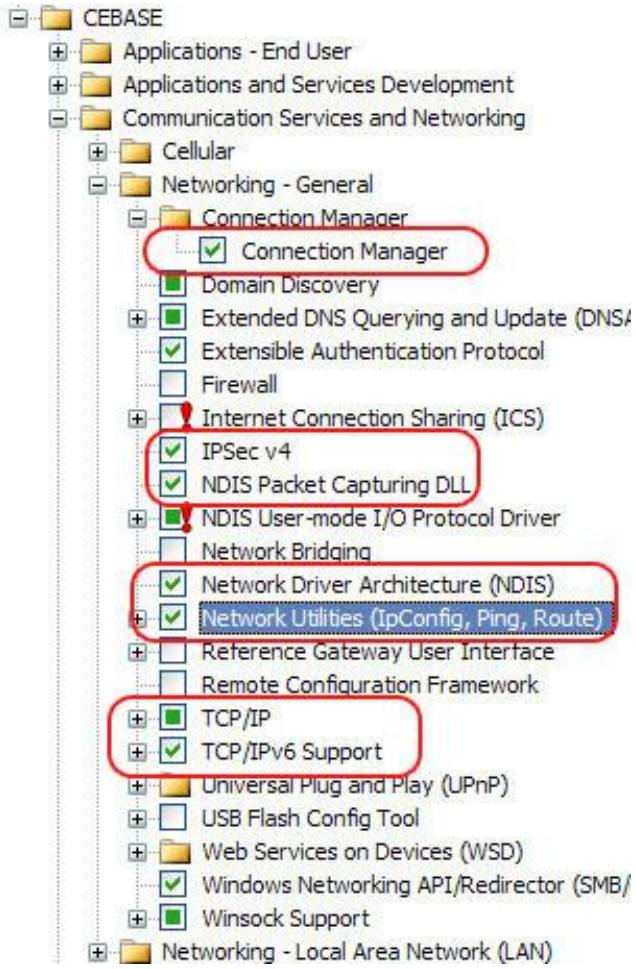
展开：Core OS > CEBASE >Communication Services and Networking

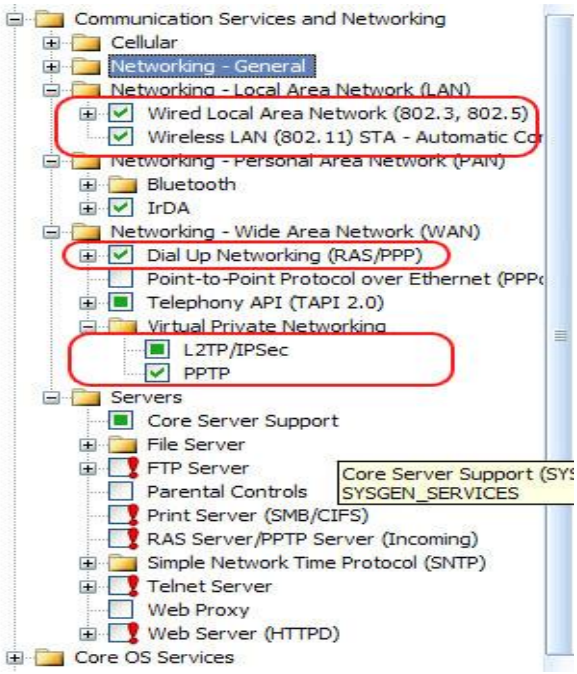
Networking - General

选择： Connection Manager (连接管理器)

各种网络协议：IPSec v4, NDIS，TCP/IP 等

Network Utilities (支持 IpConfig,Ping,Route 命令)

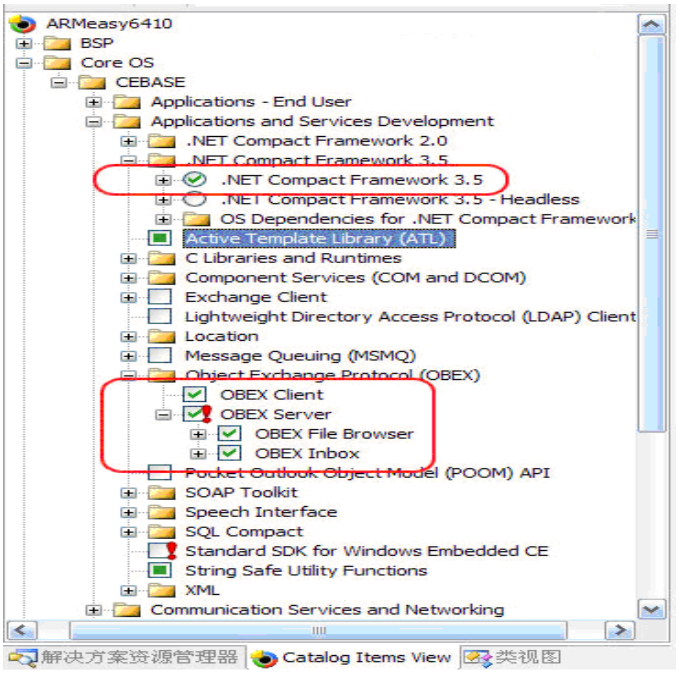




2.4.4 应用和服务开发支持组件

展开目录：Core OS > CEBASE > Application and Services Development

这个节点下面的组件，主要是为了支持开发的组件，例如.net 的 Framework，ATL 等等，不同的项目根据需要支持的类库选择支持，因为占用空间较大，所以如果没有用到的库尽量不要加，这样会使得 NK 变得很大。



2.4.5 终端应用组件

展开目录：Core OS > CEBASE > Applications - End User

ActiveSync: 用于开发板与 PC 机同步，建议选用，否则无法通过 USB 和 PC 同步

CAB Installer/Uninstaller : .CAB 文件的安装和卸载程序 (CE 应用的发布形式)

Games: WinCE 自带的小游戏，如需要自行选择 ，有两个

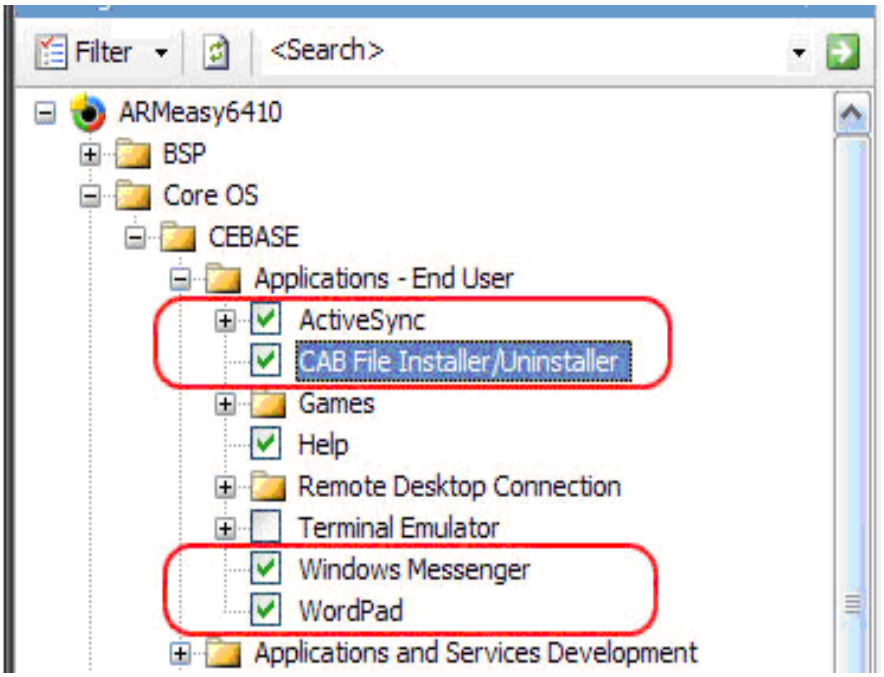
Help : WinCE 的帮助文件

Remote Desktip Connection : 远程桌面程序（有点像 PC 上远程协助那样的东东）

Termial Emulator :终端仿真软件

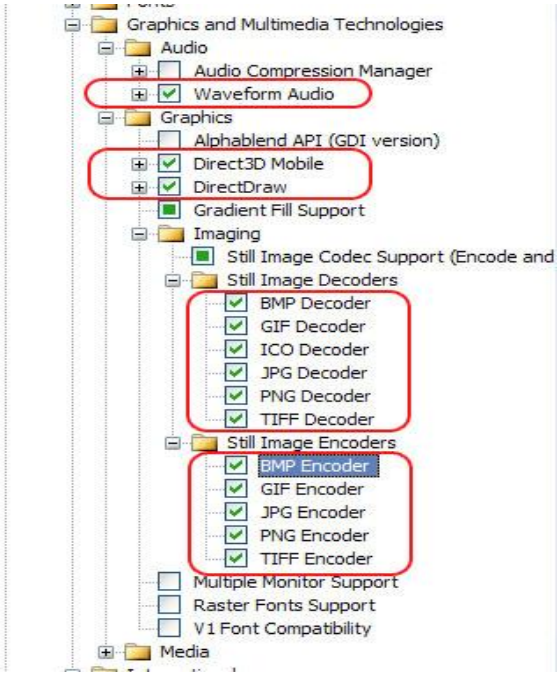
Windows Messaenger: MSN 聊天软件

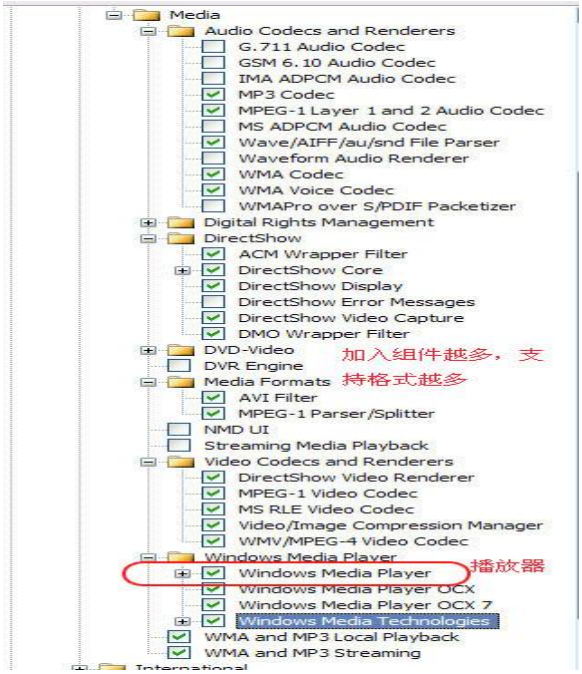
WorPad : Wince 下的记事本



2.4.6 多媒体组件

展开目录：Core OS > CEBASE > Graphics and Multimedia Technologies





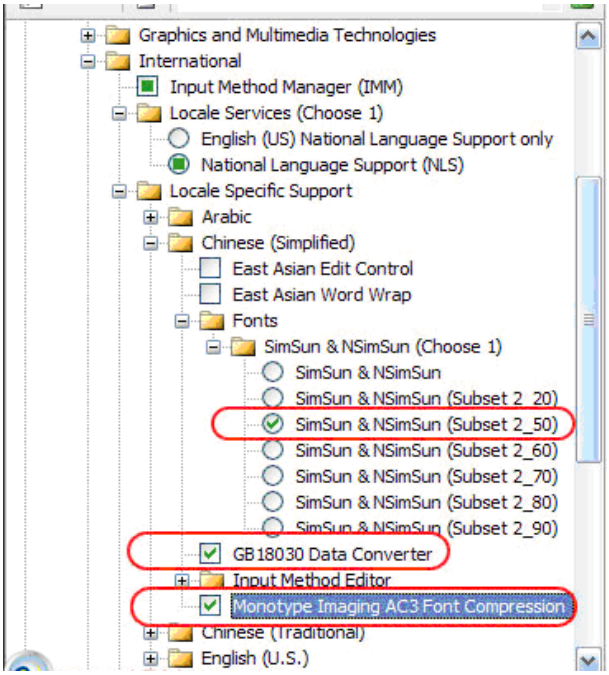
2.4.7 国际语言支持组件

展开目录：Core OS > CEBASE > International

支持中文，要加入中文的字库，SimSun & NSimSun(Subset 2_50)第三项，是一个比较常用的字库，比较小，只有 3MB。

Montype Imaging AC3 Font Compression ,字库压缩格式支持。如果不选，有些中文网页

显示和乱码。



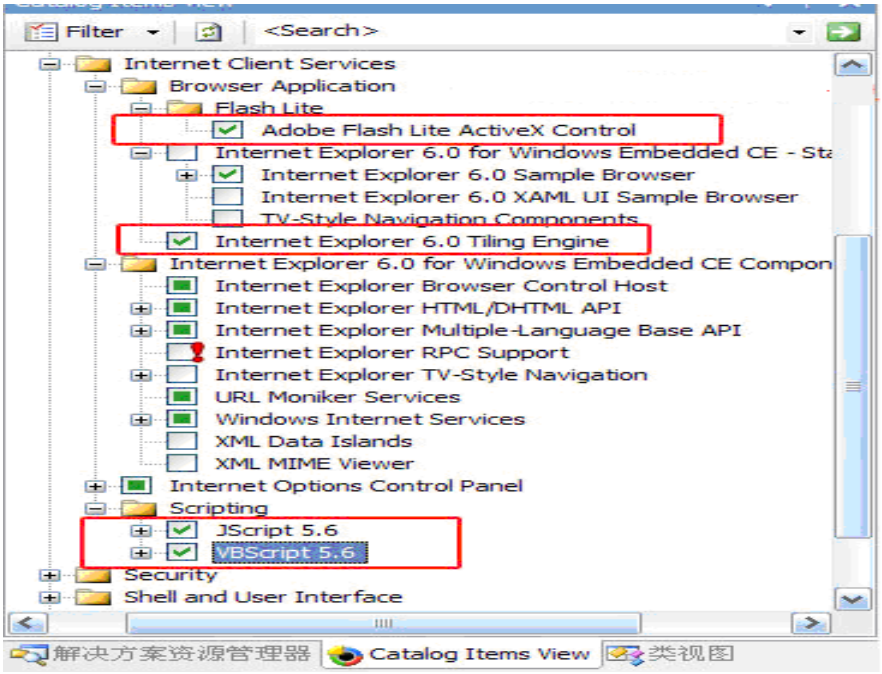
需要注意的是这里的大、小键盘是微软简体中文输入法自带的大、小键盘，却不是使用微软简体中文输入法所必须的，也可以不选，而使用 Shell Software-based Input Software Input Panel User Interface and User Interface Panel (SIP) (Choose 1 or more)下的默认的标准大、小键盘。

两者的区别在于：如果使用自带的键盘就是使用双拼输入汉字，如果使用标准的键盘就是使用全拼音输入汉字，但只能拼写一部分，有些字不能拼写。

2.4.8 网络服务相关组件

展开目录：Core OS > CEBASE > Internet Client Services

Adobe Flash Lite ActiveX Control 安装 R3 后才有的，是内嵌在 IE 的 Flash 插件。

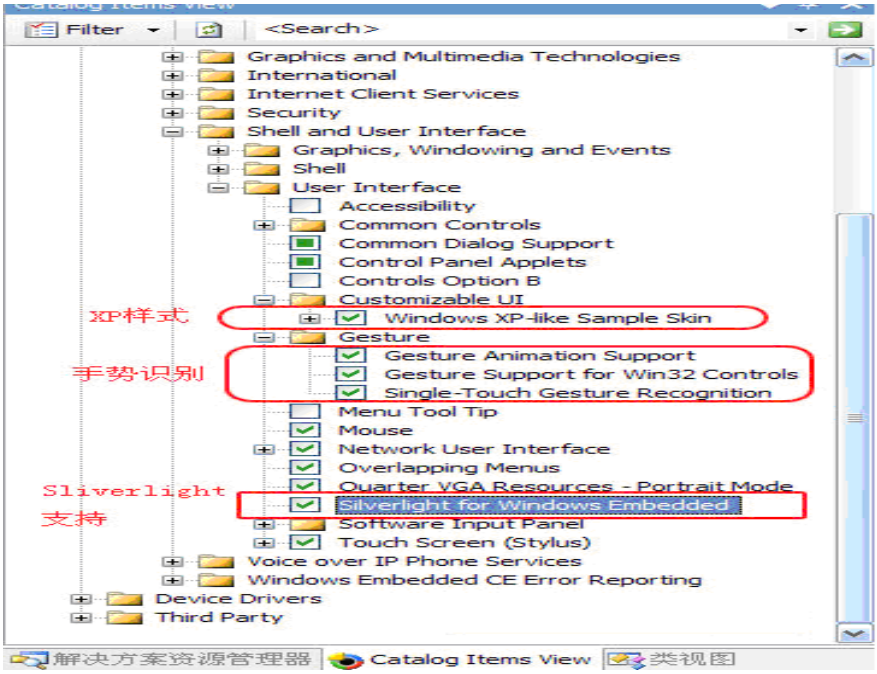


2.4.9 Shell 界面相关组件定义

展开目录：Core OS > CEBASE > Shell and User Interface

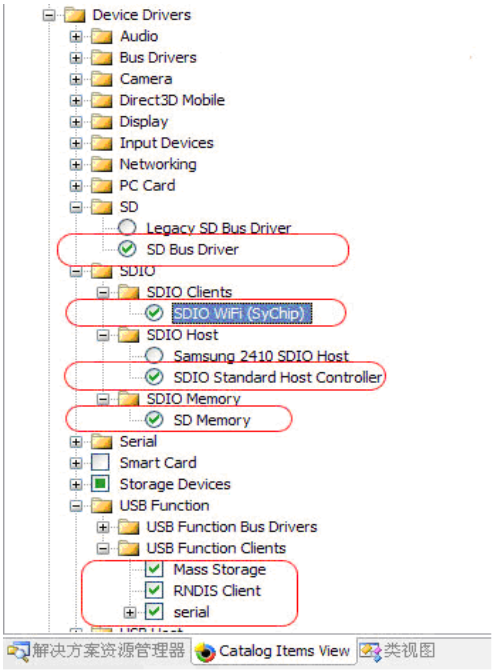
Gesture (手势识别,安装 R3 之后新加的功能)

Sliverlight for Windows Embedded （Sliverlight 支持，安装 R3 升级后才加的功能）



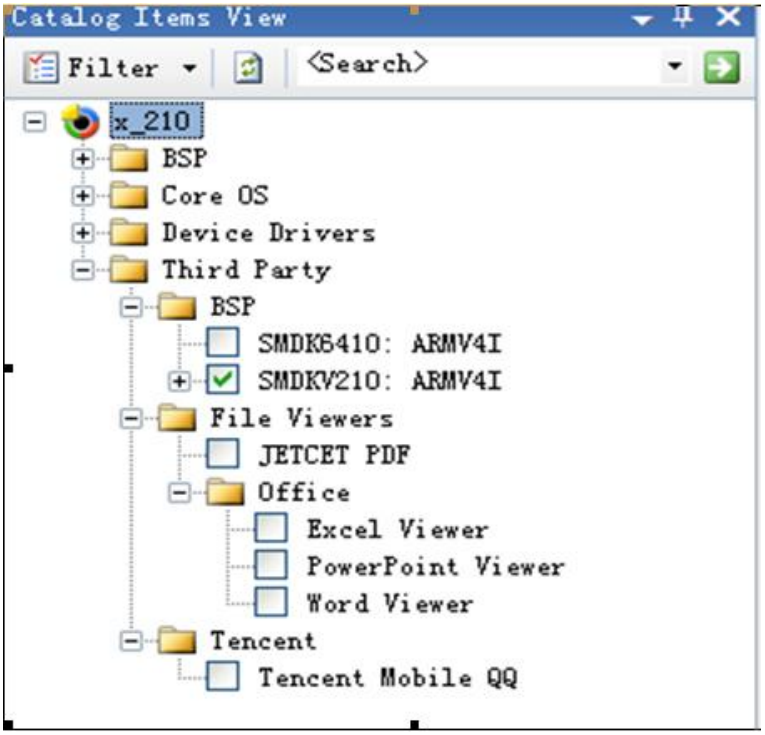
2.4.10 通用驱动的选择

展开目录：Device Drivers



2.4.11 第三方组件选择

主要就是一些 R3 添加的 OFFICE 组件，腾讯 QQ 等，



第3章 编译 WinCE6.0 内核

3.1 编译内核步骤：

如果你是 WinCE6.0 的初次使用者，请跟着下面的步骤操作：

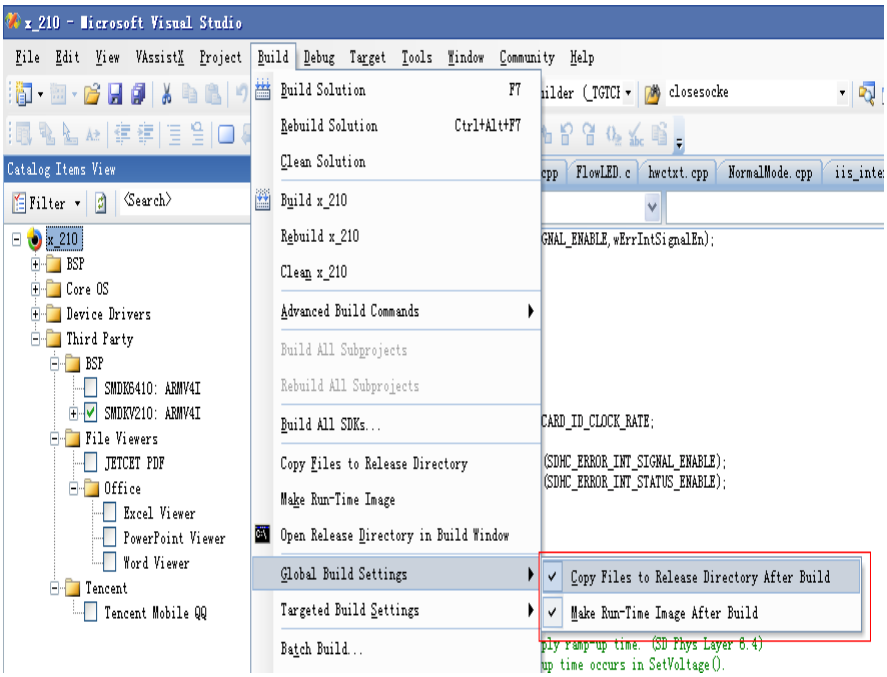
3.1.1 设置编译选项

1. 全局编译设置：

Build -> Global Build Settings

勾选 Copy Files to Release Directory After Build

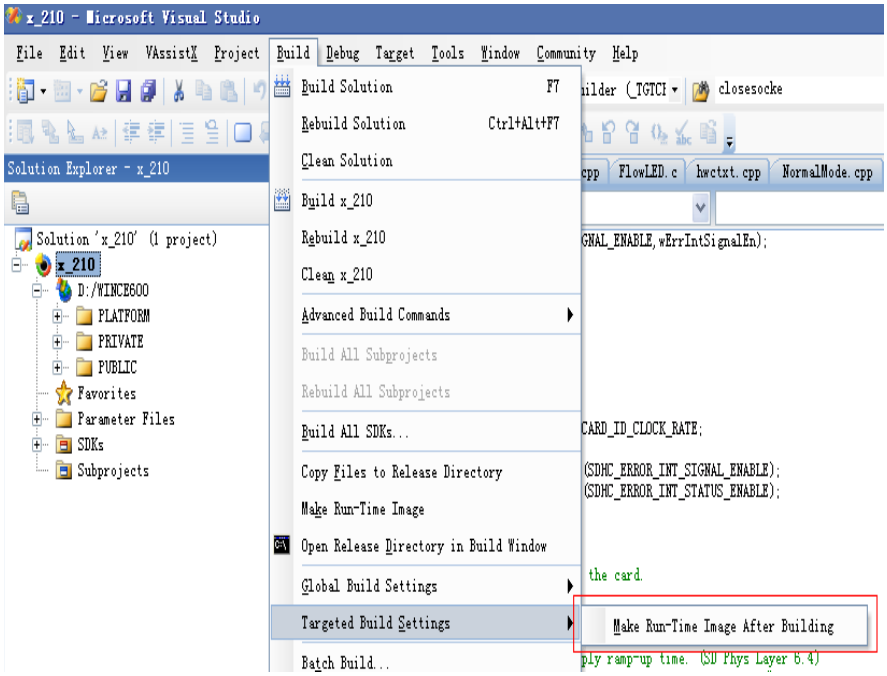
Make Run-Time Image After Build



2. 目标编译（Tarteted Build）设置

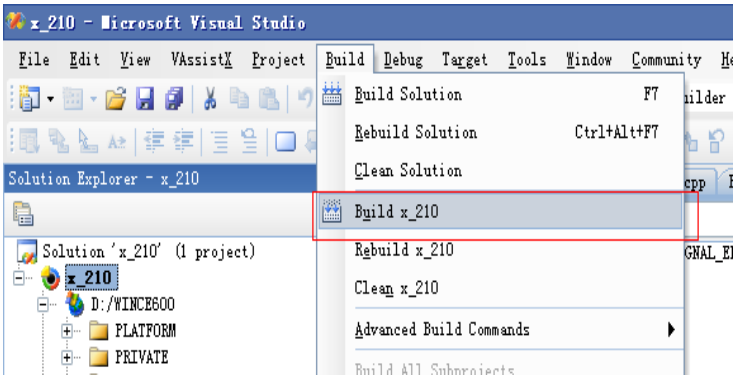
Build -> Targeted Build Settings 去掉 Make Run-Time After Building

这样是为了方便调试驱动，因为在后面的驱动开发过程中，要常先编译，看有没有语法错，很多情况下编译是不用马上生成镜像的。



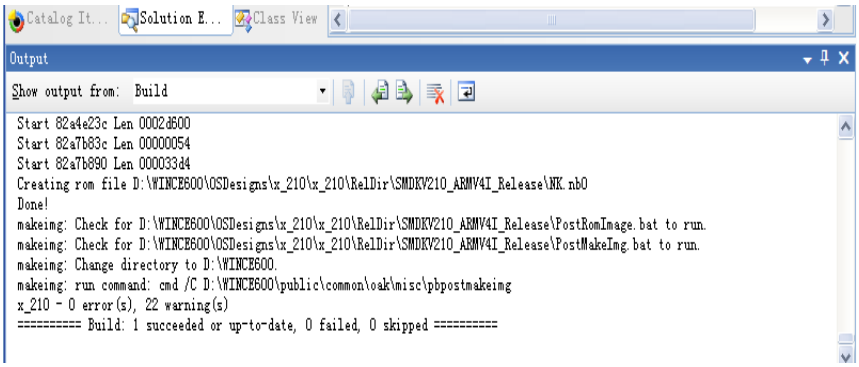
3.1.2 菜单操作：“Build-> Build x_210”

注意：我们假设你建立的工程名字是“x_210”，开始编译，



可以看到右下角正在编译的箭头在动：

3.1.3 编译完成



到这里，就实现了编译系统镜像的工作了。这时在工作目录的 Release 下会找到 NK.bin，xip.bin。例如：D:\WINCE600\OSDesigns\x_210\x_210\RelDir\SMDKV210_ARMV4I_Release

目录。

3.2 编译方式及适用场合

3.2.1 新的工程和工作环境

如果是新建立的开发环境，第一次建立的工程，也是第一次编译，则要求全编译，也就是 CE5.0 里面的“Build and sysgen”，的操作，但是 CE6.0 里，只需要“生成→生成【工程名】”这个步骤，这就包含了上面的操作，例如我们建立的工程名是：x_210，则菜单操作“Build →Build x_210”

1. 新建的工程第一次编译

建议选择两步：

“Build”→“ Advanced Build Commands ” →“ Clean Sysgen ”

“Build → Build【工程名】”这个步骤

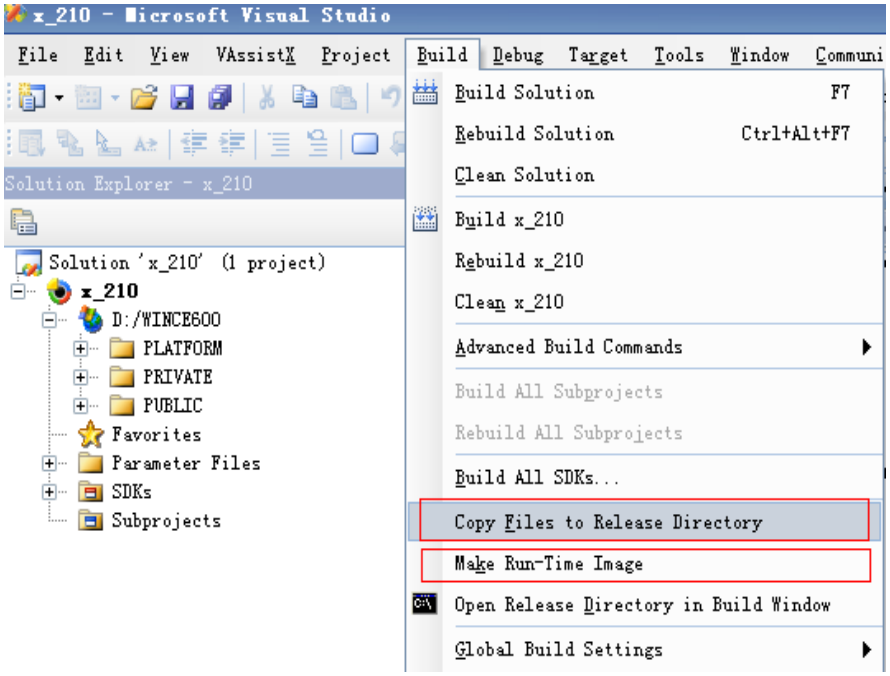
适用场合：新建的工程，第一次编译，但是开发环境已经编译过其他的工程

2. 改动了 platform.bib 或 platform.reg 文件

比如修改了 reg 里的内容，想编译到镜像里看效果，此时有更快的编译方法

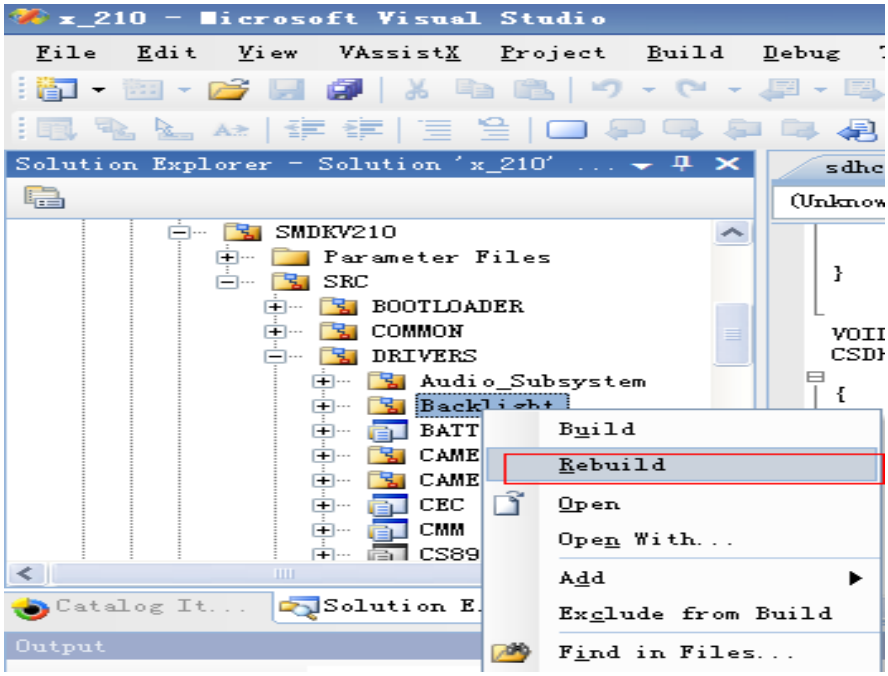
执行菜单 Build \ Copy File to Release Directory

执行菜单 Build \Make Run-Time Image

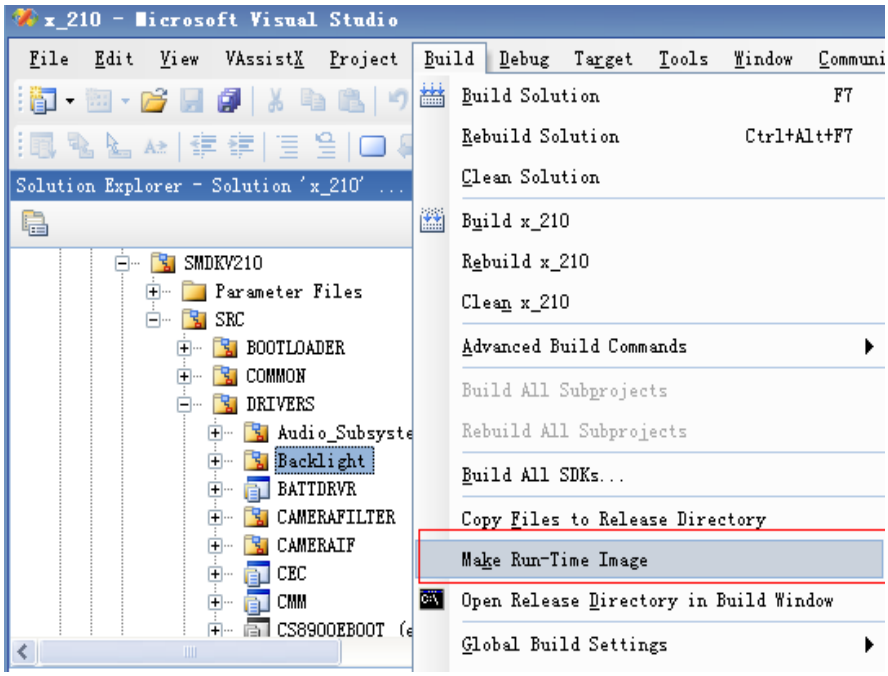


3.2.2 BSP 中部分改动

如果只改动了 BSP 中某个部分的内容，可以使用目标编译法，这样会更快。比如我们只改动了 Backlight 的内容，然后想编译进系统看效果。方法：在解决方案资源管理器里，鼠标右键选择 Backlight 文件夹，弹出菜单选择 Rebuild。



编译完成后，生成镜像。

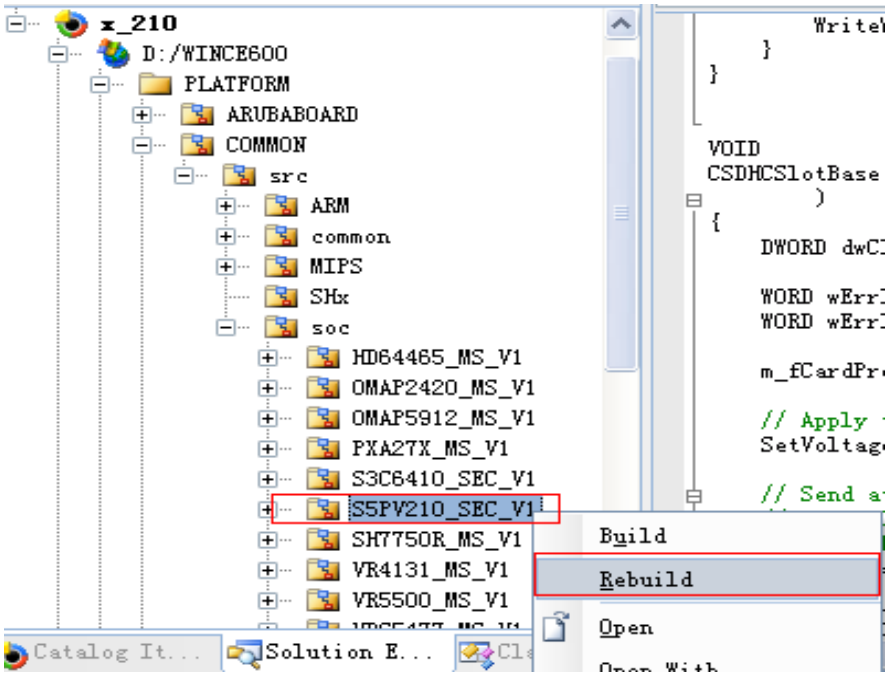


当然，如果你在 Targeted Build Settings 里面已经勾选了 Make Run-Time Image After Building,那就可以省略这一步了。

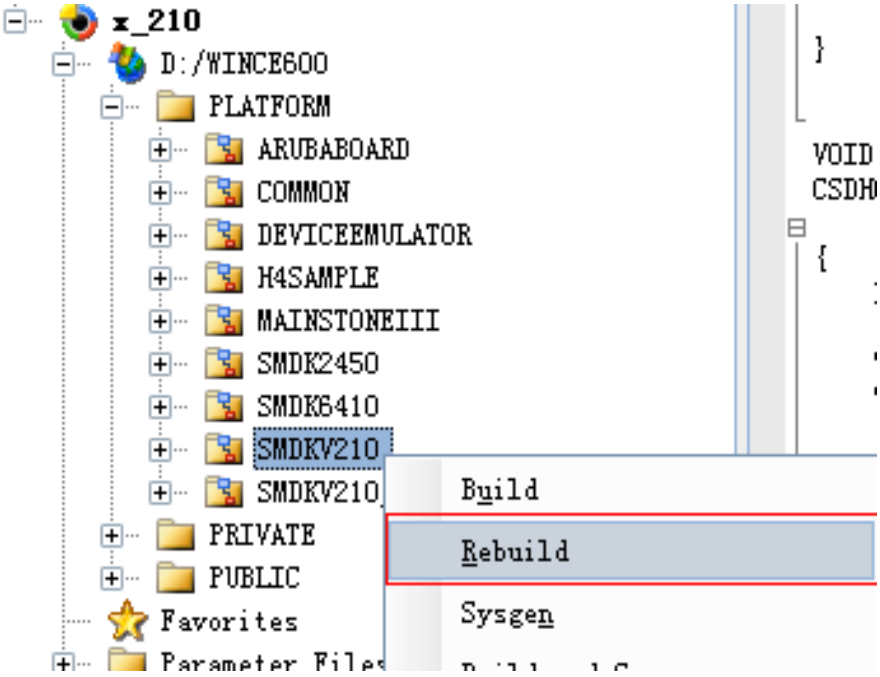
3.2.3 改动了 COMMON 里 OAL 的内容

目录 PLATFORM\COMMON\SRC\SOC\ S5PV210_SEC_V1，此时编译要分两步来做

1. 先目标编译 OAL 文件夹



2. 再编译 SMDKV210 文件夹



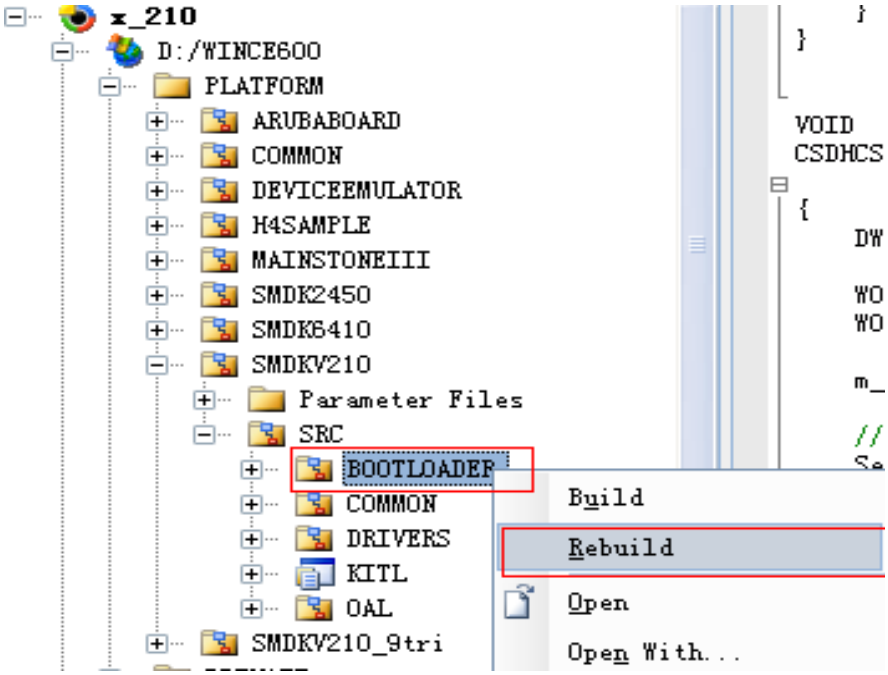
3. 然后生成镜像

Make Run-Time Image

3.2.4 BOOTLOADER 编译

BOOTLOADER 编译有点特别，因为它不是 NK 的一部份，编译它不用生成系统的 Image。

只需要目标编译 BOOTLOADER 文件夹就行了，生成 Eboot 的相关镜像。



3.2.5 编译有 Lib 依赖的驱动

有一些驱动是分开成 lib 和 dll 的，最终生成给系统使用的是 dll，但是，在编译 dll 时要用到 lib,例如 6410CE6.0 的三星 BSP 包中，camera 驱动，它生成 dll 时要使用 CAMMDD.lib，CAMPDD.lib，s3c6410_camera.lib 三个库，所以，在编译 dll 之前，要先编译以上三个 lib.

如何控制 PB 的编译顺序呢？

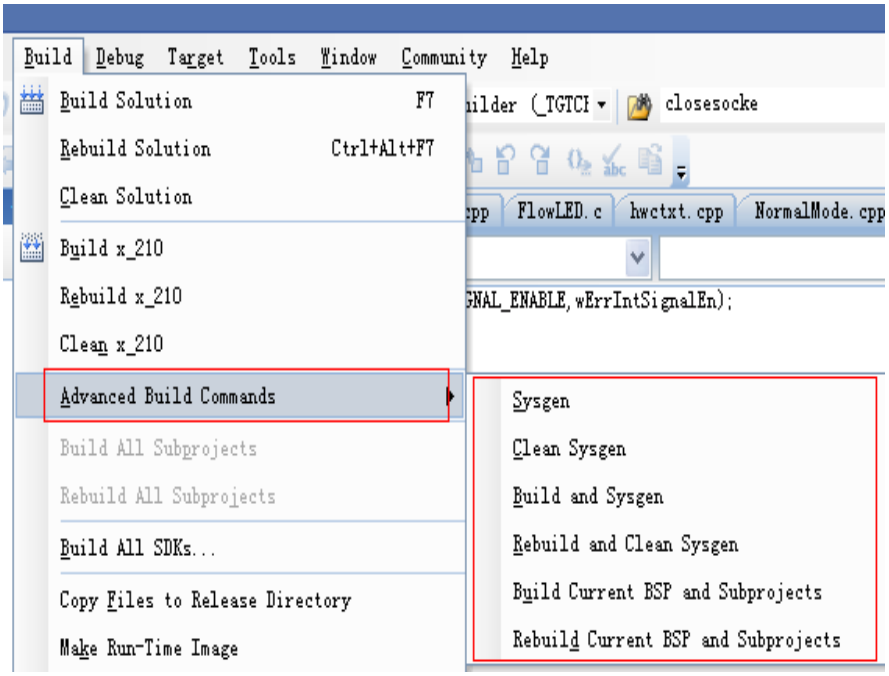
编译顺序：根据驱动文件夹里的 dir 文件的顺序如下：

```
DIRS_CE=\
        CAMERA_MDD_PM     \
        CAMERA_PDD        \
        S5K4BA_MODULE     \
        S5PV210_camera    \
        DLL
```

编译顺序：CAMERA_MDD_PM→CAMERA_PDD→S5K4BA_MODULE→S5PV210_camera → DLL，或者，我们鼠标右键，按上面顺序逐个 Rebuild。

3.3 高级编译菜单说明

在 VS2005 的"Build"菜单中，有一个“Advanced Build Commands”，其中有如下关于 WinCE 6.0 的编译选项：



3.3.1 SYSGEN

system generation 根据用户的设置,产生相应的头文件,库文件等。 比如：在\PUBLIC\COMMON\OAK\LIB\ARMV4I\DEBUG\下面有库文件 wavemdd.lib，创建 testbuild (基于 BSP device emulator) 工程后,sysgen 后，在\testbuild\testbuild\Wince600\DeviceEmulator_ARMV4I\cesysgen\oak\lib\ARMV4I\debug 下面也有 wavemdd.lib ，同时在\testbuild\testbuild\Wince600\DeviceEmulator_ARMV4I\cesysgen 下面也有很多头文件和库文件 。 用户在 VS2005 上的修改，比如选择上 ActiveSync (SYSGEN_AS_BASE),在 PBInitEnv.bat 里面就会加上一句：

```
set SYSGEN_AS_BASE=1
```

选择 Project-> testbuild property -> Locale 或者 Build Options ，做的修改也反映在 PBInitEnv.bat 里。 sysgen 会根据很多.bat 文件 (会处理依赖关系)，比如 winceos.bat .

相当于执行命令"blddemo -q”,一般第一次编译或者是改变了"Catalog"中的 item 的时候，就用这个命令。

3.3.2 Clean Sysgen

相当于执行命令"blddemo clean -q"，按照文档上的说明，当修改了%_WINCEROOT%\Public\CEBASE\OAK\Misc\Cesysgen.bat 的时候，或者改变了以 SYSGEN，BSP 为前缀的环境变量的时候，需要使用这个来编译。

一般只有第一次创建完工程的时候，才需要用"Sysgen"命令，以后只要是改变了 SYSGEN 为前缀的环境变量的设置或者是"Catalog"中的 item，就需要使用"Clean Sysgen"，而改变了以 BSP 为前缀的环境变量要看具体情况，也不一定就要用"Clean Sysgen"。

3.3.3 Build and Sysgen

相当于执行命令"blddemo",当改变了\public 目录下的代码,比如你改了 WinCE 的 patch,

你就需要用这个了。

3.3.4 Rebuild and Sysgen Clean

相当于执行命令"blddemo clean cleanplat -c"，清除上一次编译生成的所有文件，然后重新编译\public 目录和你的工程。

3.3.5 Build and Sysgen Current BSP

相当于执行命令"blddemo -qbsp"，仅编译\platform 目录下的代码。所以当改变了\platform 目录下的代码的时候或者说改变了 BSP 的代码的时候，可以用这个来编译。

3.3.6 Rebuild and Clean Sysgen Current BSP

相当于执行命令"blddemo -qbsp -c"，相当于完全重新编译\platform 目录下要编译的代码。

3.4 定制编译

也叫加载/不加载某个驱动的编译方法，在开发产品的时候，有可能一个 BSP 要兼容不同的硬件版本，针对一系列的产品，这些产品之间就是一些功能裁剪，比如第一款产品是 R01，是一个有 WIFI 功能的 MID，而 R02 是一个把 R01 的 WIFI 换成 3G 上网的机器，功能的一些变换，推出一些有针对性的产品，在开发的时候，研发人员一项一项功能往系统里加，然后，针对不同的型号，做一些定制编译，例如 R01 编译把 WIFI 功能加入镜像，而 R02 是不编译 WIFI，只编译 3G 进入镜像。

方法：在 BSP 目录中的（D:\WINCE600\PLATFORM\SMDK6410）找到 SMDKV210.bat 文件，用文本编辑器打开，修改此文件的一些项，可以允许某些驱动不参与编译或让其参与编译。

例如：

```
set BSP_NOTOUCH=
```

这是表示编译 TOUCH 驱动到系统中，

```
set BSP_NOTOUCH=1
```

这表示去掉 TOUCH 驱动。

还有 Set BSP_NOCAMERA= 等等关键字，可以通过这样的关键字，去作镜像功能裁剪，当然，这些开关项是在驱动开发时加入的，如果我们后面要开发新的驱动，比如我们要开发 FM 驱动，可以在这里加入

```
Set BSP_NOFM=
```

然后在 platform.reg 和 platform.bib 里作判断这个开关量，如果成立，把相关注册表和 DLL 文件包含进系统。

注意，一般这个 BAT 文件有个使用习惯，前面有个 NO 的，表示空为有效，例如 BSP_NOTOUCH=，没有的 NO 的当“=1”时有效，例如

```
set BSP_NOTOUCH=      //表示把 TOUCH 驱动加入系统
set BSP_AUDIO_IIS=1   //表示编译 IIS 的音频驱动进入系统
```

设置好之后，保存，然后双击 SMDKV210.bat 运行一下。才会生效，再去编译系统，如果要使用的驱动已经编译成 DLL 了，则直接 Make Run-Time Image 就可以了。

3.5 把程序或文件编译进 NK

因为嵌入式系统中，每次开机，都是从非易失性存储介质（例如 flash）里拷贝镜像到内存（RAM）去运行的，所以有时不小心删除了一些系统原有的文件，或是软件，都不要紧，下次开机，它又会出来了。有些时候，我们会想把一些重要的软件和文件编译进 NK 里面去，以免用户误操作影响机器功能。假如我们想把 GarField.wmv 这个文件编译进系统镜像里。

3.5.1 方法一：

1. 把 GarField.wmv 拷贝到 BSP 的 FILE 文件夹（D:\WINCE600\PLATFORM\SMDKV210\FILES）
2. 修改 BSP 里 platform.bib 文件，在 FILES 段添加内容

GarField.wmv	$(_FLATRELEASEDIR)\GarField.wmv	NK	U

注意：

U 是表示用户模式；

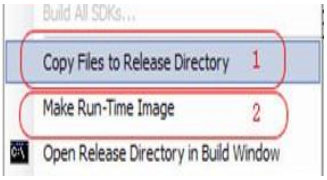
NK 表示 放在 NK 里；

（_FLATRELEASEDIR） 表示工程的 release 目录，编译时在那里提取这个文件。

3. 打开编译菜单，

执行：Copy Files to Release Directory 把 GarField.wmv，platform.bib 拷贝到工程的 Release 目录

再执行：Make Run-Time Image 编译镜像。



开发板电源打开，进入 WINCE，就能在“我的设备”→“Windows”目录下找到它了。

3.5.2 方法二：

我们也可以直接修改 Release 目录的 platform.bib 文件，Release 目录：D:\WINCE600\OSDesigns\x_210\x_210\RelDir\SMDKV210_ARMV4I_Release 把

GarField.wmv	$(_FLATRELEASEDIR)\GarField.wmv	NK	U

改成

GarField.wmv	$(_TARGETPLATROOT)\FILES\GarField.wmv	NK	U

这样，编译时就不用执行 Copy Files to Release Directory 命令，它也能找到 GarField.wmv 文件

第4章 烧写 WinCE6.0 镜像

4.1 烧写注意事项

4.1.1 必要的硬件连接

1. USB 数据线(数据线小口接 x210-II 的 mini USB 接口,另一端接 PC 机的 USB 接口)
2. 串口线 (一端接 x210-II 的 COM0,另一端接 PC 机的 COM 口)
3. 9V 直流电源

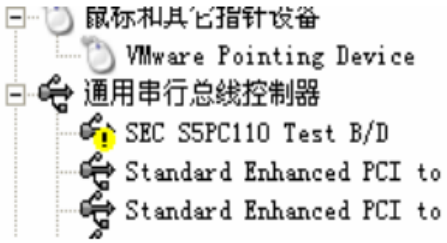
4.1.2 必要的软件烧写工具

DNW.exe 通过 DNW.exe 下载镜像文件

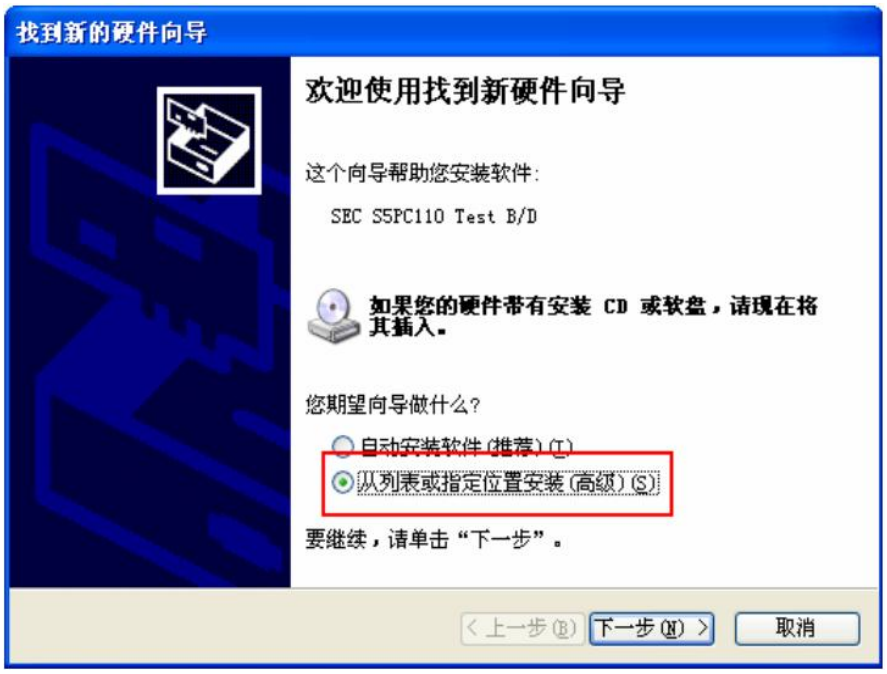
4.1.3 安装必要的驱动

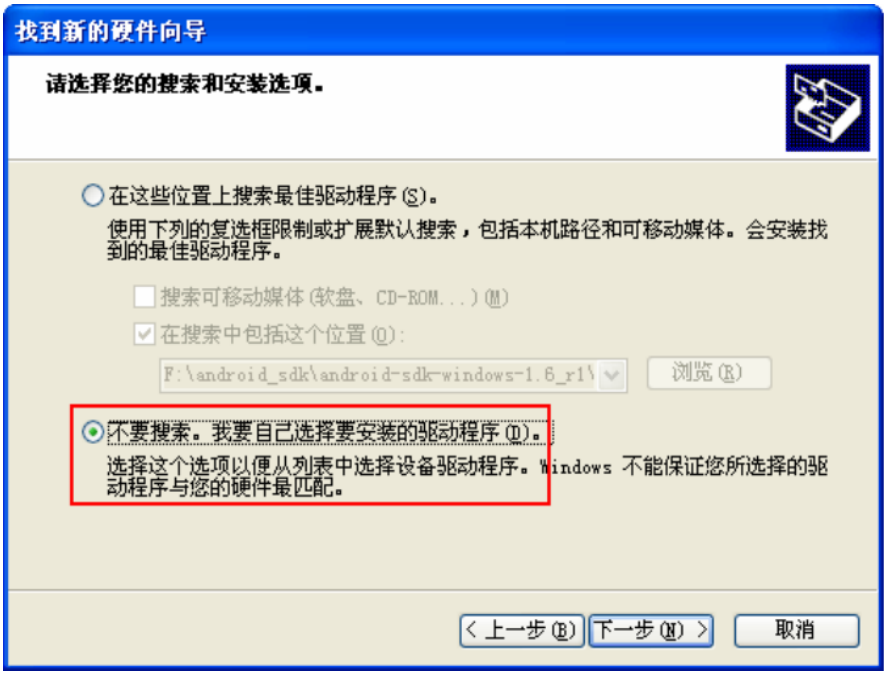
下载前需要你的 PC 上安装 USB 下载驱动程序,正常情况下,x210-II 通过 USB 数据线连接到你的 PC 机, x210-II 上电后,PC 会自动发现新硬件并要求安装驱动.如下图:

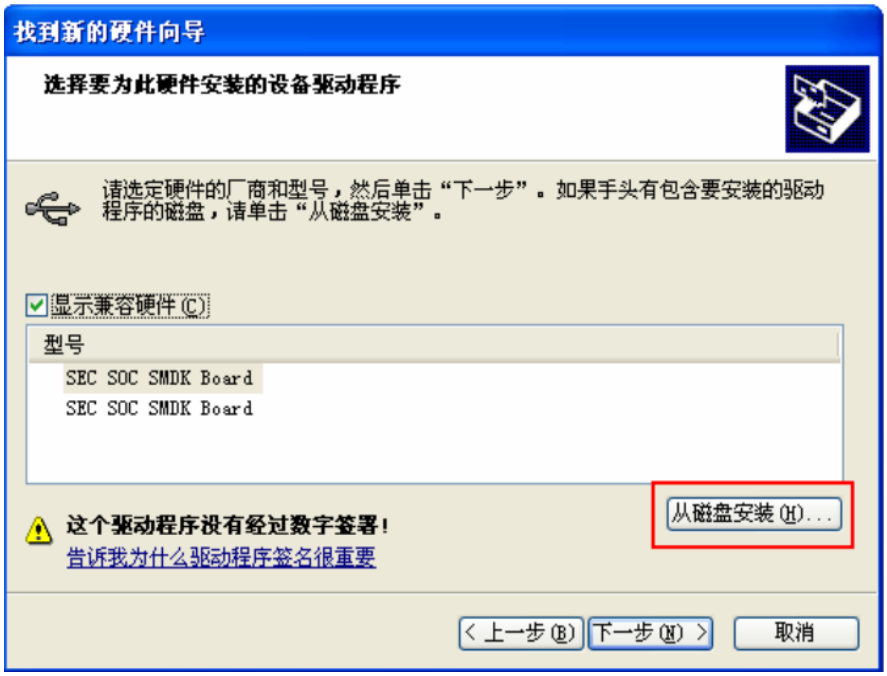
PC 设备管理器中可以看到:

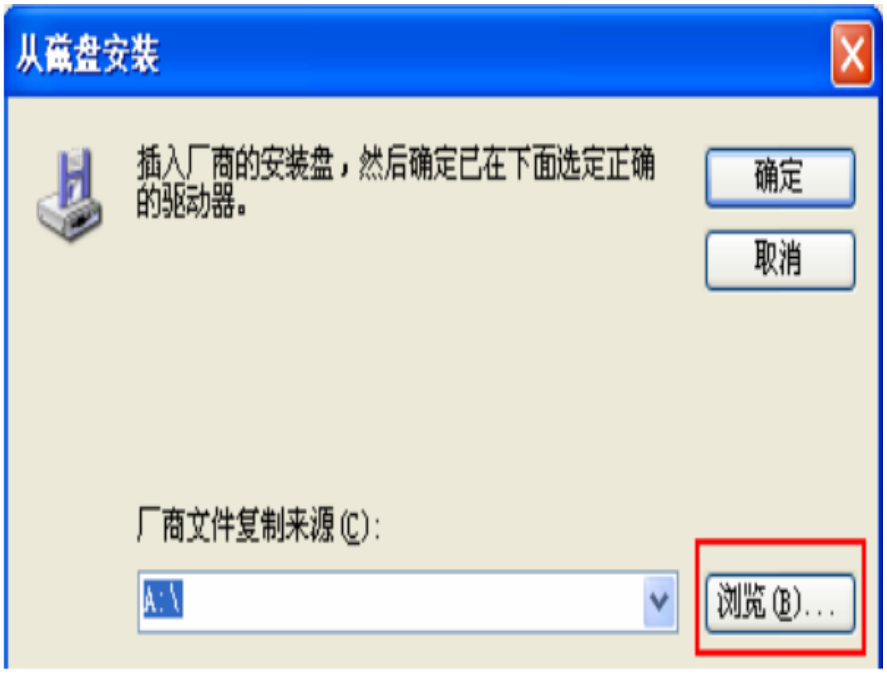


根据弹出的安装向导,将必要的驱动装完

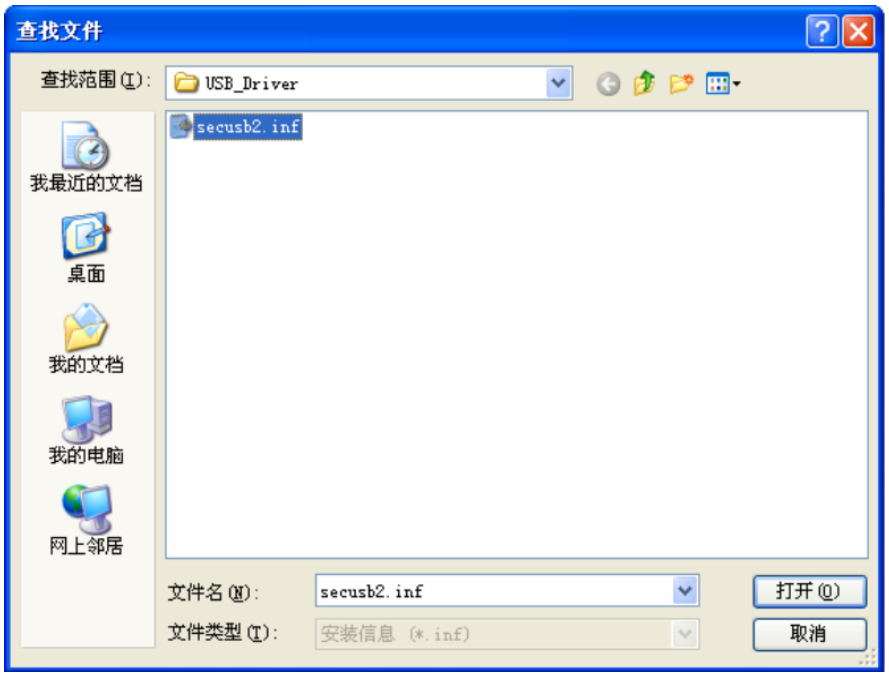

找到新的硬件向导
欢迎使用找到新硬件向导
这个向导帮助您安装软件:
SEC S5PC110 Test B/D
如果您的硬件带有安装 CD 或软盘，请现在将其插入。
您期望向导做什么?
自动安装软件(推荐)(I)
从列表或指定位置安装(高级)(S)
要继续，请单击“下一步”。
< 上一步(B)
下一步(N) >
取消


找到新的硬件向导
请选择您的搜索和安装选项。
在这些位置上搜索最佳驱动程序(S)。
使用下列的复选框限制或扩展默认搜索，包括本机路径和可移动媒体。会安装找到的最佳驱动程序。
搜索可移动媒体(软盘、CD-ROM...)(M)
在搜索中包括这个位置(O):
F:\android_sdk\android-sdk-windows-1.6_r1\
浏览(R)
不要搜索。我要自己选择要安装的驱动程序(D)。
选择这个选项以便从列表中选择设备驱动程序。Windows 不能保证您所选择的驱动程序与您的硬件最匹配。
< 上一步(B)
下一步(N) >
取消



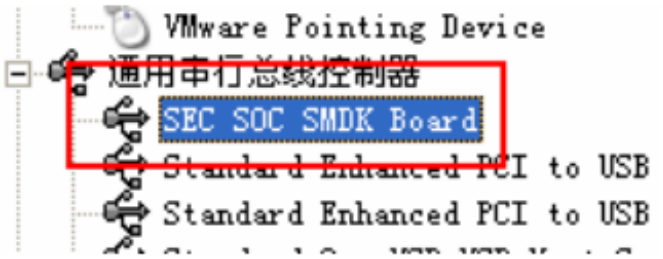


选择光盘中 \USB_Driver\secusb2.inf


查找文件
查找范围(I):
USB_Driver
secusb2.inf
我最近的文档
桌面
我的文档
我的电脑
网上邻居
文件名(N):
secusb2.inf
打开(O)
文件类型(T):
安装信息 (*.inf)
取消


找到新的硬件向导
选择要为此硬件安装的设备驱动程序
请选定硬件的厂商和型号，然后单击“下一步”。如果手头有包含要安装的驱动程序的磁盘，请单击“从磁盘安装”。
显示兼容硬件(C)
型号
SEC SOC SMDK Board
从磁盘安装(H)...
这个驱动程序没有经过数字签署！
告诉我为什么驱动程序签名很重要
< 上一步(B)
下一步(N) >
取消

安装完成后,在 PC 的资源管理器中可以看到:



4.1.4 相关的 IMAGE

1. 烧写涉及到的镜像文件

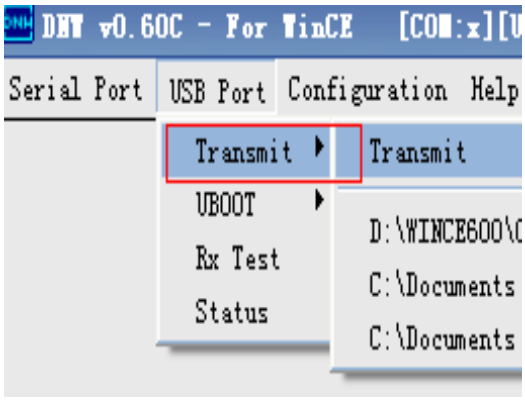
x210_usb.bin(或 V210_USB.BL2.bin)

eboot.nb0

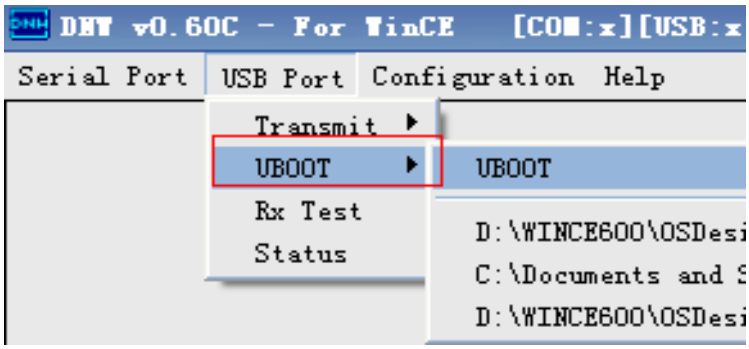
9tripod_boot.nb0（或 bootimage.nb0）

nk.bin

2. x210_usb.bin 和 eboot.nb0 只需要在 RAM 中运行,不需要写入 NAND FLASH 中,烧写时,通过 DNW 的 Transmit 来将 x210_usb.bin 和 eboot.nb0 装载到 RAM 中运行.



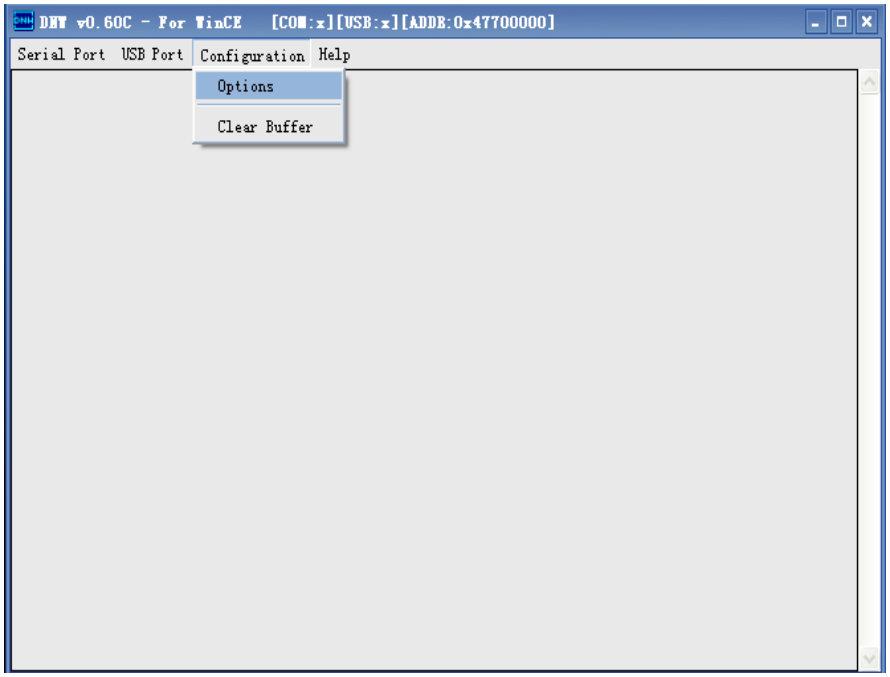
3. bootimage.nb0 和 nk.bin 是固化到存储媒体 ROM 中的,烧写的时候,是通过 DNW 的 UBOOT 来将 9tripod_boot.nb0 和 nk.bin 写入到存储媒体 ROM 中.



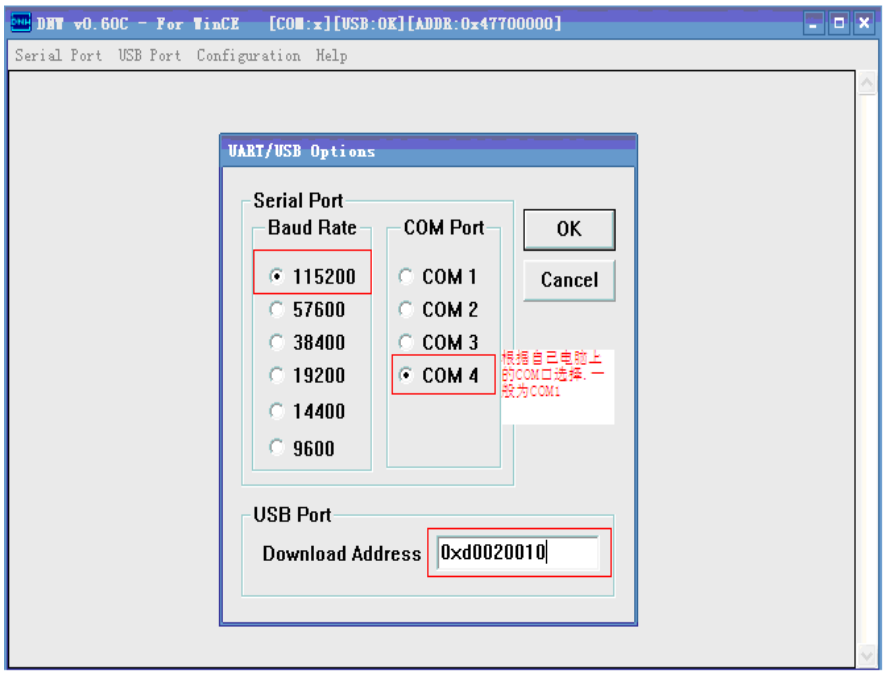
4.1.5 DNW 的设置

在 Windows XP 下打开 DNW.设置 DNW.步骤如下:

1. Configuration->Options



2. Download Address 设为 0xd0020010.



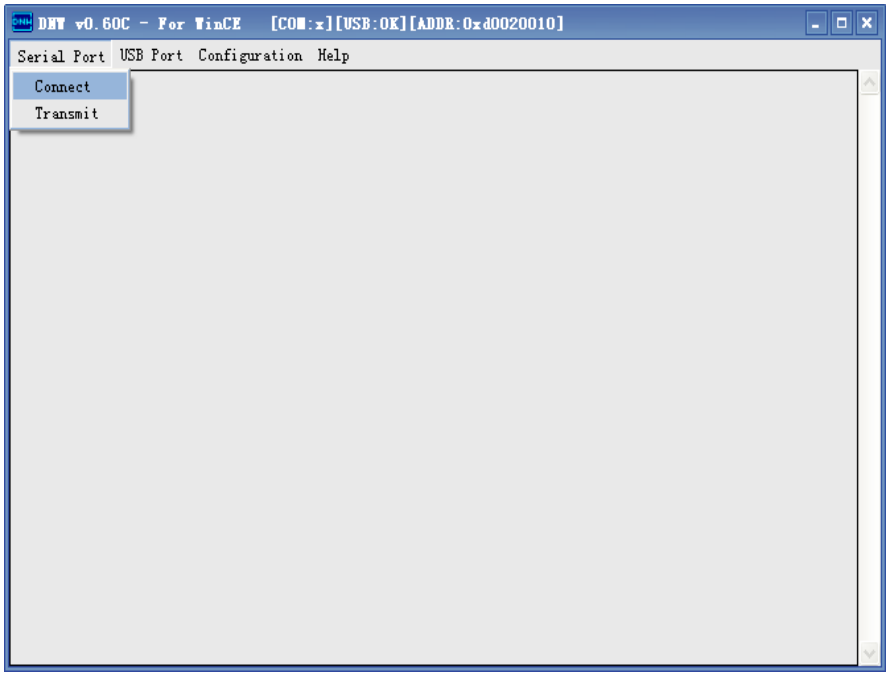
注意：

COM Port 调置成 PC 机上的 COM 端口号,一般为 COM1.这里选的是 COM4,是因为笔者用的是 USB->COM 转换器.所以选 COM4

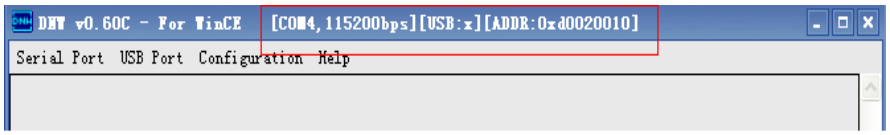
下载 V210_USB.BL2.bin 时, Download Address 设为 0xd0020010

下载 EBOOT.nb0 .bootimage.nb0 NK.bin 时, Download Address 设为 0x47700000

3. Serial Port->Connet



调置完成后，可以看到页眉的[COM:x]变成了[COM4,115200bps],如下图显示



4.2 第一次烧写 9tripod_boot.nb0(bootimage.nb0)到 ROM 中

上电时,开发板会自动运行一个引导程序(CPU 出厂时就存在)，通过该引导程序加载并运行 x210_usb.bin(V210_USB.BL2.bin)，x210_usb.bin 运行起来后,通过 USB OTG 接口将 EBOOT.nb0 加载到 RAM 中,并运行，EBOOT.nb0 运行起来后,可以通过 USB 口将 bootimage.nb0 写入 NAND FLASH 中,还可以将 NK.bin 写入 NAND FLASH 中。

此烧写方式适用于第一次烧写和当 NAND FLASH 中的 bootimage.nb0 被损坏时。

4.2.1 引导模式说明

1. S5PV210 CPU 支持 6 种引导模式。由 OM0-OM5 共计 6 个引脚的电平来决定。引导模式如下图所示。

OM5]	OM[4:1]	OM[0]	Storage
Boot Sep. 0: Storage 1: USB-> UART-> Storage	0 0 0 0	0: X-TAL	NAND 512B-4cycle
	0 0 0 1		NAND 2KB-5cycle
	0 0 1 0		NAND 4KB-5cycle 8-ECC
	0 0 1 1		NAND 4KB-5cycle 16-ECC
	0 1 0 0		OneNAND Mux(Audi)
	0 1 0 1		OneNAND DeMux(Audi)
	0 1 1 0	1: X-TAL (USB)	SD/MMC
	0 1 1 1		eMMC(4bit)
0: Storage	1 0 0 0		Reserved
	1 0 0 1		NAND 2KByte Page, 4cycle
	1 0 1 0		iROM NOR boot
	1 0 1 1		eMMC(8bit)

上图中红色线框内的拔码设计是我们将接下来说明讲解。

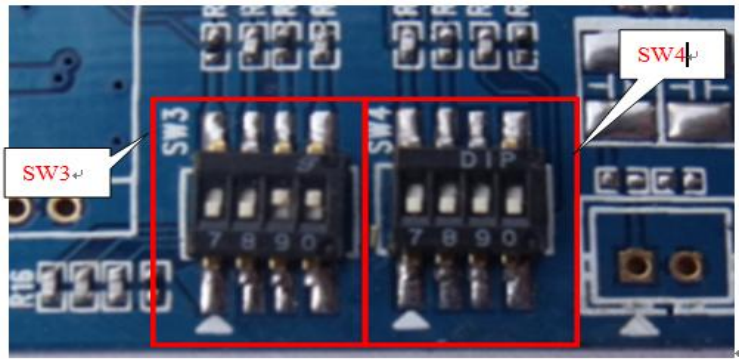
一种是 NAND FLASH 引导。

另一种是 SD/MMC/iNAND 引导。

第三种是调试模式。将按 USB→UART→Storage 顺序引导。

2. X210-II 开发板上拔码开关

X210-II 开发板拔码开关 SW3 和 SW4，左至右对应 OM0 OM1 … OM5 .最右边两个空，没有定义。**特别注意：OM0 在后来的核心板中，要求必须拔成 1，即高电平。**



3. 拔码实例如下:

1） SD/MMC 引导。对应引脚如下：

SD/MMC/iNAND boot								
SW3					SW4			
OM	0	1	2	3	4	5	N/A	N/A
1(UP)	**1**		**1**	**1**			N/A	N/A
0(DOWN)		0			0	0	N/A	N/A

2） Wince 系统和 android 系统在 nand flash 时，按下表拔码

NAND FLASH boot								
SW3					SW4			
OM	0	1	2	3	4	5	N/A	N/A

1(UP)	1	1					N/A	N/A
0(DOWN)			0	0	0	0	N/A	N/A

4.2.2 设置 USB 启动模式(调试模式)

1. NAND FLASH ROM

当开发板上的 ROM 类型为 NAND FLASH 时，拔码开关 SW3 和 SW4 拔成 USB 引导模式:如下图所示。

USB/UART/STORAGE(NAND FLASH) boot								
	SW3				SW4			
OM	0	1	2	3	4	5	N/A	N/A
1(UP)	1	1				1	N/A	N/A
0(DOWN)			0	0	0		N/A	N/A

注: SW3、SW4 拔码开关向上表示" 1"，向下表示" 0"

2. iNAND ROM

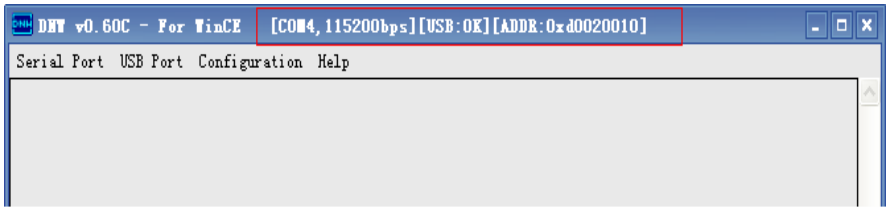
当开发板上的 ROM 类型为 iNAND 时，拔码开关 SW3 和 SW4 拔成 USB 引导模式:如下图所示。

USB/UART/STORAGE(Inand) boot								
	SW3				SW4			
OM	0	1	2	3	4	5	N/A	N/A
1(UP)	1		1	1		1	N/A	N/A
0(DOWN)		0			0		N/A	N/A

注: SW3、SW4 拔码开关向上表示" 1"，向下表示" 0"

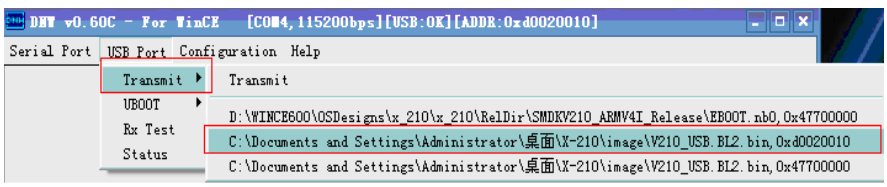
4.2.3 烧写 9tripod_boot.nb0(bootimage.nb0)到 ROM 中

1. 运行 DNW,给开发板上电,一直按住电源键,此时,USB 连接 OK.如下图



注意:现在电源按键要一直按住,因为电源自锁是靠 EBOOT 内的代码控制.而不是硬件电路或按键自锁。所以只有在启动 EBOOT 后,电源键才被锁住,那时才可以松开电源按键。

2. 下载 x210_usb.bin(V210_USB.BL2.bin),注意这里用 Transmit 下载

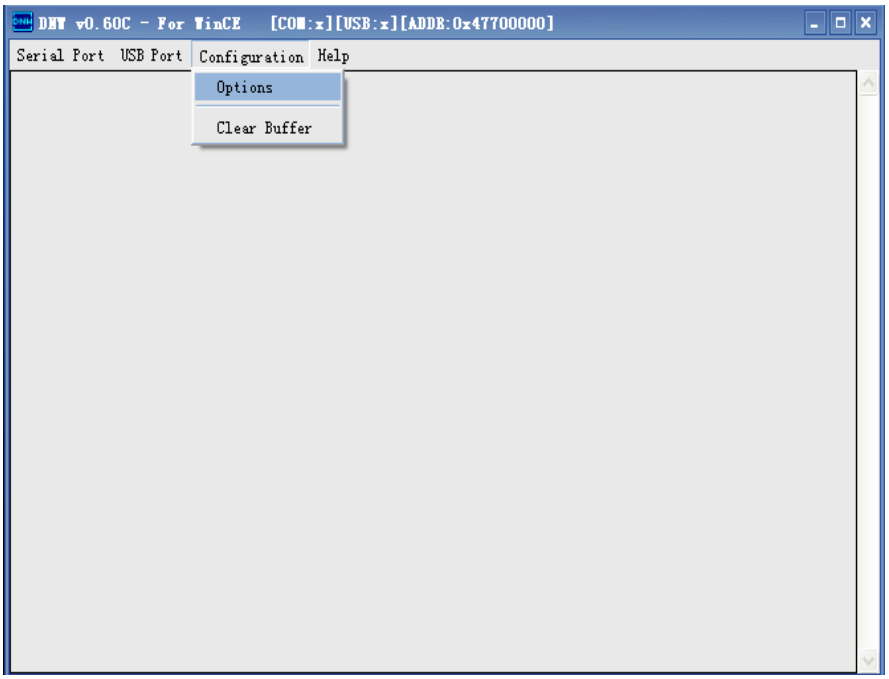


3. 此时,USB:OK 会变成 USB:X 再变到 USB:OK.

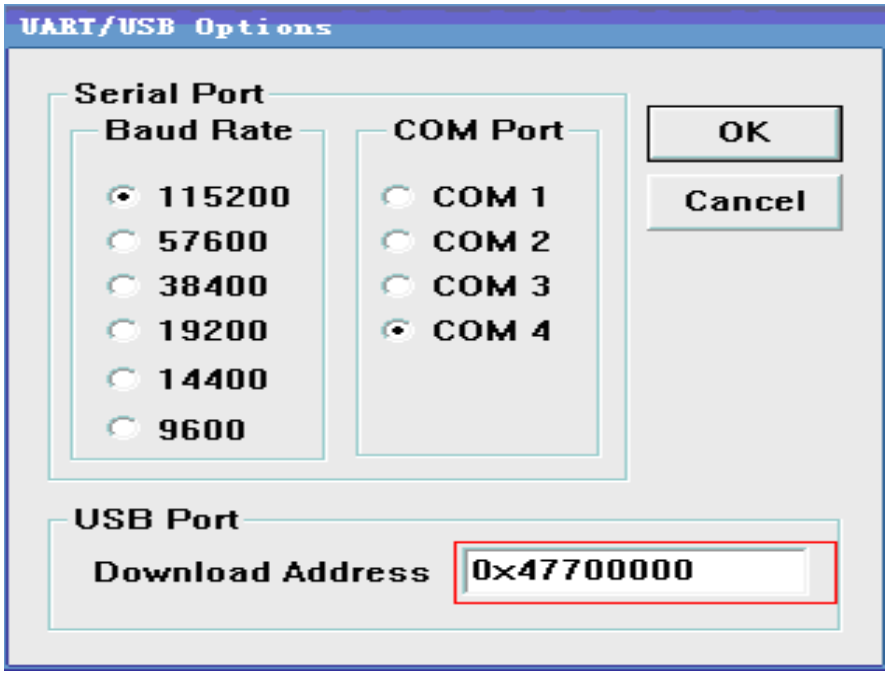
先从 USB:OK→UBS:x

再变成 USB:OK

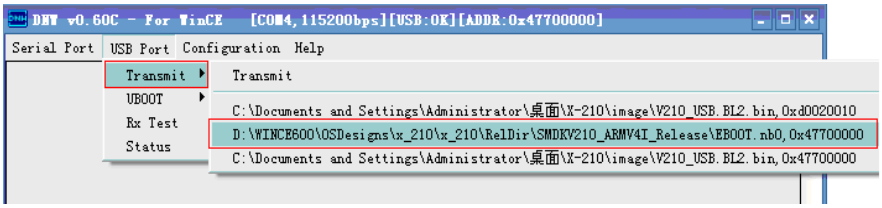
4. 下载运行 eboot.nb0



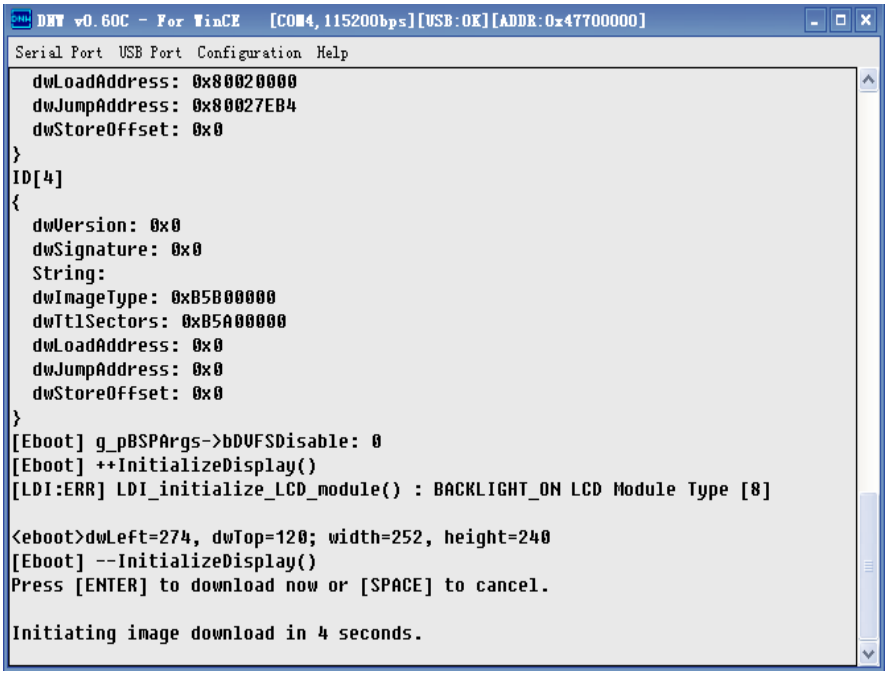
Download Address 设为 0x47700000.



下载运行 eboot.nb0

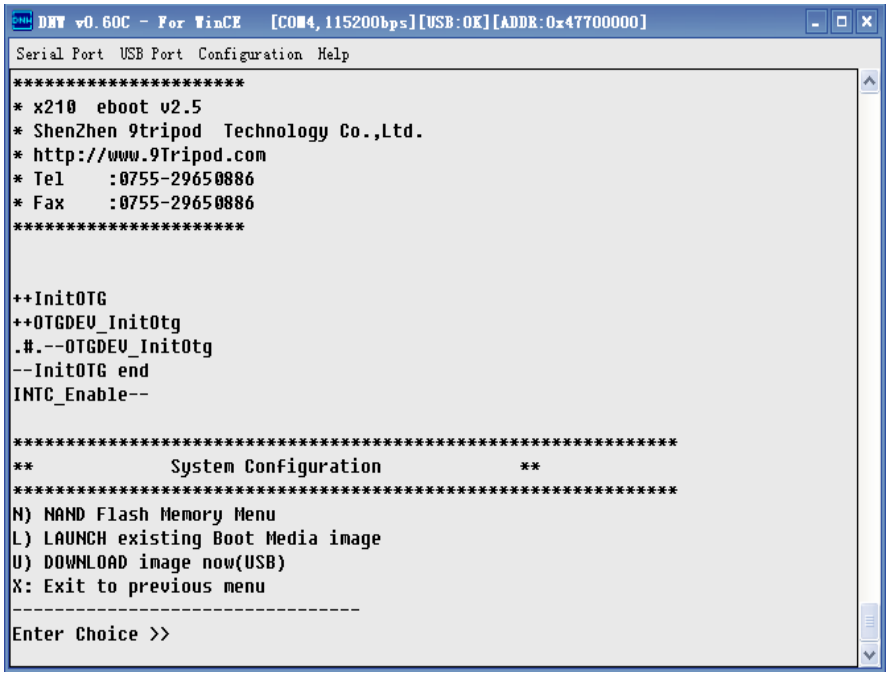


Eboot.nb0 运行起来后,进入以下 5 秒倒计时界面

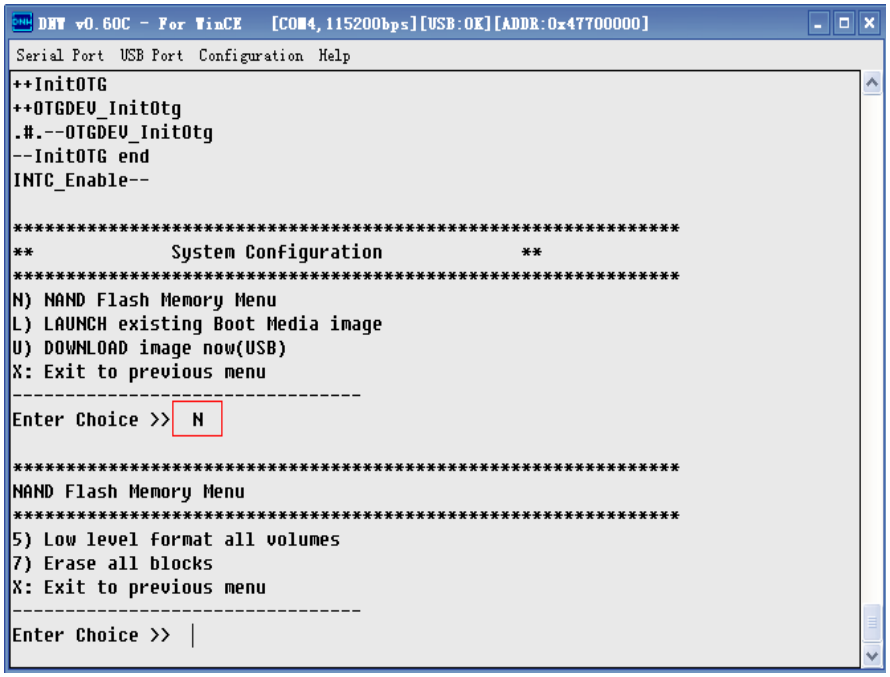


注意:此时按住的电源键此时才可松开.因为 eboot.nb0 内有代码会对电源键进行自锁。

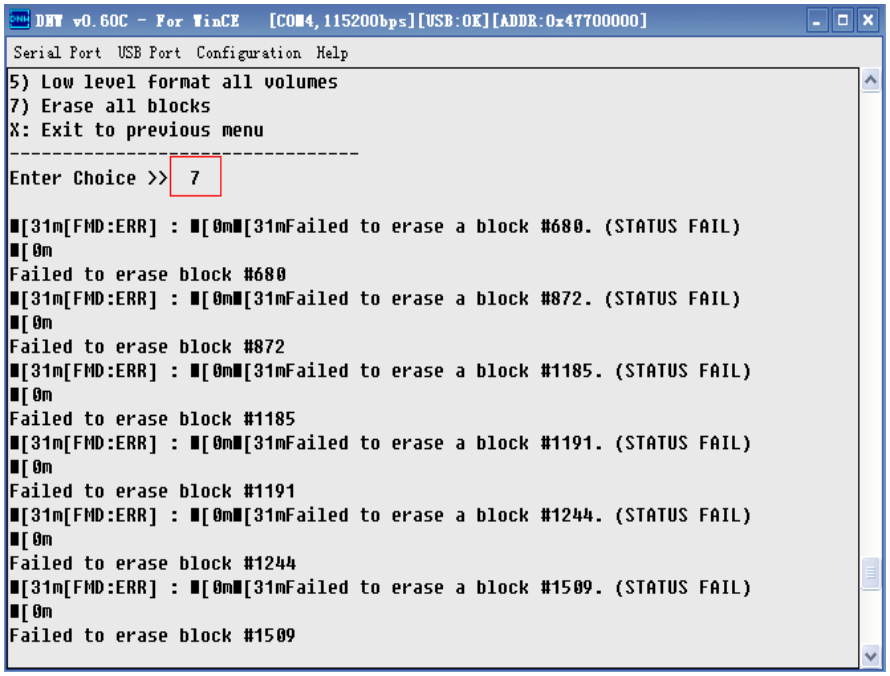
按空格键,进入菜单选择界面

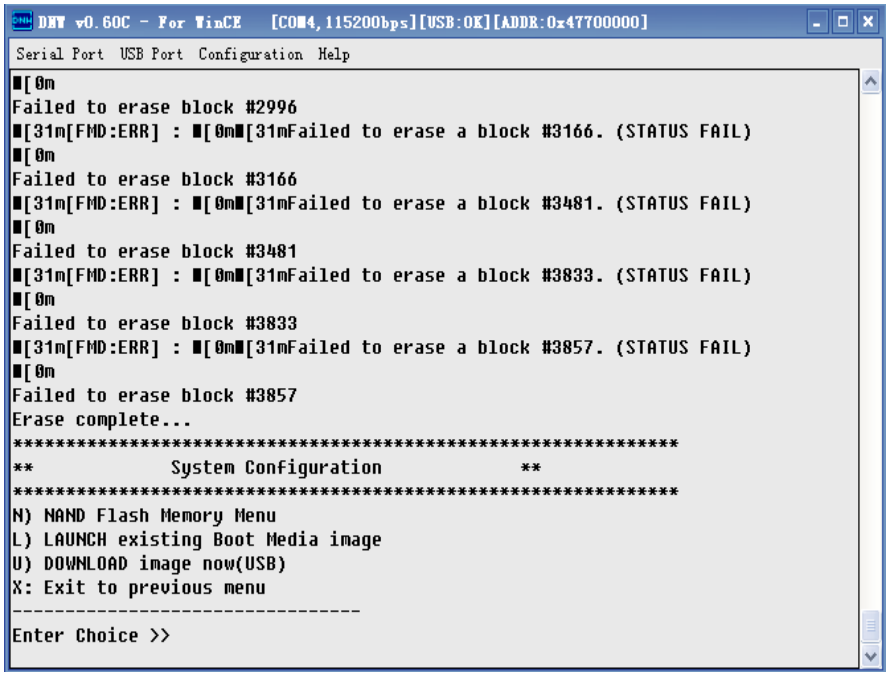


在 Enter Choice >>处输出 N

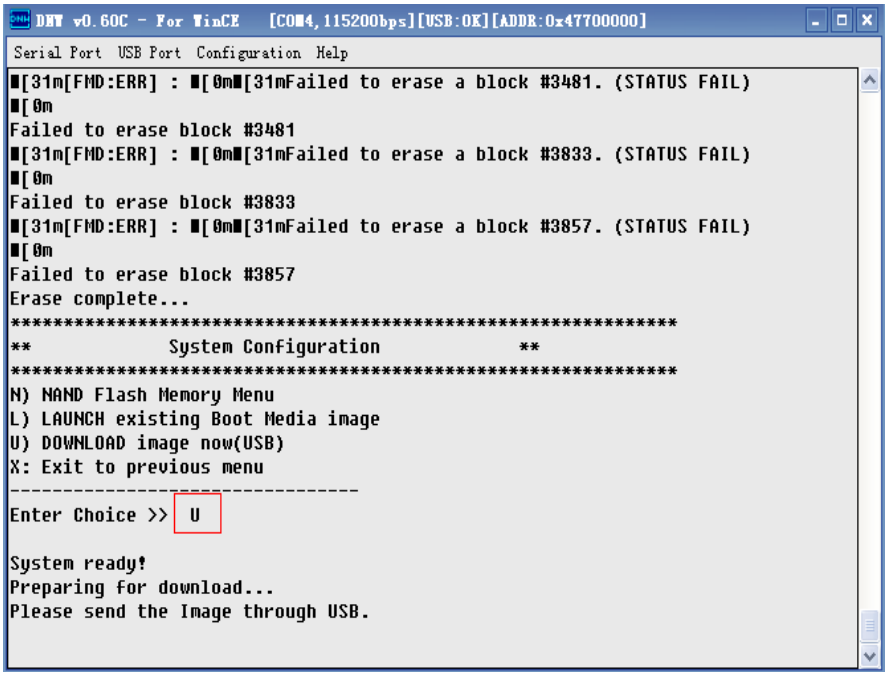


在 Enter Choice >>处输出 7

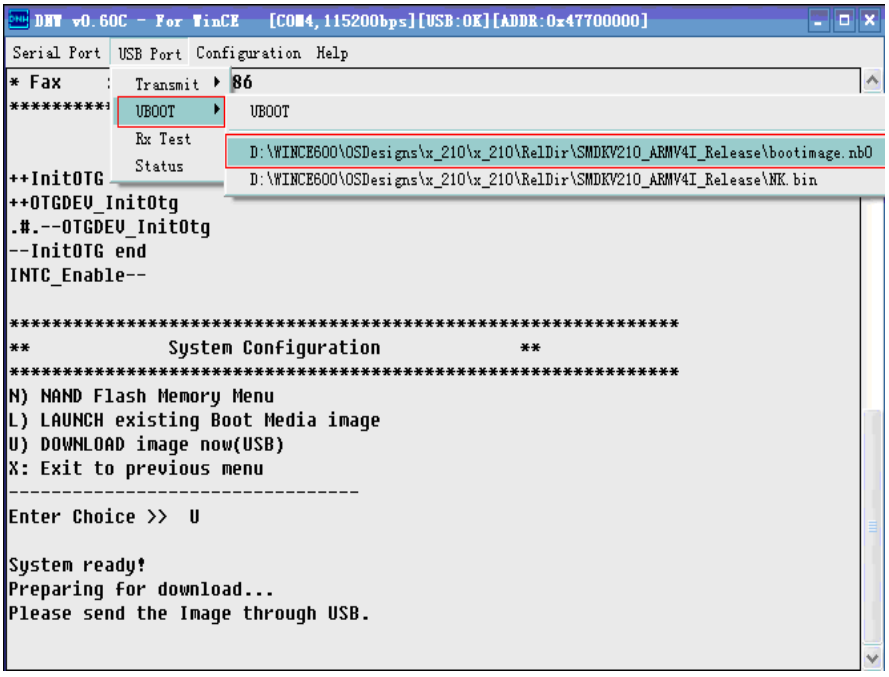




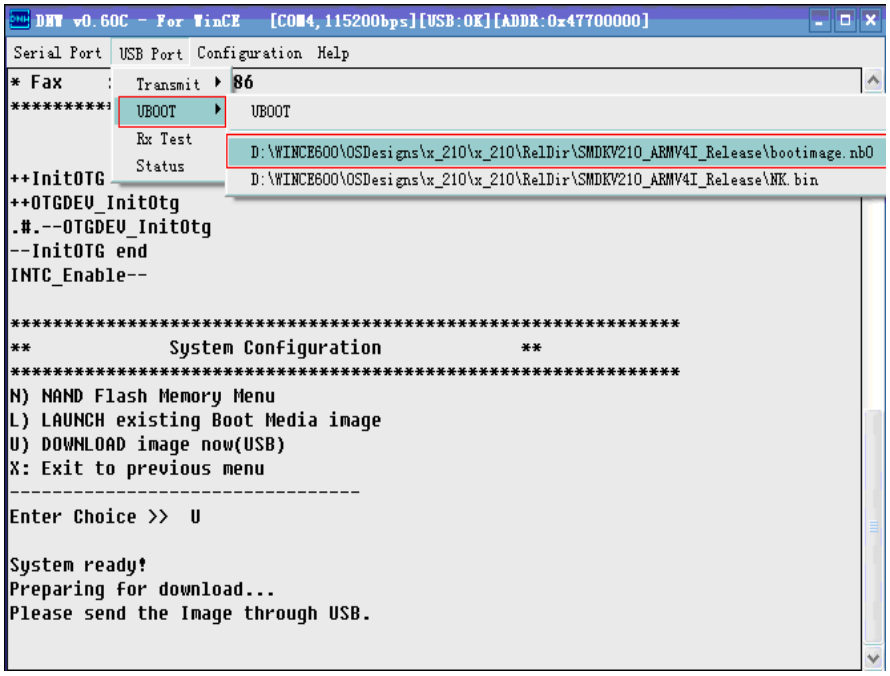
在 Enter Choice >>处输出 U

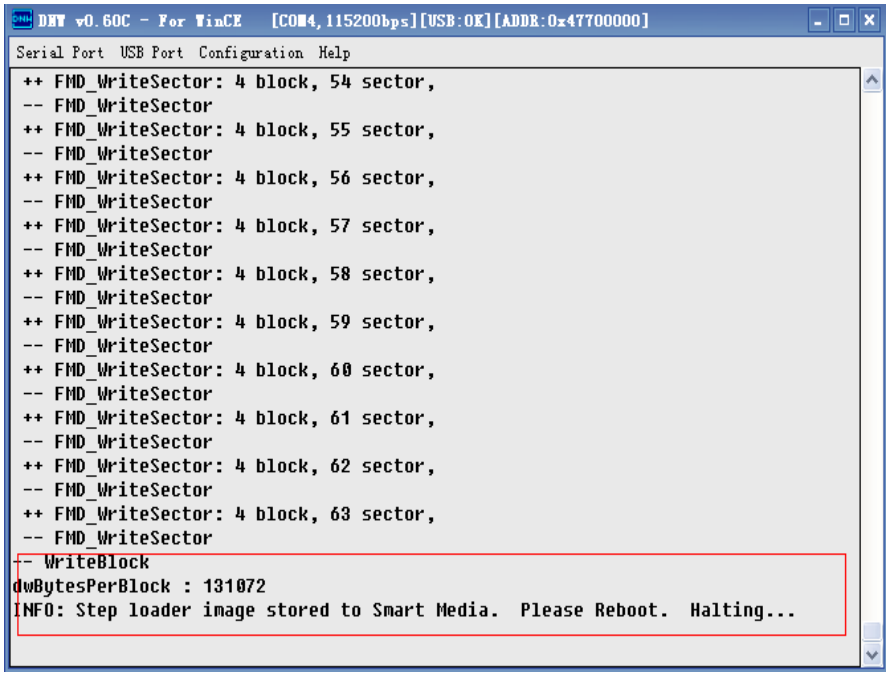


5. 通过USB Port->UBOOT,将9tripod_boot.nb0(bootimage.nb0)下载并烧写到ROM中



通过 USB Port->UBOOT,将 9tripod_boot.nb0（bootimage.nb0）下载并烧写到 ROM 中。





9tripod_boot.nb0（bootimage.nb0）已成功烧写到 ROM 中。

到这一步后，ROM 中就已有了 bootloader，所以以后更新升级就不需要重复上面的步骤。直接根据 bootloader 启来时的菜单选择来更新 9tripod_boot.nb0 或 nk.bin 等镜像文件。

4.3 烧写 NK 到 ROM 中

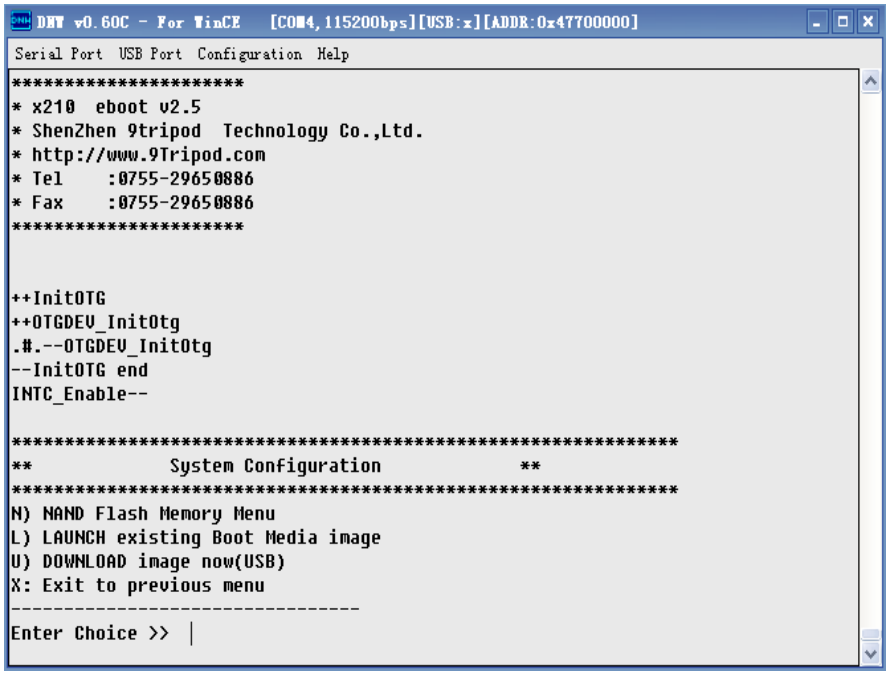
1. 此时可将 x210-II 开发板设置成 NAND FLASH 启动模式或 iNAND 启动模式(只需拔 SW4 的 OM5 到 0 即可)如下表所示：

NAND FLASH boot

SW3					SW4			
OM	0	1	2	3	4	5	N/A	N/A
1(UP)		**1**					N/A	N/A
0(DOWN)	0		0	0	0	0	N/A	N/A

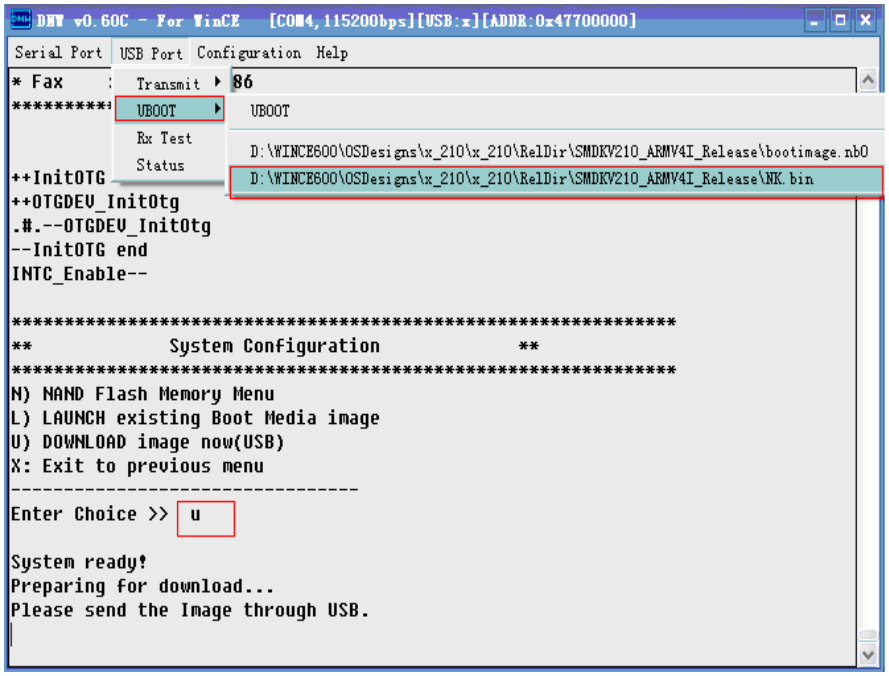
Inand boot								
SW3					SW4			
OM	0	1	2	3	4	5	N/A	N/A
1(UP)	**1**		**1**	**1**			N/A	N/A
0(DOWN)		0			0	0	N/A	N/A

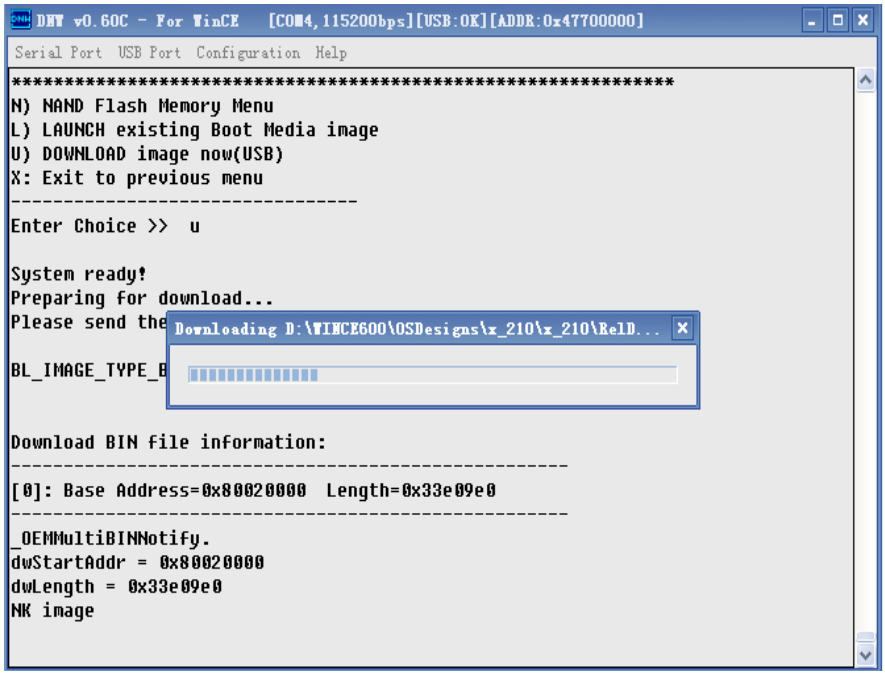
2. 重启开发板,这时开发板将从 ROM 中引导 EBOOT.在 eboot 启动 5 秒内,按 PC 机的空格键(要求口串口线连接正常),这时进入功能界面中：



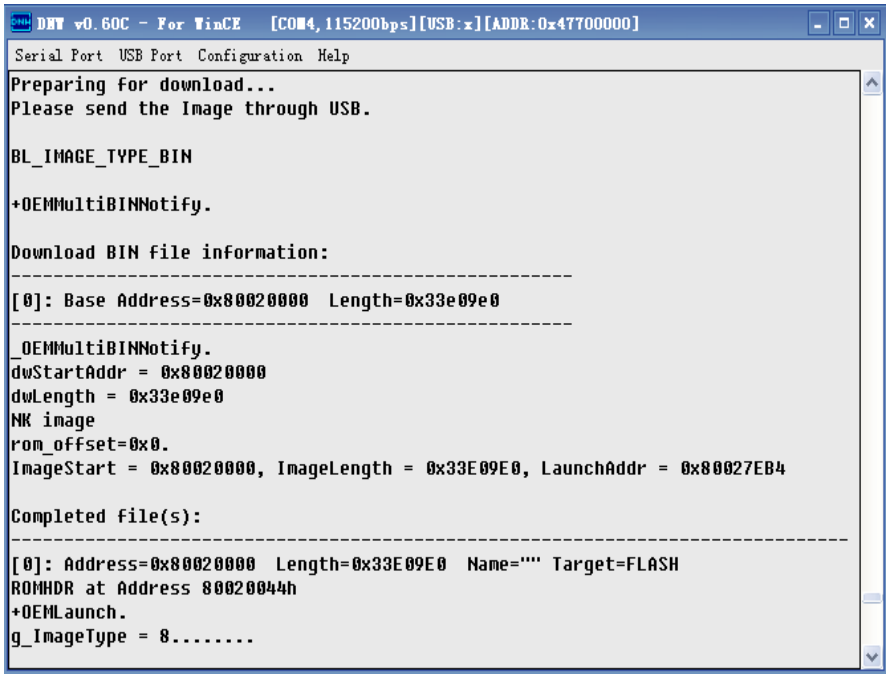
3. 烧写 NK 到存储媒体 ROM 中

在 Enter Choice >>处输出 U。进入 USB Port→UBOOT 下载并烧写 NK.bin.





下载过程如下,没有信息提示.下载完成后,系统会自行启动


DNW v0.60C - For WinCE [COM4,115200bps][USB:x][ADDR:0x47700000]
Serial Port USB Port Configuration Help
Preparing for download...
Please send the Image through USB.
BL_IMAGE_TYPE_BIN
+OEMMultiBINNotify.
Download BIN file information:
[0]: Base Address=0x80020000 Length=0x33e09e0
_OEMMultiBINNotify.
dwStartAddr = 0x80020000
dwLength = 0x33e09e0
NK image
rom_offset=0x0.
ImageStart = 0x80020000, ImageLength = 0x33E09E0, LaunchAddr = 0x80027EB4
Completed file(s):
[0]: Address=0x80020000 Length=0x33E09E0 Name="" Target=FLASH
ROMHDR at Address 80020044h
+OEMLaunch.
g_ImageType = 8.....

第5章 WinCE 使用及模块测试

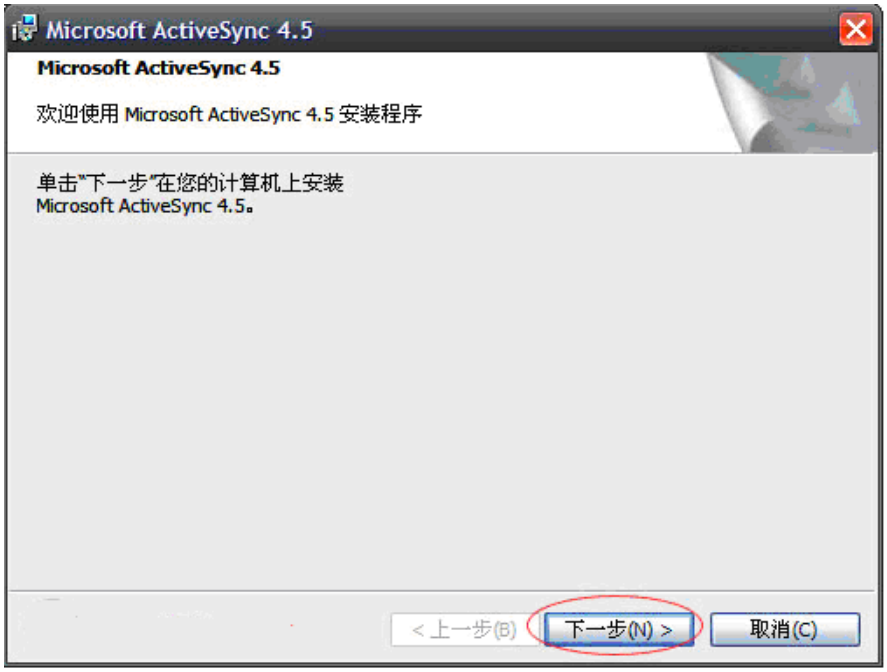
5.1 安装 Microsoft ActiveSync 4.5 同步软件

在 WINCE 嵌入式系统与 PC 同步，要使用 ActiveSync 这个软件。Microsoft ActiveSync 是基于 Windows CE 设备的最新同步软件版本。它通过 USB 从设备端口将开发板与 PC 机相连，方便文件传输和应用程序开发/调试。这是微软开发的运行于 PC 上的软件，支持 PC 机与其他移动设备的通信。

注意：在安装时 ActiveSync 时， USB DEVICE 不要连接到开发板上。

安装同步软件 Microsoft ActiveSync 4.5

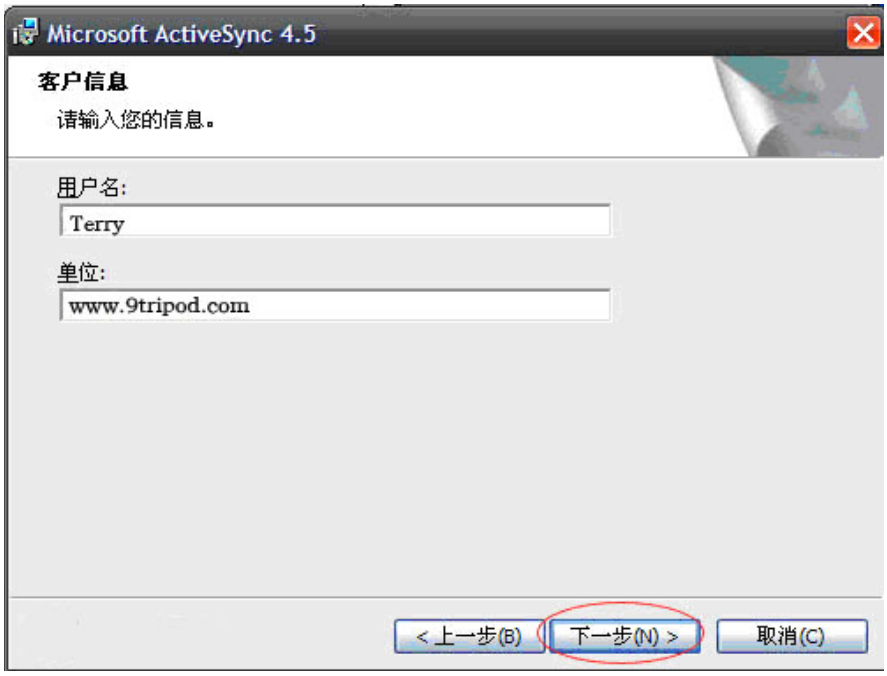
1. 双击：Microsoft ActiveSync 4.5.msi



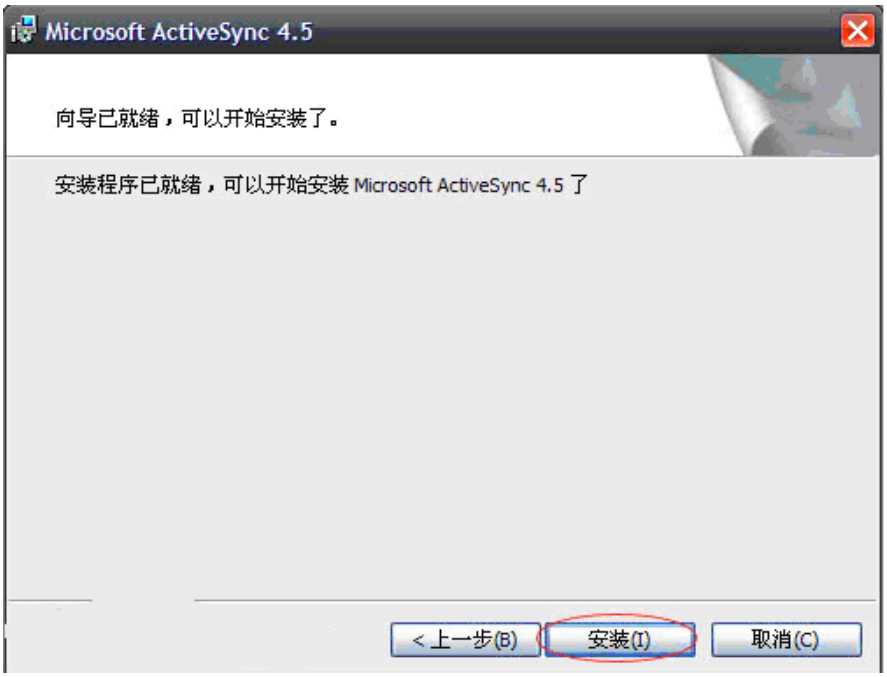
2. 同意条款，下一步

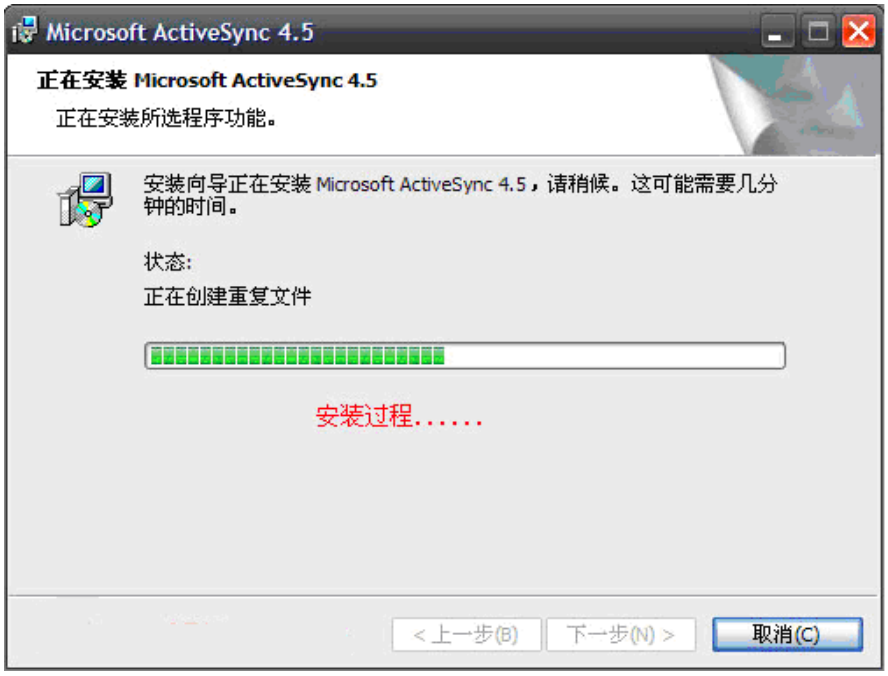
3. 选择安装路径

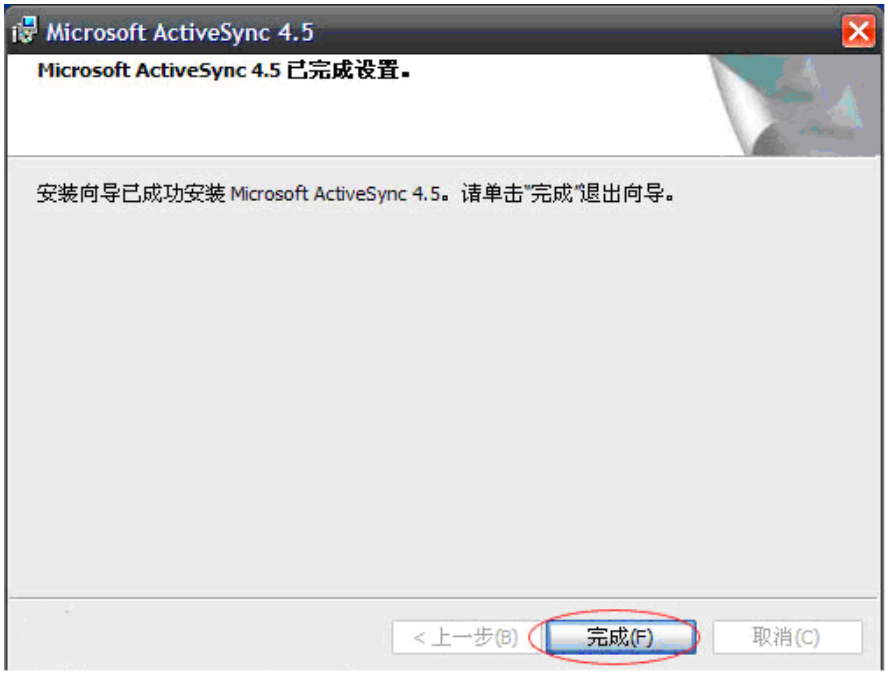
4. 填入用户信息


Microsoft ActiveSync 4.5
客户信息
请输入您的信息。
用户名:
Terry
单位:
www.9tripod.com
< 上一步(B)
下一步(N) >
取消(C)

5. 安装


Microsoft ActiveSync 4.5
向导已就绪，可以开始安装了。
安装程序已就绪，可以开始安装 Microsoft ActiveSync 4.5 了
< 上一步(B)
安装(I)
取消(C)



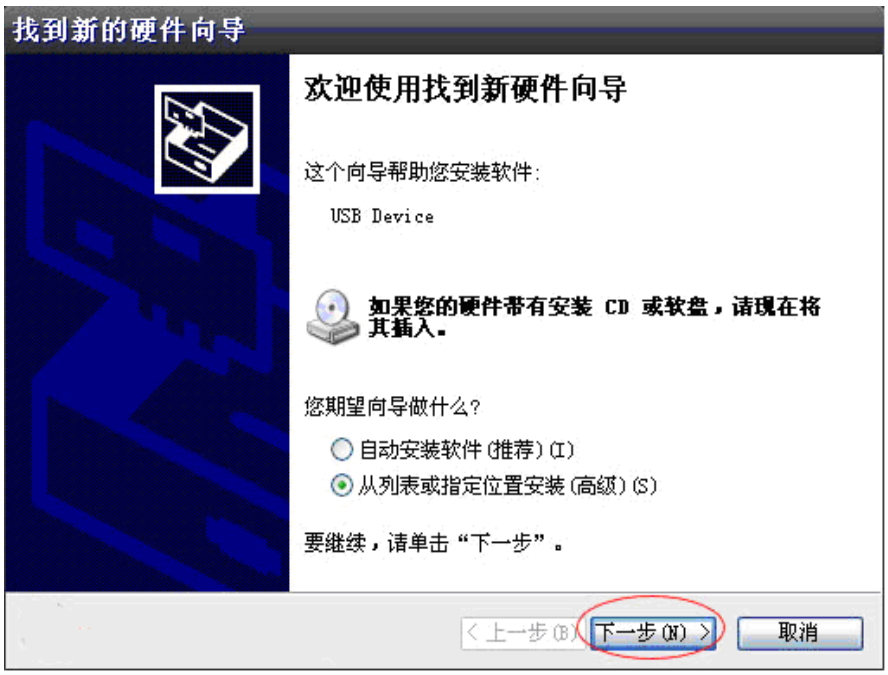


6. 完成安装

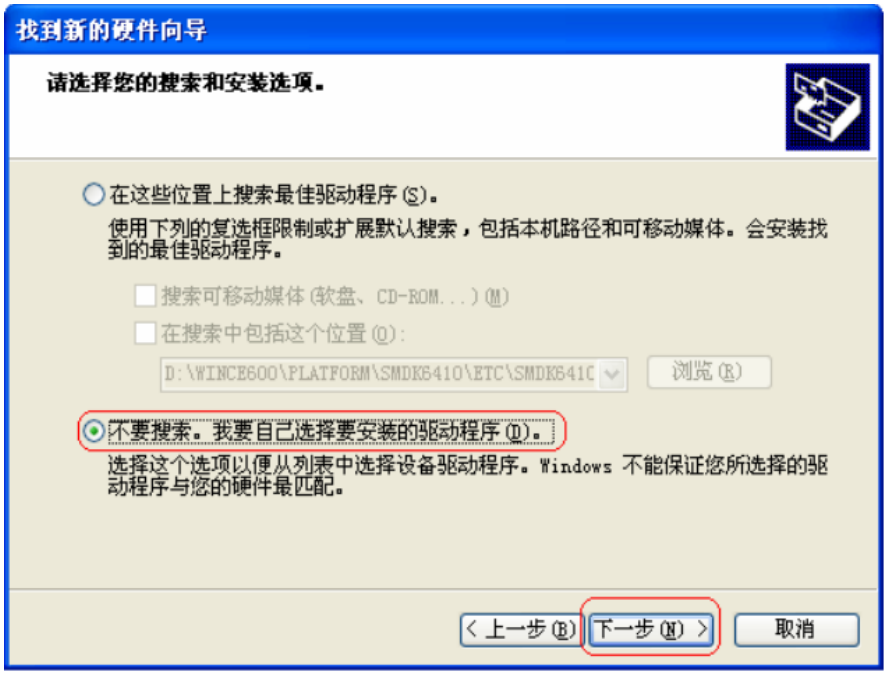
安装之后，桌面右下角应该出现了这个软件的图标，两个旋转的箭头  ，这时这个图标应该是灰色的，因为 WINCE 与 PC 机还未连接。

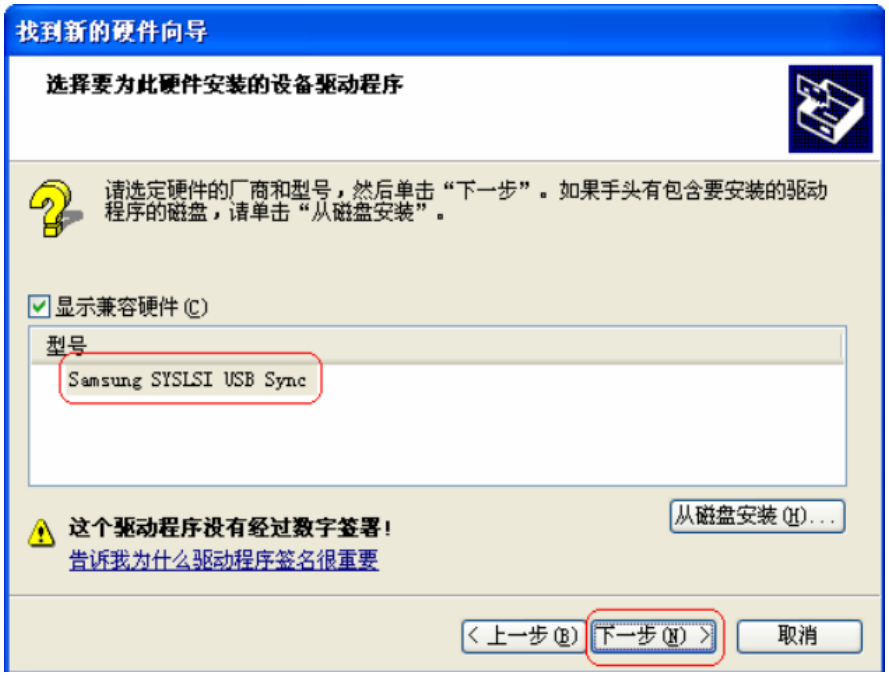
5.2 安装 USB 同步驱动

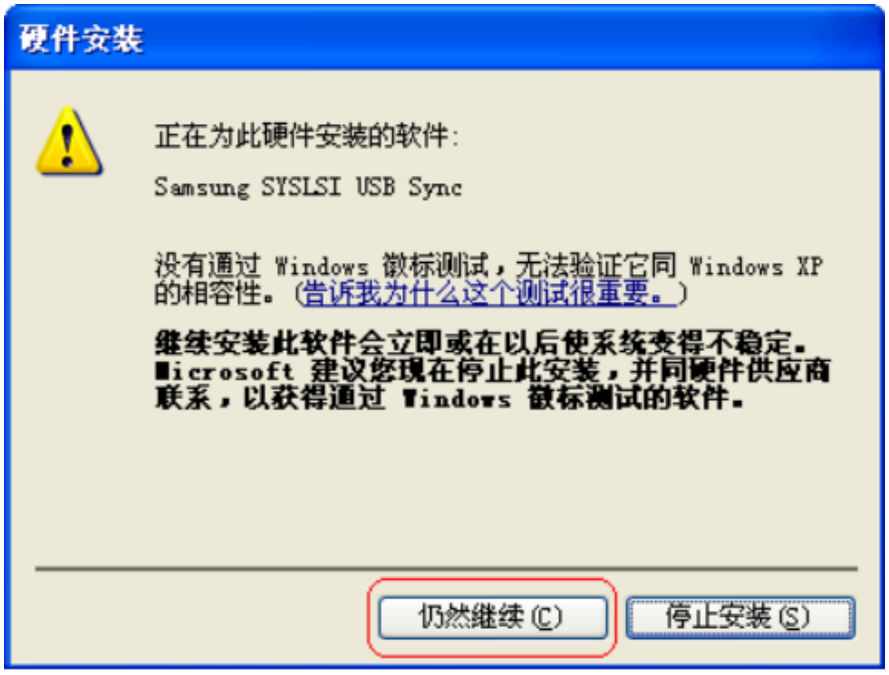
1. USB 线与开发板连接，启动开发板，进入 WINCE 系统。系统会提示找到新硬件。



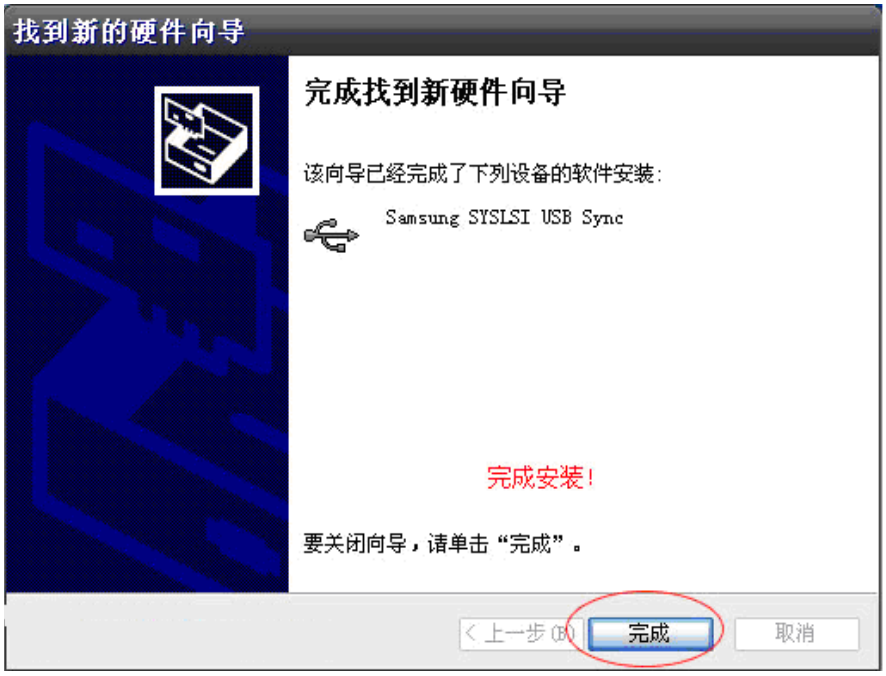
2. 指定驱动所在路径，开发板的 BSP 路径下面的 USB Driver 文件夹




找到新的硬件向导
选择要为此硬件安装的设备驱动程序
请选定硬件的厂商和型号，然后单击“下一步”。如果手头有包含要安装的驱动程序的磁盘，请单击“从磁盘安装”。
显示兼容硬件(C)
型号
Samsung SYSLSI USB Sync
从磁盘安装(H)...
这个驱动程序没有经过数字签署！
告诉我为什么驱动程序签名很重要
< 上一步(B)
下一步(N) >
取消

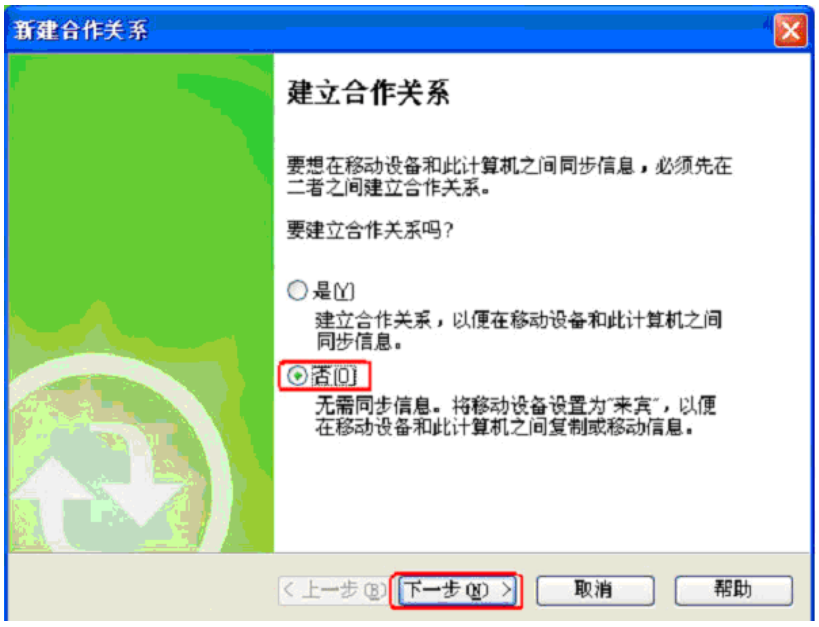

硬件安装
正在为此硬件安装的软件:
Samsung SYSLSI USB Sync
没有通过 Windows 徽标测试，无法验证它同 Windows XP 的相容性。(告诉我为什么这个测试很重要。)
继续安装此软件会立即或在以后使系统变得不稳定。Microsoft 建议您现在停止此安装，并同硬件供应商联系，以获得通过 Windows 徽标测试的软件。
仍然继续(C)
停止安装(S)

3. 完成安装

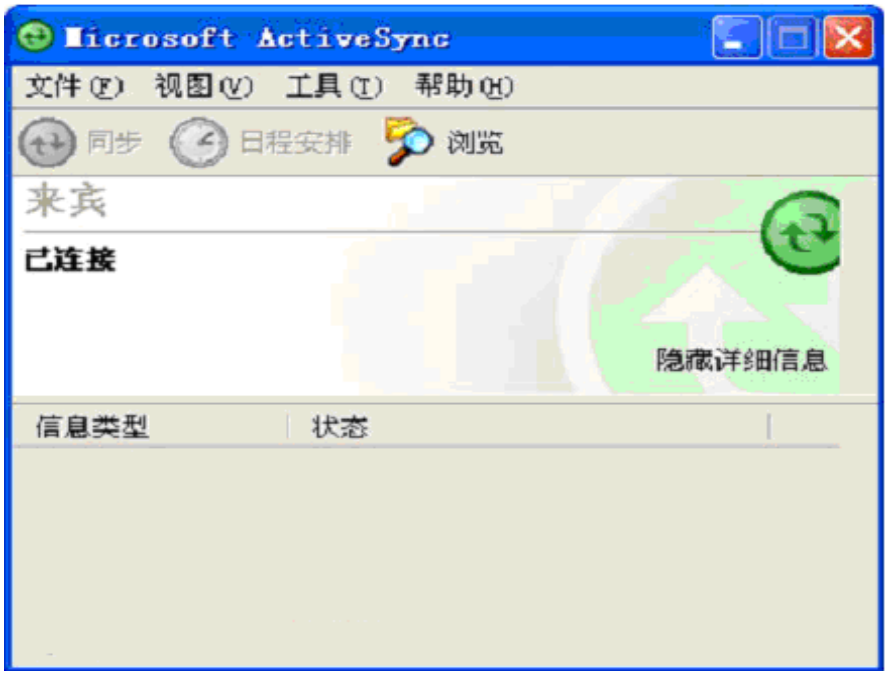


5.3 远程连接到开发板

安装完成驱动，就能看到 PC 桌面右下角任务栏一个双箭头图标  由灰色，变为彩色，在旋转，表示连接正在进行。如果这个灰色的图标没有动静，请保持 USB 线与开发板连接并重启开发板，如果连接成功，会弹出一个同步询问



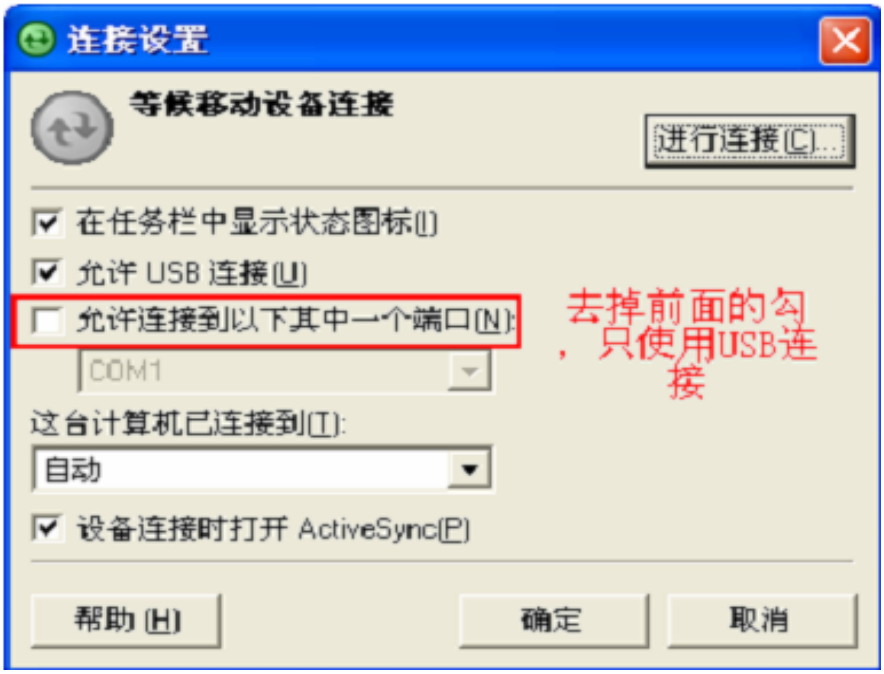
选择 否 开始同步，



到这一步，就可以之前现 VS2005 的在线调试等功能了，还可以在 PC 上直接访问 WINCE 设备上的存储空间，任务栏上的图标也变为了彩色  ，并停止旋转。

如果连接不成功，请设置一下

确保如下设置



5.3.1 访问 WINCE 设备

双击 “我的电脑”中的“移动设备

就能进入到开发板上的目录了。

5.4 远程工具的使用

安装 VS2005 之后，会有一些远程工具，例如最常用的“远程注册表编辑器”，“远程放大”。

在产品开发过程中，这都非常有用的工具，例如，远程注册表编辑器，它能在 PC 端打

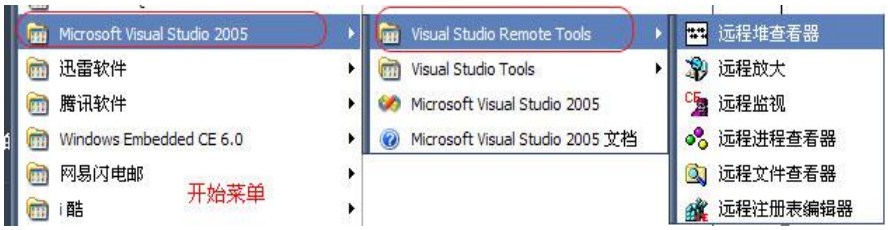
开 WINCE 设备上的注册表并修改，在这系统开发时非常实用。“远程放大”主要是在写产品说明书时，用于抓取 WINCE 设备上的界面。

5.4.1 独立使用远程工具

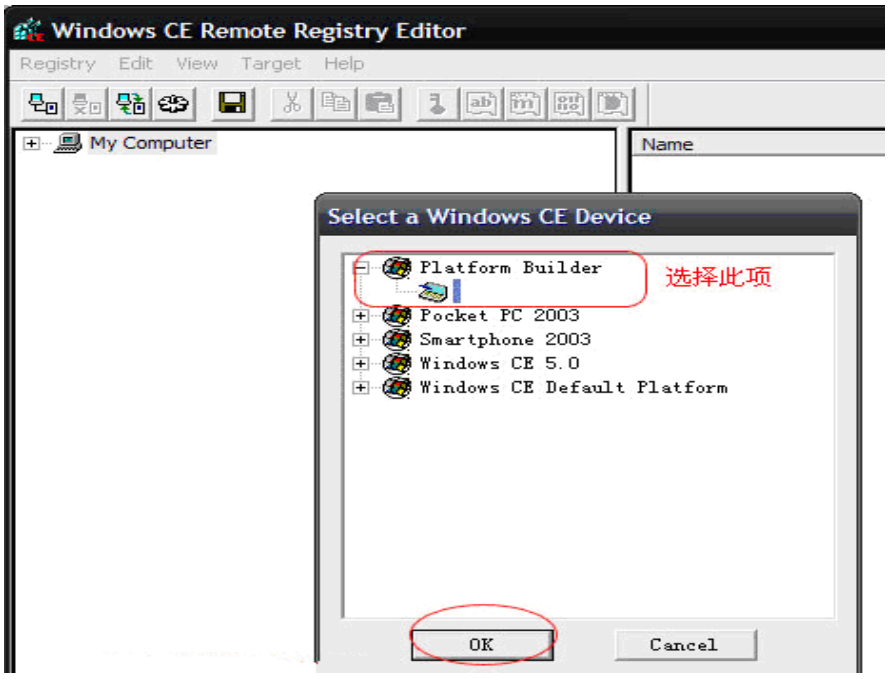
使用方法（以“远程注册表编辑器”为例子说明）

1. 启动开发板，进入 WINCE 系统，ActiveSync 同步开发板。（如 5.2 章）

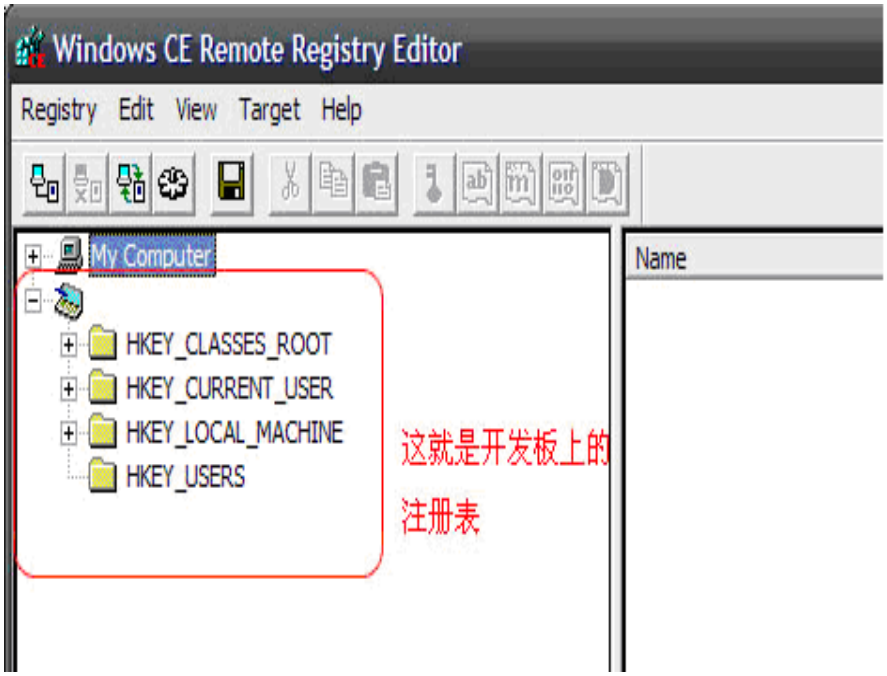
2. 打开工具 “开始” → “所有程序” → “Microsoft Visual Studio 2005” → “Visual Studio Remote Tools” → “远程注册表编辑器”



进入“远程注册表编辑器”界面，自动弹出设备选择窗口，请选择“Platform Builder”项下面的默认子项

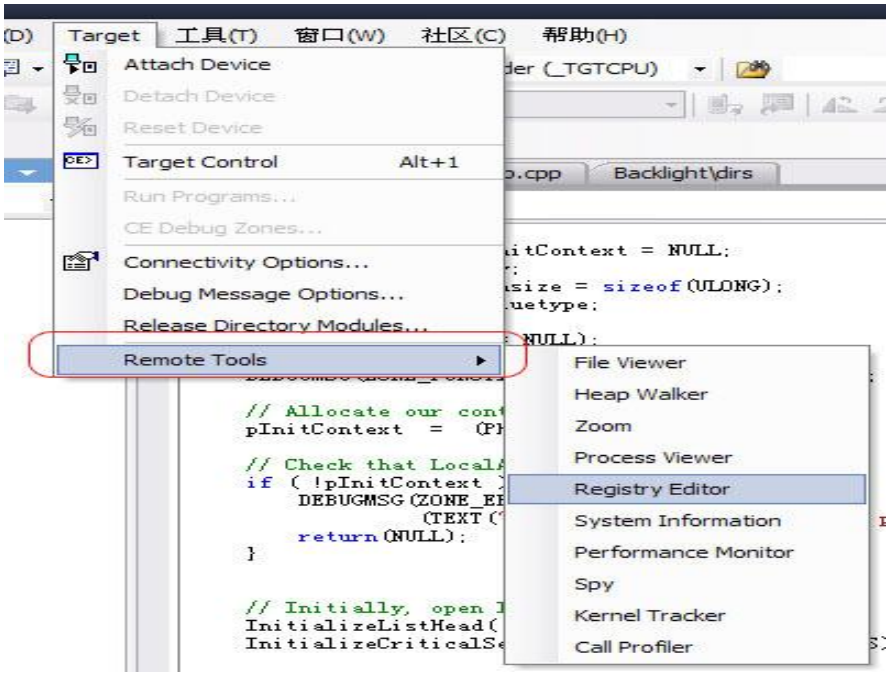


点击 OK，就开始连接过程了，同步注册表成功会如下图



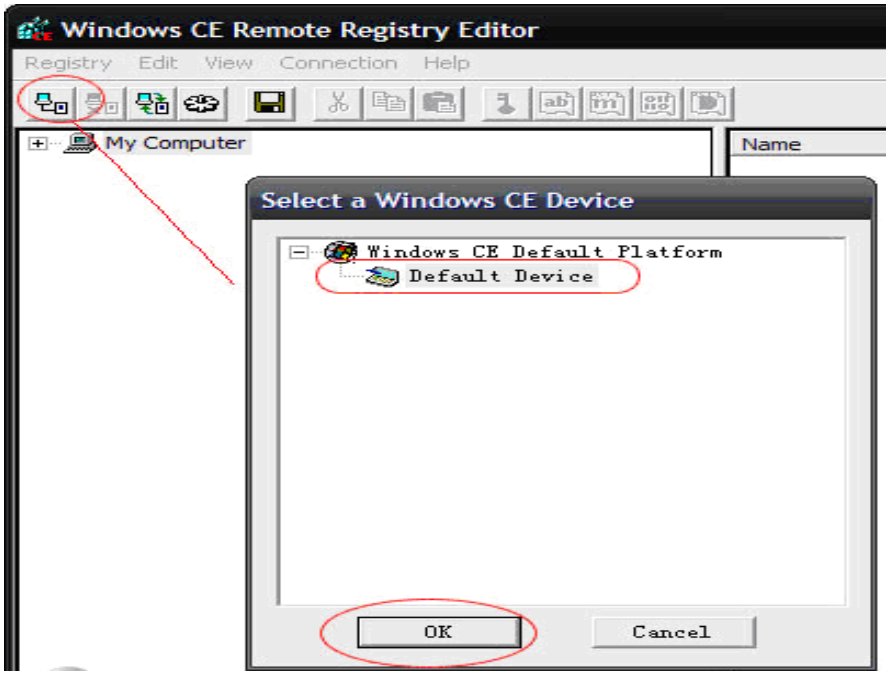
5.4.2 在 VS2005 开发环境里使用远程工具

打开工程后，在页眉菜单，“Target”> “Remote Tools” 就看到了所有远程工具，



还是以注册表工具为例

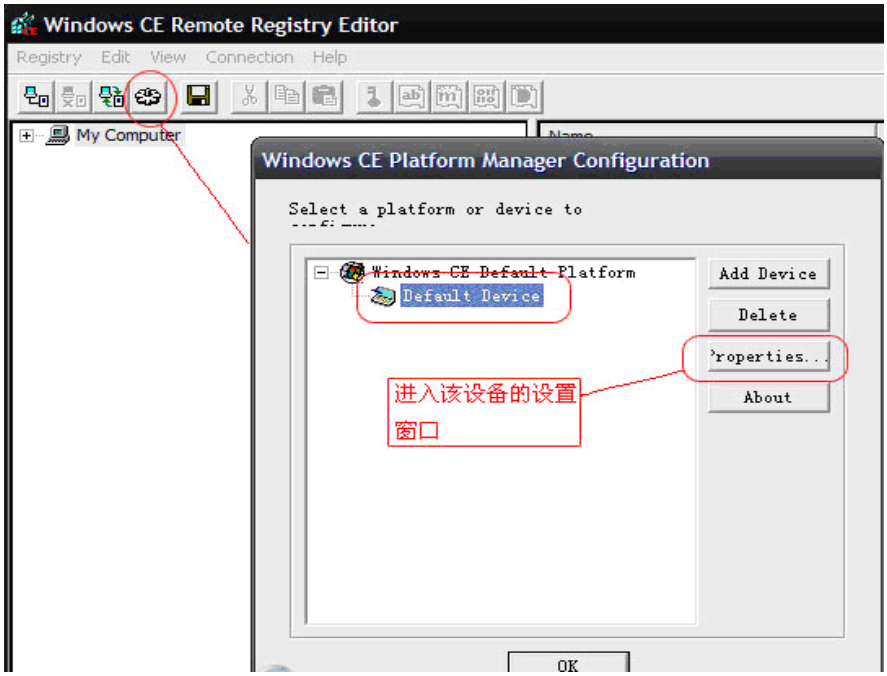
1. 先同步
2. 选择远程工具的 Registry Editor，如上图
3. 在弹出的设备选择窗，选择默认设备“Default Device”



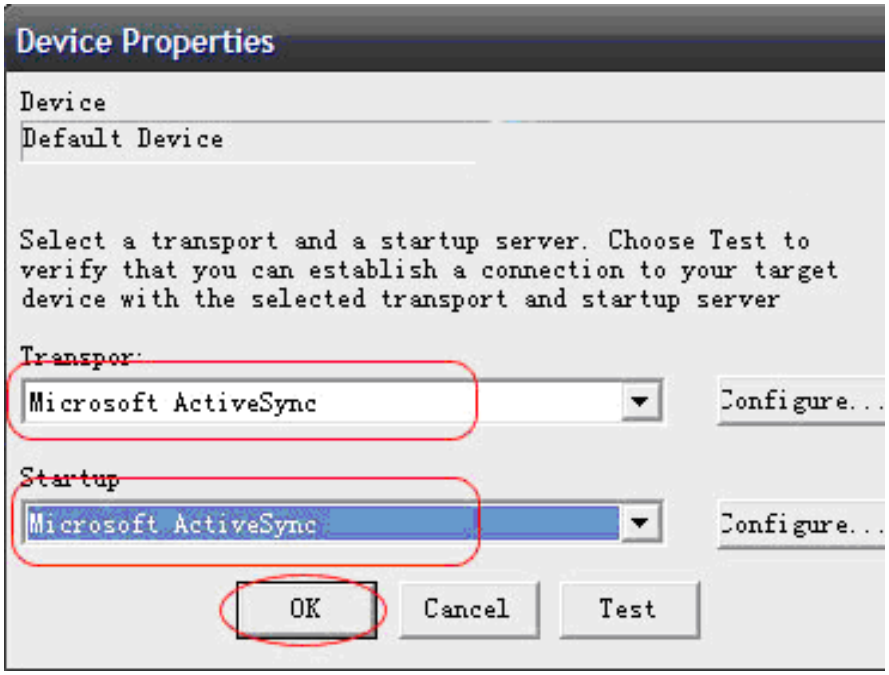
如果是第一次使用，会提示如下错误

这是因为，使用了 KITL 连接方式，我们应该选择 ActiveSync 的，

4. 解决办法，点上图确定，回到远程工具 Registry Editor 的主窗口，选择设置，如下图



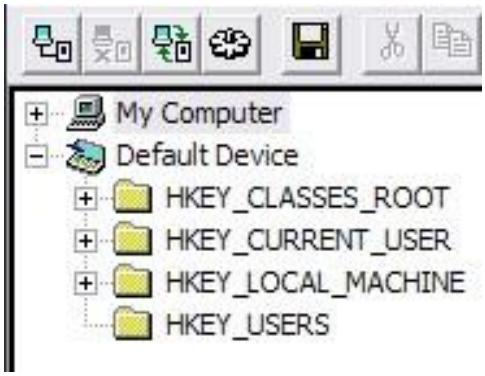
Properties 按钮进入“Default Device”的设置界面



都选择为 ActiveSync 连接方式。

5. 点 OK，回到 Registry Editor 的主窗口

点连接按钮  ,重新连接，如果有防火墙，请关闭，连接成功如下图



ERROR: 如果是第一次使用该功能，会弹出错误信息，

“The Microsoft ActiveSync reported the following error: Unable to load device side components”

这是 PB6 的一个 BUG，因为找不到目录

" C:\Program Files\Common Files\Microsoft Shared\Windows CE Tools\Platman\target\wce600\armV4"

下的有关库和工具，而系统在安装 CE6.0 时，是不会生成该目录的。

解决方法：创建目录“armV4”路径如下

" C:\Program Files\Common Files\Microsoft Shared\Windows CE Tools\Platman\target\wce600\armV4"

然后把 "C:\Program Files\Common Files\Microsoft Shared\Windows CE Tools\Platman\target\wce600\armV4i".下的东西全部拷过来，

注意，如果你的 VS2005 是装在 D 盘，那么，也要确保 C 盘 C:\Program Files\Common Files\Microsoft Shared\Windows CE Tools\Platman\target\wce600\

下面有 armV4 文件夹，没有就复制 armV4i 的过来。这样问题就能解决

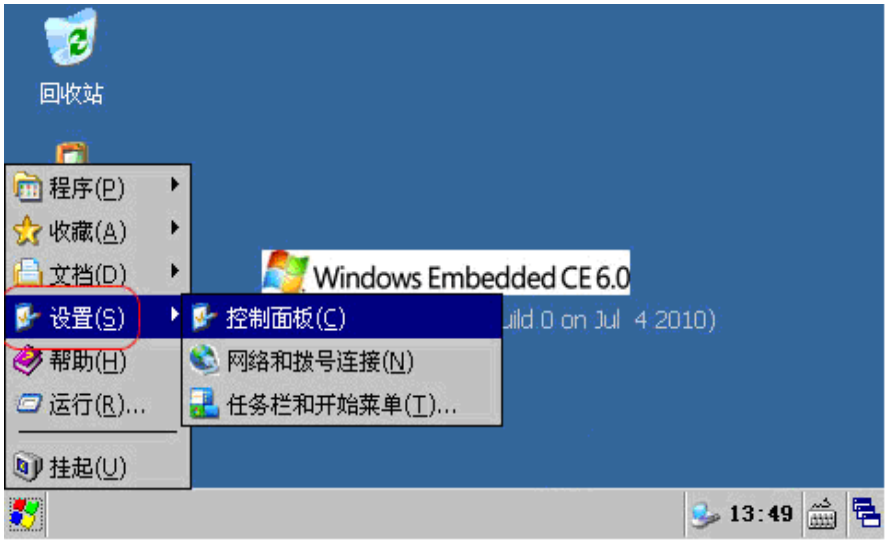
如果以上操作还不能连接，请确定打开了 DHCP 服务，因为有时候如果没开 DHCP 的话，active 是连接不上的，

打开 DHCP 服务方法：开始→运行→输入 services.msc 检查 “DHCP Client”服务，如果没启动手动启动。 基本上就没问题了。

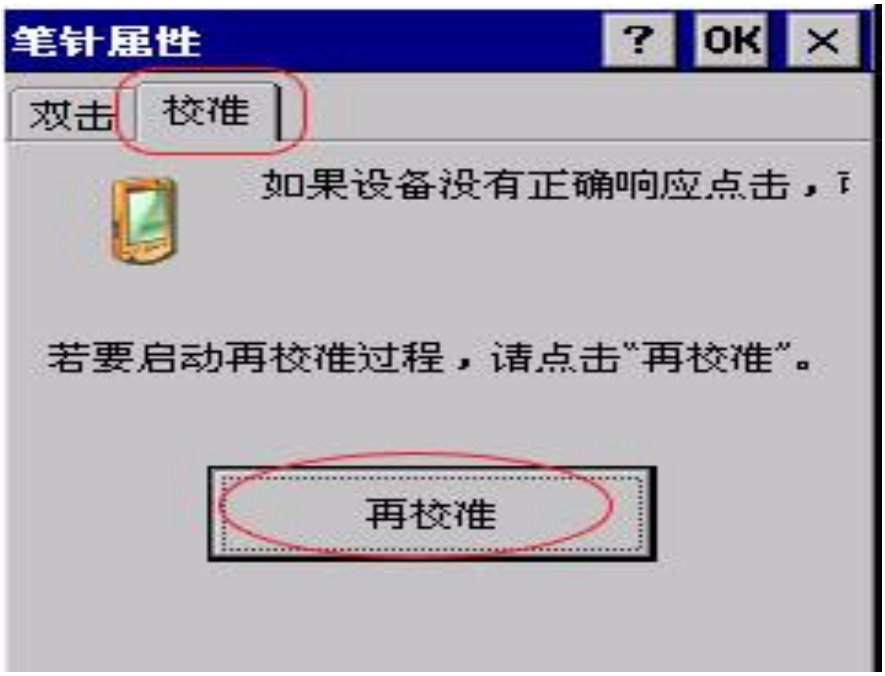
5.5 校正触摸屏

新机器，第一次出厂或者使用久了，发现触摸屏不准了，需要校正一下，方法如下：

1. 进入 WINCE 的控制面板



2. 双击“笔针”进入校屏程序



3. 依次点击以上图标的中心，

4. 5 点都校完后，自动保存并退出程序。

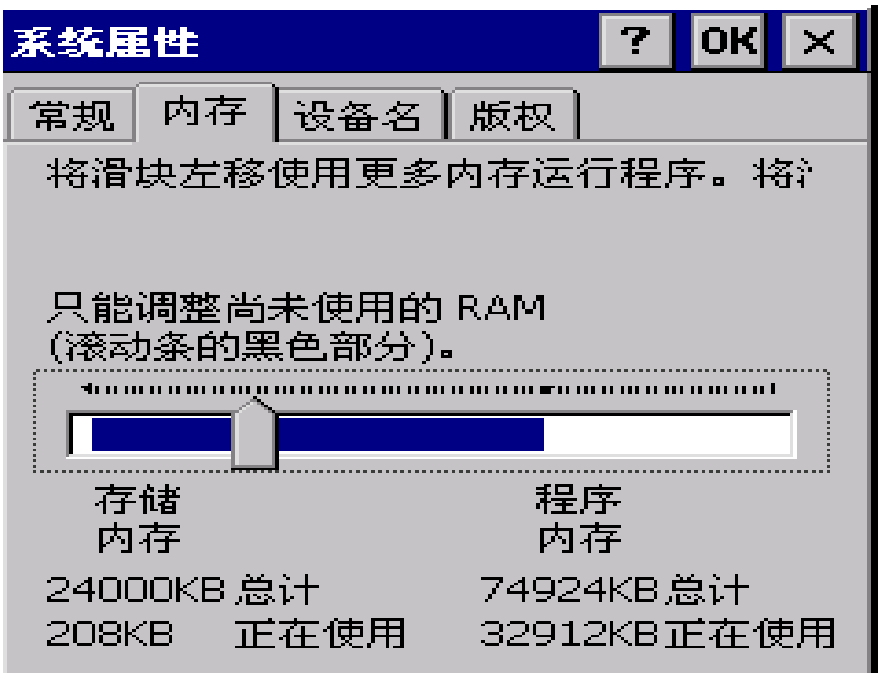
5.6 查看系统内存大小

1. 进入 控制面板

2. 双击“系统”查看系统属性

3. 查看内存大小，

从上图可以看出内存为 128MB，系统自身使用了一部分，余下的 98924KB 是可



以使用的，可以通过调整滑块来调整存储内存和程序内存的大小，右边的部份，是程序内存，如果让它大一点，那么程序会运行得更流畅，一般存储内存有 10MB 左右就够用了，两部分内存都不能调整的太小，调整的太小会导致数据丢失，甚至是死机。

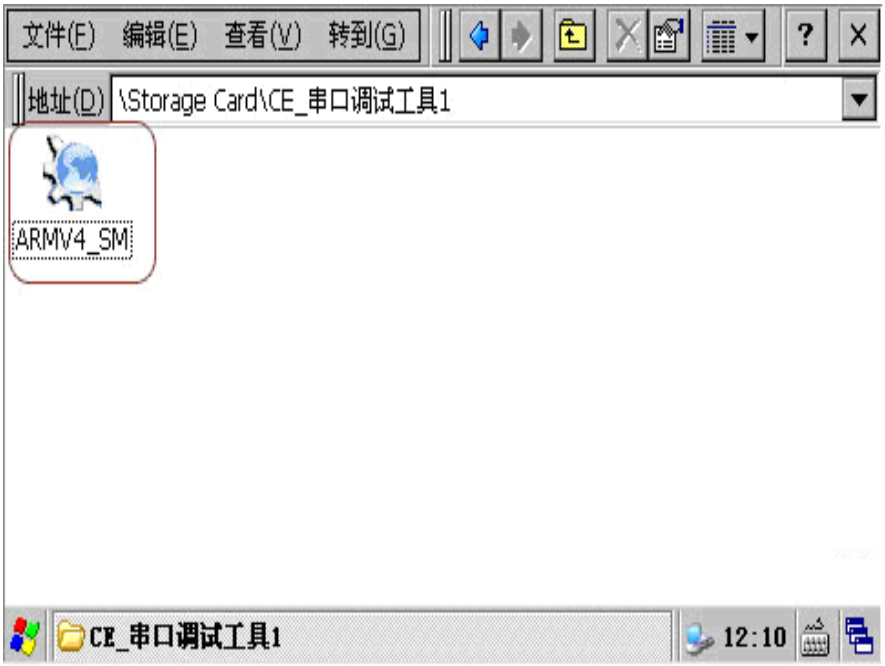
5.7 WinCE 下使用串口调试助手

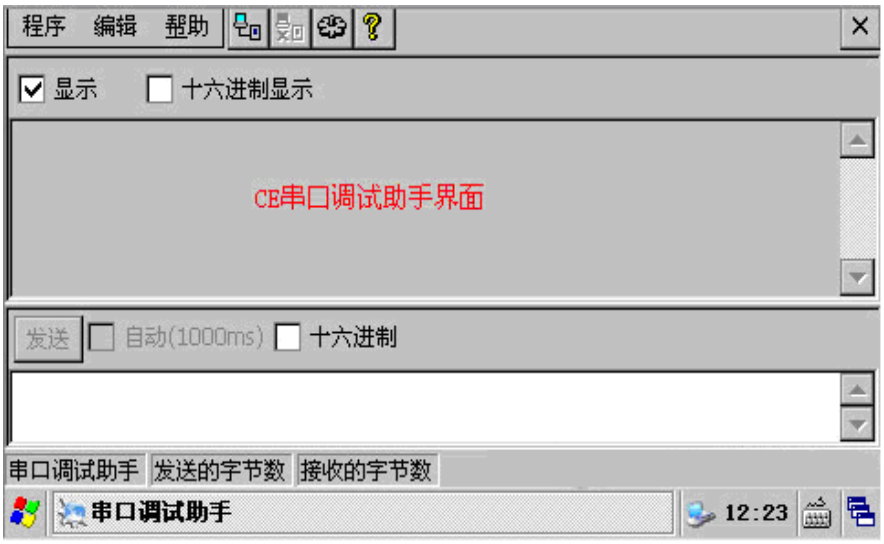
为了开发方便，我们光盘里，符上了一个 CE 下的串口调试工具，用户如果常要用到，也可以把它编译进 NK 里，这样，就要可以在“我的设备”→“Windows”目录下找到它，把第三方软件编译进 NK 的方法，请查看前面的章节“3.5 把程序或文件编译进 NK”，

该工具软件为网上牛人分享的第三方软件，没有源代码。

1. 运行

把“ARMV4_SM.EXE”拷贝到 WinCE 上，可以通过同步，也可以把 T 卡拔出来拷贝，在 WinCE 里运行它





2. WinCE 端打开 COM 口

X210-II 开发板上有两个 UART 口（串口），COM0 用于系统调试信息输出口，不能作

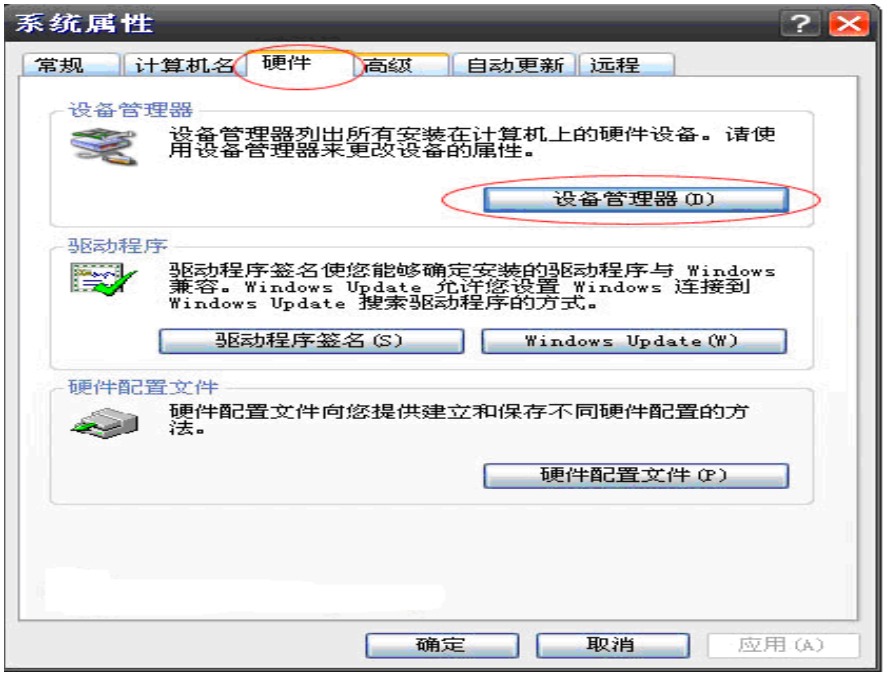
为它用，所以只能使用 COM2 为测试，

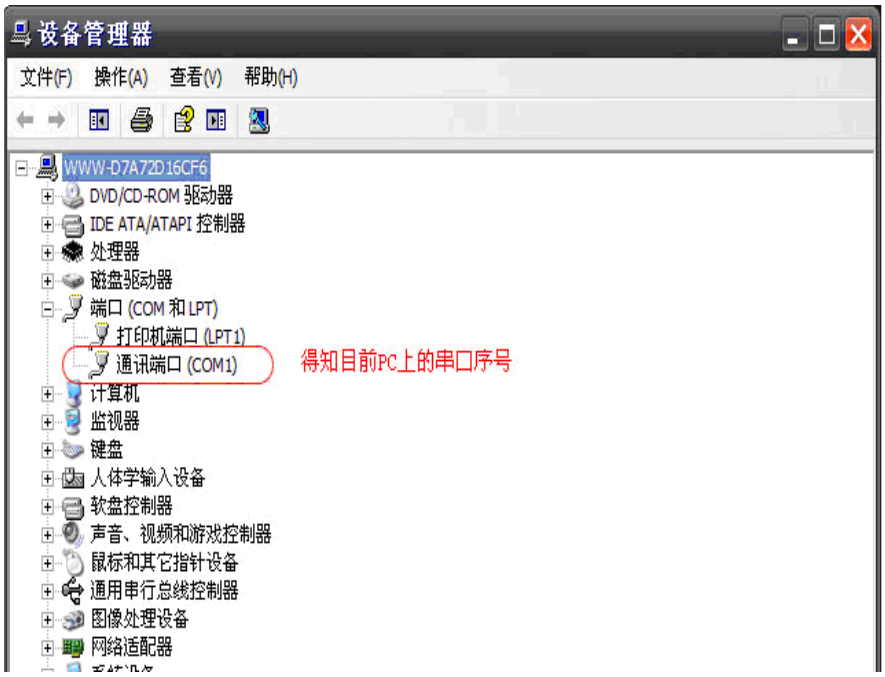
在 CE 端串口调试助手进行设置： COM2，115200，8，无，1

3. 硬件连接

用直通串口线一头接开发板的“COM2”,另一端接 PC 的串口上，注意如果 PC 上有多个口，要搞清楚实际接的为哪一个串口，可以在 PC 上的系统属性里查看串口为 COM 号，

方法：桌面鼠标右击“我的电脑”→“属性”→“硬件”→“设备管理器”→查看“端口（COM 和 LPT）”项，就能知道目前您电脑上串口是 COM 号了。





如上图可以看出，目前我电脑上的串口为 COM1，

双击上图的”COM1”可以设置它的参数，也可以在打开 PC 端的串口调试工具时再设置参数

设置串口通信参数，要与 WINCE 端的设置参数一致。

4. PC 端打开 COM 口

5.8 WinCE 驱动调试助手使用

在实际的项目开发中，驱动程序需要反复的调试，才能形成最终完善的驱动程序。如果我们每次修改那么一点代码，有时就为修改一条打印语句，也要先 build 一下生成动态链接库，再 make 生成系统映像，然后再烧写到板子上，这样调试难免效率低下。而在 Linux 平台下，要调试一个驱动程序，只用 insmod,remod 就可以了，相当的方便。那么在 WinCE 下，是否有捷径可走呢？

WINCE 的驱动包括流接口驱动和本机驱动，本机驱动相对比较复杂，还没有找到比较快捷的方法。而流接口驱动，可以借助驱动调试助手来完成。目前最新的驱动调试助手版本为 V2.9，下面将以 LED 驱动来讲解驱动调试助手的具体使用方法。

第一步：开启开发板，进入 WINCE 系统，并与 PC 机同步；

第二步：找到光盘中的驱动调试助手文件夹，将“驱动调试助手”这个文件夹拷备到开发板的 nand-disk 盘下；

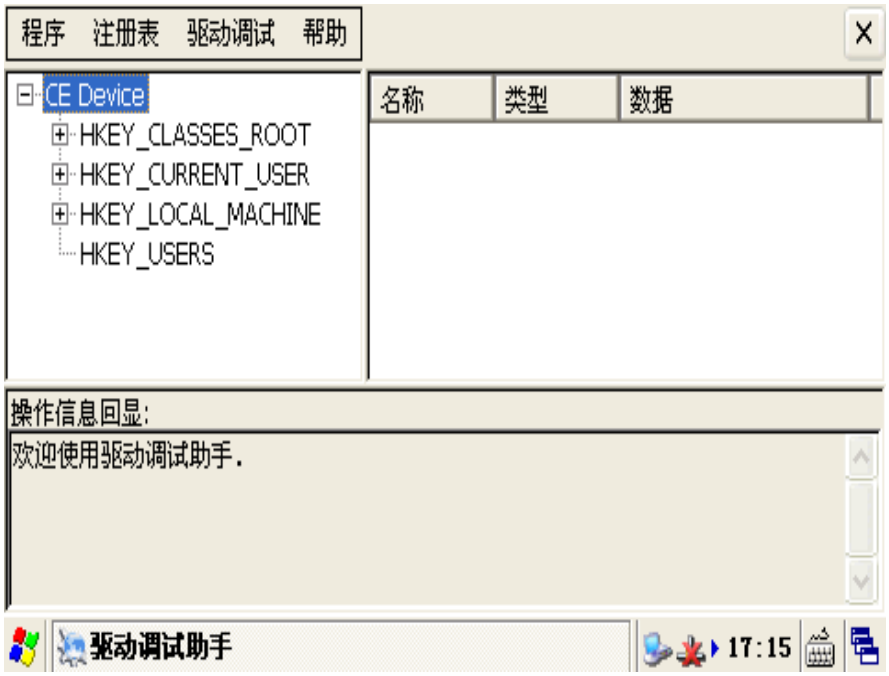
第三步：在 PB6.0 平台下编译修改后的 LED 驱动，在工程目录下会生成 ARMEasy6410_LED.dll 动态链接库，将这个新生成的库文件拷备到开发板的驱动调试助手文件夹；

第四步：修改驱动调试助手目录下的*.reg 注册表文件，修改内容如下：

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\SetLED]
    "Index"=dword:1
    "Prefix"="LED"
    "Dll"="\\ARMEasy6410_LED.dll"
    "Order"=dword:1
```

同时将该注册表文件重命名为 LED.reg；

第五步：在开发板的驱动调试助手目录双击 DM.exe 文件，弹出驱动调试助手的窗口如下图：



点击驱动调试->驱动列表，可以看到列表中有 SetLED 驱动已启用，如下图所示。

点击卸载，调试助手会提示驱动卸载成功，见下图。

点击退出，再点击驱动调试->导入 REG，弹出寻找文件的对话框，找到刚才拷到板上的注册表文件，如下图，点击 OK。

这时调试助手会提示导入注册表文件完成，同时注册表显示路径会指向 SetLED 的路径，同时显示相应的键值，如下图：

再次点击驱动调试->浏览 DLL，以相同的方式找到对应的动态链接库文件，弹出设置注册表的对话框，如下图

将 Prefix 一栏设置为驱动所定义的名称，这里填 LED，这里不区分大小写，调试助手会自动纠正为大写，点击确认，会提示设置设置注册表成功，同时注册表列表会指向 DMDrv0，这里在 Display 驱动的下一栏，右边为 DMDrv0 对应的键值，可以看到这时右边的 Dll 路径已经修改为我们存放在开发板上的路径了，见下图：

在保证选中 DMDrv0（即蓝底选中状态）的情况下，点击驱动调试->激活驱动，这时会提示驱动加载成功，见下图所示：

这时，我们就已经更新完我们需要修改的驱动程序了，我们可以通过观看我们修改的代码或是打印信息确定驱动是否已经改为我们所需要的。也许有人会觉得，怎么会要这么多步骤？那还不如我重新 make 一下来得快呢？没错，这里步骤是多了点，也仅是第一次步骤多一点而已。下面继续讲续如果我们执行了这一步骤后，又要重新更新驱动，是否会又要重新执行上面的步骤呢？

第六步：同样确保注册表选中 DMDrv0 的情况下，点击驱动调试->卸载驱动，调试助

手提示驱动卸载成功；

第七步：将我们再次修改的 DLL 文件拷备到开发板的相同文件夹，覆盖前面的 DLL 文件。注意，在执行这一步之前，一定要先卸载该驱动，否则无法覆盖成功！

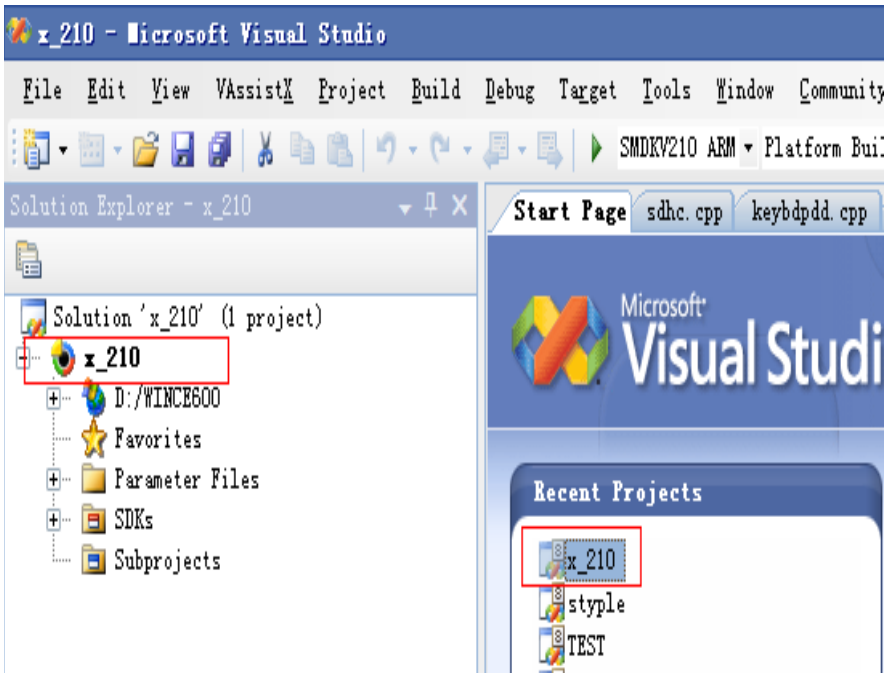
第八步：确保注册表选中 DMDrv0 的情况下，点击驱动调试->激活驱动，这时调试助手再次提示驱动加载成功，我们新更新的驱动程序已经就这样再次轻而易举的加载进去了！

第九步：如果我们重启了系统，是否要再次执行第一次加载驱动的那些烦锁步骤呢？完全没有必要。不防重启开发板，进入驱动调试助手文件夹，打开 DM.exe，这时注册表没有自动跳转到 DMDrv0 目录，我们可以手动找到这个目录，在 HKEY_LOCAL_MACHINE->Drivers->DMDrv0 这里，然后将我们新编辑的 DLL 文件拷到前面指定的路径，点击驱动调试->激活驱动，这样我们新编译的驱动就正常加载了。

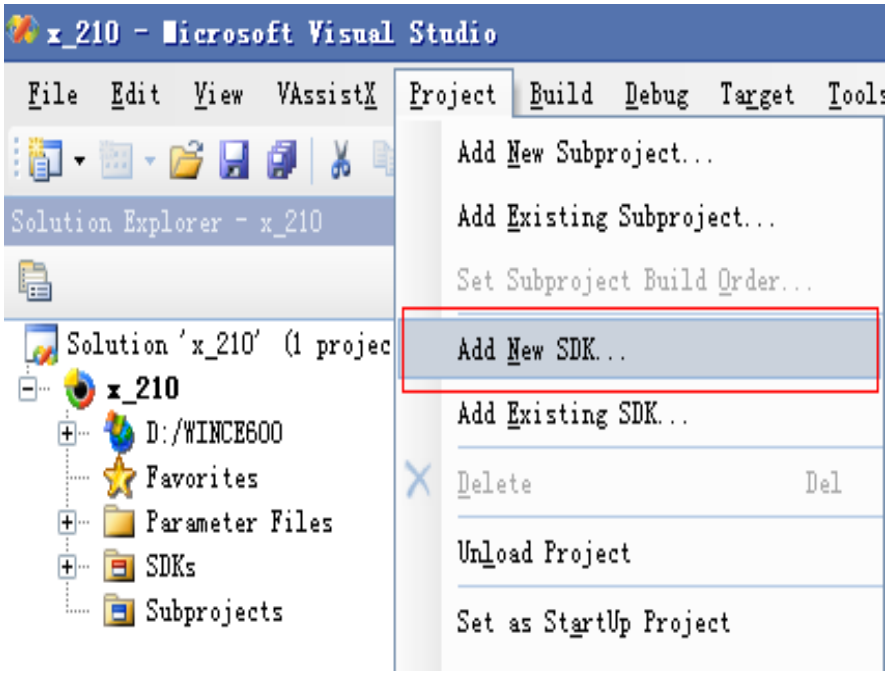
第6章 导出工程 SDK

6.1 生成并导出 SDK

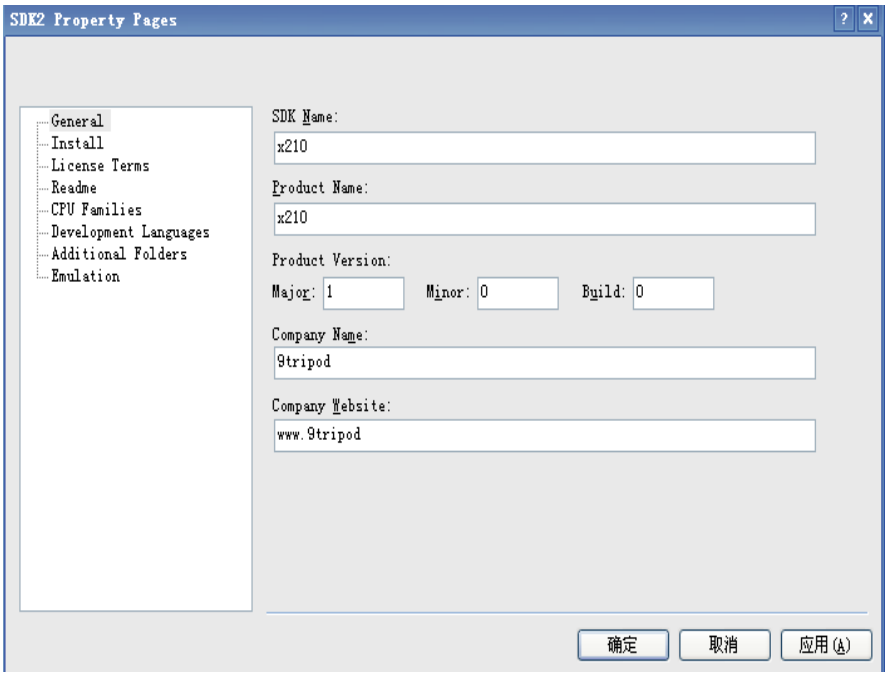
1. 打开已经定制并编译好的 WinCE6.0 工程

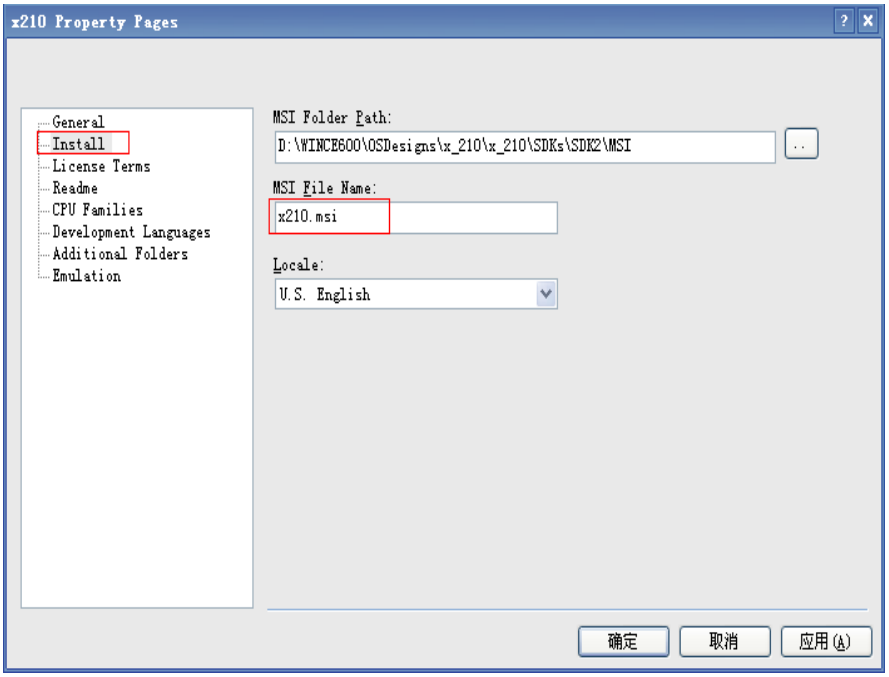


2. Project->Add New SDK

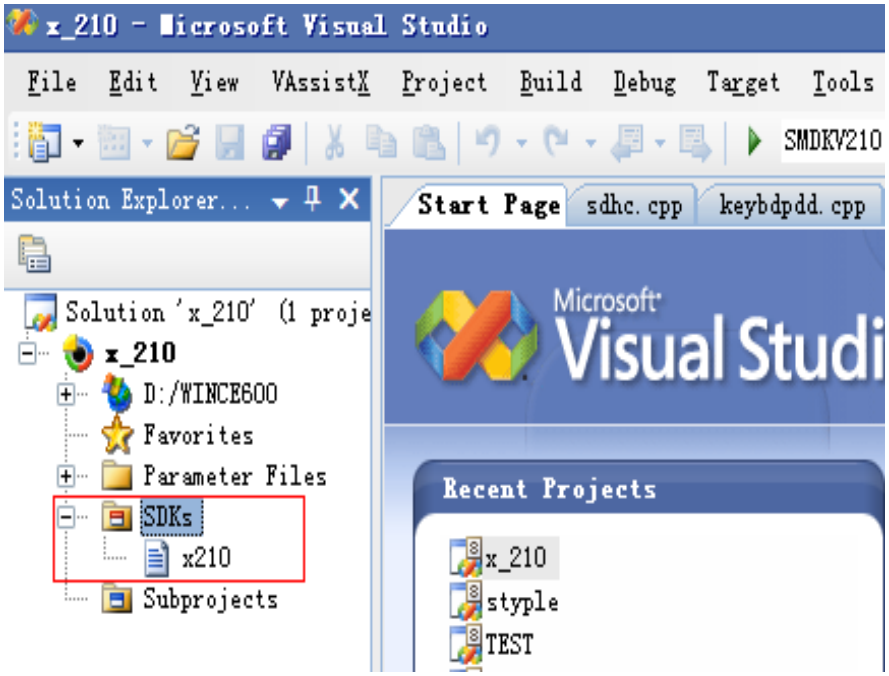


3. 填写 SDK 信息



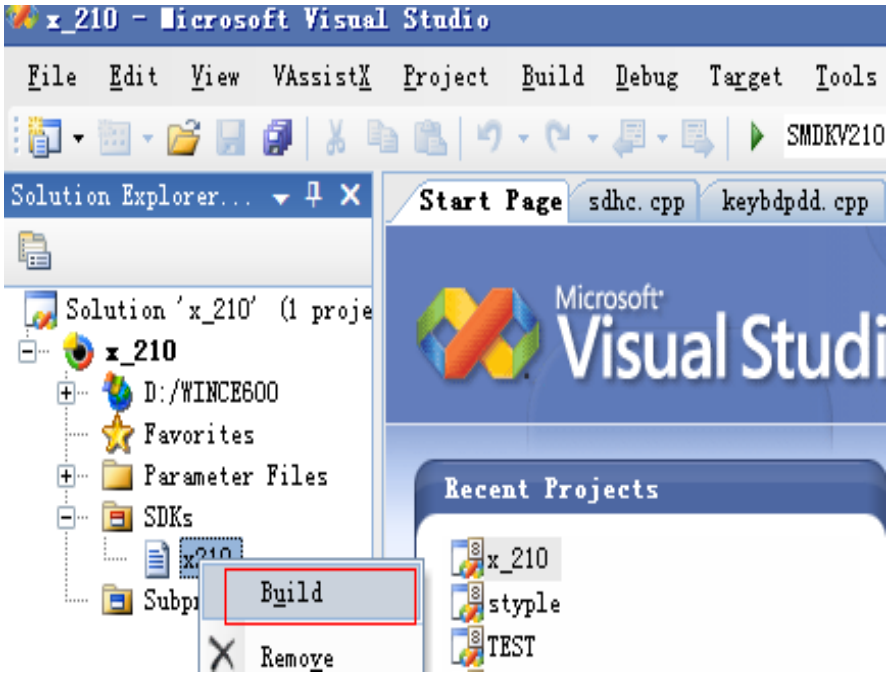


4. 成功后可看到如下界面:

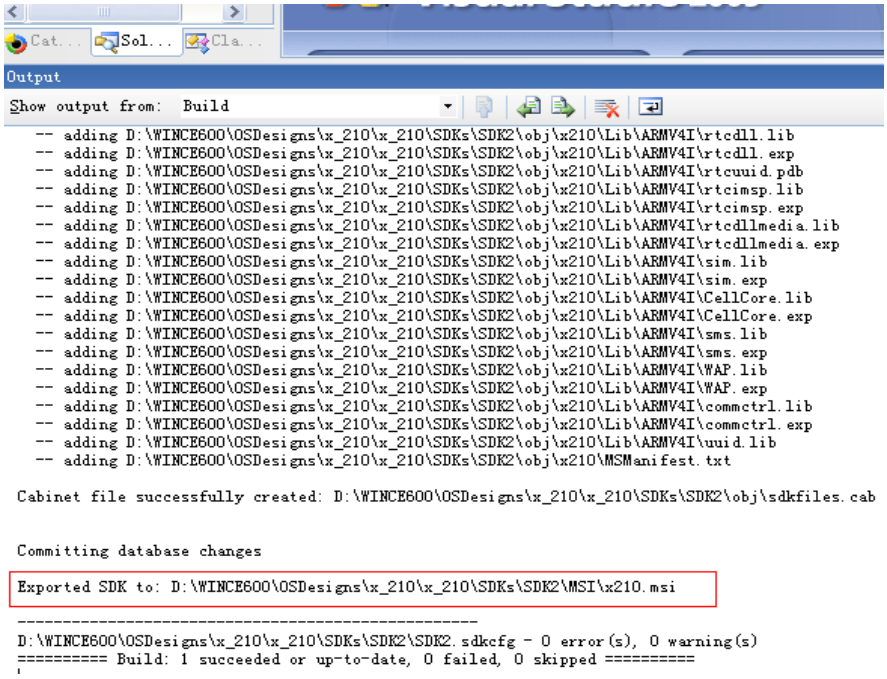


6.2 编译 SDK

1. 右键点击 SDKs->x210->Build



2. 编译后如下

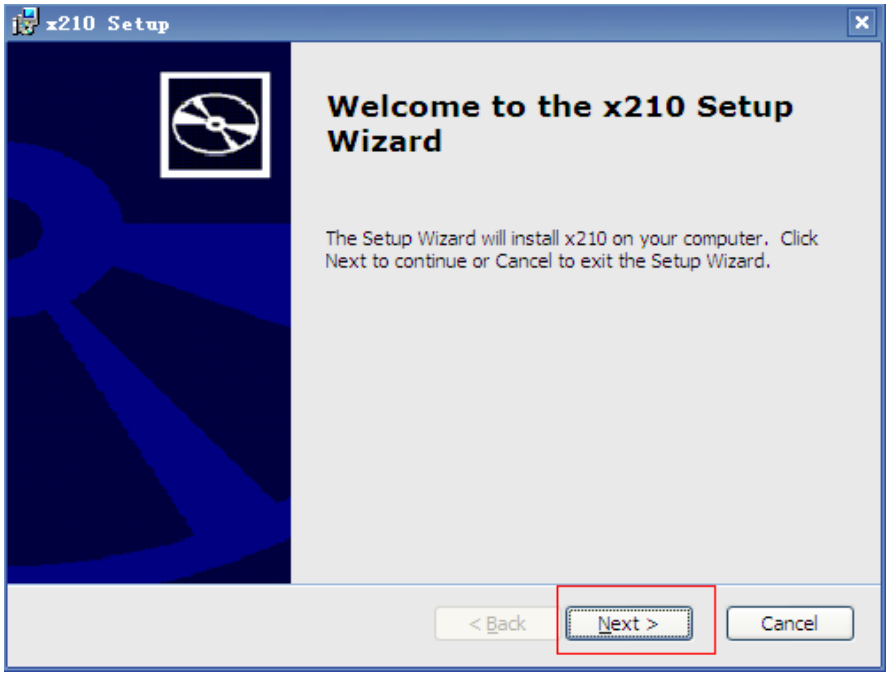


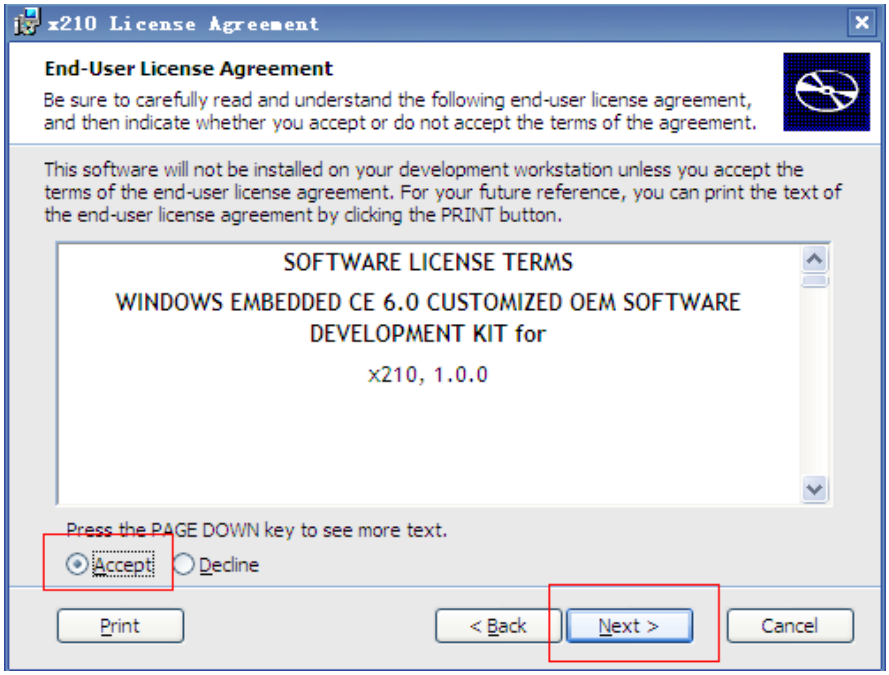
编译完成后,SDK 被成功导出.目录在 D:\WINCE600\OSDesigns\x_210\x_210\SDKs。此时，SDK 可以提交给应用程序开发人员进行完装.并开发上层应用程序了。

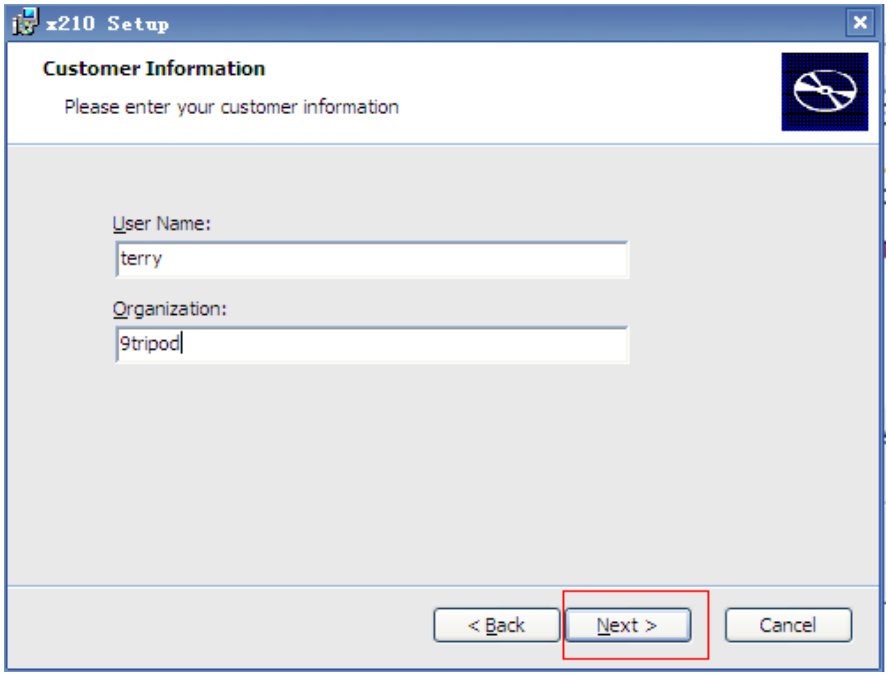
6.3 安装 SDK

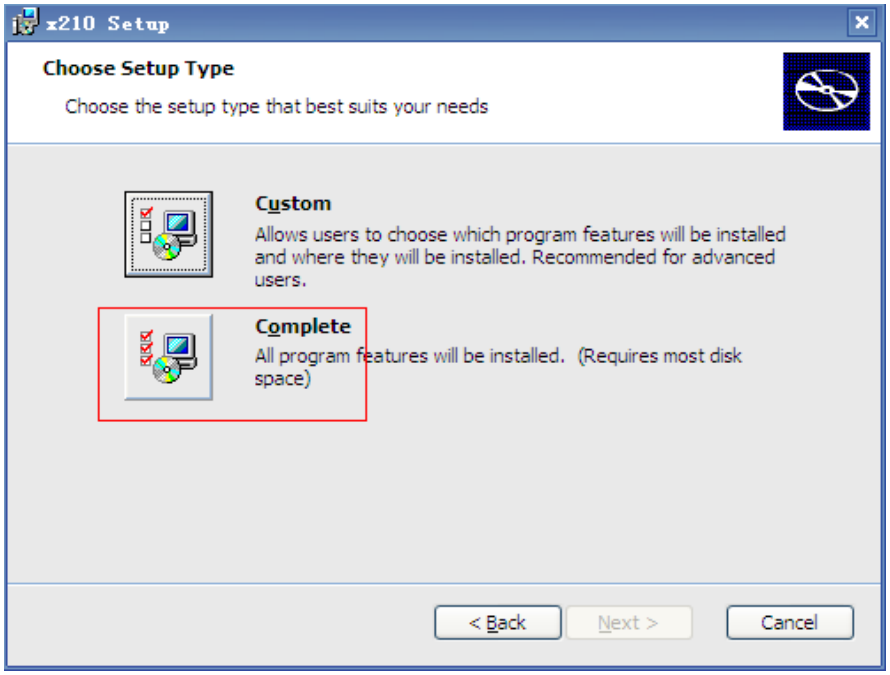
开发应用程序需先安装对应平台的 SDK。

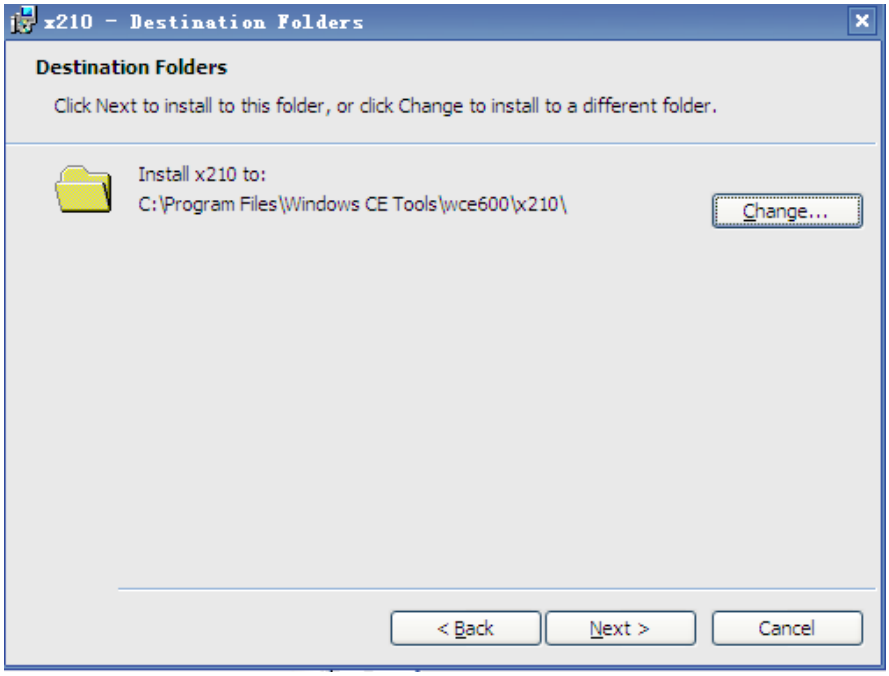
直接双击: D:\WINCE600\OSDesigns\x_210\x_210\SDKs\SDK2\MSI\ x210.msi

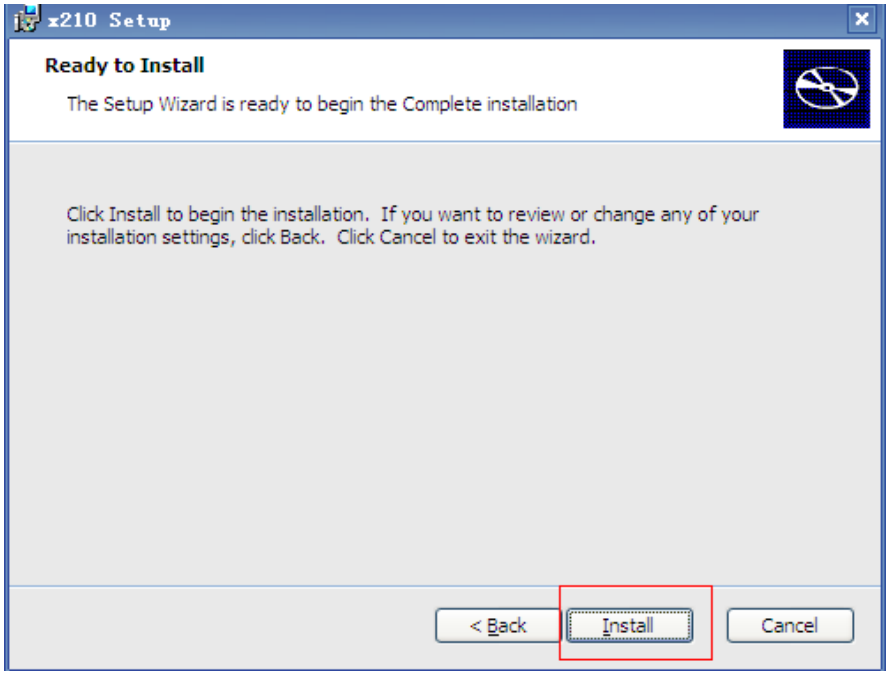

x210 Setup
Welcome to the x210 Setup Wizard
The Setup Wizard will install x210 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
< Back
Next >
Cancel

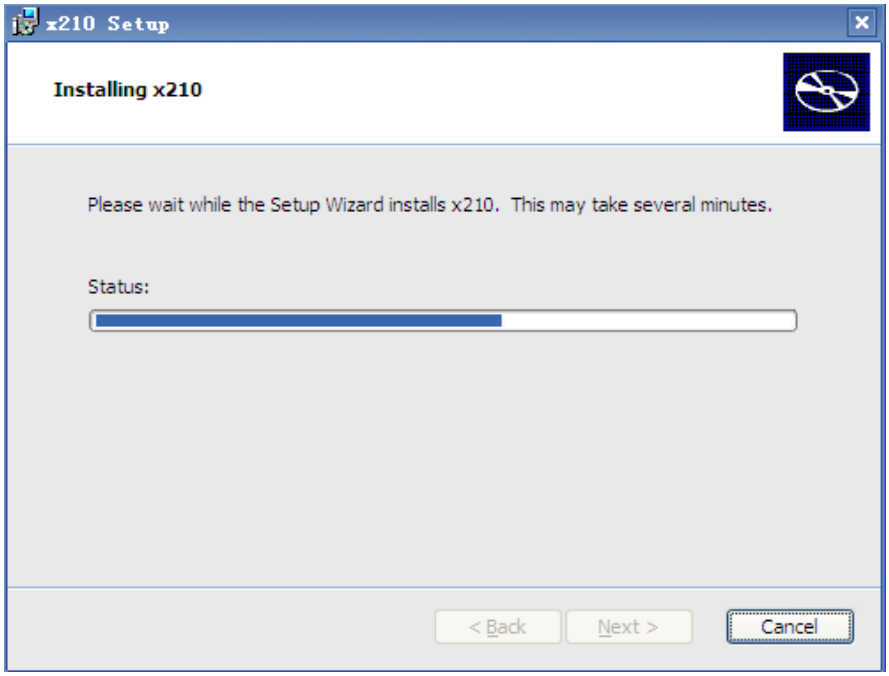

x210 License Agreement
End-User License Agreement
Be sure to carefully read and understand the following end-user license agreement, and then indicate whether you accept or do not accept the terms of the agreement.
This software will not be installed on your development workstation unless you accept the terms of the end-user license agreement. For your future reference, you can print the text of the end-user license agreement by clicking the PRINT button.
SOFTWARE LICENSE TERMS
WINDOWS EMBEDDED CE 6.0 CUSTOMIZED OEM SOFTWARE DEVELOPMENT KIT for
x210, 1.0.0
Press the PAGE DOWN key to see more text.
Accept
Decline
Print
< Back
Next >
Cancel

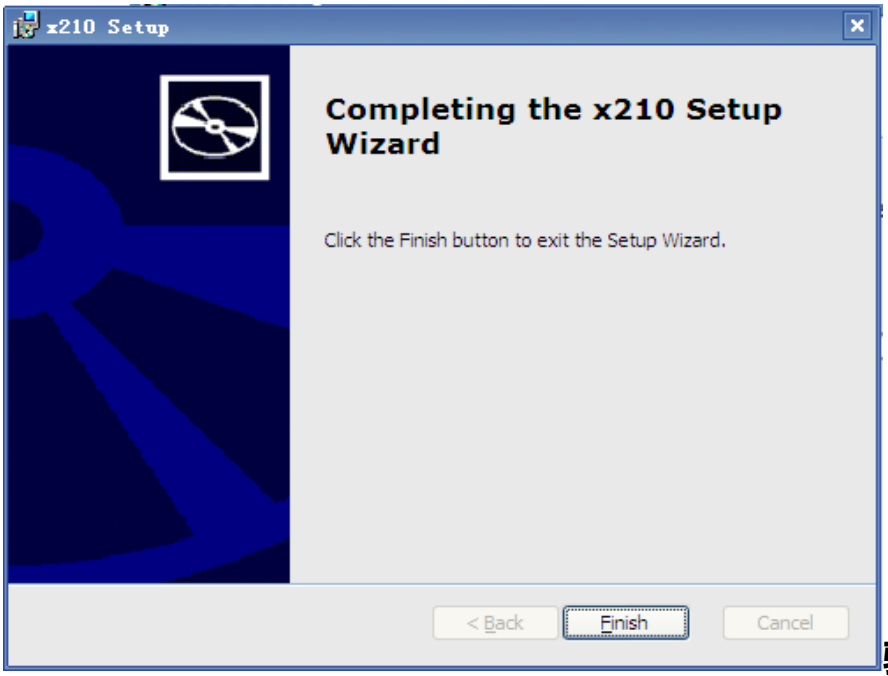

x210 Setup
Customer Information
Please enter your customer information
User Name:
terry
Organization:
9tripod
< Back
Next >
Cancel


x210 Setup
Choose Setup Type
Choose the setup type that best suits your needs
Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.
Complete
All program features will be installed. (Requires most disk space)
< Back
Next >
Cancel


x210 - Destination Folders
Destination Folders
Click Next to install to this folder, or click Change to install to a different folder.
Install x210 to:
C:\Program Files\Windows CE Tools\wce600\x210\
Change...
< Back
Next >
Cancel


x210 Setup
Ready to Install
The Setup Wizard is ready to begin the Complete installation
Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
< Back
Install
Cancel


x210 Setup
Installing x210
Please wait while the Setup Wizard installs x210. This may take several minutes.
Status:
< Back
Next >
Cancel



第7章 驱动开发实例

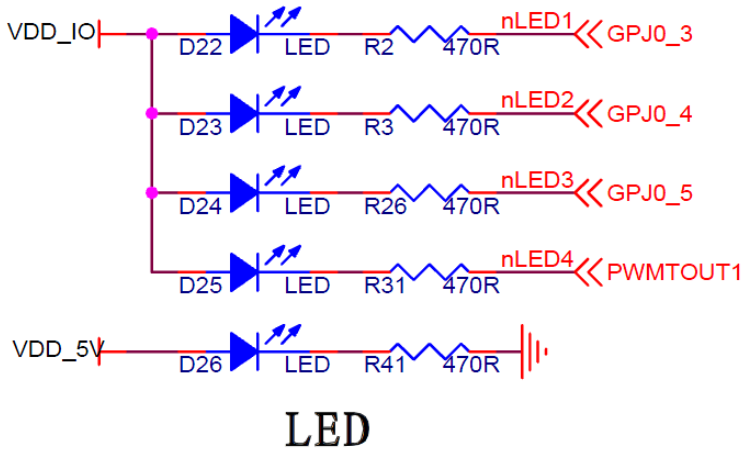
驱动开发涉及硬件，驱动，系统方面的知道，而从事驱动开发的工程师主要来自两类专业，一是计算机专业，另一类是电子专业，对于计算机专业出身的人来说，他的强项是对数据结构，OS 和应用软件方面是比较有优势的，而弱点就是对硬件电路和 CPU 寄存器的理解会相对弱些。相反，学电子的工程师对硬件原理的把控能力比较强，而应用软件方面的的知识是相对不足。本人出自电子专业，所以仅以驱动为重点，本章以 X210-II 开发板为平台。为驱动开发提供开发实例，供初学者参考。

7.1 LED 流驱动移值

流驱动是 WinCE 驱动开发中最基础，也是最重要的驱动模型之一。采用流驱动模型的驱动有 LED,UART，CAMERA，I2C，SPI，AUDIO，等等，所以把流驱动做为第一个驱动来说是很有必要的。通过本章的学习，你可了解流驱动的开发移植过程，使用到 GPIO 的设置。此时你可阅读进一步了解复杂一点的流接口驱动。

由于很多资料介绍都是以原理分析为主，对于初学者来说，在掌握原理的情况下，移植过程更为重要。所以接来的，以 LED 驱动为例，重点讲解驱动移植的详细过程。

7.1.1 LED 电路原理分析



LED

从上面的原理图来看，四个发光二极管 D22、D23、D24、D25 正极接电源 VDD_IO，而负板分别接到了 CPU 的 GPJ0_3、GPJ0_4、GPJ0_5 和 PWMTOUT1 引脚。由此，可通过控制负板的电平来点亮和灭 LED。拿 D22 举例说明，当 GPJ0_3 电平为高时，D22 两端没有压差，D22 灯灭。反之当 GPJ0_3 为低电平时，发光二极管 D22 两端有电压，通电点亮。通过调节 GPJ0_3 引脚的电平高低保持时间的长短，可做闪烁等效果。

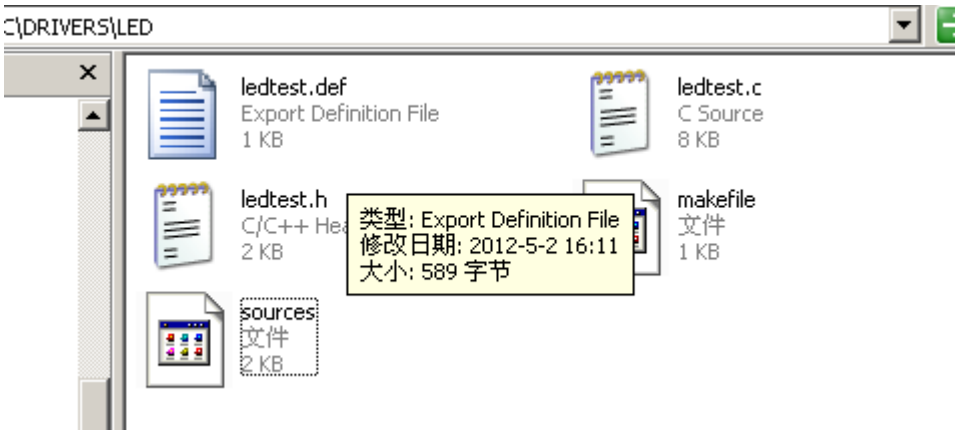
注意：D25 所接的引脚 PWMTOUT1，可直接采用 PWM 输出来控制 D25 的闪烁。 这个电路接法可用于验证 PWM 输出功能。

7.1.2 LED 驱动移植

流驱动的接口是 MS 已定义好的，我们只需要按照规定去实现即可。

1. 创建 LED 驱动文件夹

创建一个名为 LED 的文件夹，创建以下 5 个文件。



2. 改写 LEDTEST.DEF 和 SOURCE 文件

从其他驱动中拷贝 sources 和 XXX.def 文件。改写

Source 文件内容如下：

```
TARGETNAME=ledtest
TARGETTYPE=DYNLINK
```

```
RELEASETYPE=PLATFORM
DLLENTRY=_DllEntry
DEFFILE=ledtest.def
PRECOMPILED_INCLUDE=ledtest.h
TARGETLIBS= \
    $(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\coredll.lib      \
    $(_COMMONOAKROOT)\lib\$(_CPUINDPATH)\ceddk.lib      \
    $(_TARGETPLATROOT)\lib\$(_CPUINDPATH)\drvlib_mem.lib \
    $(_TARGETPLATROOT)\lib\$(_CPUINDPATH)\drvlib.lib \
SOURCES= \
    ledtest.c
```

以上红色部分是改写过的。

LEDTEST.DEF 内容如下：

```
LIBRARY     LED
EXPORTS
    LED _Init
    LED _Deinit
    LED _Open
    LED _Close
    LED _Read
    LED _Write
    LED _Seek
    LED _IOControl
    LED _PowerUp
    LED _PowerDown
```

红色字体部分是我改后的，这次示例是从 POWERCONTROL 驱动改造而来。

3. 创建 LEDTEST.H 头文件和 LEDTEST.C 源文件

这里重点介绍 3 个函数，分别是 DllEntry 函数、LED_Init 函数和 LED_IOControl 函数。

- DllEntry(HINSTANCE hinstDll, DWORD Reason, LPVOID Reserved)

这个函数是动态链接库的入口，每个动态链接库都需要输出这个函数，它只在动态库被加载和卸载时被调用，也就是设备管理器调用 LoadLibrary 而引起它被装入内存和调用 UnloadLibrary 将其从内存释放时被调用，因而它是每个动态链接库最早被调用的函数，一般用它做一些全局变量的初始化。

```
BOOL
DllEntry( HINSTANCE    hinstDll,    DWORD dwReason,    LPVOID lpReserved )
{
    if (dwReason == DLL_PROCESS_ATTACH)
    {
      ///    DEBUGREGISTER(hinstDll);
      //    DisableThreadLibraryCalls ((HMODULE)hinstDll);
        DBGMSG(LED_FUNC,(TEXT("[LED] %s : Process Attach\n"), _T(__FUNCTION__)));
```

```
    }
    else if (dwReason == DLL_PROCESS_DETACH)
    {
        DBGMSG(LED_FUNC,(TEXT("[LED] %s : Process Detach\n"), _T(__FUNCTION__)));
    }
    return TRUE;
}
```

参数：

hinstDll：DLL 的句柄，与一个 EXE 文件的句柄功能类似，一般可以通过它在得到 DLL 中的一些资源，例如对话框，除此之外一般没什么用处。

Reason: 一般我们只关心两个值： DLL_PROCESS_ATTACH 与 DLL_PROCESS_DETACH，Reason 等于前者是动态库被加载，等于后者是动态库被释放。所以，我们可以在 Reason 等于前者是初始化一些资源，等于后者时将其释放。

- DWORD LED_Init(DWORD dwContext)

它是驱动程序的动态库被成功装载以后第一个被调用的函数。其调用时间仅次与 DllEntry,而且，当一个库用来生成多于一个的驱动程序实例时仅调用一次 DllEntry，而 LED_Init 会被调用多次。驱动程序应当在这个函数中初始化硬件，如果初始化成功，就分配一个自己的内存空间(通常用结构体表示)，将自己的状态保存起来，并且将此内存块的地址做为一个 DWORD 值返回给上层。设备管理器就会用在调用 LED_Open 时将此句柄传回，我们就能访问自己的状态。如果初始化失败，则返回 0 以通知这个驱动程序没有成功加载，先前所分配的系统资源应该全部释放，此程序的生命即告终至。

当这个函数成功返回，设备管理器对这个程序就不做进一步处理，除非它设置了更多的特性。至此一个各为 LED 的设备就已经加载成功，当用户程序调用 CreateFile 来打开这个设备时，设备管理器就会调 LED_Open 函数。

```
DWORD LED_Init( DWORD dwContext)
{
    DBGMSG(LED_FUNC,(TEXT("[LED:INF] ++%s(0x%08x)\n"), _T(__FUNCTION__), dwContext));
    if (AllocResources() == FALSE)
    {
        DBGMSG(LED_INFO,(TEXT("[LED:ERR] %s->AllocResources() Failed \n"),
_T(__FUNCTION__)));
        goto CleanUp;
    }
    DBGMSG(LED_FUNC,(TEXT("[LED:INF] --%s()\n"), _T(__FUNCTION__)));
    return TRUE;
CleanUp:
    DBGMSG(LED_INFO,(TEXT("[LED:ERR] --%s : Failed\n"), _T(__FUNCTION__)));
    LED_Deinit(0);
    return FALSE;
}
```

参数：

dwContext：系统传入的注册表键，通过它可以讲到我们在注册表中设置的配置信息。

AllocResources()分配 LED 资源，如 GPIO 的映射等初始化等。

- BOOL LED_IOControl(
  DWORD pContext,
  DWORD dwCode,
  PBYTE pBufIn,
  DWORD dwLenIn,
  PBYTE pBufOut,
  DWORD dwLenOut,
  PDWORD pdwActualOut)

几乎可以说一个驱动程序的所有功能都可以在这个函数中实现。对于一类 CE 自身已经支持的设备，它们已经被定义了一套 IO 操作定，我们只需按照各类设备已经定义的内容去实现所有的 IO 操作。但当我们实现一个自定义的设备时，我们就可以随心所欲定义我们自己的 IO 操作。

```
BOOL    LED_IOControl(DWORD pContext, DWORD dwCode, PBYTE pBufIn, DWORD dwLenIn, PBYTE
pBufOut, DWORD dwLenOut, PDWORD pdwActualOut)
{
     BOOL bRet = TRUE;
    DWORD dwIndex;
    if ( !( (dwCode == IOCTL_LED_SET_ON)))              // 先判断 dwCode 是否合法
    {
        SetLastError (ERROR_INVALID_ACCESS);
        return FALSE;
    }
    switch(dwCode)
    {
    case   IOCTL_LED_SET_ON:   // 这个 dwCode 是自定义，上层应用程序调用时应做相同定义
        if ((dwLenIn < sizeof(DWORD)) || (NULL == pBufIn))
        {
            SetLastError (ERROR_INVALID_PARAMETER);
            bRet = FALSE;
            break;
        }
        dwIndex = (DWORD)(*pBufIn);
        switch(dwIndex)
        {
        case 1:      // LED ON          用户自己在下面的 CASE 中实现想要功能。此例是全亮。
              // ON
              Set_PinData(g_pGPIOreg, GPJ03_Output, 0);
              Set_PinData(g_pGPIOreg, GPJ04_Output, 0);
              Set_PinData(g_pGPIOreg, GPJ05_Output, 0);
              Set_PinData(g_pGPIOreg, GPD01_Output, 0);
```

```
                break;
            case 2:       // LED Off
            case 3:        // X210-II      OFF
                Set_PinData(g_pGPIOreg, GPJ03_Output, 1);
                Set_PinData(g_pGPIOreg, GPJ04_Output, 1);
                Set_PinData(g_pGPIOreg, GPJ05_Output, 1);
                Set_PinData(g_pGPIOreg, GPD01_Output, 1);
                break;
            default:
                SetLastError (ERROR_INVALID_PARAMETER);
                bRet = FALSE;
            }
            break;
    }
    return bRet;
}
```

参数：

pContext　LED_Open 返回给上层的那个句柄，即我们自己定义的，用来存放程序所有信息的一个结构。

dwCode　IO 操作码，如果是 CE 已经支持的设备类，就用它已经定义好码值，否则就可以自己定义。

pBufIn 传入的 Buffer,每个 IO 操作码都会定义自己的 Buffer 结构

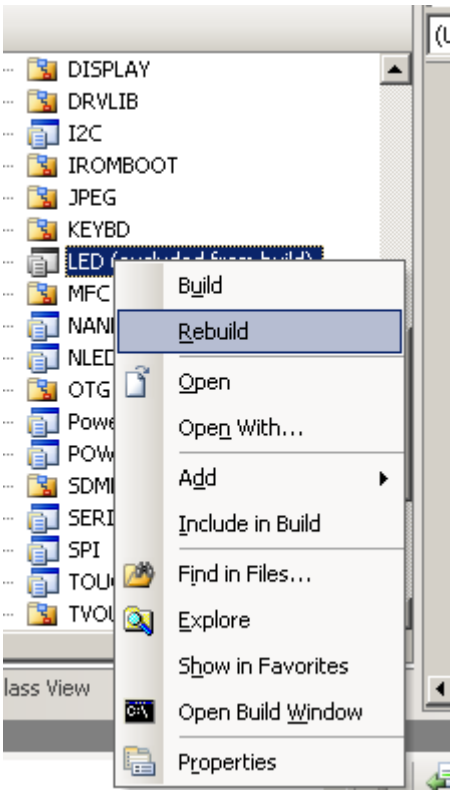
dwLenIn pBufIn 以字节记的大小

pBufOut　dwLenOut 分别为传出的 Buffer，及其以字节记的大小

pdwActualOut 驱动程序实际在 pBufOut 中填入的数据以字节记的大小

其中，前两个参数是必须的，其它的任何一个都有可能是 NULL 或 0。所以，当给 pdwActualOut 赋值时应该先判断它是否为一个有效的地址。

4. 编译 LEDTEST.DLL



编译成功后如下

```
BUILD: [00:0000000024:PROGC ] Building LIB Pass in D:\WINCE600\PLATFORM\SMDKV210\SRC\DRIVERS\LED\ directory.
BUILD: [01:0000000034:PROGC ] Linking D:\WINCE600\platform\SMDKV210\lib\ARMV4I\retail\ledtest.lib
BUILD: [00:0000000045:PROGC ] Building LINK Pass in D:\WINCE600\PLATFORM\SMDKV210\SRC\DRIVERS\LED\ directory.
BUILD: [01:0000000056:PROGC ] Linking D:\WINCE600\platform\SMDKV210\target\ARMV4I\retail\ledtest.dll
BUILD: [00:0000000084:PROGC ] Saving D:\WINCE600\PLATFORM\SMDKV210\Build.dat.
BUILD: [00:0000000086:PROGC ] Done.
BUILD: [00:0000000087:PROGC ]                          Files  Warnings  Errors
BUILD: [00:0000000088:PROGC ] Midl                         0         0       0
BUILD: [00:0000000089:PROGC ] Message                      0         0       0
BUILD: [00:0000000090:PROGC ] Precomp Header               0         0       0
BUILD: [00:0000000091:PROGC ] Resource                     0         0       0
BUILD: [00:0000000092:PROGC ] MASM                         0         0       0
BUILD: [00:0000000093:PROGC ] SHASM                        0         0       0
BUILD: [00:0000000094:PROGC ] ARMASM                       0         0       0
BUILD: [00:0000000095:PROGC ] MIPSASM                      0         0       0
BUILD: [00:0000000096:PROGC ] C++                          0         0       0
BUILD: [00:0000000097:PROGC ] C                            1         0       0
BUILD: [00:0000000098:PROGC ] Static Libraries             0         0       0
BUILD: [00:0000000099:PROGC ] Exe's                        0         0       0
BUILD: [00:0000000100:PROGC ] Dll's                        2         0       0
BUILD: [00:0000000101:PROGC ] Preprocess deffile           0         0       0
BUILD: [00:0000000102:PROGC ] Resx                         0         0       0
BUILD: [00:0000000103:PROGC ] CSharp Compile               0         0       0
BUILD: [00:0000000104:PROGC ] Other                        0         0       0
BUILD: [00:0000000105:PROGC ]
BUILD: [00:0000000106:PROGC ] Total                        3         0       0
BUILD: [00:0000000107:PROGC ]
BUILD: [00:0000000108:PROGC ] 0 Warnings,  0 Errors
BUILD: [00:0000000109:PROGC ] GetSystemTimes (seconds): Idle: 1     Kernel: 2     User: 0
BUILD: [00:0000000110:PROGC ] Elapsed  time  (seconds): 1
Build for Windows CE (Release 601) (Built on Aug 17 2006 15:18:52)
File names: Build.log Build.wrn Build.err Build.dat
D:\WINCE600\PLATFORM\SMDKV210\SRC\DRIVERS\LED\sources - 0 error(s), 0 warning(s)
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

此时可看到 ledtest.dll 已生成。

5. 加入到 NK 中

在 platform.reg 文件中加入以下注册表项

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Led]
    "Prefix"="LED"
    "Dll"="ledtest.dll"
```

```
"Flags"=dword:00010000
"Order"=dword:40
"IClass"="{A32942B7-920C-486b-B0E6-92A702A99B35}"
```

在 platform.bib 文件中加入以下内容

```
ledtest.dll     $(_FLATRELEASEDIR)\ ledtest.dll    NK        U
```

make 即可加入 NK 中。

其他函数请参考源码，同时网上很多资料介绍流驱动。

7.1.3 编写测试程序

驱动移植完成后，接下来就需要一个测试程序来检测我们的驱动功能。由于网上太多的资料介绍 win32 和 MFC 的知识，所以这里不再介绍。请参考网络上的资料。

7.1.4 GPIO 作业

当完成以上驱动移植后，基本上了解了流驱动的移植和 GPIO 的控制。现在大家可以再继续深入开发。

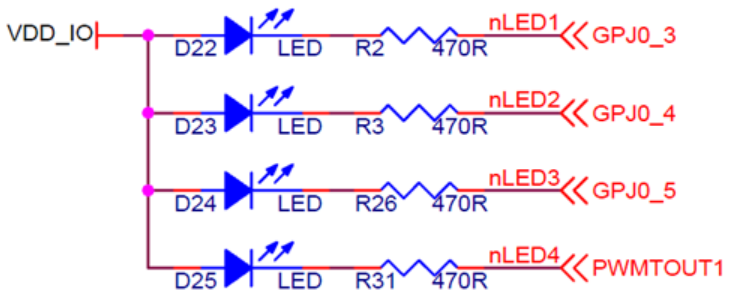
1. 流水灯的控制

流水灯的控制是单片机入门的实验之一，这是个很好的示例。在嵌入式领域里，对于计算机专业出身的人来说，这也是个很好的实验。可以更进一步了解 GPIO 的控制和应用层与驱动接口之间的联系。完成此功能可进入蜂鸣器的控制。

提示：在 LED_IOControl 接口函数中实现此功能。

2. PWM 的控制

D25(nLED4) 所接的 GPIO 是 GPH17，我们把他设置成 GPH17_Output 模式。同时这个引脚又是 PWMTOUT1 的输出口，也是就是这个引脚可以用做 PWM 输出。如果设成 PWM 输出时，此时 LED4 将会按照 PWM 的脉宽来闪烁。完成此功能，可用于创建背光驱动。背光驱动也是用 PWM 来控制亮度。



提示：此作业涉及到 PWM 控制器的配置。需熟悉 S5PV210 芯片资料。参考驱动中其他调用 PWM 的相关设置方法。

7.2 音频驱移植

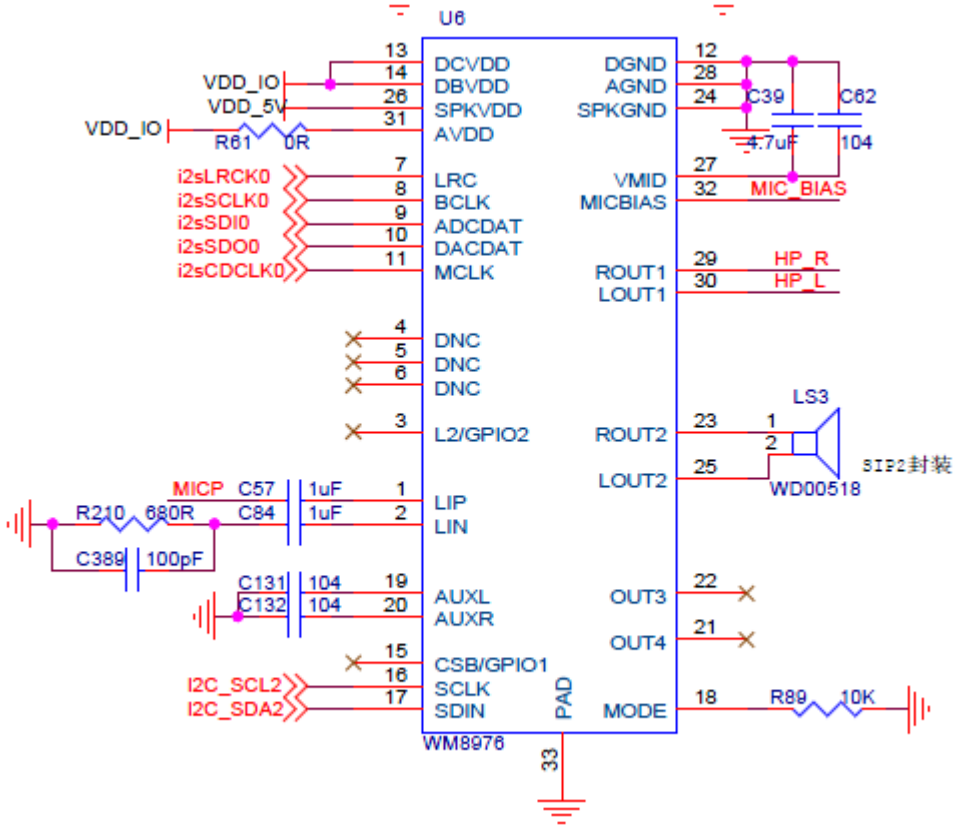
通过上一实例，我们知道简单流驱动的开发移植过程，然后接下来说下音频驱动的开发移植。这个音频驱动也是个流驱动。所以我重点讲解音频驱动的移植思路。涉及到硬件 I2S 接口，I2C 接口，GPIO 控制，MDA 控制，还有中断的控制等等。等你仔细消化了这个驱动

后。你会发现你的功力大增。

7.2.1 音频电路原理分析

原理图如下，这款 X210-II 开发板与 X210 开发板都是直接上电，没有 GPIO 控制。所以电源供电部分就无法用软件控制。两款开发板音频部分唯一区别是 I2S 数据通道做了更改。由 X210 开发板上的 I2S1 更改为 I2S0。在这里、我将讲解每个移植过程。

WM8976 音频芯片与 CPU 的接口主要关注两路。第一路是 I2C 的通讯口。用于控制 WM8976 的工作状态。对 WM8976 的初始化和配置都由这个 I2C 接口完成。另外就是 I2S 接口，音频数据流的输入和输出都是通过这个数据口来完成。



从上图中可以看出。I2C 采用的是第二路 I2C 口。对于 I2C 的协议比较简单。第一次接触 I2C 的，建议仔细看下协仪，有机会的话用 GPIO 去模拟一下 I2C 的协议，对于理解 I2C 协议和硬件的控制有很大帮助。因为 SPI 等协议控制原理都差不多，仅仅是协议内容不同而已。

言归正传，由于 S5PV210 内部已有 I2C 控制器，所以我们需要控制 I2C 的控制器。三星原厂已把 I2C 的控制器做成了流接口，所以我们在使用时，直接调用 I2C 的驱动程序即可。I2S 控制器采用 I2S0 通道，内部也有控制器，所以需要配置这一组寄存器。

7.2.2 移植思路

音频驱动是典型的流驱动.三星原厂开发板的 BSP 包有示例代码，所以根据以上 X210-II 开发板硬件原理图，所以移植从以下几方面入手。

1. 先保证调试 I2S 通道的频率正常.

I2S 音频正常工作需要以下几路信号：

IISSCLK	----------	采样时钟
IISLRCLK	----------	左右声道的时钟
IISCDCLK	----------	主时钟
IISSDO	----------	IIS 格式音频数据输出
IISSDI	----------	IIS 格式音频数据输入

S5PV210 的 IIS 通道做 master,WM8976 作为 slave.所以 IISSCLK IISLRCLK,IISCDCLK 三个时钟都由 S5PV210 产生,提供给音频 WM8976 芯片.

IISSCLK	采用 44.1Khz 的采样率.
IISLRCLK	根据 210 芯片手册采用 32fs,即 1.4112Mhz.(注:fs 就是采样频率 44.1Khz.)
IISCDCLK	根据 210 芯片手册采用 256fs ,即 11.289Mhz

经以上分析,正式开始移植, X210 开发板音频所采用的 IIS0 通道与三星原厂的 IIS0 一样,所以，不需改动。用示波器抓 IISSCLK IISLRCLK,IISCDCLK 引脚的波形,发现有时钟输出,说明 IIS 通道已有数据输出.

再测试输出频率，正常应该如下：

IISCDCLK	11.289Mhz.
IISSCLK	1.4112Mhz
IISLRCLK	44.1Khz

如果频率不对，需要查找原因。本人在调试过程中就遇到这个频率一直不正常。经仔细检查。是在屏蔽掉 I2C 对 WM8976 的代码后，频率就正常了。

2. 保证有音频数据输出

涉及 I2S 初始化的配置。

有音频数据输出时, IISLRCLK 会输出 44.1Khz 的采样频率时钟,同时 IISSDO 会在频率输出.如果没有,则查 DMA 通道是否有数据送出，通过加打印信息跟踪即可。在有了时钟和数据输出后,接下来就是控制音频芯片工作了。

3. 保证 I2C 正常通信.

I2C 的初始化和控制。

音频芯片 WM8976 的控制通信接口是采用 I2C 接口，由于 CODEC 芯片 WM8976 无法读取内置寄存器值，所以不能通过 I2C 读取 CODEC ID 来验证是否读操作是否正常。同样也不能通过读写进去的值来验证 I2C 写操作是否正常。一般是通过波器来抓波形来检测 I2C 数据是否正常发送出去。这时要注意的是要与 I2C 控制器输出的协议波形一定要与 I2C 器件（这里指的是 WM8976 音频芯片）规格书的关于 I2C 的协议部分一致，方可通信正常。

在此不得不说下 WM8976 芯片寄存器的特点。WM8976 的寄存器值有 9 位.通常我们对 I2C 器件写数据时,都是 8bit 数据.为此,在往 WM8976 芯片中写数据时,需按以下规则处理。

高 7 位寄存器地址							9b	8 位数据位							
7b	6b	5b	4b	3b	2b	1b	0b	7b	6b	5b	4b	3b	2b	1b	0b

CODEC 寄存器地址左移一位,D0 位用于存放数据寄存器的最高位 D8.函数实现如下,见蓝色字体部分:

```
Void HardwareContext::WriteCodecRegister(WORD wReg, WORD wVal)
```

```
{
    BYTE btWriteBuffer[2];
    btWriteBuffer[0] = (BYTE)(wReg<<1)|(BYTE)((wVal>>8)&1);
    btWriteBuffer[1] = (BYTE)(wVal&0xFF);
    WriteI2C_CODEC(m_hI2CCodec, btWriteBuffer, 2);
    DBGMSG(WAVE_CODEC, (L"[WAV] WRITE ADDR : 0x%04X, DATA : 0x%04X\r\n", wReg, wVal));
}
```

在 WM8976 寄存器数值的数组如下

```
// WM8976 Codec-chip Initialization Value
unsigned int WM8580_Codec_Init_Table[][2] =\
{
    {0x00,0x00},//R0 software reset
    {0x03,0xed},//R1 power management 1--PLL 0ff
    {0x05,0x80},//R2 power management 2
.....
};
```

数组 WM8580_Codec_Init_Tabl[][2]中的寄存器地址和数据已经做了移位处理，WriteCodecRegister()再进行移位处理,造成了数据全部错位.所以解决办法是只保留一次移位操作.即可.改之.声音出来了。

4. 修改 CODEC WM8976 相关控制代码.

WM8976 芯片的内部寄存器的配置。参考源码。这是 WM8976 厂家提供的初始化代码。需要改写时，请参考 WM8976 的数据手册。

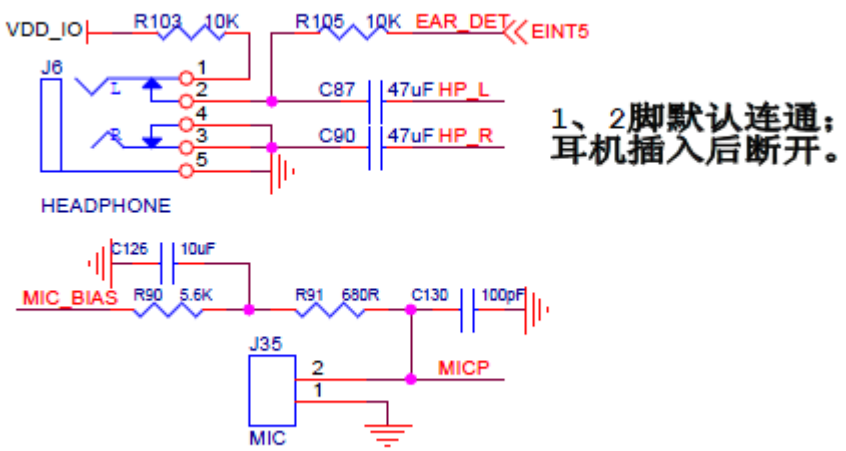
7.2.3 音频驱动作业

现有驱动有几个功能没有调试，当成作业吧。完成后大家可以提交到 bbs.9tripod.com 中。

1. 音频耳机侦测功能

目前现在的驱动代码是把音频数据同时输送到喇叭和耳机。

原理图如下：



ENIT15 作为耳机插入和拔出的检测中断脚。当没有耳机插入时，EINT5 是高电平，当有耳机插入时，EINT5 引脚电平会变成低电平。此时中断产生，线程切换音频流输出到耳机。注意要防抖处理。反之，同理。

提示：开启一线程用于侦探耳机插入和拔出事件。硬件上以中断方式，通知线程，由线程通过 I2C 接口，改写 WM8976 的寄存器值来完成通道的切换。

2. 录音功能的实现

此驱动还有一个录音功能需实现。上层应用发录音消息，用录音测试程序即可，驱动中开启 WM8976 的录音通道，把数据回传给上层。硬件上他是从 IISSDI 引脚送入。

录音效果好坏与硬件电路有很大关系，所以调试时就耐心细致的调节。

第8章 其他产品介绍

8.1 核心板系列

X6410CV10
X210CV3
X210CV4
G210CV10
I210CV20
X4412CV2
X4418CV2
X6818CV3
X3288CV3

8.2 开发板系列

x6410 开发板
x210 开发板
g210 开发板
i210 开发板
x4412 开发板
x4418 开发板
X6818 开发板
X3288 开发板
ibox4412 卡片电脑
ibox4418 卡片电脑
ibox6818 卡片电脑

说明：产品详细规格，以及更多其他产品请关注九鼎创展官方网站和论坛。