

ARM9 GEC2410v1.1 用户手册

广东省嵌入式软件公共技术中心

<http://www.gd-emb.org/>

广东省嵌入式软件公共技术中心简介 (美的集团美泰科技有限公司)

广东省嵌入式软件公共技术中心(简称“GEC”)是一个面向信息家电、汽车电子、数控机床、消费电子、网络通讯等嵌入式新技术、新产品领域的技术与产业中心。

目前, GEC 已形成嵌入式技术与产品解决方案、嵌入式系统评测认证、技术培训服务和电子商务平台四大业务体系, 以“专业诚信、开拓创新、行业引领、齐商共赢”的中心精神为指导, 努力为客户的技术创新和产品升级提供支持。

在开发板系列产品方面, GEC 目前已经形成了完善的产品体系结构, 包括以下四个方面:

1 多样化的开发板产品: 目前 GEC 开发板系列产品包括 2410 开发板、2440 开发板、实验箱等一系列产品, 目标是向有志于从事嵌入式学习及开发的高校、学生、工程师以及企业客户提供完善的嵌入式开发平台的整体解决方案。

2 严谨的产品评测过程: 本中心设立了专门的嵌入式产品评测团队, 拥有完善的评测设备, 保证每个出厂产品都经过严谨的评测过程, 从而保证了产品的质量。

3 完善的售后服务: 开发板产品出现质量问题 10 天包换, 12 个月免费保修。保修时间之长堪称国内之最。

4 强大的技术支持: 本中心拥有 30 多人的技术研发团队, 拥有强大的技术研发力量, 技术领域涉及到嵌入式开发的各个方面, 为开发板客户提供全面的技术支持。

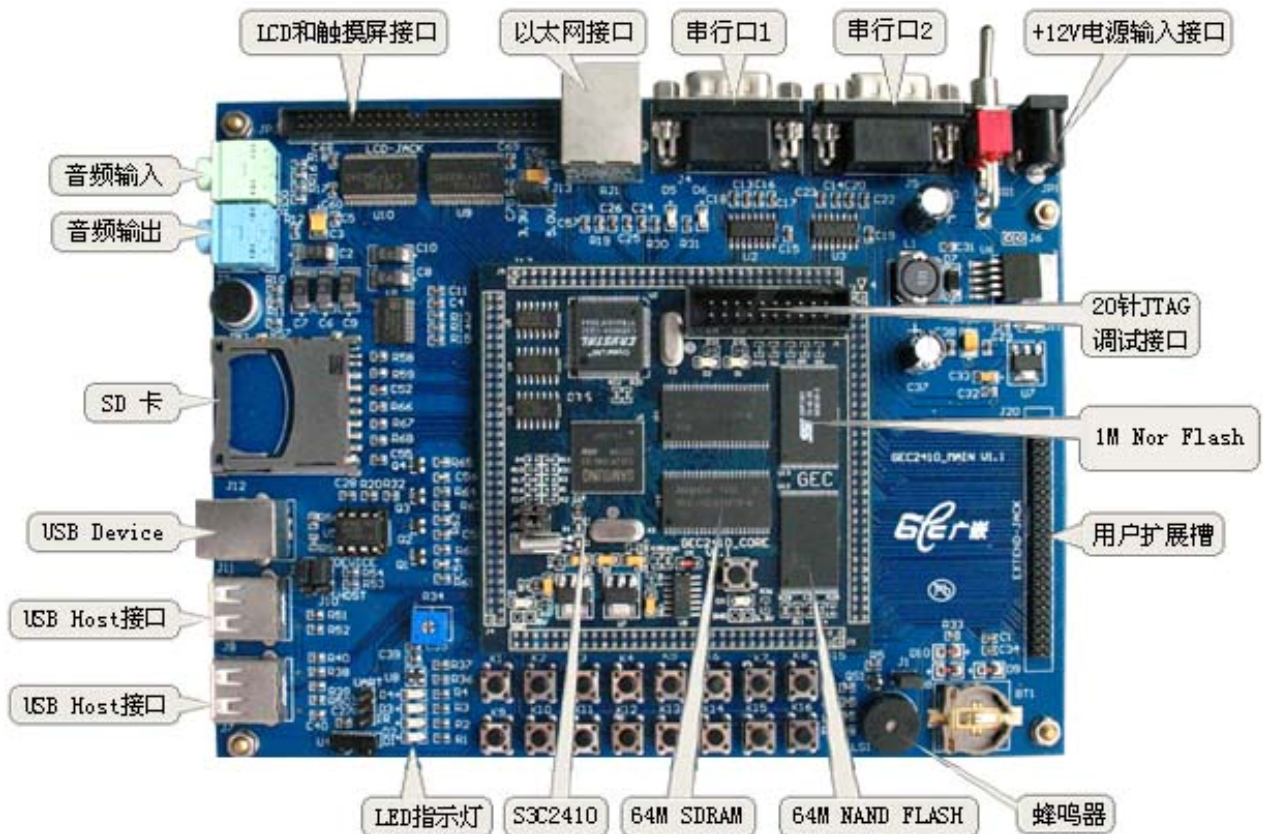
目录

第一章 GEC2410 开发板套件介绍	5
1.1 GEC2410 开发板外观	5
1.2 GEC2410 开发板硬件构成	5
1.3 硬件资源分配	6
1.4 开发套件包含的内容	8
1.5 操作系统支持的驱动	9
第二章 GEC2410 开发板使用	10
2.1 GEC2410 开发板快速使用入门	10
2.2 启动 LINUX 和 WINCE	13
2.3 GEC2410 的 BIOS 功能说明	17
2.4 非操作系统下的外围资源测试	19
2.5 Linux 操作系统下的外围资源测试	30
2.6 用 SJF2410 工具将 BIOS 烧写到 NAND FLASH	32
第三章 烧写和启动 linux	37
3.1 烧写 LINUX 内核	37
3.2 烧写根文件系统	40
3.3 启动 LINUX	42
第四章 烧写 WINCE 和启动 WINCE	47
4.1 下载运行 WINCE	47
4.2 烧写 WINCE	49
4.3 自启动 WINCE	50
第五章 用 ARM 仿真器调试 GEC2410	53
第六章 嵌入式 linux	60
6.1 安装工具链	60
6.2 配置移植 linux 内核	60
6.3 制造根文件系统	63
第七章 嵌入式 GUI--Qt 开发	65
7.1 Qt/Embedded 开发环境的安装	65
7.2 Qt/Embedded 开发工具的安装	65
7.2.1 tmake 的安装	65
7.2.2 Qt X11 的安装	65
7.2.3 Qt/Embedded 安装	68
7.2.4 交叉编译 qtopia	68
7.3 Qt/Embedded 应用开发实例	69
7.3.1 在 ARM 上的 qt/embedded 应用程序实例	69
7.3.2 在 ARM 上的 qt/embedded 应用程序实例	72
第八章 嵌入式 WINCE	74
8.1 WINCE 的安装	74
8.1.1 安装开发环境	74
8.1.2 安装目录	74

8.1.3 构建新的平台、编译	78
8.2 WINCE 的 image 的运行	90
8.3 通过以太网烧写 WINCE 映象文件	90
8.4 在 WINCE 和桌面系统之间建立通讯连接.....	100

第一章 GEC2410 开发板套件介绍

1.1 GEC2410 开发板外观



1.2 GEC2410 开发板硬件构成

硬件功能介绍:

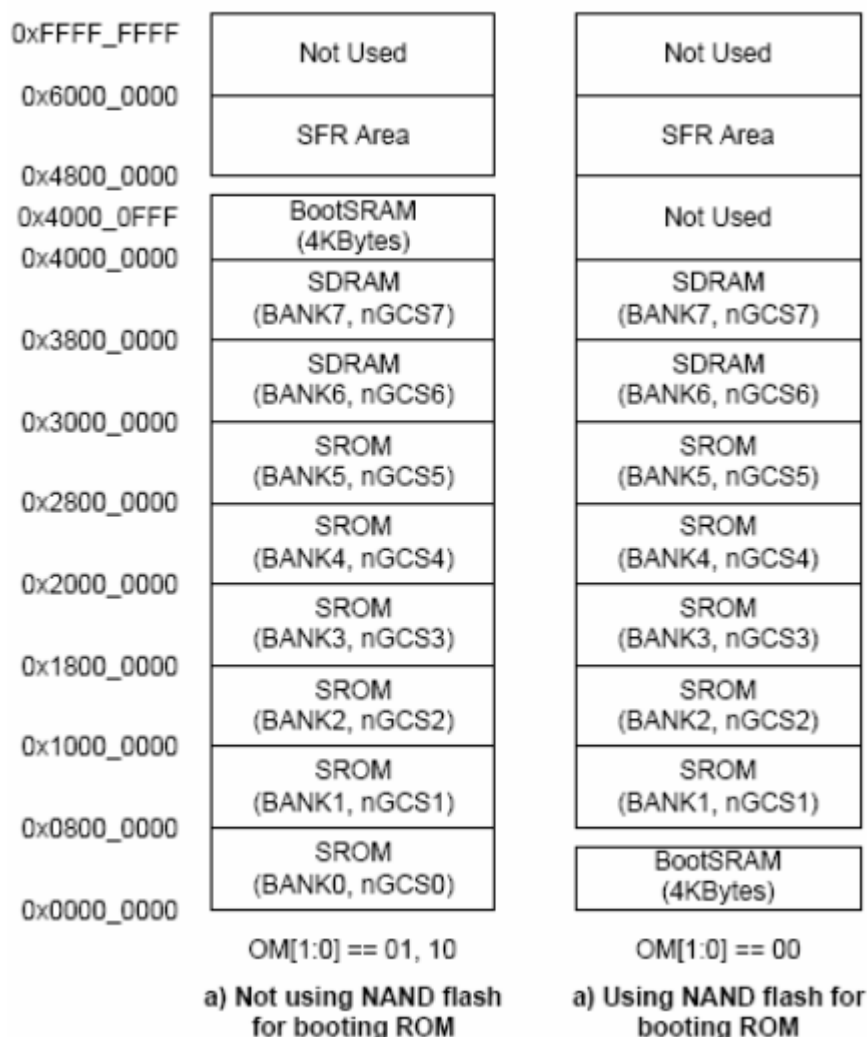
- ◆ 采用三星公司的 S3C2410，主频可达 203MHz；
- ◆ 64M 字节的 SDRAM，由两片 K4S561632 组成，工作在 32 位模式下；
- ◆ 64M 字节 NAND Flash，采用的是 K9F1208，可以兼容 16M，32M 或 128M 字节；1M 字节 NOR Flash；
- ◆ 10M 以太网接口，采用的是 CS8900Q3，带传输和连接指示灯；
- ◆ LCD 和触摸屏接口；
- ◆ 2 个 USB HOST，S3C2410 内置的，符合 USB 1.1，注意这两个 USB HOST 接口是复用的
- ◆ 一个 USB Device，S3C2410 内置的，符合 USB 1.1；
- ◆ 支持音频输入和音频输出，音频模块由 S3C2410 的 IIS 音频总线接口和 UDA1341 音频编解码器组成，板上还集成了一个 MIC，用于音频输入；
- ◆ 2 路 UART 串行口，波特率可高达 115200bps，并具有 RS232 电平转换电路；
- ◆ SD 卡接口，兼容 SD Memory Card Protocol 1.0 和 SDIO Card Protocol 1.0；
- ◆ Embedded-ICE（20 脚标准 JTAG）接口和并口式 JTAG 接口，支持 ADS，SDT 软件的下载和调试以及 FLASH 的烧写；
- ◆ 串行 EEPROM：AT24C02 4Kbytes EEPROM，IIC 接口；
- ◆ 蜂鸣器，4 个 LED 灯；

- ◆ 16 个按键
- ◆ 开关电源，分布式电源供电，3V 锂电池，提供 CPU 内置 RTC 操持电源；

1.3 硬件资源分配

1.3.1 地址空间分配以及片选信号定义

S3C2410 支持两种启动模式：一种是从 NAND FLASH 启动；一种是从外部 nGCS0 片选的 Nor Flash 启动（gec2410v1.00 开发板没有 Nor Flash）。在这两种启动模式下，各片选的存储空间分配是不同的，这两种启动模式的存储分配图如下：



a) 图是 nGCS0 片选的 Nor Flash 启动模式下的存储分配图；

b) 图是 NAND FLASH 启动模式下的存储分配图；

注意：gec2410v1.0 版本的开发板没有 nor flash

说明：SFR Area 为特殊寄存器地址空间

◆ 下面是器件地址空间分配和其片选定义

在进行器件地址说明之前，有一个点需要注意，nGCS0 片选的空间在不同的启动模式下，映射的器件是不一样的。由上图可以知道：

—在 NAND FLASH 启动模式下，内部的 4K Bytes BootSRam 被映射到 nGCS0 片选的空间。

—在 Nor Flash 启动模式(非 NAND FLASH 启动模式)下，与 nGCS0 相连的外部存储器 Nor Flash 就被映射到 nGCS0 片选的空间

地址范围	说明	片选信号
0x4800 0000~0x6000 0000	SFR(特殊寄存器)地址空间	
0x4000 0000~0x4000 0FFF	NAND FLASH 启动模式下该空间没有被使用	
	非 NAND FLASH 启动模式下，该空间为 BootSRam	
0x3000 0000~0x3400 0000	SDRAM 空间	nGCS6
0x19000300	CS8900 的 IO 基址	nGCS3
0x0000 0000~0x0020 0000	NAND FLASH 启动模式下，BootSRam 的有效地址： 0x0000 0000~0x0000 0FFF	nGCS0
	非 NAND FLASH 启动模式下，(Flash SST39VF1601) Nor flash 的有效地址空间为： 0x0000 0000~0x0020 0000	

1.3.2 接口资源及按键说明

◆ 跳线分配表

跳线名称	说明
JP1 (核心板)	决定 S3C2410 的启动模式 插上短路块从 Nand Flash 中启动，默认 不插上短路块从 Nor Flash (SST39VF1601) 中启动 (注意：GEC2410V1.1 开发板没有 nor flash)
J1	使能蜂鸣器 接上短路帽：使能蜂鸣器；默认 否则就断开蜂鸣器
J18	短路块决定 RXD2 信号是 UART2 功能还是 IrDA 功能 短路帽接在 RXD2 的标识处 (1, 2) 决定工作在 UART 功能，默认 短路帽接在 nCTS1 的标识处 (2, 3) 决定工作在 IrDA 功能
J9, J10	短路帽接在 1, 2 处选择工作在 USB HOST1 处 短路帽接在 2, 3 处选择工作在 USB DEVICE 方式，默认
J13	短路帽接在 1, 2 脚，LCD 工作在 5V，默认 短路帽接在 2, 3 脚，LCD 工作在 3.3V

◆ 按键说明

下表为核心板按键说明

按键名称	说明
S1 (Reset) (核心板)	复位按键，小按键

K1~K16 按键	4x4 键盘
-----------	--------

◆LED 指示灯说明

下表是核心板的 LED 指示灯说明

LED 名称	说明
D1, D2	I/O 口指示灯
D3	PWREN 线上的 LED
D4	核心板电源指示灯

下表为底板 LED 指示灯说明

LED 指示灯名称	说明
D5, D6	网络指示灯
D1~D4	4 个 I/O 口指示灯
D11	底板电源指示灯

◆各种接口说明

下表为核心板上的接口说明

接口名称	说明
J1 (JTAG)	20 针扩展口, 标准的 JTAG 接口

下表为底板的接口说明

接口名称	说明
RJ1 (RJ45)	以太网接口 (RJ45, 带隔离器的)
J4, J5	串行口 1, 2
U4	红外线 IrDA
J12	SD 卡接口
J7	USB HOST 0 接口
J9	USB HOST 1 接口 (与 USB DEVICE 复用)
J11	USB DEVICE 接口
J3	音频输出接口
J2	音频输入接口
U6	CCFL 背光电源输出接口
JP1	板子的开关电源接口
J20	一些扩展口
JP3 (LCD INTERFACE)	LCD 和触摸屏接口

1.4 开发套件包含的内容

GEC2410 光盘上提供的开发工具和源代码:

- 1) 测试源代码, 启动引导源代码 BIOS 和下载烧写工具 sjf2410.exe;
- 3) Linux for 2410 的内核源码包以及内核交叉编译工具, 可运行的根文件系统及根文件系统制作工具 mkcramfs;
- 4) qt-embedded-2.3.7 和 qtopia-free-1.7.0 源码, 应用程序交叉编译器。
- 5) WINCE4.2.NET 板级支持包 BSP for 2410, 及已编译好的 WINCE 镜像文件;

- 6) GEC2410 及部分板上扩展芯片的资料;
- 7) GEC2410 底板和核心板电路原理图 (pdf 格式);
- 8) GEC2410 开发板用户手册 (pdf 格式);
- 9) 以及一些实用工具, 如 DNW 。

套件包括:

- 1) 一块已测试好的 GEC2410 开发板 (包括核心板与底板)
- 2) GEC2410 用户光盘
- 3) 3.5" TFT 彩色 LCD 板一块, 带触摸屏 (选配的)
- 4) 一条交叉串口线 (两边都是母头)
- 5) 一条网线 (交叉网线)
- 6) 一条 USB 线
- 7) 触摸笔一支 (与 LCD 屏一起选配的)
- 8) 一个 +12V/1A 直流电源
- 9) 一个包装盒

1.5 操作系统支持的驱动

GEC2410 开发板支持 WINCE 和 Linux 嵌入操作系统。

嵌入 Linux 操作系统说明:

- 采用 linux2.4 以上的内核
- 支持多种文件系统, 象 Cramfs, FAT 以及用于 NAND FLASH 的 YAFFS 文件系统等等
- 支持 LCD 和触摸屏
- 支持 USB HOST
- 支持 QT
- 支持 MP3 播放和视频播放
- 支持多种网络应用, 象 FTP, HTTP, Telnet 之类的网络应用

WINCE 嵌入式操作系统说明:

- 支持 LCD 和触摸屏
- 支持 USB HOST
- 支持音频
- 支持串口
- 支持网络
- SD 卡

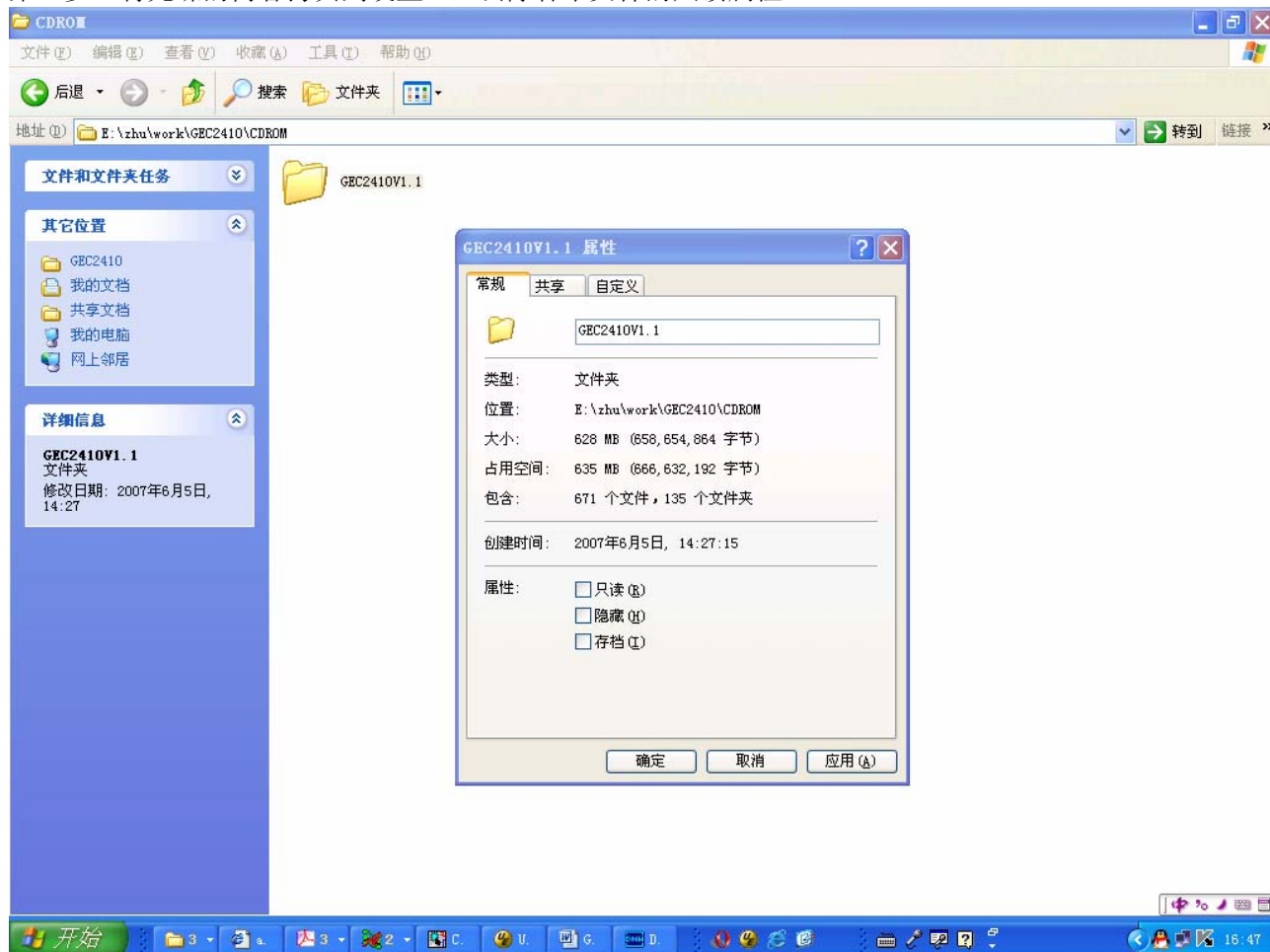
第二章 GEC2410 开发板使用

2.1 GEC2410 开发板快速使用入门

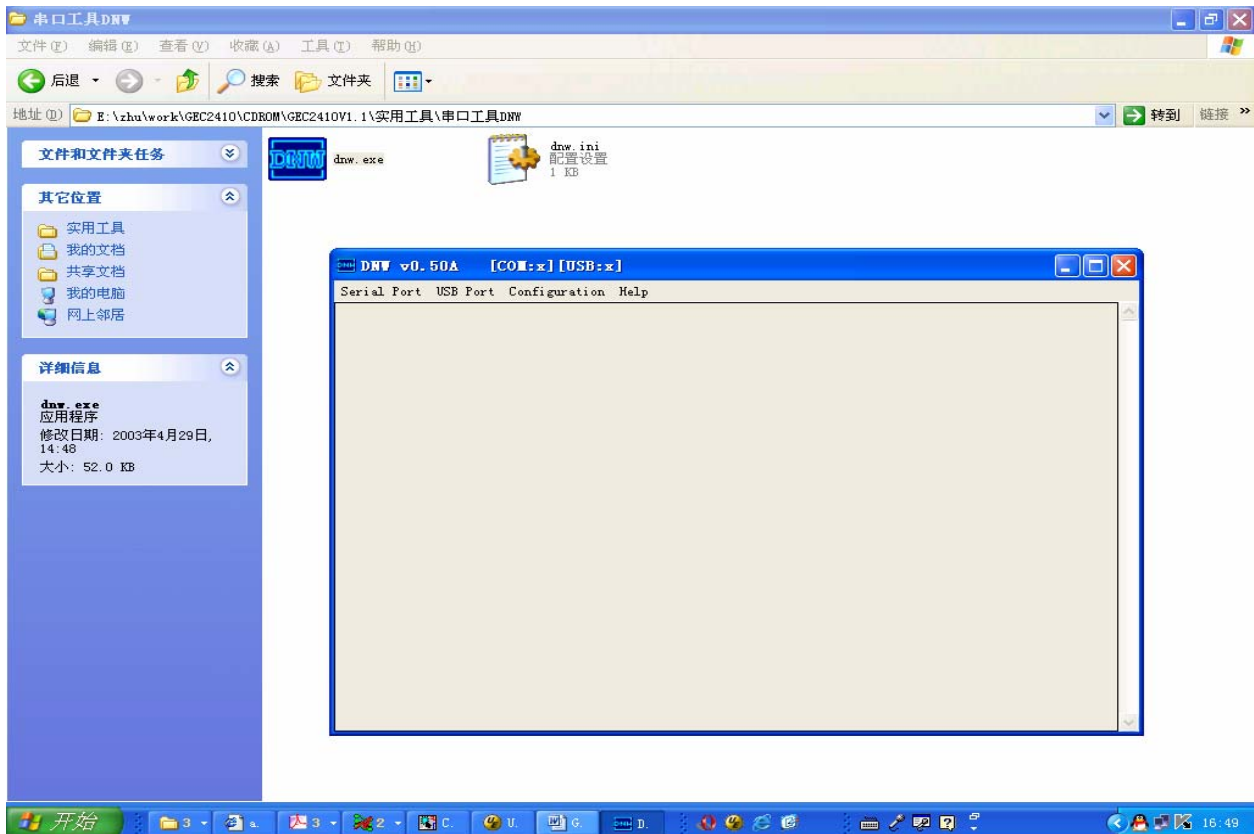
2.1.1 串口连接

第一步：连接 PC 机与开发板，GEC2410 开发板接上电源，接上交叉串口线（**串口线连接到开发板的 J4**），连上 USB 线（**USB 线连接到开发板的 J11**），连上交叉网线。

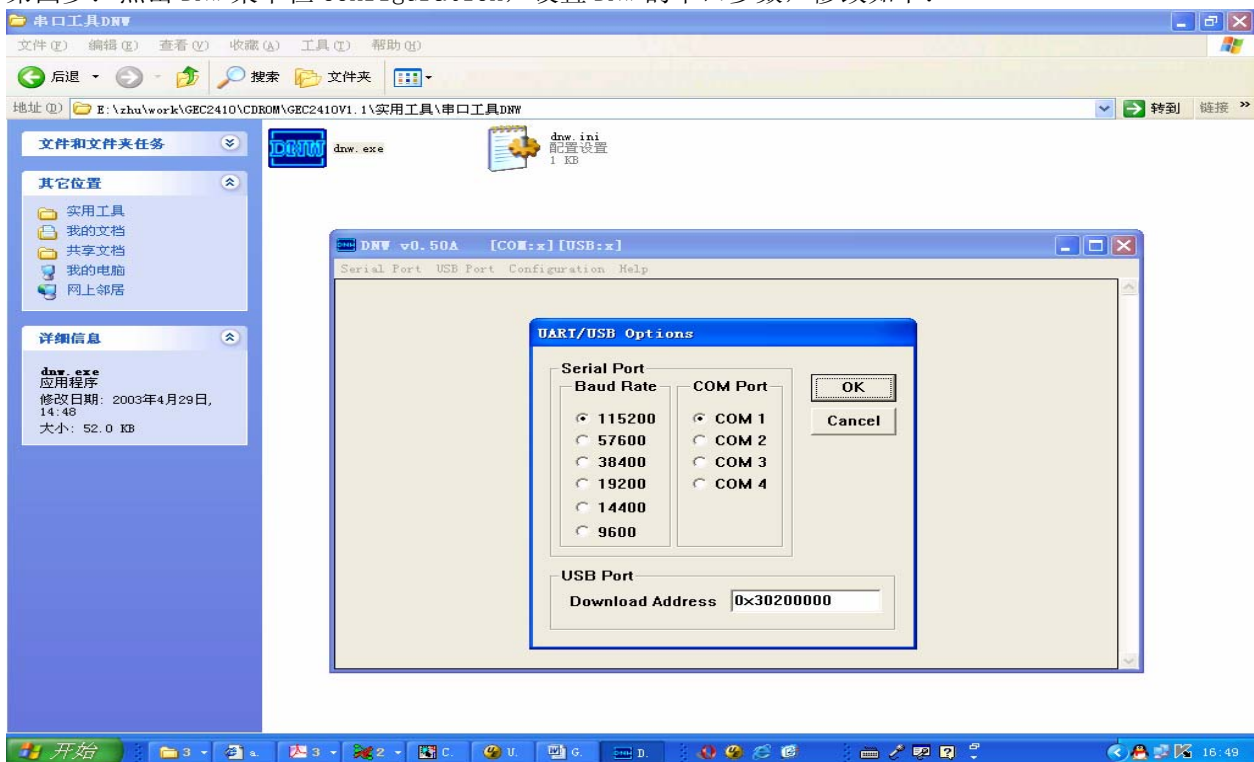
第二步：将光碟的内容拷贝到硬盘上，去除各个文件的只读属性。



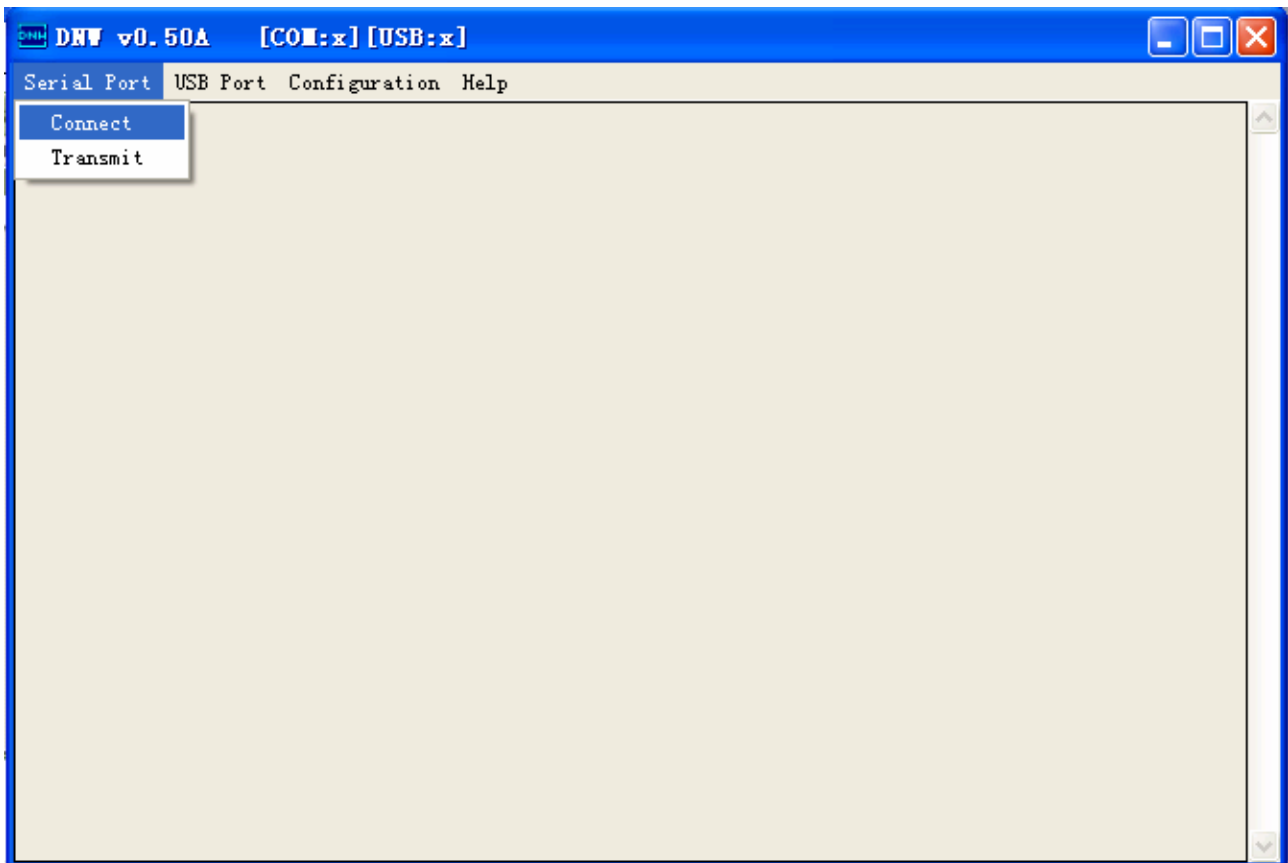
第三步：打开 gec2410v1.1\实用工具\串口工具 DNW 目录下的 dnw.exe, 如下：



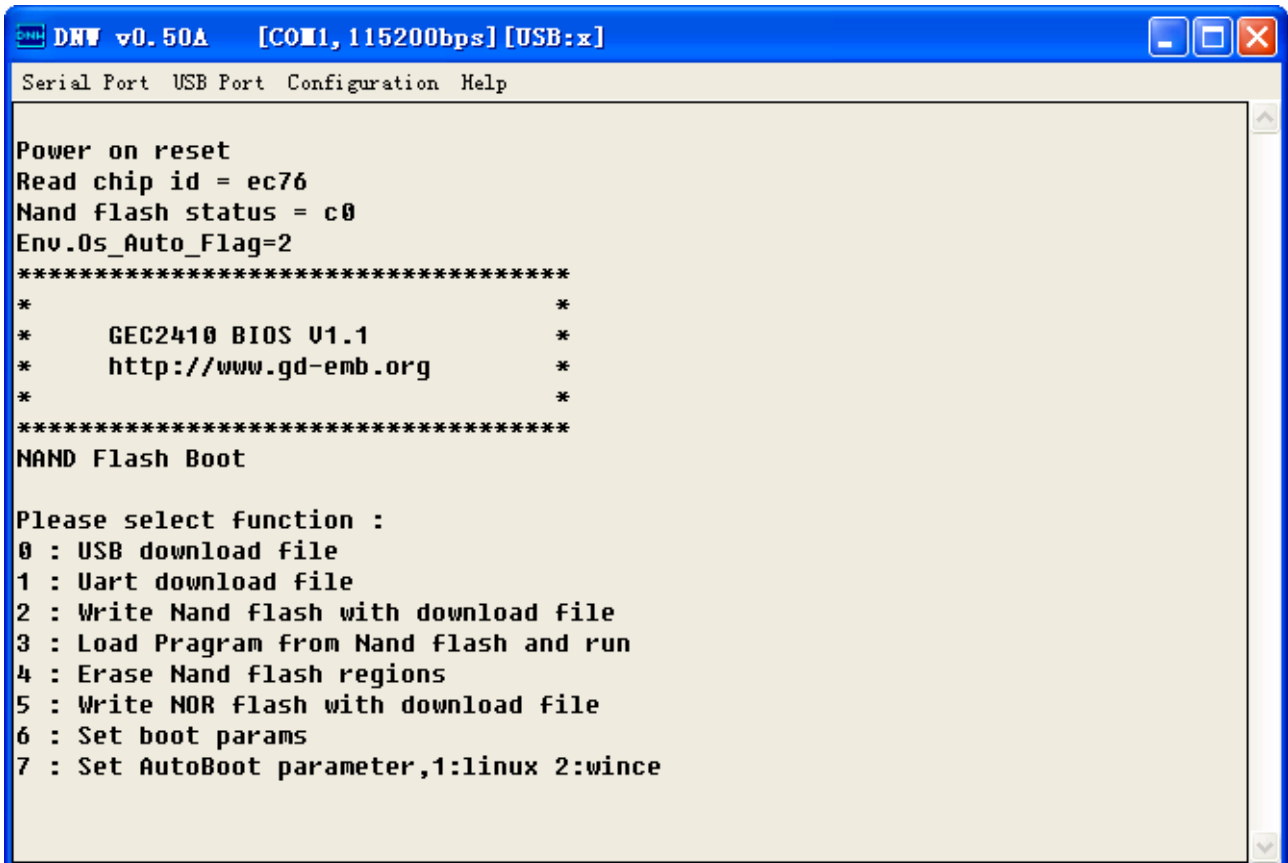
第四步：点击 DNW 菜单栏 Configuration，设置 DNW 的串口参数，修改如下：



第五步：点击 DNW 菜单栏 Serial Part 下的 Connect，连接串口，如下图：



第六步：打开开发板上的电源，就可以在 DNW 上看到启动信息：



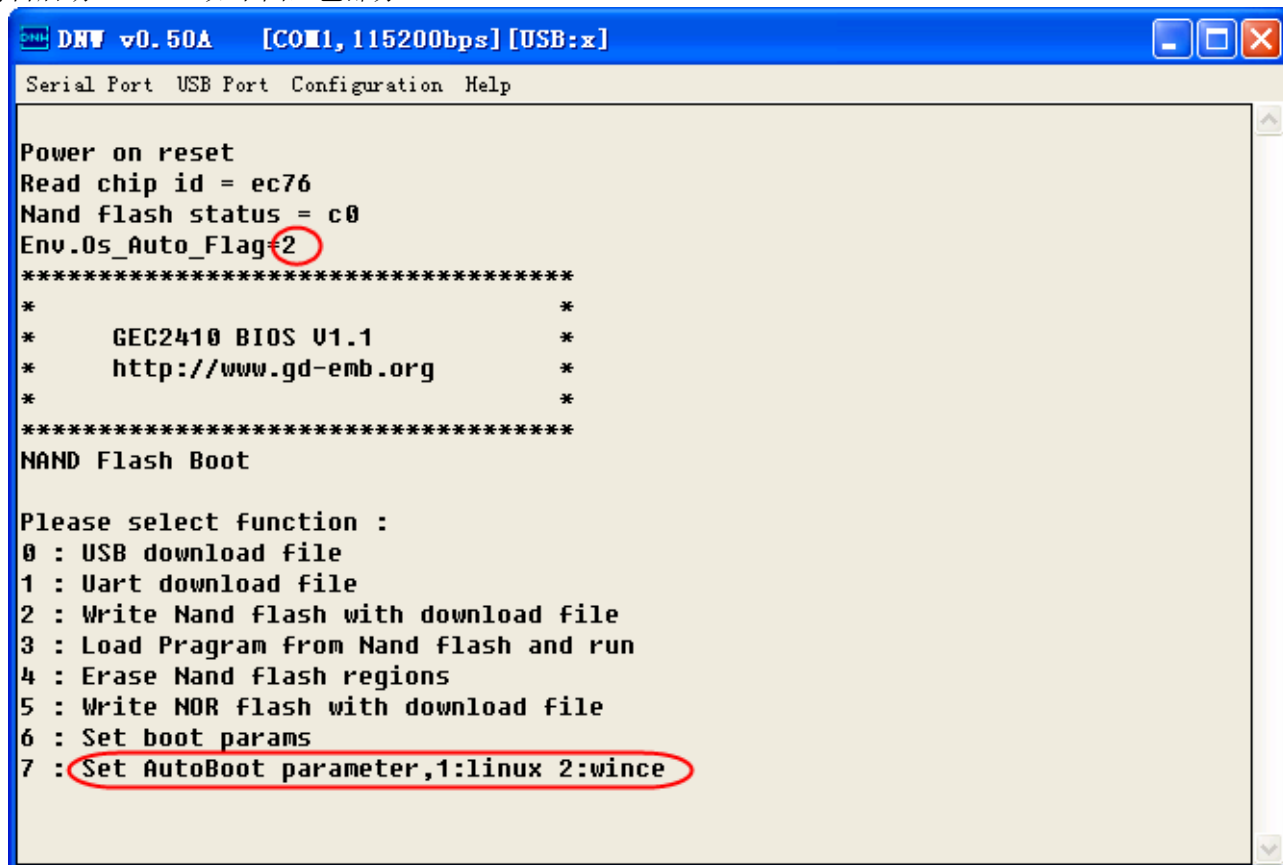
2.1.2 USB 驱动安装

请参考“配置的多媒体演示”目录下的“2 PC 端 USB 驱动安装”文件。

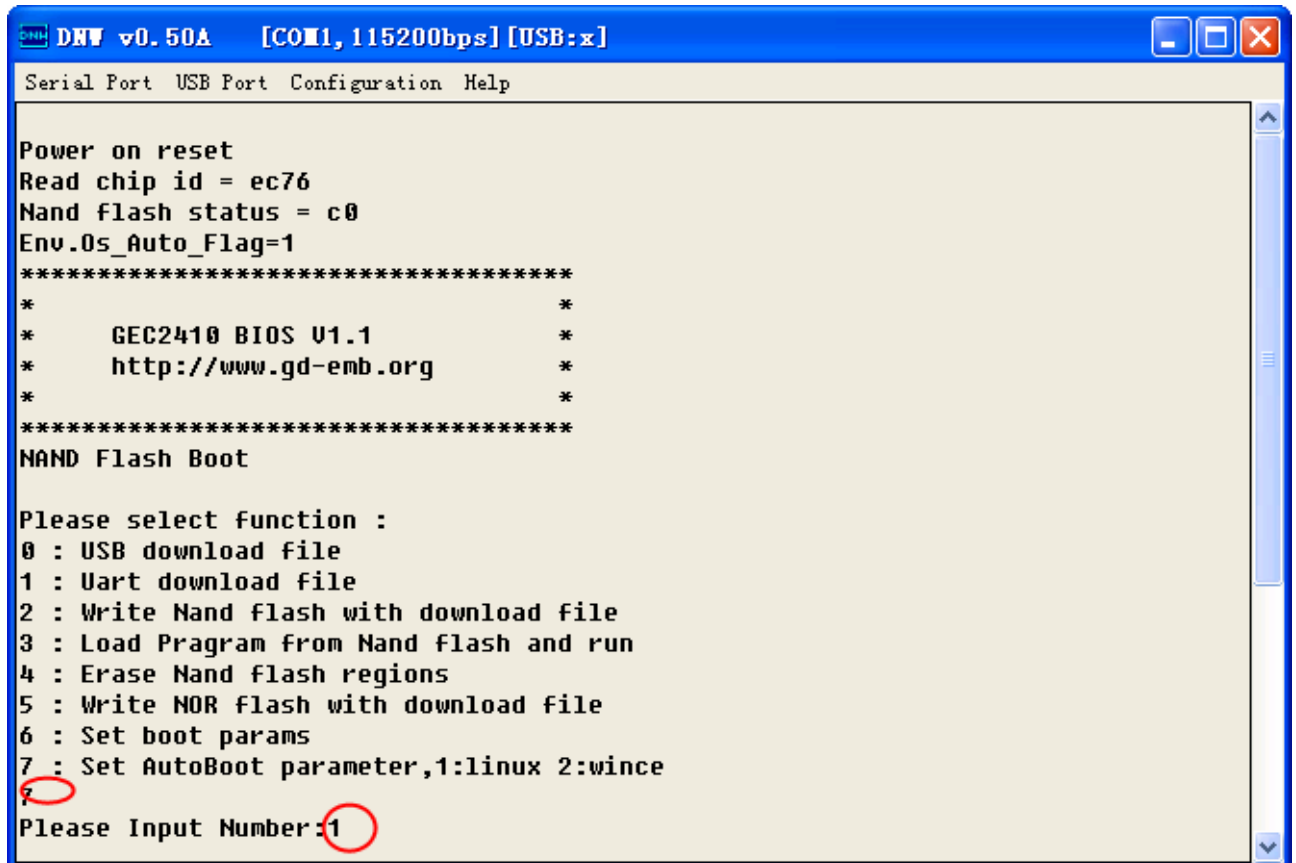
2.2 启动 LINUX 和 WINCE

2.2.1 启动 LINUX

Bootloader 的 7 号功能键是选择自动启动操作系统的功能键，1：自启动 Linux，2：自启动 wince；Env.0s_Auto_Flag 是标识自启参数，Env.0s_Auto_Flag=1 说明将自启动 Linux，Env.0s_Auto_Flag=2 说明将自启动 Wince。如下图红色部分：



在开发板上电后，在 5 秒钟之内选择 7，然后再输入 1，就设定开发板的自启系统是 linux，在按开发板上的 S1 复位键，在 5 秒钟内不作任何改动，就会自动启动 linux 系统，如下图：



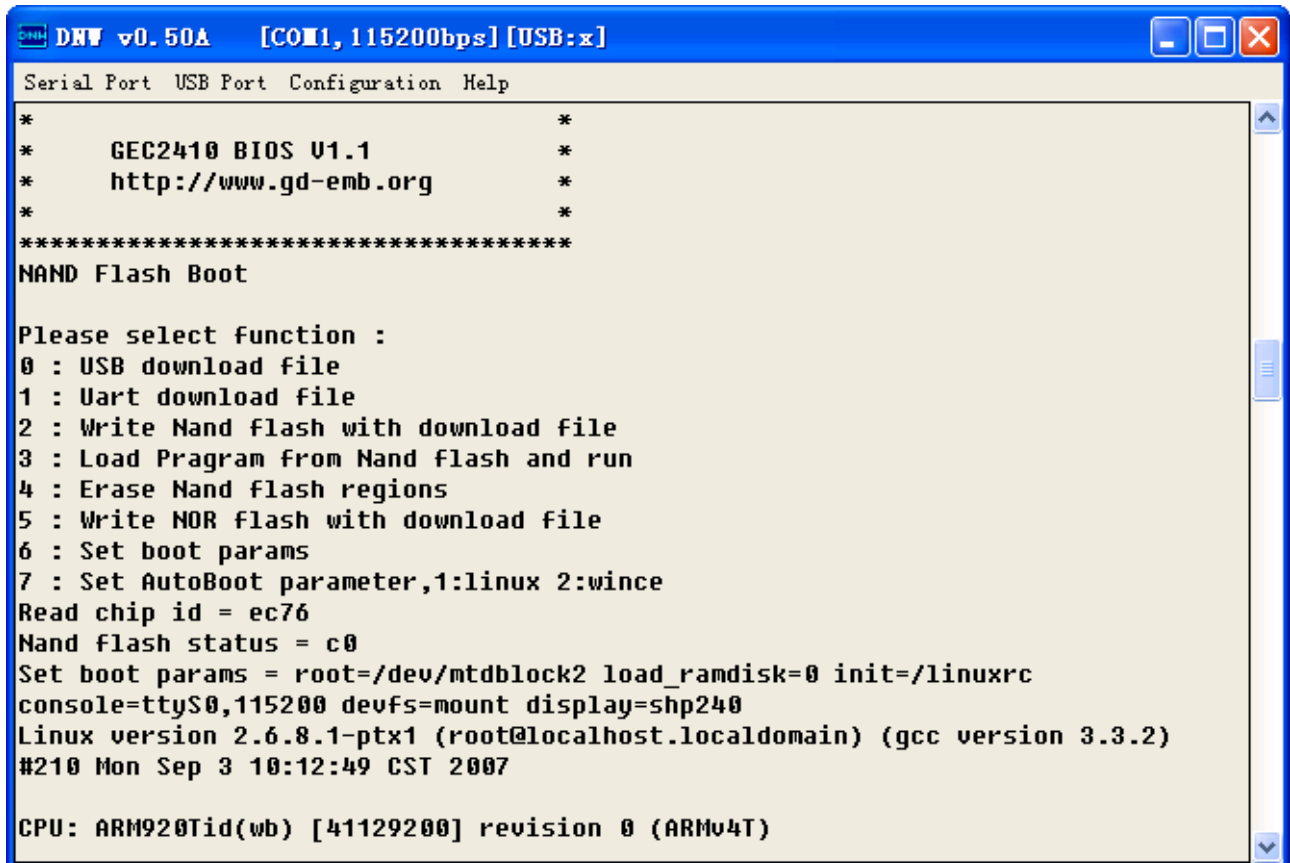
```
DNW v0.50A [COM1, 115200bps] [USB:x]
Serial Port  USB Port  Configuration  Help

Power on reset
Read chip id = ec76
Nand flash status = c0
Env.Os_Auto_Flag=1
*****
*                                     *
*      GEC2410 BIOS V1.1             *
*      http://www.gd-emb.org         *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
1
Please Input Number:1
```

这时将在串口和 LCD 屏（选配模块）有信息显示，在串口，按 Enter 键，将进入 Linux 的命令模式下，在 LCD 屏上，将有 QT 界面的显示，要进入 QT，先要进行触摸屏的校准后，才能进入 QT 的界面。

下面是 Linux 启动的串口打印信息：



```
DWV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
*
* GEC2410 BIOS V1.1
* http://www.gd-emb.org
*
*****
NAND Flash Boot

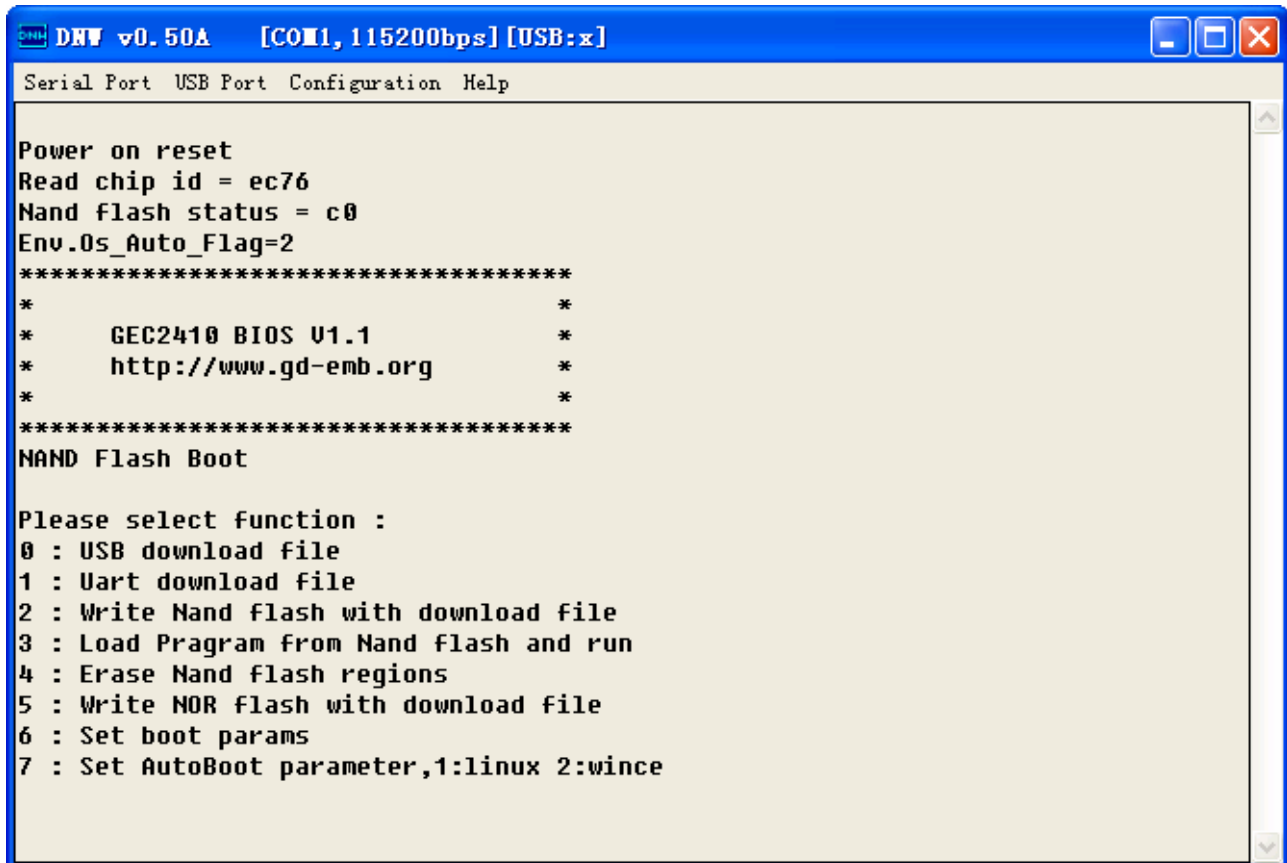
Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
Read chip id = ec76
Nand flash status = c0
Set boot params = root=/dev/mtdblock2 load_ramdisk=0 init=/linuxrc
console=ttyS0,115200 devfs=mount display=shp240
Linux version 2.6.8.1-ptx1 (root@localhost.localdomain) (gcc version 3.3.2)
#210 Mon Sep 3 10:12:49 CST 2007

CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
```

2.2.2 启动 WINCE

要设置自启动 WINCE，需要进行一些操作，准备工作与《启动 Linux》章节一样。

(1) 启动 BIOS，进入 BIOS 的主功能菜单。

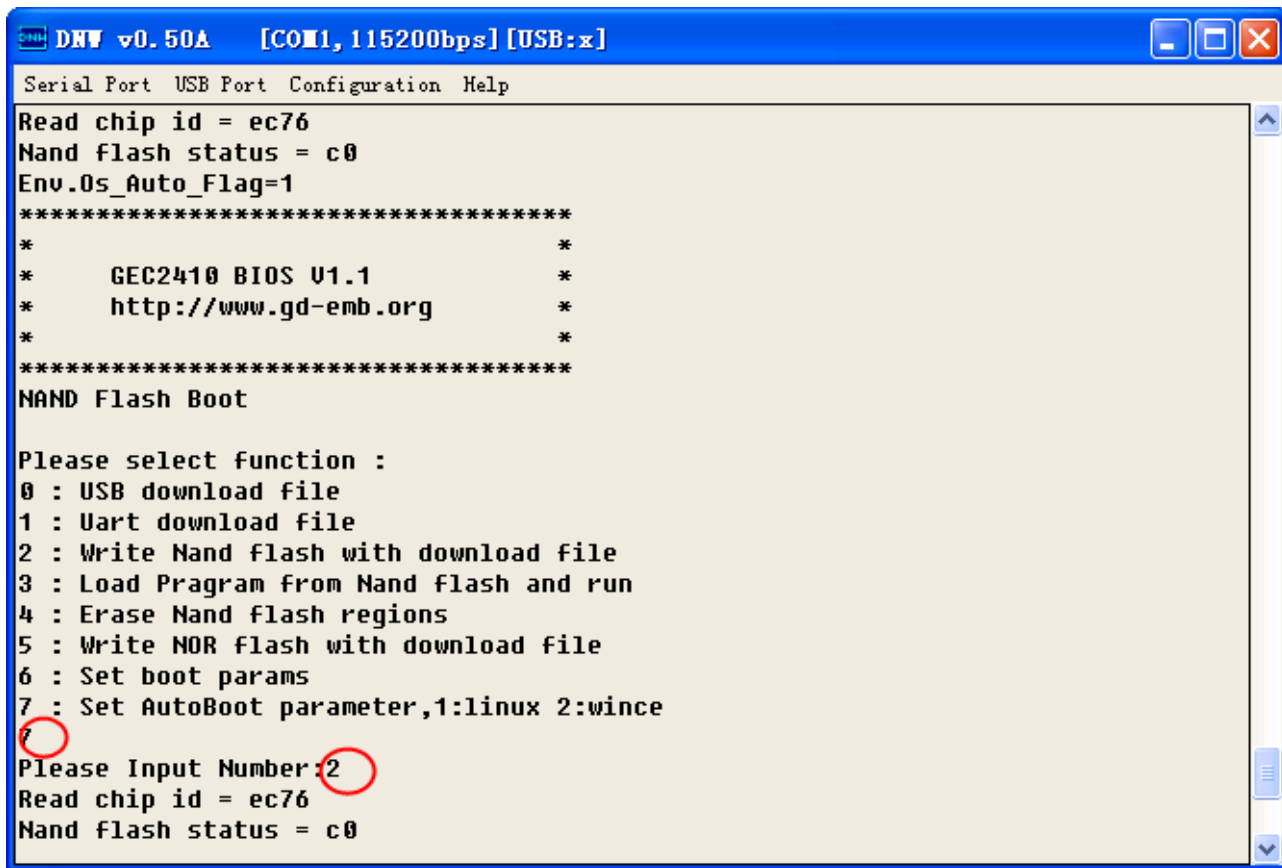


```
DNV v0.50A [COM1, 115200bps] [USB:z]
Serial Port  USB Port  Configuration  Help

Power on reset
Read chip id = ec76
Nand flash status = c0
Env.Os_Auto_Flag=2
*****
*                                     *
*   GEC2410 BIOS V1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter, 1:linux 2:wince
```

(2) 选择“7”，在“Please Input Number::”提示下输入“2”。



这样就将自启动参数保存了。

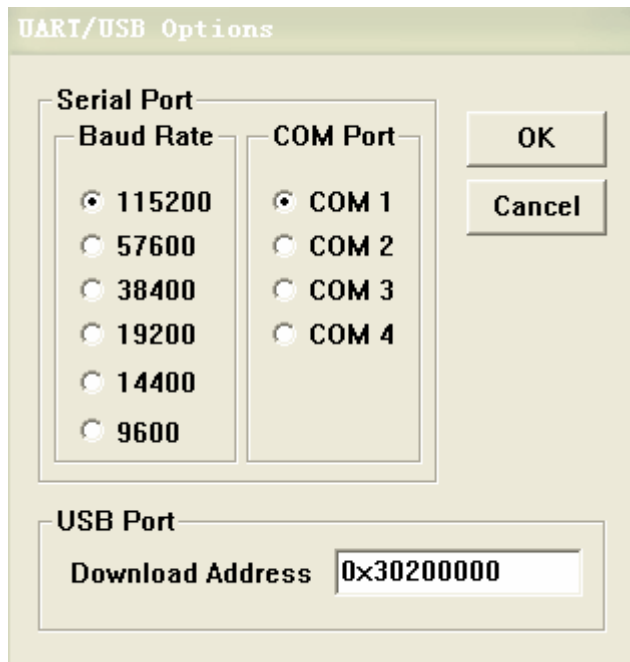
- (3) 上电复位，在 5S 内，不进行任何操作，将自动启动 Wince。
注意，WINCE 的装载要一段时间。

注意：Env.Os_Auto_Flag=1 说明将自启动 Linux
Env.Os_Auto_Flag=2 说明将自启动 Wince
Env.Os_Auto_Flag=其他，说明没有设置自启动操作系统

2.3 GEC2410 的 BIOS 功能说明

GEC2410 开发板所带的 BIOS 主要功能有文件下载（USB 和串口）FLASH 烧写（NOR 和 NAND）、启动存储在 FLASH 中的程序、设置启动参数（针对 linux），设置 Wince 或 Linux 的自启动 BIOS 菜单的 0 和 1 号功能分别是 USB 和串口下载，都是配合 DNW 这个程序使用的。

在 DNW 上的 Configuration 菜单的 Options 选项里可以选择 PC 和开发板通信的串口和波特率，还可以指定 USB 下载时开发板要将下载的文件保存到 SDRAM 中的起始地址：



对于 GEC2410 开发板，SDRAM 的物理起始地址是 0x30000000，结束地址是 0x34000000，大小是 64Mbytes，所以指定的 USB 下载地址要在此范围内。另外 BIOS 自身占用了 0x30100000 的一段空间，因此在使用 BIOS 的 USB 下载功能时应指定地址大于等于 0x30100000。用串口下载时，BIOS 内部固定了下载地址为 0x30200000。BIOS 在下载结束后，会询问是否执行下载的程序，利用此功能可引导 linux 和 WINCE 及用户自编的应用程序。

注意用 USB 下载要先在 PC 端装好驱动程序，保证 USB 连接好，第一次选择 bios 的 0 号功能，在 PC 端会提示安装驱动程序，沿着提示安装光碟下的 USB 驱动程序就可以了，有时 PC 端提示无法识别 USB 设备时，可在 BIOS 中输入 ESC 取消下载，等几秒钟再输入 0 启动 USB 下载（PC 机的操作系统最好时 windows 2000，这样 USB 下载功能会稳定很多）。

BIOS 的 2 号功能可执行 NAND FLASH 的烧写，目前支持三星的 32M 和 64M 两种型号。BIOS 对 NAND FLASH 作了简单分区，分区 0 为 BOOT 区，存储 bootloader，如 BIOS，分区 1 用做存储内核，分区 2 用作存储根文件系统。BIOS 在启动后，若等待 5S 钟没接收到控制串口的数据，会自动将存储在分区 1 的程序读到 0x30200000 地址开始的 SDRAM 中并运行。用户可在下载完数据后不运行，输入 2 后选择分区将下载的数据烧写到 NAND FLASH 里去，注意若下载的文件大于分区大小时，BIOS 会有提示，但仍可执行烧写。分区 5 主要用来存储 WINCE 系统的。

BIOS 的 3 号功能主要用来从 NAND FLASH 启动 Linux 和自己的应用程序，注意 Linux 内核和应用程序要烧写到分区 1。

BIOS 的 4 号功能可以擦除 NAND FLASH 分区。

BIOS 的 5 号功能可以烧写 NOR FLASH (SST39VF160)，此功能只在 NOR FLASH 启动后才可用，是用以更新 NOR FLASH 中的启动程序，如果用户自编了启动程序，可以在下载完后用此功能烧写 SST39VF160，注意如果启动失败后，只能用 JTAG 方式重新烧写 SST39VF160 了。

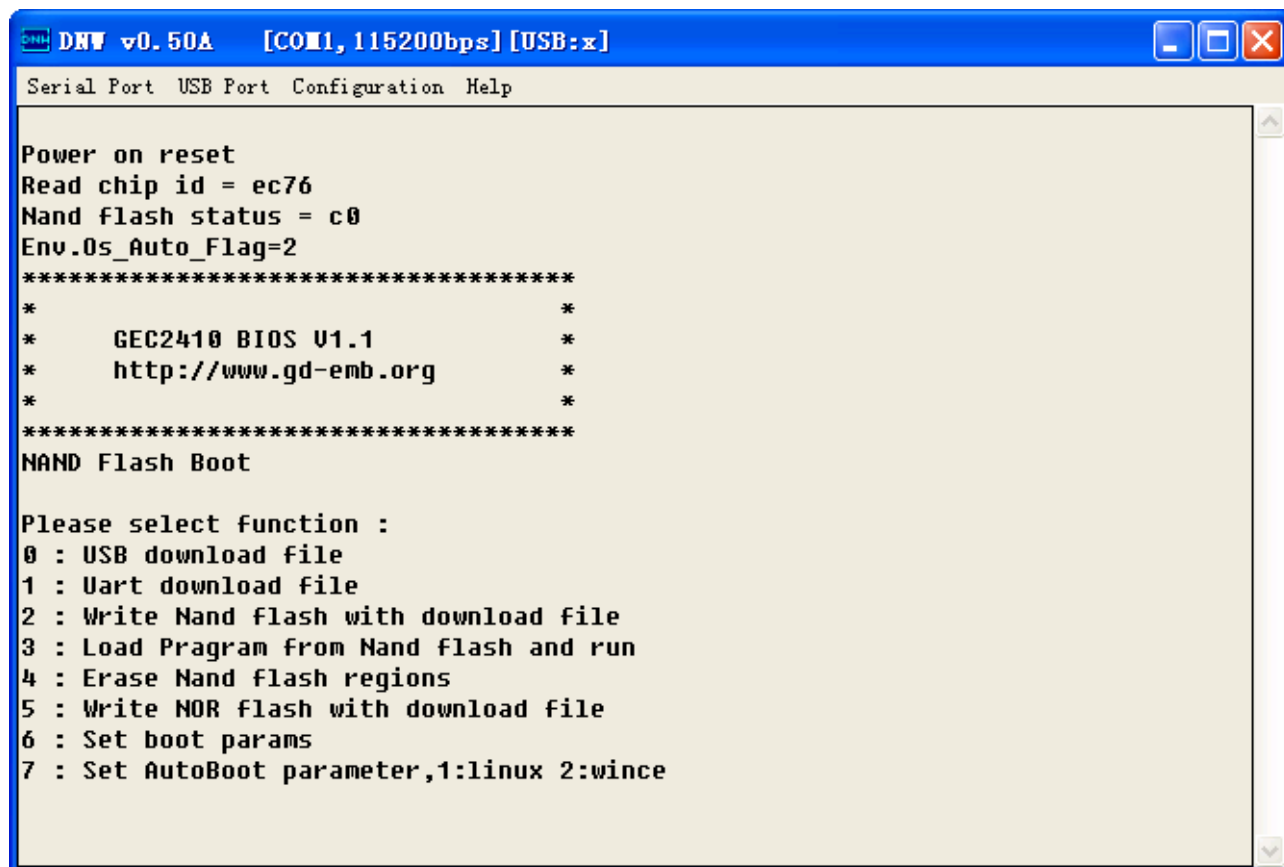
注意：烧写 NOR FLASH 的时候，要保证核心板上的 JP1 的跳线帽要取掉，即工作在 NOR FLASH 启动模式。

BIOS 的 6 号功能可设置启动参数，主要是对 linux 调试用。

BIOS 的 7 号功能是设置自启动参数，

- 1: 自启动 Linux ；
- 2: 自启动 wince ；
- 其他: 不进入自启动模式

下面是 BIOS 的启动信息。



```
DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

Power on reset
Read chip id = ec76
Nand flash status = c0
Env.0s_Auto_Flag=2
*****
*                                     *
*   GEC2410 BIOS V1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
```

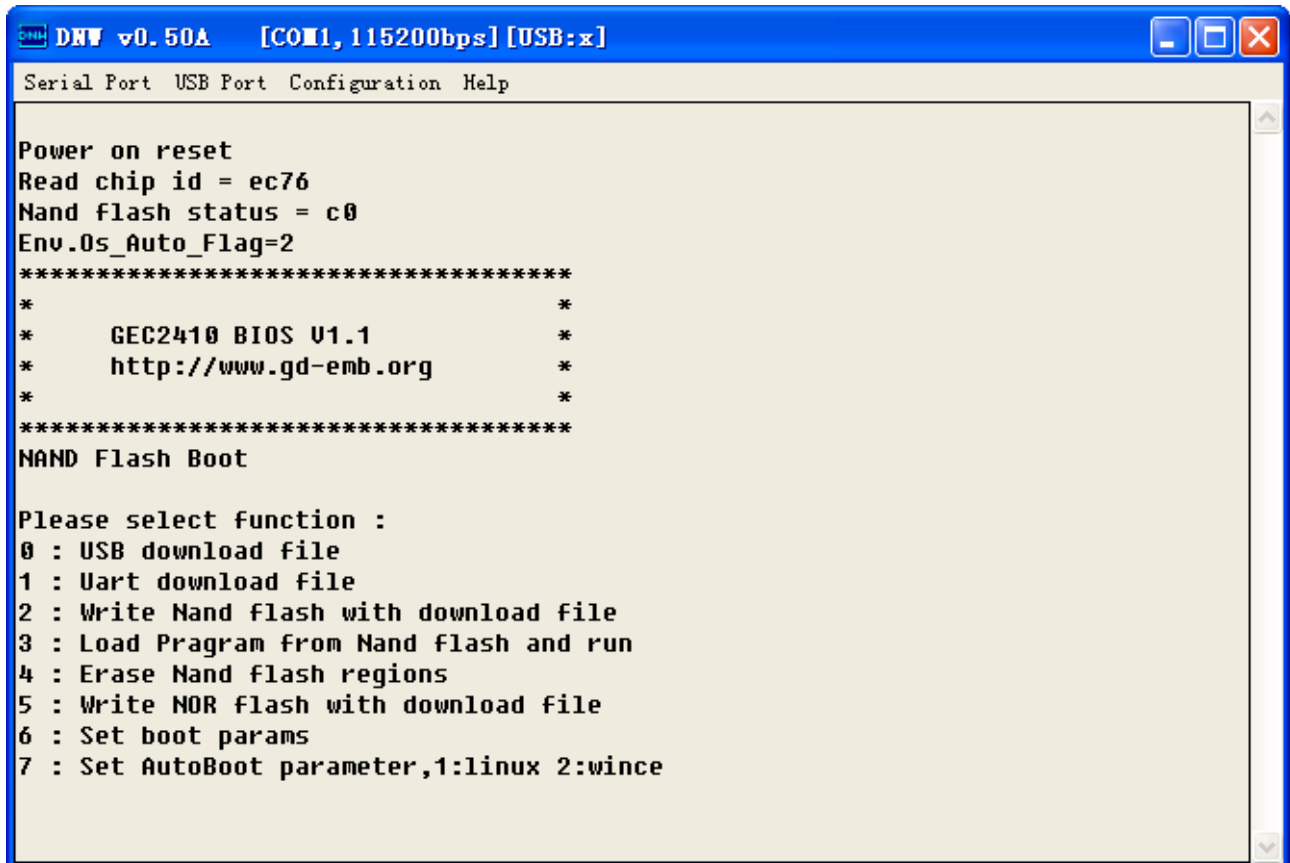
2.4 非操作系统下的外围资源测试

在非操作系统下，主要测试 RTC 测试，按键测试，蜂鸣器测试，ADC 测试，IIC 测试，触摸屏测试，LCD 屏测试，音频输出测试，音频输入测试，红外线测试，SD 卡测试，以及 COM2 测试。

2.4.1 运行测试程序

先要通过 BIOS 将测试程序 GEC2410_APP.bin（这个 bin 文件在光盘目录的“目标代码”文件夹下）下载运行，首先下载测试程序，下载步骤如下：

- （1）上电启动开发板，进入 BIOS 界面

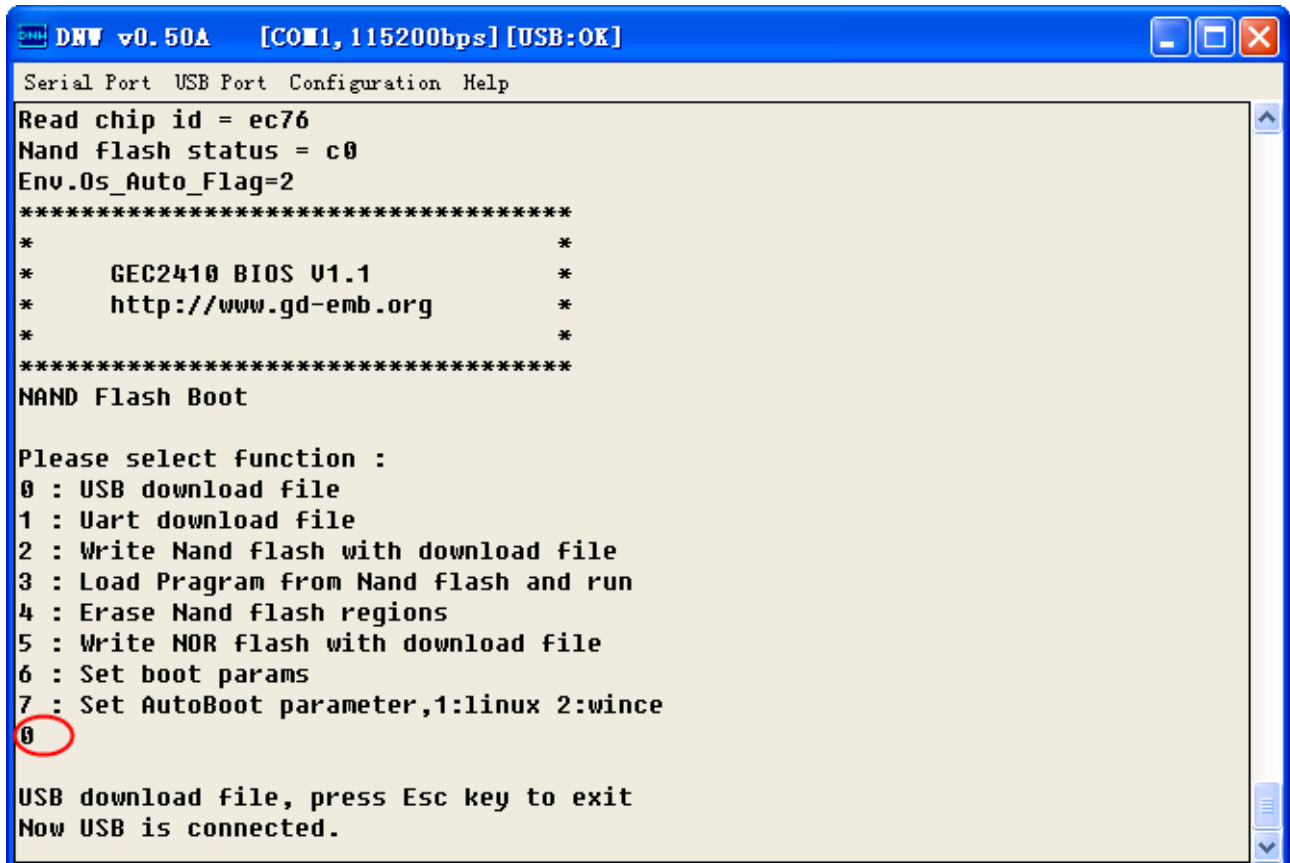


```
DNV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

Power on reset
Read chip id = ec76
Nand flash status = c0
Env.Os_Auto_Flag=2
*****
*                                     *
*   GEC2410 BIOS U1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

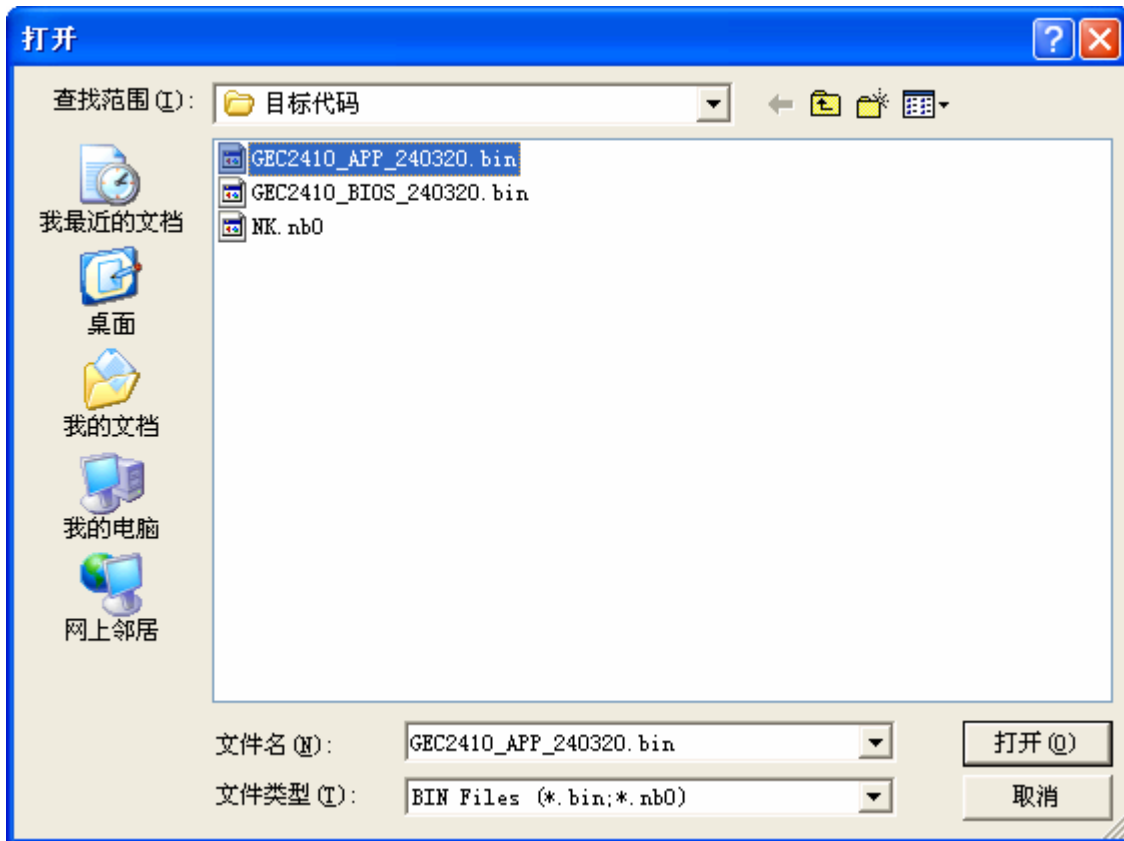
Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
```

(2) 接上 USB DEVICE 的连接线，选择“0”，选择 USB 下载，注意用 USB 下载要先在 PC 端装好驱动程序，保证 USB 连接好，有时 PC 端出现发现无法识别的 USB 设备时，可在 BIOS 中输入 ESC 取消下载，等几秒钟再输入“0”启动 USB 下载。

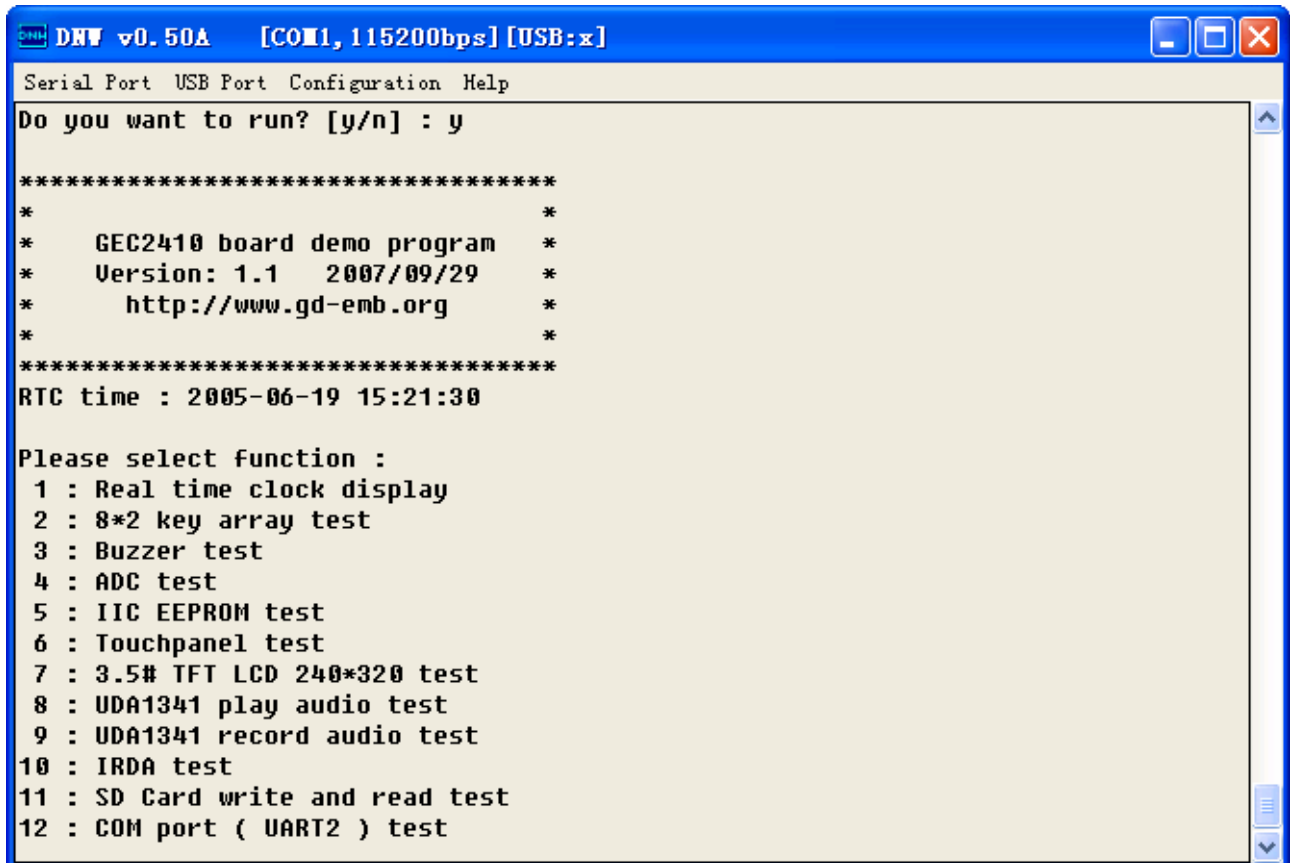


出现“Now USB is connected”说明 USB 正确枚举了，这时可以通过 USB 下载了。

(3) 点击“USB Port” ”Transmit”选项，选择 GEC2410_APP.bin（这个映像文件在光盘目录的”目标代码“文件夹下）



(4) 下载结束后，会提示是否要立即运行，这时输入“Y”，这时将运行刚刚下载的程序。



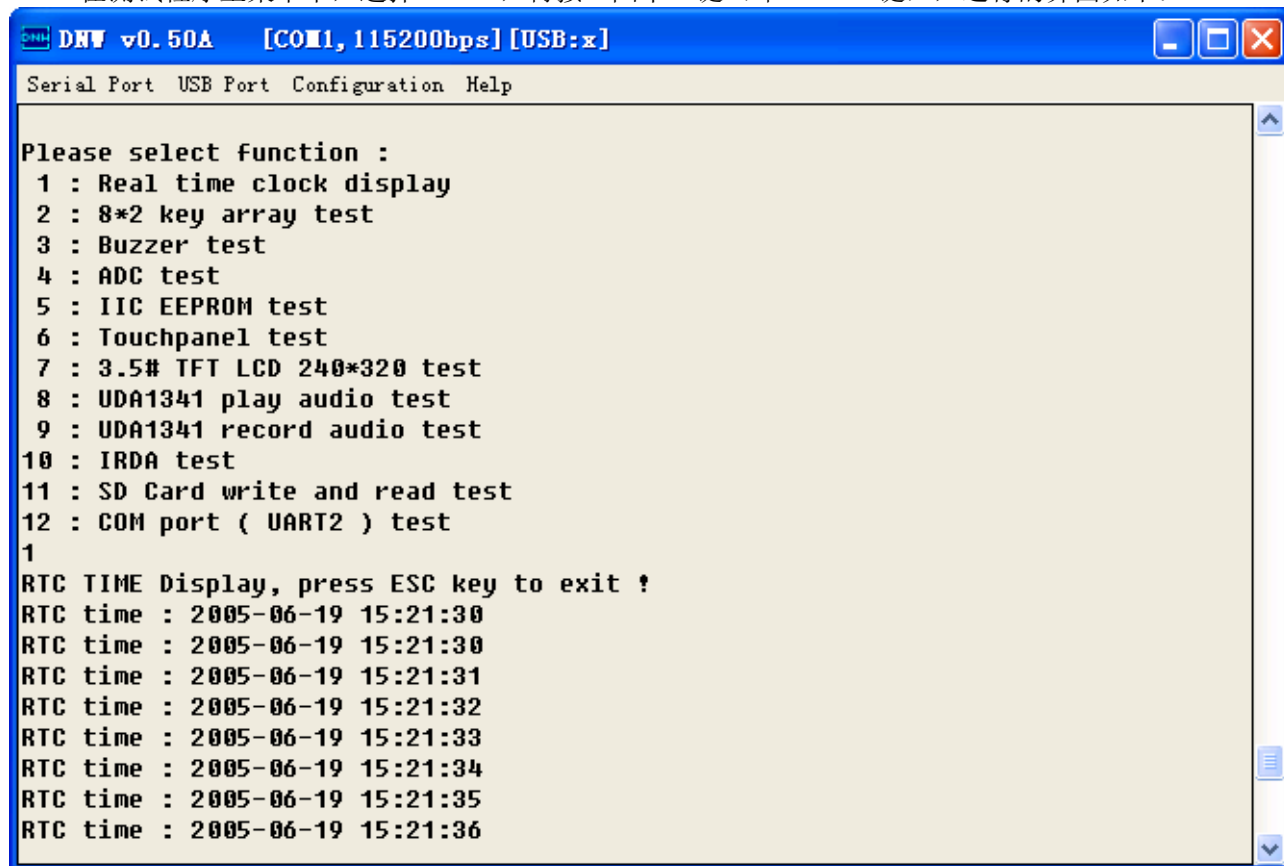
这时测试程序将会运行起来了。

2.4.2 相关外围资源测试

测试程序运行后，就可以进行相应的外围资源测试了，通过选择测试程序主功能菜单相应的选项，就可以进行测试。

（1） RTC 实时时钟测试

在测试程序主菜单中，选择“1”，再按“回车”键（即 Enter 键），运行的界面如下：



```
DNW v0.50A [COM1, 115200bps] [USB:x]
Serial Port USB Port Configuration Help

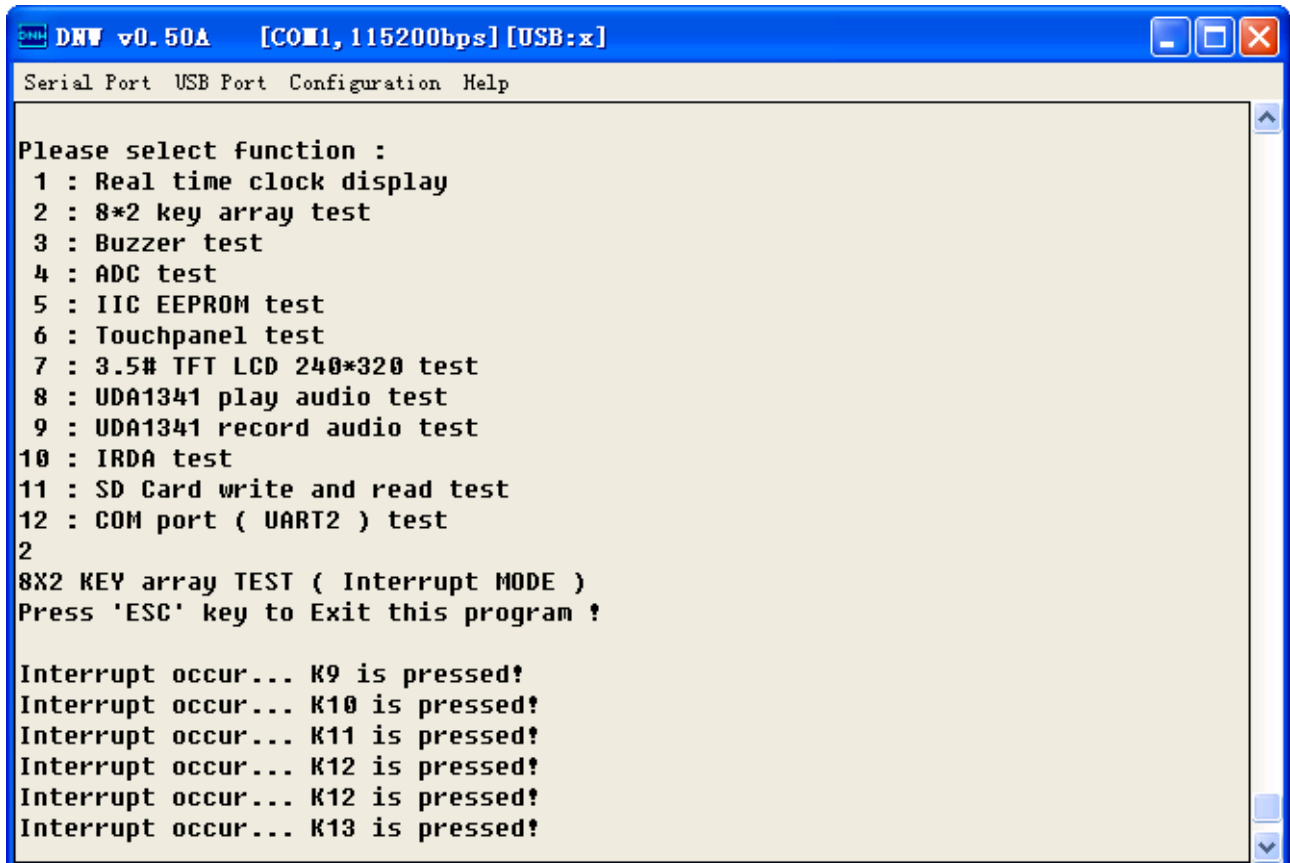
Please select function :
1 : Real time clock display
2 : 8*2 key array test
3 : Buzzer test
4 : ADC test
5 : IIC EEPROM test
6 : Touchpanel test
7 : 3.5# TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
1
RTC TIME Display, press ESC key to exit !
RTC time : 2005-06-19 15:21:30
RTC time : 2005-06-19 15:21:30
RTC time : 2005-06-19 15:21:31
RTC time : 2005-06-19 15:21:32
RTC time : 2005-06-19 15:21:33
RTC time : 2005-06-19 15:21:34
RTC time : 2005-06-19 15:21:35
RTC time : 2005-06-19 15:21:36
```

看到秒钟在不断的变化，说明 RTC 在正常工作，有一点要说明的是，这个时间并不是当前的时间。

按“ESC”键可以退出该测试，并返回到主菜单中。

（2） 按键测试

在主菜单中，输入“2”，再按“回车”键（即 Enter 键），将进行按键测试，按键测试运行起来后，可以按开发板上的 K1~K16 按键进行测试，按一个键将在串口打印相应的按键信息，运行界面如下：



```
DNV v0.50A [COM1,115200bps] [USB:x]
Serial Port  USB Port  Configuration  Help

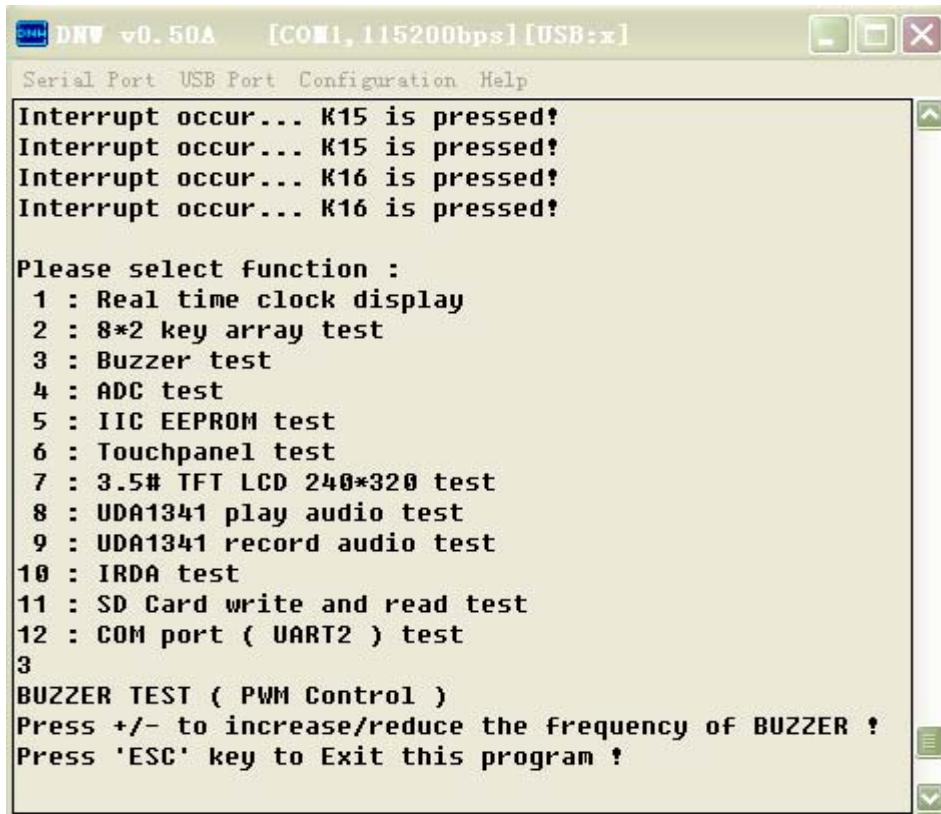
Please select function :
1 : Real time clock display
2 : 8*2 key array test
3 : Buzzer test
4 : ADC test
5 : IIC EEPROM test
6 : Touchpanel test
7 : 3.5# TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
2
8X2 KEY array TEST ( Interrupt MODE )
Press 'ESC' key to Exit this program !

Interrupt occur... K9 is pressed!
Interrupt occur... K10 is pressed!
Interrupt occur... K11 is pressed!
Interrupt occur... K12 is pressed!
Interrupt occur... K12 is pressed!
Interrupt occur... K13 is pressed!
```

按“ESC”键可以退出该测试，并返回到主菜单中。

(3) 蜂鸣器测试

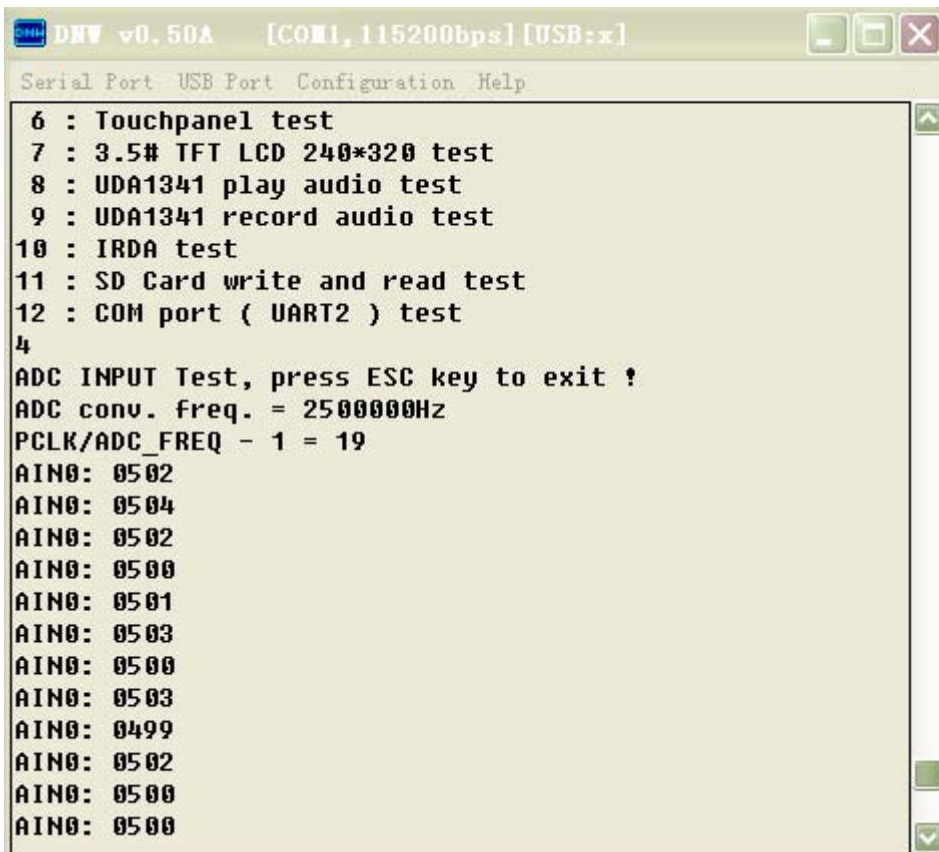
在主菜单中，输入“3”，再按“回车”键（即 Enter 键），将进行蜂鸣器测试，注意开发板上的 J1 要接上跳线帽，蜂鸣器测试运行起来，将会听到蜂鸣器发出叫声。



按“ESC”键可以退出该测试，并返回到主菜单中。

4) ADC 测试

在主菜单中，输入“4”，再按“回车”键（即 Enter 键），将进行 ADC 测试，运行界面如下图。



```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

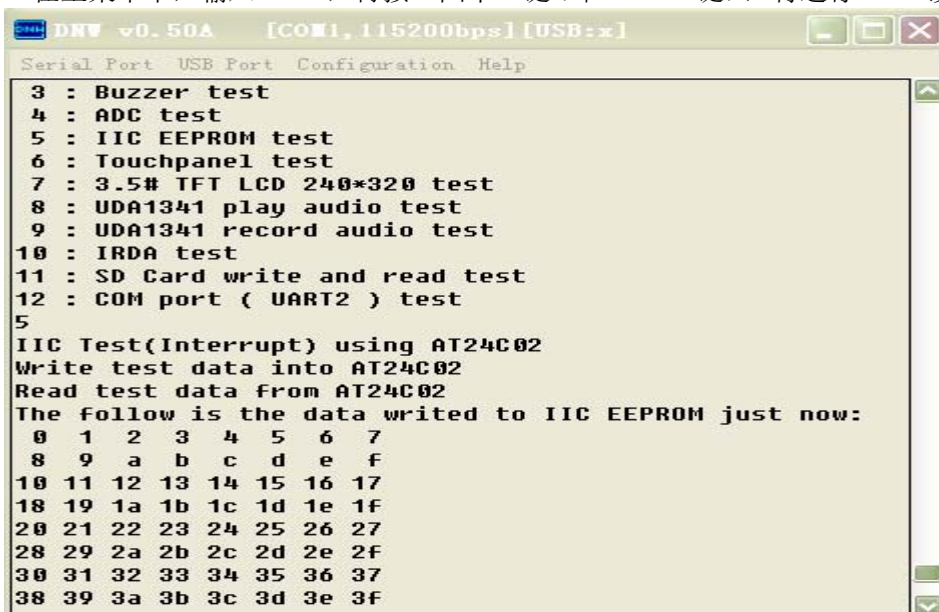
6 : Touchpanel test
7 : 3.5# TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
4
ADC INPUT Test, press ESC key to exit !
ADC conv. freq. = 2500000Hz
PCLK/ADC_FREQ - 1 = 19
AIN0: 0502
AIN0: 0504
AIN0: 0502
AIN0: 0500
AIN0: 0501
AIN0: 0503
AIN0: 0500
AIN0: 0503
AIN0: 0499
AIN0: 0502
AIN0: 0500
AIN0: 0500

```

可以用螺丝刀调节开发板上的 R34 (AIN0) ，可以看到 ADC 的值在不断的变化。
按“ESC”键可以退出该测试，并返回到主菜单中。

(5) IIC 测试

在主菜单中，输入“5”，再按“回车”键(即 Enter 键)，将进行 IIC 读写测试，运行界面如下图。



```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

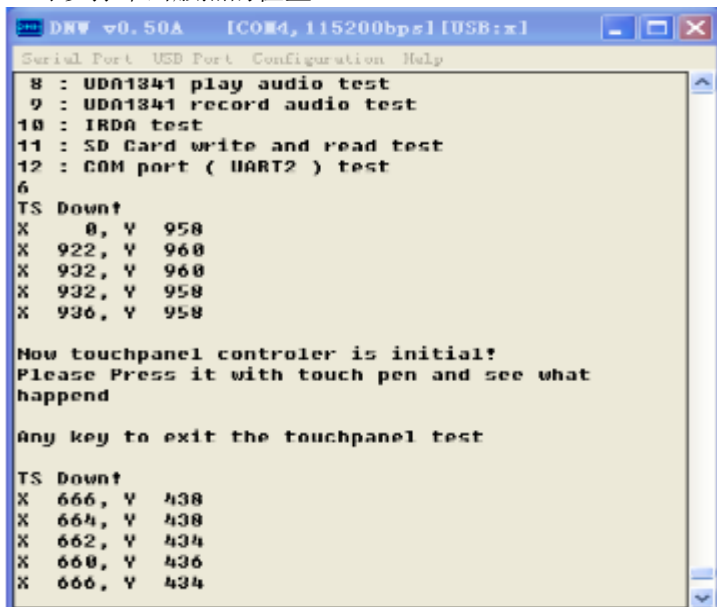
3 : Buzzer test
4 : ADC test
5 : IIC EEPROM test
6 : Touchpanel test
7 : 3.5# TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
5
IIC Test(Interrupt) using AT24C02
Write test data into AT24C02
Read test data from AT24C02
The follow is the data writed to IIC EEPROM just now:
0 1 2 3 4 5 6 7
8 9 a b c d e f
10 11 12 13 14 15 16 17
18 19 1a 1b 1c 1d 1e 1f
20 21 22 23 24 25 26 27
28 29 2a 2b 2c 2d 2e 2f
30 31 32 33 34 35 36 37
38 39 3a 3b 3c 3d 3e 3f

```

这个测试，主要是通过向 AT24C02 写 0~36 的数据，然后读出来。
这个测试结束后，会自动退回到主菜单中。

(6) 触摸屏测试

如果选购了 LCD 屏，请用 50 针的排线，将 LCD 屏与开发板的 LCD 接口连接起来。在主菜单中，输入“6”，再按“回车”键（即 Enter 键），将进行触摸屏测试，接着点击触摸屏，可以看到串口可以打印出触点的位置。



```

DNW v0.50A [COM4, 115200bps] [USB:x]
Serial Port USB Port Configuration Help
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
6
TS Down!
X 0, Y 958
X 922, Y 960
X 932, Y 960
X 932, Y 958
X 936, Y 958

How touchpanel controler is initial!
Please Press it with touch pen and see what
happend

Any key to exit the touchpanel test

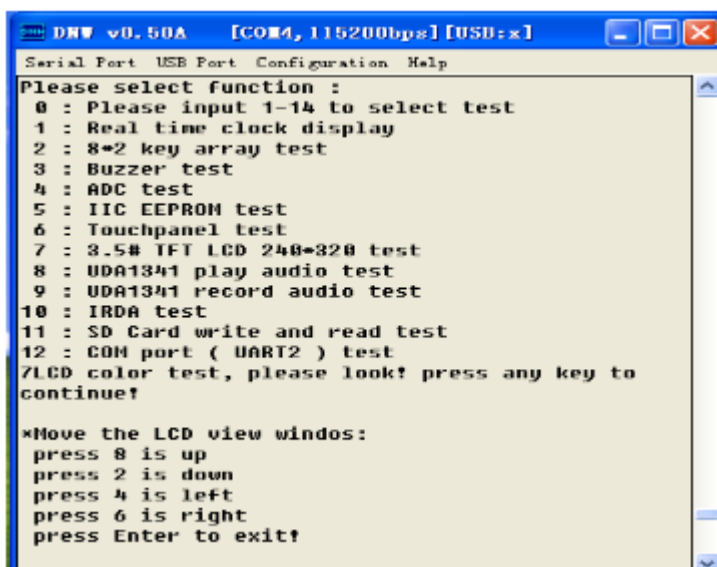
TS Down!
X 666, Y 438
X 664, Y 438
X 662, Y 434
X 660, Y 436
X 666, Y 434

```

按“ESC”键可以退出该测试，并返回到主菜单中。

(7) LCD 测试

如果选购了 LCD 屏，请用 50 针的排线，将 LCD 屏与开发板的 LCD 接口连接起来。在主菜单中，输入“7”，再按“回车”键（即 Enter 键），将进行 LCD 测试，按任意键将显示继续，接着再按“回车”键（即 Enter 键），将退出 LCD 测试。



```

DNW v0.50A [COM4, 115200bps] [USB:x]
Serial Port USB Port Configuration Help
Please select function :
0 : Please input 1-14 to select test
1 : Real time clock display
2 : 8*2 key array test
3 : Buzzer test
4 : ADC test
5 : IIC EEPROM test
6 : Touchpanel test
7 : 3.5" TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
7LCD color test, please look! press any key to
continue!

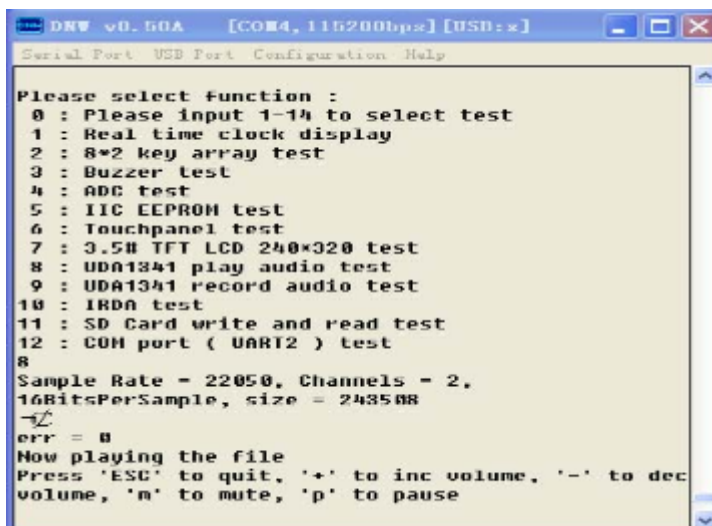
*Move the LCD view windos:
press 8 is up
press 2 is down
press 4 is left
press 6 is right
press Enter to exit!

```

(8) 音频输出测试

先将音箱接到开发板的 J3 接口。

在主菜单中，输入“8”，再按“回车”键（即 Enter 键），将进行音频输出测试，这时将从音箱听到声音。

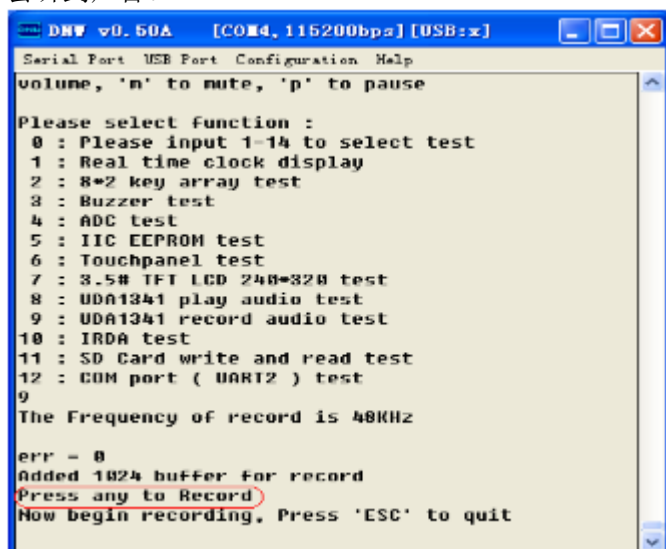


按“+”或“-”可以增加或减少音量，按“ESC”键可以退出测试，返回主菜单。

(9) 音频输入测试

先将音箱接到开发板的 J3 接口，并将麦克风与开发板的 J2 相连。

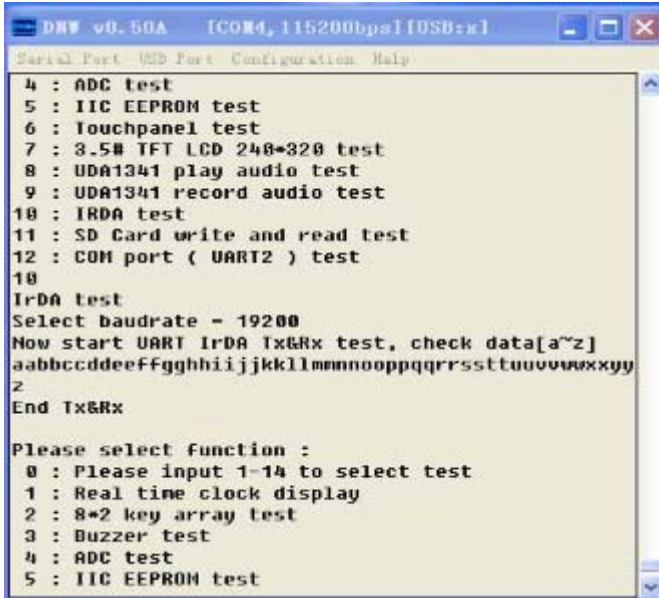
在主菜单中，输入“9”，再按“回车”键（即 Enter 键），将进行音频输入测试，这时按照该测试的提示，输入任意键，将会进行录音测试，可以对着开发板的 MIC（MK1）或麦克风发声，在音箱处将会听到声音。



按“ESC”键可以退出该测试，并返回到主菜单中。

(10) 红外线测试

在主菜单中，输入“10”，再按“回车”键（即 Enter 键），将进行红外线测试，运行界面如下：



```

DNV v0.50A [COM4,115200bps][USB:x]
Serial Port USB Port Configuration Help
4 : ADC test
5 : IIC EEPROM test
6 : Touchpanel test
7 : 3.5 inch TFT LCD 240*320 test
8 : UDA1341 play audio test
9 : UDA1341 record audio test
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
18
IrDA test
Select baudrate - 19200
Now start UART IrDA Tx&Rx test, check data[a~z]
aabbccddeeffghhiijjkkllmmnnnooppqrrssttuuvvwxxyy
z
End Tx&Rx

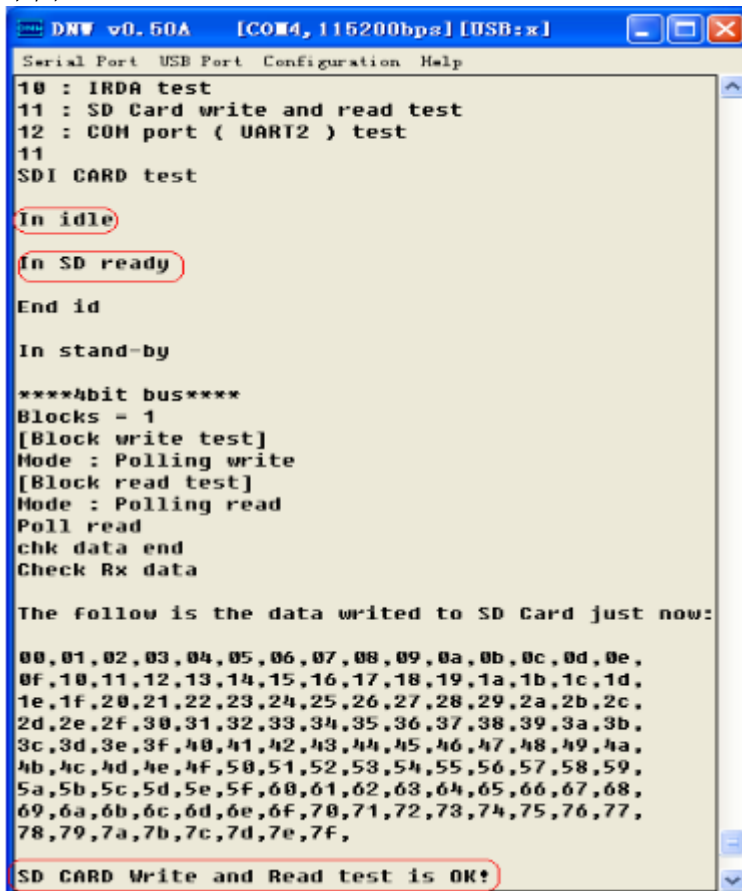
Please select function :
0 : Please input 1-14 to select test
1 : Real time clock display
2 : 8*2 key array test
3 : Buzzer test
4 : ADC test
5 : IIC EEPROM test
    
```

红外线进行的是自收发模式测试，测试结束后，会自动返回到主菜单界面。

(11) SD 卡测试

先将 SD 卡插入开发板的 SD 卡座。

在主菜单中，输入“11”，再按“回车”键（即 Enter 键），将进行 SD 卡读写测试，运行界面如下图。



```

DNV v0.50A [COM4,115200bps][USB:x]
Serial Port USB Port Configuration Help
10 : IRDA test
11 : SD Card write and read test
12 : COM port ( UART2 ) test
11
SDI CARD test
In idle
In SD ready
End id
In stand-by
****4bit bus****
Blocks = 1
[Block write test]
Mode : Polling write
[Block read test]
Mode : Polling read
Poll read
chk data end
Check Rx data

The follow is the data writed to SD Card just now:

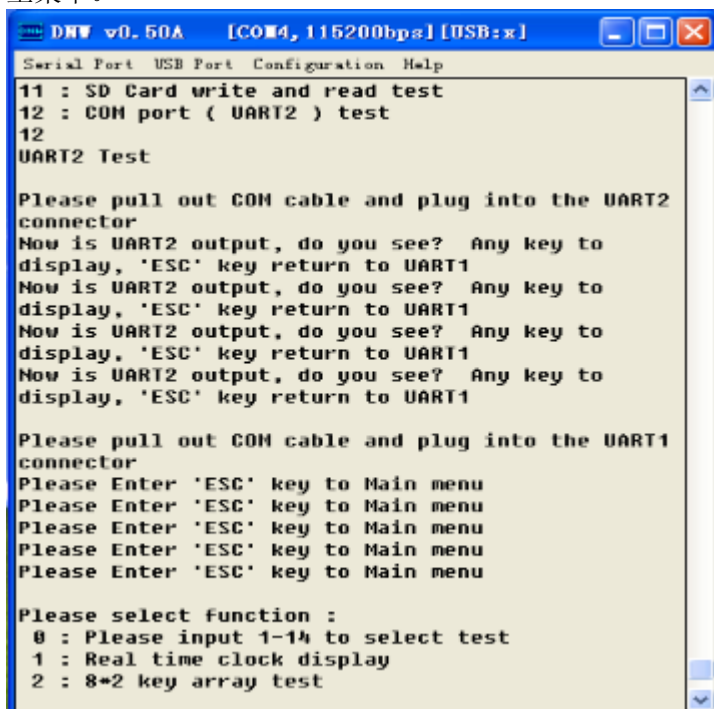
00,01,02,03,04,05,06,07,08,09,0a,0b,0c,0d,0e,
0f,10,11,12,13,14,15,16,17,18,19,1a,1b,1c,1d,
1e,1f,20,21,22,23,24,25,26,27,28,29,2a,2b,2c,
2d,2e,2f,30,31,32,33,34,35,36,37,38,39,3a,3b,
3c,3d,3e,3f,40,41,42,43,44,45,46,47,48,49,4a,
4b,4c,4d,4e,4f,50,51,52,53,54,55,56,57,58,59,
5a,5b,5c,5d,5e,5f,60,61,62,63,64,65,66,67,68,
69,6a,6b,6c,6d,6e,6f,70,71,72,73,74,75,76,77,
78,79,7a,7b,7c,7d,7e,7f,

SD CARD Write and Read test is OK!
    
```

出现上面红框的提示，说明 SD 测试是 OK 的，测试完成后，会自动退回到主菜单。

(12) 串口 2 测试

在主菜单中，输入“12”，再按“回车”键（即 Enter 键），将进行 COM2 测试，这时将连在 UART1 的串口线，换成连接到 UART2 上，这时将在串口工具（DNW），看到串口 2 的打印信息。要返回串口 1，请按“ESC”键，然后将串口线重新接到串口 1 上，接着按“回车”键（即 Enter 键），将返回主菜单。



2.5 Linux 操作系统下的外围资源测试

在操作系统下主要测试网络，USB HOST 接口。

启动 BIOS 后，“3”将会启动 Linux 系统输入，（当然要保证 Linux 烧写到 NAND FLASH 的相应的分区中，关于 Linux 的烧写，请看后面的章节）。

(1) 网络测试

用交叉网线将 PC 的网络接口与开发板的网络接口连接起来。

Linux 启动后，按“回车”键，进入命令行模式下，界面如下图。

```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
channel QPE/IME added
Reading data from NAND FLASH without ECC is not recommended
Reading data from NAND FLASH without ECC is not recommended
Create pluginlibman in libqpe
Use QPEApplication's PluginLibraryManager
QuickLauncher running
channel QPE/Application/quicklauncher added
channel QPE/QuickLauncher-59 added
Registered QPE/QuickLauncher-59

In: `/dev/touchscreen/0': File exists
Reading data from NAND FLASH without ECC is not recommended

[root@none] /]#
[root@none] /]# Reading data from NAND FLASH without ECC is not recommended

Cannot suspend - no APM support in kernel

[root@none] /]#

```

输入命令: `ifconfig eth0 192.168.3.100`

用 `ifconfig` 命令来配置开发板的 IP 地址, 注意, 开发板的 IP 地址要与 PC 的 IP 地址要设置为同一网段, 这里设置为 192.168.3.100, 同时 PC 上的网络防火墙要关闭。

IP 地址配置好后, 打开 PC 端命令窗口, 输入 PING 命令来 PING 开发板。

`ping -t 192.168.3.100`

```

C:\WINDOWS\System32\cmd.exe - ping -t 192.168.3.100
C:\>
C:\>ping -t 192.168.3.100

Pinging 192.168.3.100 with 32 bytes of data:
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255
Reply from 192.168.3.100: bytes=32 time<1ms TTL=255

```

出现上面的信息, 就说明网络是正常的。

(2) USB HOST 测试

将 U 盘插入 USB_HOST0 接口 (J7), Linux 将检测 U 盘, 检测到后, 自动给 U 盘分配一个设备节点, 正确检测到 U 盘的信息如下:



```

DHV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

[root@(none) /]# hub.c: USB new device connect on bus1/1,
assigned device number 2

scsi0 : SCSI emulation for USB Mass Storage devices

  Vendor: Multi      Model: Flash Reader      Rev: 1.00
  Type:   Direct-Access      ANSI SCSI revision:
02

Attached scsi removable disk sda at scsi0, channel 0, id 0, lun
0

SCSI device sda: 3920896 512-byte hdwr sectors (2007 MB)
sda: Write Protect is on
Partition check:

/dev/scsi/host0/bus0/target0/lun0: p1
  
```

正确检测到 U 盘后，接下来就可以挂接 U 盘了。

挂接 U 盘的命令如下：

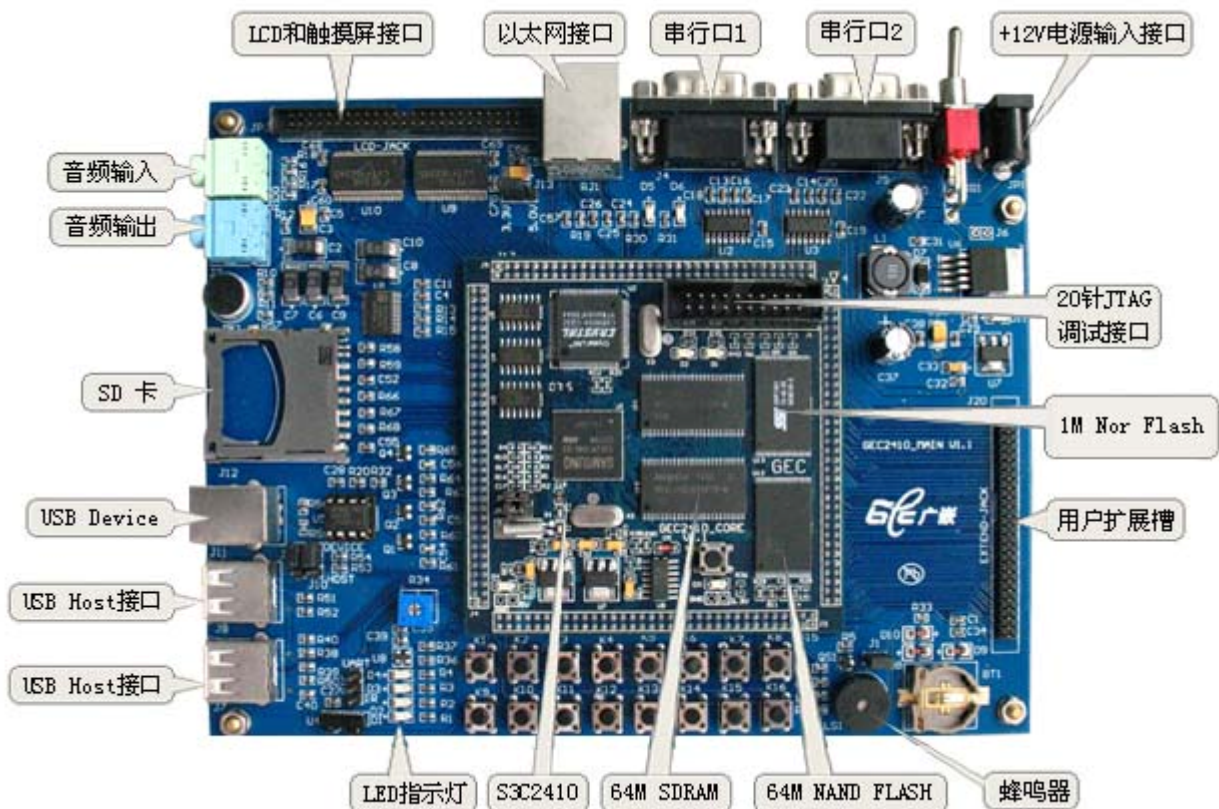
```
mkdir /tmp/usb
```

```
mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part1 /tmp/usb
```

这样就将 U 盘挂接到/tmp/usb 目录下，这时我们就可以进入/tmp/usb 目录，对 U 盘进行读写，删除，建立文件等操作了。

2.6 用 SJF2410 工具将 BIOS 烧写到 NAND FLASH

(1) 用 20 针排线将 GEC2410 的 20 针 JTAG 接口 (J1) 与 JTAG 小板的相连，再用并口线将 JTAG 小板与 PC 机相连。然后将光盘里的“Flash 烧写”文件夹拷贝到 C 盘（硬盘的其他地方也行）。如下图红色部分：



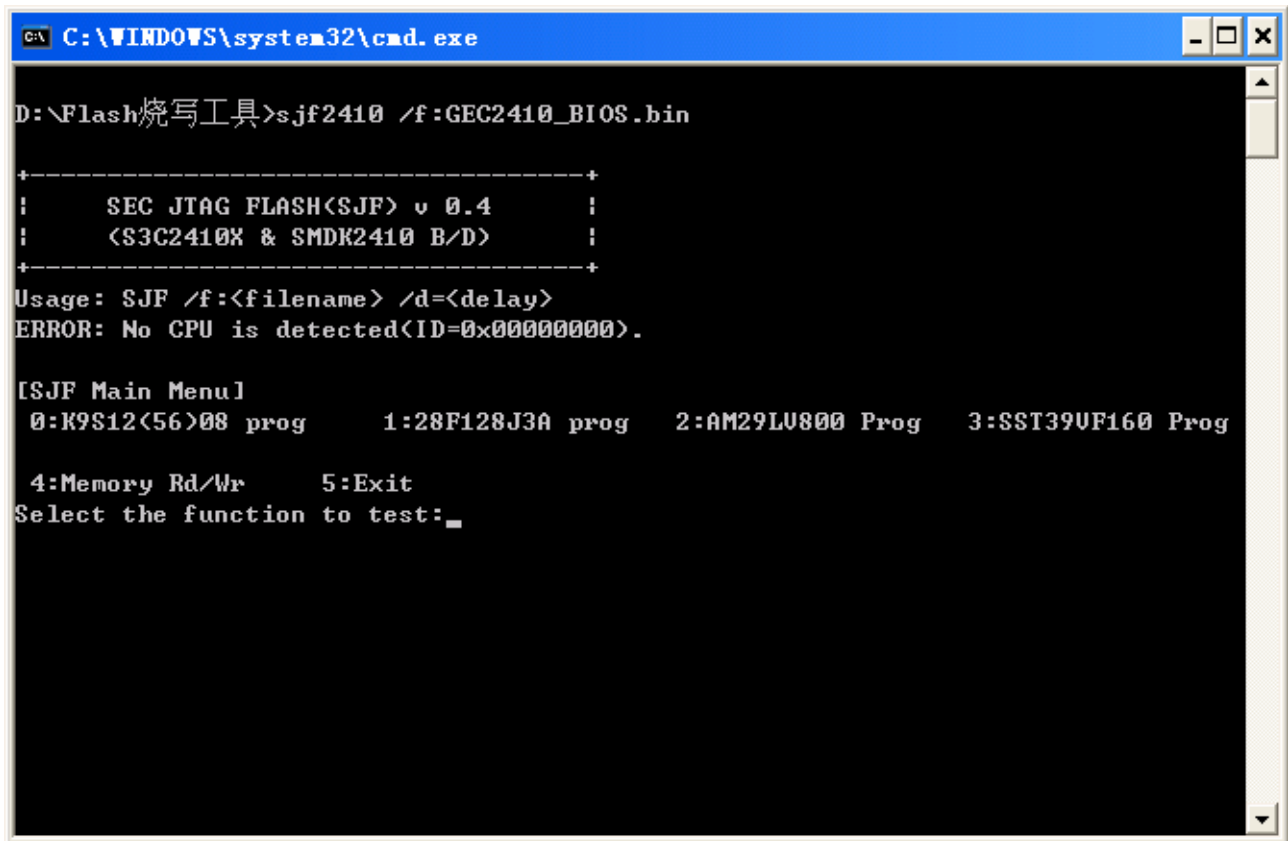
(2) 安装 giveio 驱动，进入“Flash 烧写”文件夹下，点击“安装驱动.exe”，将弹出如下界面。



先点击“InStall Parallel Port Driver”栏目下的“Remove”按钮，然后点击该栏目下的“Install”按钮。

出现“Service is installed and run”说明 giveio 驱动安装成功。

(3) 然后点击该文件夹下的 SJF2410_BIOS.BAT 批处理文件，显示信息如下：



```
C:\WINDOWS\system32\cmd.exe

D:\Flash烧写工具>sjf2410 /f:GEC2410_BIOS.bin

+-----+
|      SEC JTAG FLASH(SJF) v 0.4      |
|      (S3C2410X & SMDK2410 B/D)      |
+-----+
Usage: SJF /f:<filename> /d=<delay>
ERROR: No CPU is detected(ID=0x00000000).

[ SJF Main Menu ]
0:K9S12<56>08 prog      1:28F128J3A prog      2:AM29LV800 Prog      3:SST39VF160 Prog

4:Memory Rd/Wr      5:Exit
Select the function to test: _
```

烧写程序所支持的 FLASH 都列出来了，有 K9S1208（NAND，64M）、28F128J3A、AM29LV800、SST39VF160/1 等。

（4）在出现上面的界面后，我们在“Select the Function to test:”提示下，输入‘0’，然后回车，这时将选择 K9S1208 进行烧写，显示如下：

```

C:\WINDOWS\system32\cmd.exe

D:\Flash烧写工具>sjf2410 /f:GEC2410_BIOS.bin

+-----+
|      SEC JTAG FLASH(SJF) v 0.4      |
|      <S3C2410X & SMDK2410 B/D>      |
+-----+

Usage: SJF /f:<filename> /d=<delay>
ERROR: No CPU is detected(ID=0x00000000).

[ SJF Main Menu ]
0:K9S12<56>08 prog      1:28F128J3A prog      2:AM29LV800 Prog      3:SST39VF160 Prog

4:Memory Rd/Wr      5:Exit
Select the function to test:0

[K9S12<56>08 NAND Flash JTAG Programmer]
K9S12<56>08 is detected. ID=0xec76
0:K9S12<56>08 Program      1:K9S12<56>08 Pr BlkPage      2:Exit
Select the function to test :
  
```

(5) 接着在“Select the function to test :”提示下，输入“0”，然后回车，选择 K9F1208 进行烧写，接着在“Input target block number:”栏下输入偏移地址“0”，显示信息如下：

```

C:\WINDOWS\system32\cmd.exe

+-----+
|      SEC JTAG FLASH(SJF) v 0.4      |
|      <S3C2410X & SMDK2410 B/D>      |
+-----+

Usage: SJF /f:<filename> /d=<delay>
ERROR: No CPU is detected(ID=0x00000000).

[ SJF Main Menu ]
0:K9S12<56>08 prog      1:28F128J3A prog      2:AM29LV800 Prog      3:SST39VF160 Prog

4:Memory Rd/Wr      5:Exit
Select the function to test:0

[K9S12<56>08 NAND Flash JTAG Programmer]
K9S12<56>08 is detected. ID=0xec76
0:K9S12<56>08 Program      1:K9S12<56>08 Pr BlkPage      2:Exit
Select the function to test :0

[SMC(K9S1208U0M) NAND Flash Writing Program]

Source size:0h~92d3h

Available target block number: 0~4095
Input target block number:0
  
```

接着回车，回车后，开始烧写程序，程序烧写完成后，会自动退出。

[illegible]

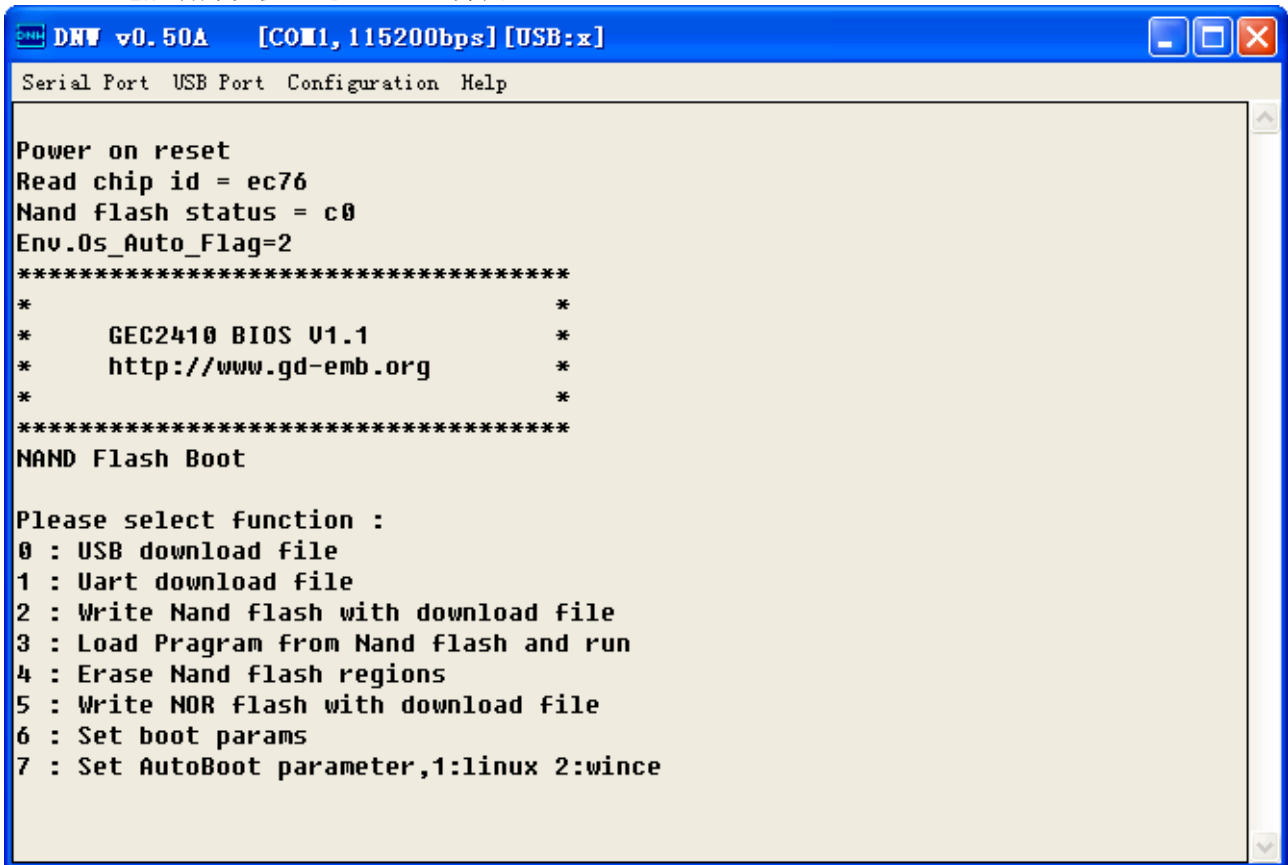
(6) 关闭电源，拔掉 JTAG 插头，将核心板的 JP1 的短路帽接上，将 PC 的串口和开发板的串口 0 通过串口线连接好，在 PC 上启动 DNW 程序，并通过 Configuration 选项设置好 PC 的串口和波特率，点击 OK 后再在 Serial Port 菜单下选择 Connect，注意不要有其他程序占用你所选中的串口。打开开发板电源，烧进 NAND FLASH 的程序 BIOS 会启动运行。

第三章 烧写和启动 linux

工作环境是 WINDOWS。

3.1 烧写 LINUX 内核

(1) 上电启动开发板，进入 BIOS 界面



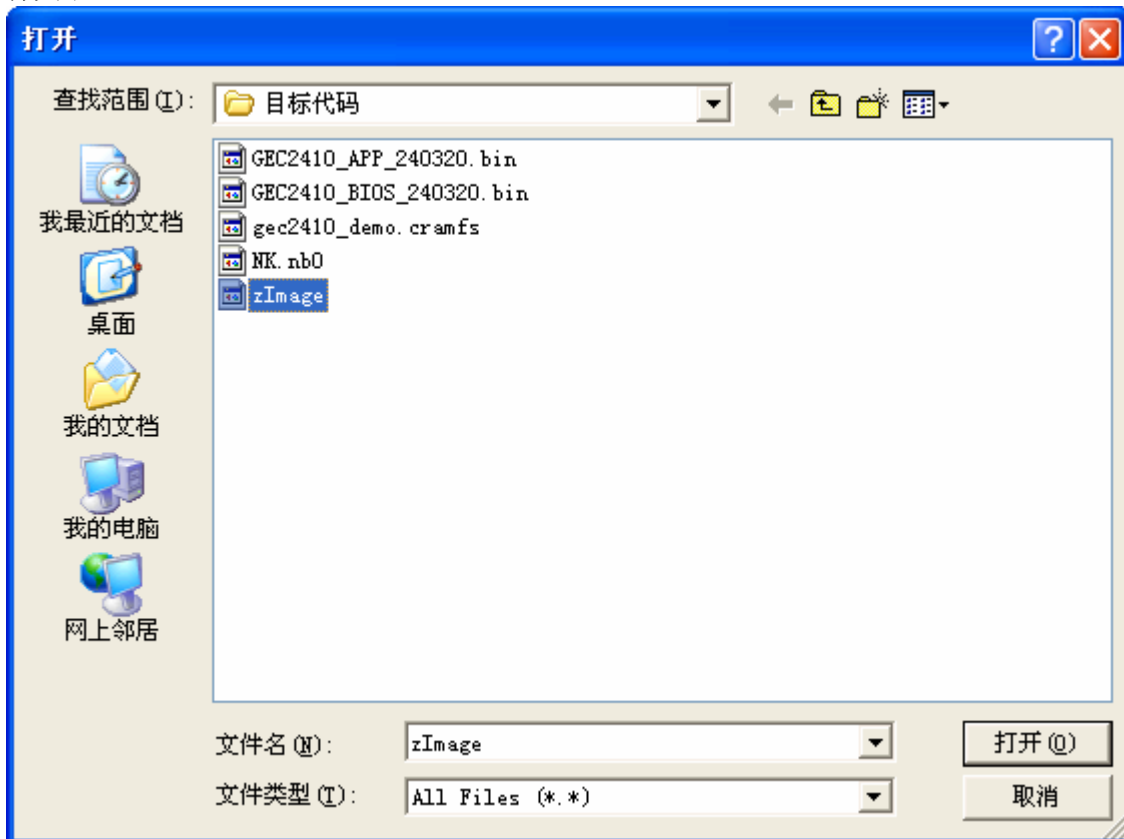
(2) 接上 USB DEVICE 的连接线，选择“0”，选择 USB 下载，注意用 USB 下载要先在 PC 端装好驱动程序，保证 USB 连接好，有时 PC 端出现发现无法识别的 USB 设时，可在 BIOS 中输入 ESC 取消下载，等几秒钟再输入“0”启动 USB 下载。出现“Now USB is connected”说明 USB 正确枚举了，这时可以通过 USB 下载了。

```

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
0

USB download file, press Esc key to exit
Now USB is connected.
    
```

(3) 点击“USB Port” ”Transmit”选项，选择 zImage（这个映像文件在光盘目录的”目标代码“文件夹下）



下载结束后，会提示是否要立即运行，这时输入“n”，将再次进入主功能菜单。

```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
* GEC2410 BIOS V1.1 *
* http://www.gd-emb.org *
* *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
0

USB download file, press Esc key to exit
Now USB is connected.
Now, Downloading [ADDRESS:30200000h,TOTAL:1147462]
ddNow, Checksum calculation
Download O.K.

Do you want to run? [y/n] : n

```

(4) 下载成功后，在出现主功能菜单后，选择“2”，将出现如下界面。

```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
Now, Downloading [ADDRESS:30200000h,TOTAL:1147462]
ddNow, Checksum calculation
Download O.K.

Do you want to run? [y/n] : n

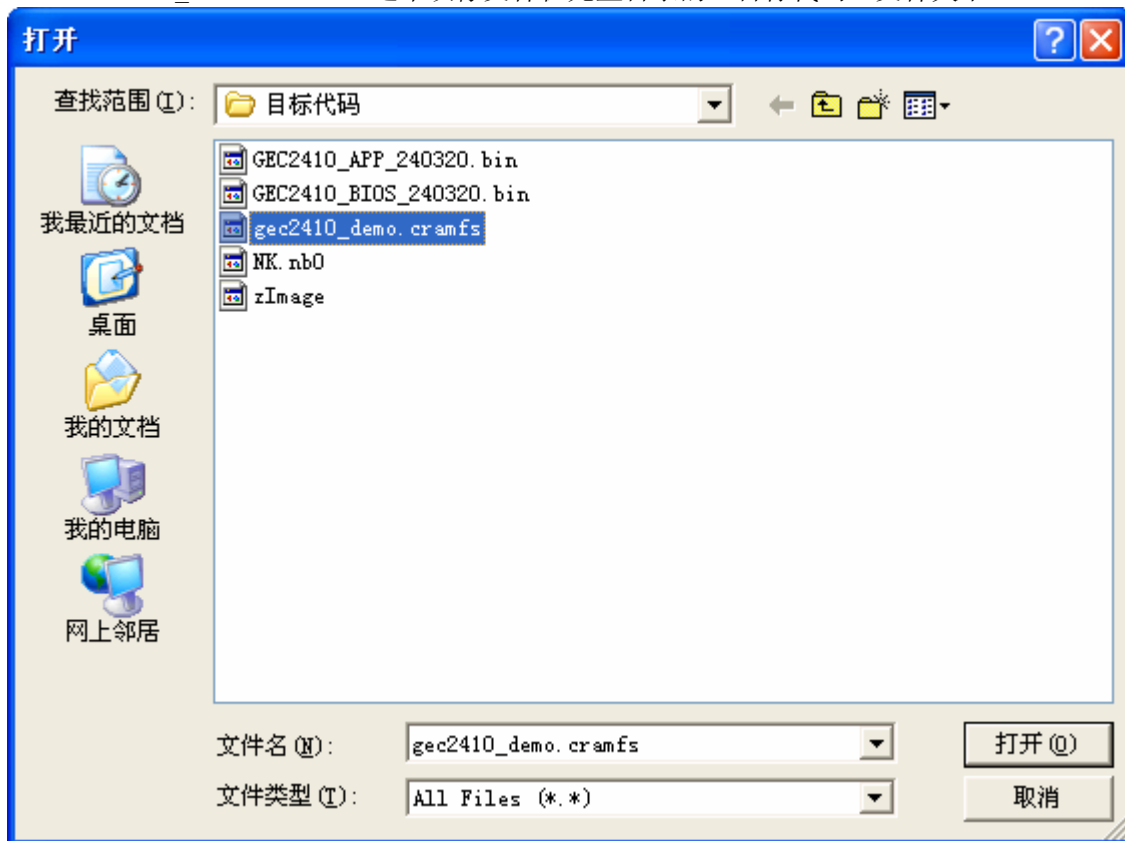
Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
2
Read chip id = ec76
Nand flash status = c0
Please select which region to write : Esc to abort
0 : offset 0x0 , size 0x30000 [boot]
1 : offset 0x30000 , size 0x1d0000 [kernel]
2 : offset 0x200000 , size 0x1e00000 [cramfs]
3 : offset 0x2000000 , size 0x2000000 [ext-fs3]

```

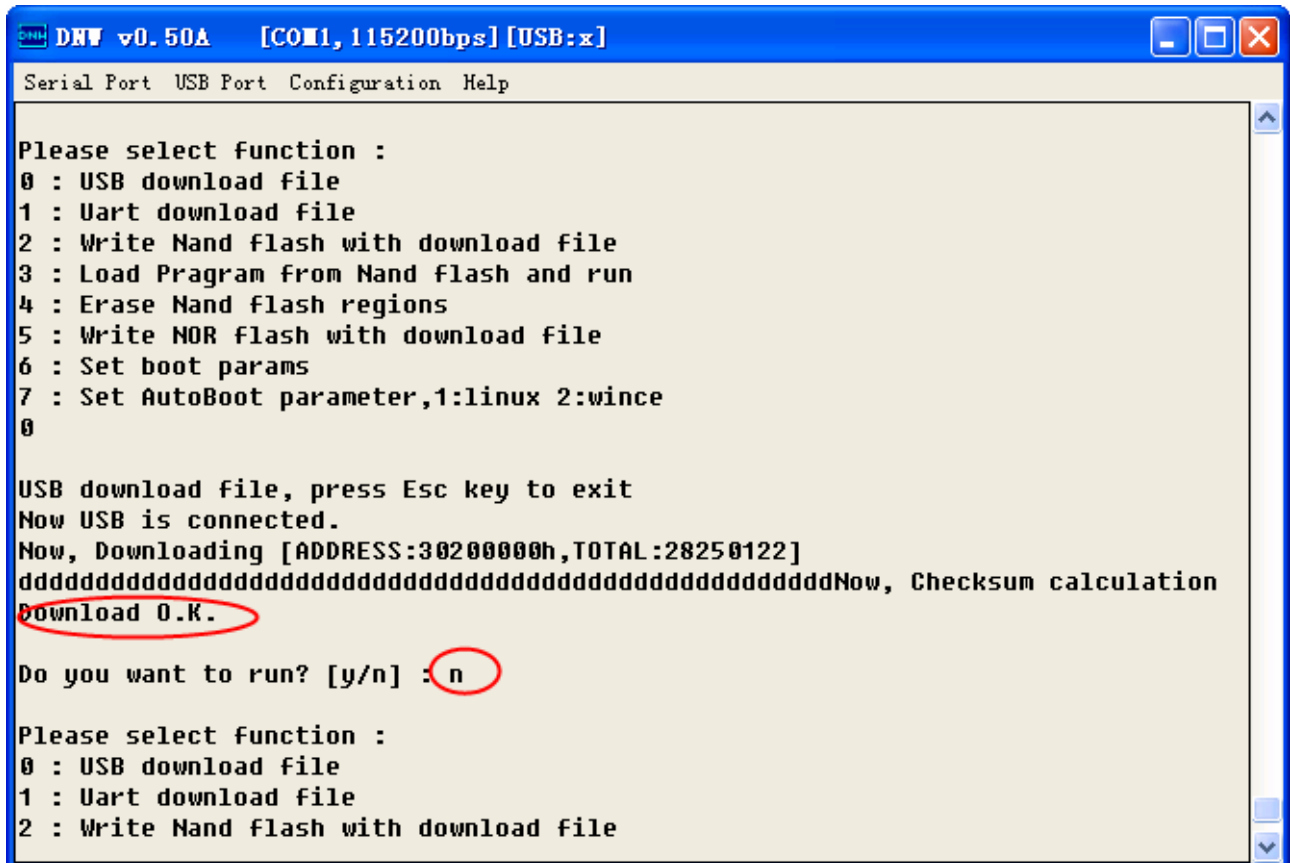
在出现上面的提示后，输入“1”，在接下来的提示输入“Y”，将 zImage 烧写到 NAND FLASH 的分区 1 中。烧写成功后，会自动进入主功能菜单。

3.2 烧写根文件系统

- (1) 烧写 LINUX 根文件系统的前面步骤与上一节《烧写 Linux 内核》的（1）和（2）步骤是一样的。
- (2) 当 USB DEVICE 成功枚举后，点击“USB Port” ”Transmit”选项，选择 GEC2410_demo.cramfs（这个映像文件在光盘目录的”目标代码“文件夹下）



下载结束后，会提示是否要立即运行，这时输入“n”，将再次进入主功能菜单。



```
DNV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

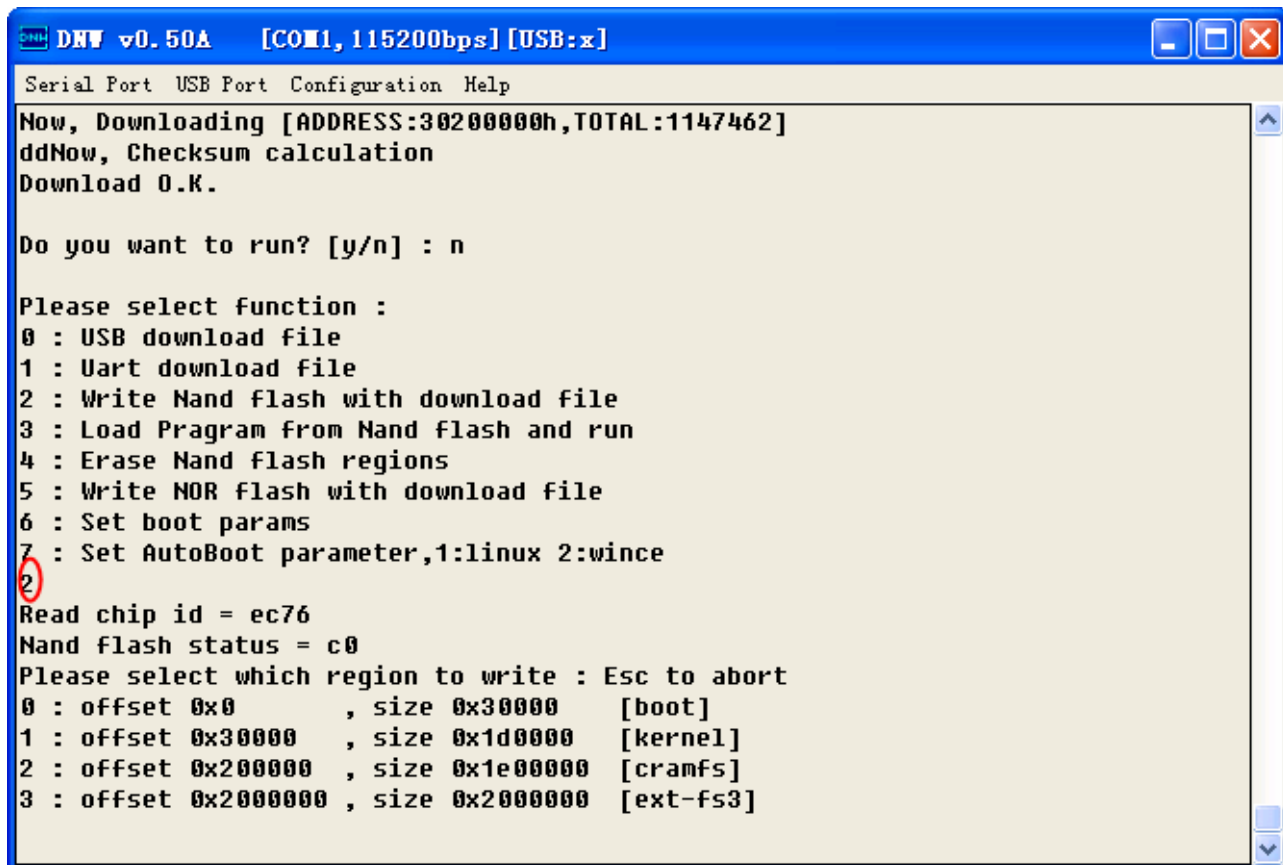
Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
0

USB download file, press Esc key to exit
Now USB is connected.
Now, Downloading [ADDRESS:30200000h,TOTAL:28250122]
ddddddddddddddddddddddddddddddddddddddddddddddddddddddNow, Checksum calculation
Download O.K.

Do you want to run? [y/n] : n

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
```

(4) 下载成功后，在出现主功能菜单后，选择“2”，将出现如下界面。



```

DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
Now, Downloading [ADDRESS:30200000h,TOTAL:1147462]
ddNow, Checksum calculation
Download O.K.

Do you want to run? [y/n] : n

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
2
Read chip id = ec76
Nand flash status = c0
Please select which region to write : Esc to abort
0 : offset 0x0 , size 0x30000 [boot]
1 : offset 0x30000 , size 0x1d0000 [kernel]
2 : offset 0x200000 , size 0x1e0000 [cramfs]
3 : offset 0x200000 , size 0x200000 [ext-fs3]
  
```

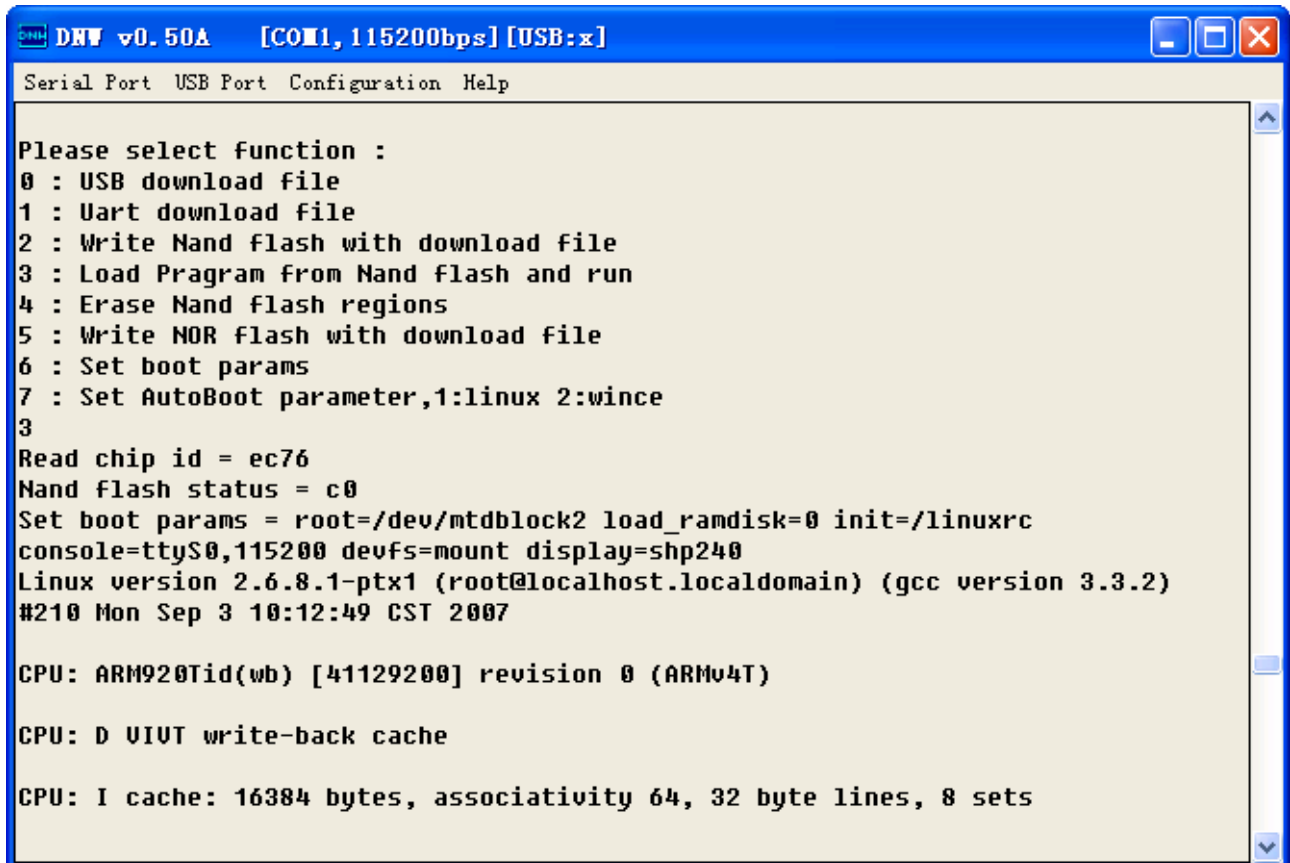
在出现上面的提示后，输入“2”，在接下来的提示输入“Y”，将根文件系统烧写到 NAND FLASH 的分区 2 中（注意，根文件系统的大小超过了该分区大小，不用担心，会自动夸区烧写，不用理会）。烧写成功后，会自动进入主功能菜单。

3.3 启动 LINUX

将 Linux 内核和 Linux 根文件系统烧写好之后，接下来就是启动 Linux 了，启动 Linux 有两种方式，一种是通过 BIOS 的 3 号功能来启动，另一种是设置 Linux 自启动。

3.3.1 通过 BIOS 的 3 号功能启动 LINUX

开发板进入 BIOS 主功能菜单后，选择“3”，这时将启动 Linux, 启动界面如下图。



```
DNW v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
3
Read chip id = ec76
Nand flash status = c0
Set boot params = root=/dev/mtdblock2 load_ramdisk=0 init=/linuxrc
console=ttyS0,115200 devfs=mount display=shp240
Linux version 2.6.8.1-ptx1 (root@localhost.localdomain) (gcc version 3.3.2)
#210 Mon Sep 3 10:12:49 CST 2007

CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)

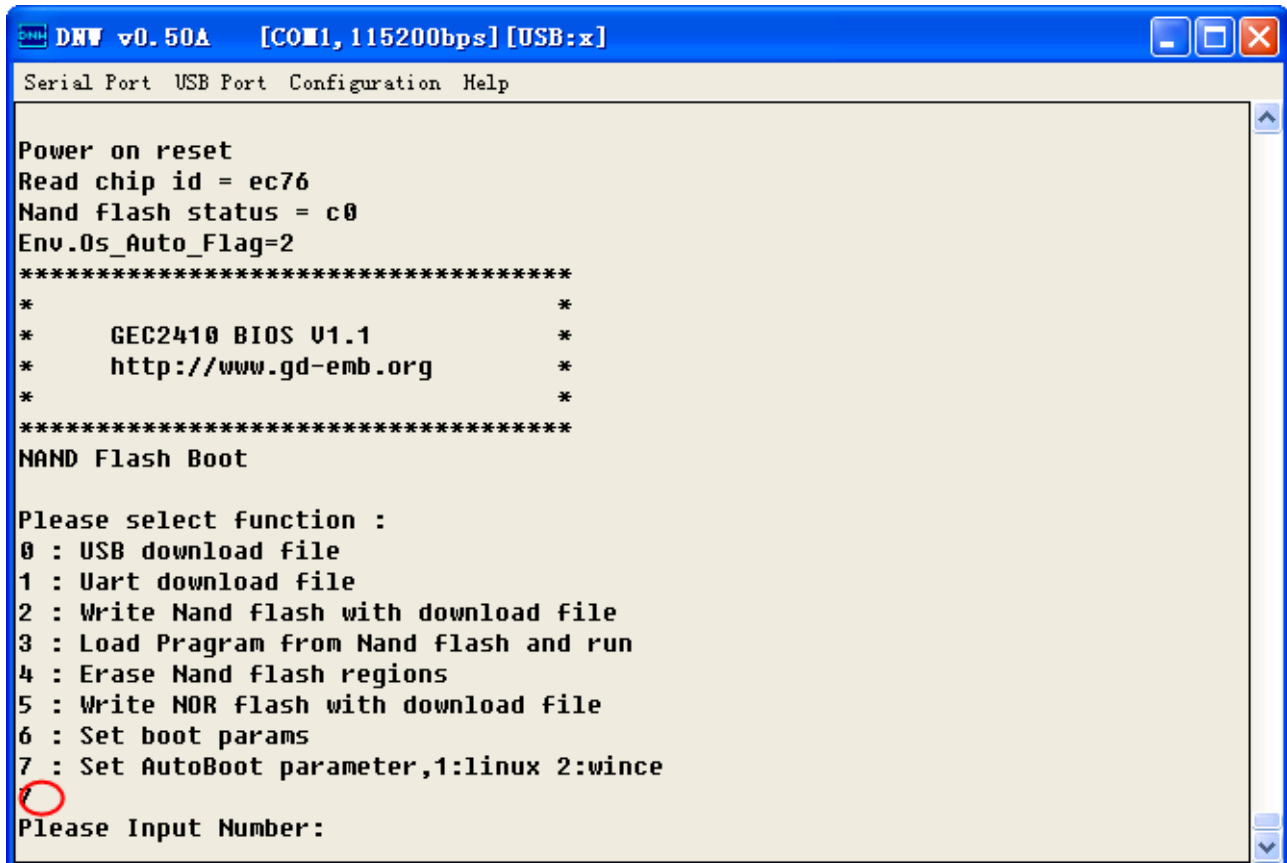
CPU: D VIWT write-back cache

CPU: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
```

该 Linux 的根文件系统是带 QT 的界面，在 LCD 屏处，需要进行触摸屏校准才能进入 QT 界面，在串口工具（DNW 或超级终端）处，按“Enter”键，将进入 Linux 的命令行模式下。

3.3.2 LINUX 的自启动

进入 BIOS 的主功能菜单后，选择“7”，提示界面如下图。

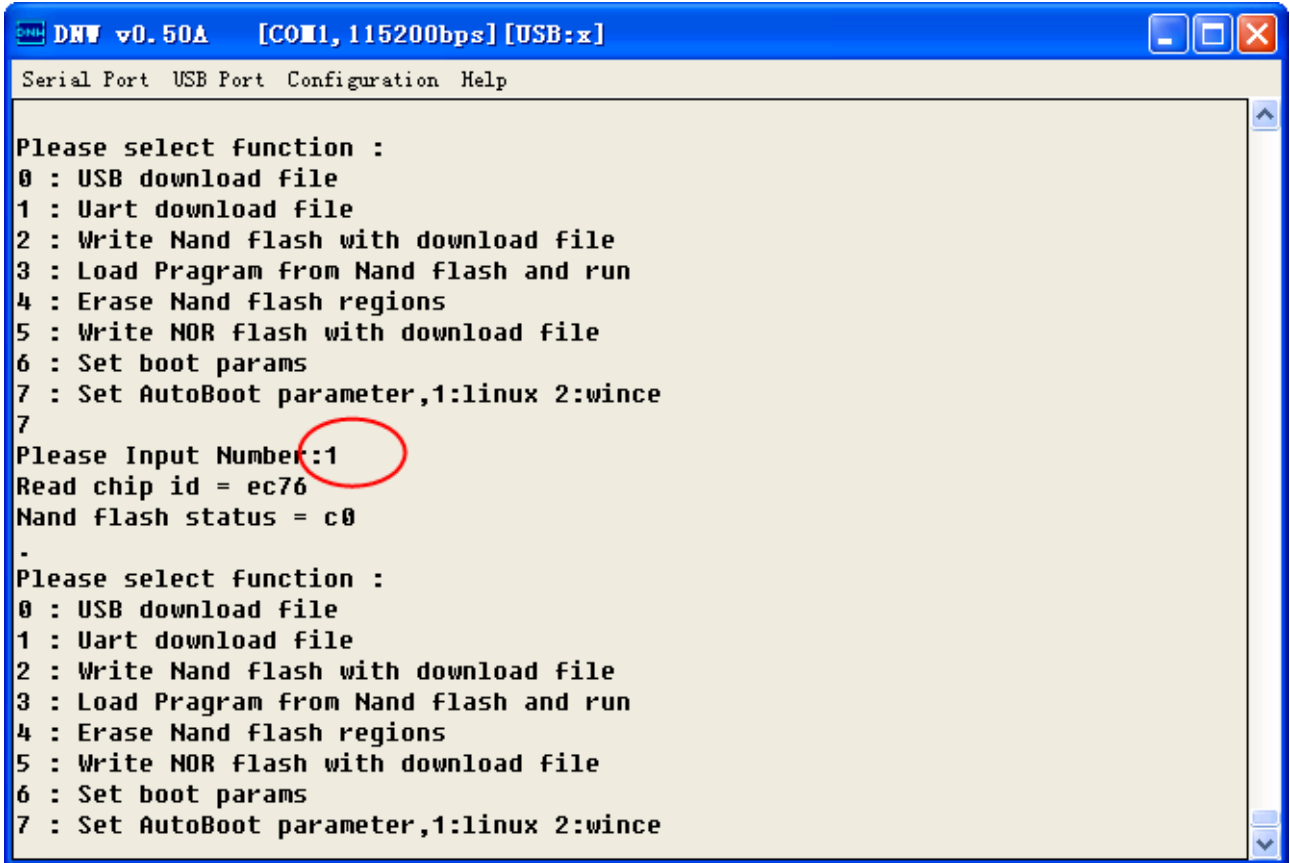


```
DNV v0.50A [COM1, 115200bps] [USB:x]
Serial Port  USB Port  Configuration  Help

Power on reset
Read chip id = ec76
Nand flash status = c0
Env.Os_Auto_Flag=2
*****
*                                     *
*   GEC2410 BIOS V1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
1
Please Input Number:
```

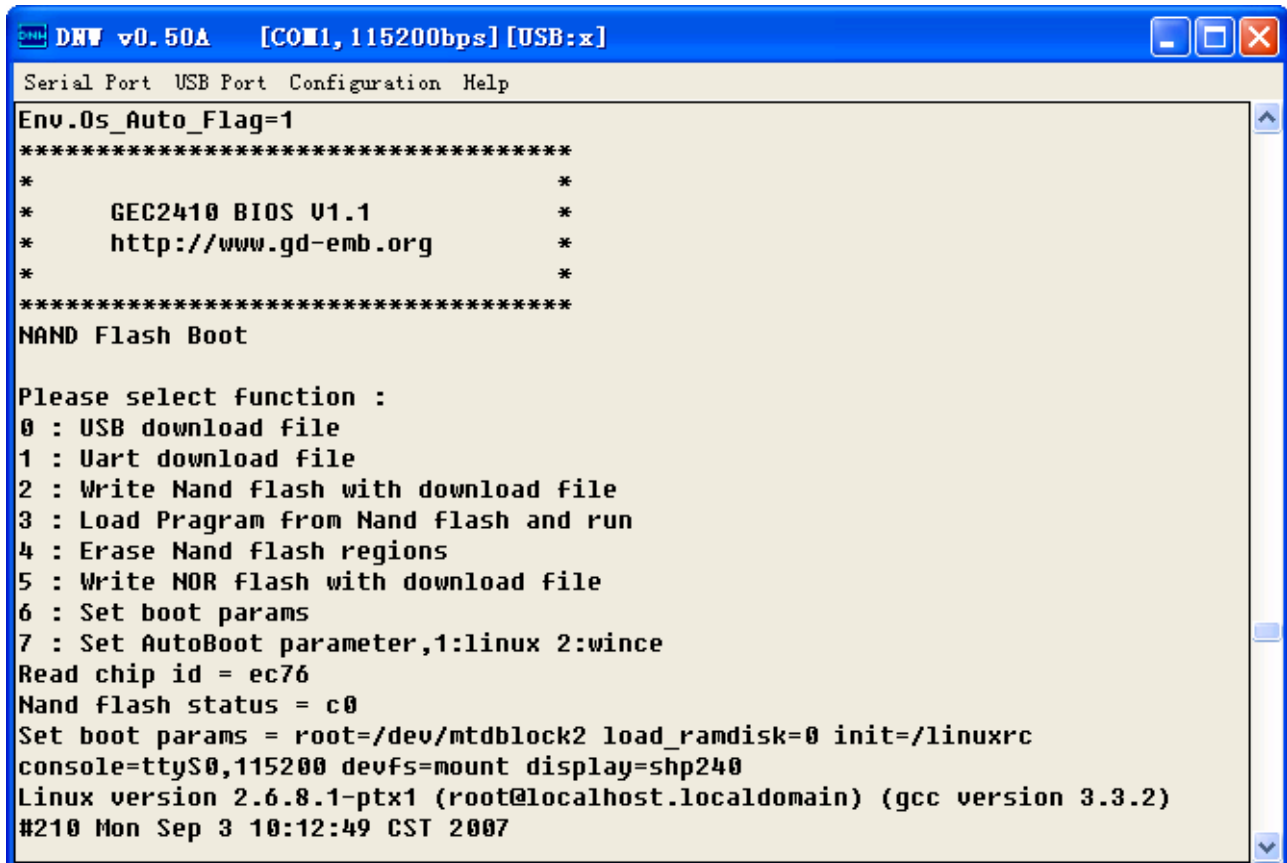
接着输入“1”，这时将会将这个参数保存起来。



```
DNV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
7
Please Input Number:1
Read chip id = ec76
Nand flash status = c0
.
Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
```

上电复位，在 5S 内，不进行任何操作，将自动启动 Linux,关于 Linux 的操作与上一节的 Linux 操作一样。



```
DNV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
Env.Os_Auto_Flag=1
*****
*                                     *
*   GEC2410 BIOS V1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
Read chip id = ec76
Nand flash status = c0
Set boot params = root=/dev/mtdblock2 load_ramdisk=0 init=/linuxrc
console=ttyS0,115200 devfs=mount display=shp240
Linux version 2.6.8.1-ptx1 (root@localhost.localdomain) (gcc version 3.3.2)
#210 Mon Sep 3 10:12:49 CST 2007
```

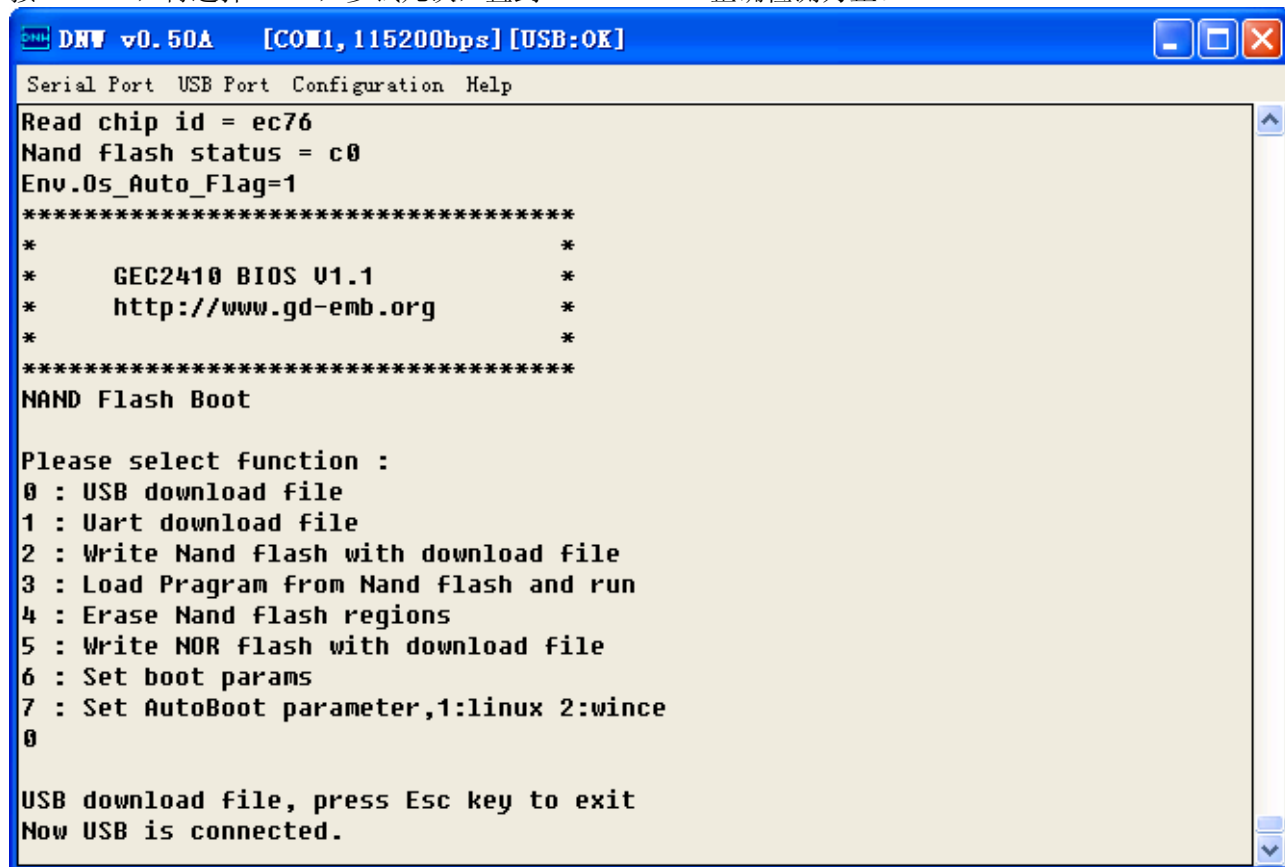
注意: Env.Os_Auto_Flag=1 说明将自启动 Linux
Env.Os_Auto_Flag=2 说明将自启动 Wince (暂不能用)
Env.Os_Auto_Flag=其他, 说明没有设置自启动操作系统

第四章 烧写 WINCE 和启动 WINCE

4.1 下载运行 WINCE

这一章主要来介绍 WINCE 映像 RAM 中运行。

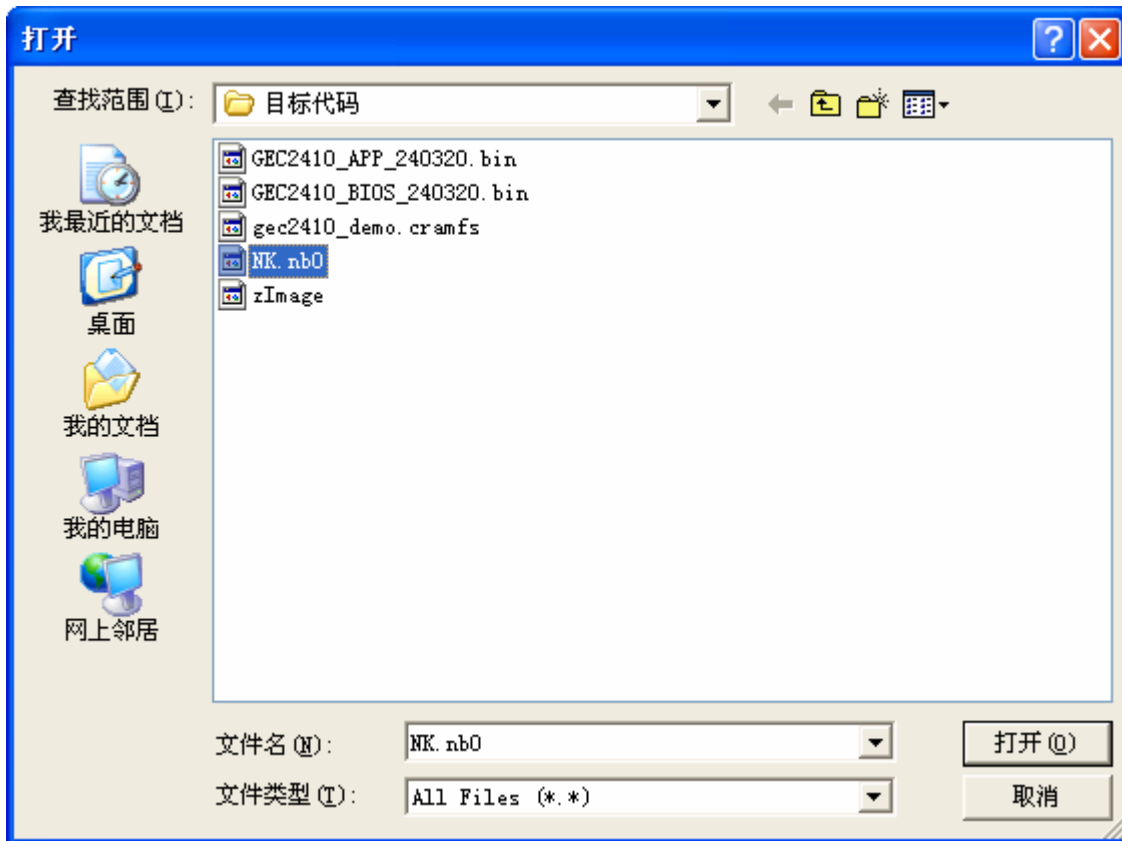
(1) 启动 BIOS，进入 BIOS 主功能菜单后，选择“0”，确定 USB 连接好，如果没有正确连接，请按”ESC“，再选择“0”，多试几次，直到 USB DEVICE 正确检测为止。



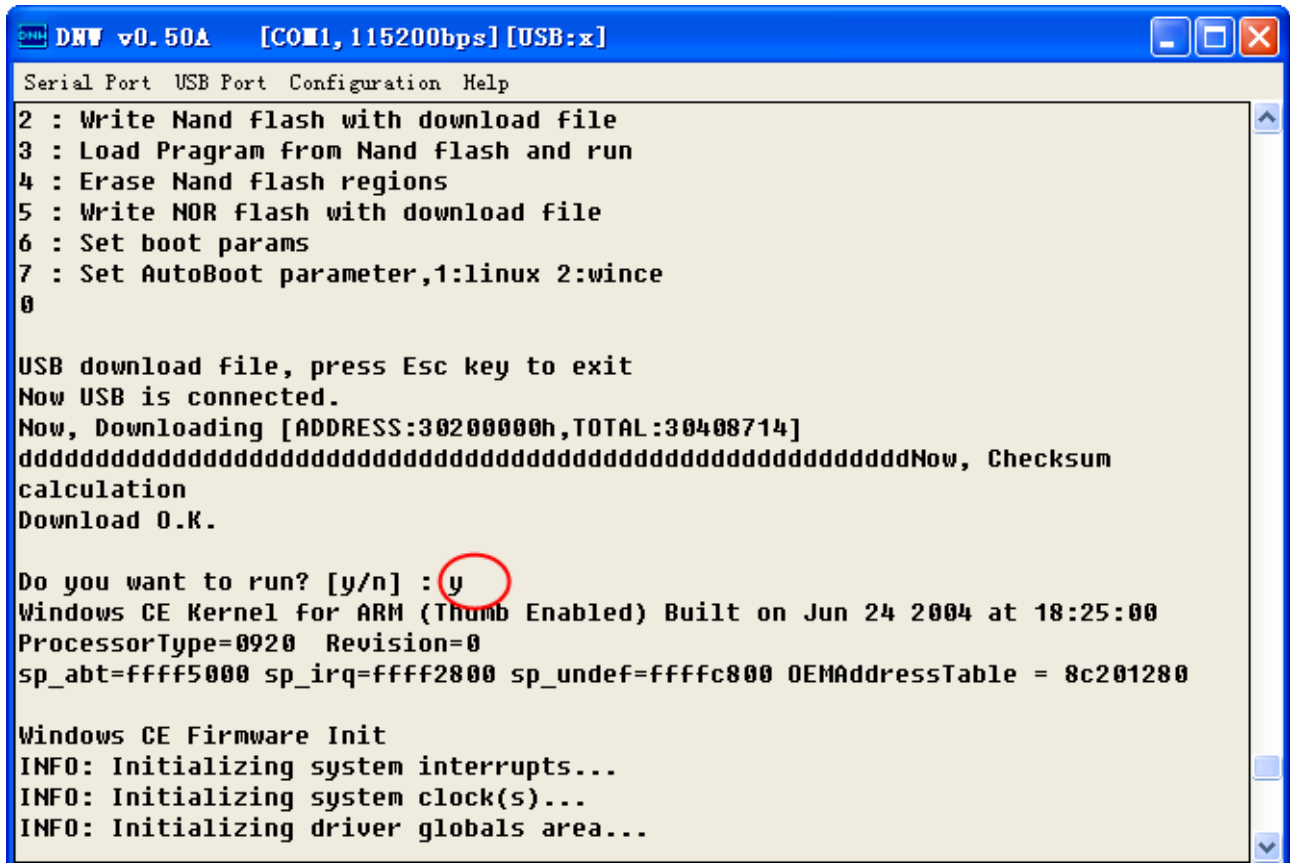
出现“Now USB is connected.”后，说明正确检测到了 USB，接下来就可以进行传输了。

(2) 设置 USB 下载地址为 0x30200000。

(3) 点击“USB Port” ”Transmit”选项，选择 NK.nb0（这个映像文件在光盘目录的”目标代码“文件夹下）



(4) 下载结束后，会提示是否要立即运行，这时输入 “Y” 这时将运行刚刚下载的 WINCE。



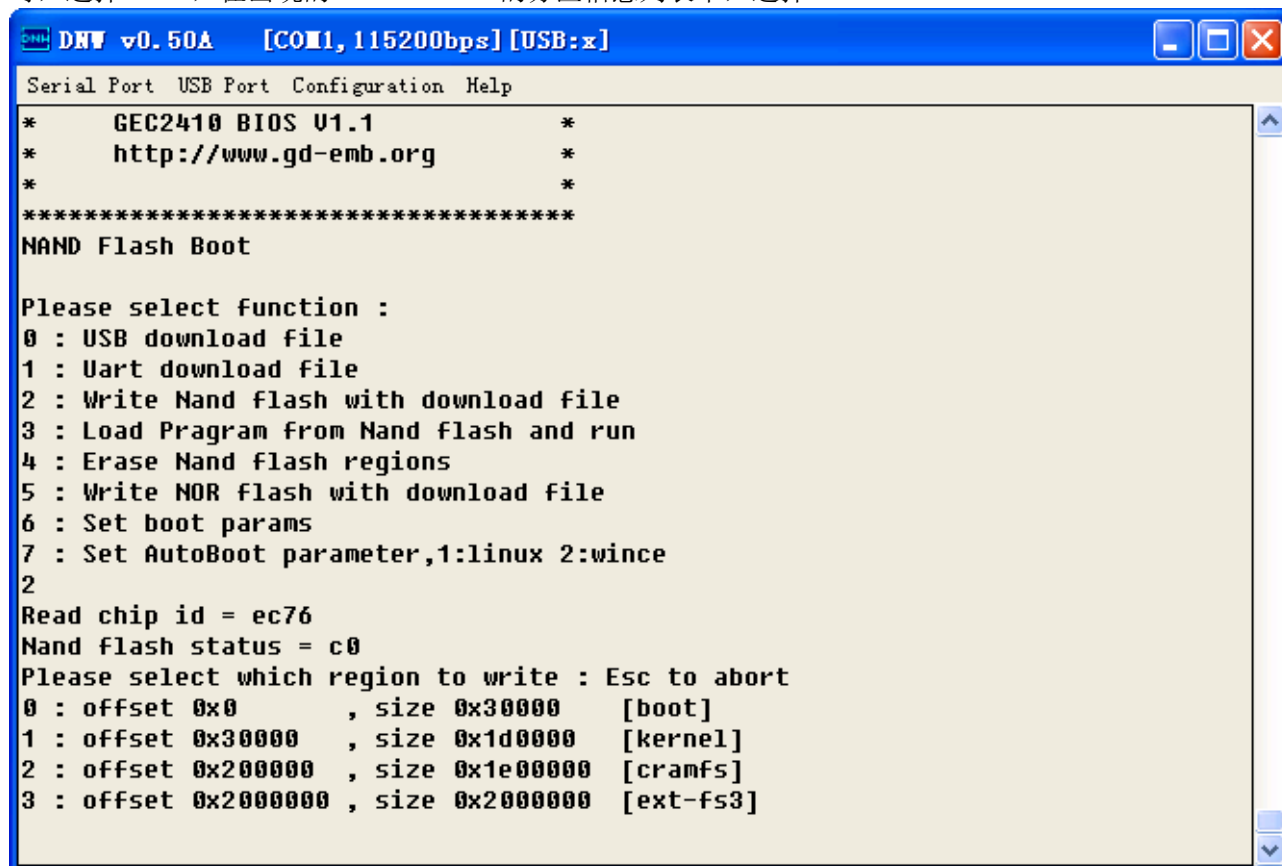
WINCE 启动后，将进入 WINCE 界面了。

4.2 烧写 WINCE

这一章节主要用来烧写 WINCE，将 WINCE 烧写到 NAND FLASH 的分区 3 中，这个分区的大小为 32M。

(1) 下载 WINCE 的 NK.nb0 的步骤与《下载运行 WINCE》节的 (1) 和 (2) 的步骤相同。

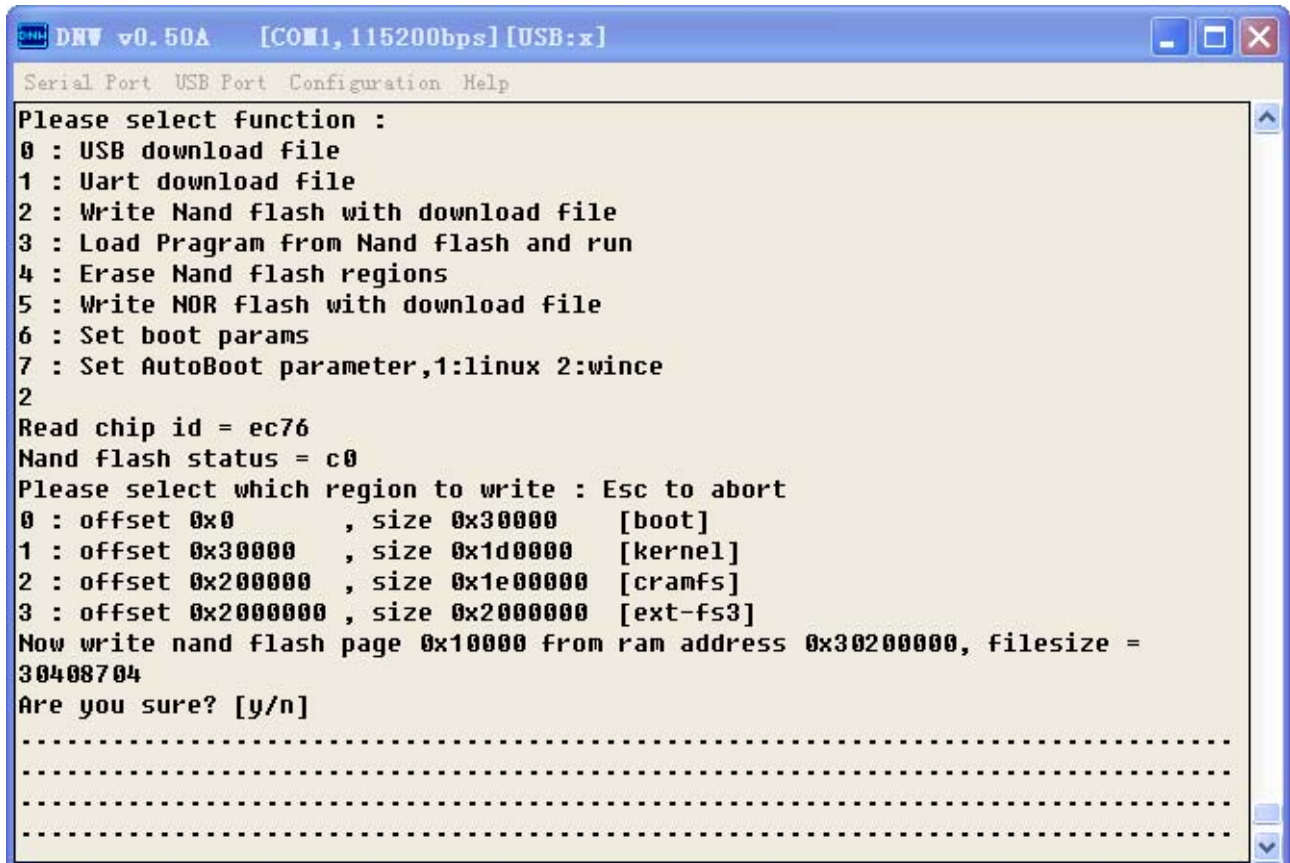
(2) NK.nb0 下载结束后，会提示是否要立即运行，这时输入“n”，将进入 BIOS 的主功能菜单，这时，选择“2”，在出现的 NAND FLASH 的分区信息列表中，选择“3”。



```
Serial Port USB Port Configuration Help
*      GEC2410 BIOS U1.1      *
*      http://www.gd-emb.org  *
*                               *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
2
Read chip id = ec76
Nand flash status = c0
Please select which region to write : Esc to abort
0 : offset 0x0      , size 0x30000  [boot]
1 : offset 0x30000  , size 0x1d0000 [kernel]
2 : offset 0x200000 , size 0x1e0000 [cramfs]
3 : offset 0x200000 , size 0x200000 [ext-fs3]
```

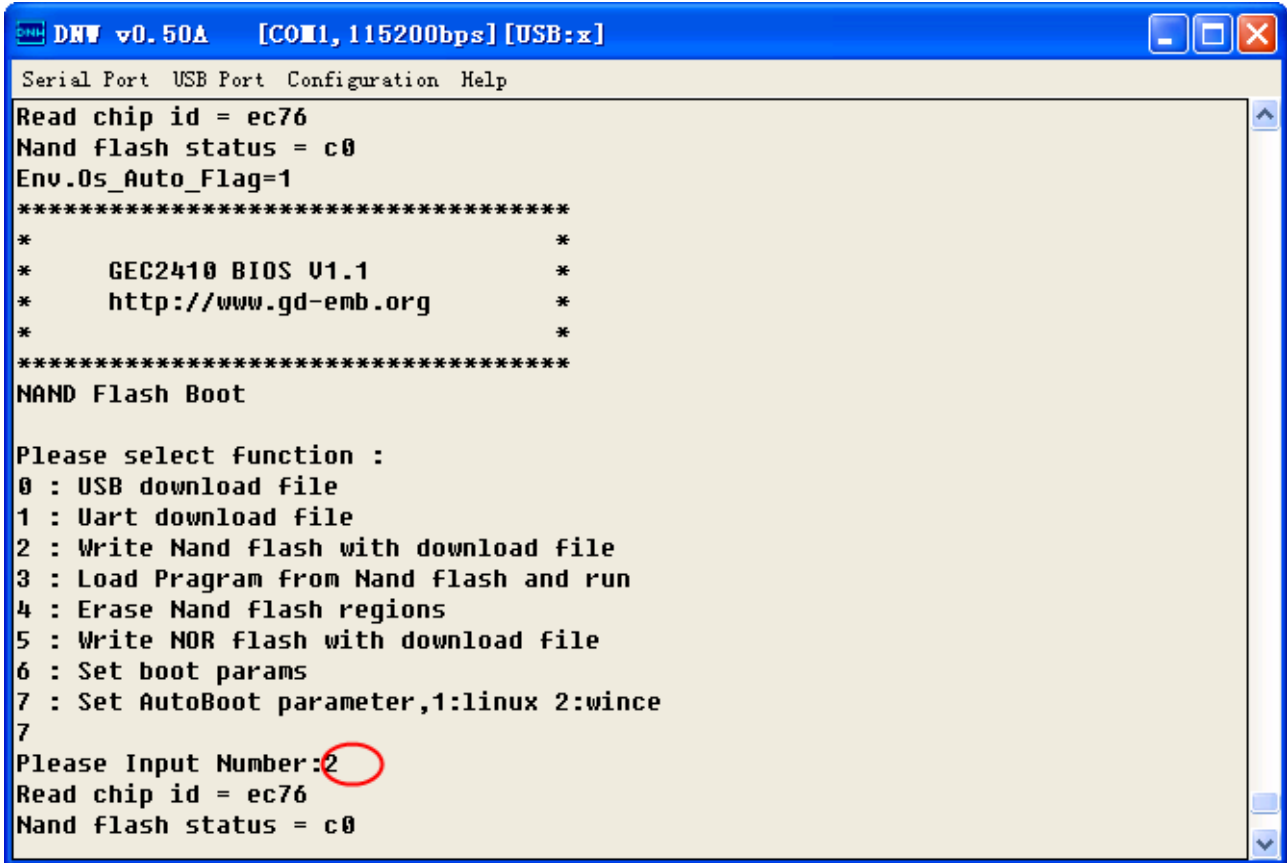
接着会提示是否要烧写，这时输入“y”，这样就将 NK.nb0 烧写到 NAND FLASH 的分区 3 中。



这样就将 WINCE 烧写到 NAND FLASH 分区 5 了。

4.3 自启动 WINCE

进入 BIOS 的主功能菜单后，选择“7”，提示界面如下图。

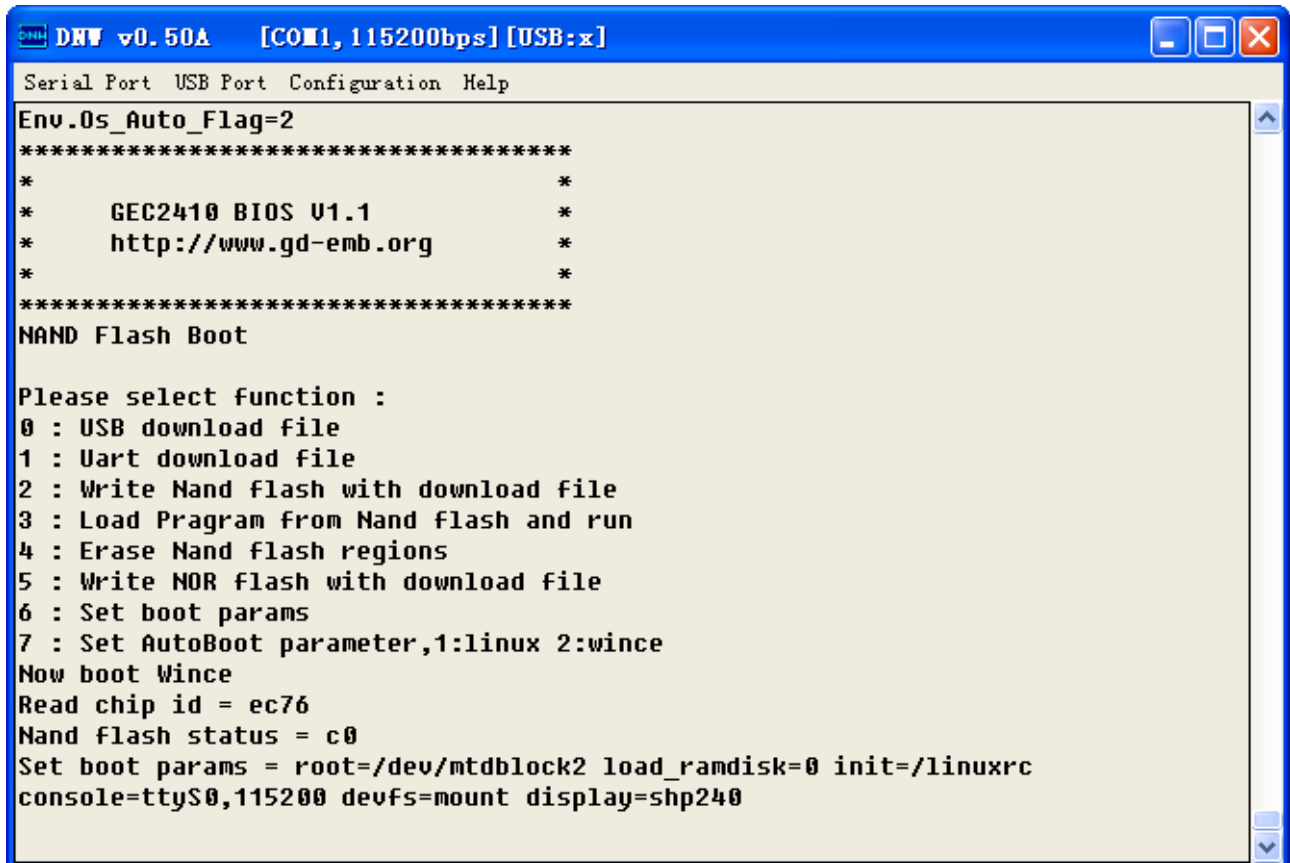


```
DNV v0.50A [COM1, 115200bps] [USB:x]
Serial Port USB Port Configuration Help
Read chip id = ec76
Nand flash status = c0
Env.Os_Auto_Flag=1
*****
*                                     *
*   GEC2410 BIOS V1.1               *
*   http://www.gd-emb.org           *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter, 1:linux 2:wince
7
Please Input Number: 2
Read chip id = ec76
Nand flash status = c0
```

输入“2”，这时会将这个参数保存起来。

注意：WINCE 的装载要一段时间。



```
DWV v0.50A [COM1,115200bps] [USB:x]
Serial Port USB Port Configuration Help
Env.Os_Auto_Flag=2
*****
*                                     *
*      GEC2410 BIOS V1.1             *
*      http://www.gd-emb.org         *
*                                     *
*****
NAND Flash Boot

Please select function :
0 : USB download file
1 : Uart download file
2 : Write Nand flash with download file
3 : Load Program from Nand flash and run
4 : Erase Nand flash regions
5 : Write NOR flash with download file
6 : Set boot params
7 : Set AutoBoot parameter,1:linux 2:wince
Now boot Wince
Read chip id = ec76
Nand flash status = c0
Set boot params = root=/dev/mtdblock2 load_ramdisk=0 init=/linuxrc
console=ttyS0,115200 devfs=mount display=shp240
```

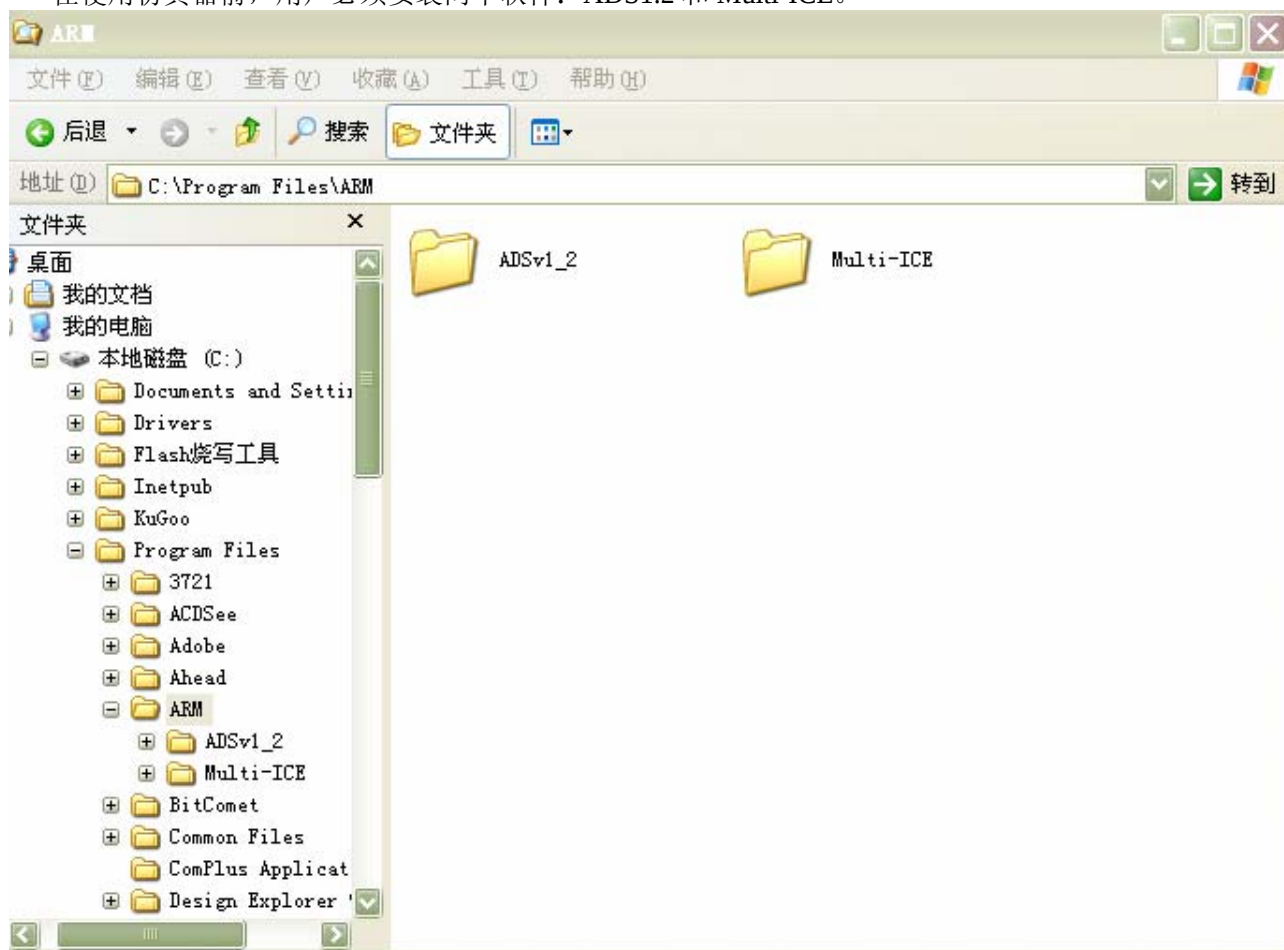
注意: Env.Os_Auto_Flag=1 说明将自启动 Linux
Env.Os_Auto_Flag=2 说明将自启动 Wince
Env.Os_Auto_Flag=其他, 说明没有设置自启动操作系统

第五章 用 ARM 仿真器调试 GEC2410

GEC2410v1.1 开发板套件没有 ARM 仿真器，用户需要另外购买 ARM 仿真器，Multi-ICE 仿真器是比较常用的 ARM 仿真器。以下的说明是针对 ARM 仿真器的，没有 ARM 仿真器的用户可以跳过本章。

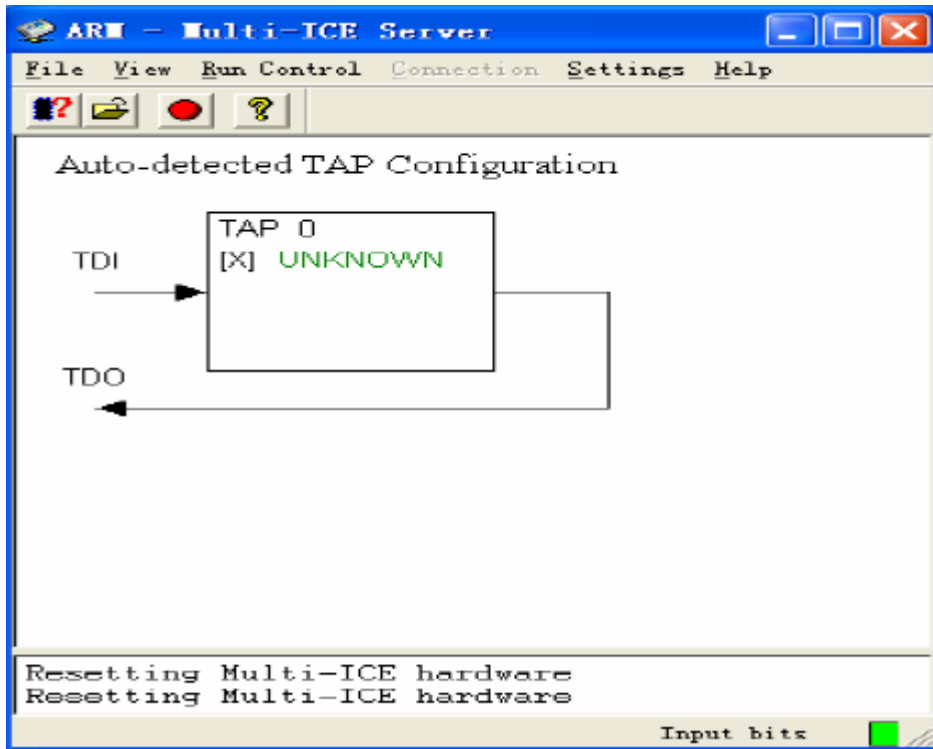
本章节工作平台：2000/xp

在使用仿真器前，用户必须安装两个软件：ADS1.2 和 Multi-ICE。

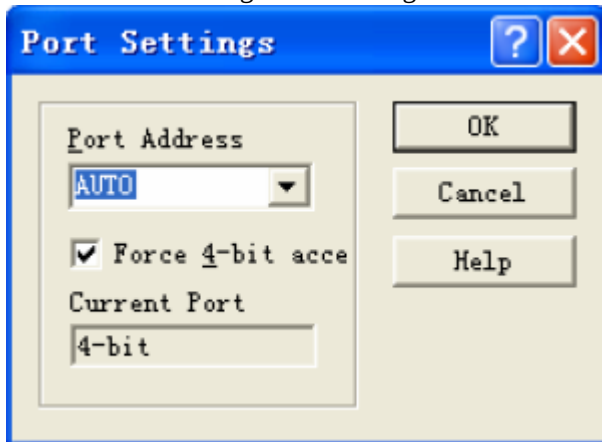


在使用仿真器之前，首先把提供的仿真器配置文件 ARM920t.cfg 拷贝到 Multi-ICE 的安装目录 Program Files\ARM\Multi-ICE 下面，通常配置文件由仿真器厂商提供。

把开发板用 20 芯排线连接到 ARM 仿真器上面，把仿真器用并口线连接到 PC 机并口，然后给开发板上电，再运行 Milti-ICE server 软件，应该可以看到下面的画面：



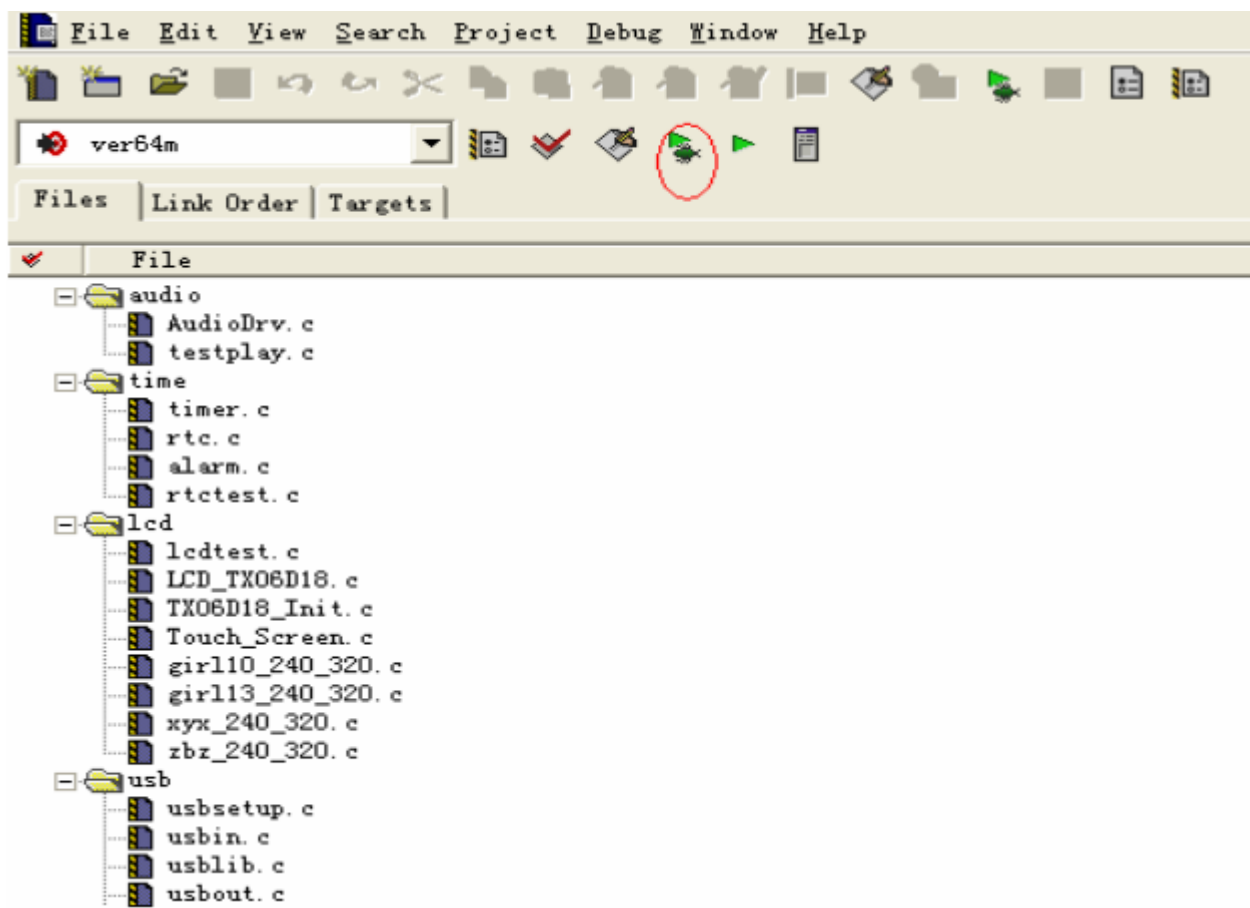
点击菜单 Settings/Ports Setting 按照下面图片进行设置：



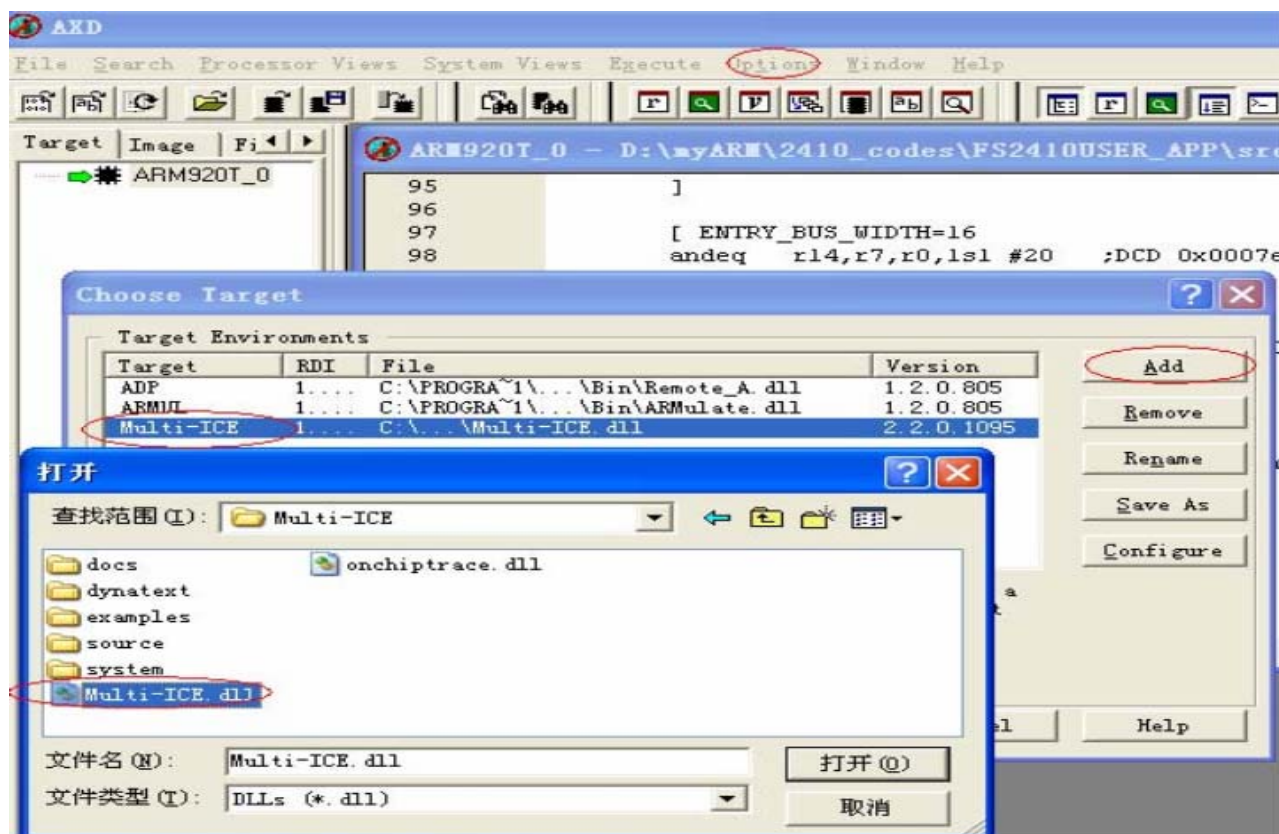
然后点击菜单 File/Load Configuration 打开刚才拷贝到 Program Files\ARM\Multi-ICE 目录下面的配置文件 ARM920t.cfg:

从上图可以看到已经识别出 ARM 核的信号，是 ARM920T 的内核。

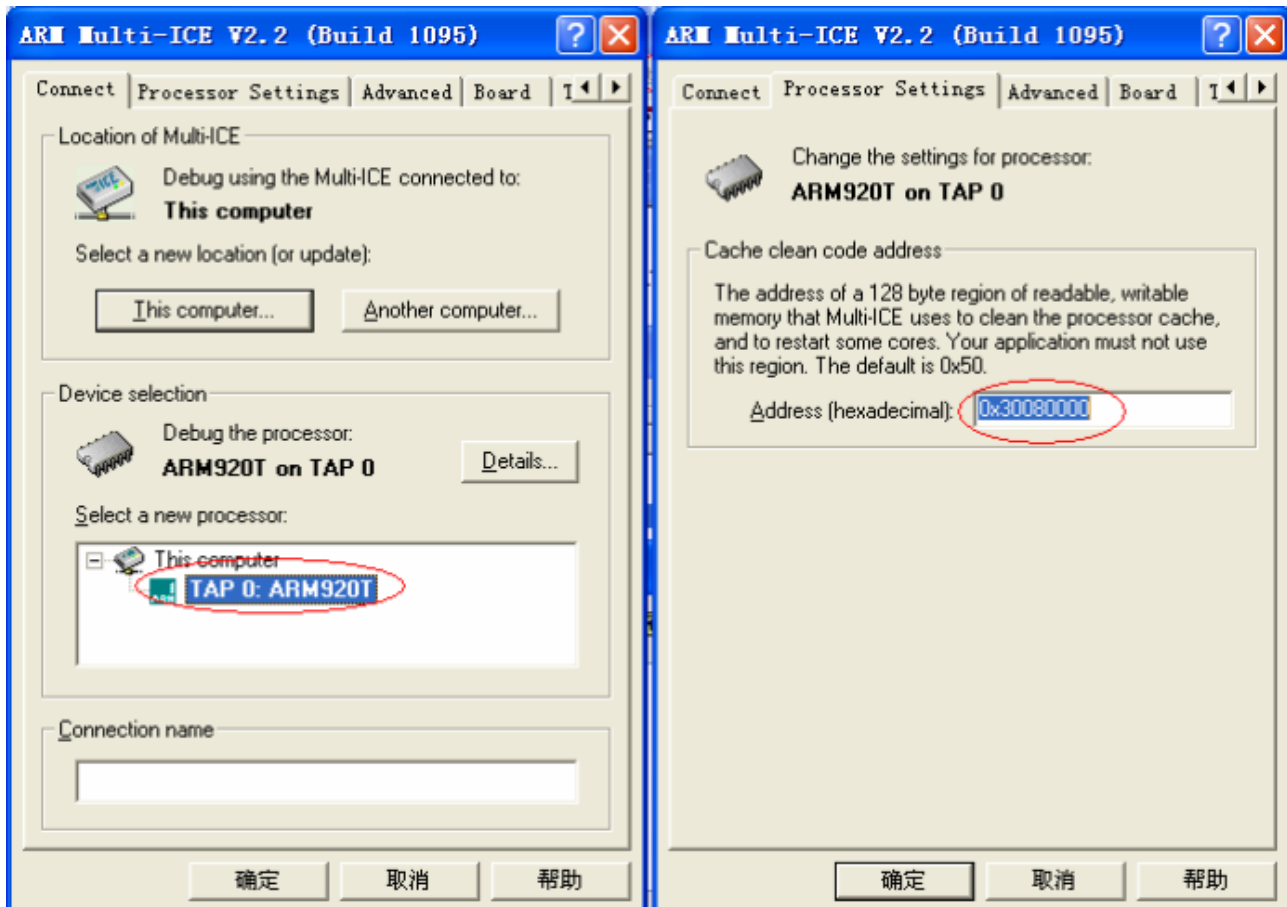
运行 ADS1.20，打开一个 S3C2410 的 Project，点击下图中画红色圆圈的图标进行 Debug:



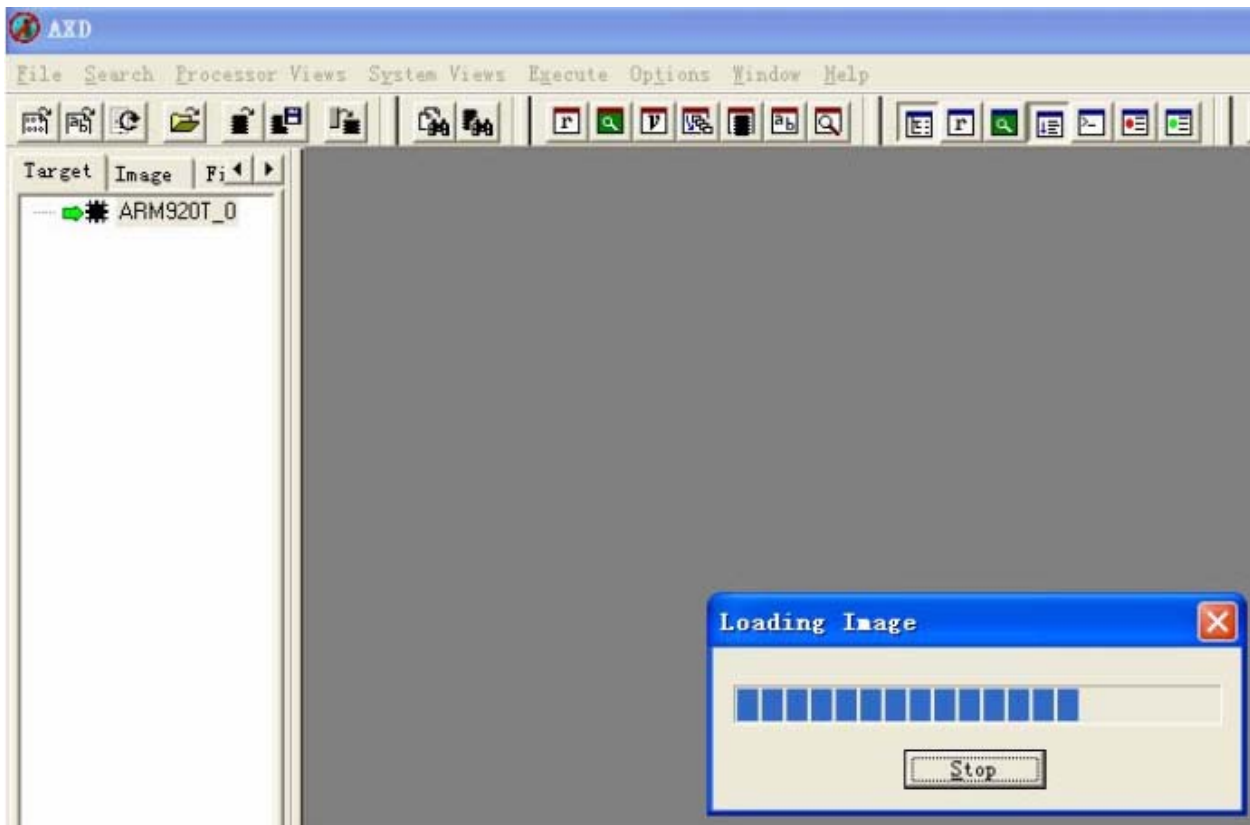
在弹出的 AXD 调试窗口里面点击菜单 Options/Configure Target，在弹出的菜单里面点击 add，添加 Multi-ICE 的驱动：



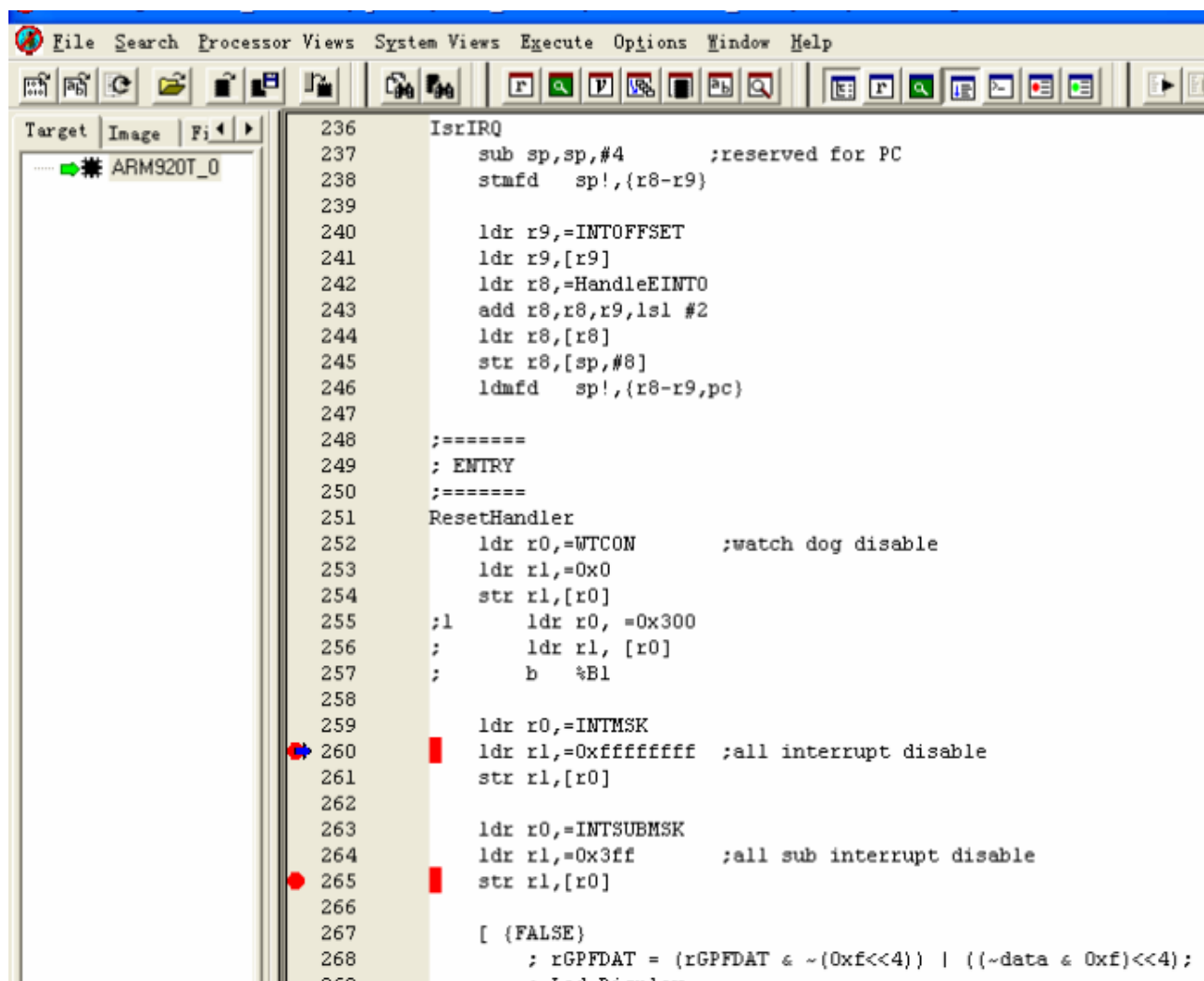
点击上图中右下角的 **Configure** 按钮，在弹出的窗口中按照下面的图片进行设置，注意 **Address** 设置为 **0x30080000**：



按确定，然后关闭 AXD.exe，然后在 ADS1.20 里面点击 Debug 图片重新启动 AXD.exe，就可以看到 Load Image 的进度条了：



Load 完毕可以看到源码，现在就可以进行单步或全速 Debug 了：



```

236 IsrIRQ
237     sub sp,sp,#4           ;reserved for PC
238     stmfid sp!,{r8-r9}
239
240     ldr r9,=INTOFFSET
241     ldr r9,[r9]
242     ldr r8,=HandleEINT0
243     add r8,r8,r9,lsr #2
244     ldr r8,[r8]
245     str r8,[sp,#8]
246     ldmfd sp!,{r8-r9,pc}
247
248     ;=====
249     ; ENTRY
250     ;=====
251 ResetHandler
252     ldr r0,=WTCON          ;watch dog disable
253     ldr r1,=0x0
254     str r1,[r0]
255     ;l     ldr r0, =0x300
256     ;     ldr r1, [r0]
257     ;     b    %B1
258
259     ldr r0,=INTMSK
260     ldr r1,=0xffffffff     ;all interrupt disable
261     str r1,[r0]
262
263     ldr r0,=INTSUBMSK
264     ldr r1,=0x3ff          ;all sub interrupt disable
265     str r1,[r0]
266
267     [ {FALSE}
268         ; rGPFDAT = (rGPFDAT & ~(0xf<<4)) | ((~data & 0xf)<<4);
269         ; Led Display

```

第六章 嵌入式 linux

6.1 安装工具链

假定宿主机的操作系统为 redhat 9.0。编译嵌入式 linux 内核前，要先安装交叉编译工具 toolchain，随板光盘已附带两个编译工具。一个是 cross-3.3.2.tar.bz2，安装此编译器只需要在/usr/local 目录下建一个 arm 的目录，进入 arm 目录再执行解包命令 tar xjvf /mnt/cdrom/linux/toolchain/cross-3.3.2.tar.bz2 即可，之后可编辑/etc/bashrc 文件，在最后增加路径 export PATH=/usr/local/arm/3.3.2/bin:\$PATH，以后编译内核或其他应用程序均可用 arm-linux- 来指定交叉编译器。

安装工具链 cross-3.3.2.tar.bz2

- 1 在/usr/local 目录下建一个 arm 目录

```
mkdir /usr/local/arm
```

- 2 将 cross-3.3.2.tar.bz2 拷贝到上面建立的目录下

```
cp cross-3.3.2.tar.bz2 /usr/local/arm
```

- 3 解压 cross-3.3.2.tar.bz2

```
tar xzvf cross-3.3.2.tar.bz2
```

这样会生成一个 3.3.2 目录。

- 4 添加环境变量

在文件/etc/bashrc 文件最后添加：

```
export PATH=/usr/local/arm/3.3.2/bin:$PATH
```

或者

修改/etc/profile 文件

找到：pathmunge /usr/local/sbin

在其下面添加一行

```
Pathmunge /usr/local/arm/3.3.2/bin
```

工具链建立完成。

注意：

添加环境变量，也可以修改/etc/profile文件：

```
pathmunge /sbin
```

```
pathmunge /usr/bin
```

```
pathmunge /usr/local/sbin
```

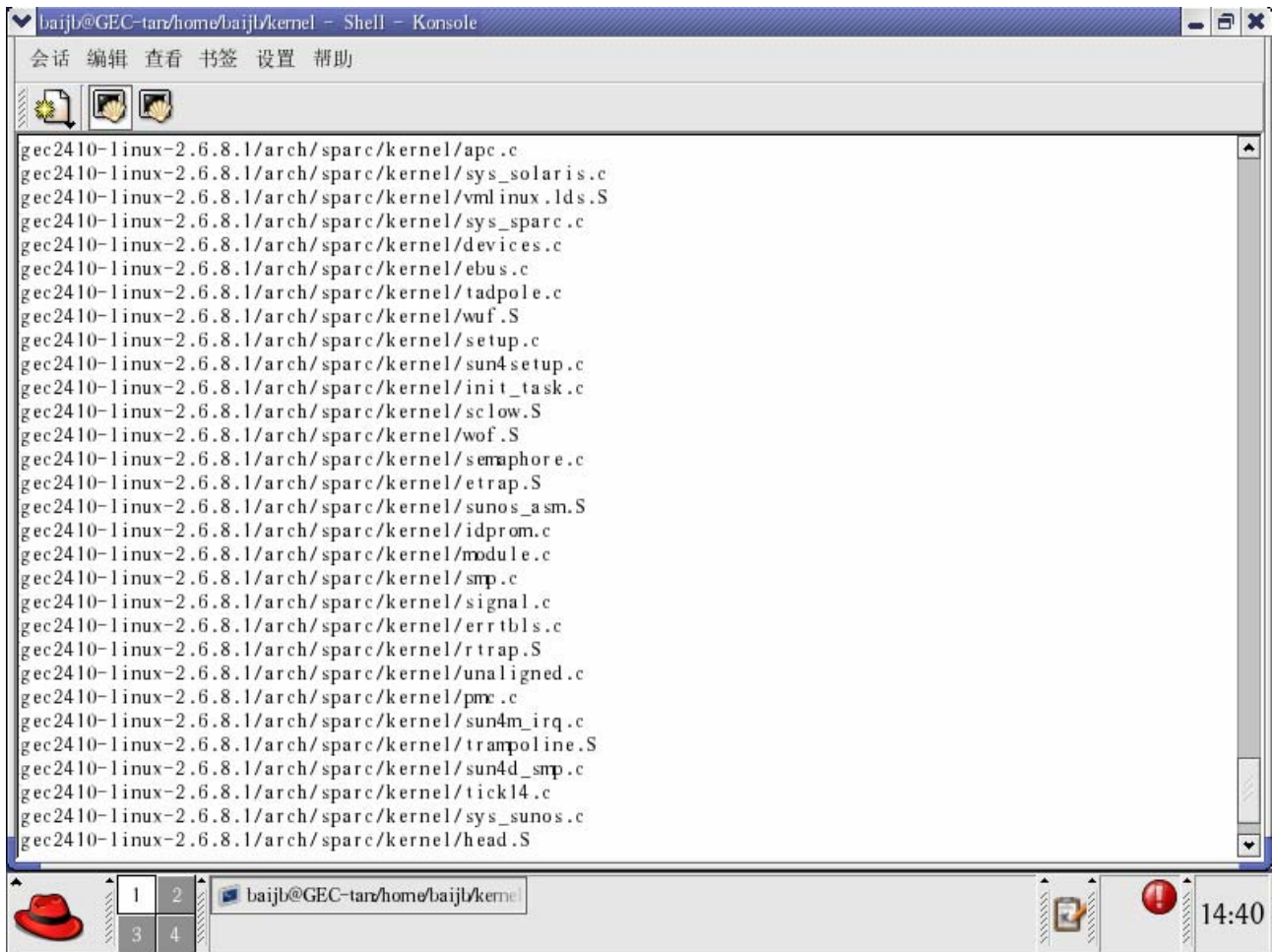
```
pathmunge /usr/local/arm/3.3.2/bin
```

6.2 配置移植 linux 内核

- 1) 解压内核包

将提供的 linux-2.6 内核压缩包解包到您所想要的目录下，输入如下命令：

```
tar xjvf /mnt/cdrom/linux/gec2410-linux-2.6.8.tar.bz2
```



```

baijb@GEC-tar/home/baijb/kernel - Shell - Konsole
会话 编辑 查看 书签 设置 帮助

gec2410-linux-2.6.8.1/arch/sparc/kernel/apc.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/sys_solaris.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/vmlinux.lds.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/sys_sparc.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/devices.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/ebus.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/tadpole.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/wuf.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/setup.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/sun4setup.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/init_task.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/sclow.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/wof.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/semaphore.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/etrap.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/sunos_asm.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/idprom.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/module.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/smp.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/signal.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/errtbls.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/rtrap.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/unaligned.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/pmc.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/sun4m_irq.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/trampoline.S
gec2410-linux-2.6.8.1/arch/sparc/kernel/sun4d_smp.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/tick14.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/sys_sunos.c
gec2410-linux-2.6.8.1/arch/sparc/kernel/head.S
  
```

2) 编辑 Makefile 文件

进入解压的目录后，运行命令：

vi Makefile

找到 CROSS_COMPILE = 这行，将它改为

CROSS_COMPILE = arm-linux-，或者，直接在此行写上/usr/local/arm/3.3.2/bin/arm-linux-

```

baijb@GEC-tan/usr/local/src/ge2410-linux-2.6.8.1 - Shell - Konsole
会话 编辑 查看 书签 设置 帮助

# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile

# ARCH          ?= $(SUBARCH)
ARCH            ?= arm
#CROSS_COMPILE  ?= arm-software-linux-gnu-
CROSS_COMPILE   ?= /usr/local/arm/3.3.2/bin/arm-linux-

# Architecture as present in compile.h
UTS_MACHINE := $(ARCH)

# SHELL used by kbuild
CONFIG_SHELL := $(shell if [ -x "$$BASH" ]; then echo $$BASH; \
    else if [ -x /bin/bash ]; then echo /bin/bash; \
    else echo sh; fi ; fi)

HOSTCC          = gcc
HOSTCXX         = g++
HOSTCFLAGS      = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
HOSTCXXFLAGS    = -O2

#       Decide whether to build built-in, modular, or both.
#       Normally, just do built-in.

KBUILD_MODULES :=
KBUILD_BUILTIN := 1

#       If we have only "make modules", don't compile built-in objects.
#       When we're building modules with modversions, we need to consider
"Makefile" [已转换] 1197L, 38235C
200,1 14%

```

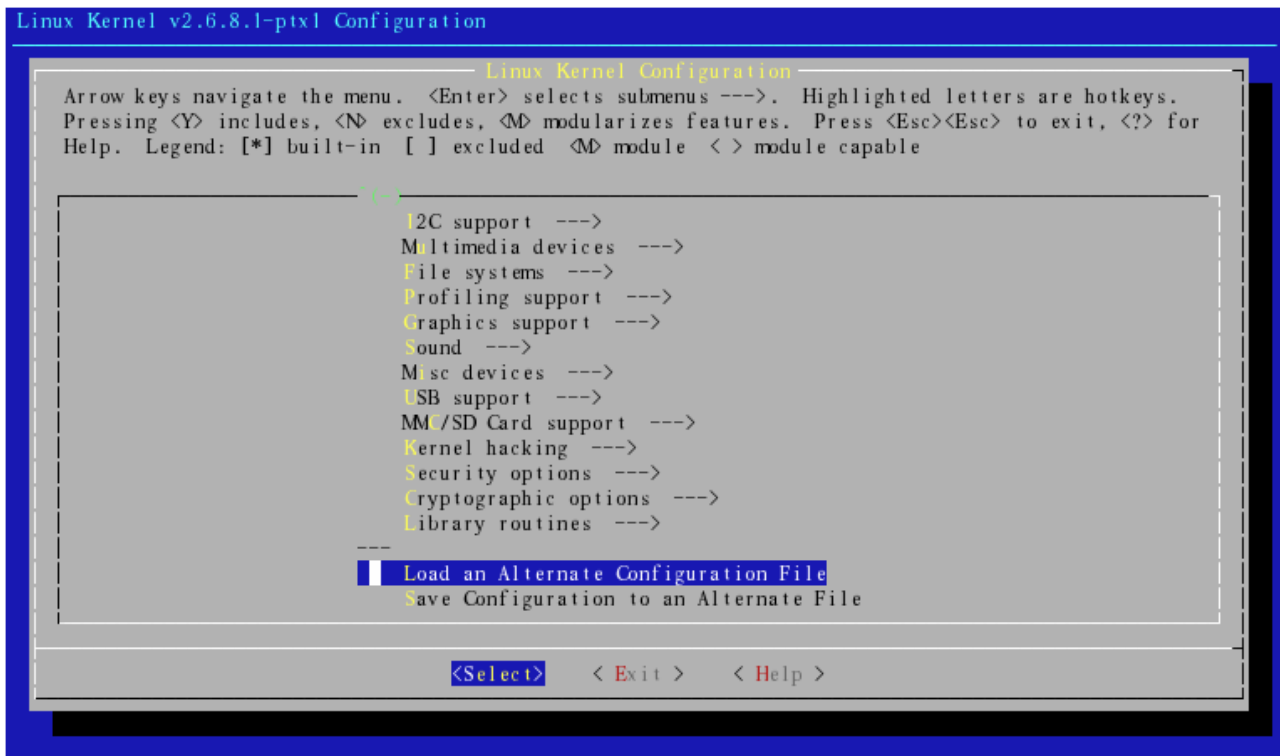
设置好后编译器再保存退出

3) 配置内核

输入命令: **make menuconfig**

进入配置栏的“Load an Alternate Configuration File”

输入配置文件名 **GEC2410.cfg** 并回车, 在主菜单里选择 **Exit** 退出并保存设置。



5)接着输入命令 make

6)编译结束后, 可在 arch/arm/boot 目录下得到 linux 内核压缩映像 zImage。

6.3 制做根文件系统

Cramfs 根文件系统的制作

将程序和模块添加到根文件系统中, 以 gec2410 中的 gec2410_demo.cramfs 根文件系统的添加为例。

1 将 gec2410_demo.cramfs 拷贝到任意目录下

```
cp gec2410_demo.cramfs /home
```

2 在该目录下建立两个文件:

```
mkdir chang
```

```
mkdir guo
```

3 将 Gec2410_demo.cramfs 挂接到 chang 目录下

```
mount -t cramfs -o loop Gec2410_demo.cramfs chang
```

4 将 chang 目录下的内容压缩

```
cd chang
```

```
tar -cvf chang.tar *
```

这样将产生一个 chang.tar 包.

然后将隐藏的文件要拷贝过来: ls -a 显示所有的文件, cp .bashrc ../guo/ 将.bashrc 文件拷贝到 guo 目录下。

5 将包解压到 guo 目录下

```
umount chang ; 卸载挂接
```

```
cd .. ; 进入上一级目录
```

```
mv chang.tar guo
```

```
cd guo
```

```
tar -xvf chang.tar ; 将打包的根文件系统里的内容解压
```

```
rm chang.tar
```

6 将自己的驱动 `gpiodrv.ko` 和应用程序 `gpio_test` 拷贝到相应的目录下。

将 `gpiodrv.ko` 拷贝到 `guo/usr/` 目录下

7 经过上面的步骤就可以将自己的驱动和应用程序添加到 `cramfs` 根文件系统了

现在将开始制作 `cramfs` 根文件系统

先将 `mkcramfs` 文件拷贝到 `guo` 所在的目录

在这个目录下运行命令：

```
mkcramfs guo Gec2410_demo.cramfs
```

运行成功后，会在该目录下生成 `Gec2410_demo.cramfs` 根文件系统

8 根文件系统制做成功后，就可以将它烧写到相应的地方。

7.4 linux 下的设备驱动程序

1、LCD 驱动：`drivers/video/s3c2410fb.c`

2、触摸屏驱动：`drivers/char/s3c2410ts-adc.c`

3、AD 驱动：`drivers/char/s3c2410ts-adc.c`

3、音频驱动：`sound/oss/s3c2410-oss.c`

4、SD 卡驱动：`drivers/mmc/s3c2410_sdi.c`

5、NAND FLASH 驱动：`drivers/mtd/nand/s3c2410_nand.c`

6、USB HOST 驱动：`drivers/usb/host/ohci-s3c2410.c`

7、RTC 驱动：`drivers/char/s3c2410-rtc.c`

8、网卡驱动：`drivers/net/cirrus.c`

9、串口驱动：`drivers/serial/s3c2410.c`

第七章 嵌入式 GUI--Qt 开发

7.1 Qt/Embedded 开发环境的安装

准备以下源码包和文件(在 GEC2410 光碟的 Linux 源码包/gui 目录下找到)

```
tmake-1.11.tar.gz
qt-embedded-2.3.7.tar.gz
qt-x11-2.3.2.tar.gz
qconfig-local.h
qtopia-free-1.7.0.tar.bz2
```

假设源码包放在/home/build_qte/目录下

```
#cd /home/build_qte/
```

7.2 Qt/Embedded开发工具的安装

7.2.1 tmake的安装

解压 tmake 源码包

```
#tar zxvf tmake-1.11.tar.gz
```

进入解压目录

```
#cd tmake-1.11
```

建立 shell 文件, 在文件中添加如下内容

```
#vi env.sh
export TMAKEDIR=$PWD
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-arm-g++
export PATH=$TMAKEDIR/bin:$PATH
```

保存退出.

添加环境变量

```
#source env.sh
```

7.2.2 Qt X11 的安装

你可能需要root 帐号, 这取决于你要安装Qt 的路径的权限。从 TrollTech 公司的官方 ftp 站点获得qt-x11-2.3.2.tar.gz 软件包。解开和解压缩软件包:

```
# tar -vxzf qt-x11-2.3.2.tar.gz
```

进入解开后的文件, 设置一些环境变量

```
# cd qt-2.3.2
```

```
# vi env.sh
```

```
export QTDIR=$PWD
export PATH=$QTDIR/bin:$PATH
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

```
#source env.sh
```

在配置之前, 我们先看一下配置的选项:

```
#!/configure -help
```

主要的配置信息如下:

The defaults (*) are usually acceptable. Here is a short explanation of each option:

- * -release Compile and link Qt with debugging turned off.
- debug Compile and link Qt with debugging turned on.

```

* -shared ..... Create and use a shared Qt library (libqt.so)
  -static ..... Create and use a static Qt library (libqt.a)
* -no-gif ..... Do not compile in GIF reading support.
  -gif ..... Compile in GIF reading support. See src/kernel/qgif.h
  -no-sm ..... Do not support X Session Management.
* -sm ..... Support X Session Management, links in -lSM -lICE.
* -no-thread ..... Do not compile with Threading Support
  -thread ..... Compile with Threading Support
* -qt-zlib ..... Use the zlib bundled with Qt. -system
  -zlib ..... Use a zlib from the operating system
                        http://www.info-zip.org/pub/infozip/zlib
* -qt-libpng ..... Use the libpng bundled with Qt.
  -system-libpng ..... Use a libpng from the operating system. See
                        http://www.libpng.org/pub/png
* -no-mng ..... Do not compile in MNG I/O support.
  -system-libmng ..... Use libmng from the operating system. See
http://www.libmng.com
* -no-jpeg ..... Do not compile in JPEG I/O support.
  -system-jpeg ..... Use jpeglib from the operating system. See
http://www.ijg.org
* -no-nas-sound ..... Do not compile in NAS sound support.
  -system-nas-sound .. Use NAS libaudio from the operating system. see
http://radscan.com/nas.html
  -no-<module> ..... Disables a module, where module can
                        can be one of: opengl table network canvas
* -kde ..... Builds the Qt Designer with KDE 2 support, so that
                        KDE 2 widgets can be used directly in the Qt Designer.
                        $KDEDIR must be set to point to a KDE 2 installation.
                        See http://www.kde.org
  -no-g++-exceptions . Disable exceptions on platforms using the GNU
                        C++ compiler by
                        using the -fno-exceptions flag.
* -no-xft ..... Disable support for Anti-Aliased fonts through the
                        Xft extension library (XFree86 4.0.2 and newer).
* -xft ..... Enable support for Anti-Aliased fonts.
                        Xft support is auto-detected, but you may use these
                        flags to explicitly enable/disable support.
  -platform target ... The platform you are building on (linux-g++)
  -xplatform target .. The platform when cross-compiling.
                        See the PLATFORMS file for a list of supported
operating systems and compilers.
  -Istring ..... Add an explicit include path.
  -Lstring ..... Add an explicit library path.

```

-Rstring Add an explicit dynamic library runtime search path.
-lstring Add an explicit library.

Qt/Embedded only:

-qconfig local Use src/tools/qconfig-local.h rather than the default (qconfig.h).
-depths list Comma-separated list of supported bit-per-pixel depths, from: v, 4, 8, 16, 24, and 32. 'v' is VGA16.
-accel-voodoo3 Enable Voodoo3 acceleration.
-accel-mach64 Enable Mach64 acceleration.
-accel-matrox Enable Matrox MGA acceleration.
-qvfb Enable X11-based Qt Virtual Frame Buffer.
-vnc Enable VNC server (requires network module).

选择我们需要的配置和平台:

#!/configure -no-xft -no-opengl

按照下面的选择进行一步一步, 并且会有以下打印信息:

Type 'Q' to view the Q Public License.

Type 'G' to view the GNU General Public License.

Type 'yes' to accept this license offer.

Type 'no' to decline this license offer.

Do you accept the terms of the license?

yes

This target is using the GNU C++ compiler (linux-g++).
Recent versions of this compiler automatically include
code for exceptions, which increase both the size of
the Qt library and the amount of memory taken by your
applications.

You may elect to re-run ./configure with the
-no-g++-exceptions

option to compile Qt without exceptions. This is
completely binary compatible, and existing
applications will continue to work (like KDE 2).

Build type: linux-g++-shared

Thread support..... yes

GIF support..... yes

MNG support..... no

JPEG support..... yes

OpenGL support yes

NAS sound support..... no

Session management..... yes

Xft support (Anti-Aliased Fonts) .. no

XKB support yes

Creating makefiles...

Qt is now configured for building. Just run make.

To reconfigure, run make clean and configure.

生成Makefile 后，就可以进行安装

#make

安装成功后，将会有以下打印信息：

The Qt library is now built in ./lib

The Qt examples are built in the directories in ./examples

The Qt tutorials are built in the directories in ./tutorial

Note: be sure to set \$QTDIR to point to here or to wherever you move these directories.

Enjoy! – the Trolltech team

编译生成qvfb、qembed工具

#make -C tools/qvfb

#make -C tools/qembed

7.2.3 Qt/Embedded 安装

解开软件包：

#tar zxvf qt-embedded-2.3.7.tar.gz

进入软件包中并设置一些环境变量：

#cd qt-2.3.7

#export PATH=/usr/local/arm/3.3.2/bin:\$PATH

#export QTDIR=\$PWD

#export QTEDIR=\$QTDIR

在 qconfig_local.h 文件中添加一行：

#define QT_QWS_IPAQ_RAW

添加完之后将该文件拷贝到 src/tools/目录

#cp /qconfig-local.h src/tools/

将 qt-2.3.2 源码编译后的 uic 工具拷贝到 bin 目录

#cp qt-2.3.2/bin/uic ./bin

如果是交叉编译，设置编译选项：

#./configure -xplatform linux-arm-g++ -shared -qconfig local -qvfb -depths 4,8,16,32 -gif -system-jpeg

如果不是交叉编译，想在qvfb 上运行的话，那么则去掉“**xplatform linux-arm-g++**”选项即可；

生成Makefile 之后，就可以进行编译了：

#make

注意：这里交叉编译qte源码的时候要用3.3.2版本的交叉工具链，使用2.95.3的工具链编译qte可能会出现错误。

7.2.4 交叉编译 qtopia

如果已经安装好了qt-embedded，就可以进行Qttopia 的安装了。

解压源码包

#tar zxvf qtopia-free-1.7.0.tar.gz

设置环境变量：

#export QTDIR=\$QTEDIR

#export QPEDIR=\$PWD

#export PATH=\$QPEDIR/bin/:\$PATH

接下来就可以配置了：

```
#cd src/
```

```
#!/configure -platform linux-arm-g++
```

如果是在qvfb 上运行的话，则为：

```
#!/configure -platform linux-x86-g++
```

生成Makefile 之后，就可以进行编译了：

```
#make
```

7.3 Qt/Embedded 应用开发实例

7.3.1 在X86上的qvfb运行的qt/embedded应用程序实例

建立 qte 开发的工作目录

```
#mkdir /home/qte
```

```
#cd /home/qte
```

建立 shell 文件, 设置环境变量

```
#vi set-qte-x86.sh
```

在文件中添加如下

```
#!/bin/bash
```

```
export QTDIR=/home/build_qte/qt-2.3.7-x86 ;编译 qt-2.3.7 的目录
```

```
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

```
export TMAKEDIR=/home/build_qte/tmake-1.11 ;tmake-1.11 的目录
```

```
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-x86-g++
```

```
export PATH=$QTDIR/bin:$TMAKEDIR/bin:$PATH
```

保存退出

先不要执行#source set-qte-x86.sh （不要设置环境变量）

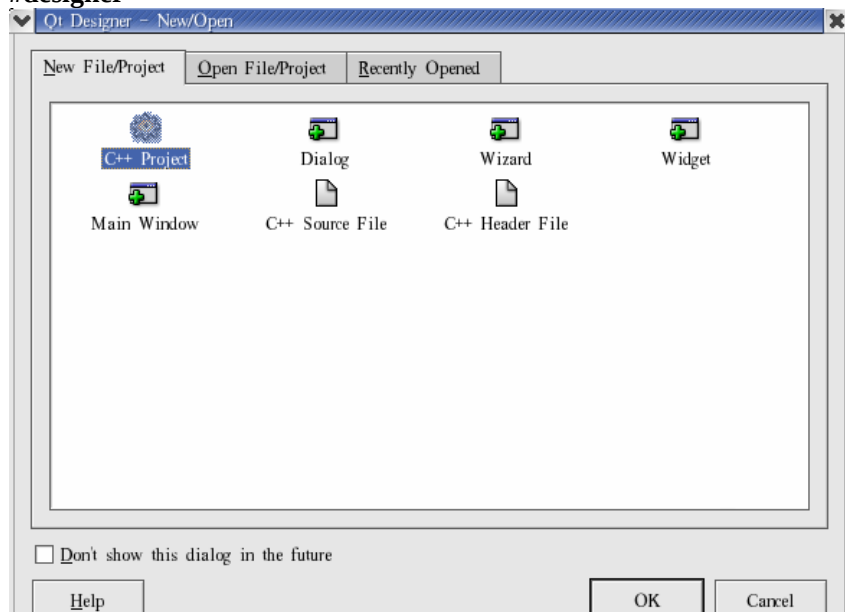
建立工程目录

```
#mkdir hello
```

```
#cd hello
```

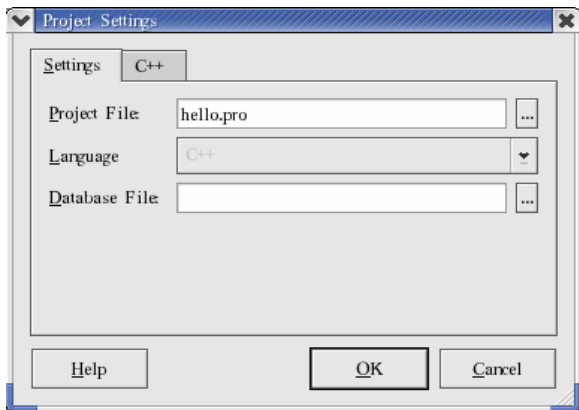
启动 QT 的 designer 生成工程文件(hello.pro)，窗体文件(hello.ui)，和 main.cpp 文件，如下：

```
#designer
```

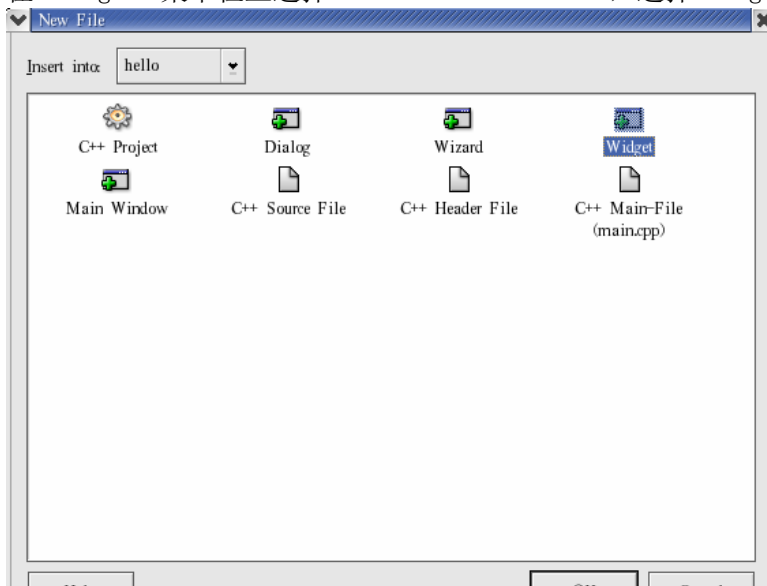


选择“C++ Project”，单击“OK”进入下一步；

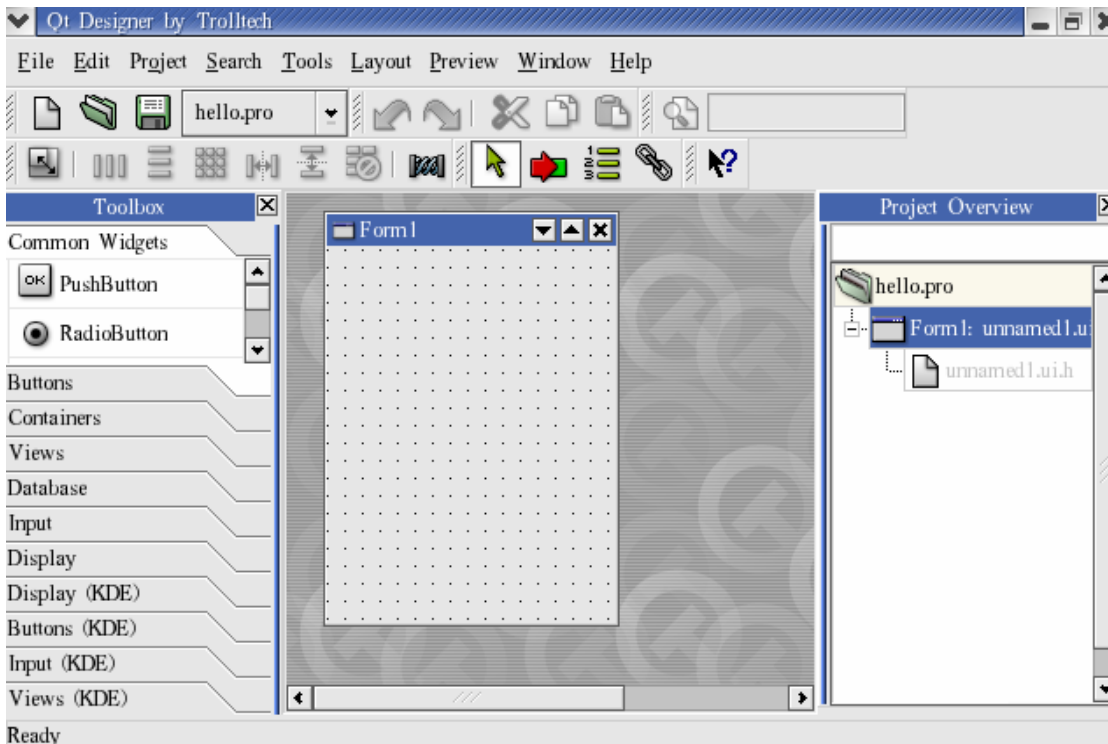
输入工程名称，单击“OK”即可，如下：



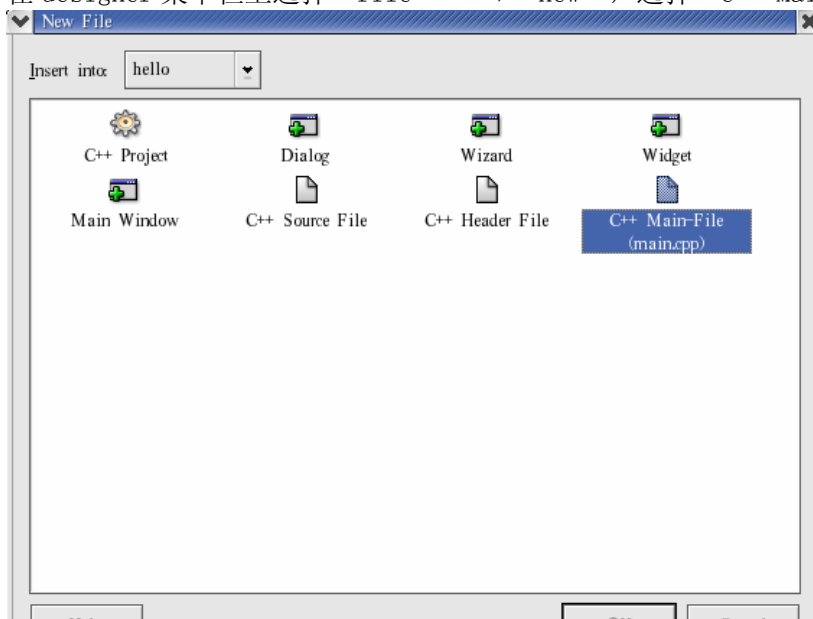
在 designer 菜单栏上选择“file”——>“new”，选择“widget”，生产一个窗体文件；



调整窗体的大小如下：



在 designer 菜单栏上选择“file”——>“new”，选择“C++ Main-File”，生产一个 main.cpp 文件；



选择“OK”，保存退出 designer；

设置环境变量：

#source ../set-qt-x86.sh

用 uic 工具生成源文件

#uic hello.ui -o hello.h

#uic hello.ui -i hello.h -o hello.cpp

删除 hello.pro, hello.ui 文件，省下 hello.h, hello.cpp, main.cpp 三个文件

生成工程文件：

#progen -o hello.pro

生成 Makefile 文件

#tmake hello.pro -o Makefile

开始编译

```
#make
```

在 PC 机的 qvfb 上运行

```
#qvfb &
```

新建终端，在新建的终端中执行如下

```
#cd /home/qte
```

```
#source set-qte-x86.sh
```

```
#cd hello
```

执行刚才编译的 hello

```
#./hello -qws
```

7.3.2 在 ARM 上的 qt/embedded 应用程序实例

如果qt-embedded已经交叉编译成ARM平台的库（参考QT/embbbed的安装），就可以开发ARM平台的QT应用程序了。

建立 qte 开发的工作目录

```
#mkdir /home/qte-arm
```

```
#cd /home/qte-arm
```

#建立 shell 文件, 设置环境变量

```
#vi set-qte-arm.sh
```

在文件中添加如下

```
#!/bin/bash
```

```
export QTDIR=/home/build_qte/qt-2.3.7-arm
```

#根据自己的路径来修改交叉编译 qt-2.3.7-arm 的目录的路径

```
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

```
export TMAKEDIR=/usr/tmake-1.11
```

#根据自己安装的 tmake-1.11 来添加 tmake 目录,我这里 tmake 的路径是:/usr/tmake-1.11

```
export TMAKEPATH=$TMAKEDIR/lib/qws/linux-arm-g++
```

```
export PATH=$QTDIR/bin:$TMAKEDIR/bin:/usr/local/arm/3.3.2/bin:$PATH
```

```
echo $PATH
```

```
echo $TMAKEPATH
```

保存退出

先不要执行 source set-qte-arm.sh（不要设置环境变量）

建立工程目录

```
#mkdir hello
```

```
#cd hello
```

用 designer 生成工程文件(hello.pro)，窗体文件(hello.ui)，和 main.cpp 文件。

设置环境变量：

```
#source ../set-qte-arm.sh
```

生成源文件

```
#uic hello.ui -o hello.h
```

```
#uic hello.ui -i hello.h -o hello.cpp
```

删除 hello.pro, hello.ui 文件，省下 hello.h, hello.cpp, main.cpp 三个文件

用 progen 生成工程文件

```
#progen -o hello.pro
```

用 tmake 工具生成 Makefile 文件

```
#tmake hello.pro -o Makefile
```

```
#vi Makefile ;编辑刚刚生成的 Makefile 文件
```

把 LINK = arm-linux-gcc 改为

```
LINK = arm-linux-g++
```

把 LIBS = \$(SUBLIBS) -L\$(QTDIR)/lib -lm -lqte 改为

```
LIBS = $(SUBLIBS) -L/usr/local/arm/3.3.2/lib -L$(QTDIR)/lib -lm -lqte
```

保存退出

修改完 Makefile 之后，就可以编译了；

```
#make
```

编译完之后, 生成 hello 的可执行文件, 把可执行文件下载到开发板上用以下命令运行

```
./hello -qws ; 执行编译的 hello.
```

第八章 嵌入式 WINCE

8.1.WINCE 的安装

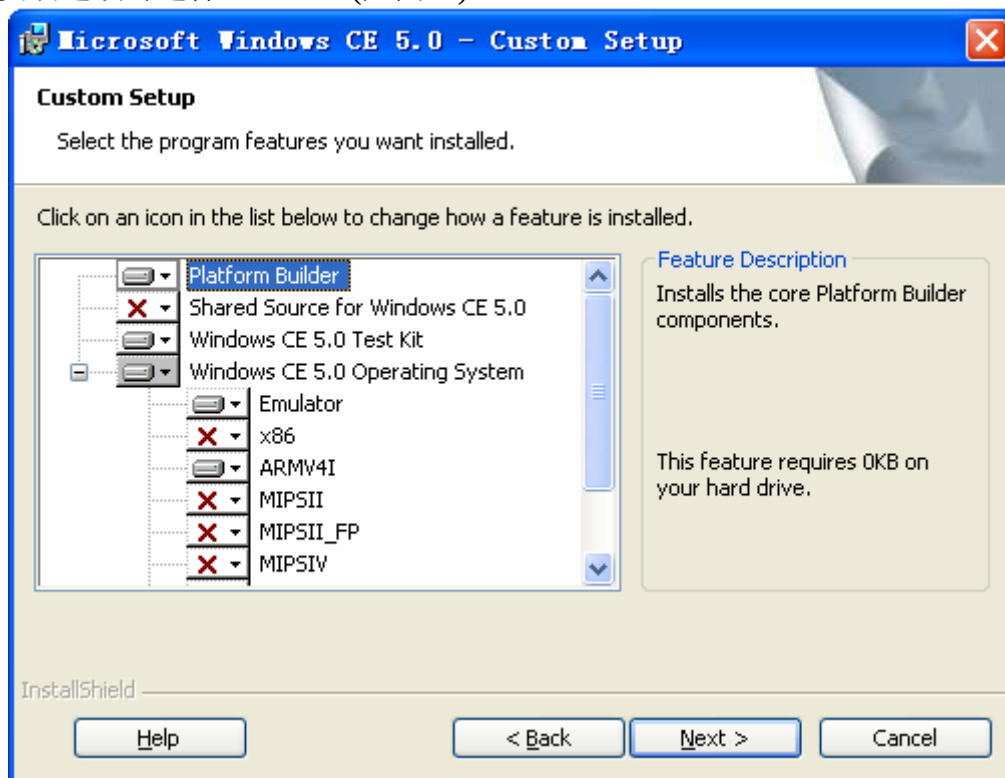
8.1.1 安装开发环境

为了建立 WINCE5.0 的应用开发环境，您需要准备好由微软公司发布的 PB (Platform Builder 5.0) 安装光盘。这张光盘包含了微软公司的 Windows® CE .NET 5.0 操作系统安装程序，以及把操作系统编译移植到指定目标硬件平台的工具—平台建立器 (Platform Builder 5.0)。

下面介绍如何从这张光盘安装 WINCE5.0 操作系统和 Platform Builder 5.0。首先将光盘的内容全部拷贝到同一个目录下，例如：D:\PB5SET。

运 SETUP.EXE, 输入序列号

在出现安装选项时选择 ARMV4I(如图一)

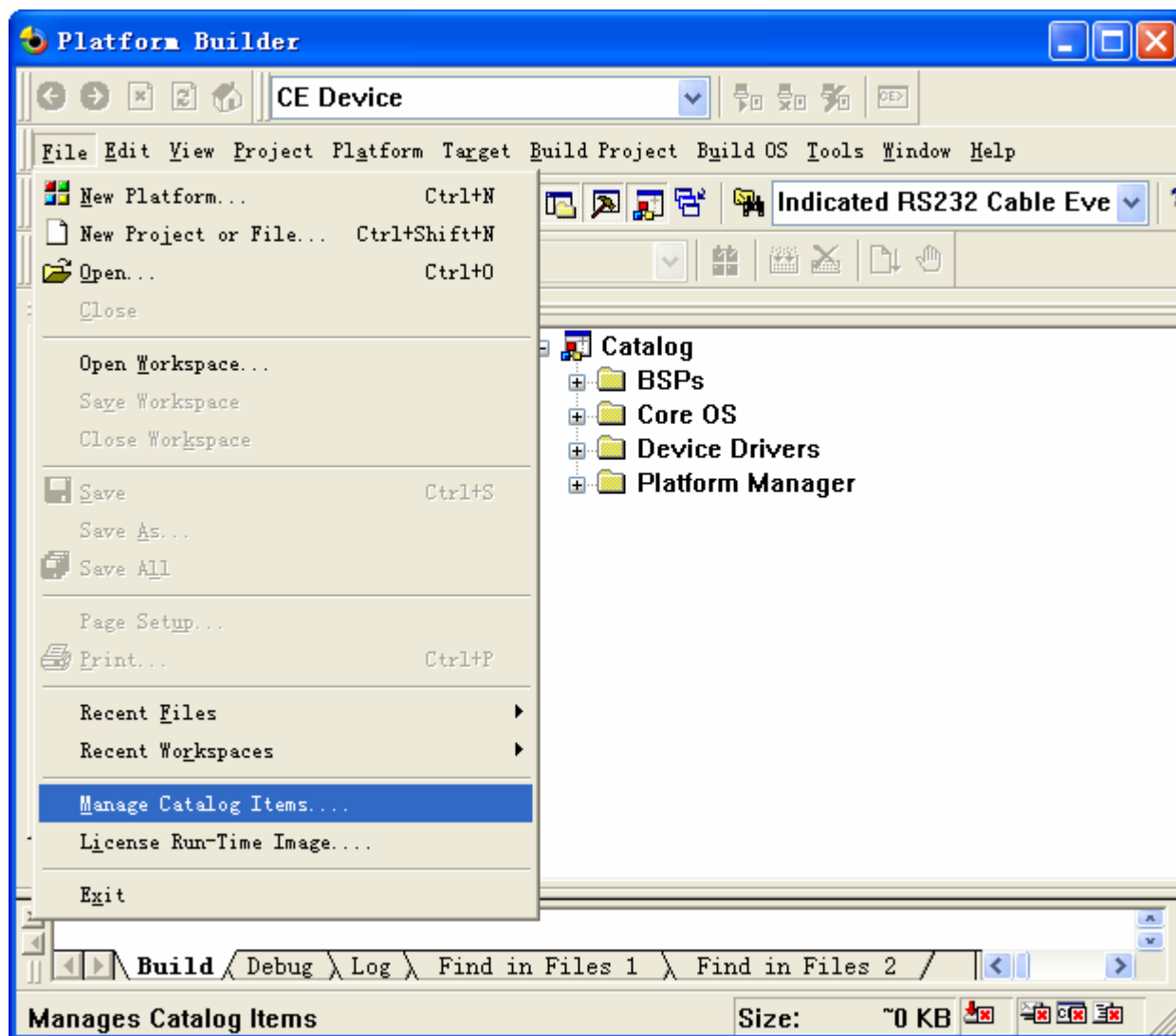


(图一)

安装结束时，WINCE 目录有 3.5G 左右的文件。

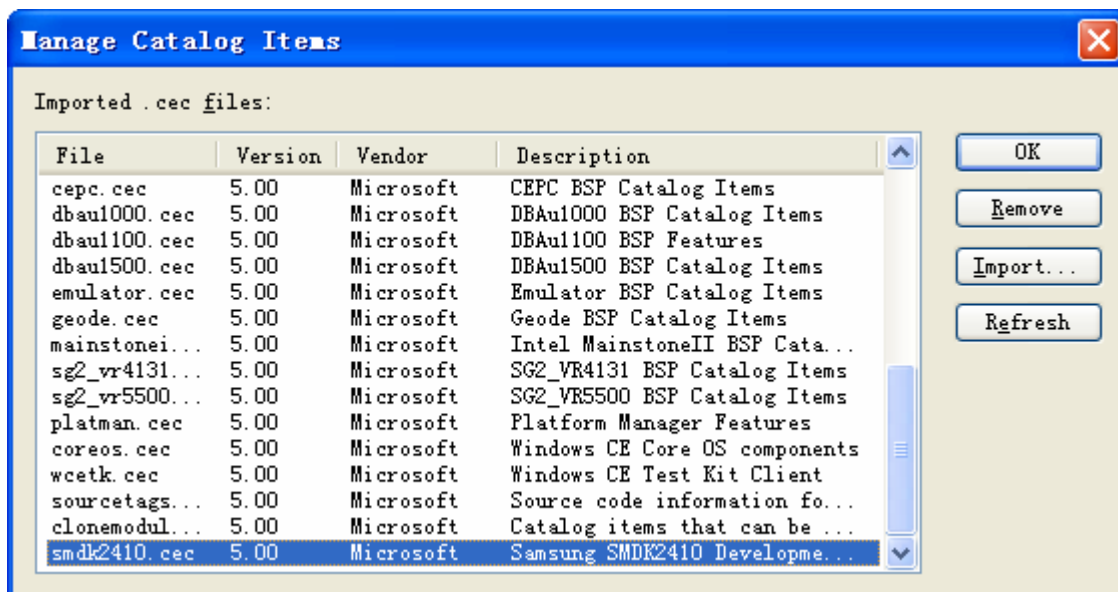
8.1.2 安装目录

前面已经安装了平台建立器 Platform Builder 5.0，下一步，要运行 Platform Builder，并对它进行一系列的设置，目的是为编译 WINCE5.0 操作系统映像做好准备。点击 PB 主菜单“File”下的“Manage Catalog Items”菜单项，如下图：



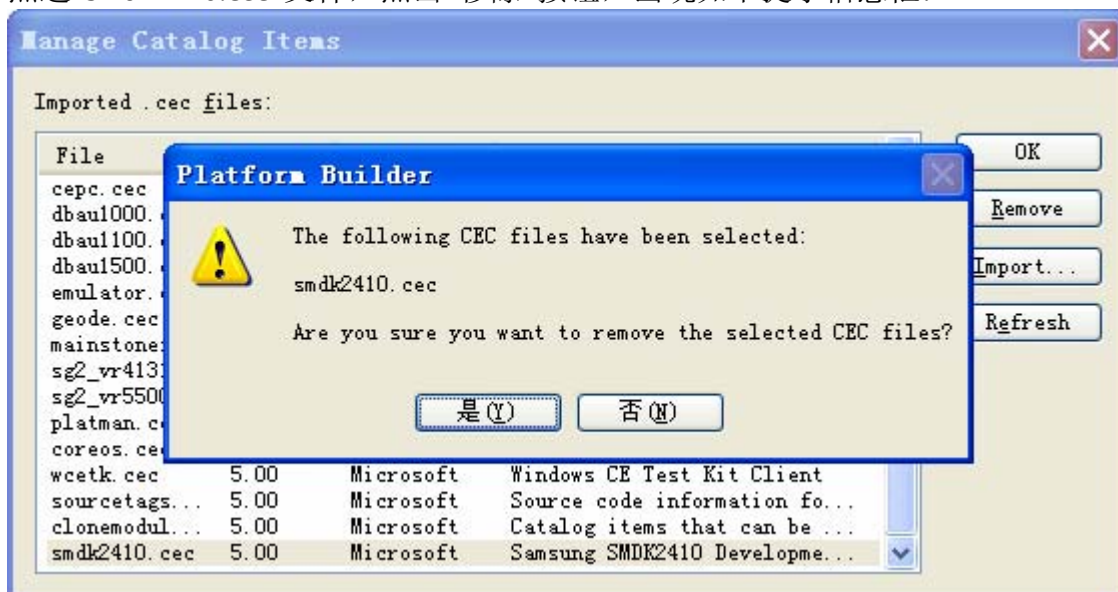
(图二)

您可以看到如下图所示的 Platform Builder 里的“SMDK2410.CEC”文件。这是系统中以前安装过的，您必须先删除掉该文件。



(图三)

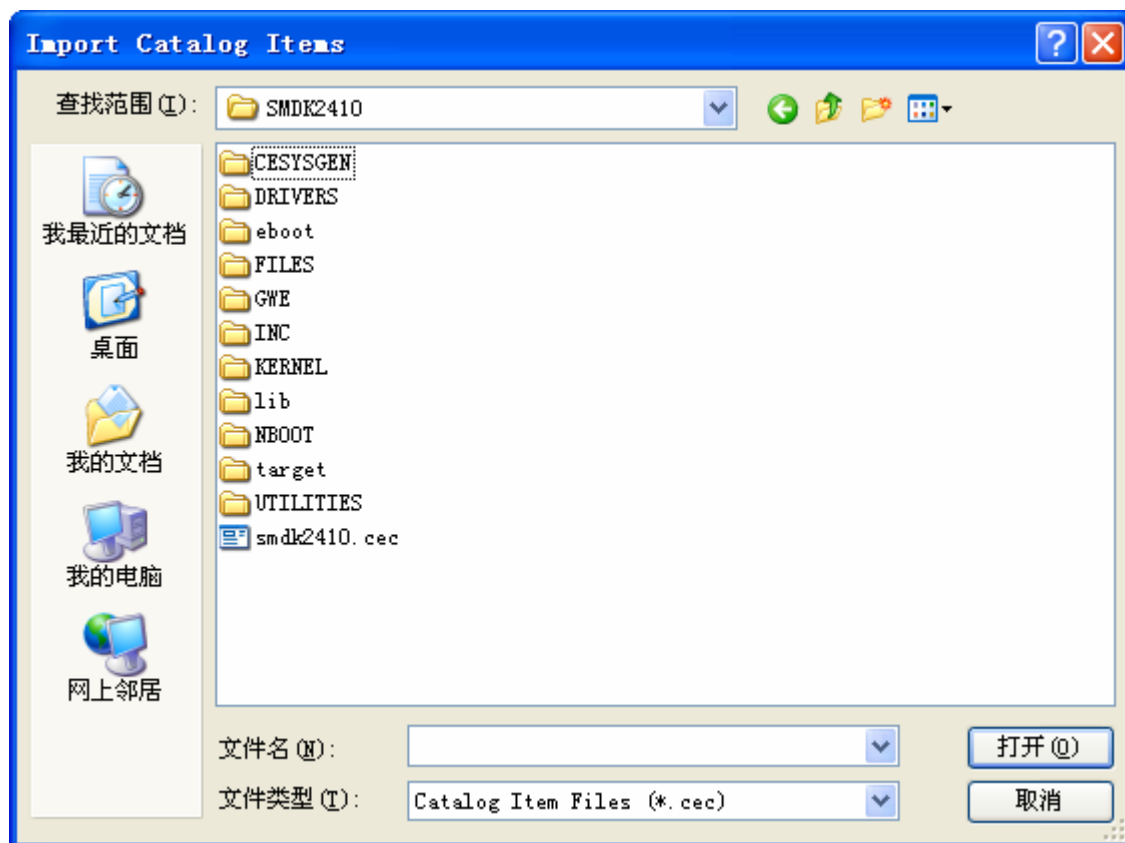
点选“smdk2410.cec”文件，点击“移除”按钮，出现如下提示信息框：



(图四)

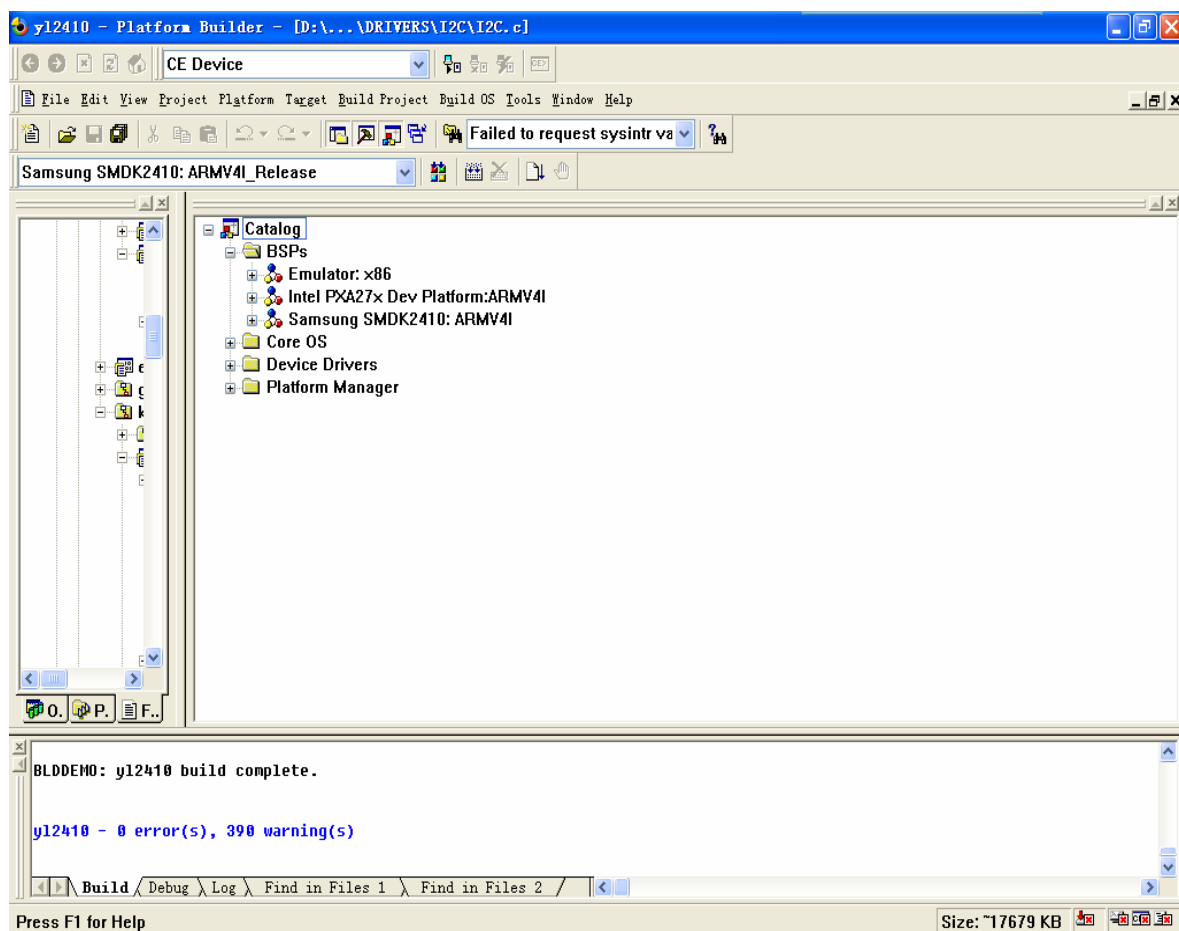
点击按钮“Yes”确定移除操作。

然后，您需要“导入”（Import）新的“SMDK2410.CEC”文件。新的“SMDK2410.CEC”文件是从 2410 用户光盘的 wince 目录中的 wincebsp.rar 压缩包解压出来的，一般我们将这个压缩包解压缩到 wince500 的安装目录下的 PLATFORM 子目录，例如 WINCE500\PLATFORM；然后就可以点击“Import” 按钮导入新的“SMDK2410.CEC”文件了，打开 WINCE500\PLATFORM\SMDK2410 目录，选中该目录下的“SMDK2410.CEC”文件，如下图：



(图五)

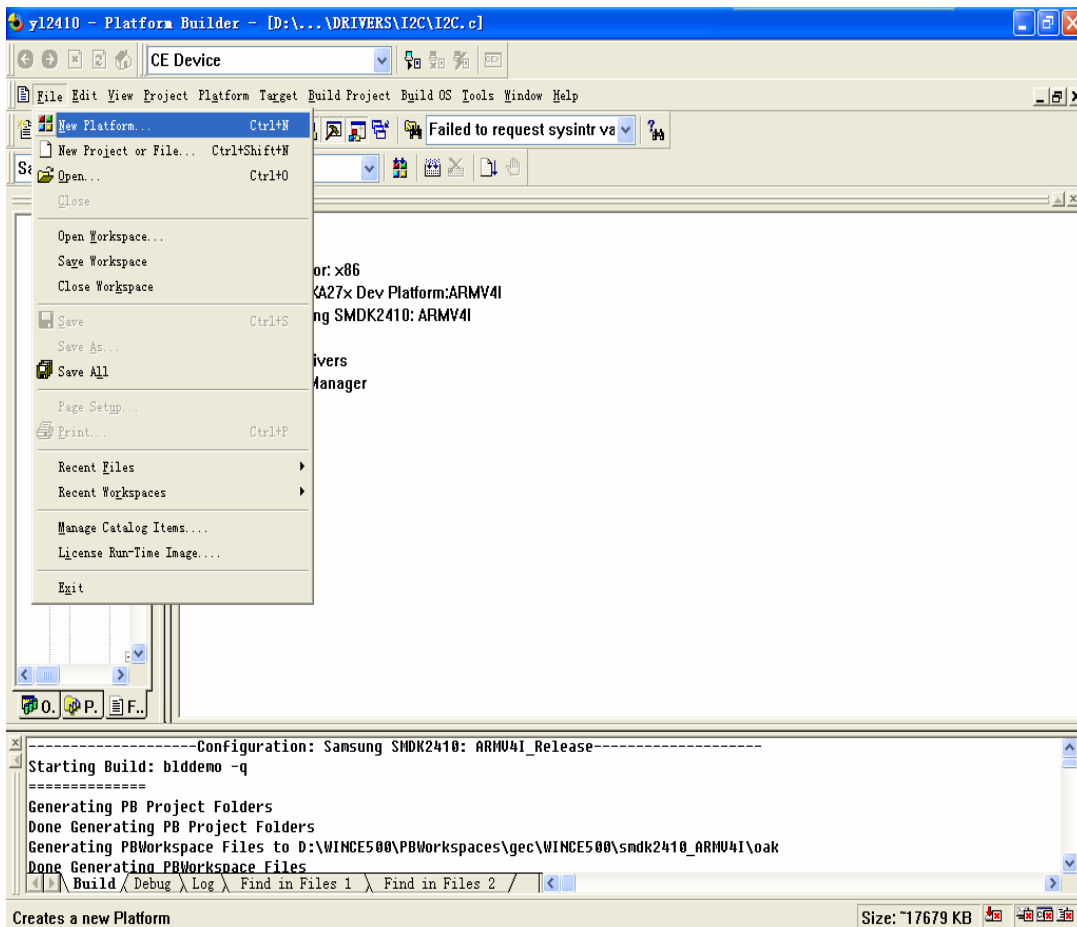
导入之后，在 PB 的目录查看器上将会看到“Samsung SMDK2410:ARMV4”，“SamsungSMDK2410:ARMV4I” 列，如下图：



(图六)

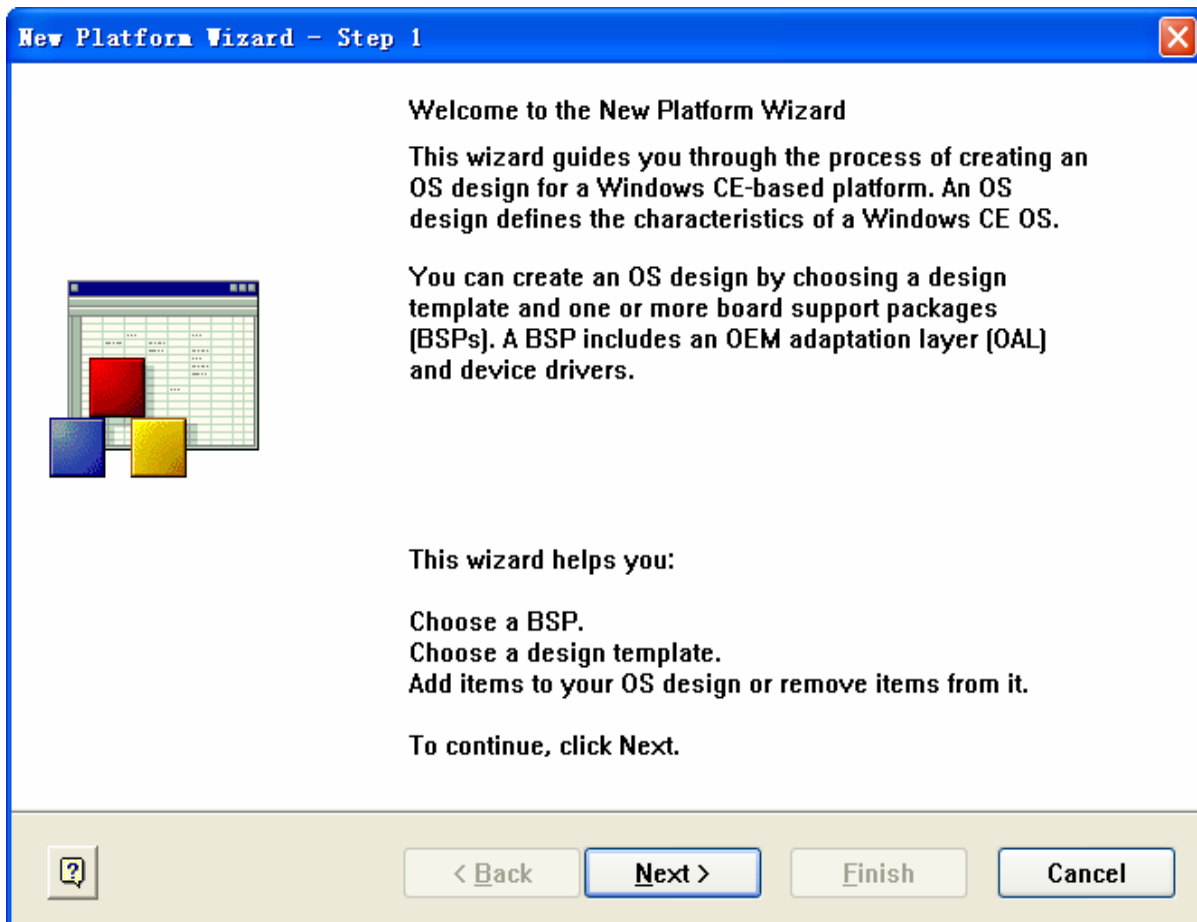
81.3 构建新的平台、编译

点击 PB 主菜单“File”下的“New Platform”菜单项，如下图：



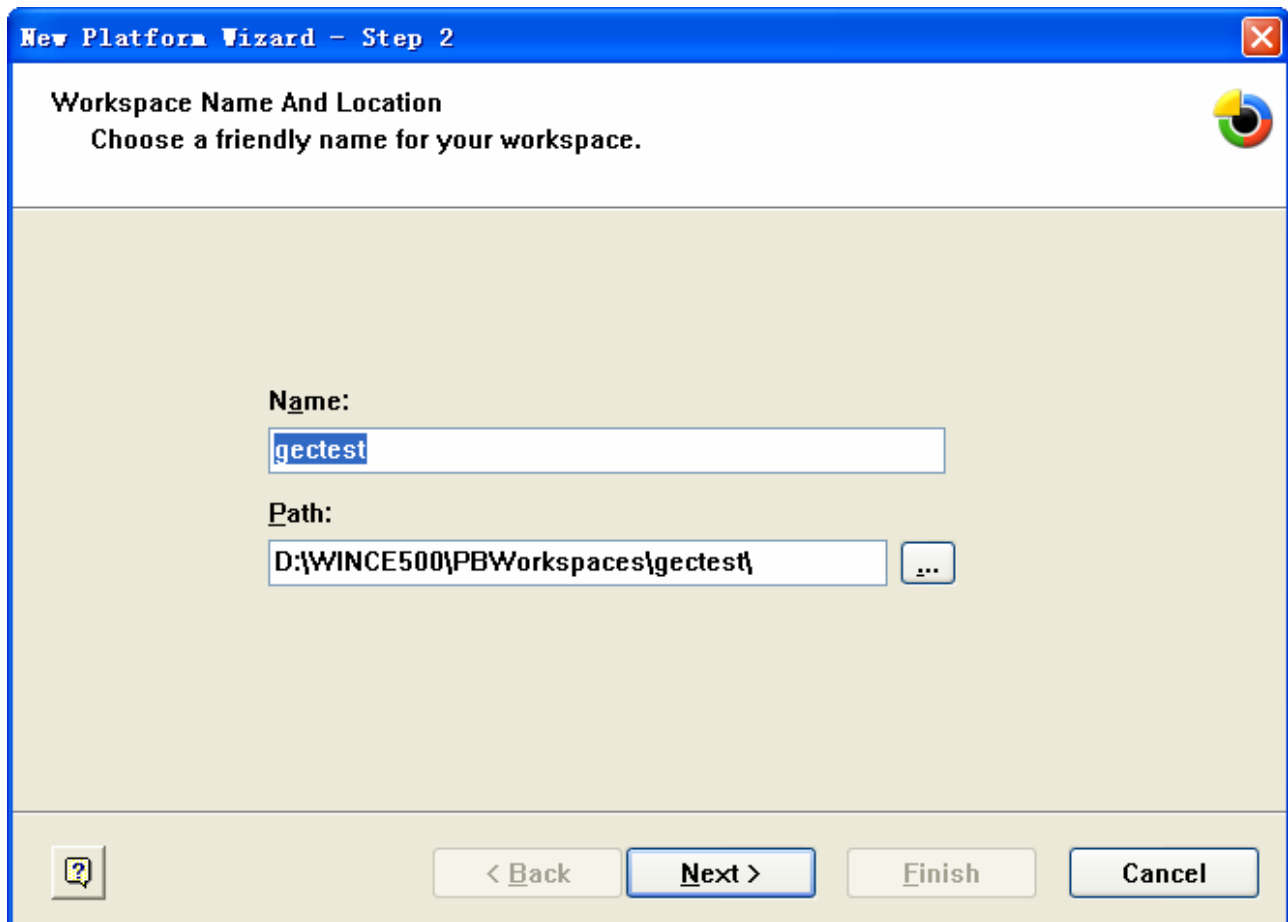
(图七)

将会出现“New Platform Wizard – Step 1”框，点击“Next”按钮：



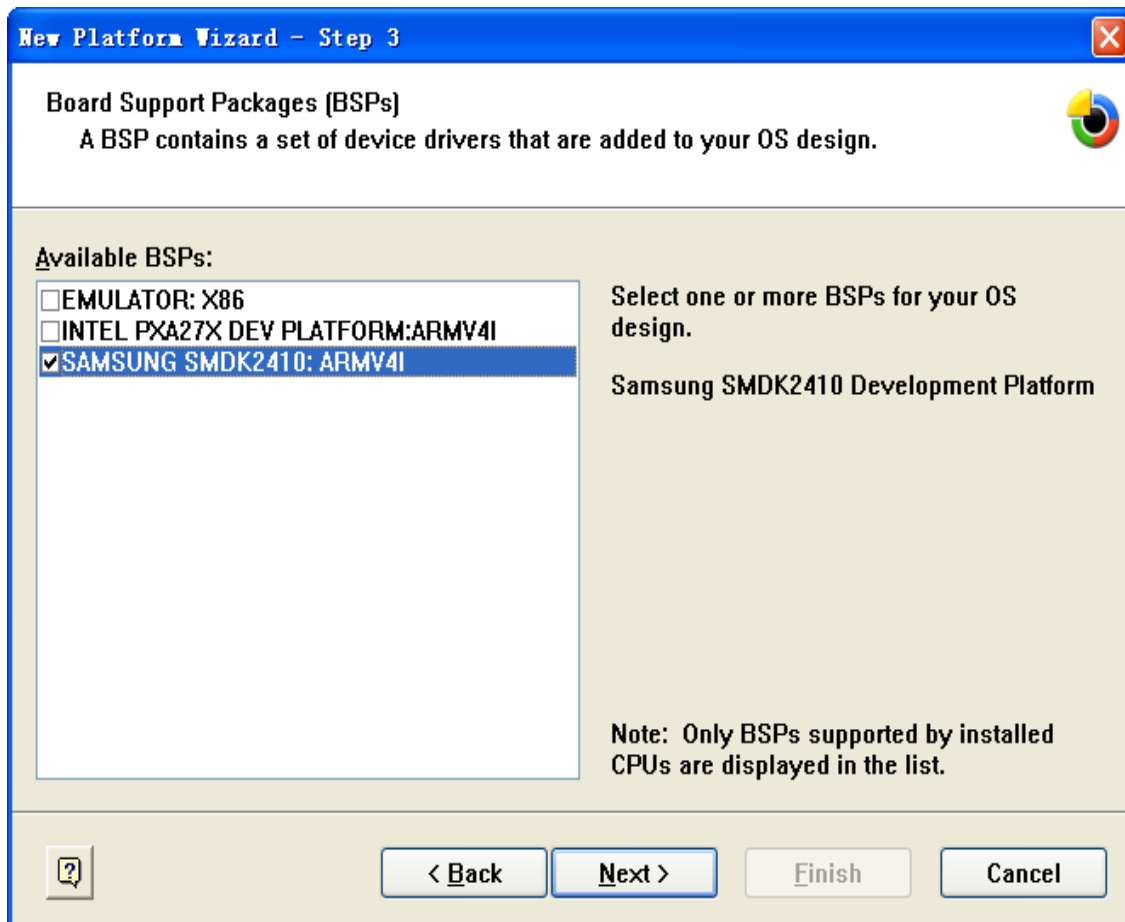
(图八)

接着出现“New Platform Wizard – Step 2”框，您可看到” Name” 的文本编辑框。编写名字，再点击“Next”按钮，如下图：



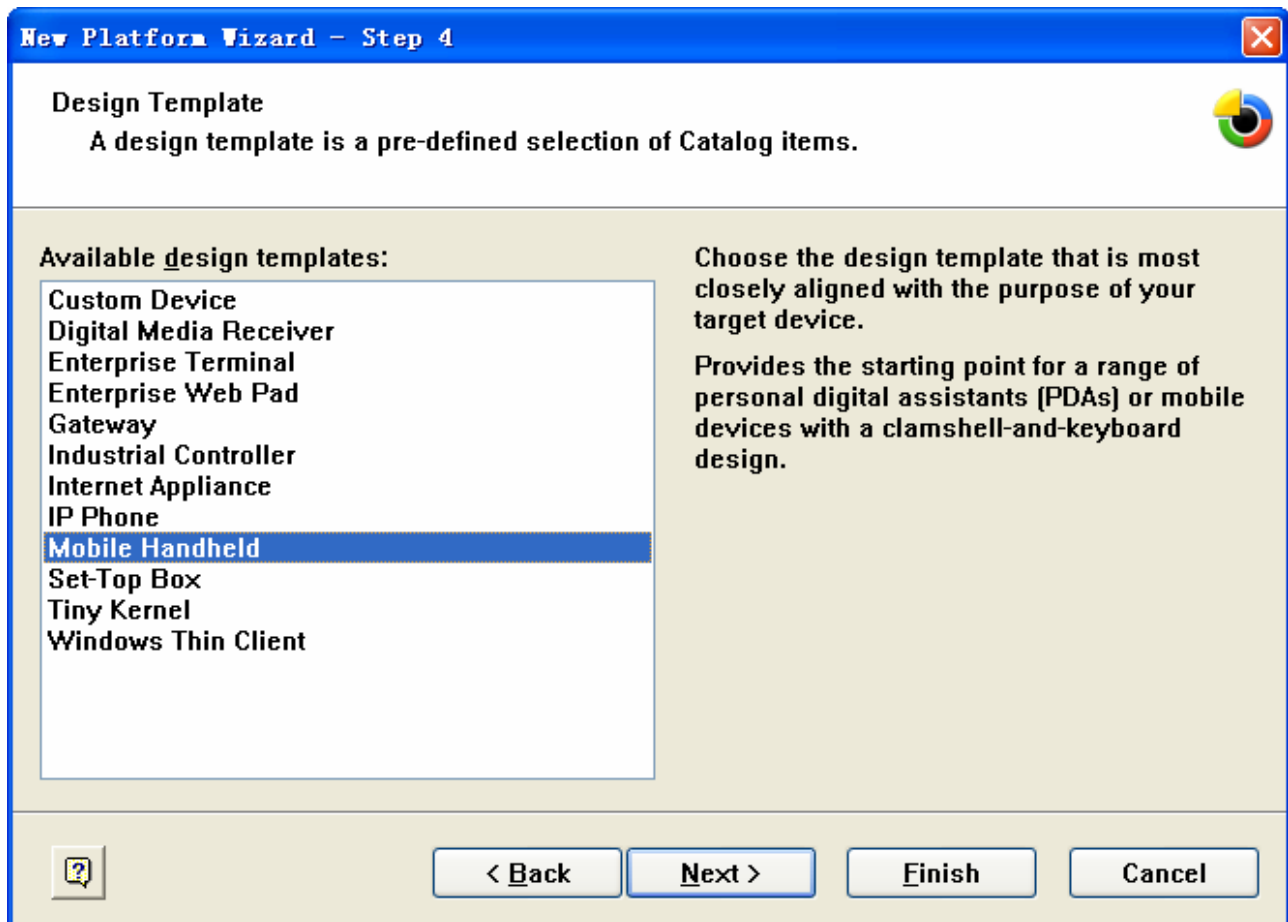
(图九)

接着出现“New Platform Wizard – Step 3”框，您可看到“SAMSUNG SMDK2410:ARMV4”的 BSP。点击“SAMSUNG SMDK2410:ARMV4”、再点击“Next”按钮，如下图：



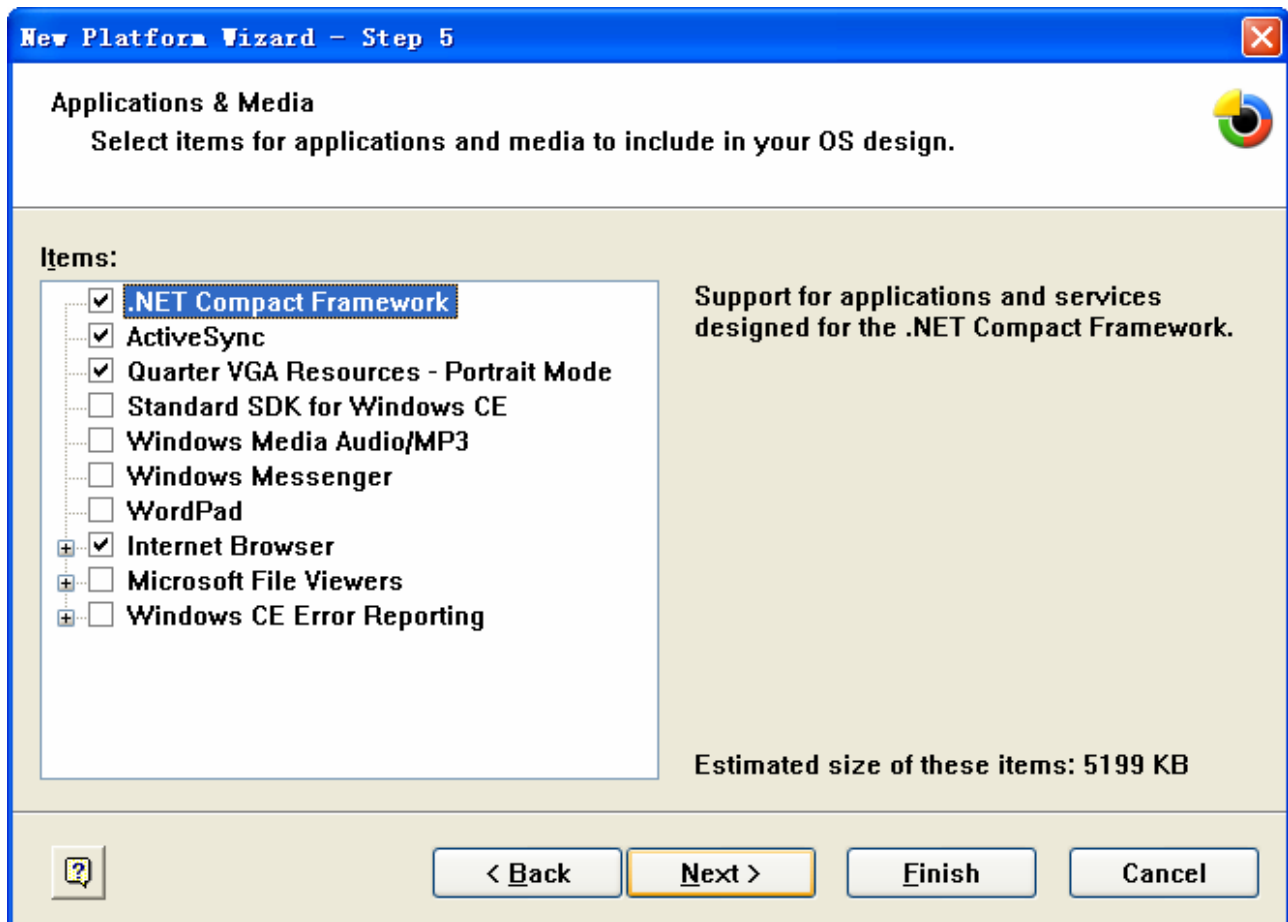
(图十)

出现“New Platform Wizard – Step 4”对话框，在“Available configurations”列表中选择您希望的配置，在“Platform name:”框中输入平台的名称 gec2410、再点击“Next”按钮，如下图：



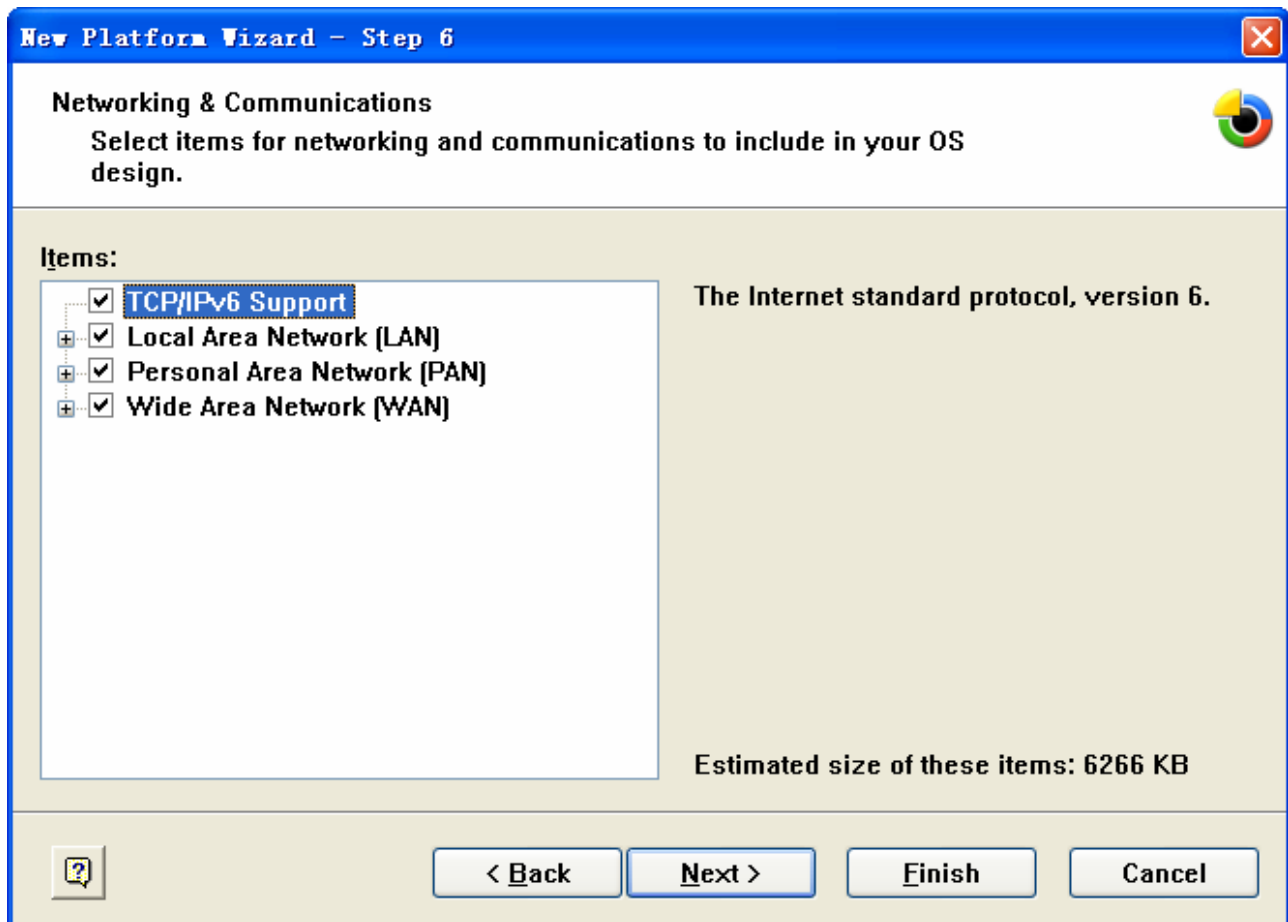
(图十一)

接着出现“New Platform Wizard – Step 5”框，选择您需要的“Application& Media”，如下图：



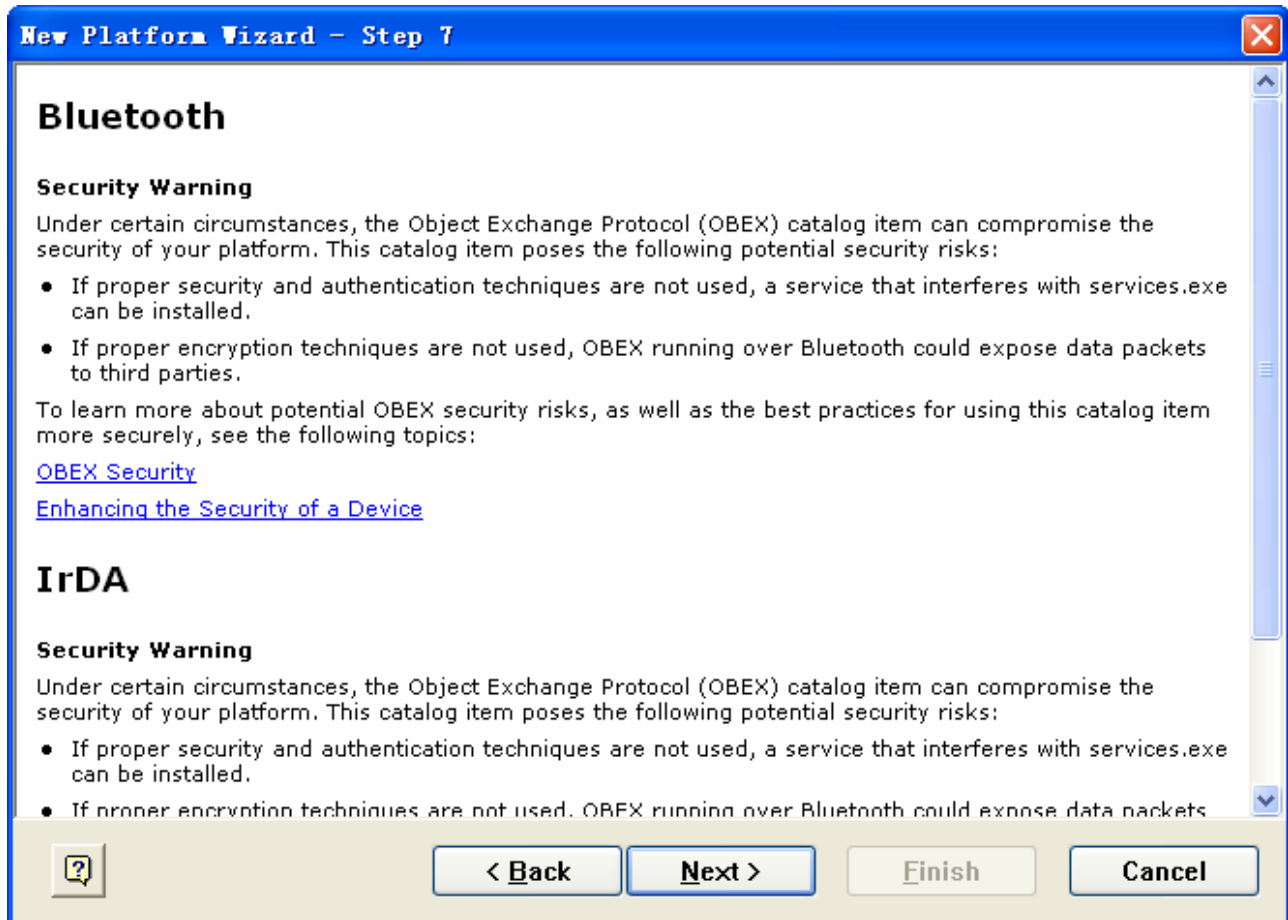
(图十二)

接着出现“New Platform Wizard-Step 6”框，选择您需要的“Networking & Communications”，如下图：



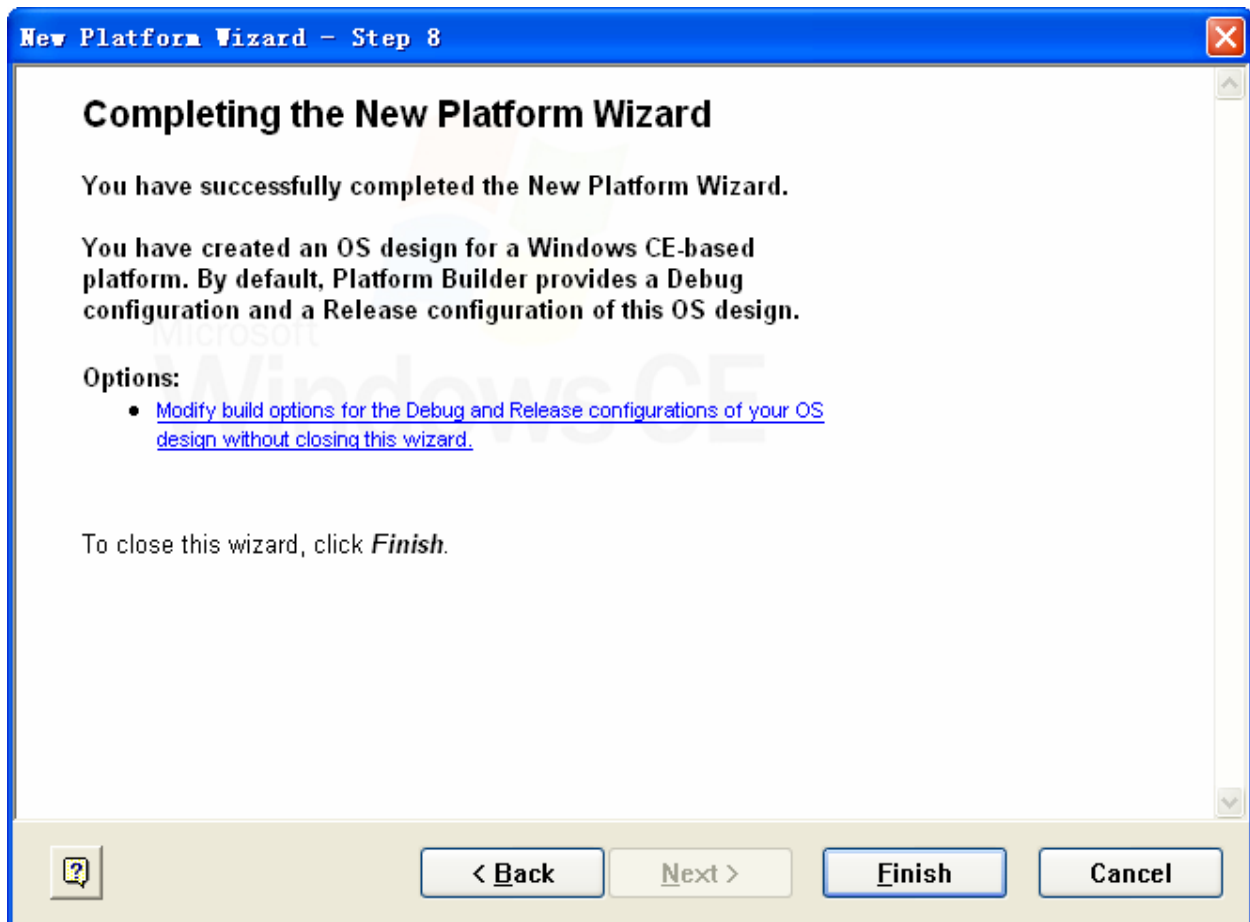
(图十三)

再点击”Next”按钮，出现以下的对话框。



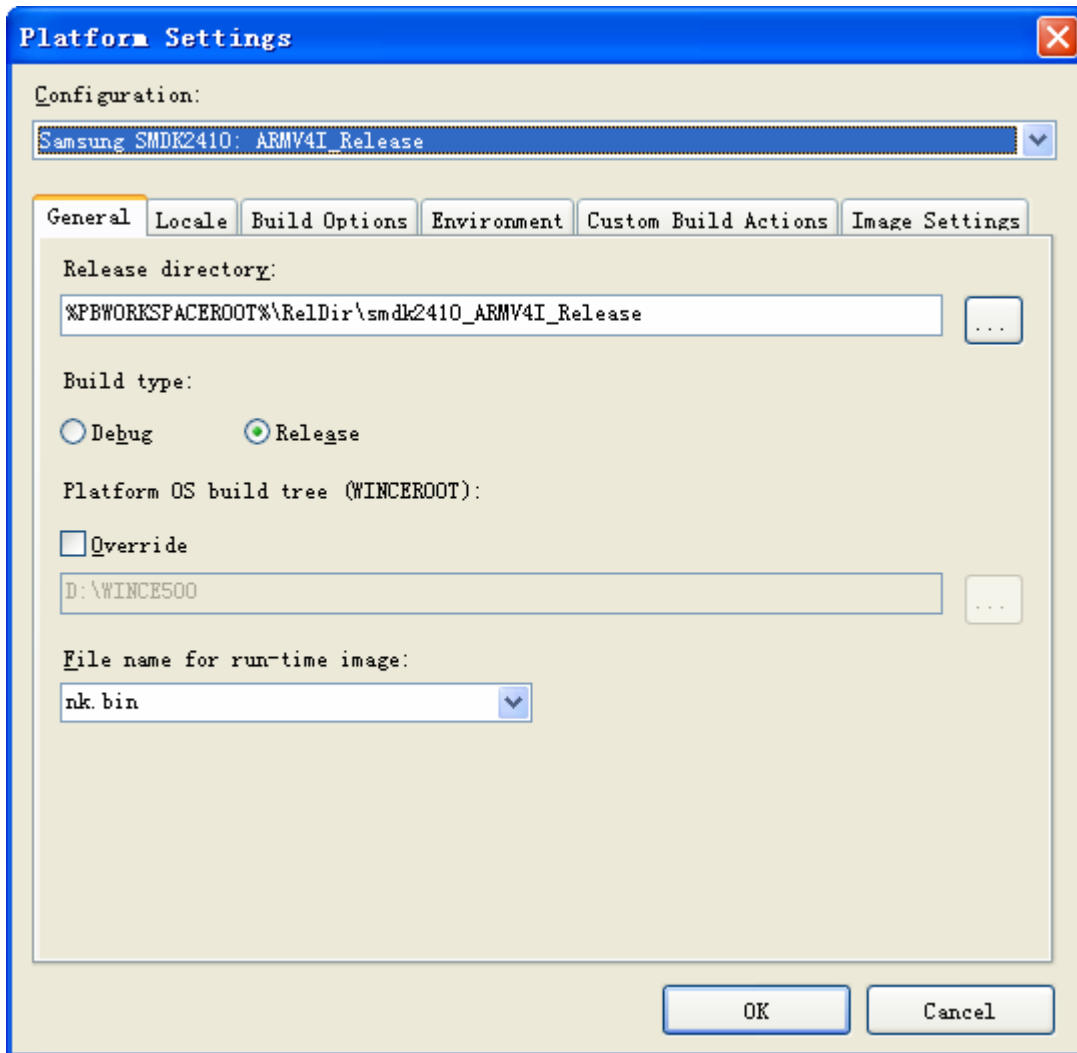
(图十四)

再点击“Next”按钮，您将看到“Completing the New Platform Wizard”对话框。建立新平台的所有设置步骤已经完成了。请点击“Finish”按钮。



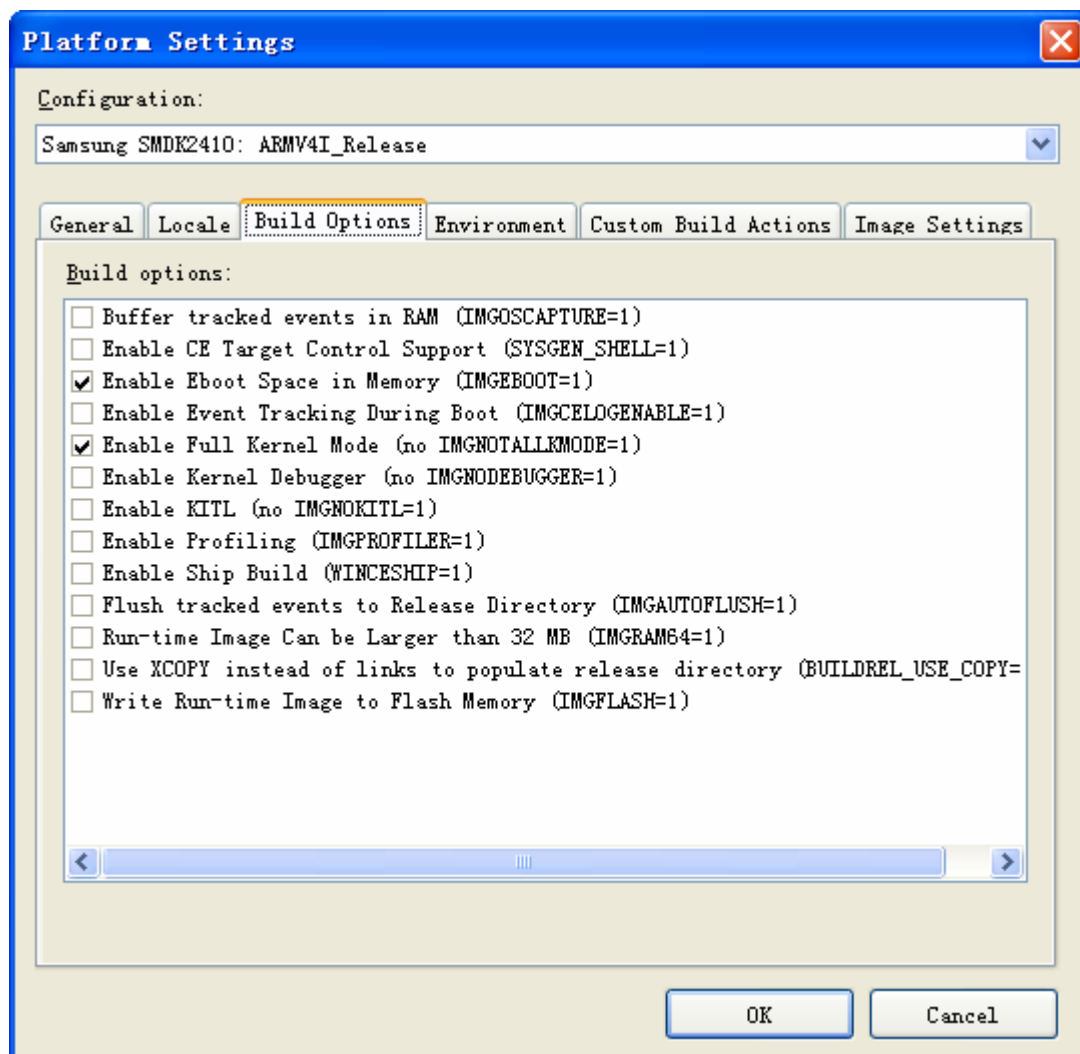
(图十五)

下一步设置平台，点击 PB 的 Platform| Setting 菜单，如下图：



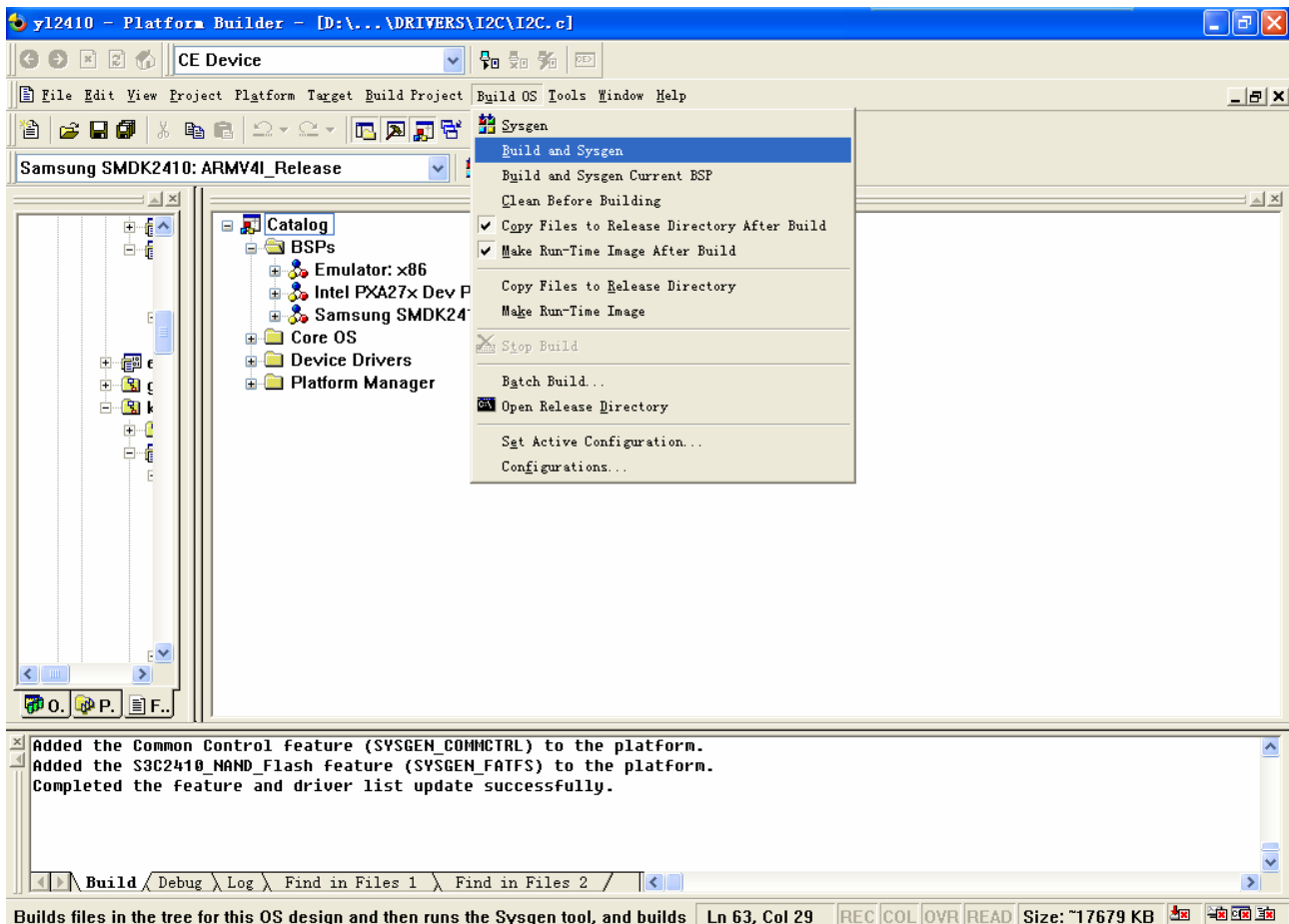
(图十六)

在弹出的“Platform Settings”设置框的“Build Options”标签页，作如下修改：取消选项[Enable CE Target Control Support]、取消选项[Enable KITL]，选中[Enable Full Kernel Mode]，选中[Enable Eboot Space in Memory]，如下图



(图十七)

设置完成后，点击 Build | Build Platform 开始编译平台



(图十八)

编译完成后，您就有了 SMDK2410 的二进制的 image：“nk.bin”和“nk.nb0”，一般而言，这两个文件位于编译平台时生成的文件夹
“WINCE500\PUBLIC\[PlatformName]\RelDir\SAMSUNG_SMDK2410_ARMV4Release”。

8.2.WINCE 的 image 的运行

前面编译平台生成的文件 nk.nb0，就是 wince5.0 操作系统以 2410 硬件平台为目标而编译生成的二进制映像文件（image）。这个文件可以被下载到 GEC2410 的 RAM 空间执行，执行的结果就是在 RAM 中直接运行了 wince5.0 操作系统，这种方式并不会把 wince 映像固化到 2410 的 FLASH 中，系统调电之后在 RAM 中的 wince 就不复存在了。可以通过以下两种方式把 nk.nb0 映像文件下载到 GEC2410 的 RAM 空间执行。

关于 WINCE 的 IMAGE 在 RAM 空间运行，请看《GEC2410 用户手册》相关章节。

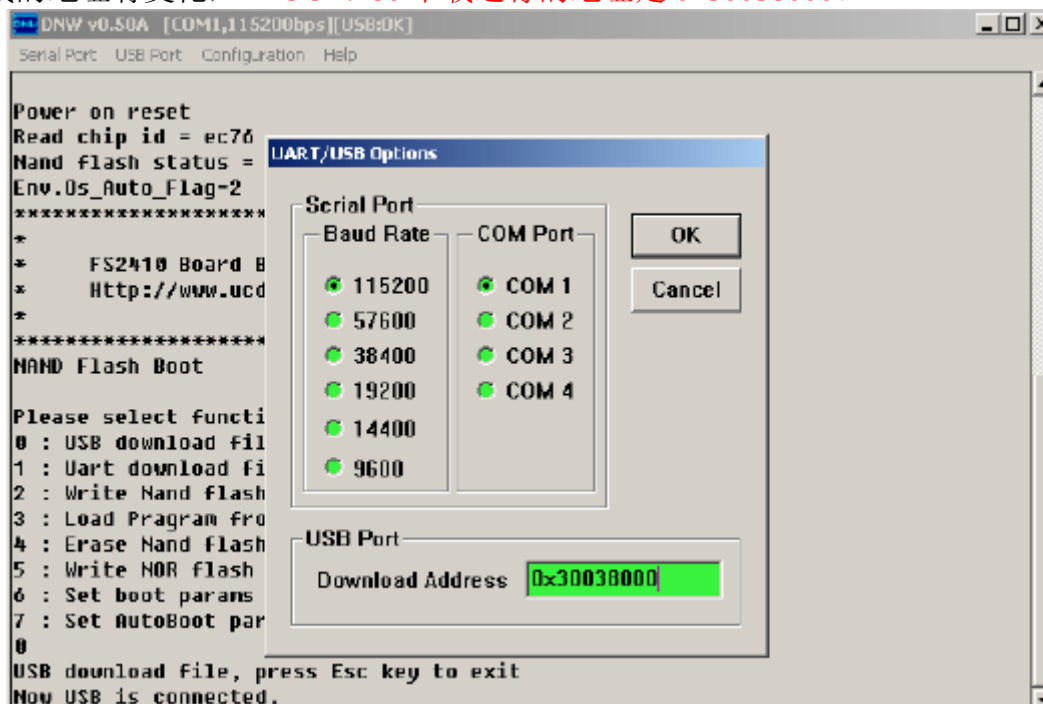
8.3.通过以太网烧写 WINCE 映像文件

将 wince5.0 映像文件 nk.nb0 固化（烧写）到 FLASH 后，就不用每次下载 wince5.0 到内存中运行了。目前只能通过用以太网下载的方式烧写 wince5.0 映像文件或者相关章节的方法烧写，下面具体描述网络烧写的过程：

- (1) 首先确保 PB 程序已经启动,并且打开了先前所建立平台时生成的工作空间（WORKSPACE），打开工作空间的方法是点击 PB 的菜单 File | Open WorkSpace

(2) 通过 USB 口下载引导设置程序 EBOOT.nb0

通过 USB 下载运行 EBOOT.nb0 与上一节的下载运行 WINCE 的 NK.nb0 类似，只是下载的地址有变化，**EBOOT.nb0 下载运行的地址是 0x30038000。**

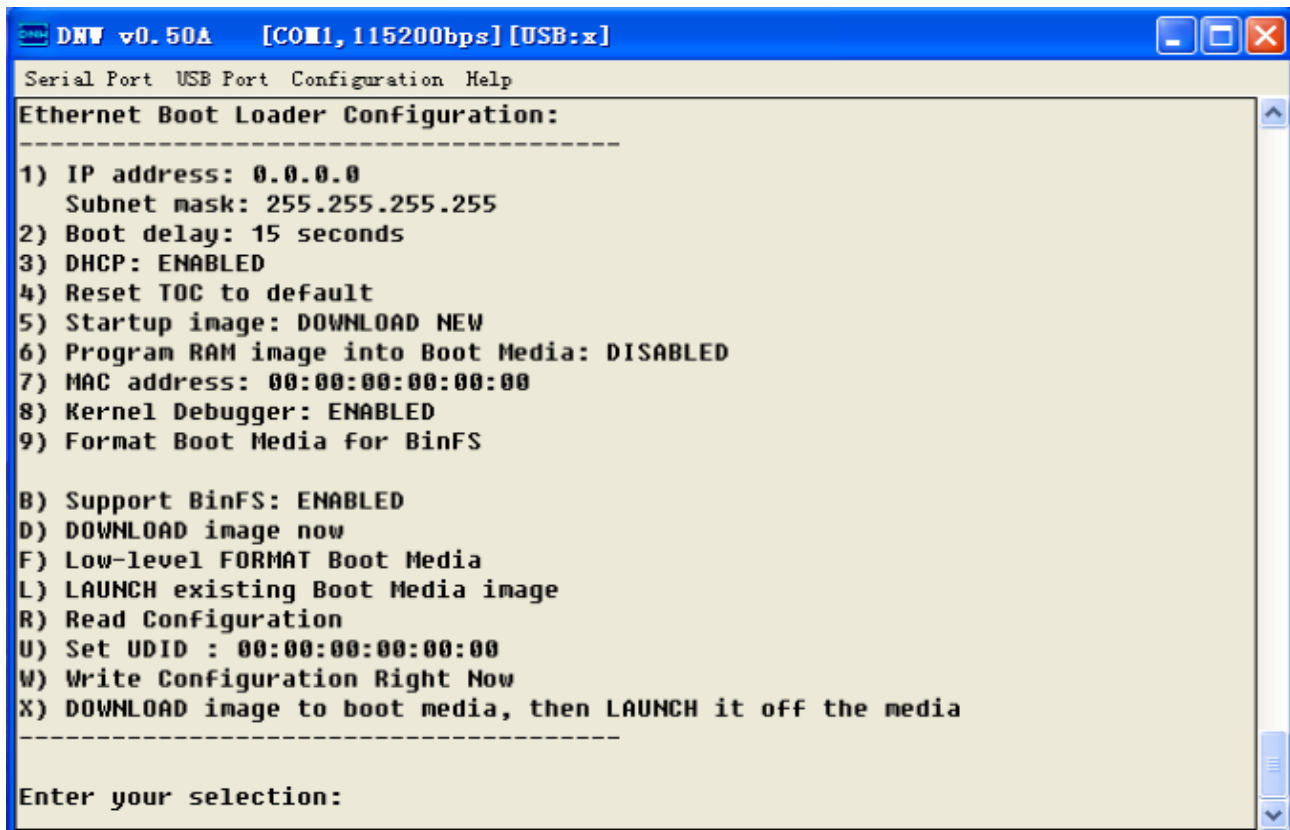


(图十九)

接下来就可以选择 DNW 的菜单“USB port”下的“Transmit”子菜单项来下载 EBOOT.nb0 一般而言，EBOOT.nb0 文件位于编译平台时生成的文件夹。

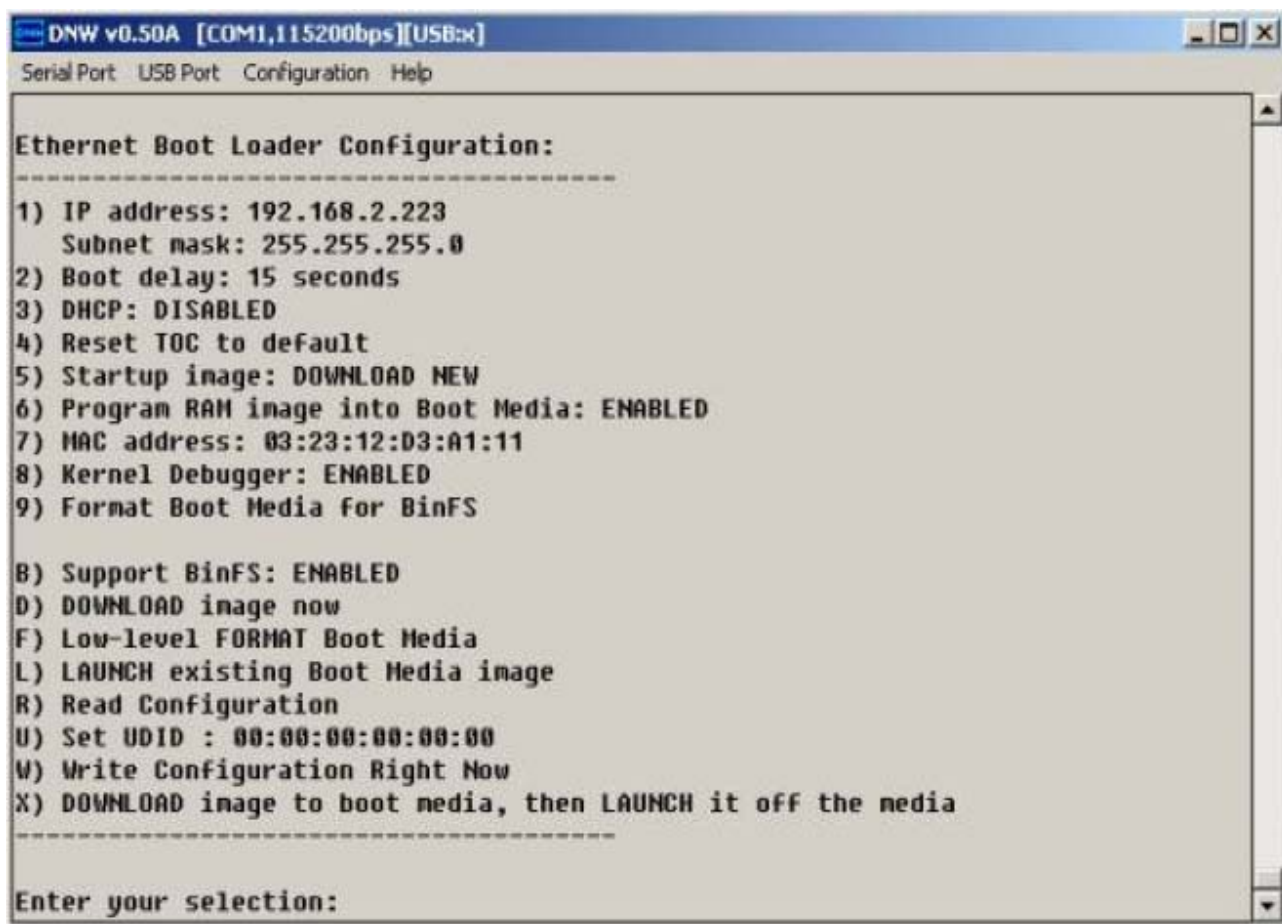
“WINCE420\PUBLIC\[PlatformName]\RelDir\SAMSUNG_SMDK2410_ARMV4Release”。

当文件 EBOOT.nb0 下载完毕后，会自动运行 EBOOT.nb0 这个程序，程序运行后会提示“Press [ENTER] to download image now, or [SPACE] to enter boot monitor”这时按“SPACE”键，进入的界面如下：



(图二十)

接着进入了 WINCE 引导设置界面，见下图



(图二十一)

在这个设置界面，您需要按顺序设置以下几个参数：

① 设置 MAC 地址

按 PC 键盘的“7”键进入 MAC 地址设置，MAC 地址是 wince 网卡的地址，您只要按 XX.XX.XX.XX.XX.XX 的格式随意输入六部分十六进制的数就行了，例如 12. 11. 1A. 2C. 33. 21 （回车）

② 设置系统的 IP 地址

按 PC 键盘的“1”键进入 IP 地址设置，您必须按 xxx.xxx.xxx.xxx 的格式输入 IP 地址，例如 192. 168. 2. 223 （回车）

要注意这个 IP 地址必须和你的 PC 机网卡设的 IP 地址处在同一子网段，否则将不能建立网络通讯。然后再设置子网掩码，一般是 255. 255. 255. 0

③ 设置烧写到 FLASH 为使能

按 PC 键盘的“6”键设置将 RAM 内的映像烧写到 FLASH，也就是按下“6”键后，设置菜单的第 6) 项应该为“6) Program RAM image into Boot Media: ENABLED”，如果该项设置为 DISABLE,那么我们就只是把映像下载到内存而不是烧写它。

④ 设置 Support BinFS 为使能

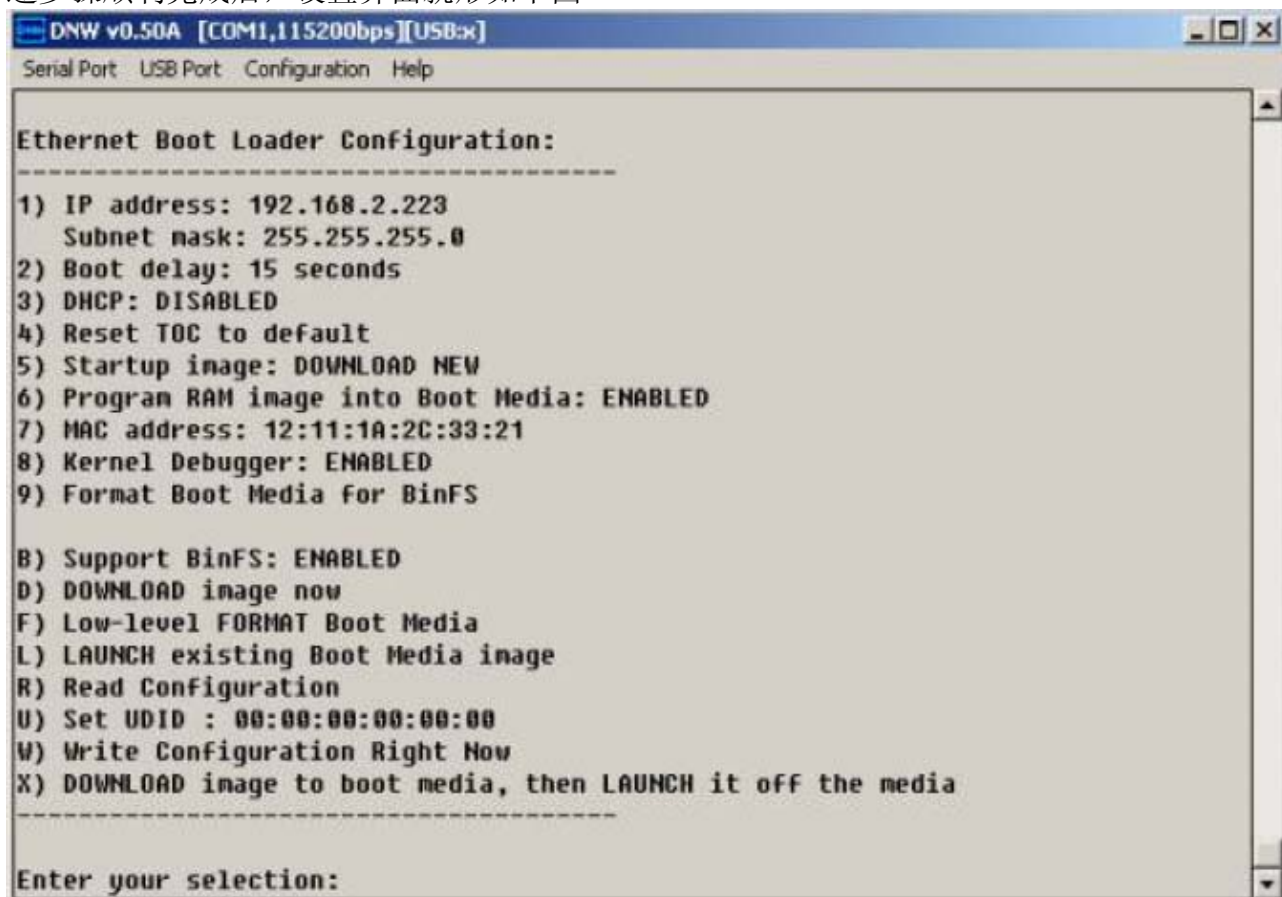
按 PC 键盘的“B”键将 Support BinFS 为设置为使能。

⑤ 低级格式化 FLASH

按 PC 键盘的“F”键低级格式化 FLASH

⑥ 高级格式化 FLASH

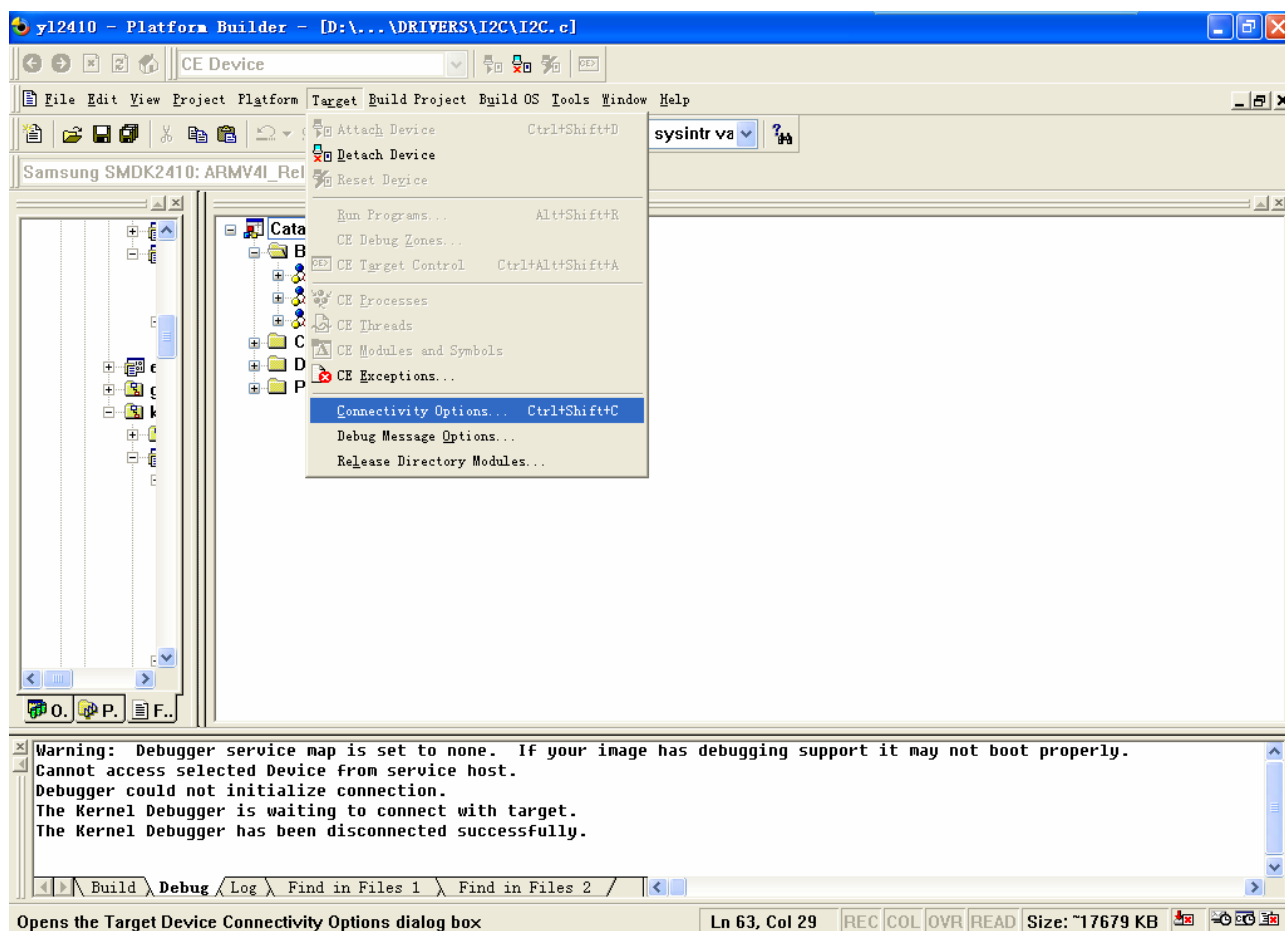
按 PC 键盘的“9”键高级格式化 FLASH，这一步骤可能需要几分钟时间，请耐心等待。上述步骤顺利完成后，设置界面就形如下图



(图二十二)

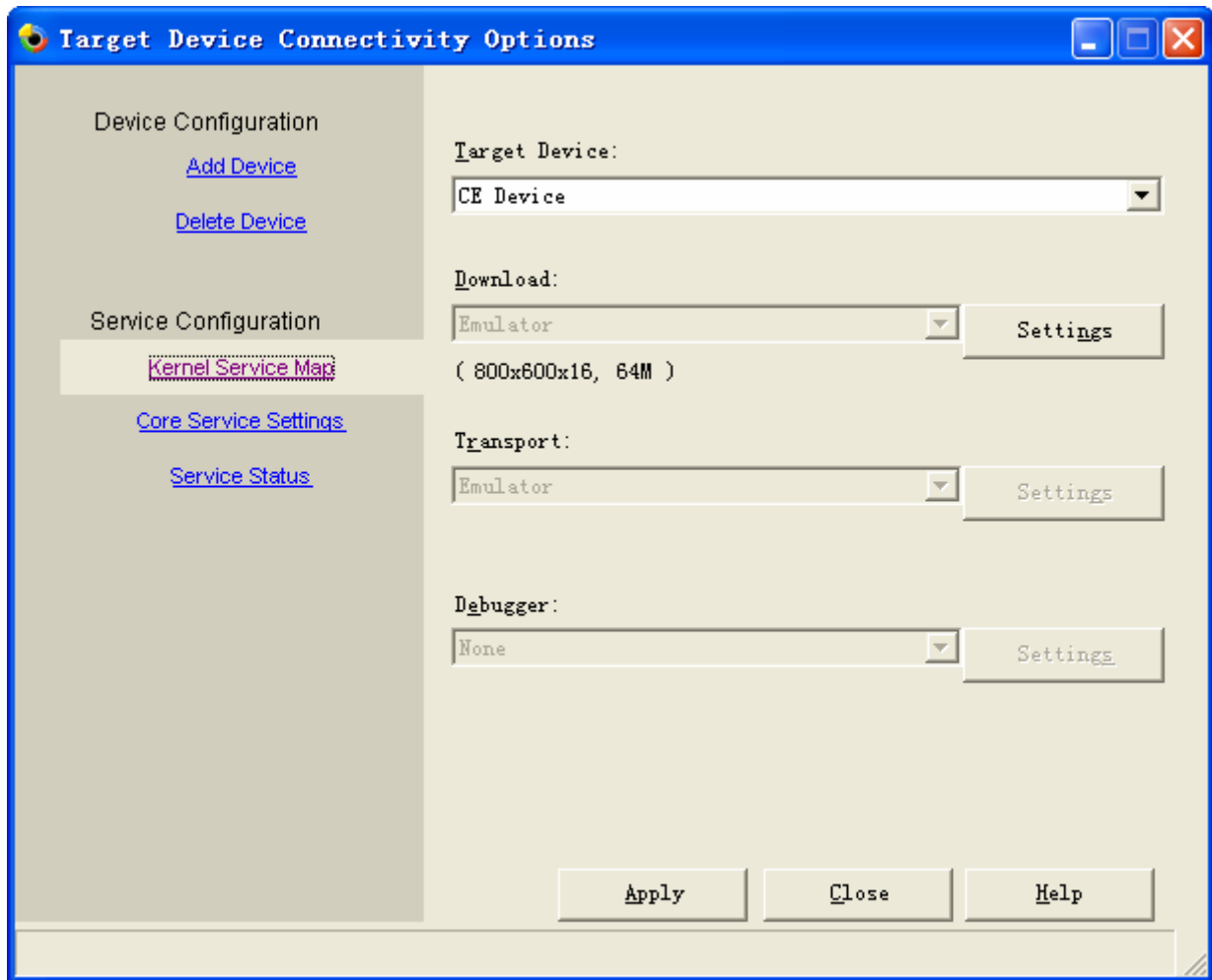
最后一步，就可以按“D”键，将 wince 映像文件 nk.nb0 下载到板子上。

按下“D”键后，2410 上运行的 EBOOT 程序开始尝试 PC 机上的 PB 程序进行联络，这时工作就要转移到 PB 程序上来。打开 PB 的菜单 Target | Configure Remote Connection，要设置一些远程连接选项，目的是和 GEC2410 进行网络连接，



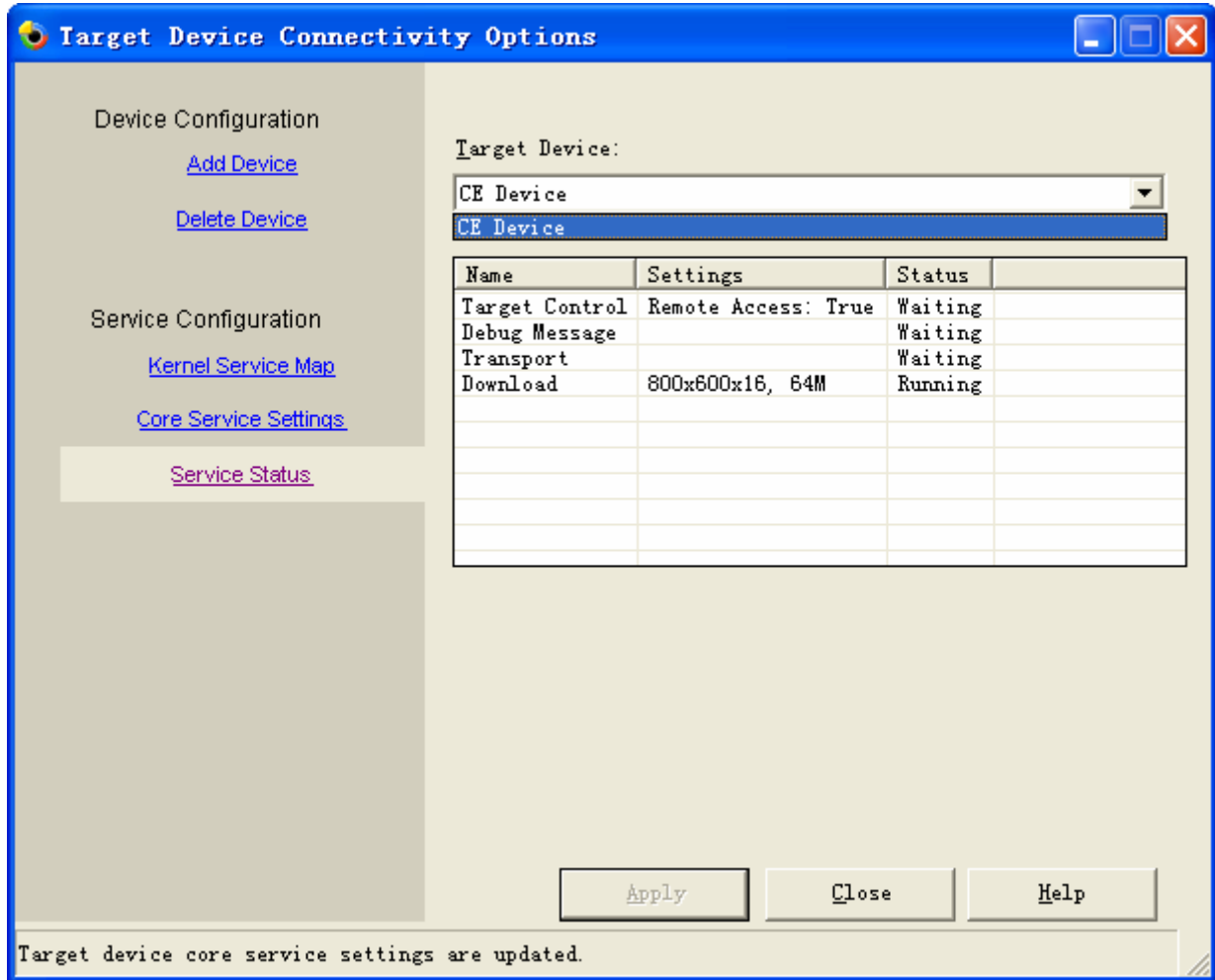
(图二十三)

在出现的配置对话框中点击“Services”标签页，然后在 active named connection 下拉列表中选择你的平台名，在“download”下拉列表中选择“Ethernet”，在“Kernel”下拉列表中也同样选择“Ethernet”，如下图



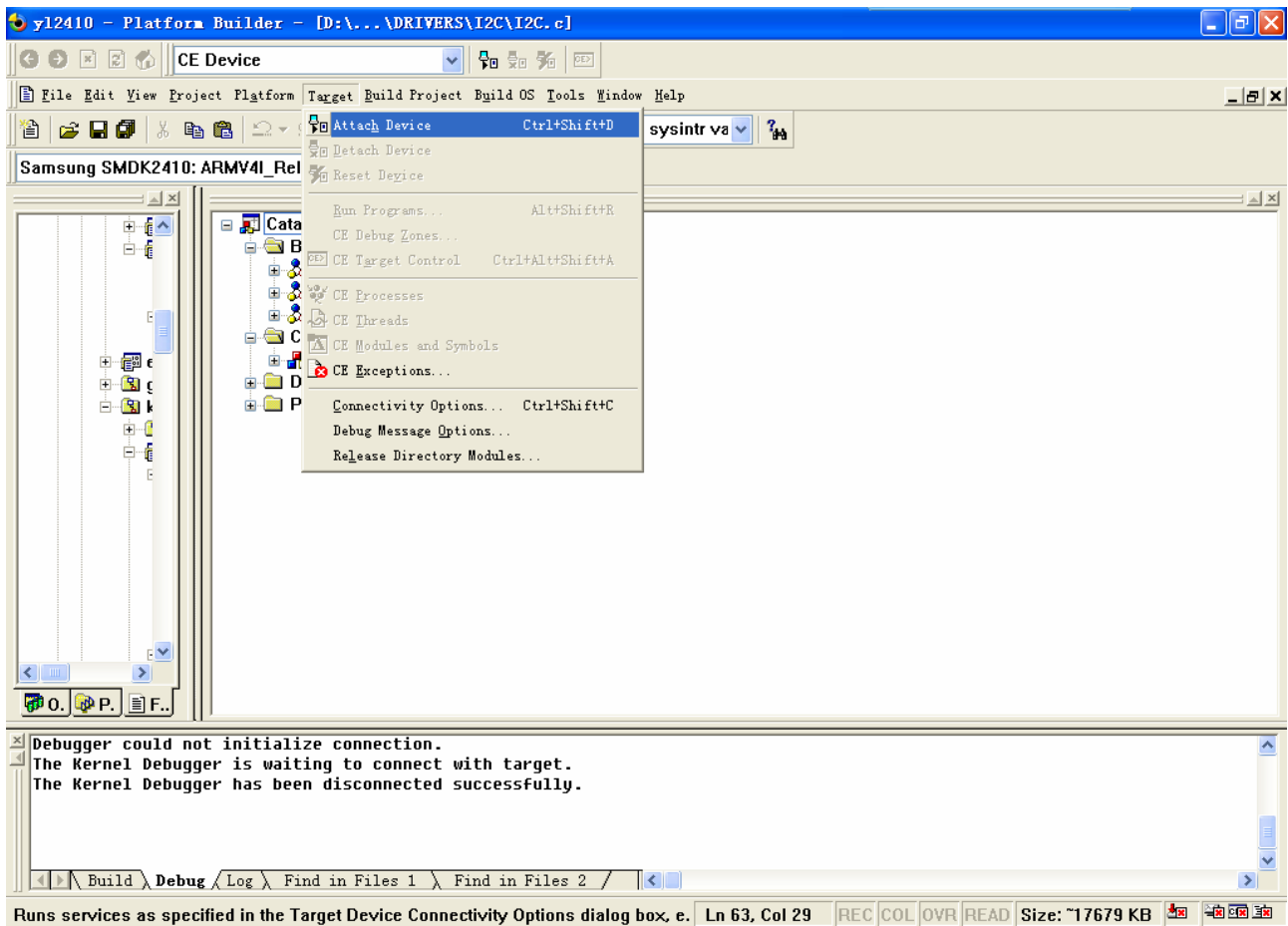
(图二十四)

接着，在这个对话框可以看到“Target Device”列表框列出了可用的远程连接设备名称，按“Apply”键退出，如下图，



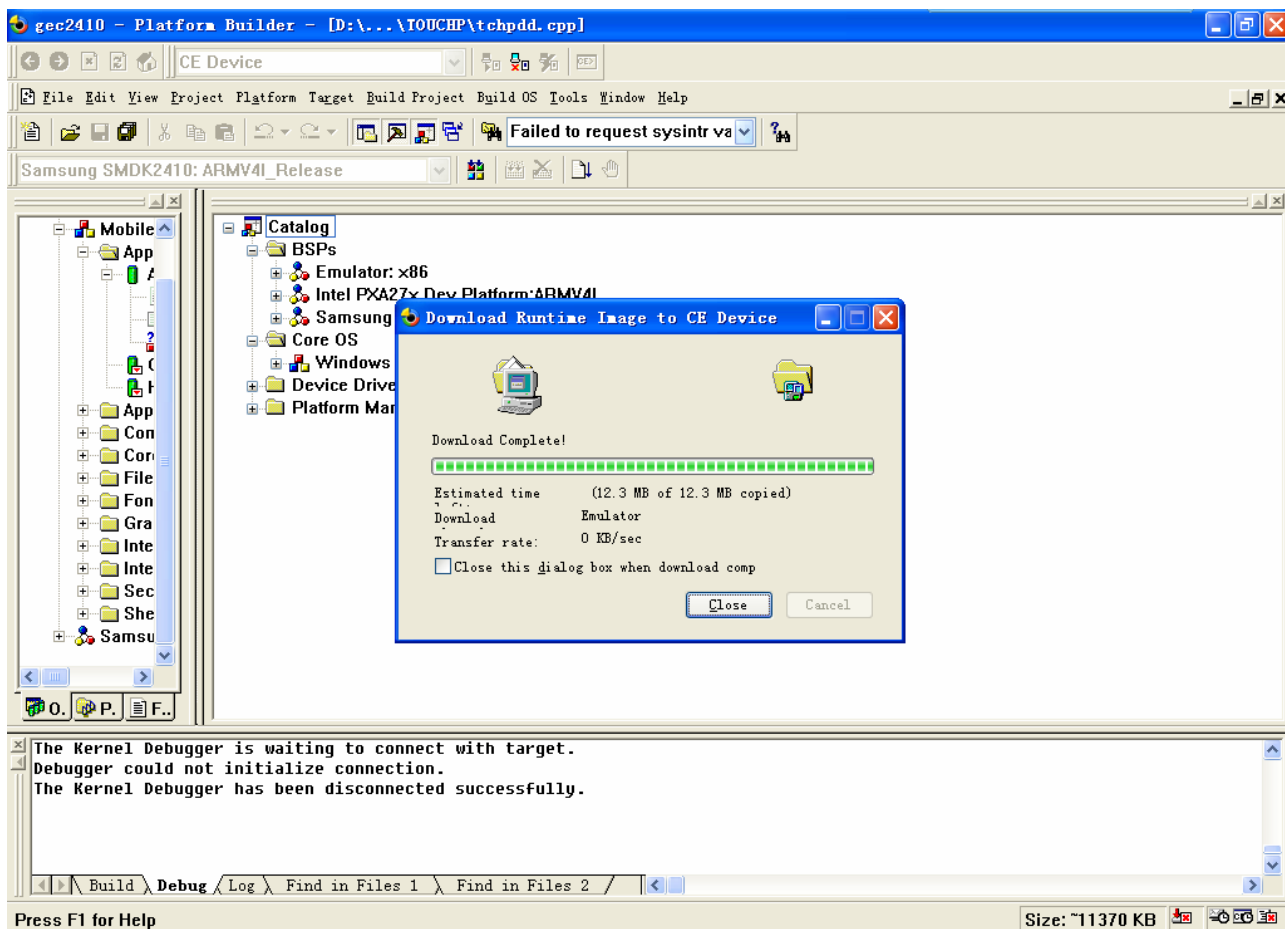
(图二十五)

在“Configure Remote Connection”对话框的配置结束后，按“确定”键退出，回到PB的主界面，然后点击 Target | Download/initialize 菜单，开始下载 NK.nb0,如下图



(图二十六)

下载过程如下图，



(图二十七)

下载完毕后，稍等一分钟左右的，会在 GEC2410 的 LCD 上看到运行起来的 wince 操作系统，以此同时，wince 操作系统映像也成功被烧写到 NAND FLASH 了。Wince 运行起来后，如果 windows 系统提示是否要在 PC 上安装一个和 wince 通讯的 USB 驱动程序，那么该驱动位于 bsp 包解压后的\SMDK2410\DRIVERS\USB\FUNCTION 目录下，请安装一下。

还有最后一步，为了让系统复位后能够自动从 NAND FLASH 引导 wince 操作系统，需要烧写“2410loader.bin”引导程序，2410loader.bin 这个文件是之前从光盘的 wince 目录的 wincebsp.rar 压缩文件中解压缩出来的，一般它位于“WINCE420\Platform\smdk2410\NBOOT”目录下。烧写过程如下：

首先要从 NOR FLASH 启动 bios（即将 JP1 短路块拔出来），因为 NAND FLASH 的 bios 已经被 wince 的格式化破坏掉了。

下一步运行 dnw.exe，选择菜单“configuration/option”，设置“UART/USB”选项的下载地址为 0x30038000。

设置完毕后，给 2410 复位以进入 bios 操作菜单界面，点击菜单，USB Port | Transmit, 选中要传输的文件为 2410loader.bin，然后开始传送 2410loader.bin 到 GEC2410 开发板中。

2410loader.bin 下载完毕后，程序会询问是否执行它，输入“n”，即不执行，然后按“2”键，再按“0”键，目的是将该文件烧写到 NAND 的 BOOT 区，烧写完毕后，断电，然后将 2410 上的 JP1 短路块接上，上电，从 NAND flash 启动，这时 wince 操作系统就运行起来了！（如下图）



（图二十八）

8.4 在 WINCE 和桌面系统之间建立通讯连接

可以使用微软公司提供的一些工具，并在进行了一些必要的设置后，就可以在安装了 WINCE 操作系统的移动设备和 windows 桌面系统之间进行通讯连接，从而可以实现文件上传下载，远程调试等功能。

首先下载 ActiveSync 工具的安装程序 MSASync.EXE（这个工具可以单独从网上下载），安装，运行。

然后开始对 ActiveSync 进行设置，点击菜单 File | Connection Settings，在弹出的设置对话框中，选择允许通过 USB 口、以太网口进行通讯连接，串口就不用选上了，以免同 DNW 冲突。如下图所示：



(图二十九)

ActiveSync 可以通过串口，USB 口，网口等方式建立连接，下面仅以 USB 连接方式为例介绍如何通过 ActiveSync 在桌面系统与 WINCE 之间建立通讯连接。

第一步，在 WinCE 系统启动后再连接上桌面系统的 USB 口和 GEC2410 的方形 USB 口间的 USB 线缆，这时 PC 机会弹出发现新硬件的对话框，添加 USB 驱动再按照以下步骤建立通讯，该驱动在 wince bsp 包解开后的 DRIVERS\USB\FUNCTION 里面有一个 inf 文件，设置 GEC2410 上运行的 wince 操作系统如下：

打开【我的电脑】—》打开【控制面板】—》打开【网络和拨号连接】—》点击【新建连接】，在“新建连接”设置对话框中，选择连接类型为“直接连接”，如下图：



(图三十)

点击“下一步”，在出现的“选择设备”下拉列表中选择“S3C2410 USB Cable: ” 如下图:



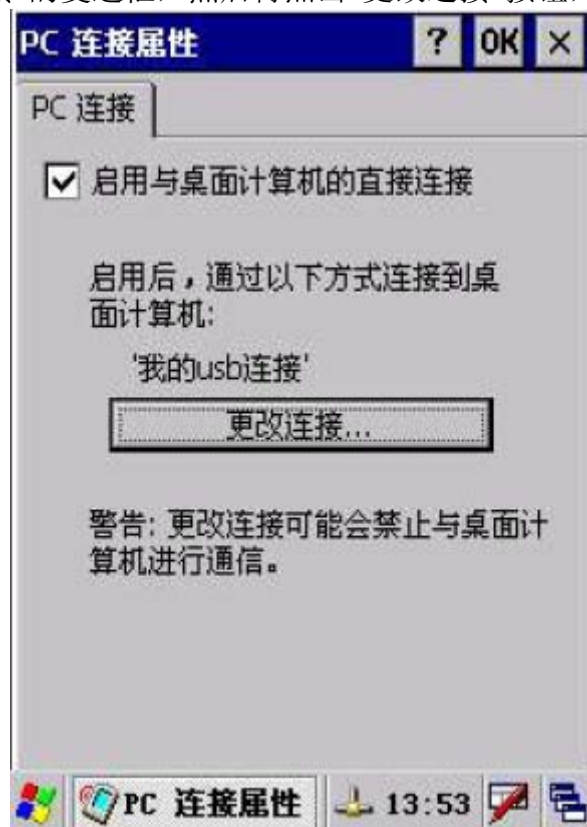
(图三十一)

点击完成，这时就出现“我的 USB 连接”图标，如下图:



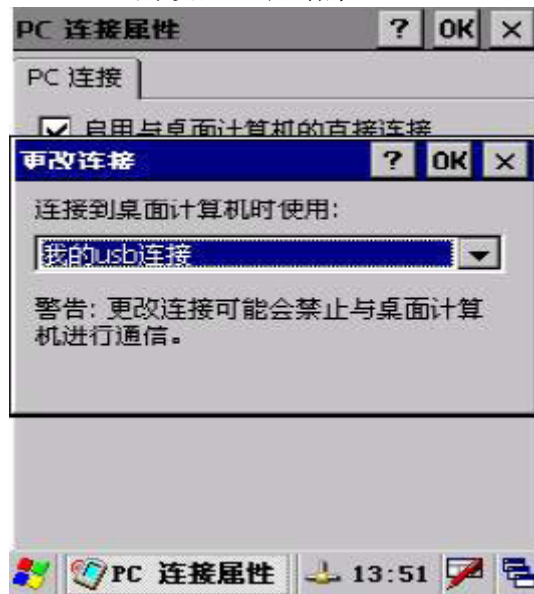
(图三十二)

回到“控制面板”，点击【PC 连接】图标，进入“PC 连接属性”设置对话框，选中“启用与桌面计算机的直接连接”的复选框，然后再点击“更改连接”按钮，如下图：



(图三十三)

在“更改连接”设置对话框的下拉列表框中选择刚才新建的连接“我的 USB 连接”，然后按“OK”键退出。这样 WINCE 这边的设置宣告结束。

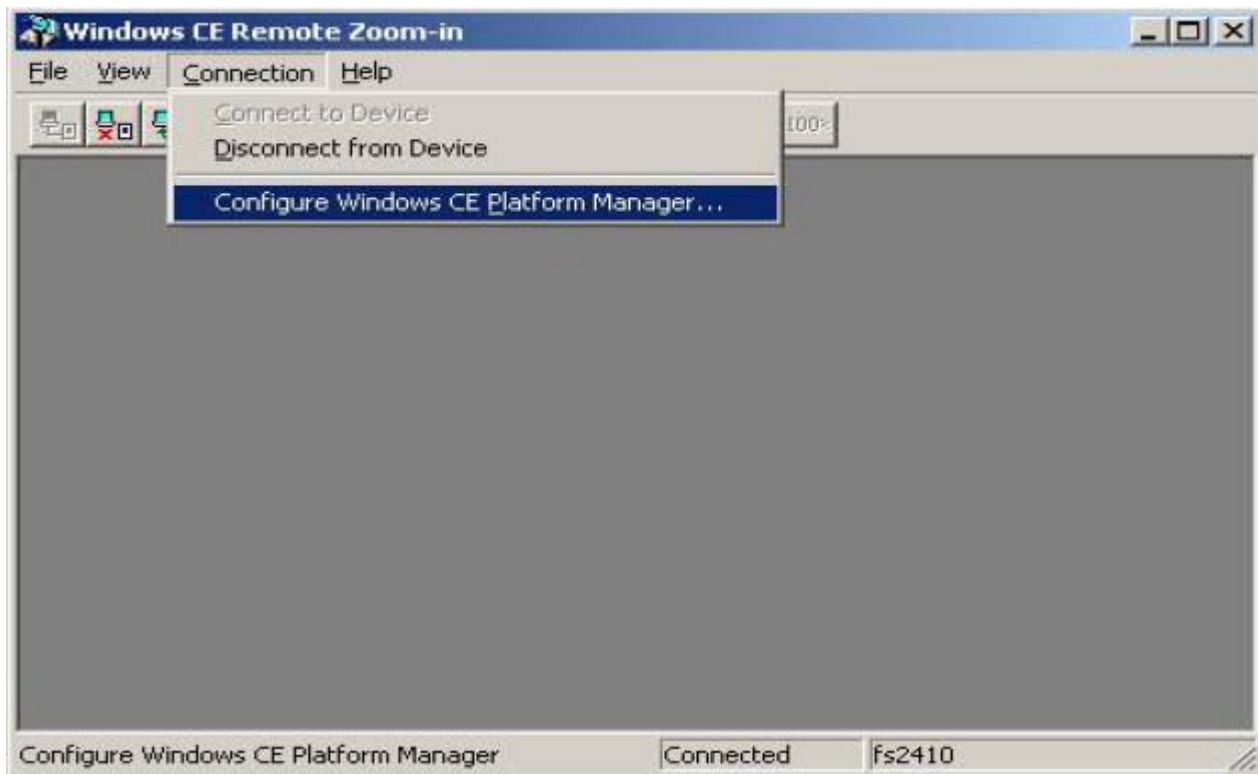


(图三十四)

最后，回到 PC 机上来，打开 ActiveSync 的 File | Get Connected 菜单项，开始尝试和 WINCE 操作系统进行连接

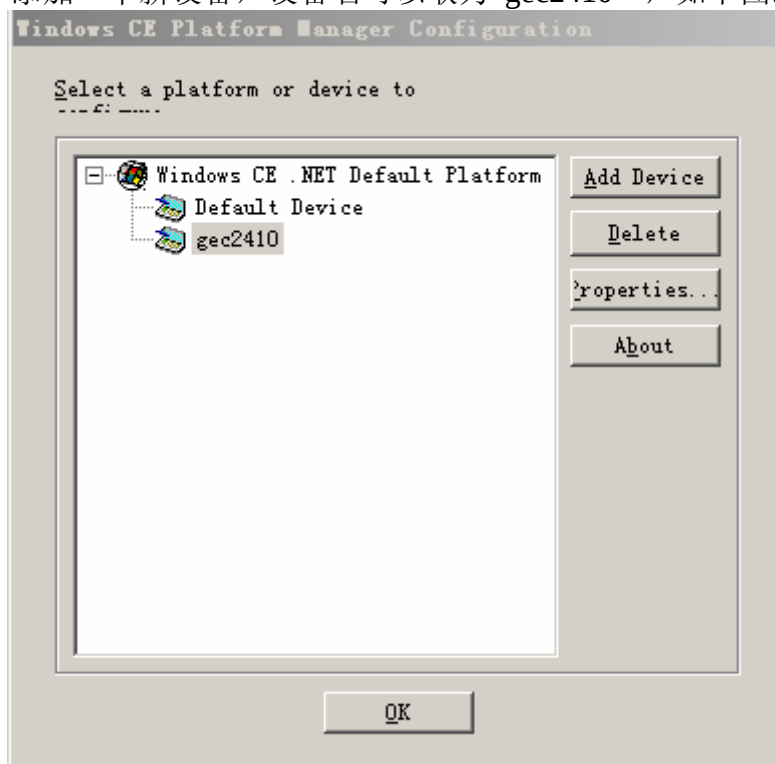
连接成功后，ActiveSync 的图标会变成另外一种颜色，并且提示连接成功。这时，打开菜单 File | Explore，就可以浏览 WINCE 系统上的资源，也可以通过复制/粘贴的方式在系统之间拷贝文件。

ActiveSync 的成功连接是使用所有微软远程连接工具的基础。ActiveSync 成功连接在后，就可以进一步使用到 PB 程序的“Tools”菜单项里面的许多远程工具。例如点击 PB 的 Tools | Remote zoom-in 菜单，以运行远程图片缩放工具，这个程序可以实现对远程移动设备显示屏幕的截屏。Remote zoom-in 程序运行起来后，首先要配置一下平台管理器，点击它的 Connection | Configure windows ce platform manager 菜单，如下图：



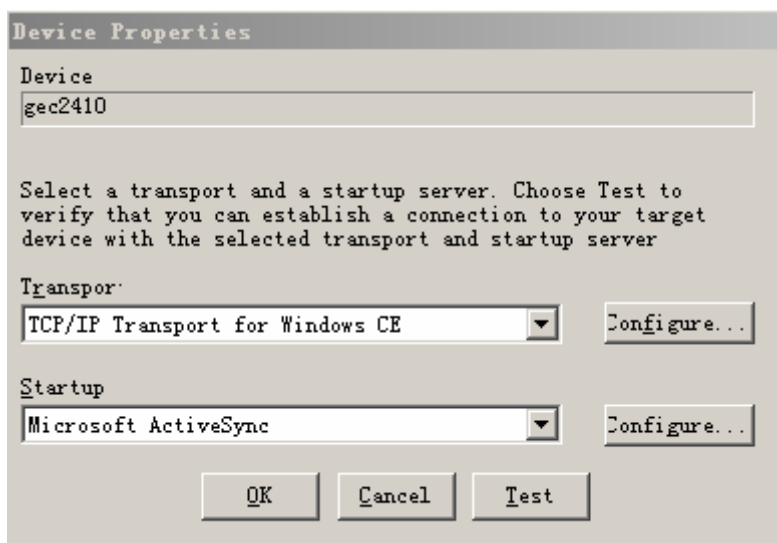
(图三十五)

在弹出的对话框中添加一个新设备，设备名可以取为“gec2410”，如下图:



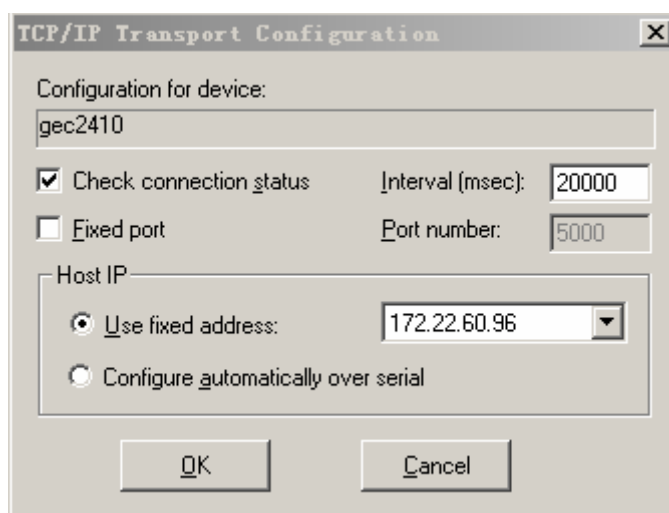
(图三十六)

设置“gec2410”设备的属性，如下图:



(图三十七)

点击“Transport”下拉框右边的“Configure”按钮，以设置 TCP/IP 传输的设置，HOST IP 一项填入您的桌面机器的与 WINCE 相连的以太网卡的 IP 地址，例如下图：



(图三十八)

设置结束后，确信您的网线已经连好 GEC2410 的网口和桌面机器的网卡，然后点击 Remotezoom-in 程序的菜单 File | New Bitmap ,开始进行截屏。

如果用 EVC 或 EVB 编译了自己编写的 WINCE 应用程序，并点击在“tools”菜单的“Configure platform manager”菜单项完成配置平台管理器后，点击运行按钮，就可以自动的把程序发送到板子上运行，此外也可以利用 sync 工具的 explorer 将编译好的程序上传到 GEC2410 上运行。以下是我方编写的例子程序“Hello”在 GEC2410 的 wince5.0 操作系统上执行的结果，如下图：



(图三十九)