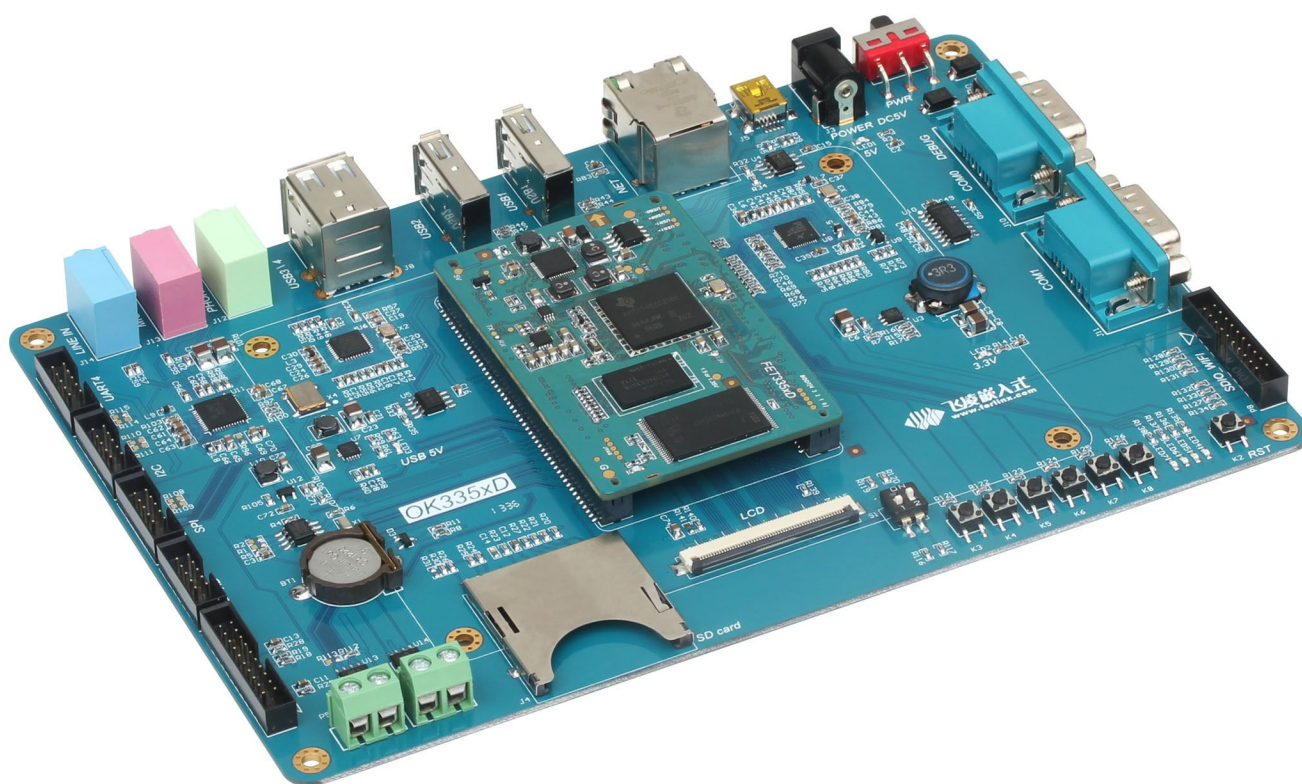




# OK335xD

## Linux 软件手册



Devoted to create the best embedded products

## 注意事项与维护

### 产品使用环境

供电电压：DC5V  $\pm$  10%

运行温度：商用级 0 – 80 °C

工业级 -40 – 85 °C

湿度：10 – 90%（不结露）



### 注意事项

- **请勿带电插拔核心板及外围模块！**
- 请遵循所有标注在产品上的警示和指引信息。
- 请保持本产品干燥。如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干。
- 使用中注意本产品的通风散热，避免温度过高造成元器件损坏。
- 请勿在多尘、脏乱的环境中使用或存放本产品。
- 请勿将本产品应用在冷热交替环境中，避免结露损坏元器件。
- 请勿粗暴对待本产品，跌落、敲打或剧烈晃动都可能损坏线路及元器件。
- 请勿使用有机溶剂或腐蚀性液体清洗本产品。
- 请勿自行修理、拆卸本公司产品，如产品出现故障请及时联系本公司进行维修。
- 擅自修改或使用未经授权的配件可能损坏本产品，由此造成的损坏将不予以保修。

如果产品出现故障，请联系飞凌技术服务部。

## 版权声明

本手册所有权由保定市飞凌嵌入式有限公司独家持有。未经本公司的书面许可，任何单位和个人不得以任何形式复制、传播、转载本手册的任何部分，否则一切后果由违者自负。

飞凌嵌入式

## 更新记录

日期	版本号	说明
2013.12	V1.3	1 增加 uboot 启动菜单和一键烧写进度显示 2 增加 QT 中文支持 3 增加系列图形测试程序 4 增加触摸 USB 鼠标切换 5 增加 USB 摄像头支持 6 增加 4.3/10.4 寸电阻屏和电容屏支持 7 增加 tftp 在 uboot 和 linux 使用文档
2013.10	V1.2	1 增加 5/7/8 寸电阻屏和电容屏支持 2 增加 uboot 命令设置传向内核的电阻、电容和屏幕尺寸参数 3 增加 QT 编译及编程文档支持 4 增加 USB WIFI 驱动和文档支持 5 增加 USB 3G 上网卡驱动和文档支持
2013.09	V1.1	1 增加 7 寸电容触摸屏的支持 2 增加 QT 图形文件系统 3 增加基于 Qt 的图形界面测试程序 4 增加最小文件系统，提供给不需要图形系统的用户使用
2013.08	V1.0	OK335xD-Linux 用户手册第一版



# 技术支持与更新

## 1 技术支持范围

- 1.1 本公司产品的软、硬件资源提供情况咨询；
- 1.2 本公司产品的软、硬件手册使用过程中遇到的问题；
- 1.3 本公司提供的 OEM、ODM 售后技术支持；
- 1.4 已购买本公司产品用户的资料丢失、更新后重新获取；
- 1.5 本公司产品的故障判断及售后维修服务；

## 2 技术讨论范围（非必解决问题）

- 2.1 源码的修改以及理解；
- 2.2 操作系统如何移植；
- 2.3 用户在自行修改以及开发中遇到的软硬件问题；

## 3 技术支持方式

- 3.1 电话（即时）：0312-3119192
- 3.2 邮箱（非即时）：
  1. Linux 技术支持：linux@forlinux.com
  2. Win CE 技术支持：wince@forlinux.com
  3. Android 技术支持：android@forlinux.com
- 3.3 论坛（非即时）：<http://bbs.witech.com.cn>

## 4 技术支持时间

周一至周五：

上午	9:00—11:30
下午	13:30—17:00

公司按照国家法定节假日安排休息，在此期间无法提供技术支持，有问题请发邮箱或论坛技术支持区，我们会在工作日尽快给您回复。

## 5 资料下载方法

请登陆“[bbs.witech.com.cn](http://bbs.witech.com.cn)”找到“[开发板资料下载](#)”选择对应平台下载

# 目 录

注意事项与维护.....	1
产品使用环境.....	1
注意事项.....	1
版权声明.....	2
更新记录.....	3
技术支持与更新.....	4
1 技术支持范围.....	4
2 技术讨论范围（非必解决问题）.....	4
3 技术支持方式.....	4
4 技术支持时间.....	4
5 资料下载方法.....	4
目 录.....	5
第一章 OK335xD 简介.....	8
第二章 运行 Linux 系统.....	9
2.1 制作 SD 卡.....	11
2.1.1 制作用于一键烧写 Linux 系统的 SD 卡.....	11
2.1.2 制作用于启动 LINUX 的 SD 卡.....	12
2.2 SD 卡启动 Linux.....	15
2.2.1 拷贝镜像到 SD 卡准备启动.....	15
2.3 NandFlash 启动 Linux.....	18
2.3.1 利用 SD 卡进行批量烧写.....	18
2.3.2 命令行烧写 Linux 系统到 NandFlash.....	18
2.3.3 单步更新系统.....	20
2.4 通过 Uboot 设置屏幕参数.....	21
2.4.1 更改屏幕类型.....	21
2.4.2 更改屏幕大小.....	23
第三章 OK335xD 功能测试.....	26
3.1 测试功能列表.....	26
3.2 准备工作.....	27
3.3 命令行测试.....	28
3.3.1 SD 卡的测试.....	28
3.3.2 按键测试.....	29
3.3.3 USB 测试.....	30

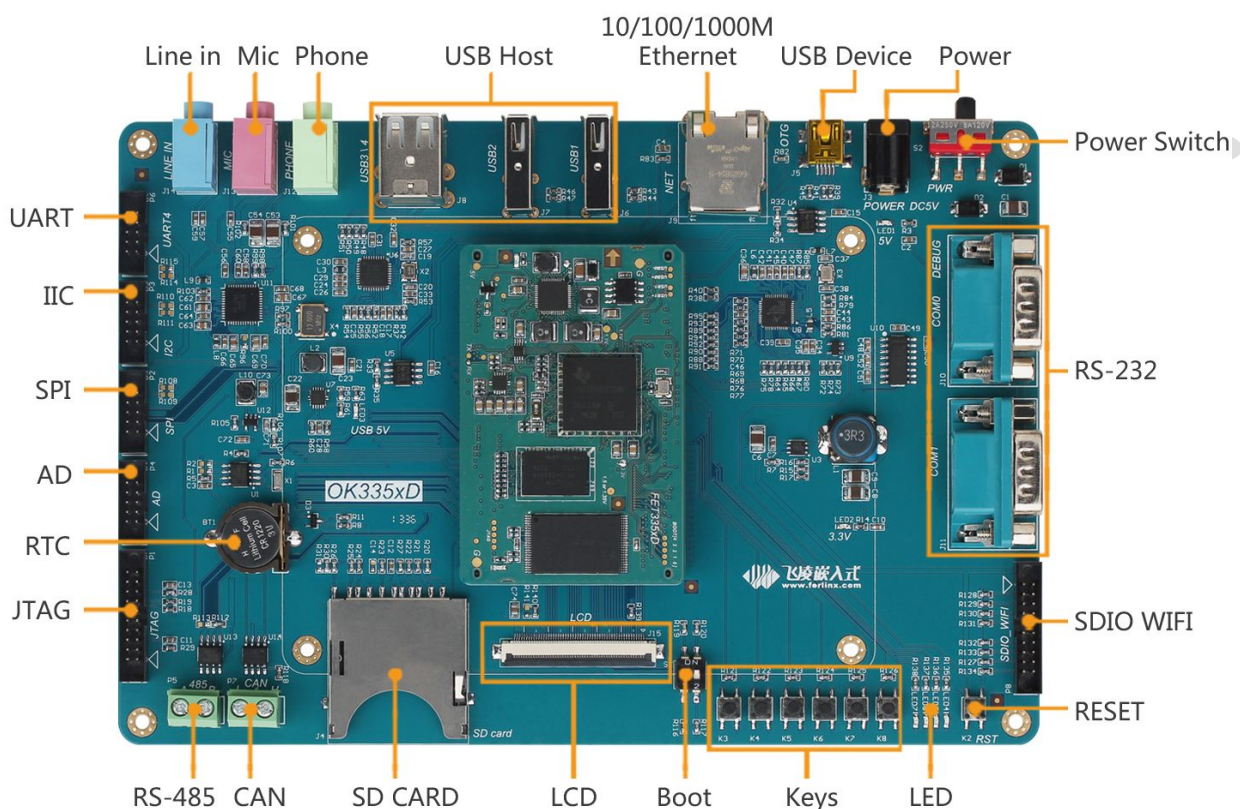
3.3.4	有线网卡测试.....	31
3.3.5	串口测试.....	32
3.3.6	音频测试.....	33
3.3.7	复位测试.....	39
3.3.8	RS485 测试.....	40
3.3.9	CAN 测试.....	44
3.3.10	SPI 测试.....	46
3.3.11	看门狗测试.....	47
3.3.12	LED 测试.....	48
3.3.13	背光测试.....	49
3.3.14	RTC 测试.....	50
3.3.15	EEPROM 测试.....	52
3.3.16	GPRS 测试.....	53
3.3.17	USB 3G 测试.....	56
3.3.18	USB WIFI 测试.....	62
2.3.19	USB 摄像头测试.....	71
3.4	图形界面测试.....	76
3.4.1	LCD 及触摸测试.....	76
3.4.2	按键测试.....	77
3.4.3	USB 测试.....	79
3.4.4	网卡配置.....	80
3.4.5	PING 测试.....	81
3.4.6	音频测试.....	82
3.4.7	视频测试.....	85
3.4.8	LED 测试.....	85
3.4.9	背光测试.....	86
3.4.10	串口测试.....	87
3.4.11	RS485 测试.....	88
3.4.12	SPI 测试.....	91
3.4.13	CAN 测试.....	92
3.4.14	RTC 测试.....	93
3.4.15	EEPROM 测试.....	94
3.4.16	看门狗测试.....	96
3.4.17	GPRS 测试.....	97
3.4.18	GPS 测试.....	99
3.4.19	WEB 测试.....	100
第四章	Linux 系统的编译.....	102
4.1	安装编译器.....	102
4.2	U-boot 的编译.....	103
4.2.1	普通版本 uboot 的编译.....	103
4.2.2	批量烧写版本的 uboot 编译.....	104
4.3	Linux 的编译.....	105
4.4	文件系统制作.....	106

4.5 驱动代码路径.....	107
第五章 Qt 开发指南.....	108
5.1 安装并设置交叉编译器环境.....	108
5.1.1 安装交叉编译器.....	108
5.1.2 设置交叉编译器环境变量.....	108
5.1.3 导入环境变量.....	108
5.2 移植 tslib.....	109
5.2.1 安装 autoconf、automake、libtool 包.....	109
5.2.2 拷贝压缩文件.....	109
5.2.3 移植 tslib 到 ok335x 目标板.....	109
5.2.4 导出 tslib 环境变量.....	110
5.2.5 在 ok335x 目标板下测试 tslib.....	110
5.3 移植 qt4.8.5.....	111
5.3.1 拷贝压缩文件.....	111
5.3.2 测试 arm 版 qt 是否安装并配置成功.....	112
5.3.3 移植 qt4.8.5 到 ok335x 目标板.....	112
5.3.4 导出 qt 环境变量.....	112
5.3.4 ok335x 目标板下测试 Qt4.8.5.....	113
5.4 Qt Creator 开发环境搭建及编程.....	114
5.4.1 安装 Qt Creator.....	114
5.4.2 设置 Qt Creator 使其支持交叉编译.....	114
5.4.3 新建 hello Qt 程序进行测试.....	115
5.4.4 在开发板上测试 qt 应用程序.....	118
附录 1: JTAG 调试.....	119
F1.1 CCS 简介.....	119
F1.2 安装 CCSV5.5.....	119
F1.3 连接开发板进行测试.....	120
F1.4 调试 SPL 代码.....	123
附录 2: TFTP 服务搭建.....	133
F2.1 安装服务器、客户端和守护进程.....	133
F2.2 服务器配置.....	133
F2.3 重新启动服务.....	133
F2.4 测试服务器.....	134
F2.5 Uboot 使用 tftp.....	134
F2.6 Linux 使用 tftp.....	135

## 第一章 OK335xD 简介

OK335xD 开发板基于 TI AM335X 处理器，运行主频最高 1G，支持 Linux，WinCE，Android 三大操作系统，及 StarterWare 裸机程序，可用于工业产品设计。OK335xD 由核心板和底板组成，核心板主要芯片有：CPU，NandFlash，Memory，PowerManage，EEPROM。使用我们的核心板，只需要根据您的业务需求开发自己的底板，这样可加速您的产品上市时间，让您从平台搭建的复杂环境中脱离。下面我们具体描述 OK335xD 核心板和底板资源。

下图为 OK335xD 接口简述，详细硬件介绍请参见硬件手册。





## 第二章 运行 Linux 系统

OK335xD 的所有软件都向用户开源，用户可以通过光盘获取软件和硬件的文档及源码, Linux 内核版本为 3.2.0，资料位于光盘目录 linux 文件夹下，以下文档中提到的所有目录（bin，src，filesystem，tools，doc 等）都以 **OK335xD 用户光盘\OK335xD 用户光盘（A）\Linux\**为根目录，

注意：我们提供了 QT 版本的文件系统 rootfs.tar.bz2 和非 QT 版本的文件系统 rootfs-mini.tar.bz，默认情况下烧写到 NandFlash 的文件系统为全功能文件系统，也即是带有 Qt 图形系统，如果您不使用 Qt 或不使用 LCD 屏幕，建议您使用 Mini 文件系统，该文件系统占用 NandFlash 空间小，烧写到 NandFlash 的速度快，方便您产品开发。

**问：如何设置拨码开关从 SD 卡启动或者 NandFlash 启动呢？**

答：

1. SD 卡启动设置： 1 On, 2 On (On 代表拨到上方，Off 代表拨到下方)
2. NandFlash 启动设置： 1 Off ,2 On (On 代表拨到上方，Off 代表拨到下方)

**问：如何查看内核的打印信息？**

答：

1. PC 和开发板的 COM0 通过开发板附带的串口线连接。
2. 打开超级终端进行如下的设置：

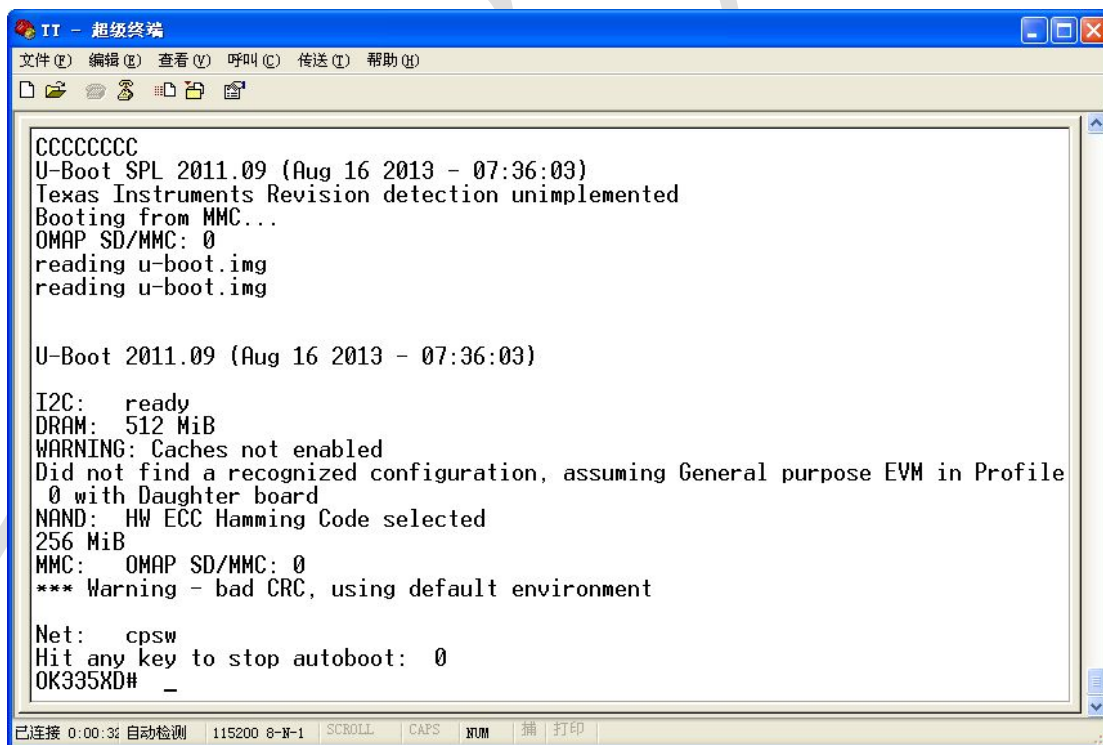


根据您的串口连接进行选择。





设置完成后，从 SD 卡或者 NandFlash 启动就可以看到调试信息了。



## 2.1 制作 SD 卡

### 2.1.1 制作用于一键烧写 Linux 系统的 SD 卡

OK335xD 开发板支持一键烧写功能，一键烧写的过程如下：

1. 用工具软件制作 SD 卡，SD 卡用普通的 4G SDHC 卡即可，推荐使用 ScanDisk .
2. 把要烧写的文件复制到 SD 卡中，把 SD 卡插入到开发板的卡槽
3. 把开发板的启动方式设置为 SD 卡启动，
4. 上电开机，系统会自动把 SD 卡中的文件烧写到 NandFlash。

目前 OK335xD 开发板的 Linux 系统，Android 系统均支持一键烧写，一键烧写过程中不需要 PC 参与，也不需要 LCD 显示，也不需要输入命令，烧写工作全部自动完成，适用于批量产品客户使用。

一键烧写需要制作专用的 SD 卡，下面是具体的制作过程：

1. 安装 SD 卡格式化工具
2. 请把 Ok335xD 用户光盘(A)\Tools 目录下的 HPUSBFW.EXE 安装到 PC 上,该工具支持 WindowXP, Win7 32 位，Win7 64 位操作系统.

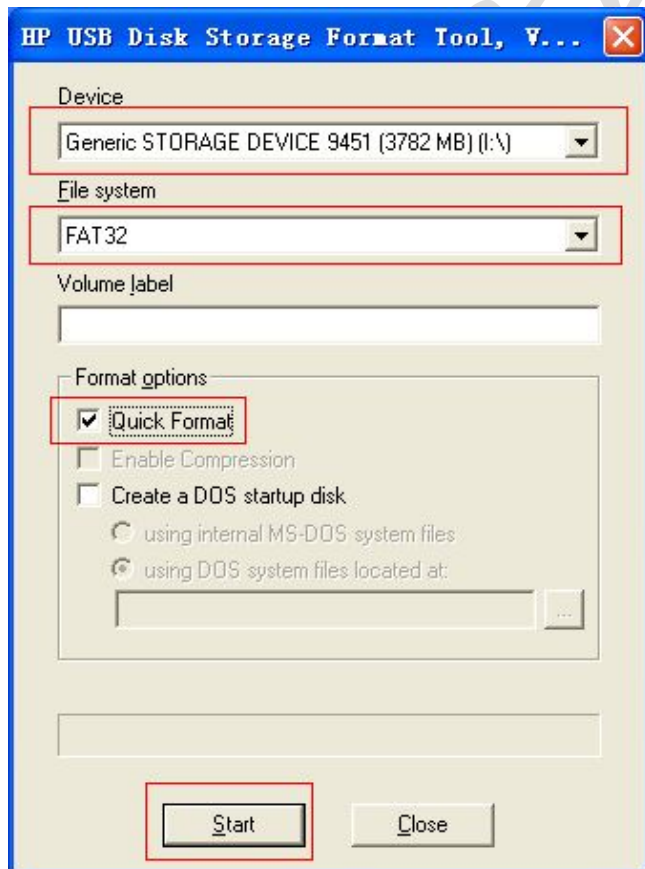
3. 把 SD 卡通过读卡器接入 PC

4. 运行 HPUSBFW 工具，会扫描到您的 SD 卡

注意：请确认 HPUSBFW 工具扫描到的 SD 卡 就是您的 SDHC 卡，以防出现误操作。

下图为运行 HPUSBFW 工具后的图片：

5. 选择“FAT32”系统格式
6. 点击“Start”
7. 等待格式化完成，点击“OK”



使用 SD 卡时，必须使用 HPUSBFW.EXE 工具或其  
它可将 SD 卡分区置为活动分区的进行格式化操  
作，否则设备无法识别 SD 卡信息。

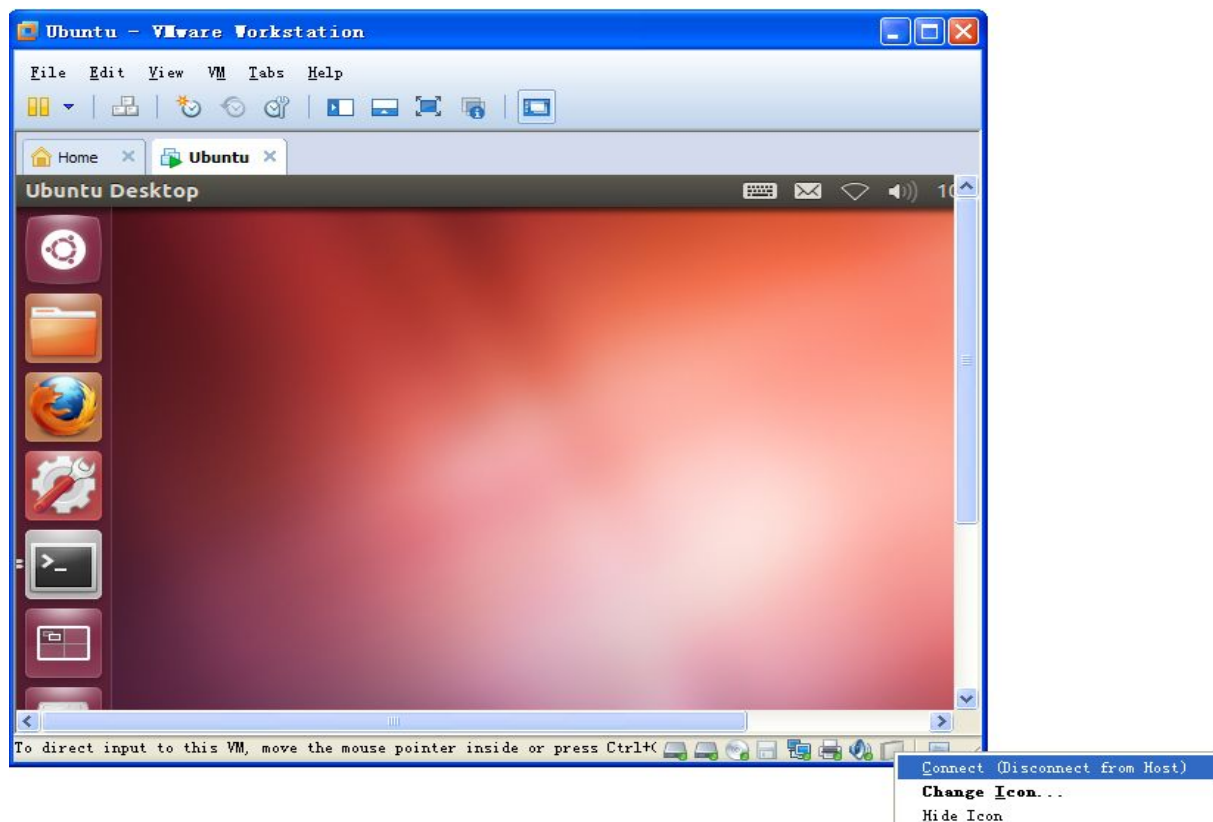


## 2.1.2 制作用于启动 LINUX 的 SD 卡

**注意：**该方法主要用于 SD 卡运行 Linux 系统，如果您不需要此功能，可以跳过该节。

**SD 卡运行 Linux 系统用于双系统备份，或者 NandFlash 空间容量有限的情况下使用。**

1. 将 tools 目录下 create-sdcard.sh 脚本拷贝到 ubuntu 系统的任一目录，假设为 /home/forlinux/work
2. 使用 USB 读卡器把 SD 卡插入到电脑的 USB 端口（VMware 虚拟机用户如果优盘没有被虚拟机识别，可以使用如下方式将优盘连接到虚拟机）



### 3. ./create-sdcard.sh (进入/home/forlinx/work 目录，执行脚本)

执行上述命令后，终端会列出电脑的硬盘或优盘，对应选择自己的 SD 卡（判定自己的优盘是 **sda/sdb/sdc** 可以根据容量进行判断，比如自己的优盘容量为 4G，则其 size 为 3872256 字节≈4G，建议用户执行此操作时不要同时插入多个优盘，以免混淆），回车。

```

root@forlinx: /usr/local/ti-sdk-am335x-evm/bin
File Edit View Search Terminal Help
#####

This script will create a bootable SD card from custom or pre-built binaries.

The script must be run with root permissions and from the bin directory of
the SDK

Example:
$ sudo ./create-sdcard.sh

Formatting can be skipped if the SD card is already formatted and
partitioned properly.

#####

Available Drives to write images to:

# major   minor   size   name
1:    8      16    3872256 sdb

Enter Device Number: 1

```

输入 y，回车。

```

root@ubuntu: /usr/local/ti-sdk-am335x-evm/bin
2:    8      32    3872256 sdc

Enter Device Number: 2

sdc was selected

Checking the device is unmounted
  unmounted /dev/sdc1
  unmounted /dev/sdc2

sdc1  sdc2  sdc3
72261 3791340

#####

Detected device has 2 partitions already

Re-partitioning will allow the choice of 2 or 3 partitions

#####

Would you like to re-partition the drive anyways [y/n] :

```



输入 2，选择创建 2 个分区，分区 1 为 FAT32 分区，分区 2 为 EXT3 分区

```

root@forlinx: /usr/local/ti-sdk-am335x-evm/bin
File Edit View Search Terminal Help

Re-partitioning will allow the choice of 2 or 3 partitions
#####
Would you like to re-partition the drive anyways [y/n] : y

Now partitioning sdb ...

#####

Select 2 partitions if only need boot and rootfs (most users)
Select 3 partitions if need SDK & CCS on SD card. This is usually used
by device manufacturers with access to partition tarballs.

****WARNING**** continuing will erase all data on sdb

#####

Number of partitions needed [2/3] : 

```

输入 n，这样就完成了创建。

```

root@forlinx: /usr/local/ti-sdk-am335x-evm/bin
File Edit View Search Terminal Help

8176 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

#####

Partitioning is now done
Continue to install filesystem or select 'n' to safe exit

**Warning** Continuing will erase files any files in the partitions

#####

Would you like to continue? [y/n] : n

```

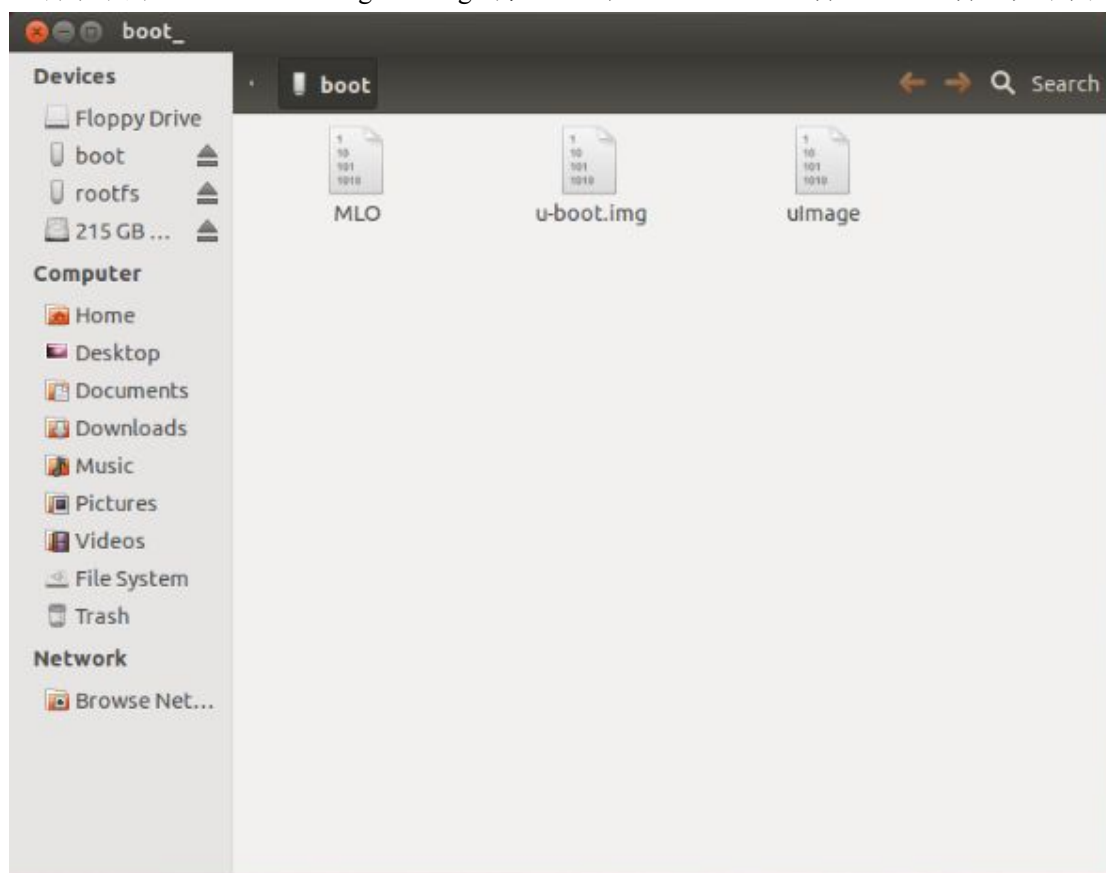
通过以上三个步骤就会在 SD 卡上创建两个分区，卷标分别为 boot(FAT32)、rootfs (ext3)，并且会自动挂载到 /media/boot 目录和 /media/rootfs 目录，用户可以直接使用。

## 2.2 SD 卡启动 Linux

**注意：**该方法主要用于 SD 卡运行 Linux 系统，如果您不需要此功能，可以跳过该节。  
SD 卡运行 Linux 系统可用于双系统备份，或者 NandFlash 空间容量有限的情况下使用。  
由于要从 SD 卡启动 Linux，所以 SD 要使用 2.1.2 的方式制作。

### 2.2.1 拷贝镜像到 SD 卡准备启动

1、将 bin 目录下的 MLO、u-boot.img、uImage 拷入 SD 卡 boot（FAT32）分区，boot 分区如下图：



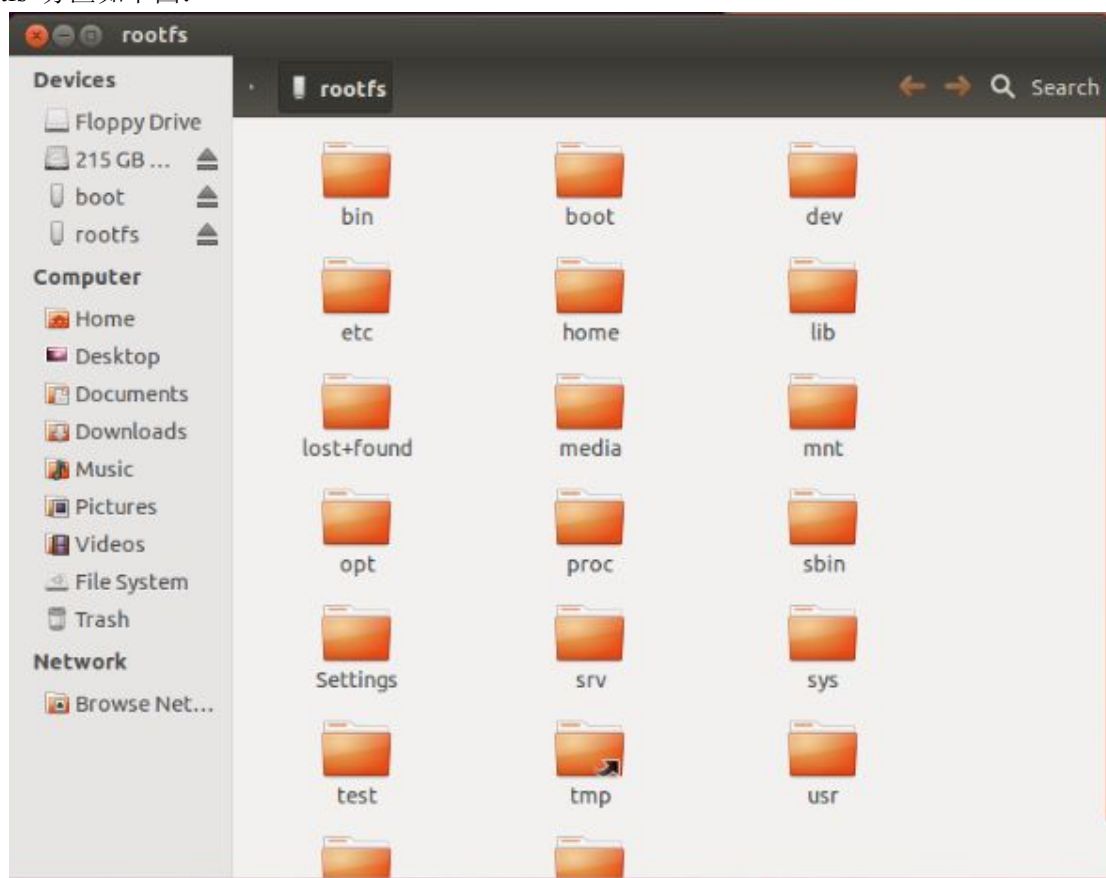


2、将 filesystem 目录下 rootfs.tar.bz2 拷贝到/home/forlinux/work 目录，执行以下命令

```
cd /home/forlinux/work
```

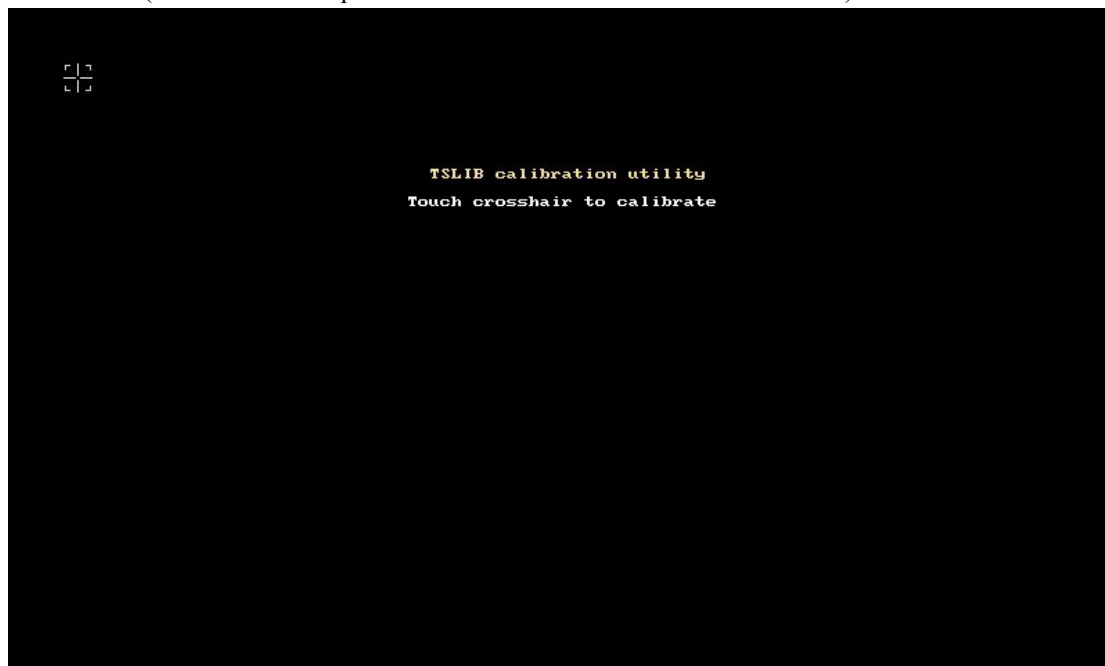
```
tar xvf rootfs.tar.bz2 -C /media/rootfs
```

rootfs 分区如下图：



3、将 SD 卡插入开发板的 SD 插槽里面，拨码开关拨到 SD 启动方式，上电启动。

首次启动需要校准(校准文件为/etc/pointercal，删除后重启系统可以重新校准):



进入系统后 lcd 将显示以下界面:



## 2.3 NandFlash 启动 Linux

SD 卡烧写 Linux 系统到 NandFlash 有两种方法，一种方法是在超级终端中手动输入烧写命令，另外一种方法不需要有手动干预，开机自动完成烧写功能，也就是一键烧写，这里推荐使用一键烧写方式。

**注意：由于 SD 仅仅用于烧写 NAND，故用户可以使用 2.1.1 或者 2.1.2 教程制作的 SD 卡**

### 2.3.1 利用 SD 卡进行批量烧写

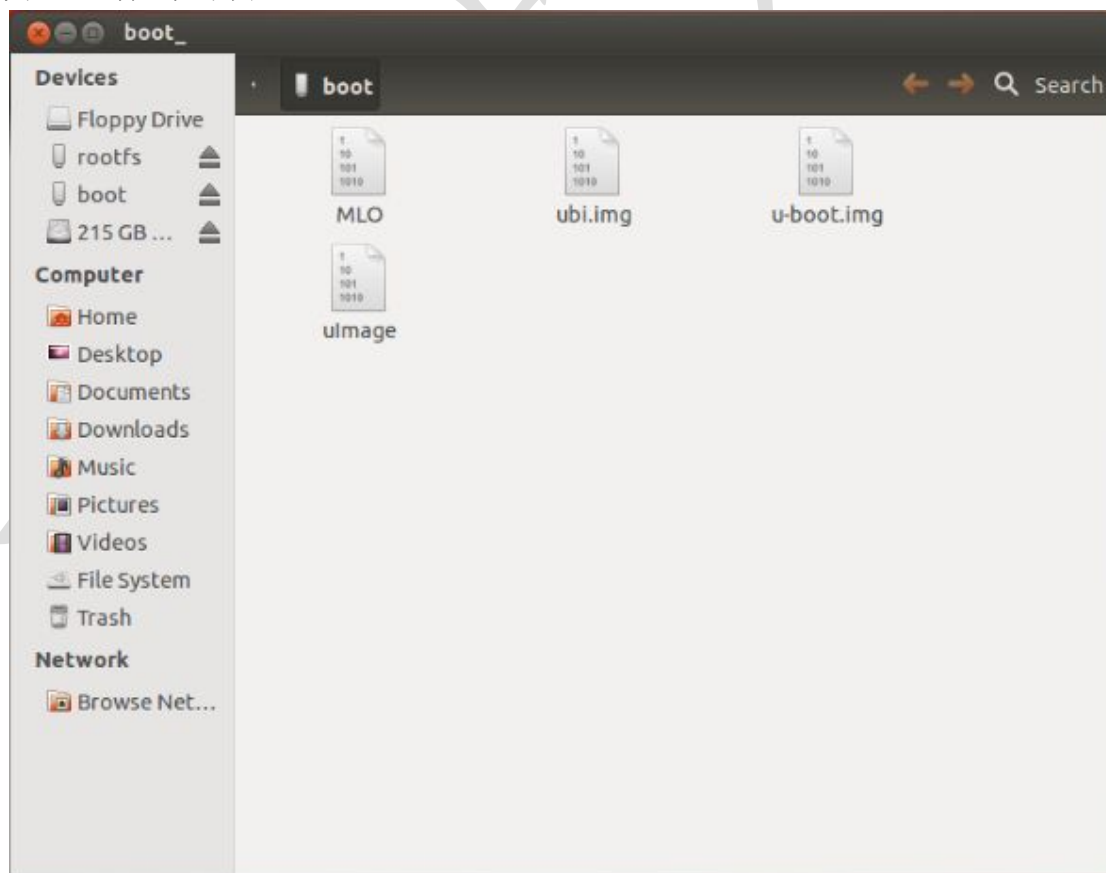
bin/batch 目录下提供了 SD 卡批量烧写 NAND 的 uboot 版本，使用该版本的 uboot 进行烧写，用户无需敲入任何命令。

- 1、将 bin/batch 目录下的 MLO、u-boot.img 以及 bin 目录下的 uImage、ubi.img 拷入 SD 卡 boot(FAT32) 分区。
- 2、将拨码开关拨到 SD 卡启动项。然后把 SD 卡插入到开发板的 SD 卡插槽中上电启动。
- 3、烧写过程中 led 灯会有流水效果，LCD 和串口会有进度显示，烧写成功后 led 长亮并且串口打印：  
Update system to nand success。

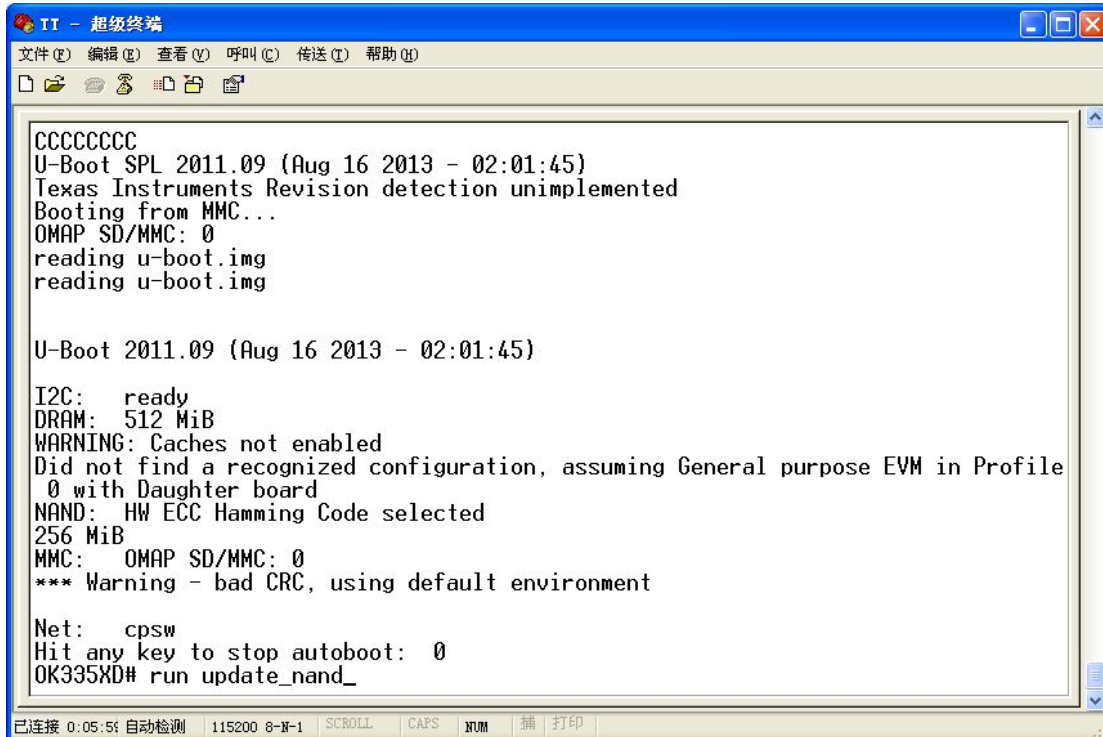
**注意：该模式下的 uboot 在烧写成功后便处于死循环状态，用户可断电拔卡后进行另一台设备烧写。批量烧写方式适用于批量产品客户，使用此方法不需要人工干预，不需要连接 PC，一张 SD 卡即可把系统快速烧写到设备中。**

### 2.3.2 命令行烧写 Linux 系统到 NandFlash

- 1、将 bin 目录下的 MLO、u-boot.img、uImage、ubi.img 拷入 SD 卡 boot (FAT32) 分区，此步骤与 rootfs 分区无关，boot 分区如下图：



- 2、拨码开关拨到 SD 启动项，把 SD 卡插入到开发板的 SD 卡插槽中，开机进入 U-boot 命令行，执行 “run update\_nand” 命令，如下图：



```

II - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]
CCCCCCCC
U-Boot SPL 2011.09 (Aug 16 2013 - 02:01:45)
Texas Instruments Revision detection unimplemented
Booting from MMC...
OMAP SD/MMC: 0
reading u-boot.img
reading u-boot.img

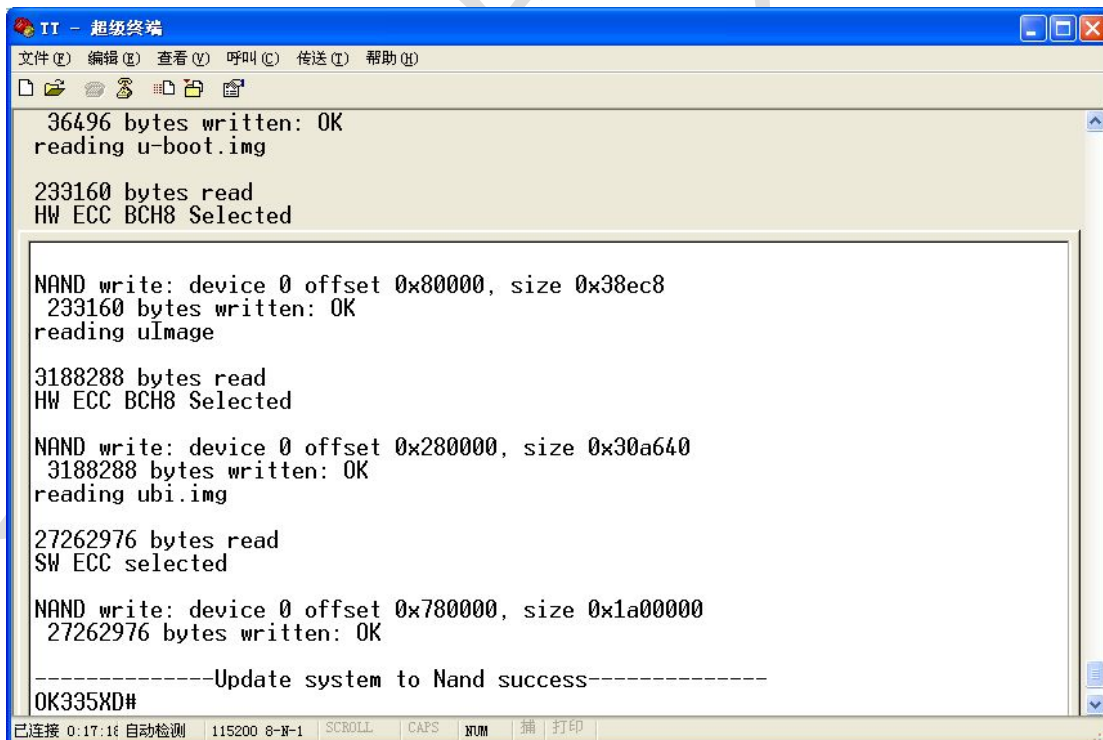
U-Boot 2011.09 (Aug 16 2013 - 02:01:45)

I2C: ready
DRAM: 512 MiB
WARNING: Caches not enabled
Did not find a recognized configuration, assuming General purpose EVM in Profile
0 with Daughter board
NAND: HW ECC Hamming Code selected
256 MiB
MMC: OMAP SD/MMC: 0
*** Warning - bad CRC, using default environment

Net: cpsw
Hit any key to stop autoboot: 0
OK335XD# run update_nand_

```

- 3、串口信息显示 Update system to nand success 表示烧写完成，如下图：



```

II - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]
36496 bytes written: OK
reading u-boot.img

233160 bytes read
HW ECC BCH8 Selected

NAND write: device 0 offset 0x80000, size 0x38ec8
233160 bytes written: OK
reading uImage

3188288 bytes read
HW ECC BCH8 Selected

NAND write: device 0 offset 0x280000, size 0x30a640
3188288 bytes written: OK
reading ubi.img

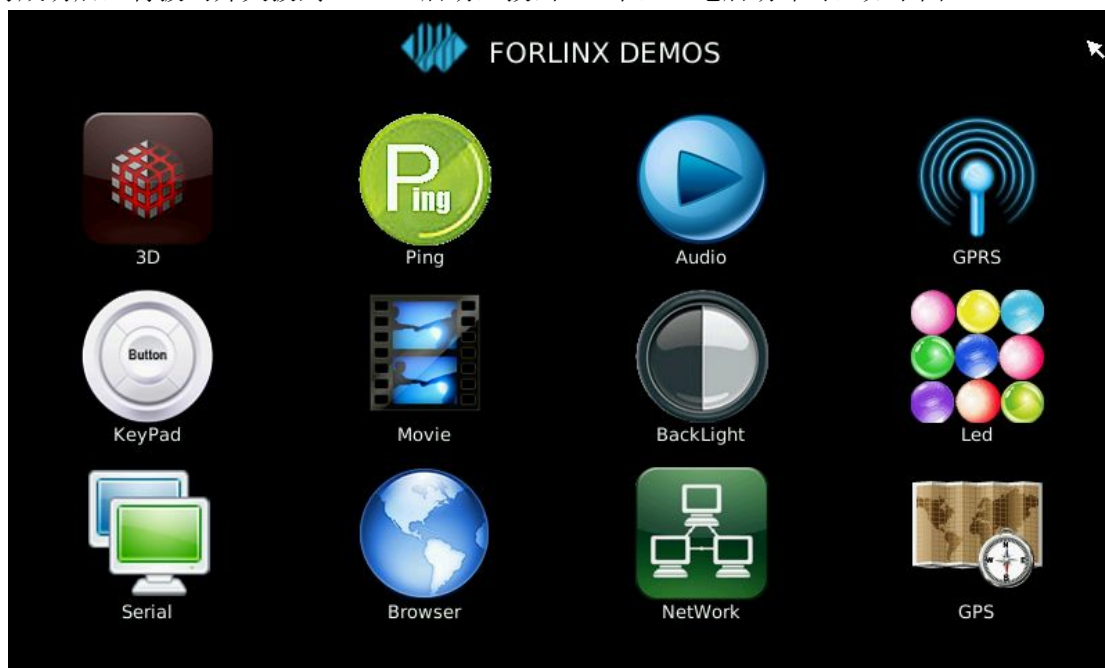
27262976 bytes read
SW ECC selected

NAND write: device 0 offset 0x780000, size 0x1a00000
27262976 bytes written: OK

-----Update system to Nand success-----
OK335XD#

```

4、烧写成功后，将拨码开关拨到 NAND 启动，拔出 SD 卡，上电启动即可，如下图：



### 2.3.3 单步更新系统

单步更新系统是指更新 nand 中的单个文件而不是整个系统，用户只需将要更新的文件拷入 SD 卡，然后设置开发板从 SD 卡启动，进入 uboot 命令行后执行以下操作：

#### 1. 更新 MLO, u-boot.img 文件

```
mmc rescan
fatload mmc 0 0x82000000 MLO
nand erase 0x0 0x20000
nandeccl hw 2
nand write.i 0x82000000 0x0 0x20000
mmc rescan
fatload mmc 0 0x82000000 u-boot.img
nand erase 0x80000 0x40000
nandeccl hw 2
nand write.i 0x82000000 0x80000 0x40000
```

#### 2. 更新内核文件

```
mmc rescan
fatload mmc 0 0x82000000 uImage
nand erase 0x280000 0x500000
nandeccl hw 2
nand write.i 0x82000000 0x280000 0x380000
```

#### 3. 更新文件系统

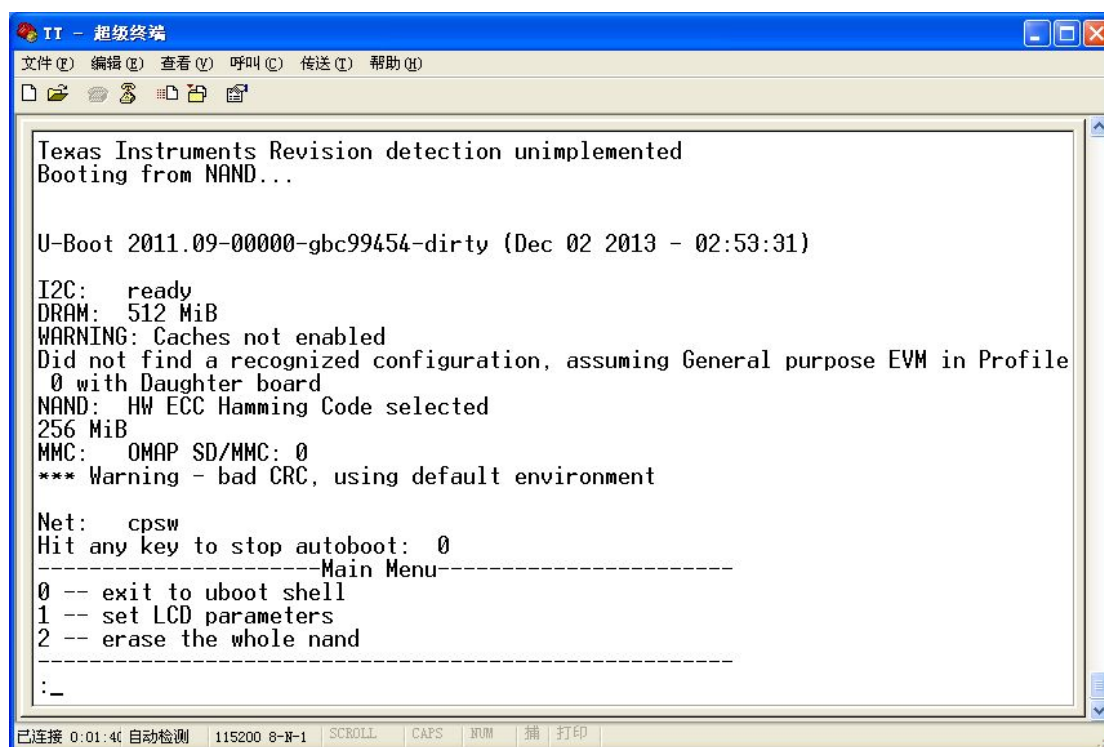
```
mmc rescan
fatload mmc 0 0x82000000 ubi.img
nand erase 0x780000 0xC800000
nandeccl sw
nand write.i 0x82000000 0x780000 0xC800000
```

## 2.4 通过 Uboot 设置屏幕参数

### 2.4.1 更改屏幕类型

OK335xD 产品默认支持电阻屏，如果用户需要使用电容屏可以通过 uboot 进行参数设置，方法如下：

一. Uboot 启动时按任意键进入如下菜单：



```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

Texas Instruments Revision detection unimplemented
Booting from NAND...

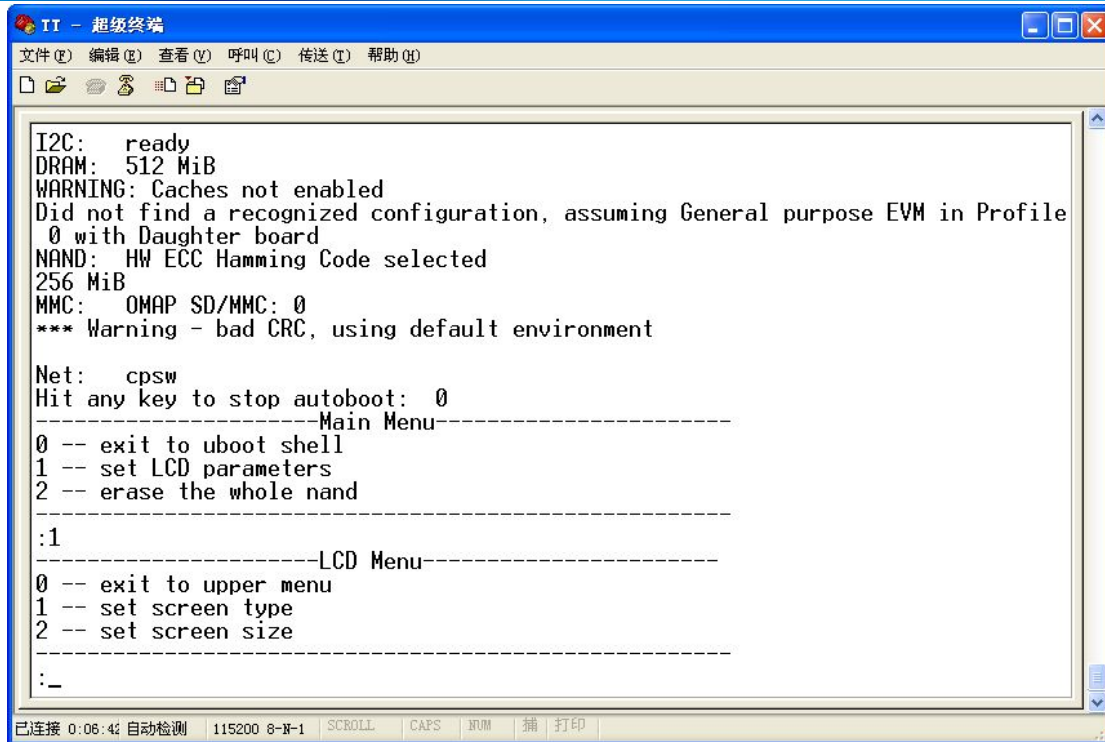
U-Boot 2011.09-00000-gbc99454-dirty (Dec 02 2013 - 02:53:31)

I2C:  ready
DRAM:  512 MiB
WARNING: Caches not enabled
Did not find a recognized configuration, assuming General purpose EVM in Profile
0 with Daughter board
NAND:  HW ECC Hamming Code selected
256 MiB
MMC:   OMAP SD/MMC: 0
*** Warning - bad CRC, using default environment

Net:   cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand
-----
:~
    
```

二. 输入 1 进入如下界面：





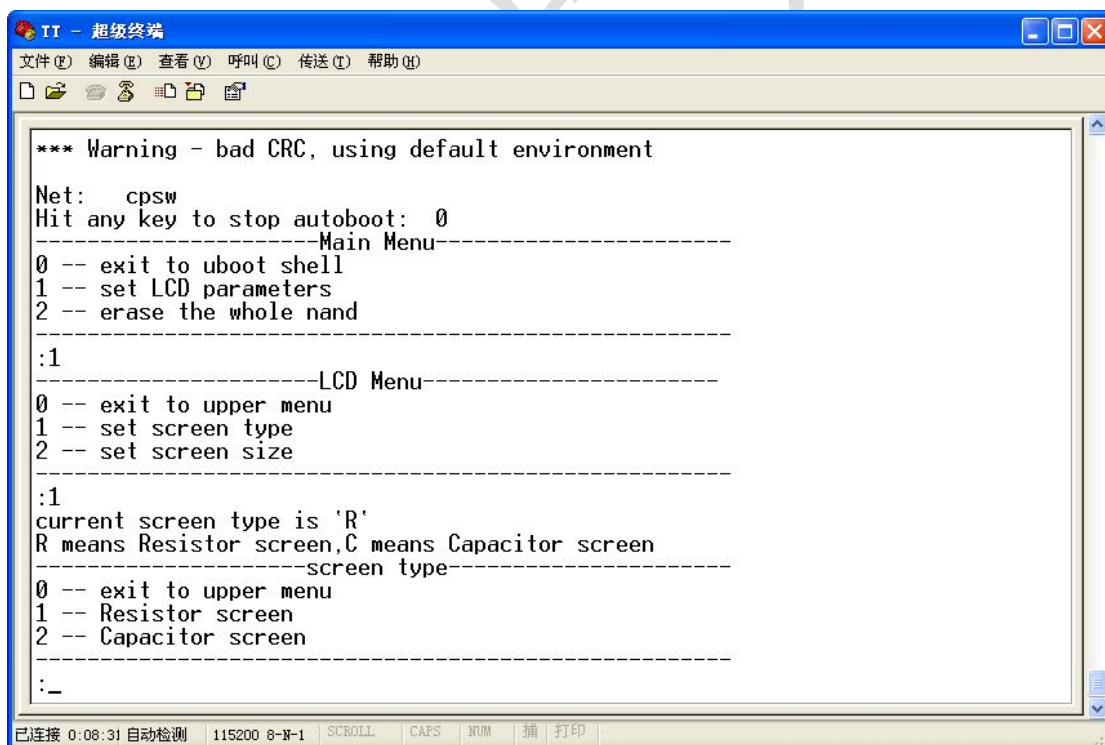
```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
I2C:  ready
DRAM:  512 MiB
WARNING: Caches not enabled
Did not find a recognized configuration, assuming General purpose EVM in Profile
0 with Daughter board
NAND:  HW ECC Hamming Code selected
256 MiB
MMC:   OMAP SD/MMC: 0
*** Warning - bad CRC, using default environment

Net:   cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand
-----
:1
-----LCD Menu-----
0 -- exit to upper menu
1 -- set screen type
2 -- set screen size
-----
:_
已连接 0:06:42 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

三. 输入 1 进入类型（电容或电阻）设置菜单，程序打印出当前的屏幕类型和供设置的类型菜单：



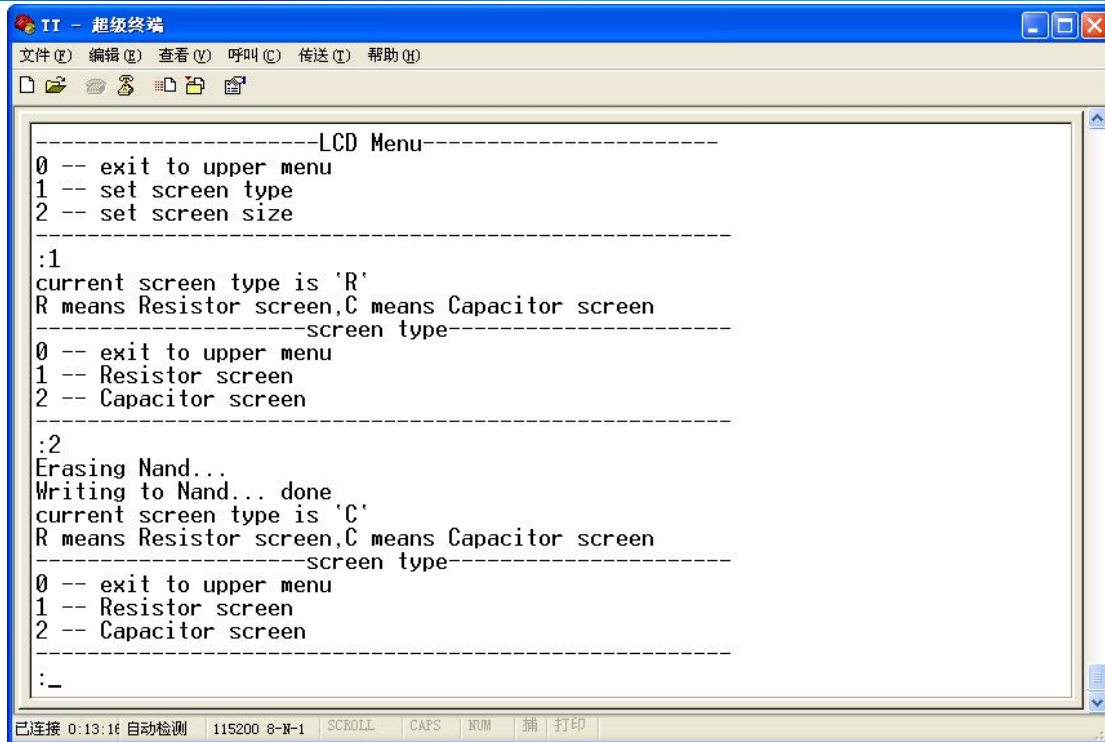
```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
*** Warning - bad CRC, using default environment

Net:   cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand
-----
:1
-----LCD Menu-----
0 -- exit to upper menu
1 -- set screen type
2 -- set screen size
-----
:1
current screen type is 'R'
R means Resistor screen,C means Capacitor screen
-----screen type-----
0 -- exit to upper menu
1 -- Resistor screen
2 -- Capacitor screen
-----
:_
已连接 0:08:31 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

四. 假如用户要设置为电容屏，则输入 2 会显示如下界面，表示设置电容屏成功：



```

-----LCD Menu-----
0 -- exit to upper menu
1 -- set screen type
2 -- set screen size

:1
current screen type is 'R'
R means Resistor screen,C means Capacitor screen
-----screen type-----
0 -- exit to upper menu
1 -- Resistor screen
2 -- Capacitor screen

:2
Erasing Nand...
Writing to Nand... done
current screen type is 'C'
R means Resistor screen,C means Capacitor screen
-----screen type-----
0 -- exit to upper menu
1 -- Resistor screen
2 -- Capacitor screen

:~

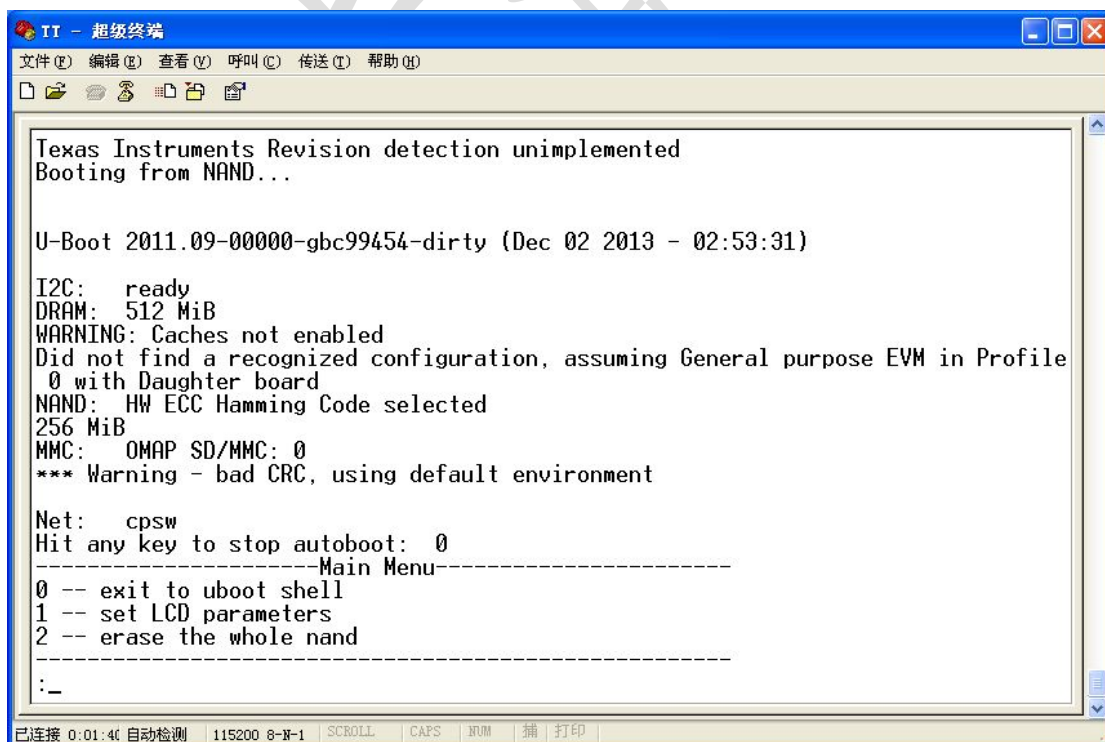
```

注：设置成功后重启系统，或者退回到 uboot 命令行执行 bootd 命令即可使用新参数启动系统。

## 2.4.2 更改屏幕大小

OK335xD 产品默认支持 7 寸屏，如果用户需要使用 4/5/8/10 寸屏幕可以通过 uboot 进行参数设置，方法如下：

一. Uboot 启动时按任意键进入如下菜单：



```

Texas Instruments Revision detection unimplemented
Booting from NAND...

U-Boot 2011.09-00000-gbc99454-dirty (Dec 02 2013 - 02:53:31)

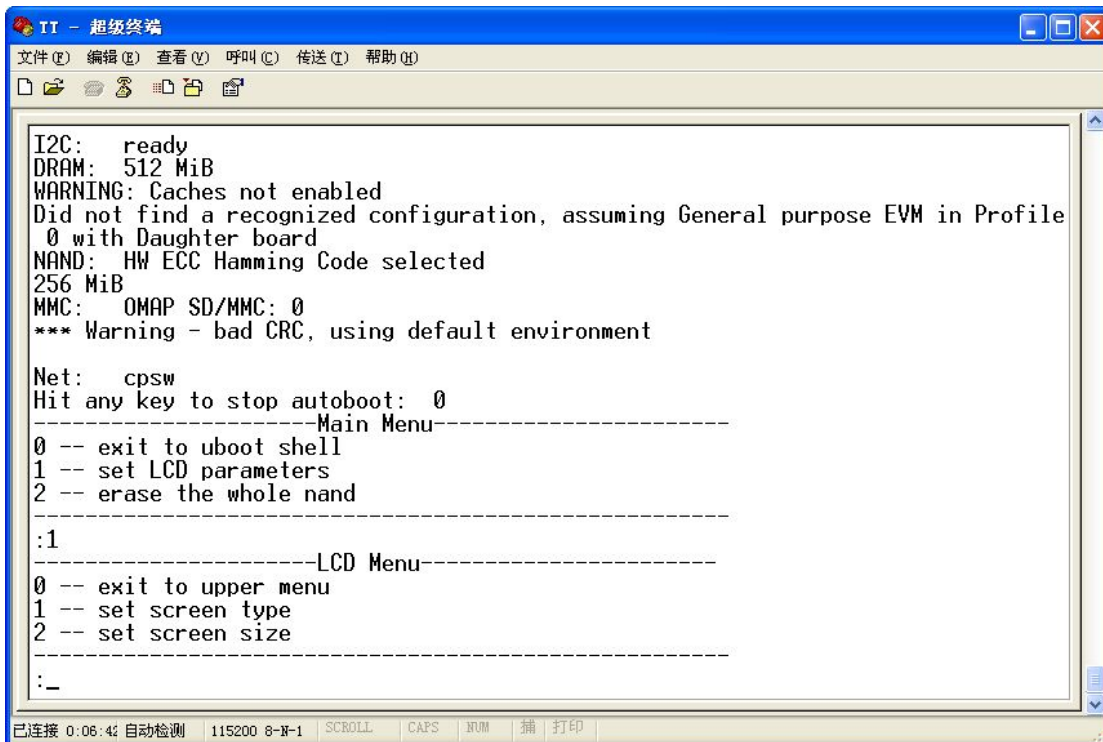
I2C:  ready
DRAM:  512 MiB
WARNING: Caches not enabled
Did not find a recognized configuration, assuming General purpose EVM in Profile
0 with Daughter board
NAND:  HW ECC Hamming Code selected
256 MiB
MMC:   OMAP SD/MMC: 0
*** Warning - bad CRC, using default environment

Net:   cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand

:~

```

二. 输入 1 进入如下界面:



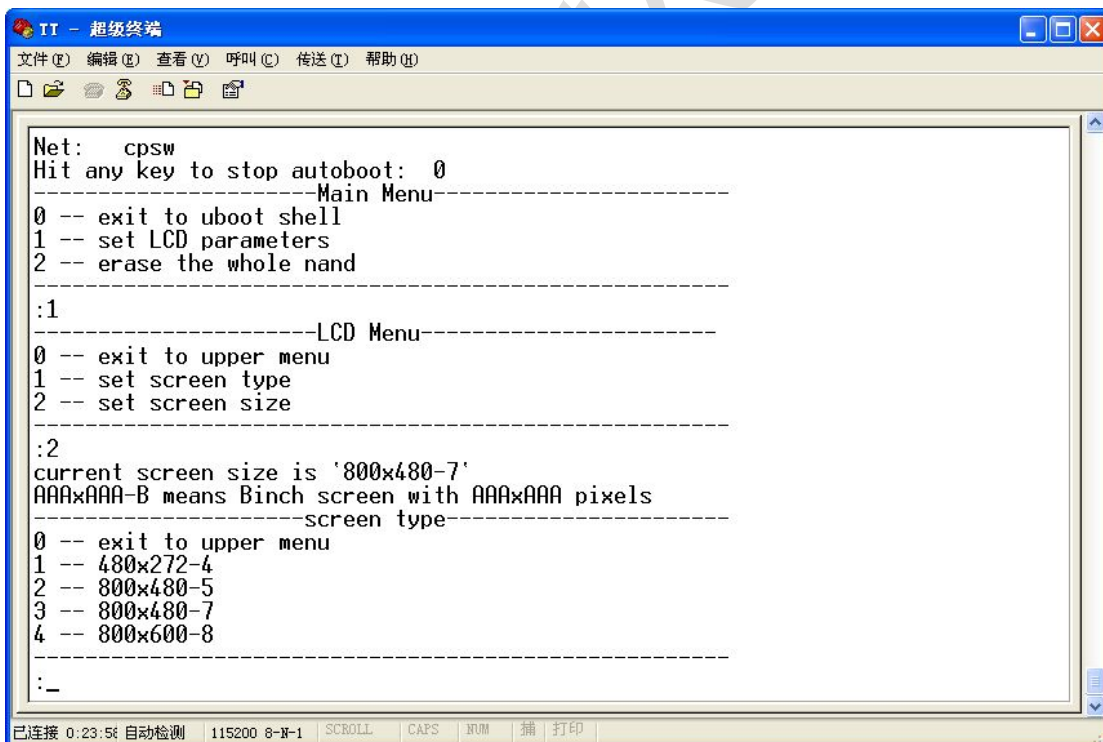
```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
I2C:  ready
DRAM: 512 MiB
WARNING: Caches not enabled
Did not find a recognized configuration, assuming General purpose EVM in Profile
0 with Daughter board
NAND:  HW ECC Hamming Code selected
256 MiB
MMC:  OMAP SD/MMC: 0
*** Warning - bad CRC, using default environment

Net:  cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand
-----
:1
-----LCD Menu-----
0 -- exit to upper menu
1 -- set screen type
2 -- set screen size
-----
:_
已连接 0:06:42 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印

```

三. 输入 2 进入屏幕大小设置菜单, 程序打印出当前的屏幕大小和供设置的大小菜单:

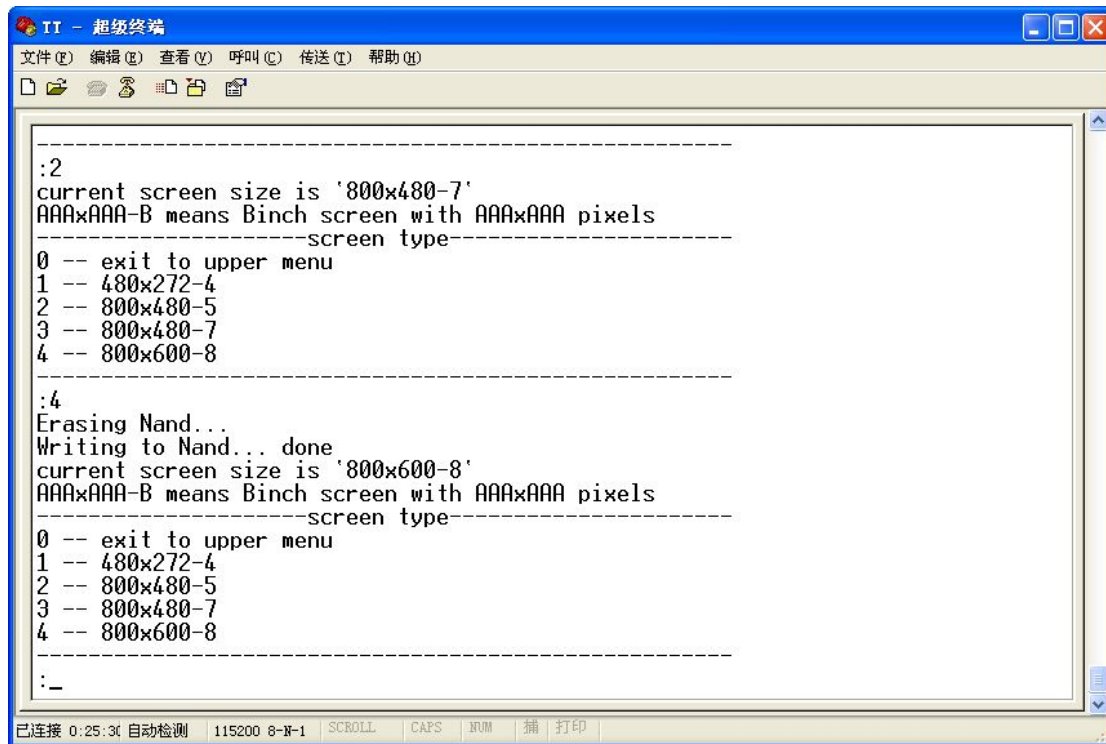


```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
Net:  cpsw
Hit any key to stop autoboot:  0
-----Main Menu-----
0 -- exit to uboot shell
1 -- set LCD parameters
2 -- erase the whole nand
-----
:1
-----LCD Menu-----
0 -- exit to upper menu
1 -- set screen type
2 -- set screen size
-----
:2
current screen size is '800x480-7'
AAAA-B means Binch screen with AAAxAAA pixels
-----screen type-----
0 -- exit to upper menu
1 -- 480x272-4
2 -- 800x480-5
3 -- 800x480-7
4 -- 800x600-8
-----
:_
已连接 0:23:56 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印

```

四. 假如用户要设置为 8 寸屏, 则输入 4 会显示如下界面, 表示设置 8 寸屏成功:



```

TI - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
-----
:2
current screen size is '800x480-7'
AAA×AAA-B means Binch screen with AAA×AAA pixels
-----screen type-----
0 -- exit to upper menu
1 -- 480x272-4
2 -- 800x480-5
3 -- 800x480-7
4 -- 800x600-8
-----
:4
Erasing Nand...
Writing to Nand... done
current screen size is '800x600-8'
AAA×AAA-B means Binch screen with AAA×AAA pixels
-----screen type-----
0 -- exit to upper menu
1 -- 480x272-4
2 -- 800x480-5
3 -- 800x480-7
4 -- 800x600-8
-----
:_
已连接 0.25:36 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕捉 打印

```

注: 设置成功后重启系统, 或者退回到 uboot 命令行执行 bootd 命令即可使用新参数启动系统。

## 第三章 OK335xD 功能测试

### 3.1 测试功能列表

OK335xD 底板需要测试的功能如下：

命令行测试：

- [1.SD 卡测试](#)
- [2.按键测试](#)
- [3.USB 测试](#)
- [4.有线网卡测试](#)
- [5.串口测试](#)
- [6.音频测试](#)
- [7.复位测试](#)
- [8.RS485 测试](#)
- [9.CAN 测试](#)
- [10.SPI 测试](#)
- [11.看门狗测试](#)
- [12.LED 测试](#)
- [13.背光测试](#)
- [14.RTC 测试](#)
- [15.EEPROM 测试](#)
- [16.GPRS 测试](#)
- [17.USB 3G 测试](#)
- [18.USB WIFI 测试](#)
- [19.USB 摄像头测试](#)

图形测试:

[1.LCD 测试（含触摸屏测试）](#)

[2.按键测试](#)

[3.USB 测试](#)

[4.网卡测试](#)

[5.PING 测试](#)

[6.音频测试](#)

[7.视频测试](#)

[8.LED 测试](#)

[9.背光测试](#)

[10.串口测试](#)

[11.RS485 测试](#)

[12.SPI 测试](#)

[13.CAN 测试](#)

[14.RTC 测试](#)

[15.EEPROM 测试](#)

[16.看门狗测试](#)

[17.GPRS 测试](#)

[18.GPS 测试](#)

[19.WEB 测试](#)

## 3.2 准备工作

1. 一块工作良好的 OK335xD 核心板。
2. 用于测试 USB 功能的 USB 鼠标或者 U 盘。
3. 测试网卡的网线。
4. 测试音频播放的耳机, Mic.
5. 串口测试工装:  
需要 DB9 母头一个, 2, 3 短路, 用于测试 COM1。  
2.0 mm 间距, 20 Pin 双排针插头 1 个, 同样收发需要短接, 用于测试 TTL 电平的 UART。
6. RS485 测试有两种方法一种需要两块 OK335xD 板子, 一种方法需要串口转换 RS485 模块, 模块与电脑串口相连, PC 端需要串口调试助手。
7. Can 测试需要两根普通的导线, 连接两块开发板的 Can 总线 H,L 接口。
8. 纽扣电池, 用于开发板 RTC 测试。

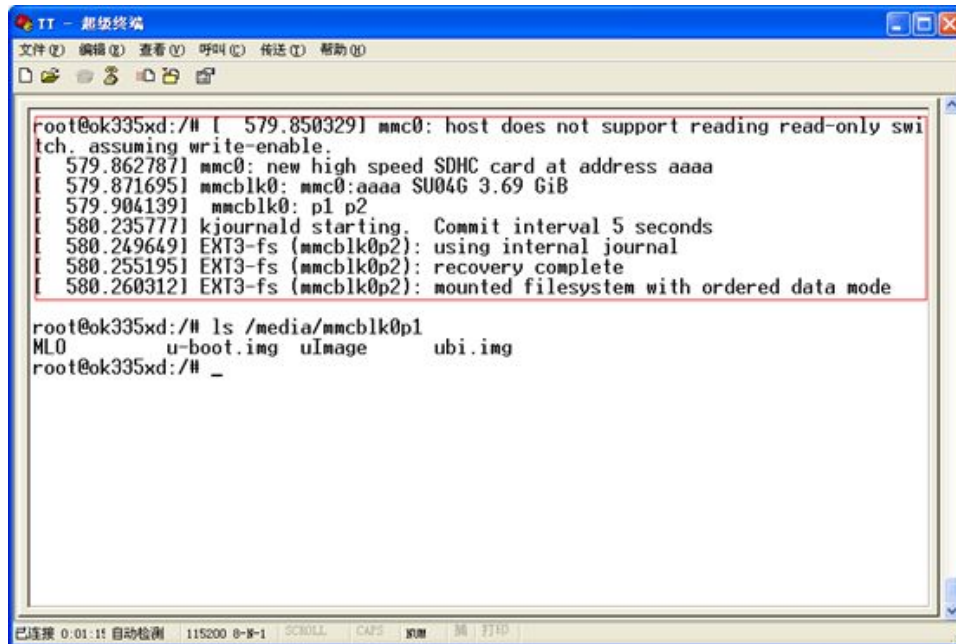


## 3.3 命令行测试

### 3.3.1 SD 卡的测试

1. 拨码开关设置成 NandFlash 启动：1 Off 2 On 。
2. 连接好 5V 电源，开机，启动 Linux 系统。
3. 把刚才制作好的 SD 卡插入到底板的卡槽中。

如果 SD 卡能正常工作，则超级终端会打印如下的信息：

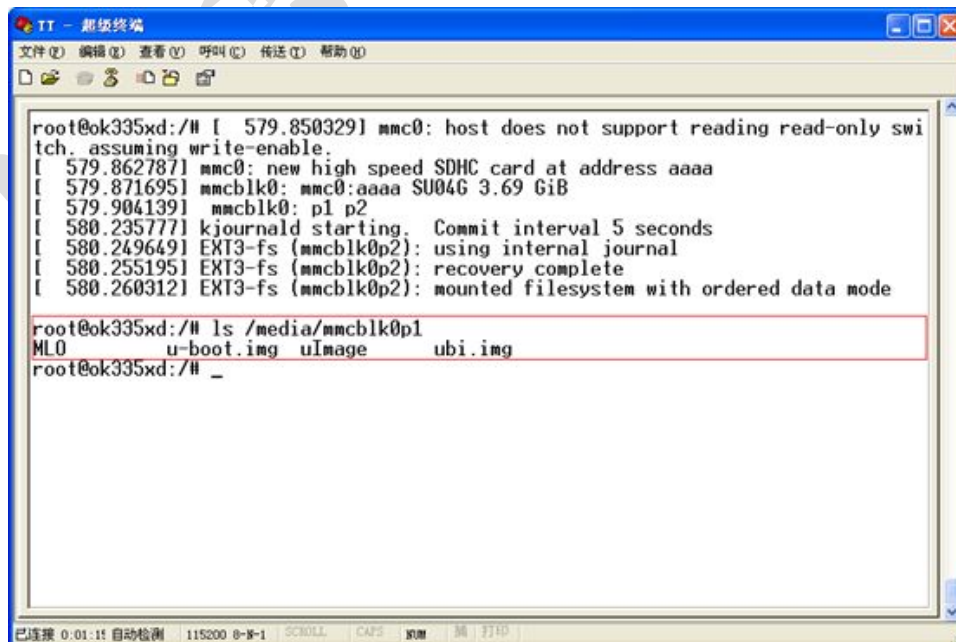


```

root@ok335xd:/# [ 579.850329] mmc0: host does not support reading read-only swi
tch. assuming write-enable.
[ 579.862787] mmc0: new high speed SDHC card at address aaaa
[ 579.871695] mmcblk0: mmc0:aaaa SU04G 3.69 GiB
[ 579.904139] mmcblk0: p1 p2
[ 580.235777] kjournal starting. Commit interval 5 seconds
[ 580.249649] EXT3-fs (mmcblk0p2): using internal journal
[ 580.255195] EXT3-fs (mmcblk0p2): recovery complete
[ 580.260312] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode

root@ok335xd:/# ls /media/mmcblk0p1
ML0      u-boot.img  uImage     ubi.img
root@ok335xd:/# _
    
```

在超级终端中执行 “ls /media/mmcblk0p1” 命令，如果能看到如下的信息，则代表 SD 卡工作正常。



```

root@ok335xd:/# [ 579.850329] mmc0: host does not support reading read-only swi
tch. assuming write-enable.
[ 579.862787] mmc0: new high speed SDHC card at address aaaa
[ 579.871695] mmcblk0: mmc0:aaaa SU04G 3.69 GiB
[ 579.904139] mmcblk0: p1 p2
[ 580.235777] kjournal starting. Commit interval 5 seconds
[ 580.249649] EXT3-fs (mmcblk0p2): using internal journal
[ 580.255195] EXT3-fs (mmcblk0p2): recovery complete
[ 580.260312] EXT3-fs (mmcblk0p2): mounted filesystem with ordered data mode

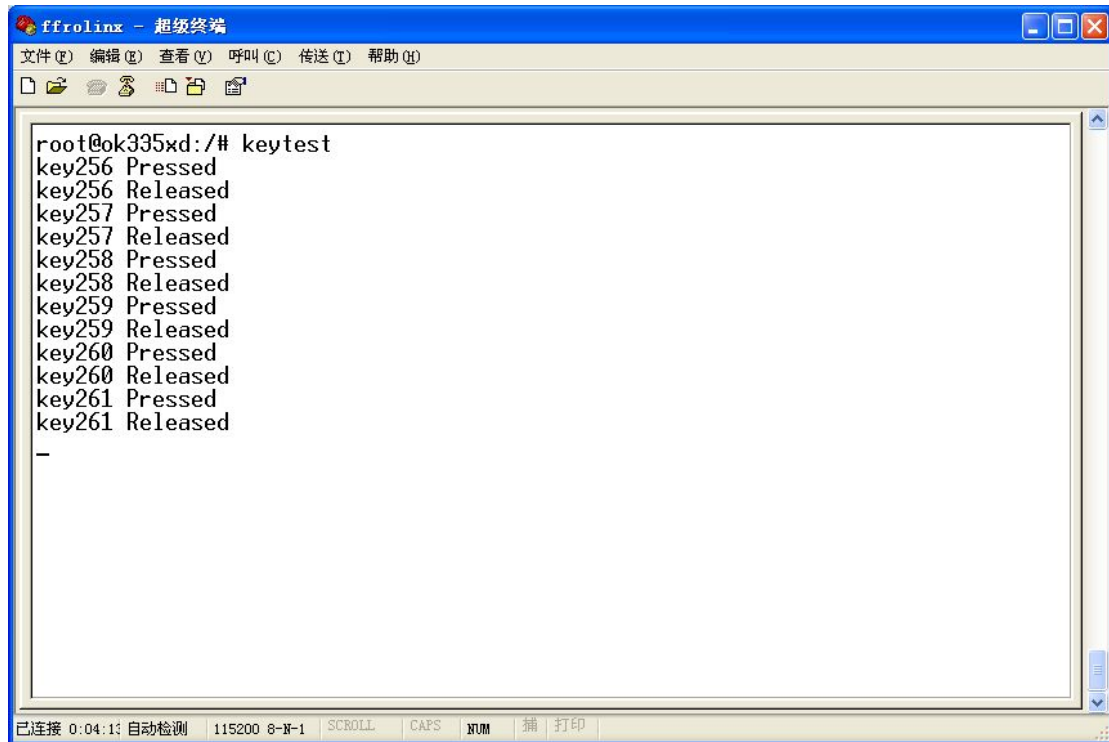
root@ok335xd:/# ls /media/mmcblk0p1
ML0      u-boot.img  uImage     ubi.img
root@ok335xd:/# _
    
```

### 3.3.2 按键测试

OK335xD 有 6 个用户按键, 按键测试需要在超级终端中输入以下 Linux 命令, 命令输入可以在任意路径下, 输入以下命令:

```
root@ok335xd:/# keytest
```

1. 依次将 K3--K8 按键按下抬起, 终端显示如下:



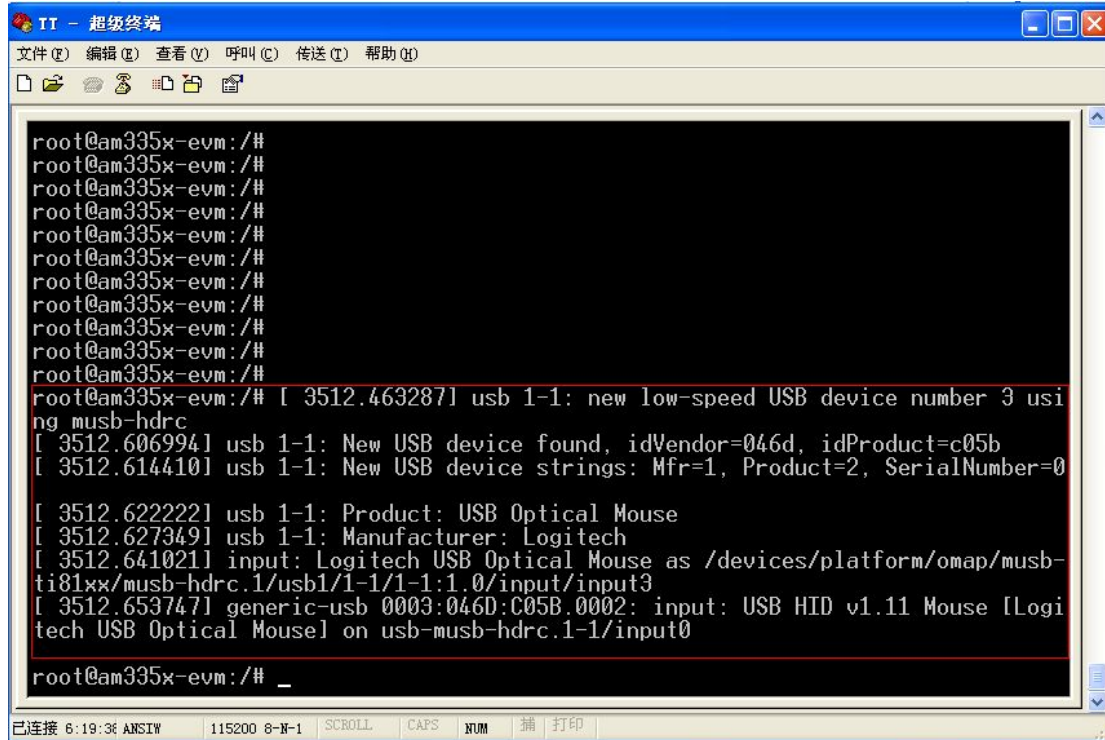
**注:** 每按下一个按键, 会有一个按键值上报给用户, 用户可以根据按键值实现相应的功能。  
**Key256--key261** 分别表示底板中标示符为 K3--K8 的按键, 256--261 分别表示底板上按键 K3--K8 上报给用户的按键值, **Pressed** 表示按键按下, **Released** 表示按键抬起。

2. 按下键盘上面的 Ctrl+C 键将结束按键测试。

### 3.3.3 USB 测试

OK335xD 引出 4 路的 USB HOST 2.0 和一路 OTG2.0 接口,这里我们使用 USB 鼠标或者 U 盘来测试 USB 功能。

把 USB 鼠标插入 USB 口, 超级终端打印如下信息, 则说明 USB 工作正常。

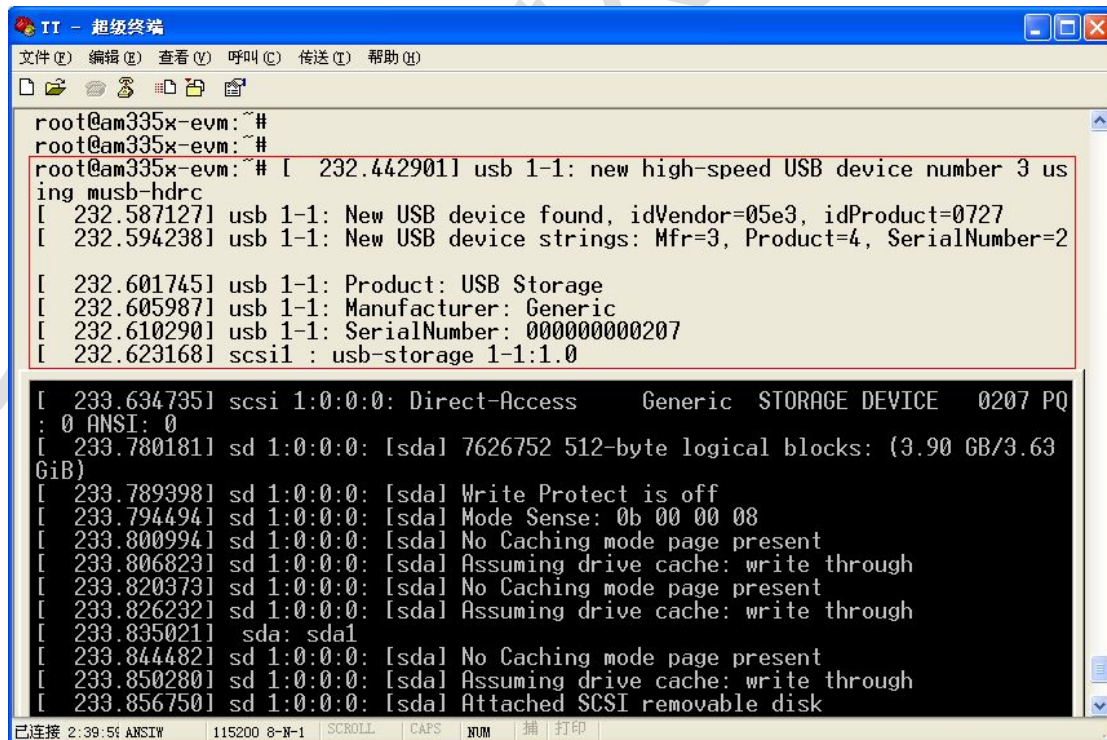


```

root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/# [ 3512.463287] usb 1-1: new low-speed USB device number 3 using musb-hdrc
[ 3512.606994] usb 1-1: New USB device found, idVendor=046d, idProduct=c05b
[ 3512.614410] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 3512.622221] usb 1-1: Product: USB Optical Mouse
[ 3512.627349] usb 1-1: Manufacturer: Logitech
[ 3512.641021] input: Logitech USB Optical Mouse as /devices/platform/omap/musb-ti81xx/musb-hdrc.1/usb1/1-1/1-1:1.0/input/input3
[ 3512.653747] generic-usb 0003:046D:C05B.0002: input: USB HID v1.11 Mouse [Logitech USB Optical Mouse] on usb-musb-hdrc.1-1/input0
root@am335x-evm:/# _

```

U 盘插入到 USB 口, 超级终端会打印如下信息:



```

root@am335x-evm:/#
root@am335x-evm:/#
root@am335x-evm:/# [ 232.442901] usb 1-1: new high-speed USB device number 3 using musb-hdrc
[ 232.587127] usb 1-1: New USB device found, idVendor=05e3, idProduct=0727
[ 232.594238] usb 1-1: New USB device strings: Mfr=3, Product=4, SerialNumber=2
[ 232.601745] usb 1-1: Product: USB Storage
[ 232.605987] usb 1-1: Manufacturer: Generic
[ 232.610290] usb 1-1: SerialNumber: 0000000000207
[ 232.623168] scsi1 : usb-storage 1-1:1.0
[ 233.634735] scsi 1:0:0:0: Direct-Access Generic STORAGE DEVICE 0207 PQ: 0 ANSI: 0
[ 233.780181] sd 1:0:0:0: [sda] 7626752 512-byte logical blocks: (3.90 GB/3.63 GiB)
[ 233.789398] sd 1:0:0:0: [sda] Write Protect is off
[ 233.794494] sd 1:0:0:0: [sda] Mode Sense: 0b 00 00 08
[ 233.800994] sd 1:0:0:0: [sda] No Caching mode page present
[ 233.806823] sd 1:0:0:0: [sda] Assuming drive cache: write through
[ 233.820373] sd 1:0:0:0: [sda] No Caching mode page present
[ 233.826232] sd 1:0:0:0: [sda] Assuming drive cache: write through
[ 233.835021] sda: sda1
[ 233.844482] sd 1:0:0:0: [sda] No Caching mode page present
[ 233.850280] sd 1:0:0:0: [sda] Assuming drive cache: write through
[ 233.856750] sd 1:0:0:0: [sda] Attached SCSI removable disk

```

如果 U 盘能自动挂载到/media/sdxx 目录下, 可以确定 USB 口是正常的。

### 3.3.4 有线网卡测试

OK335xD 具有一路 1000M 网口，测试前将 PC 与开发板直接使用网线连接（注意将 PC 防火墙关闭，否则网络可能不通）。

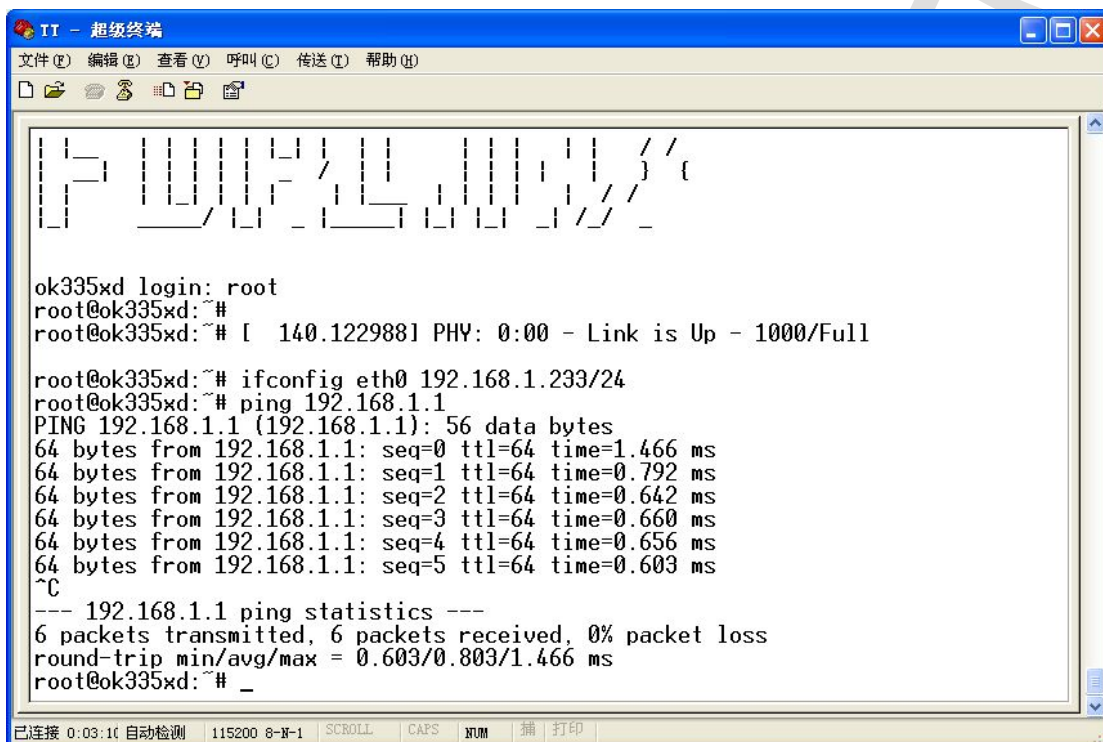
在超级终端输入如下命令：

root@ok335x:/# ifconfig eth0 192.168.1.233 **注：设置开发板的 IP 地址 192.168.1.233**

首先确认您的 PC 的 IP 地址，假如 IP 地址为 192.168.1.1，则输入 ping 命令。

root@ok335x:/# ping 192.168.1.1

如果有线网络工作正常，则会显示如下信息：



```

II - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

ok335xd login: root
root@ok335xd:~#
root@ok335xd:~# [ 140.122988] PHY: 0:00 - Link is Up - 1000/Full

root@ok335xd:~# ifconfig eth0 192.168.1.233/24
root@ok335xd:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=64 time=1.466 ms
64 bytes from 192.168.1.1: seq=1 ttl=64 time=0.792 ms
64 bytes from 192.168.1.1: seq=2 ttl=64 time=0.642 ms
64 bytes from 192.168.1.1: seq=3 ttl=64 time=0.660 ms
64 bytes from 192.168.1.1: seq=4 ttl=64 time=0.656 ms
64 bytes from 192.168.1.1: seq=5 ttl=64 time=0.603 ms
^C
--- 192.168.1.1 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.603/0.803/1.466 ms
root@ok335xd:~# _
    
```

Ctrl+C 则会终止 ping 测试。



### 3.3.5 串口测试

OK335xD 开发板中有三个串口，在开发板上的标识符分别为 COM0，COM1，UART。COM0 为调试口，不需要测试。

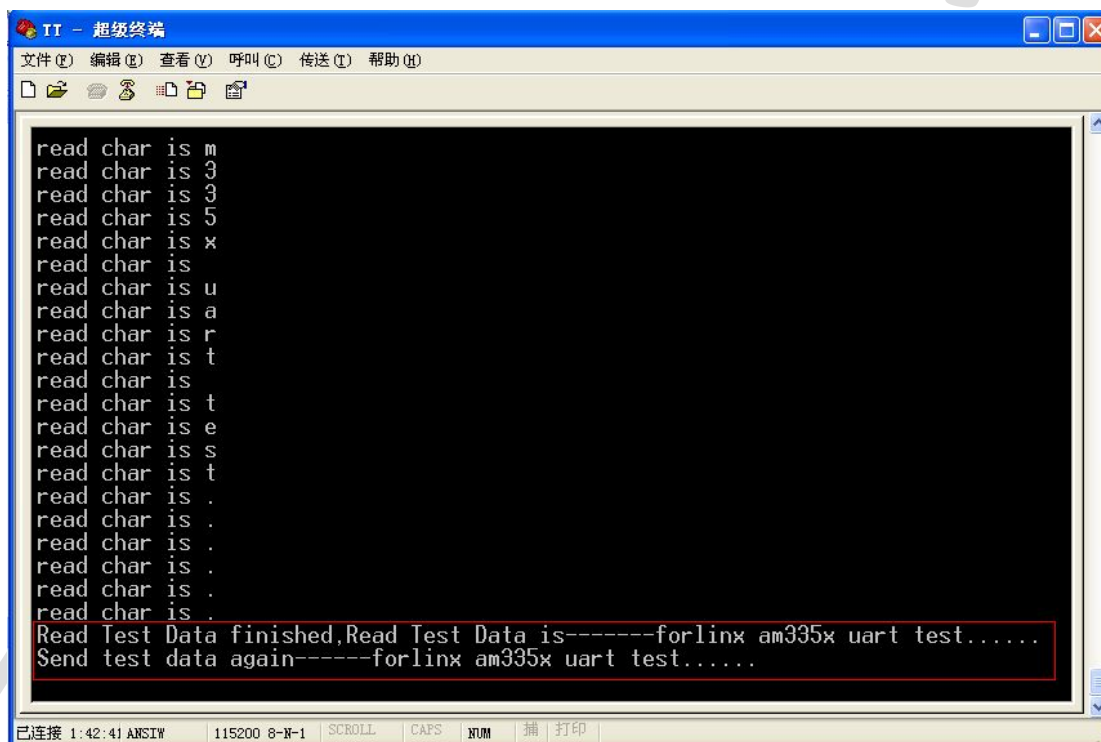
- 1) COM0 口，RS232 电平，用于调试口，在 Linux 中的设备名称： /dev/ttyO0;
- 2) COM1 口，RS232 电平，使用 DB9 公头，在 Linux 中的设备名称： /dev/ttyO1;
- 3) UART 口，TTL 电平，使用插针接口，在 Linux 中的设备名称： /dev/ttyO4;

测试方法如下：

1. 关掉开发板电源，将相应串口的 RX 和 TX 管脚短接，分别执行 COM1 口和 UART 口的测试命令，输入以下命令。

```
root@ok335xd:/# uarttest /dev/ttyO1    测试 COM1 口
root@ok335xd:/# uarttest /dev/ttyO4    测试 UART 口
```

2. 可以看到接收到的数据为：“forlinx am335x uart test .....”，这说明串口的收发功能正常，终端中显示信息如下：



3. Ctrl+C 终止测试程序的运行。



### 3.3.6 音频测试

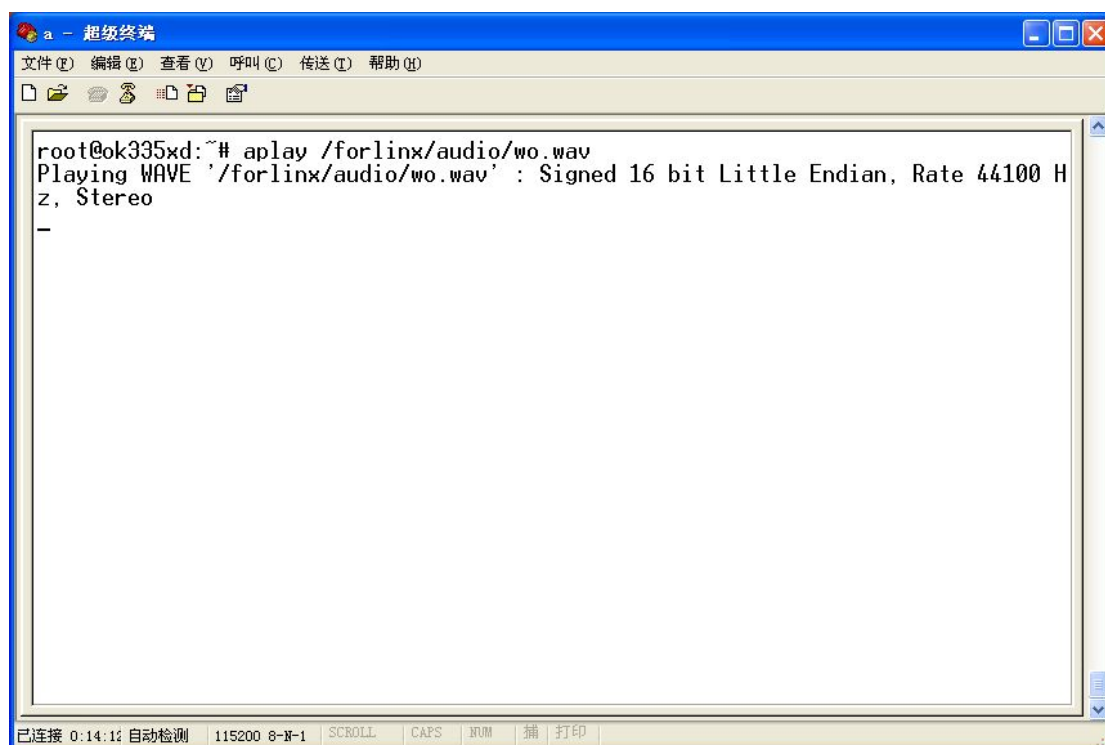
用户可以使用文件系统内带的 **alsa** 音频录制、播放工具进行测试。

#### 1. 放音测试：

将耳机插头插入开发板 PHONE 插孔，在超级终端中执行下面命令：

root@ok335xd:~# aplay /forlinx/audio/wo.wav (wo.wav 为/forlinx/audio 目录下用于测试的歌曲文件)

如下图：



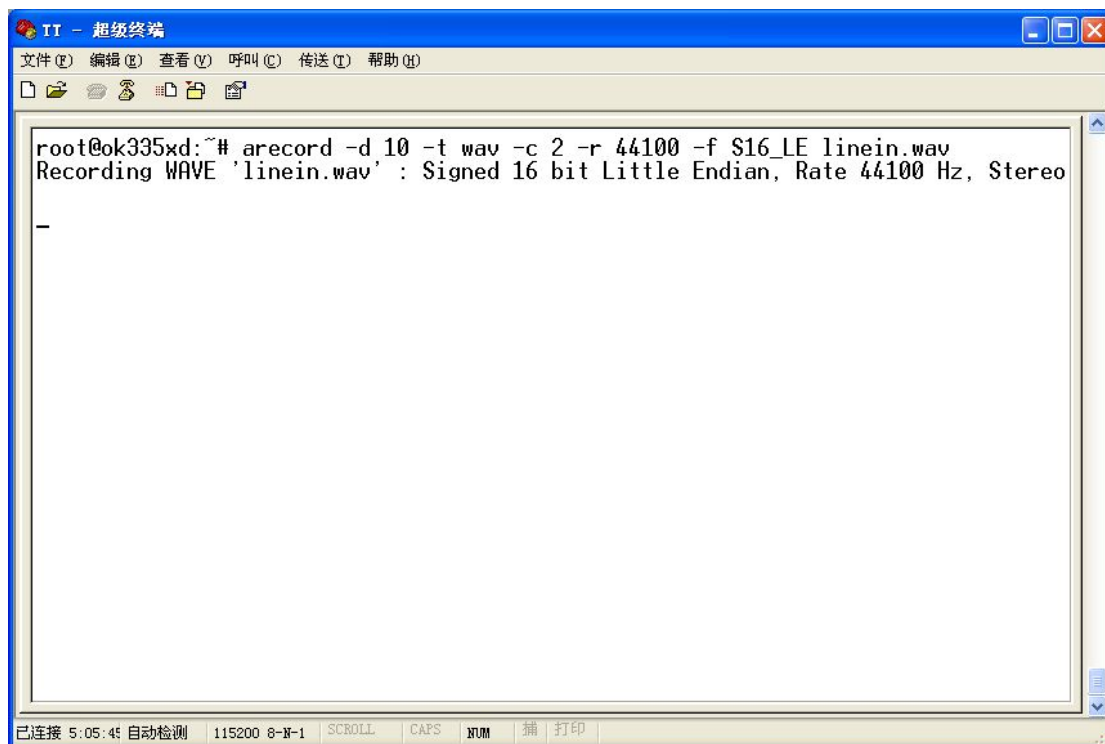
## 2. Line in 录音测试：

将 Line in 一端插入开发板 LINE IN 插孔，将 Line in 线另一端插入音源设备，使用 arecord 工具进行录音并保存，然后使用 aplay 播放保存的录音文件，如果可以播放录音文件，说明 Line in 接口正常。

```
root@ok335xd# arecord -d 10 -t wav -c 2 -r 44100 -f S16_LE linein.wav
```

（-d 录音时间（秒），-t 输出格式 -c 通道数 -r 采样率 -f 保存格式）

如下图：

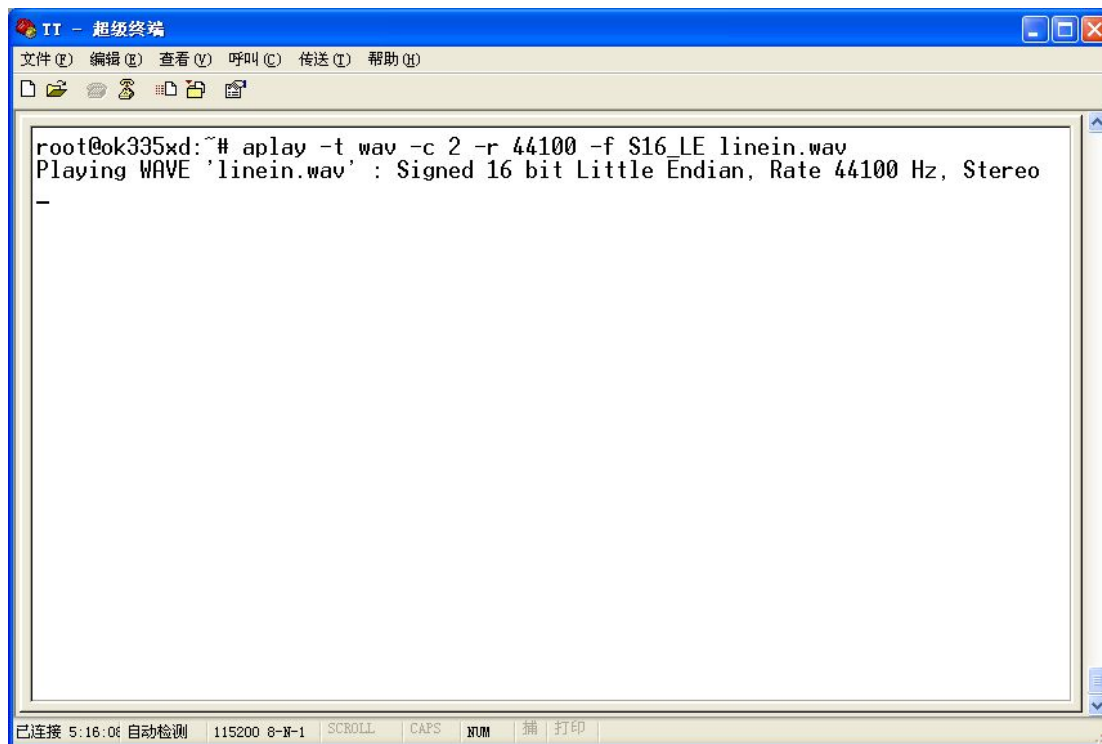


**注：进行 Line In 录音时请将音源设备的输出音量（增益）调小，以防输入波形超出采集范围而失真**

### 3. Line in 播放：

```
root@ok335xd# aplay -t wav -c 2 -r 44100 -f S16_LE linein.wav
```

如下图：



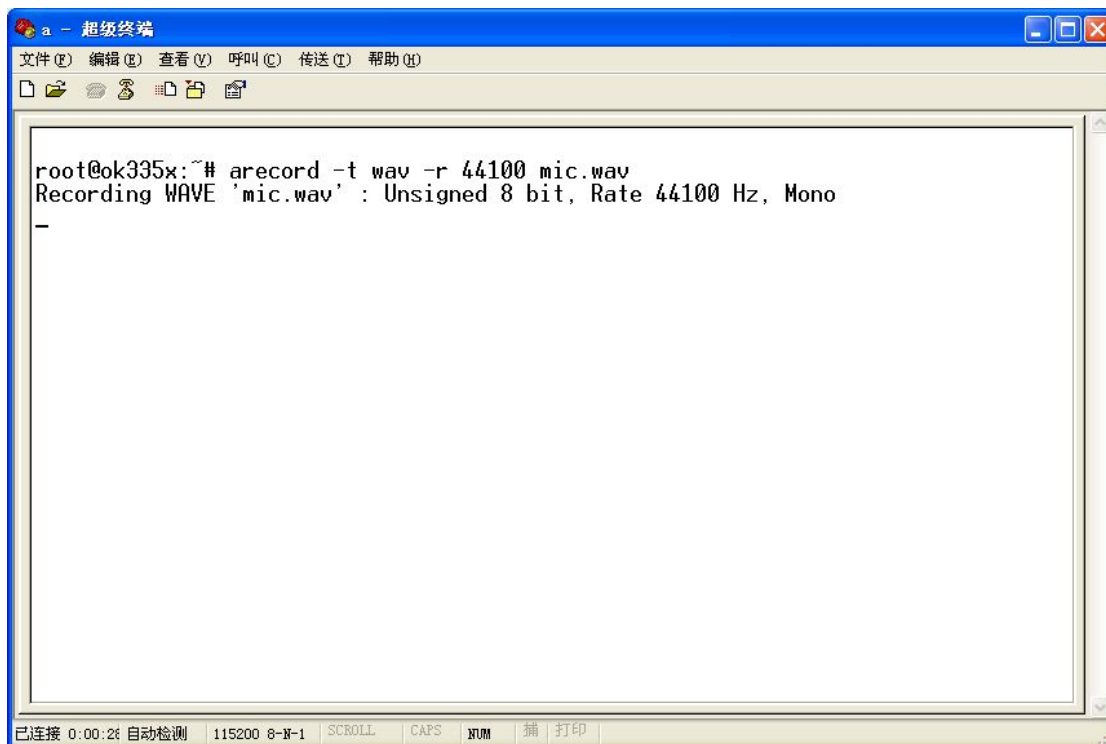
#### 4. Mic 录音测试：

将 Mic 插头插入开发板 MIC 插孔，对着 mic 进行说话，使用 arecord 工具进行录音并保存， 然后使用 aplay 播放保存的录音文件， 如果可以播放录音文件，说明 Mic 接口正常。

Mic 录制：

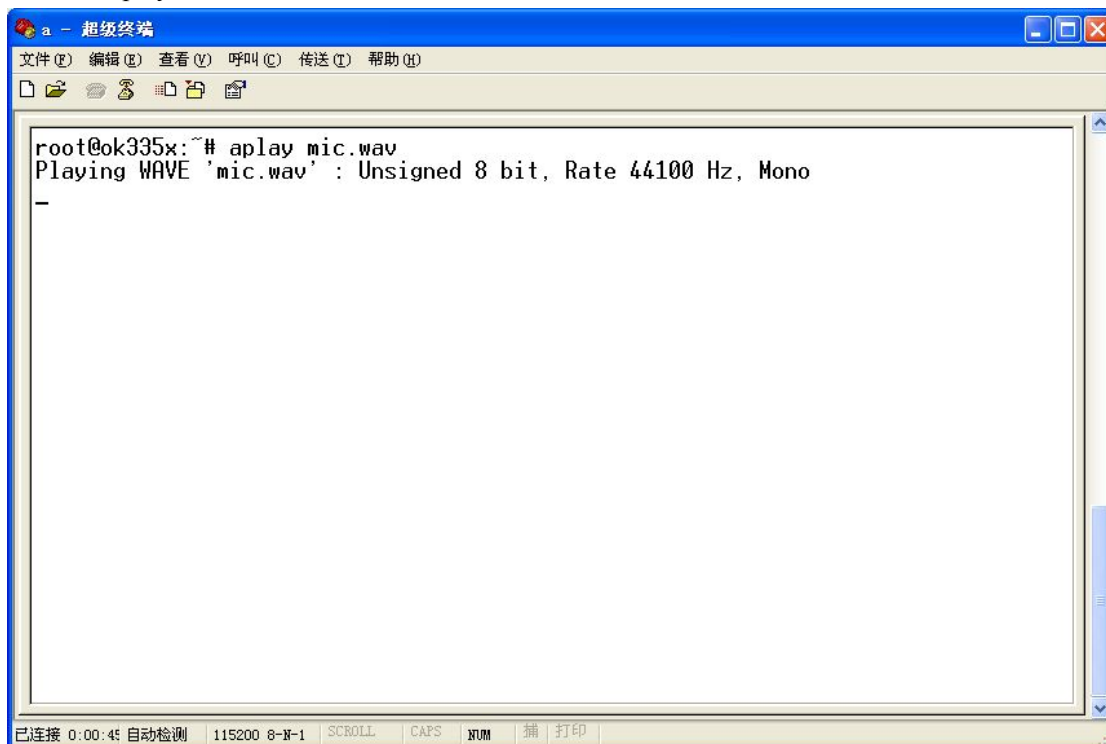
```
root@ok335xd# arecord -t wav -r 44100 mic.wav
```

如下图：



## 5. Mic 播放：

root@ok335xd# aplay mic.wav 如下图：

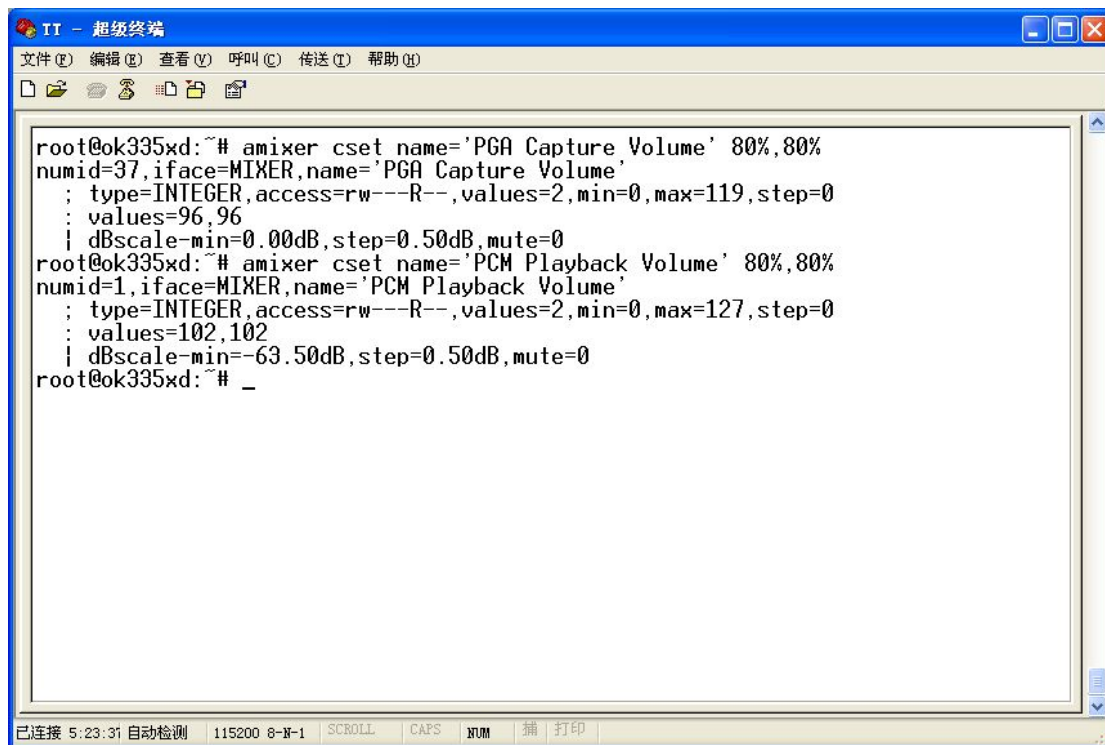




## 6. Amixer 调节音量：

root@ok335xd:~# amixer cset name='PGA Capture Volume' x%,x% (x between 0 - 100) — 设置放音音量

root@ok335xd:~# amixer cset name='PCM Playback Volume' x%,x% (x between 0 - 100) — 设置录音音量  
如下图：



```

root@ok335xd:~# amixer cset name='PGA Capture Volume' 80%,80%
numid=37,iface=MIXER,name='PGA Capture Volume'
; type=INTEGER,access=rw---R--,values=2,min=0,max=119,step=0
; values=96,96
; dBscale-min=0.00dB,step=0.50dB,mute=0
root@ok335xd:~# amixer cset name='PCM Playback Volume' 80%,80%
numid=1,iface=MIXER,name='PCM Playback Volume'
; type=INTEGER,access=rw---R--,values=2,min=0,max=127,step=0
; values=102,102
; dBscale-min=-63.50dB,step=0.50dB,mute=0
root@ok335xd:~# _
    
```

如果在测试 Iline-in 的时候，出现杂音现象，可以通过调节 capture 来减小噪音。

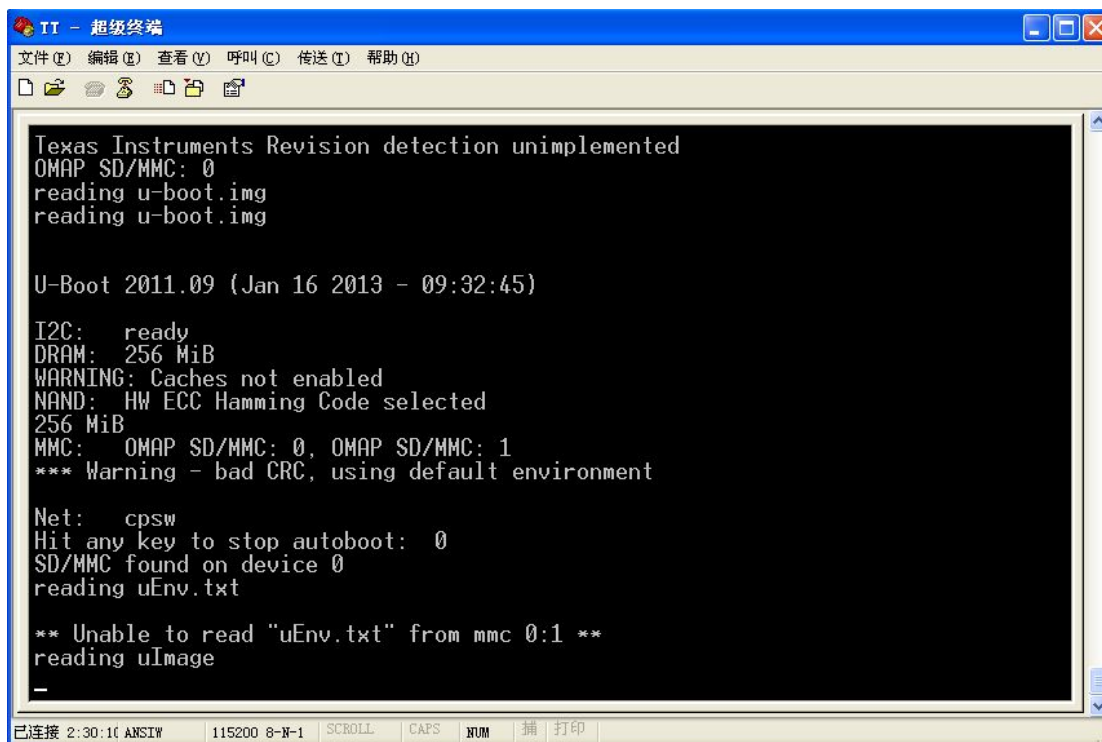
amixer cset name='PGA Capture Volume' 10%, 10%

### 3.3.7 复位测试

复位测试，主要测试系统的硬启动，按键位于 SD 卡右侧，丝印标号：K2。

按下按键后系统重新启动，则说明 RST 键工作正常。

超级终端打印如下信息，代表系统重新启动：



```

Texas Instruments Revision detection unimplemented
OMAP SD/MMC: 0
reading u-boot.img
reading u-boot.img

U-Boot 2011.09 (Jan 16 2013 - 09:32:45)

I2C:  ready
DRAM:  256 MiB
WARNING: Caches not enabled
NAND:  HW ECC Hamming Code selected
       256 MiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
*** Warning - bad CRC, using default environment

Net:   cpsw
Hit any key to stop autoboot:  0
SD/MMC found on device 0
reading uEnv.txt

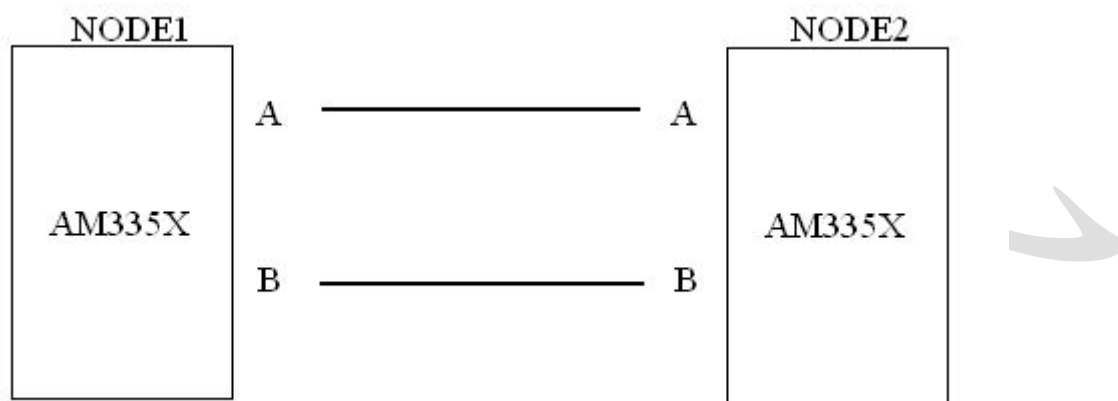
** Unable to read "uEnv.txt" from mmc 0:1 **
reading uImage
-
    
```

### 3.3.8 RS485 测试

OK335xD 可以作为一个 RS485 的挂载设备。文档中提供两种测试方式，用户可以选择方便的方式进行测试，具体步骤如下。

方式一

按照下图所示连接两块 OK335xD 开发板，并参考原理图找到对应的引脚，用连接线连接 OK335xD 的 RS485 接口和另一块开发板的 RS485 接口。



1. 485 通信只支持半双工通信，即通信一端同一时间只能发送或者只能接收信息，一端应先处于接收状态，发送端才能发送，否则会出现数据丢失的情况。在终端中执行如下命令：

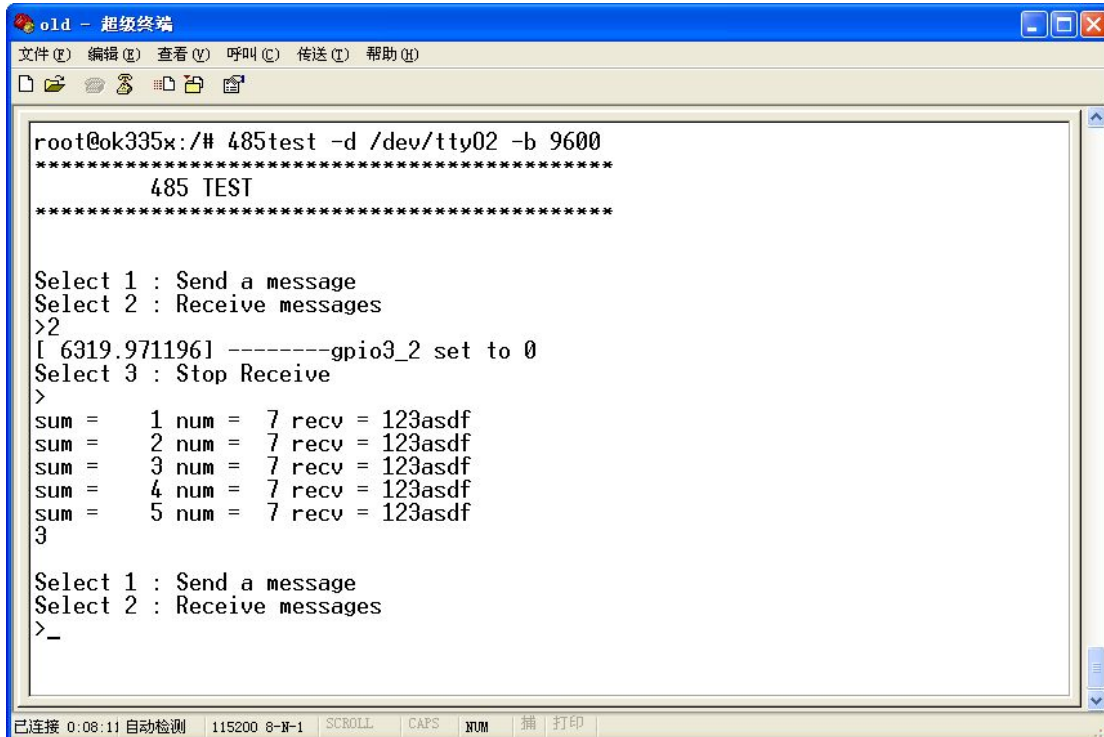
```
root@ok335xd:/# 485test -d /dev/ttyO2 -b 9600
```

2. 终端中显示测试信息如下所示：

```

new - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# 485test -d /dev/tty02 -b 9600
*****
485 TEST
*****
Select 1 : Send a message
Select 2 : Receive messages
>_
已连接 2:35:16 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
    
```

一端选择“2”接收信息，终端中打印出接收到的信息，“sum”表示接收数据个数，“num”表示单次接收字符的个数，“recv”接收到的字符串，在选择“3”停止接收前，接收端一直处于接收状态：



```

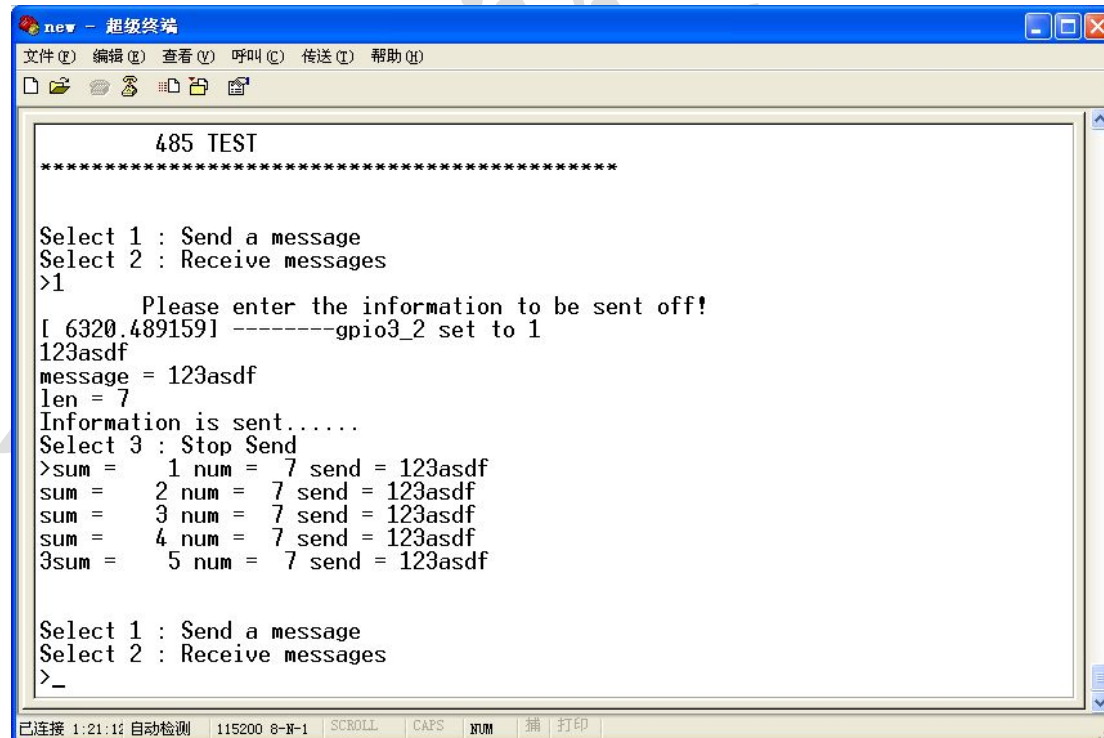
old - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

root@ok335x:/# 485test -d /dev/tty02 -b 9600
*****
485 TEST
*****

Select 1 : Send a message
Select 2 : Receive messages
>2
[ 6319.971196] -----gpio3_2 set to 0
Select 3 : Stop Receive
>
sum = 1 num = 7 recv = 123asdf
sum = 2 num = 7 recv = 123asdf
sum = 3 num = 7 recv = 123asdf
sum = 4 num = 7 recv = 123asdf
sum = 5 num = 7 recv = 123asdf
3
Select 1 : Send a message
Select 2 : Receive messages
>_
已连接 0:08:11 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

一端选择“1”发送信息，输入要发送的信息，如“123asdf”，“sum”表示发送数据个数，“num”表示单次发送字符的个数，“recv”发送的字符串，在选择“3”停止发送之前发送端处于循环发送状态，接收端可以一直接收到数据：



```

new - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

485 TEST
*****

Select 1 : Send a message
Select 2 : Receive messages
>1
Please enter the information to be sent off!
123asdf
[ 6320.489159] -----gpio3_2 set to 1
message = 123asdf
len = 7
Information is sent.....
Select 3 : Stop Send
>sum = 1 num = 7 send = 123asdf
sum = 2 num = 7 send = 123asdf
sum = 3 num = 7 send = 123asdf
sum = 4 num = 7 send = 123asdf
sum = 5 num = 7 send = 123asdf
3sum = 5 num = 7 send = 123asdf

Select 1 : Send a message
Select 2 : Receive messages
>_
已连接 1:21:14 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

### 3. Ctrl+C 终止测试程序的运行

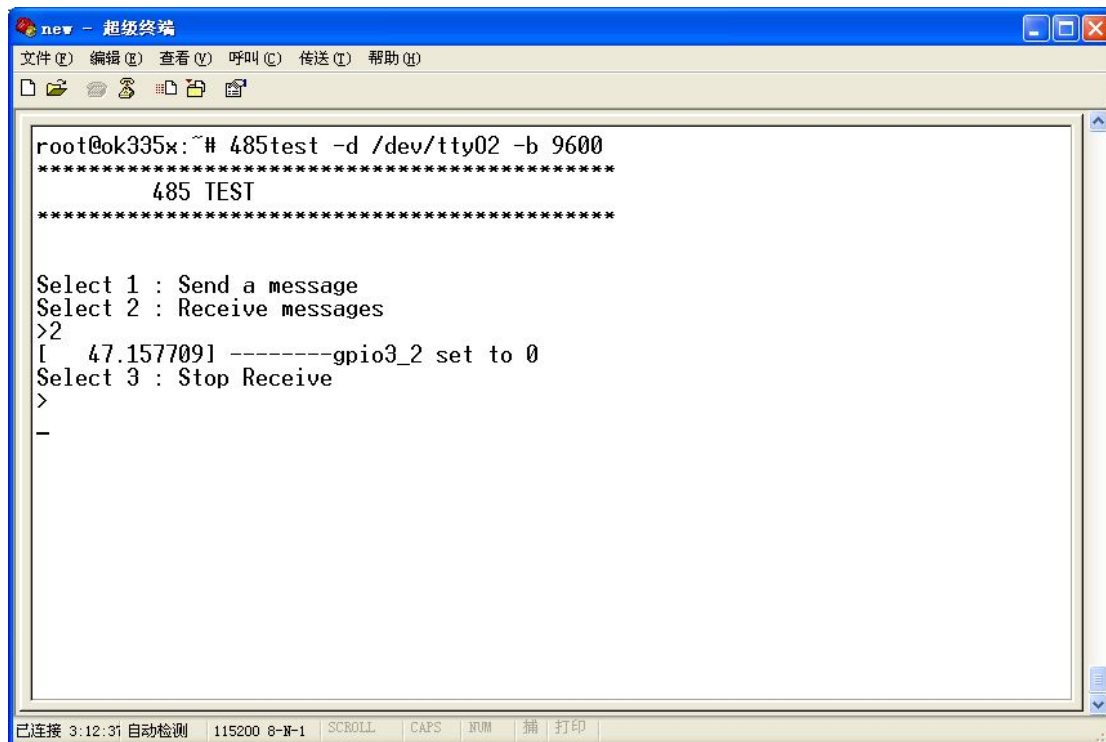
## 方式二

将一个 RS232 转 485 模块和我们的开发板连接，如果是使用我们提供的转换模块，连接方式是 A+（转换模块）接 A（开发板）、B+（转换模块）接 B（开发板）。

1. 开发板端使用超级终端，执行测试命令，选择接收：

```
485test -d /dev/ttyO2 -b 9600
```

终端中显示如下：



2. 打开串口调试助手（RS232 转 485 模块端使用），我们使用的是 scom32.exe，设置对应的串口号、波特率、数据位、停止位、校验位和流控制，输入要发送的字符，如“12345qwer”，点击发送，设置如下：





3. 超级终端中会显示出刚才发送的字符, 选择“3”, 结束接收, 按照提示选择发送“1”, 如“1234567asdf”:

```

new - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

Select 1 : Send a message
Select 2 : Receive messages
>2
[ 287.080525] -----gpio3_2 set to 0
Select 3 : Stop Receive
>
sum = 1 num = 9 recv = 1234567asdf
3

Select 1 : Send a message
Select 2 : Receive messages
>1
Please enter the information to be sent off!
[ 308.349785] -----gpio3_2 set to 1
1234567asdf
message = 1234567asdf
len = 11
Information is sent.....
Select 3 : Stop Send
>sum = 1 num = 11 send = 1234567asdf
sum = 2 num = 11 send = 1234567asdf
sum = 3 num = 11 send = 1234567asdf
-

已连接 0:05:51 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
    
```

4. 串口调试助手中会显示发送的字符, 终端显示如下:

```

SSCOM3.2 (作者:聂小猛(丁丁), 主页http://www.mcu51.com,...)

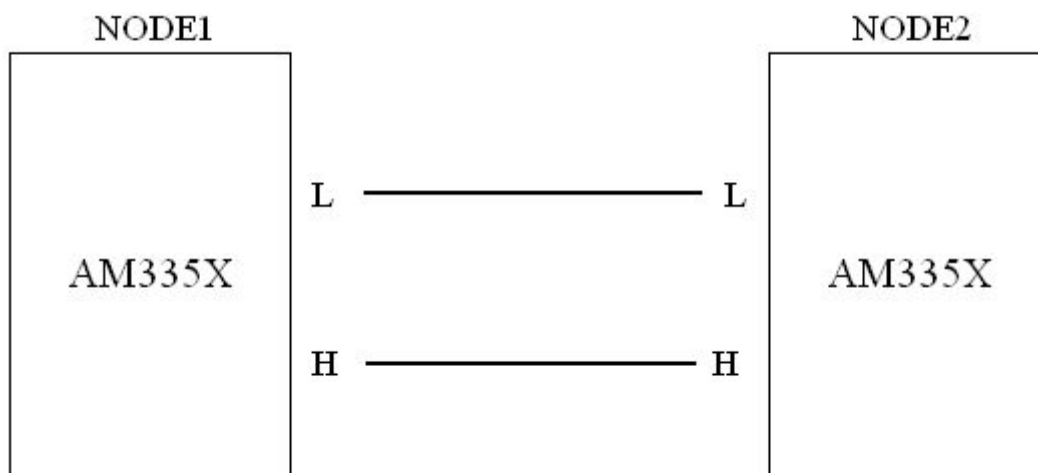
1234567asdf1234567asdf1234567asdf

打开文件 文件名 发送文件 保存窗口 清除窗口  HEX显示
串口号 COM1 关闭串口 帮助 WWW.MCU51.COM 扩展
波特率 9600 DTR RTS
数据位 8 定时发送 1000 ms/次
停止位 1 HEX发送 发送新行
校验位 None 字符串输入框: 发送
流控制 None 1234567asdf

www.mcu51.com S:0 R:33 COM1已打开 9600bps 8 1 CTS=0 DSR=0 RLSD=0
    
```

### 3.3.9 CAN 测试

OK335xD 可以作为一个 CAN 设备使用。按照下图所示连接两块 OK335xD 开发板，并参考原理图找到对应的引脚，用连接线连接 OK335xD 的 CAN 接线端子的 L、H 和另一块开发板的 CAN 接口，用户也可以使用自己的 Can 设备进行测试。



测试方法如下：

1. 将OK335xD和另一个CAN设备的通信波特率都设置为125KBPS，并使能CAN设备。  

```
root@ok335xd:/# canconfig can0 bitrate 125000 ctrlmode triple-sampling on
```

```
root@ok335xd:/# canconfig can0 start
```
2. 在OK335xD和另一个CAN设备两端分别执行接收和发送数据的命令, 接收端应先处于接收状态，然后发送端再发送。  
 执行以下命令接收数据包：  

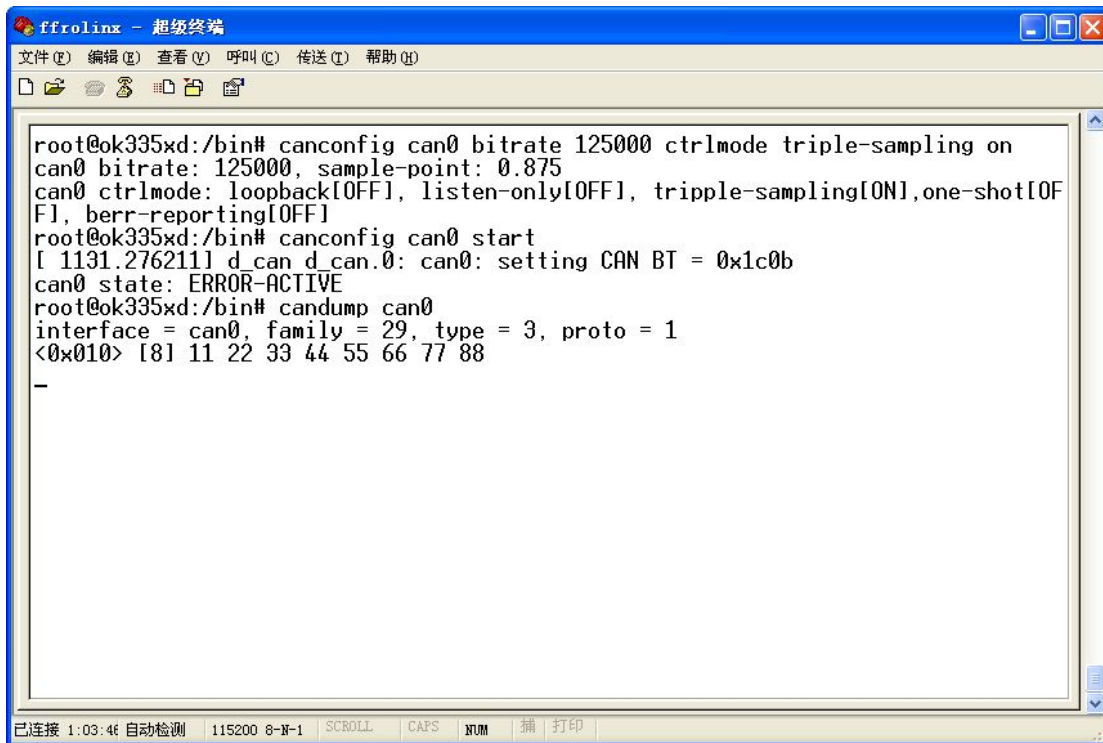
```
root@ok335xd:/# candump can0
```

 执行以下命令发送数据包：  

```
root@ok335xd:/# cansend can0 -i 0x10 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
```

### 3. 终端中显示测试结果如下:

接收端测试结果如下:

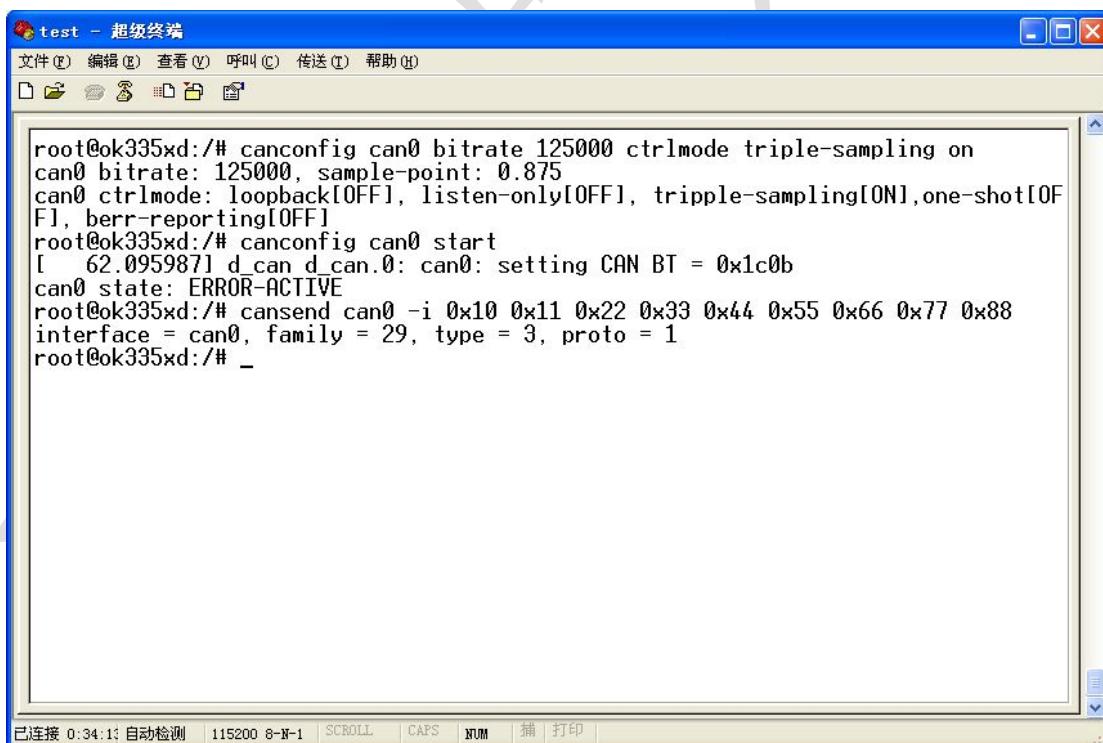


```

root@ok335xd:/bin# canconfig can0 bitrate 125000 ctrlmode triple-sampling on
can0 bitrate: 125000, sample-point: 0.875
can0 ctrlmode: loopback[OFF], listen-only[OFF], tripple-sampling[ON], one-shot[OFF],
berr-reporting[OFF]
root@ok335xd:/bin# canconfig can0 start
[ 1131.276211] d_can d_can.0: can0: setting CAN BT = 0x1c0b
can0 state: ERROR-ACTIVE
root@ok335xd:/bin# candump can0
interface = can0, family = 29, type = 3, proto = 1
<0x010> [8] 11 22 33 44 55 66 77 88
-

```

发送端测试结果如下:



```

root@ok335xd:/# canconfig can0 bitrate 125000 ctrlmode triple-sampling on
can0 bitrate: 125000, sample-point: 0.875
can0 ctrlmode: loopback[OFF], listen-only[OFF], tripple-sampling[ON], one-shot[OFF],
berr-reporting[OFF]
root@ok335xd:/# canconfig can0 start
[ 62.095987] d_can d_can.0: can0: setting CAN BT = 0x1c0b
can0 state: ERROR-ACTIVE
root@ok335xd:/# cansend can0 -i 0x10 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
interface = can0, family = 29, type = 3, proto = 1
root@ok335xd:/# _

```

#### 4. 关闭设备

```
root@ok335xd:/# canconfig can0 stop
```

注：可以根据以上命令进行相互收发测试，还可以设置不同的波特率进行通信，在设置不同波特率之前必须执行第 4 条命令“关闭设备”，然后再重新设置。两个 CAN 设备必须工作在同一波特率值，发送之前应确保另一端处于接收状态，这样接收端才会打印发送端发送的信息。以上测试是通过 CAN 工具进行的命令行测试，通过网络套接字测试的源码（cantest.c）和可执行文件（cantest）在用户光盘中提供，用户可以在两块开发板上分别执行 cantest 命令，自行测试。

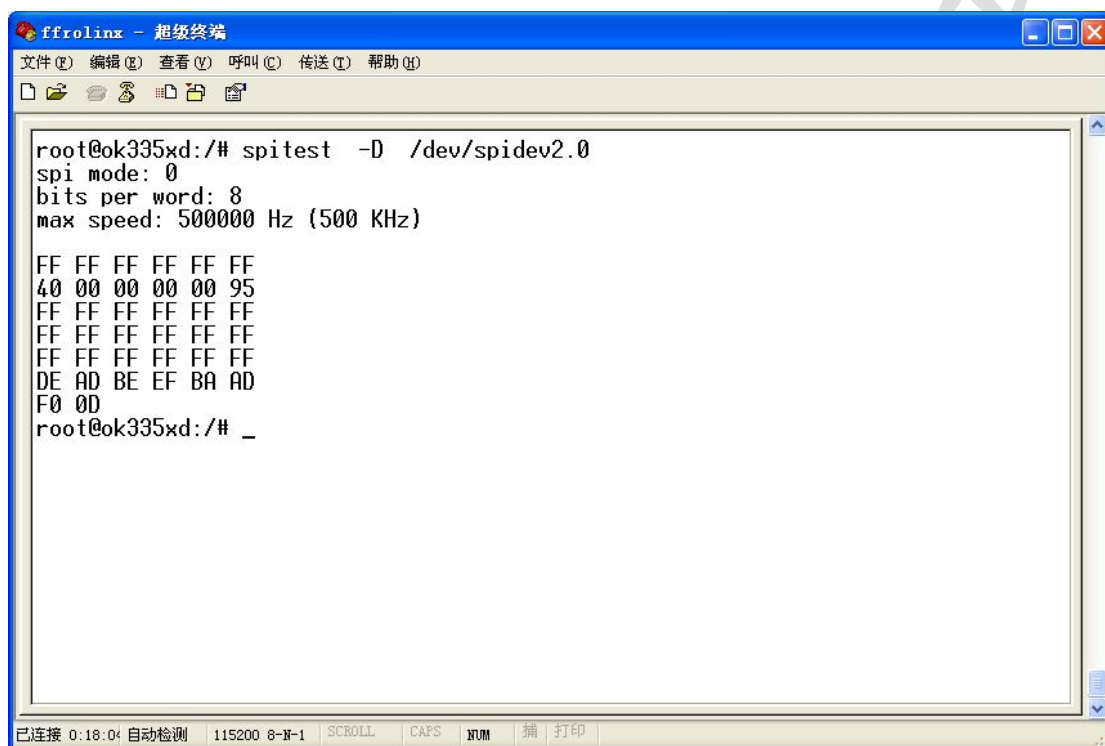
### 3.3.10 SPI 测试

OK335xD 开发板提供了一路 TTL 电平的插针 SPI 接口，将原理图中的 SPI1-D0 和 SPI1\_D1 引脚短接，查看原理图找到开发板上对应的引脚。

#### 1. 在终端中执行如下命令：

```
root@ok335xd:/# spitest -D /dev/spidev2.0
```

#### 2. 在终端中出现如下测试信息则测试成功：



```
ffrolinx - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335xd:/# spitest -D /dev/spidev2.0
spi mode: 0
bits per word: 8
max speed: 500000 Hz (500 KHz)

FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D
root@ok335xd:/# _
```

注：在测试程序中发送数据如下所示，当 SPI 的数据引脚短接后，终端中打印出发送的数据包。

发送的数据如下：

```
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x40, 0x00, 0x00, 0x00, 0x00, 0x95,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xDE, 0xAD, 0xBE, 0xEF, 0xBA, 0xAD,
0xF0, 0x0D
```

### 3.3.11 看门狗测试

“看门狗”，全称 Watchdog timer，是一个在软件出错的时候可以复位计算机系统的硬件定时器。通常一个用户空间守护进程会在正常的时间间隔内通过/dev/watchdog 特殊设备文件来通知内核的 Watchdog 驱动，用户空间一切正常。如果用户空间出现问题（如 RAM 错误，内核 BUG 等），则通知将会停止，然后硬件 Watchdog 将在超时后复位系统。

1. 在终端中执行以下命令：

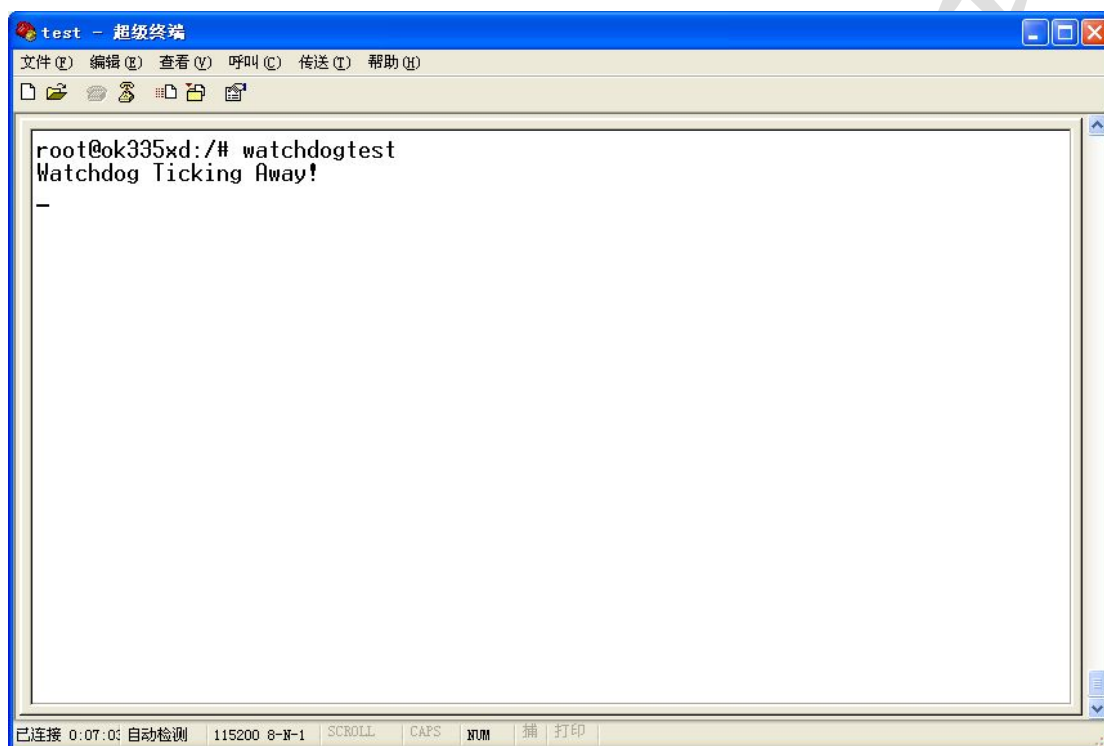
```
root@ok335xd:/# restarttest
```

现象：开发板 60 秒后重启。测试程序中打开了/dev/watchdog 设备文件，启动 Watchdog，程序进入忙循环状态，由于没有喂狗，60 秒后系统复位。

2. 在终端中执行以下命令：

```
root@ok335xd:/# watchdogtest
```

终端中现象如下，测试程序中打开/dev/watchdog 设备文件，启动了 Watchdog，每过一秒钟喂狗一次，系统不会重启：



```
test - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335xd:/# watchdogtest
Watchdog Ticking Away!
-
已连接 0:07:01 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
```



### 3.3.12 LED 测试

OK335xD 底板上的 LED4、LED5、LED6、LED7 为用户 LED 灯，分别对应/sys/class/leds 目录下的 usr0、usr1、usr2、usr3 目录。

LED4--LED7 测试命令如下，此测试在终端中进行：

1. 控制 LED4 亮: `echo 1 > /sys/class/leds/usr0/brightness`
2. 控制 LED4 灭: `echo 0 > /sys/class/leds/usr0/brightness`
3. 控制 LED5 亮: `echo 1 > /sys/class/leds/usr1/brightness`
4. 控制 LED5 灭: `echo 0 > /sys/class/leds/usr1/brightness`
5. 控制 LED6 亮: `echo 1 > /sys/class/leds/usr2/brightness`
6. 控制 LED6 灭: `echo 0 > /sys/class/leds/usr2/brightness`
7. 控制 LED7 亮: `echo 1 > /sys/class/leds/usr3/brightness`
8. 控制 LED7 灭: `echo 0 > /sys/class/led /usr3/brightness`

用户也可以通过操作/dev/led 来控制 LED，示例程序参见图形测试中的 LED 测试。

### 3.3.13 背光测试

背光的亮度设置范围为（0—100），100 表示亮度最高。0 表示关闭背光亮亮度。

进入系统后在终端下输入如下命令进行背光测试。

1. 查看背光的亮度默认值, 系统默认值为“60”

```
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
```

2. 设置背光亮亮度为0并查看背光值是否设置成功

```
root@ok335xd:/# echo 0 > /sys/class/backlight/pwm-backlight/brightness
```

```
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
```

此时背光关闭，屏幕为黑的

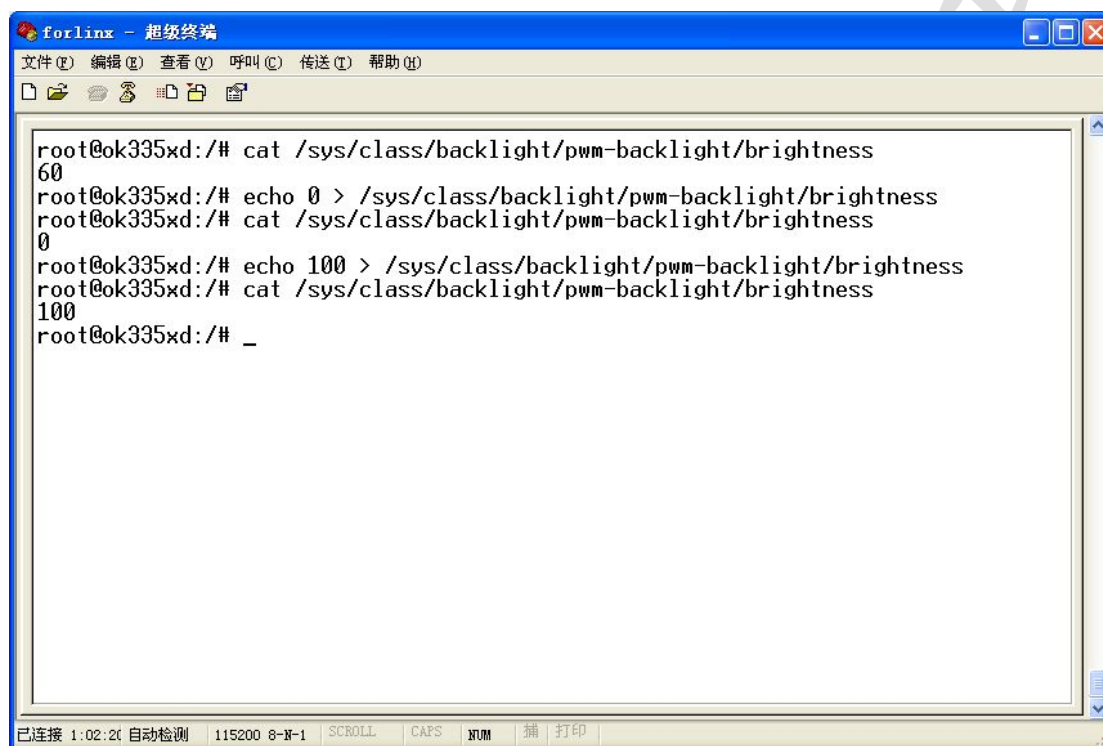
3. 设置背光亮亮度为100并查看背光值是否设置成功

```
root@ok335xd:/# echo 100 > /sys/class/backlight/pwm-backlight/brightness
```

```
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
```

此时屏幕处于最亮的状态。

测试结果在终端中显示如下：



```
forlinx - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
60
root@ok335xd:/# echo 0 > /sys/class/backlight/pwm-backlight/brightness
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
0
root@ok335xd:/# echo 100 > /sys/class/backlight/pwm-backlight/brightness
root@ok335xd:/# cat /sys/class/backlight/pwm-backlight/brightness
100
root@ok335xd:/# _
```

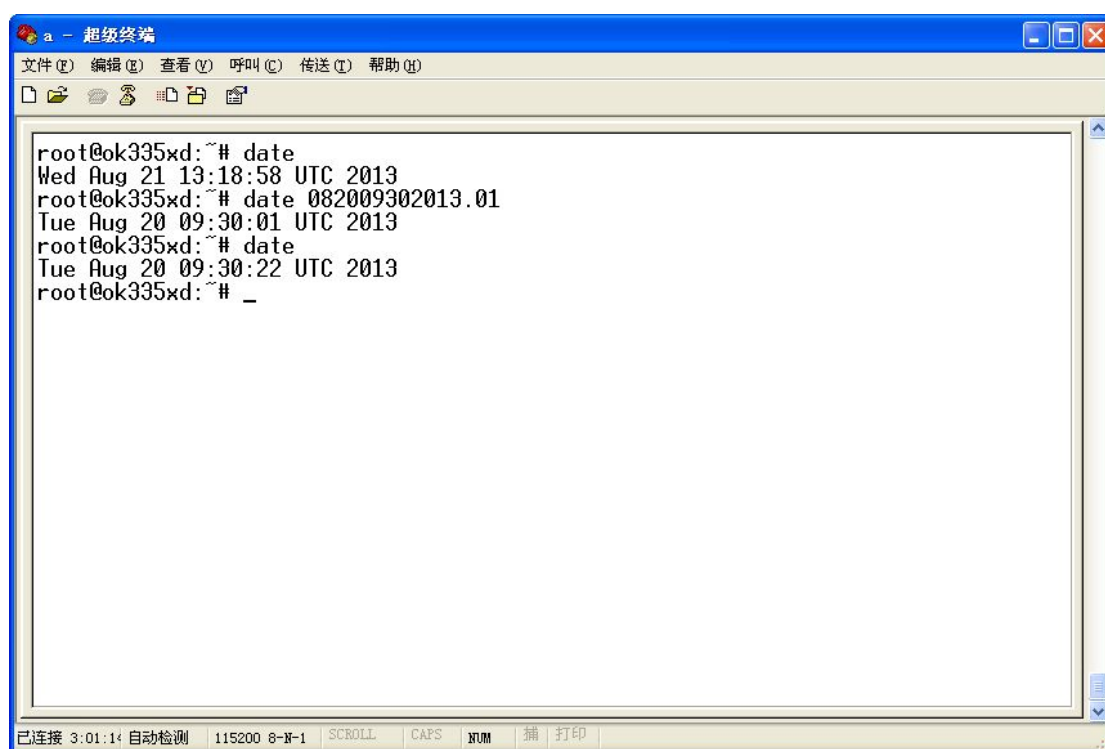
### 3.3.14 RTC 测试

RTC 测试，主要通过使用 `date` 和 `hwclock` 工具设置软、硬件时间，测试当操作系统重启的时候，软件时钟读取 RTC 时钟是否同步（**注意：确保板子上已经安装了纽扣电池**）。

1. `date` — 查看下系统时间
2. `date 082009302013.01` — `date` 设置系统时间，假如在系统启动的时候没有设置的话，系统就会从 1970 年开始。

`date` 设置的格式：MMDDhhmm[[YY]YY][.ss], MM 是月份，DD 是日期，hh 是小时，mm 是分钟，年份可以写2位的 YY 或4位 YYYY，秒可以用.ss 表示。

如下图：

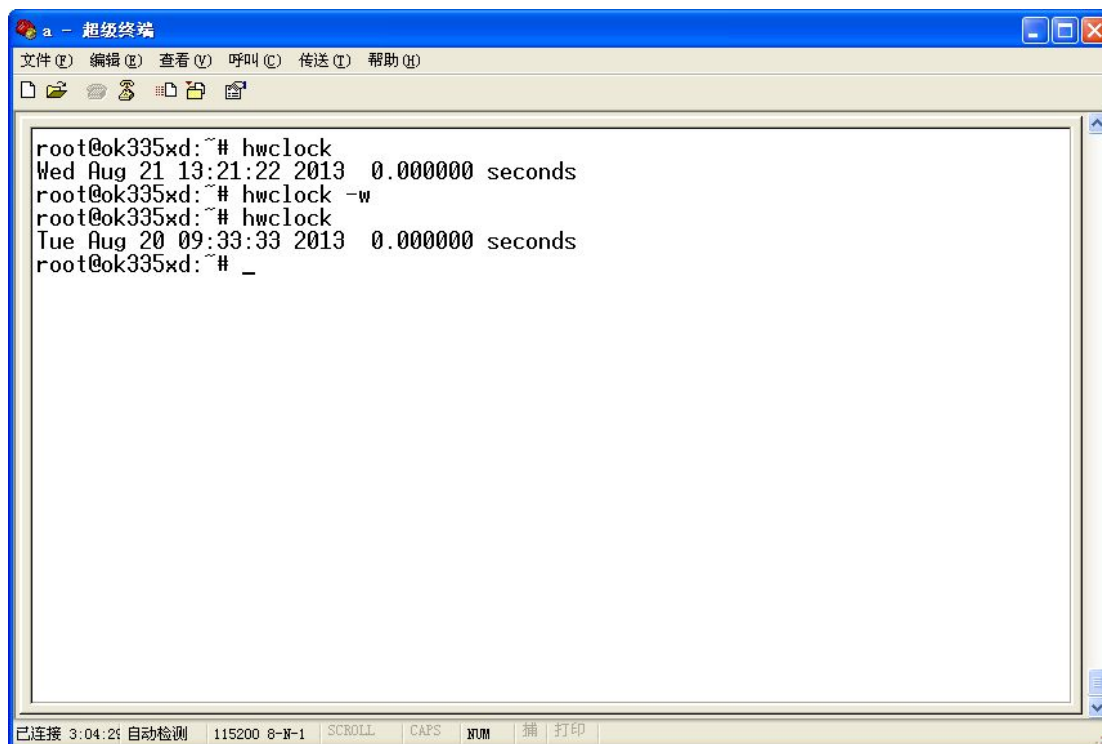


```

root@ok335xd:~# date
Wed Aug 21 13:18:58 UTC 2013
root@ok335xd:~# date 082009302013.01
Tue Aug 20 09:30:01 UTC 2013
root@ok335xd:~# date
Tue Aug 20 09:30:22 UTC 2013
root@ok335xd:~# _
    
```

3. `hwclock` — 查看下硬件时间
4. `hwclock -w` — 将系统时间设置到时钟芯片里面，此时 `rtc` 就可以使用了。如果没有这步，下次启动时，系统时间是不会更新的。

如下图：



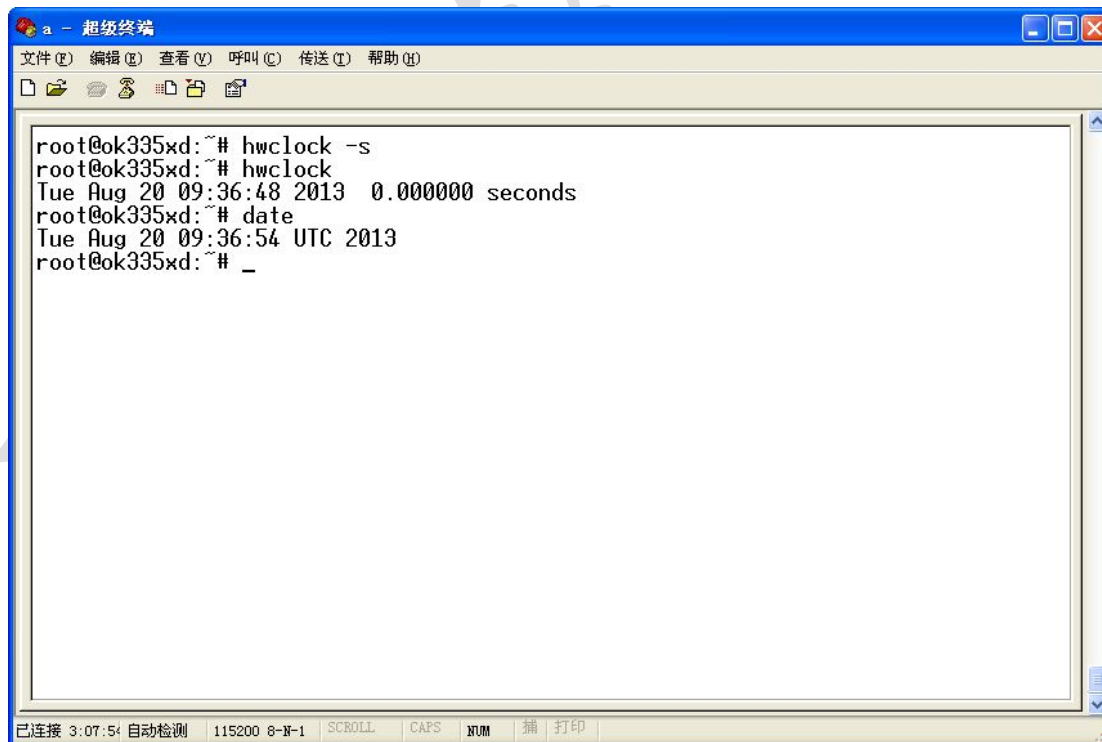
```

a - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335xd:~# hwclock
Wed Aug 21 13:21:22 2013 0.000000 seconds
root@ok335xd:~# hwclock -w
root@ok335xd:~# hwclock
Tue Aug 20 09:33:33 2013 0.000000 seconds
root@ok335xd:~# _
已连接 3:04:24 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

## 5. 断电重启系统

hwclock -s 读取 RTC 时间，并同步给系统时间。如下图：



```

a - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335xd:~# hwclock -s
root@ok335xd:~# hwclock
Tue Aug 20 09:36:48 2013 0.000000 seconds
root@ok335xd:~# date
Tue Aug 20 09:36:54 UTC 2013
root@ok335xd:~# _
已连接 3:07:54 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

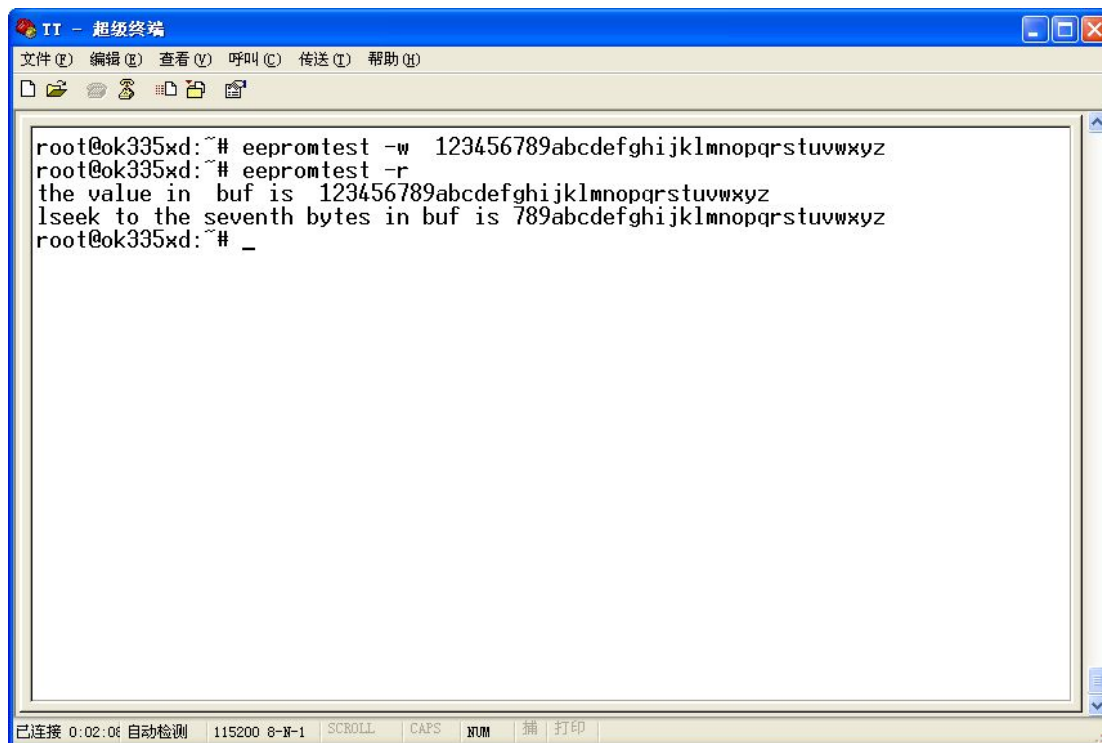
```

此时，软、硬件时间已经同步，说明 RTC 工作正常，驱动没问题。

### 3.3.15 EEPROM 测试

在终端中输入如下命令：  
`eeptest -w 123456789abcdefghijklmnopqrstuvwxyz`  
`eeptest -r`

在测试终端中显示如下：



```

root@ok335xd:~# eepertest -w 123456789abcdefghijklmnopqrstuvwxyz
root@ok335xd:~# eepertest -r
the value in buf is 123456789abcdefghijklmnopqrstuvwxyz
lseek to the seventh bytes in buf is 789abcdefghijklmnopqrstuvwxyz
root@ok335xd:~# _
    
```

“123456789abcdefghijklmnopqrstuvwxyz”为用户测试字符串，可以任意输入，如果是任意输入的字符，”eepertest -r”第一个 buf 值为用户输入的源字符串，第二个 buf 显示的是第六个字符以后的字符串。



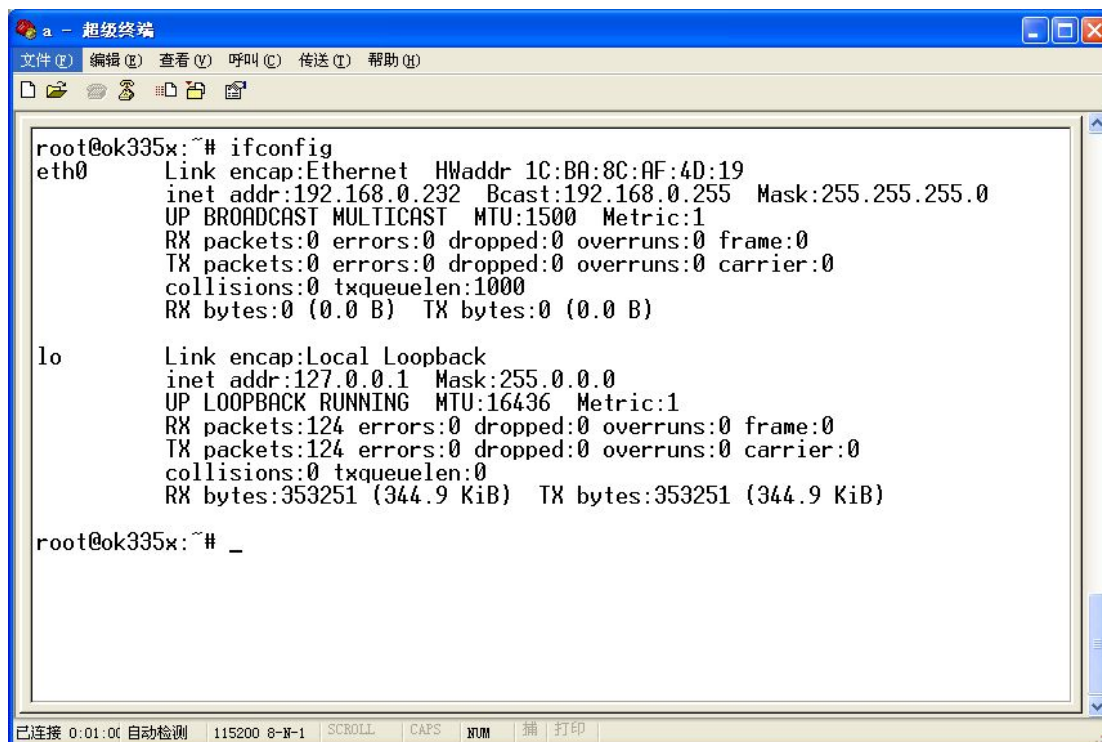
### 3.3.16 GPRS 测试

**注意：**该项功能默认情况下可以不进行测试，如果客户有 GPRS 的需求，再按照下面的步骤操作。

GPRS 模块与开发板之间采用串口连接，客户可以使用飞凌公司自产的 GPRS 模块，也可以使用自己购买的串口 GPRS 模块。

1. 在确保模块和串口连接、上电 ok 情况下，启动开发板子，进入命令行终端，查看板子现有网络接口，  
root@ok335xd:/# ifconfig

如下图：



```

root@ok335x:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 1C:BA:8C:AF:4D:19
          inet addr:192.168.0.232  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

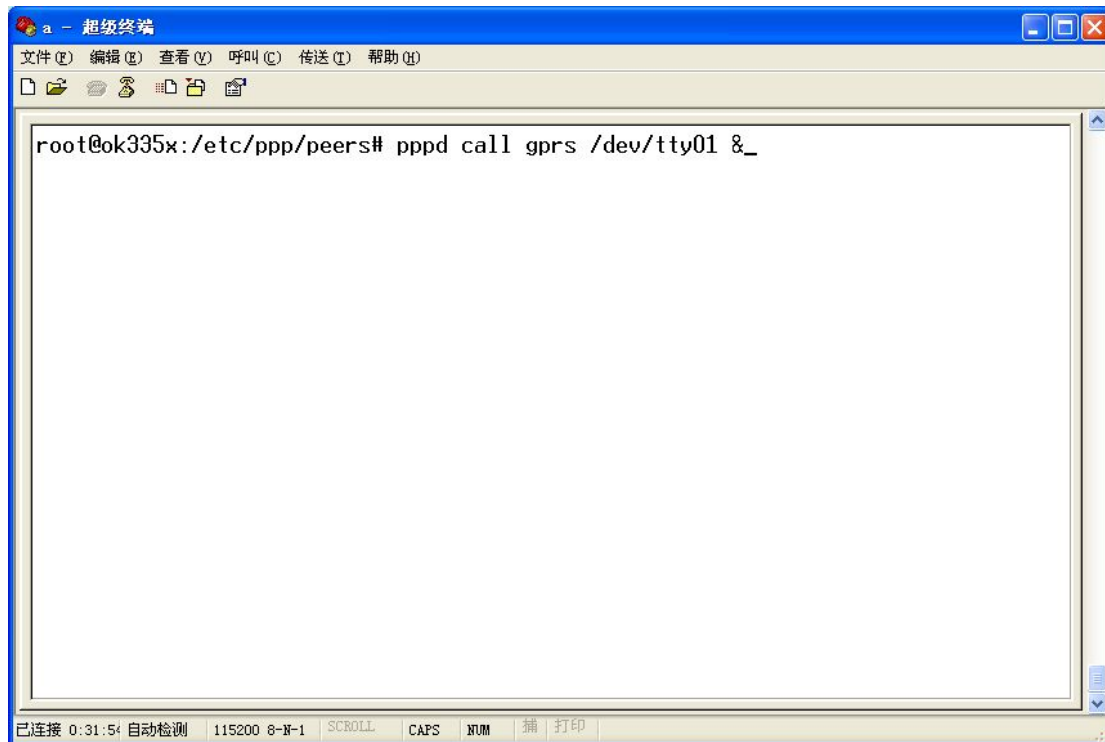
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:353251 (344.9 KiB)  TX bytes:353251 (344.9 KiB)

root@ok335x:~# _
    
```

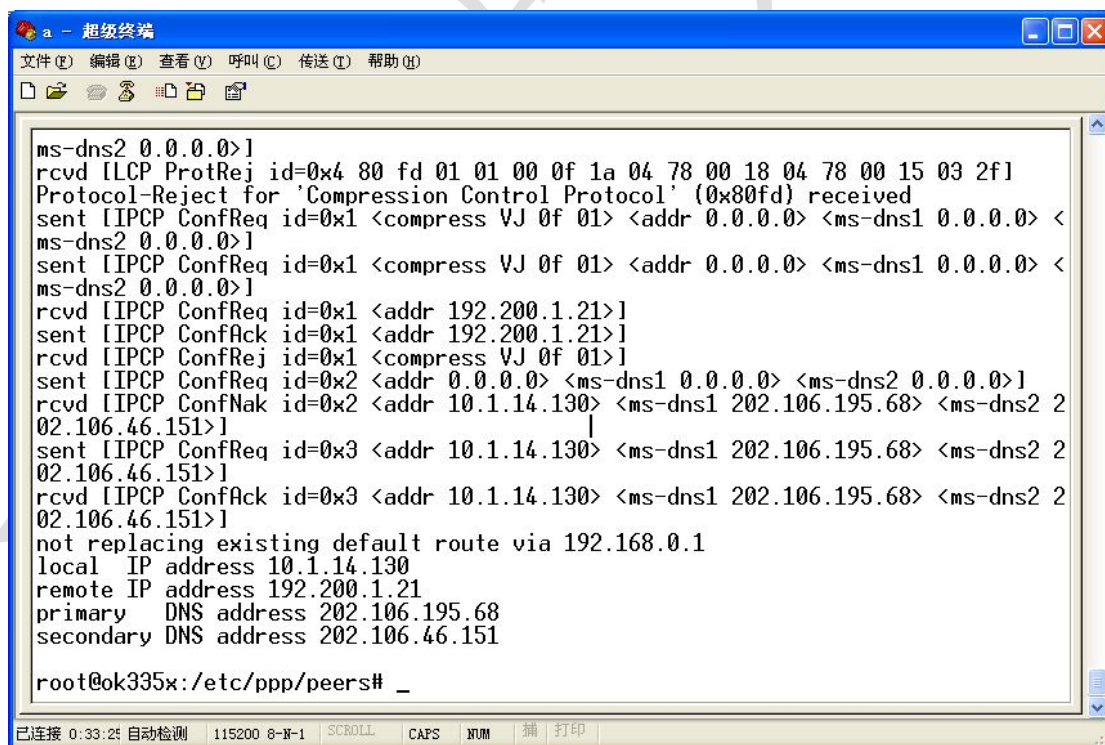
创建 pppd 拨号链接，

```
root@ok335xd:/# pppd call gprs /dev/ttyO1 &
```

如下图：



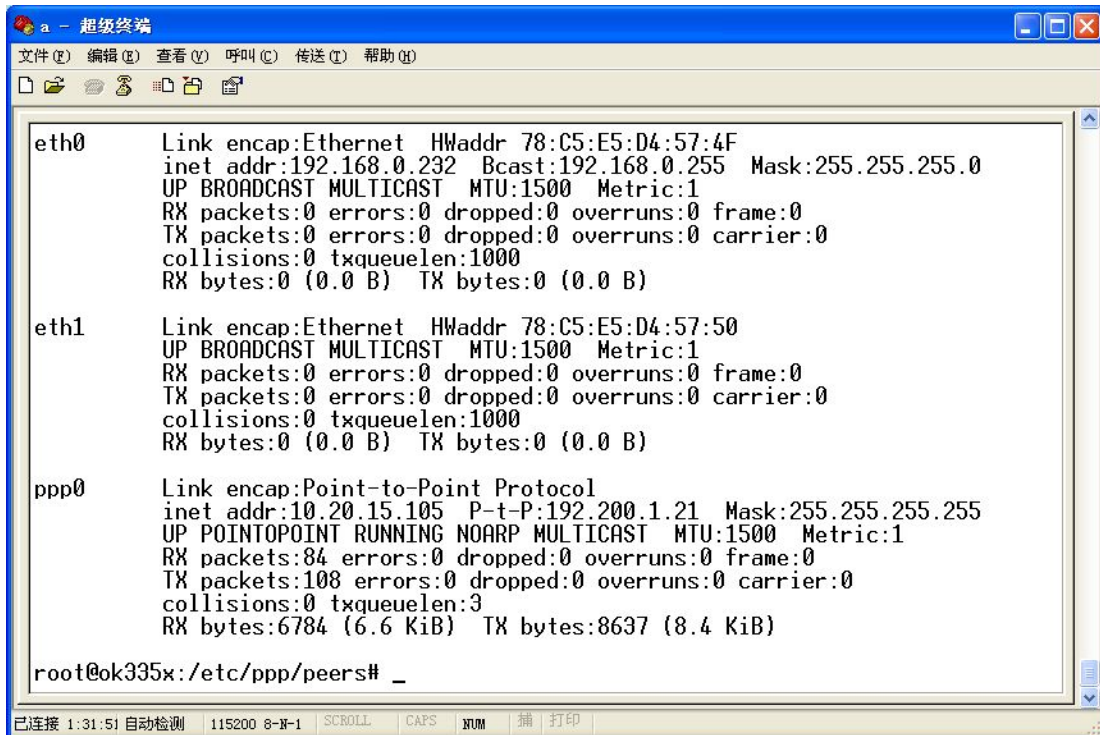
如果终端打印如下图所示的信息，说明拨号链接创建成功



再次查看网络配置，拨号链接创建成功后会出现 ppp0 网络接口

```
root@ok335xd:/# ifconfig
```

如下图：



```

a - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

eth0    Link encap:Ethernet  HWaddr 78:C5:E5:D4:57:4F
        inet addr:192.168.0.232  Bcast:192.168.0.255  Mask:255.255.255.0
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1    Link encap:Ethernet  HWaddr 78:C5:E5:D4:57:50
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0    Link encap:Point-to-Point Protocol
        inet addr:10.20.15.105  P-t-P:192.200.1.21  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
        RX packets:84 errors:0 dropped:0 overruns:0 frame:0
        TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:3
        RX bytes:6784 (6.6 KiB)  TX bytes:8637 (8.4 KiB)

root@ok335x:/etc/ppp/peers# _

```

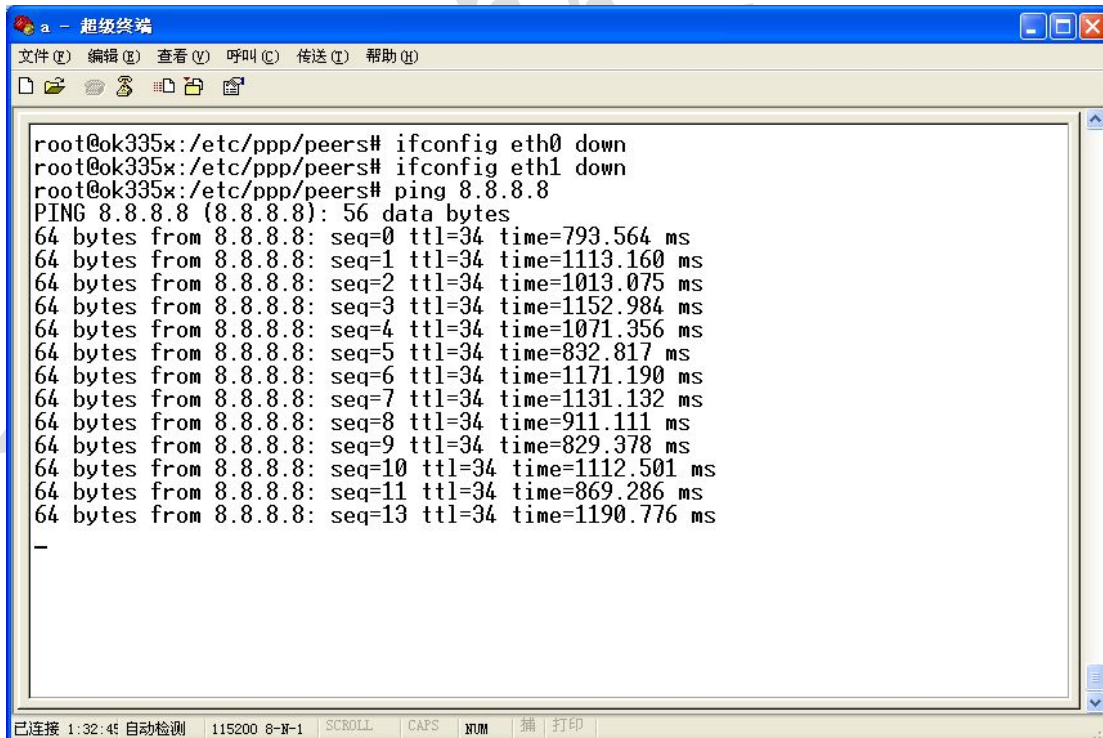
2. 关闭有线网卡 ( 以免与拨号初始化冲突 ), 使用 ping 命令通过 ppp0 连接外网

```
root@ok335xd:/# ifconfig eth0 down
```

```
root@ok335xd:/# ifconfig eth1 down
```

```
root@ok335xd:/# ping 8.8.8.8
```

如果可以 ping 通外网, 说明 gprs 拨号上网功能正常, 如下图:



```

a - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

root@ok335x:/etc/ppp/peers# ifconfig eth0 down
root@ok335x:/etc/ppp/peers# ifconfig eth1 down
root@ok335x:/etc/ppp/peers# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=34 time=793.564 ms
64 bytes from 8.8.8.8: seq=1 ttl=34 time=1113.160 ms
64 bytes from 8.8.8.8: seq=2 ttl=34 time=1013.075 ms
64 bytes from 8.8.8.8: seq=3 ttl=34 time=1152.984 ms
64 bytes from 8.8.8.8: seq=4 ttl=34 time=1071.356 ms
64 bytes from 8.8.8.8: seq=5 ttl=34 time=832.817 ms
64 bytes from 8.8.8.8: seq=6 ttl=34 time=1171.190 ms
64 bytes from 8.8.8.8: seq=7 ttl=34 time=1131.132 ms
64 bytes from 8.8.8.8: seq=8 ttl=34 time=911.111 ms
64 bytes from 8.8.8.8: seq=9 ttl=34 time=829.378 ms
64 bytes from 8.8.8.8: seq=10 ttl=34 time=1112.501 ms
64 bytes from 8.8.8.8: seq=11 ttl=34 time=869.286 ms
64 bytes from 8.8.8.8: seq=13 ttl=34 time=1190.776 ms
-

```

### 3.3.17 USB 3G 测试

**注意：**

本产品目前支持以下 3G 上网卡：

**WCDMA：华为 E1750、E261、中兴 MF190。**

**CDMA2000：中兴 AC2787、AC582。**

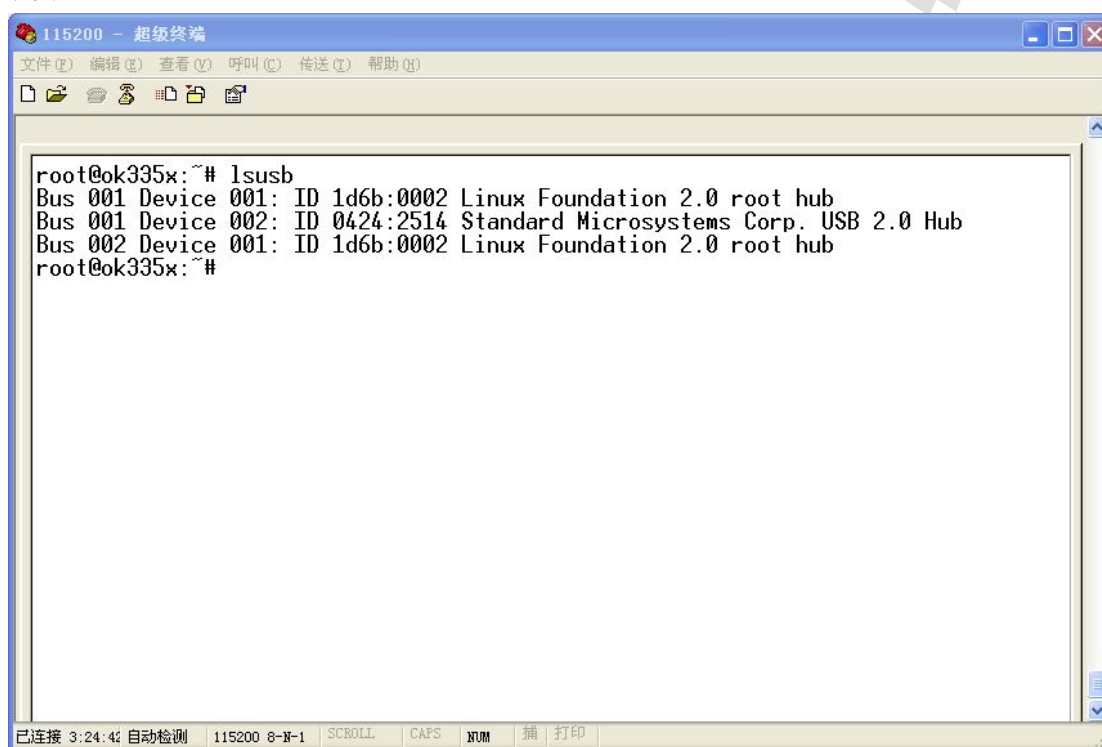
**TDSCDMA：中兴 A356。**

**FL-AD3812 上网卡忽略 3)、4)步。**

1) 开发板上电，打开超级终端。输入命令查看 USB 状态。

```
root@ok335x:~# lsusb
```

如下图：

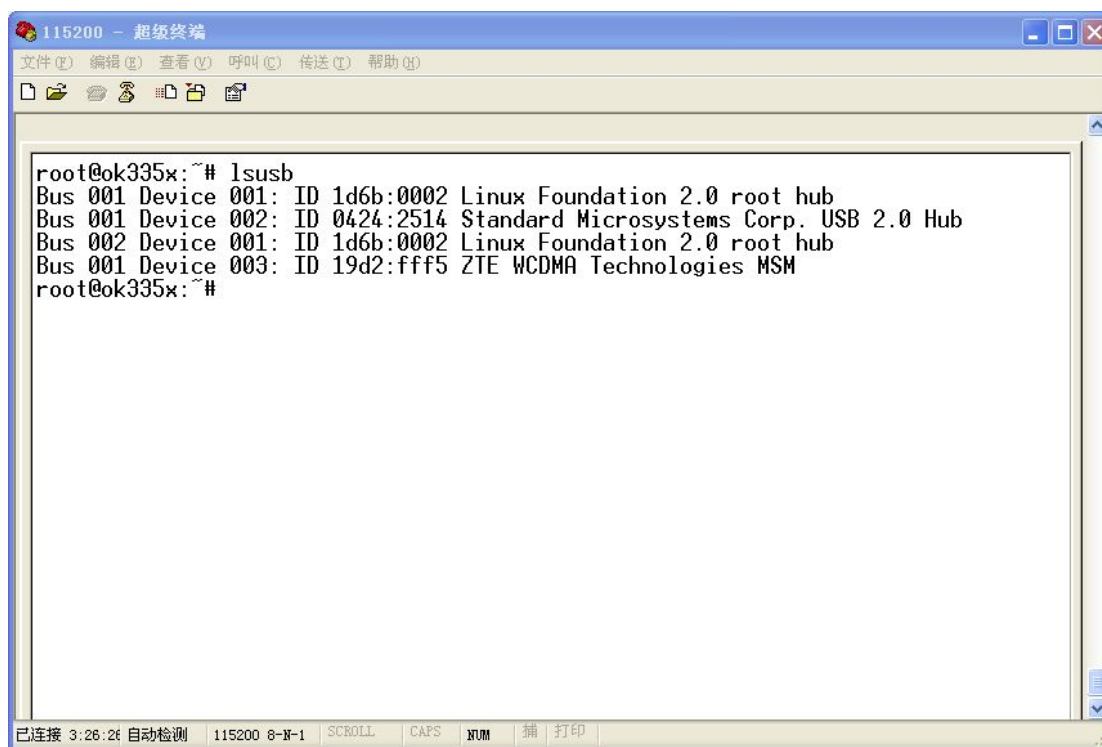


```
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@ok335x:~#
```

2) 插入上述指定的 3G 上网卡，再次输入命令查看 USB 状态，可以看到插入的 3G 上网卡信息。

```
root@ok335x:~# lsusb
```

如下图：



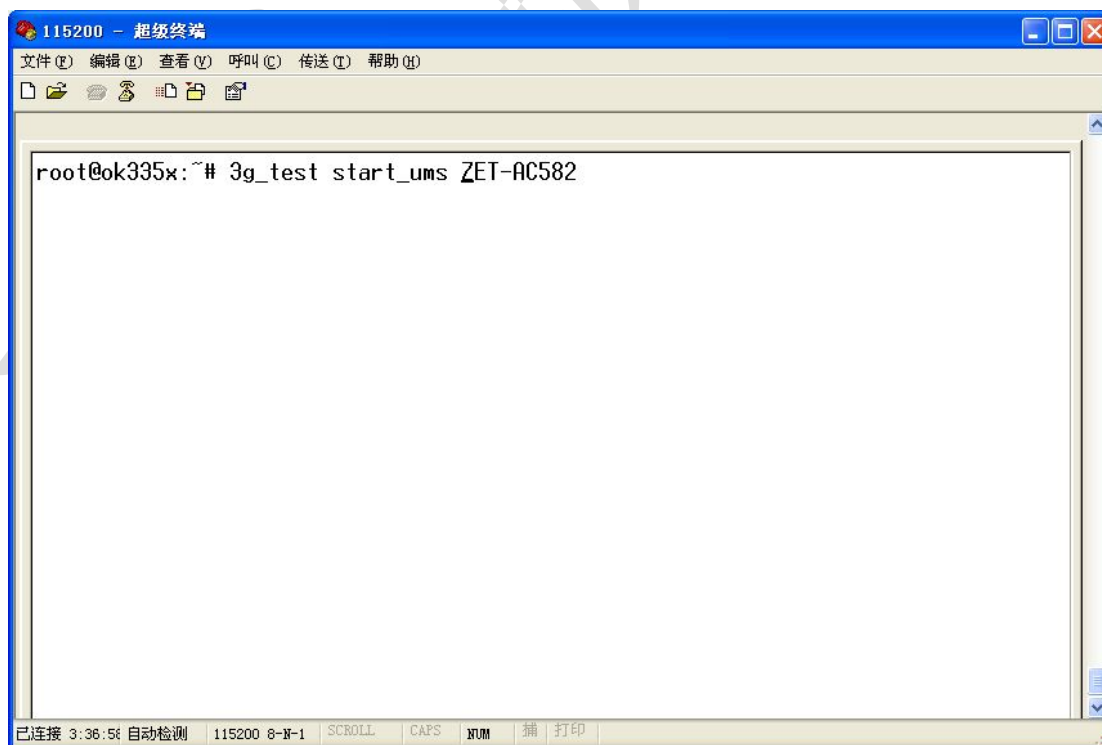
```

root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 19d2:fff5 ZTE WCDMA Technologies MSM
root@ok335x:~#
    
```

3) 输入命令进行 USB 模式转换，此处以中兴 AC582 上网卡为例，下同。

```
root@ok335x:~# 3g_test start_ums ZET-AC582
```

如下图：



```

root@ok335x:~# 3g_test start_ums ZET-AC582
    
```



若转化成功输出如下信息，可看见“ttyUSBx”串口生成：

```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
/forlinx/bin/3g_test: line 55: 0: command not found

SUCCESS: starting usb_modeswitch.
start_usb_modeswitch
root@ok335x:~# [ 2197.488978] usb 1-1.4: USB disconnect, device number 3
[ 2198.751494] usb 1-1.4: new full-speed USB device number 4 using musb-hdrc
[ 2198.872822] usb 1-1.4: New USB device found, idVendor=19d2, idProduct=fff1
[ 2198.880004] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber
=0
[ 2198.887655] usb 1-1.4: Product: ZTE CDMA Tech
[ 2198.892210] usb 1-1.4: Manufacturer: ZTE, Incorporated
[ 2198.900751] option 1-1.4:1.0: GSM modem (1-port) converter detected
[ 2198.908286] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB0
[ 2198.920924] option 1-1.4:1.1: GSM modem (1-port) converter detected
[ 2198.928716] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB1
[ 2198.937267] option 1-1.4:1.2: GSM modem (1-port) converter detected
[ 2198.944809] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB2
[ 2198.953281] option 1-1.4:1.3: GSM modem (1-port) converter detected
[ 2198.960564] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB3
[ 2198.969164] option 1-1.4:1.4: GSM modem (1-port) converter detected
[ 2198.976692] usb 1-1.4: GSM modem (1-port) converter now attached to ttyUSB4
[ 2198.985166] scsi1 : usb-storage 1-1.4:1.5
[ 2199.994265] scsi 1:0:0:0: Direct-Access      ZTE          USB Storage FFF1 2.31 PQ
: 0 ANSI: 2
[ 2200.018553] sd 1:0:0:0: [sda] Attached SCSI removable disk
已连接 3:37:14 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

4) 再次输入命令查看 USB 模式转换结果，可看见 USB 设备 PID 的变化

root@ok335x:~# lsusb

如下图：

```

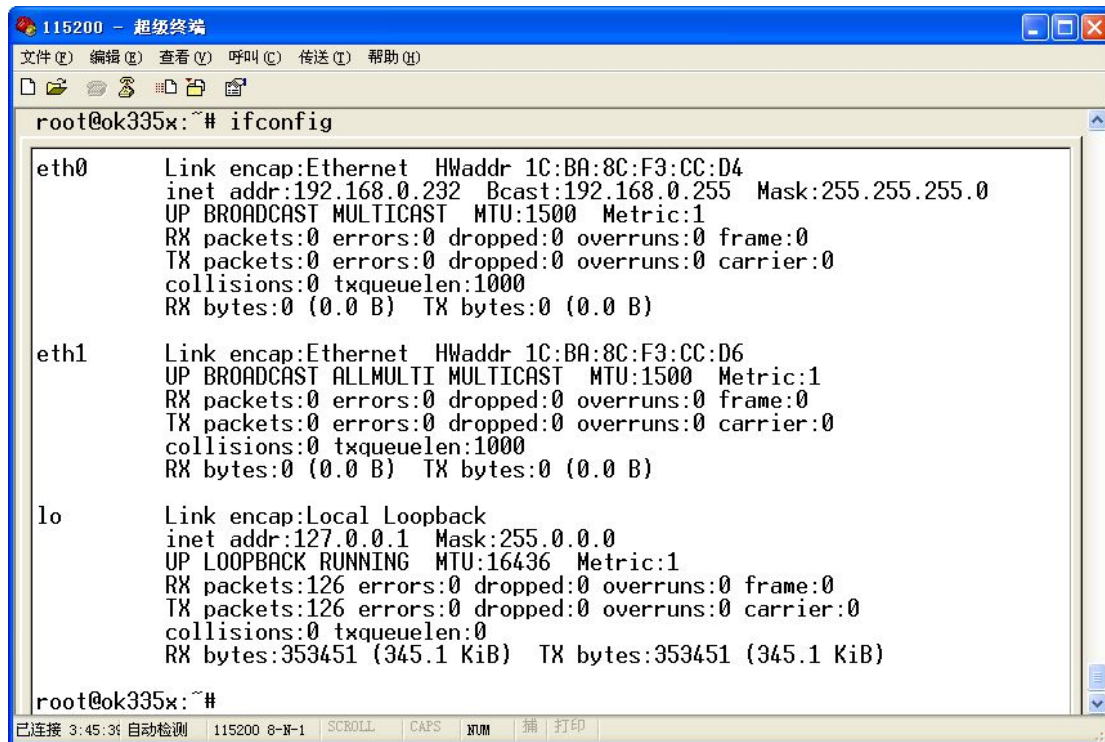
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 004: ID 19d2:fff1 ZTE WCDMA Technologies MSM
root@ok335x:~# _
已连接 3:43:21 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

5) 输入命令查看当前网络状态，可见只有以太网及回环接口。

```
root@ok335x:~# ifconfig
```

如下图：



```
root@ok335x:~# ifconfig

eth0      Link encap:Ethernet  HWaddr 1C:BA:8C:F3:CC:D4
          inet addr:192.168.0.232  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 1C:BA:8C:F3:CC:D6
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

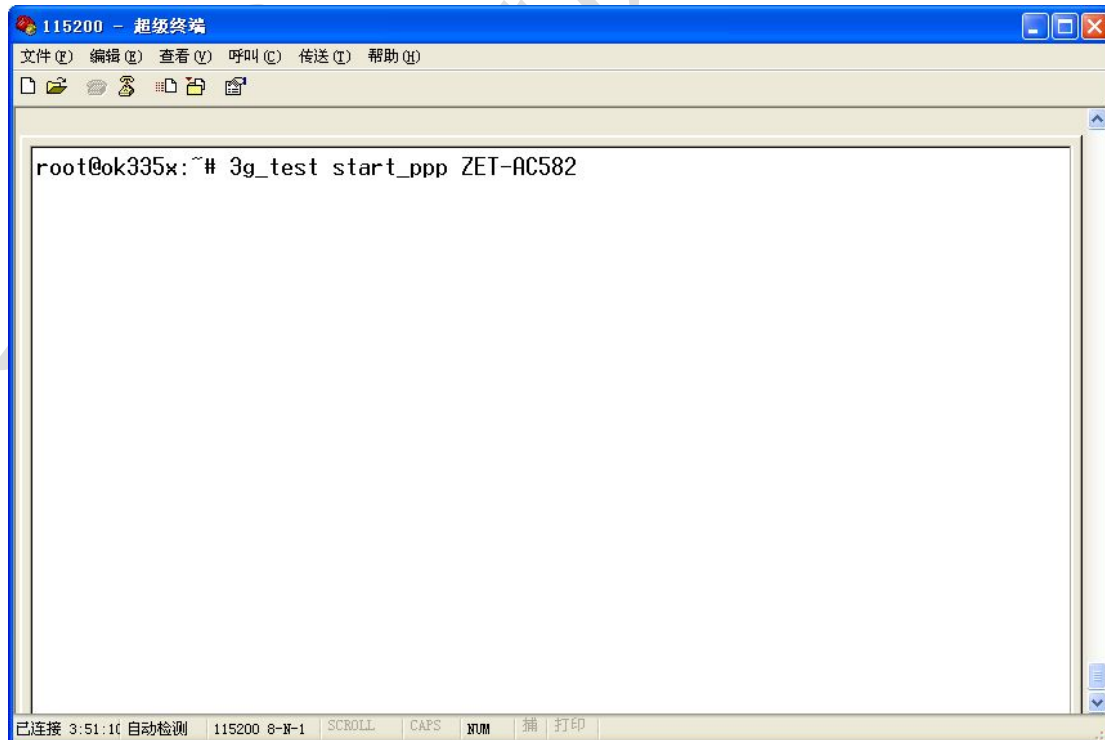
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:353451 (345.1 KiB)  TX bytes:353451 (345.1 KiB)

root@ok335x:~#
```

6) 输入命令进行拨号。

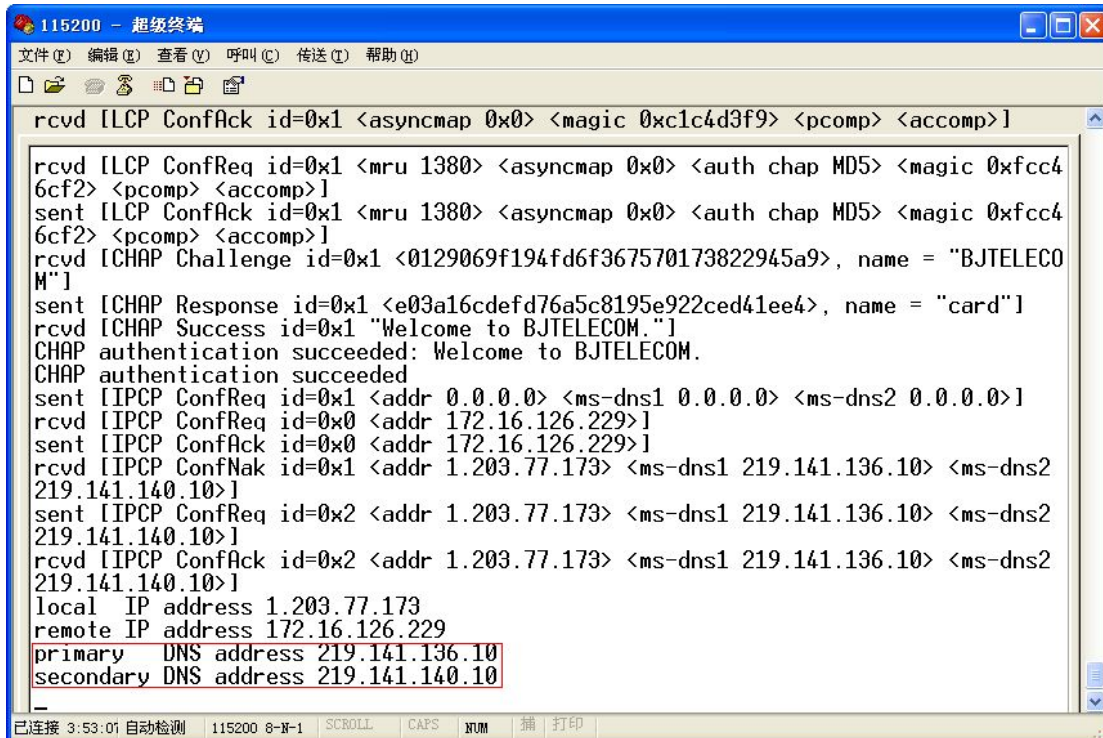
```
root@ok335x:~# 3g_test start_ppp ZET-AC582
```

如下图：



```
root@ok335x:~# 3g_test start_ppp ZET-AC582
```

若拨号成功则打印如下信息，可见分配的 DNS：



```

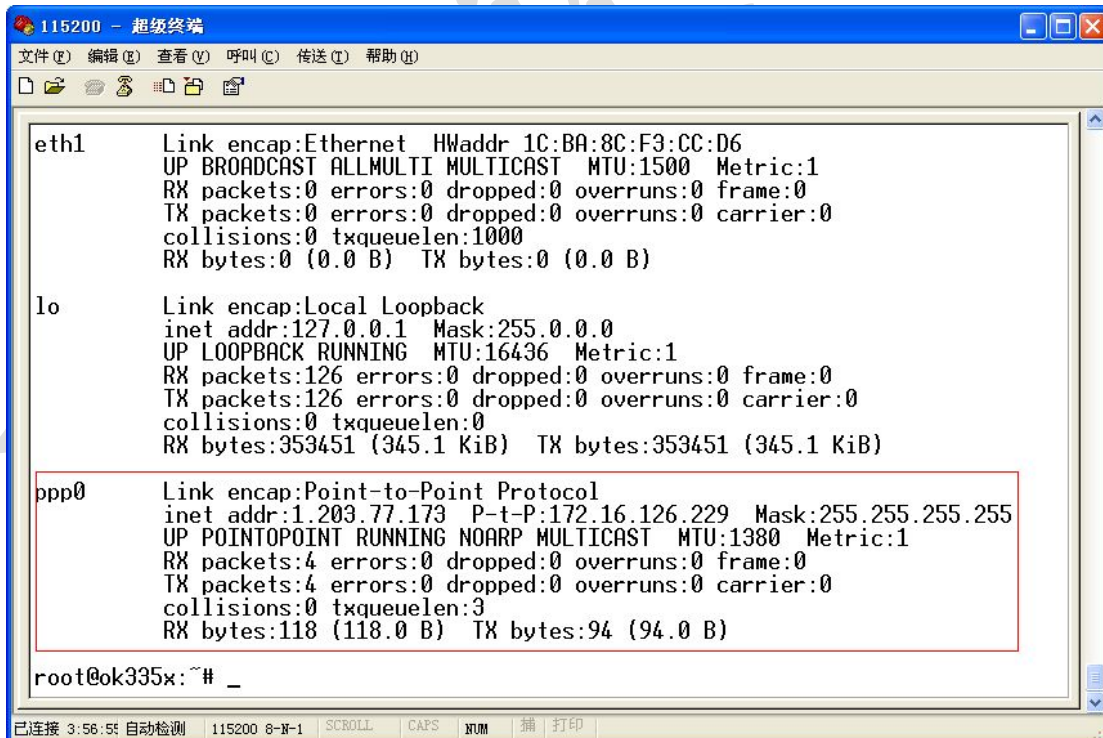
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xc1c4d3f9> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <mru 1380> <asyncmap 0x0> <auth chap MD5> <magic 0xfcc4
6cf2> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <mru 1380> <asyncmap 0x0> <auth chap MD5> <magic 0xfcc4
6cf2> <pcomp> <accomp>]
rcvd [CHAP Challenge id=0x1 <0129069f194fd6f367570173822945a9>, name = "BJTELECO
M"]
sent [CHAP Response id=0x1 <e03a16cdefd76a5c8195e922ced41ee4>, name = "card"]
rcvd [CHAP Success id=0x1 "Welcome to BJTELECOM."]
CHAP authentication succeeded: Welcome to BJTELECOM.
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x0 <addr 172.16.126.229>]
sent [IPCP ConfAck id=0x0 <addr 172.16.126.229>]
rcvd [IPCP ConfNak id=0x1 <addr 1.203.77.173> <ms-dns1 219.141.136.10> <ms-dns2
219.141.140.10>]
sent [IPCP ConfReq id=0x2 <addr 1.203.77.173> <ms-dns1 219.141.136.10> <ms-dns2
219.141.140.10>]
rcvd [IPCP ConfAck id=0x2 <addr 1.203.77.173> <ms-dns1 219.141.136.10> <ms-dns2
219.141.140.10>]
local IP address 1.203.77.173
remote IP address 172.16.126.229
primary DNS address 219.141.136.10
secondary DNS address 219.141.140.10
已连接 3:53:01 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

7) 输入命令查看当前网络状态，可见新增的 ppp0 接口。

root@ok335x:~# ifconfig

如下图：



```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
eth1      Link encap:Ethernet  HWaddr 1C:BA:8C:F3:CC:D6
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:353451 (345.1 KiB)  TX bytes:353451 (345.1 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:1.203.77.173  P-t-P:172.16.126.229  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1380  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:118 (118.0 B)  TX bytes:94 (94.0 B)

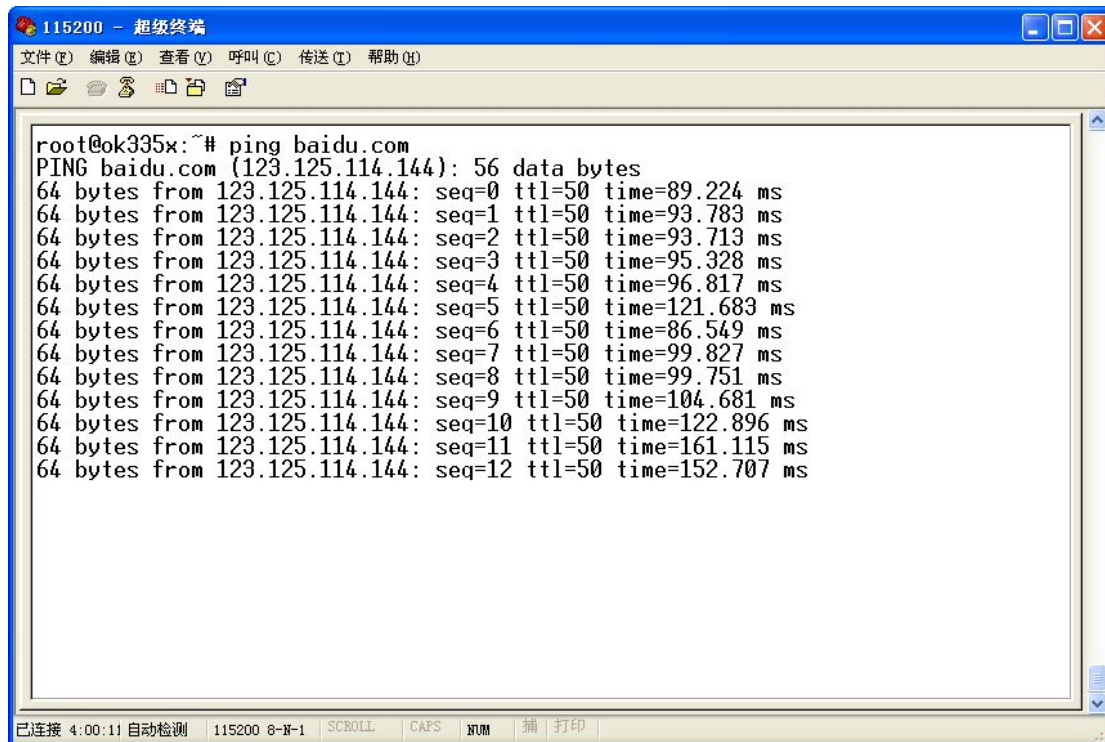
root@ok335x:~# _
已连接 3:56:55 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印

```

8) 输入命令 ping 任意广域网域名或 IP，若 PING 通则 3G 上网卡已正常工作。

root@ok335x:~# ping baidu.com

如下图：



```

root@ok335x:~# ping baidu.com
PING baidu.com (123.125.114.144): 56 data bytes
64 bytes from 123.125.114.144: seq=0 ttl=50 time=89.224 ms
64 bytes from 123.125.114.144: seq=1 ttl=50 time=93.783 ms
64 bytes from 123.125.114.144: seq=2 ttl=50 time=93.713 ms
64 bytes from 123.125.114.144: seq=3 ttl=50 time=95.328 ms
64 bytes from 123.125.114.144: seq=4 ttl=50 time=96.817 ms
64 bytes from 123.125.114.144: seq=5 ttl=50 time=121.683 ms
64 bytes from 123.125.114.144: seq=6 ttl=50 time=86.549 ms
64 bytes from 123.125.114.144: seq=7 ttl=50 time=99.827 ms
64 bytes from 123.125.114.144: seq=8 ttl=50 time=99.751 ms
64 bytes from 123.125.114.144: seq=9 ttl=50 time=104.681 ms
64 bytes from 123.125.114.144: seq=10 ttl=50 time=122.896 ms
64 bytes from 123.125.114.144: seq=11 ttl=50 time=161.115 ms
64 bytes from 123.125.114.144: seq=12 ttl=50 time=152.707 ms
    
```

附：3g\_test 命令行参数 2 可选列表：

ZET-MF190  
 HW-E1750  
 HW-E261  
 ZET-AC582  
 ZET-AC2787  
 ZET-A356  
 FL-AD3812



### 3.3.18 USB WIFI 测试

注意：

1 本产品目前支持以下 WIFI 网卡：

**BL-LW05-AR5:** 芯片型号 RTL8188CUS

**EP-N8508GS:** 芯片型号 RTL8188CUS。

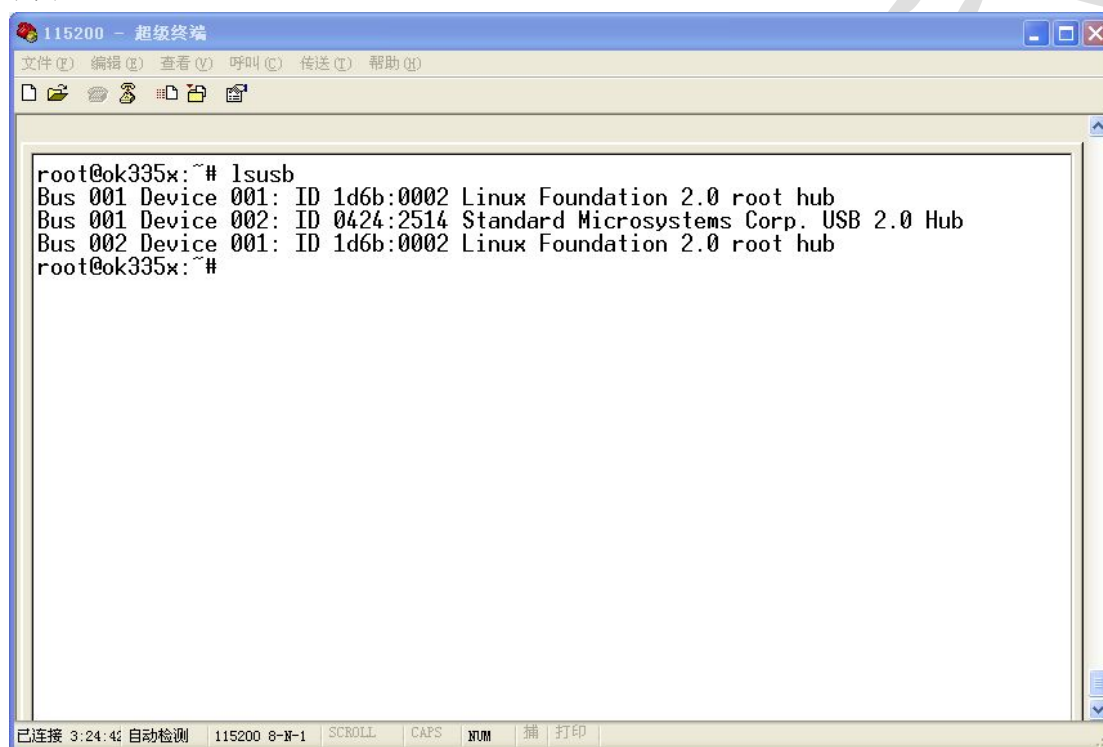
2 不要在开发板上电前插入 USB 网卡，连接 AP 后不要将网卡 down 掉或将 wpa\_supplicant 进程 kill 掉。  
(此问题将在后续版本中解决。)

3 为了减少调试过程中的麻烦，在测试前请确认 AP 工作在 WPA-PSK/WPA2-PSK 模式下（此模式为目前手机、PC 常用的上网模式），并使 AP 与开发板在较近距离内，并确保无障碍物。

1) 开发板上电，打开超级终端。输入命令查看 USB 状态。

```
root@ok335x:~# lsusb
```

如下图：



```

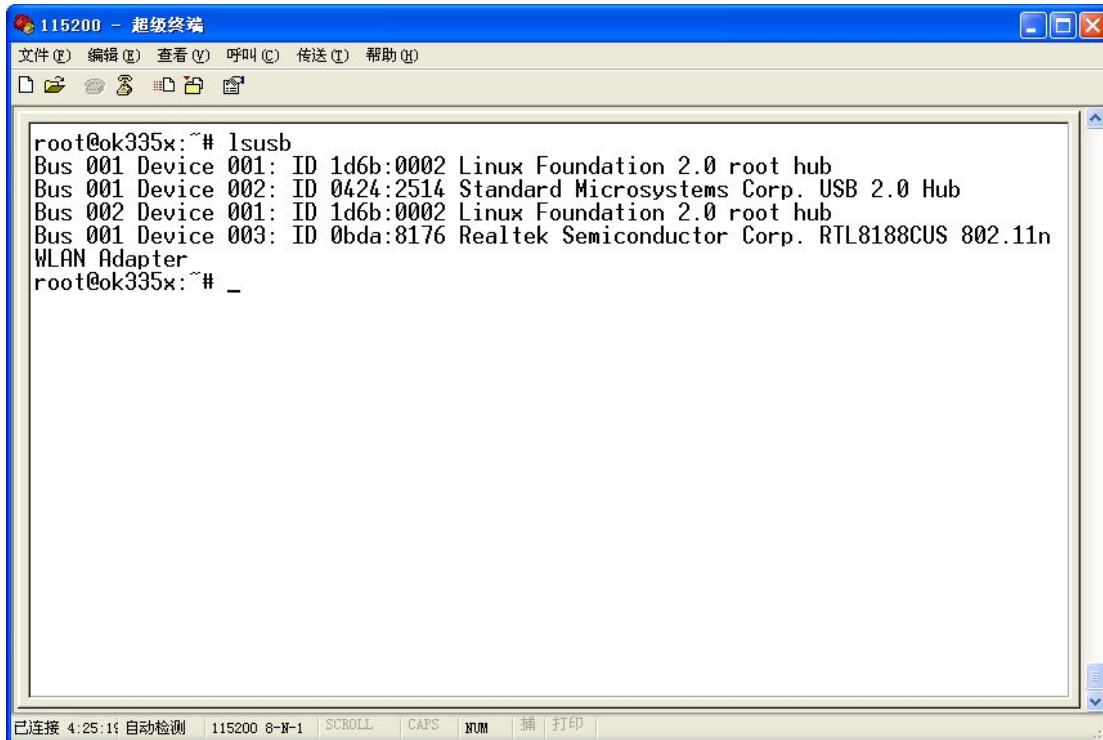
root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@ok335x:~#
    
```



2) 插入上述指定的 WIFI 网卡，再次输入命令查看 USB 状态，可以看到插入的 WIFI 网卡信息。

```
root@ok335x:~# lsusb
```

如下图：



```

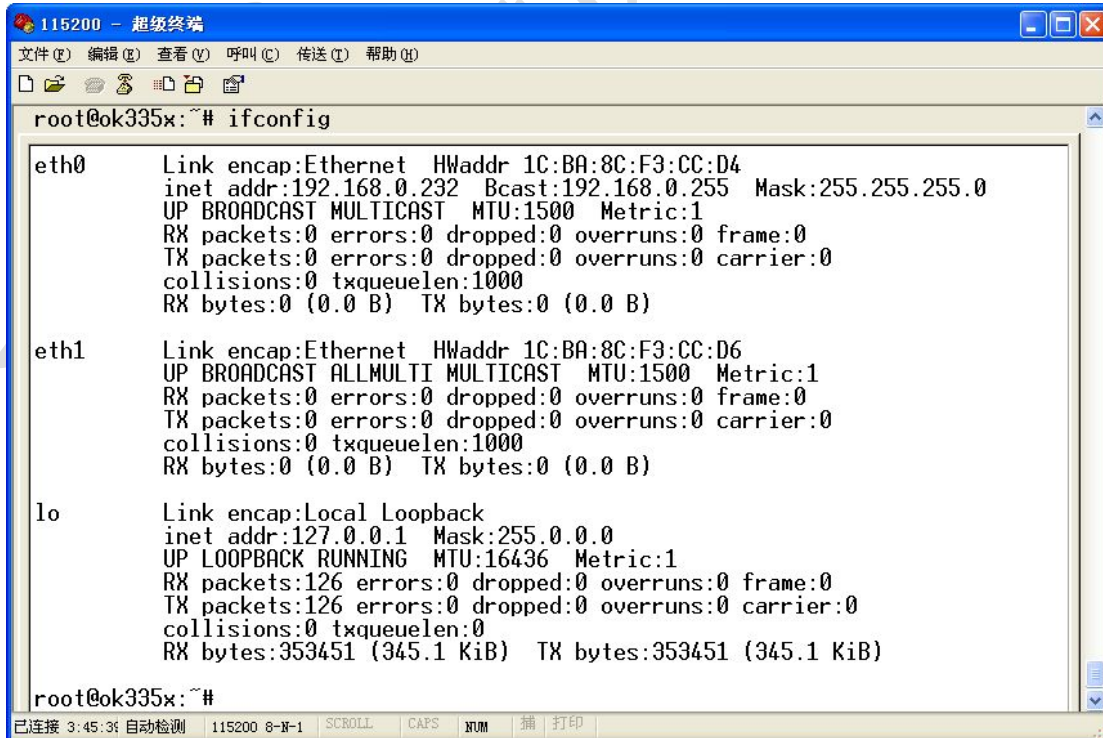
root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n
WLAN Adapter
root@ok335x:~# _

```

3) 输入命令查看当前网络状态，可见只有以太网及回环接口。

```
root@ok335x:~# ifconfig
```

如下图：



```

root@ok335x:~# ifconfig

eth0      Link encap:Ethernet  HWaddr 1C:BA:8C:F3:CC:D4
          inet addr:192.168.0.232  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 1C:BA:8C:F3:CC:D6
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:353451 (345.1 KiB)  TX bytes:353451 (345.1 KiB)

root@ok335x:~#

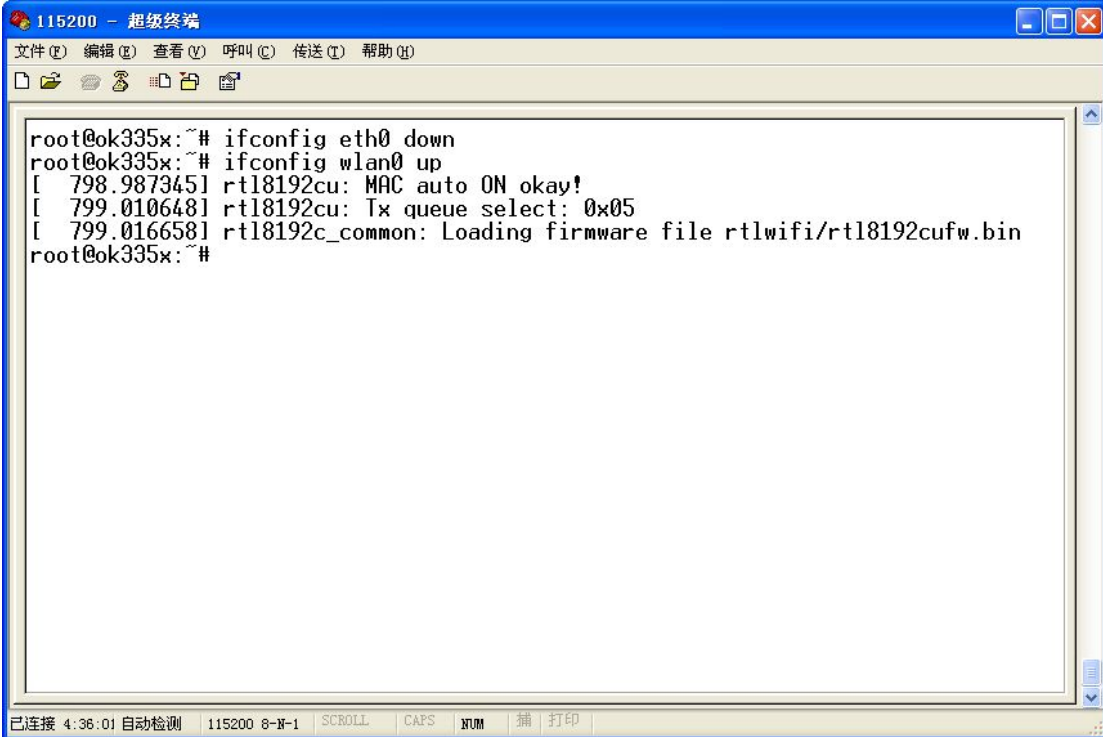
```

4) 输入命令 down 掉以太网卡、up WIFI 网卡。

```
root@ok335x:~# ifconfig eth0 down
```

```
root@ok335x:~# ifconfig wlan0 up
```

如下图：



The screenshot shows a terminal window titled "115200 - 超级终端". The terminal displays the following commands and output:

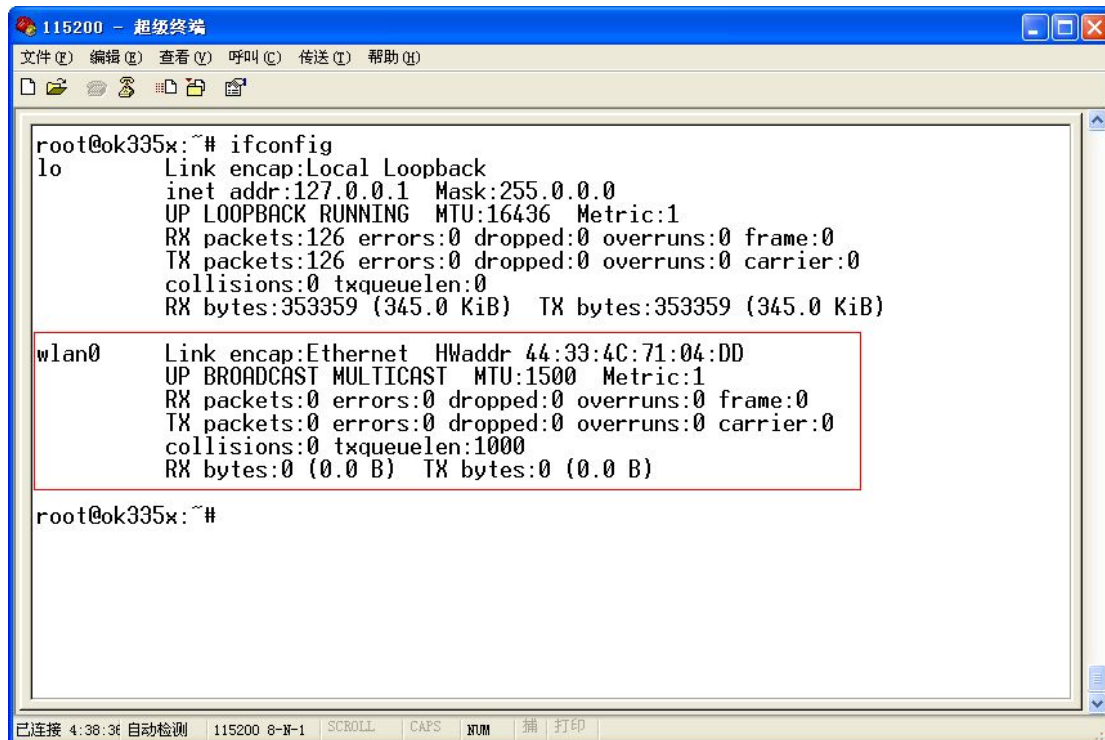
```
root@ok335x:~# ifconfig eth0 down
root@ok335x:~# ifconfig wlan0 up
[ 798.987345] rtl8192cu: MAC auto ON okay!
[ 799.010648] rtl8192cu: Tx queue select: 0x05
[ 799.016658] rtl8192c_common: Loading firmware file rtlwifi/rtl8192cufw.bin
root@ok335x:~#
```

The terminal window has a menu bar with options: 文件(F), 编辑(E), 查看(V), 呼叫(C), 传送(T), 帮助(H). The status bar at the bottom shows: 已连接 4:36:01 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印.

5) 输入命令查看当前网络状态，可见新增的 wlan0 接口。由于还没有连接 AP，可见接收、发送数据包计数为 0。

```
root@ok335x:~# ifconfig
```

如下图：



```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

root@ok335x:~# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:126 errors:0 dropped:0 overruns:0 frame:0
            TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:353359 (345.0 KiB)  TX bytes:353359 (345.0 KiB)

wlan0       Link encap:Ethernet  HWaddr 44:33:4C:71:04:DD
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

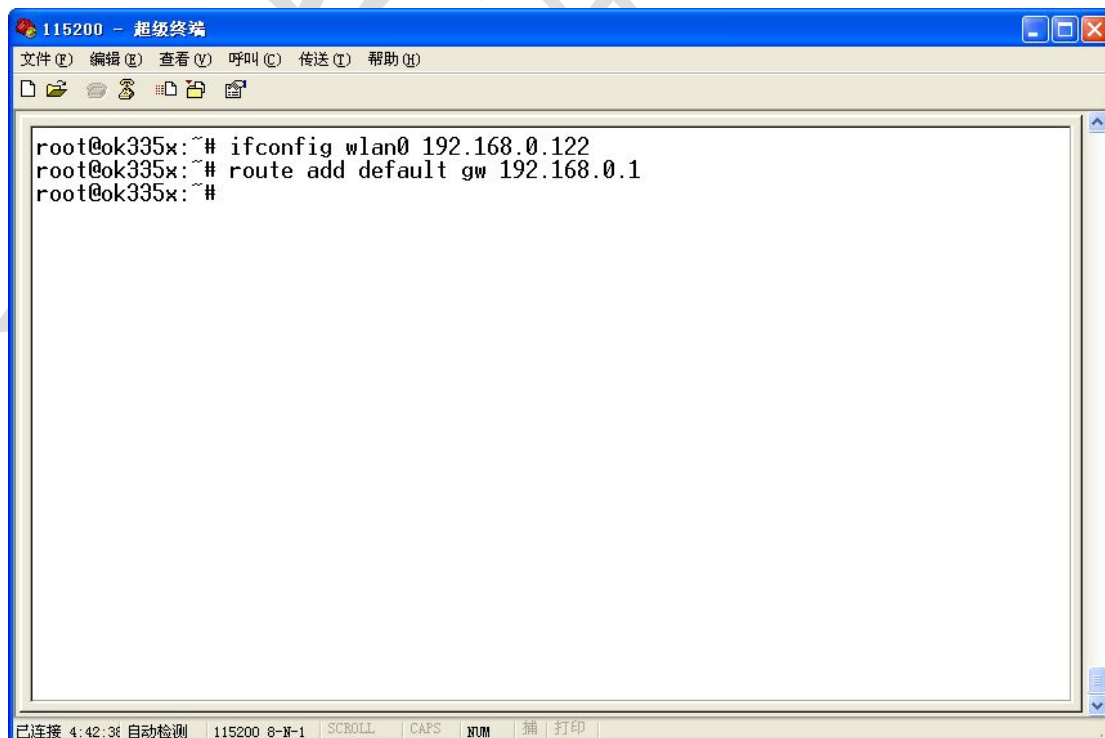
root@ok335x:~#
    
```

6) 输入命令配置 wlan0 的 IP，并添加缺省路由。

```
root@ok335x:~# ifconfig wlan0 192.168.0.122
```

```
root@ok335x:~# route add default gw 192.168.0.1
```

如下图：



```

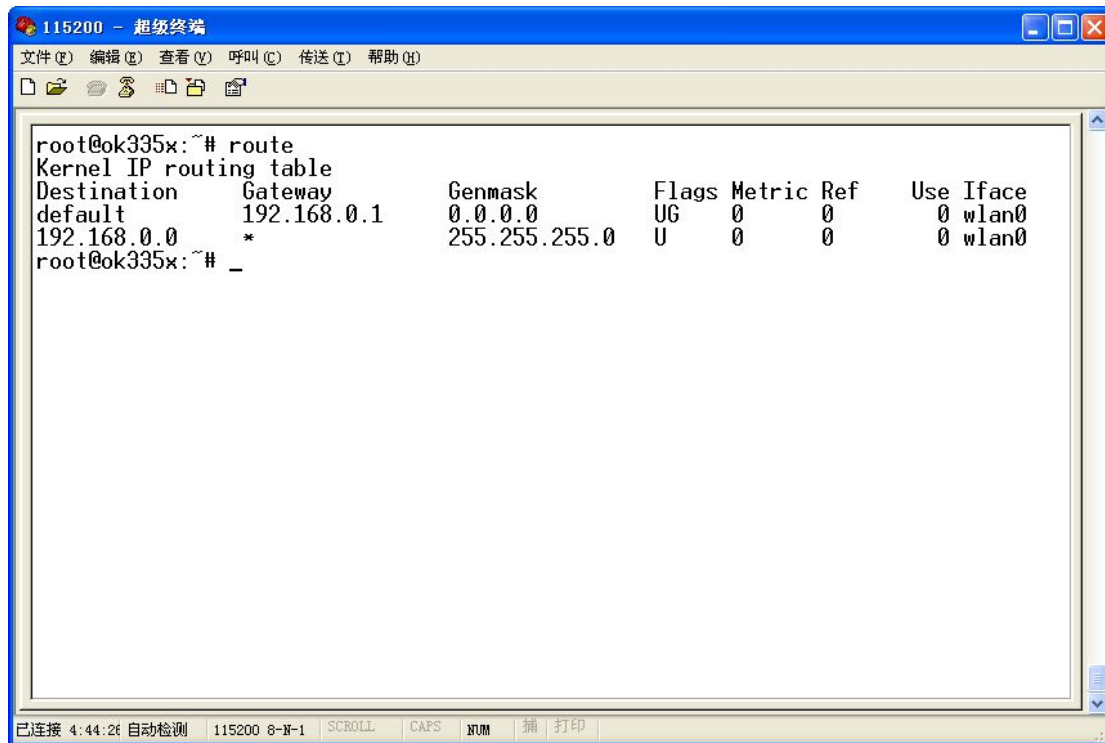
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

root@ok335x:~# ifconfig wlan0 192.168.0.122
root@ok335x:~# route add default gw 192.168.0.1
root@ok335x:~#
    
```

7) 输入命令查看添加路由是否成功。

```
root@ok335x:~# route
```

如下图：



```

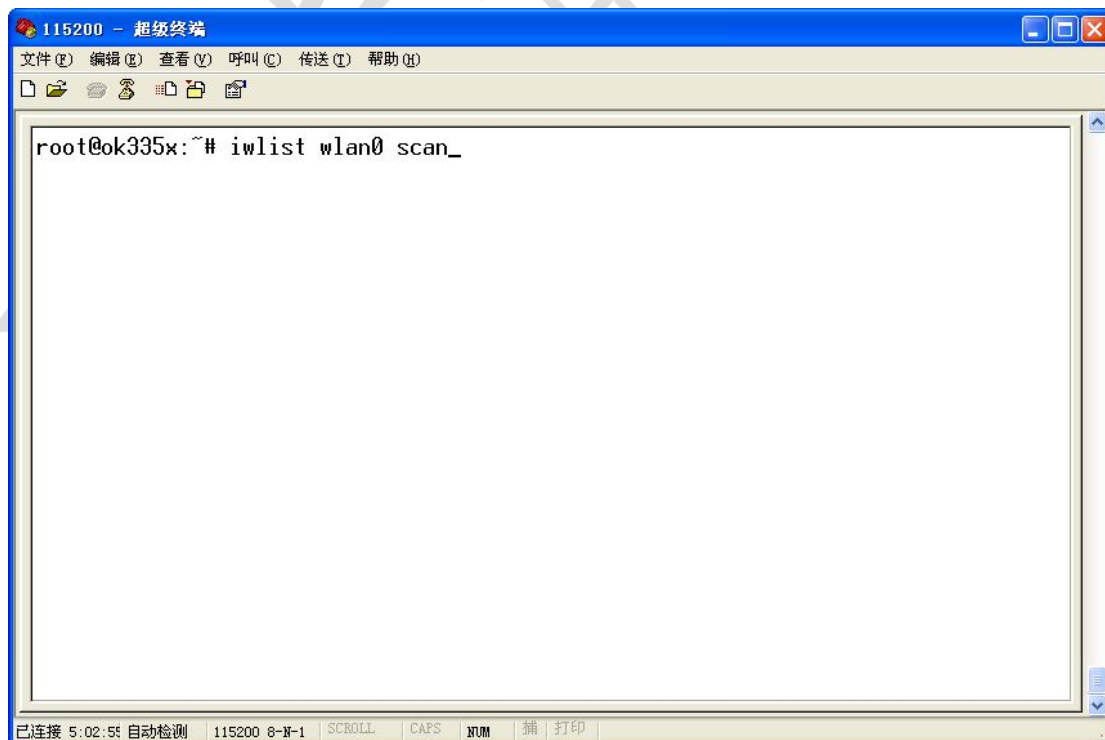
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# route
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
default        192.168.0.1     0.0.0.0         UG    0      0      0 wlan0
192.168.0.0    *               255.255.255.0   U      0      0      0 wlan0
root@ok335x:~# _
已连接 4:44:26 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印

```

8) 输入命令查看扫描周围的 AP。

```
root@ok335x:~# iwlist wlan0 scan
```

如下图：

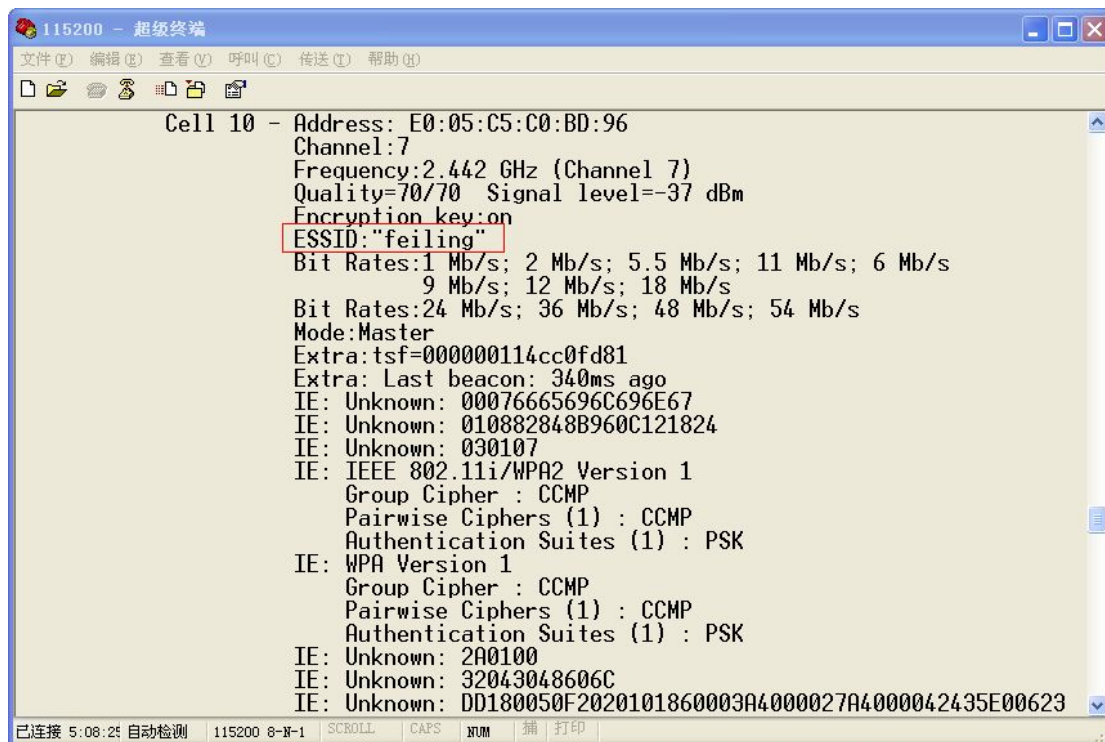


```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# iwlist wlan0 scan_
已连接 5:02:55 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印

```

扫描结果如下，找到所要连接的 AP 的 ESSID，本例选择 “feiling”：



```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

Cell 10 - Address: E0:05:C5:C0:BD:96
Channel:7
Frequency:2.442 GHz (Channel 7)
Quality=70/70 Signal level=-37 dBm
Encryption key:on
ESSID:"feiling"
Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
          9 Mb/s; 12 Mb/s; 18 Mb/s
Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
Mode:Master
Extra:tsf=000000114cc0fd81
Extra: Last beacon: 340ms ago
IE: Unknown: 00076665696C696E67
IE: Unknown: 010882848B960C121824
IE: Unknown: 030107
IE: IEEE 802.11i/WPA2 Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : PSK
IE: WPA Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : PSK
IE: Unknown: 2A0100
IE: Unknown: 32043048606C
IE: Unknown: DD180050F2020101860003A4000027A4000042435E00623

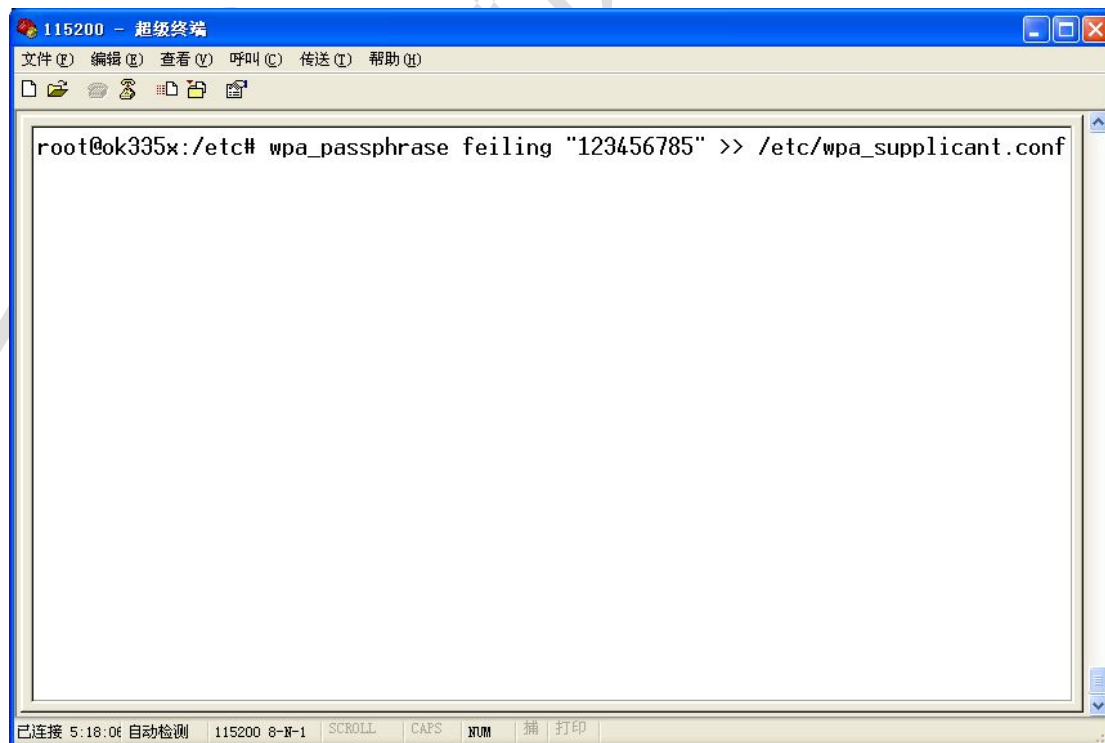
已连接 5:08:24 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
  
```

#### 9) 编辑配置文件 wpa\_supplicant.conf。

若 PSK 密码为 “123456785”，则输入命令如下：

```
root@ok335x:/etc# wpa_passphrase feiling "123456785" >> /etc/wpa_supplicant.conf
```

如下图：



```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

root@ok335x:/etc# wpa_passphrase feiling "123456785" >> /etc/wpa_supplicant.conf

已连接 5:18:06 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
  
```



[illegible]

115200 - 超级终端

文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

#PSK/TKIP

```
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="feiling"
    scan_ssid=1
    key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
    pairwise=TKIP CCMP
    group=CCMP TKIP WEP104 WEP40
    #psk="123456785"
    psk=f7c8f9ad253fefcd5c9344522efb569214e05a2c1d1adcfdbbe28c3db394d705
}
```

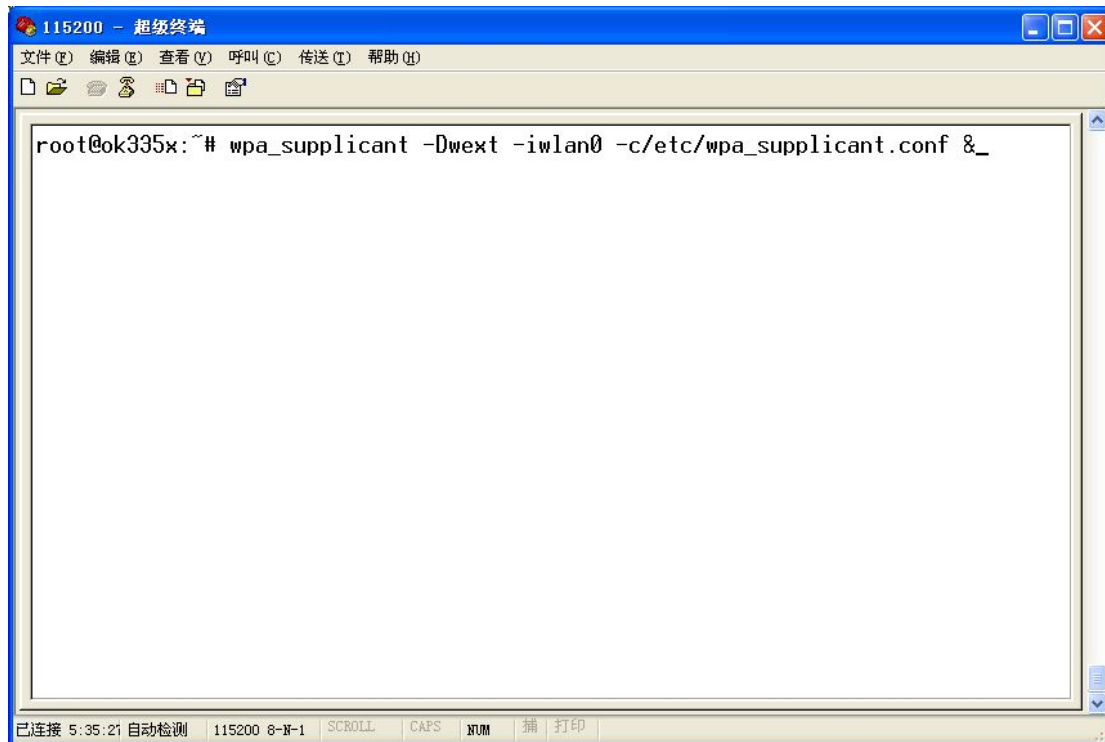
- wpa\_supplicant.conf 1/13 7%

已连接 5:23:31 自动检测 | 115200 8-W-1 | SCROLL CAPS NUM 捕 打印

10) 输入命令连接 AP。

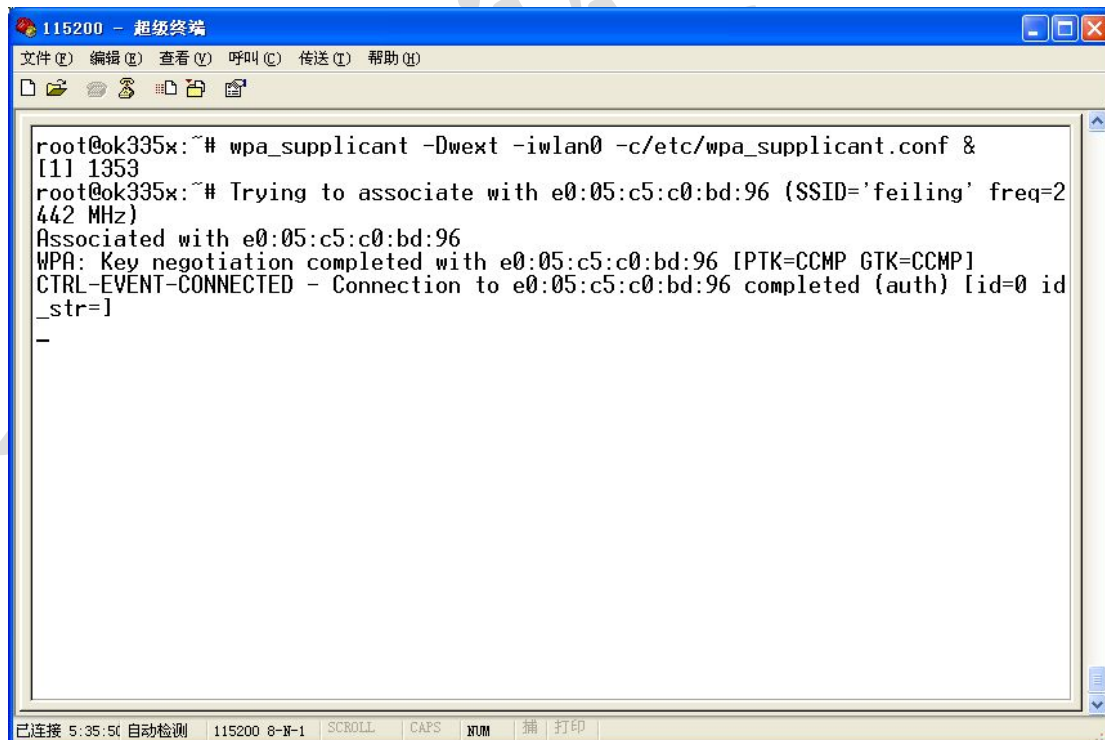
```
root@ok335x:~# wpa_supplicant -Dwext -iwlan0 -c/etc/wpa_supplicant.conf &
```

如下图：



```
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# wpa_supplicant -Dwext -iwlan0 -c/etc/wpa_supplicant.conf &_
已连接 5:35:21 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
```

若连接成功将打印如下信息：



```
115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# wpa_supplicant -Dwext -iwlan0 -c/etc/wpa_supplicant.conf &
[1] 1353
root@ok335x:~# Trying to associate with e0:05:c5:c0:bd:96 (SSID='feiling' freq=2
442 MHz)
Associated with e0:05:c5:c0:bd:96
WPA: Key negotiation completed with e0:05:c5:c0:bd:96 [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to e0:05:c5:c0:bd:96 completed (auth) [id=0 id
_str=]
-
已连接 5:35:54 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
```

- 11) 输入命令查看当前网络状态, 可见 wlan0 接收、发送数据包已有计数, 说明连接 AP 成功。

```
root@ok335x:~# ifconfig
```

如下图:

```

Tx excessive retries:0 Invalid misc:0 Missed beacon:0

eth0      no wireless extensions.
can0      no wireless extensions.

root@ok335x:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:353555 (345.2 KiB)  TX bytes:353555 (345.2 KiB)

wlan0     Link encap:Ethernet  HWaddr 44:33:4C:71:04:DD
          inet addr:192.168.0.122 Bcast:192.168.0.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10571 (10.3 KiB)  TX bytes:288 (288.0 B)

root@ok335x:~#
  
```

- 12) 输入命令 ping 任意广域网域名或 IP, 若 PING 通则 WIFI 网卡已正常工作。

```
root@ok335x:~# ping baidu.com
```

如下图:

```

root@ok335x:~# ping baidu.com
PING baidu.com (123.125.114.144): 56 data bytes
64 bytes from 123.125.114.144: seq=0 ttl=50 time=89.224 ms
64 bytes from 123.125.114.144: seq=1 ttl=50 time=93.783 ms
64 bytes from 123.125.114.144: seq=2 ttl=50 time=93.713 ms
64 bytes from 123.125.114.144: seq=3 ttl=50 time=95.328 ms
64 bytes from 123.125.114.144: seq=4 ttl=50 time=96.817 ms
64 bytes from 123.125.114.144: seq=5 ttl=50 time=121.683 ms
64 bytes from 123.125.114.144: seq=6 ttl=50 time=86.549 ms
64 bytes from 123.125.114.144: seq=7 ttl=50 time=99.827 ms
64 bytes from 123.125.114.144: seq=8 ttl=50 time=99.751 ms
64 bytes from 123.125.114.144: seq=9 ttl=50 time=104.681 ms
64 bytes from 123.125.114.144: seq=10 ttl=50 time=122.896 ms
64 bytes from 123.125.114.144: seq=11 ttl=50 time=161.115 ms
64 bytes from 123.125.114.144: seq=12 ttl=50 time=152.707 ms
  
```

## 2.3.19 USB 摄像头测试

注意:

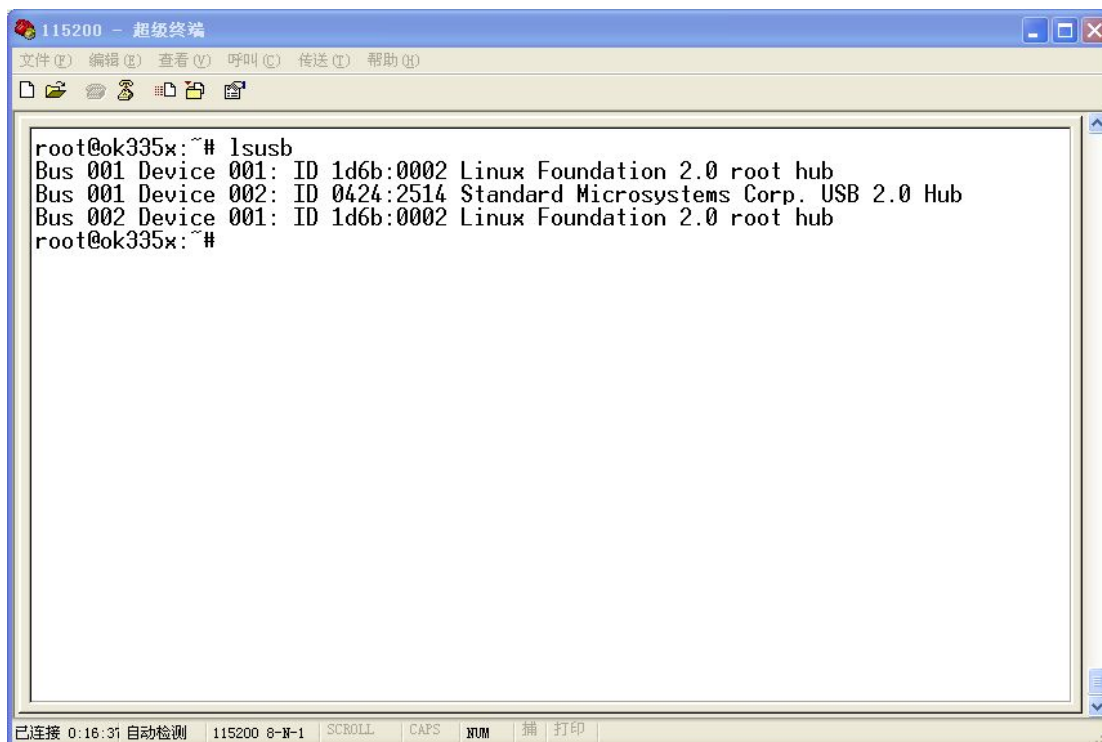
本产品目前支持以下 USB 摄像头:

**Webcam C270**

1) 开发板上电, 打开超级终端。输入命令查看 USB 状态。

```
root@ok335x:~# lsusb
```

如下图:



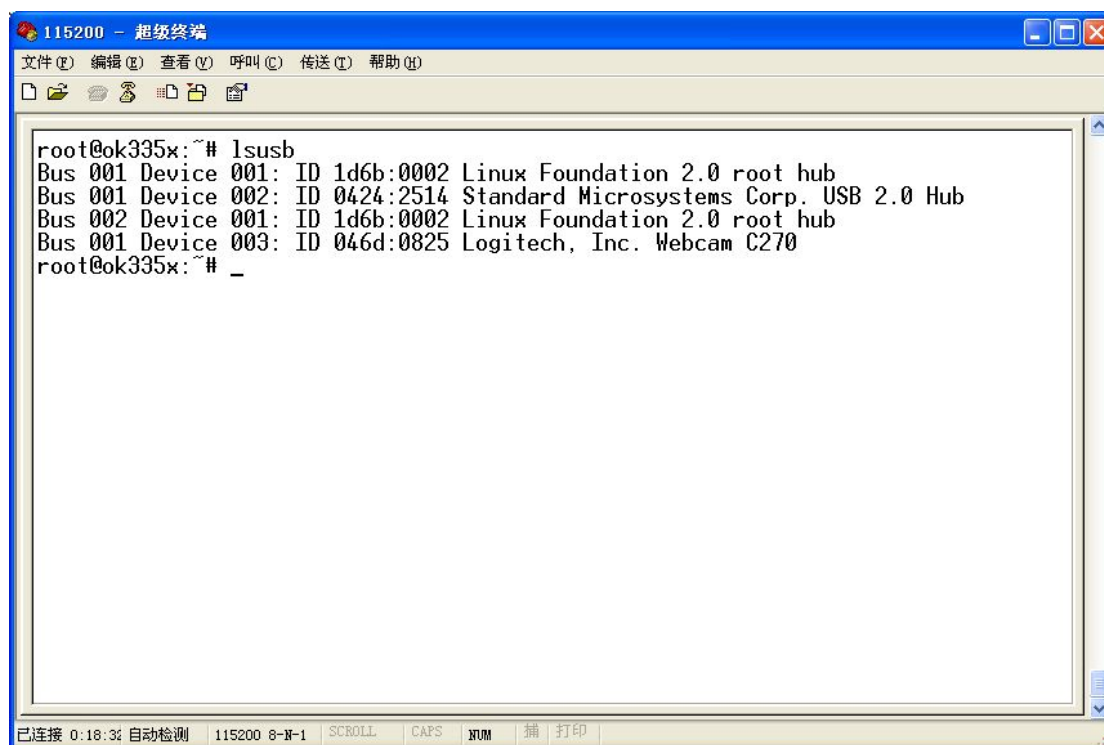
```

root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@ok335x:~#
    
```

2) 插入上述指定的 USB 摄像头，再次输入命令查看 USB 状态，可以看到插入的 USB 摄像头信息。

```
root@ok335x:~# lsusb
```

如下图：



```

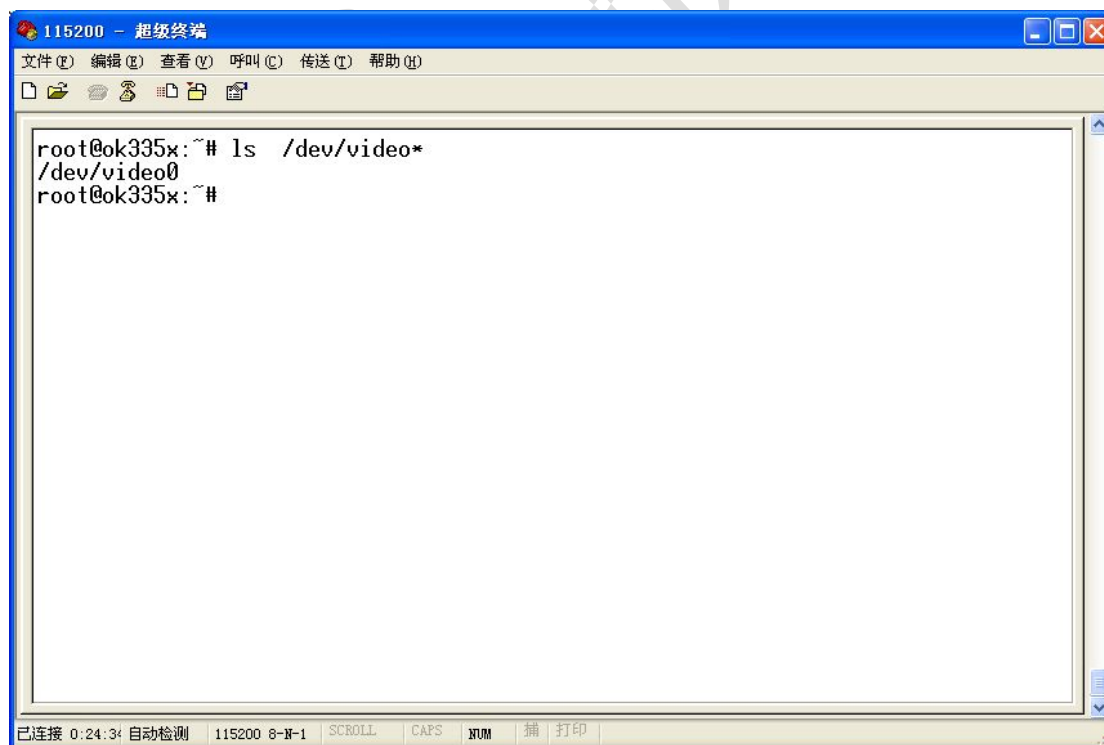
root@ok335x:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 046d:0825 Logitech, Inc. Webcam C270
root@ok335x:~# _

```

3) 输入命令查看 USB 摄像头的设备节点，可见设备节点 video0。

```
root@ok335x:~# ls /dev/video*
```

如下图：



```

root@ok335x:~# ls /dev/video*
/dev/video0
root@ok335x:~#

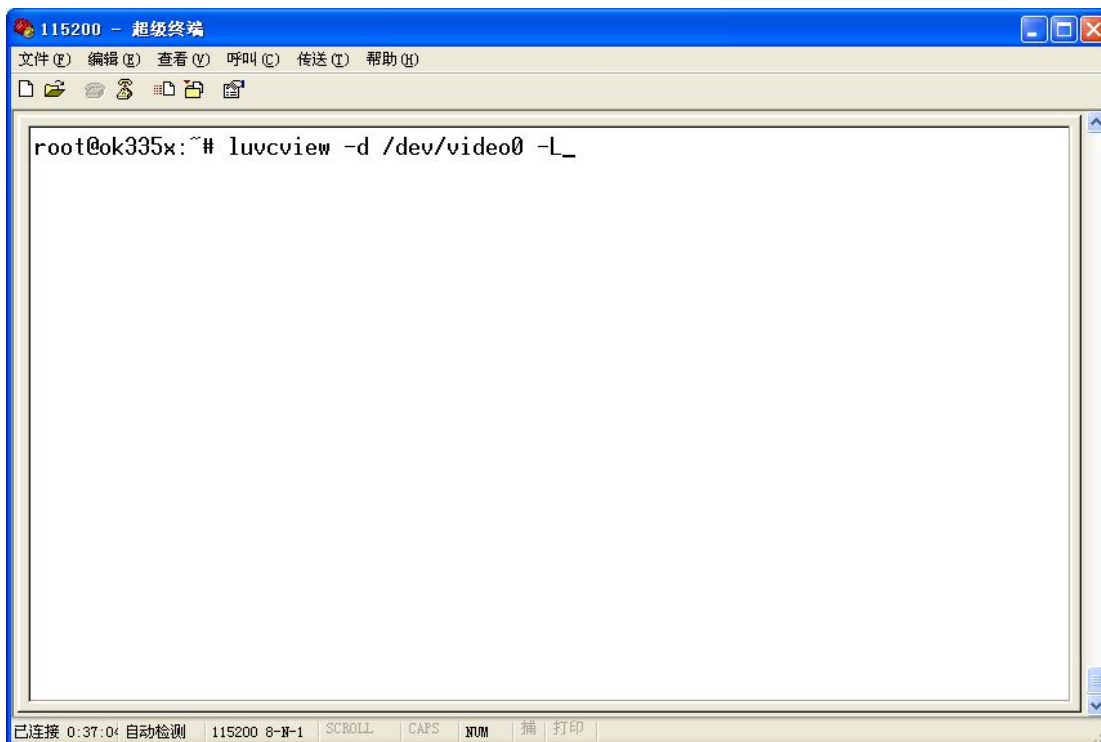
```



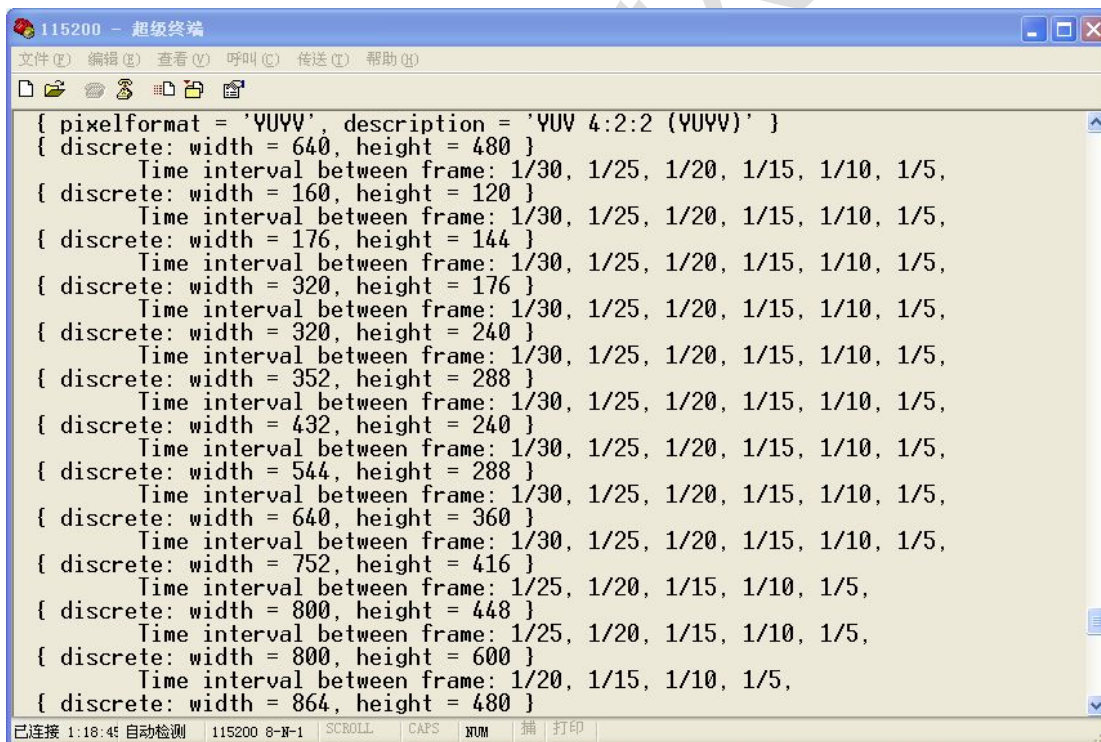
4) 输入命令查看摄像头支持的分辨率和帧速率。

```
root@ok335x:~# luvview -d /dev/video0 -L
```

如下图:

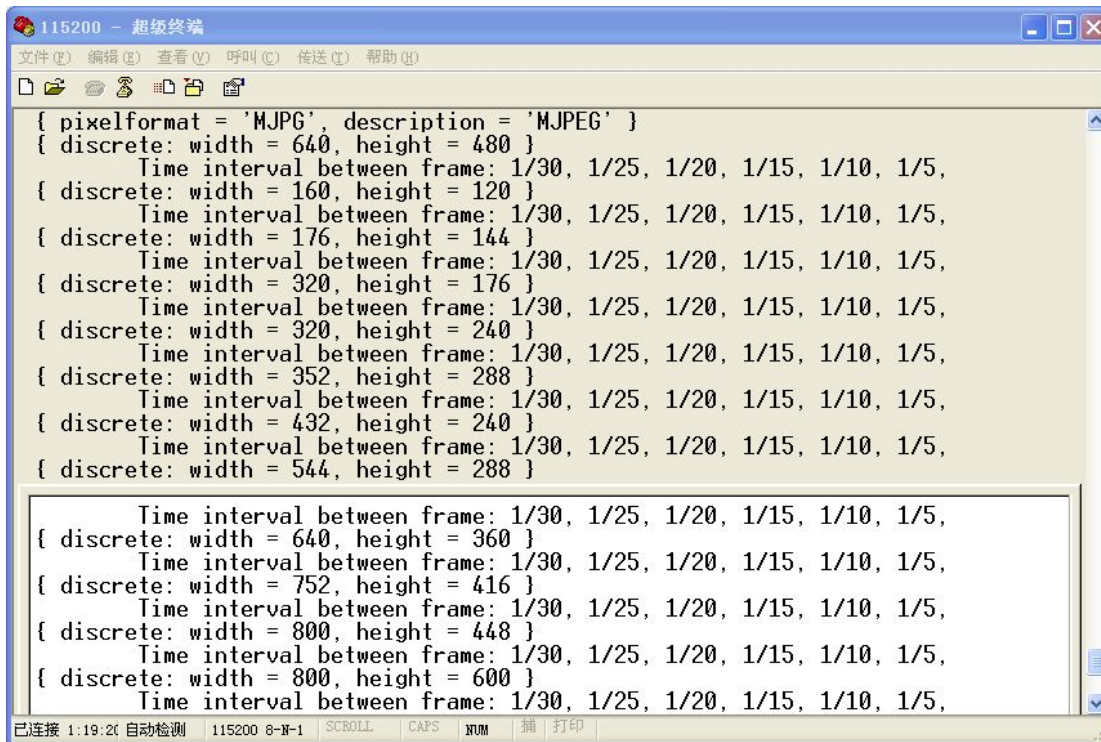


YUV 模式支持的分辨率和帧速率，如下图:





MJPEG 模式支持的分辨率和帧速率，如下图：



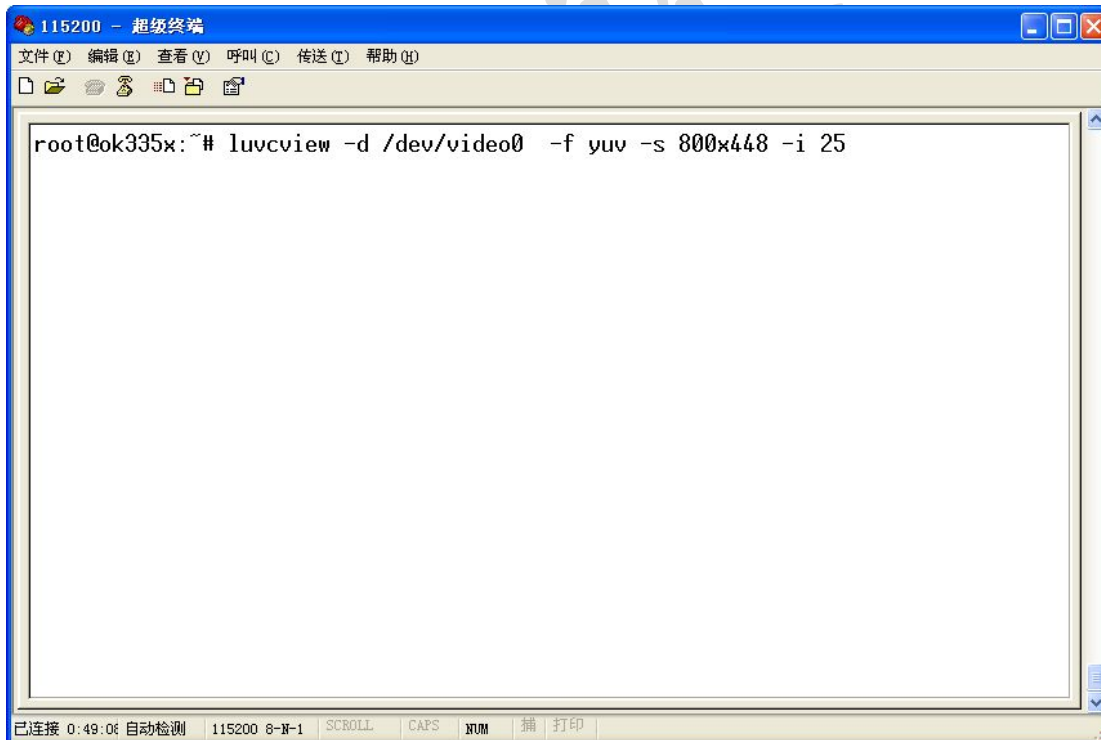
```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
{ pixelformat = 'MJPG', description = 'MJPEG' }
{ discrete: width = 640, height = 480 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 160, height = 120 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 176, height = 144 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 320, height = 176 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 320, height = 240 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 352, height = 288 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 432, height = 240 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 544, height = 288 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 640, height = 360 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 752, height = 416 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 800, height = 448 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
{ discrete: width = 800, height = 600 }
    Time interval between frame: 1/30, 1/25, 1/20, 1/15, 1/10, 1/5,
已连接 1:19:20 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
    
```

5) 输入命令进行 YUV 模式图像采集，可在液晶屏上预览采集的图像。

```
root@ok335x:~# luvcview -d /dev/video0 -f yuv -s 800x448 -i 25
```

如下图：



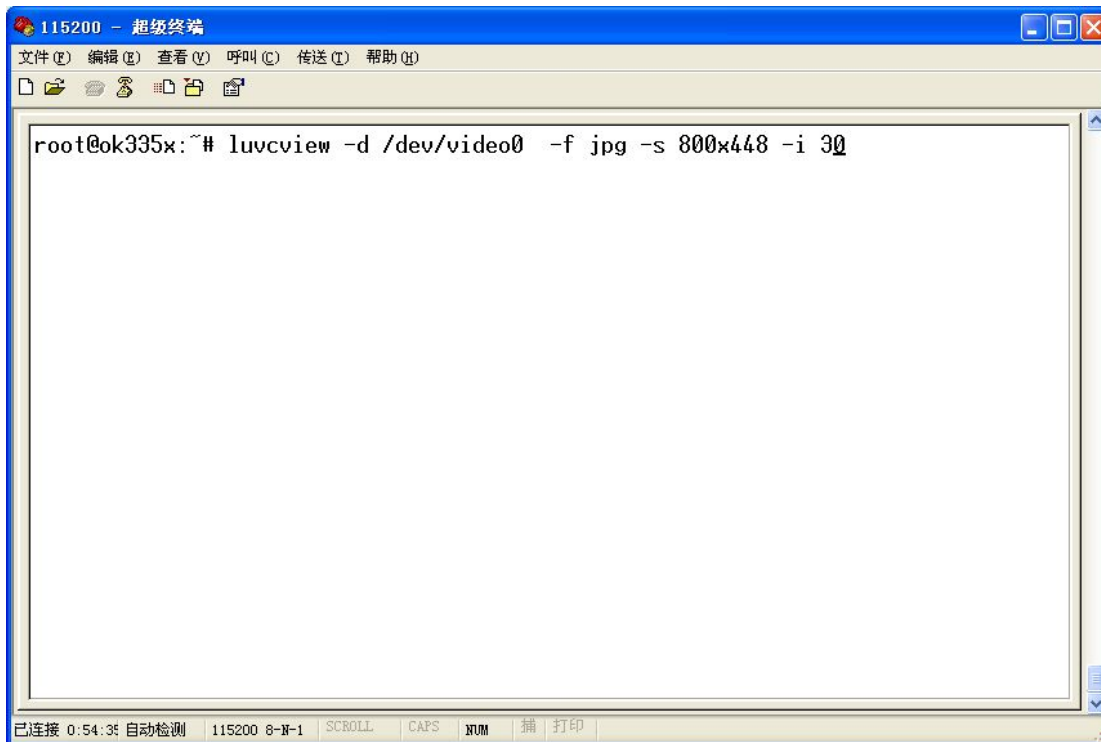
```

115200 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
root@ok335x:~# luvcview -d /dev/video0 -f yuv -s 800x448 -i 25
已连接 0:49:00 自动检测 115200 8-N-1 SCROLL CAPS NUM 插 打印
    
```

6) 输入命令进行 MJPEG 模式图像采集，可在液晶屏上预览采集的图像。

```
root@ok335x:~# luvview -d /dev/video0 -f jpg -s 800x448 -i 30
```

如下图：



## 3.4 图形界面测试

以下图形测试需要使用带 QT 版本的文件系统（rootfs.tar.bz2）。

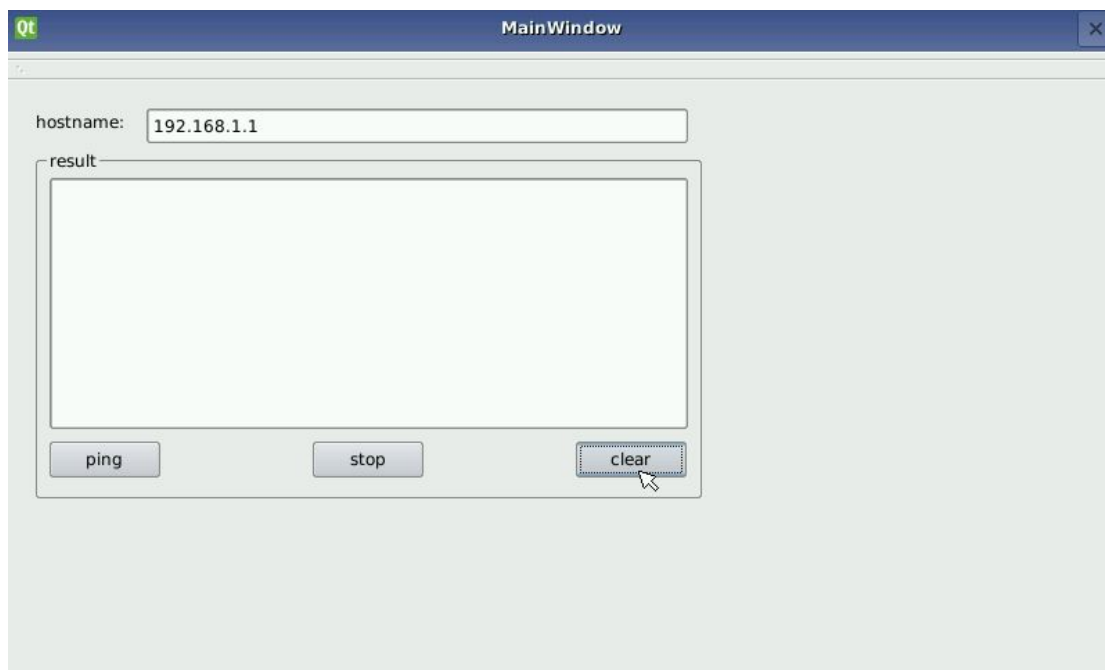
注：飞凌提供的图形系统默认支持触摸屏操作，用户如果想要使用 USB 鼠标可以点击屏幕上的 **touch2mice** 图标，系统会自动切换的鼠标操作模式（此时插上 USB 鼠标即可移动光标），用户需要再次点击该图标来完成鼠标到触摸的切换。

### 3.4.1 LCD 及触摸测试

系统启动后查看 LCD 上面是否有如下的图像输出：



如果有则说明 LCD 显示正常，点击 Ping 图标，如果显示如下界面，则说明触摸功能正常：

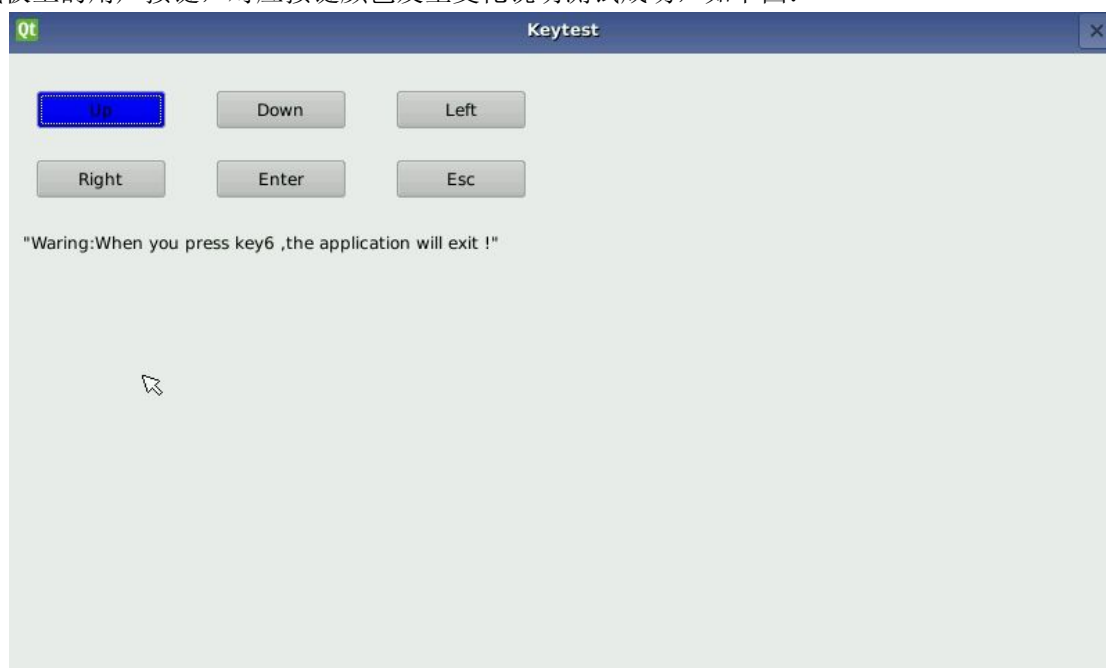


### 3.4.2 按键测试

在 LCD 主菜单界面点击 KeyPad 进入如下界面：



点击底板上的用户按键，对应按键颜色发生变化说明测试成功，如下图：



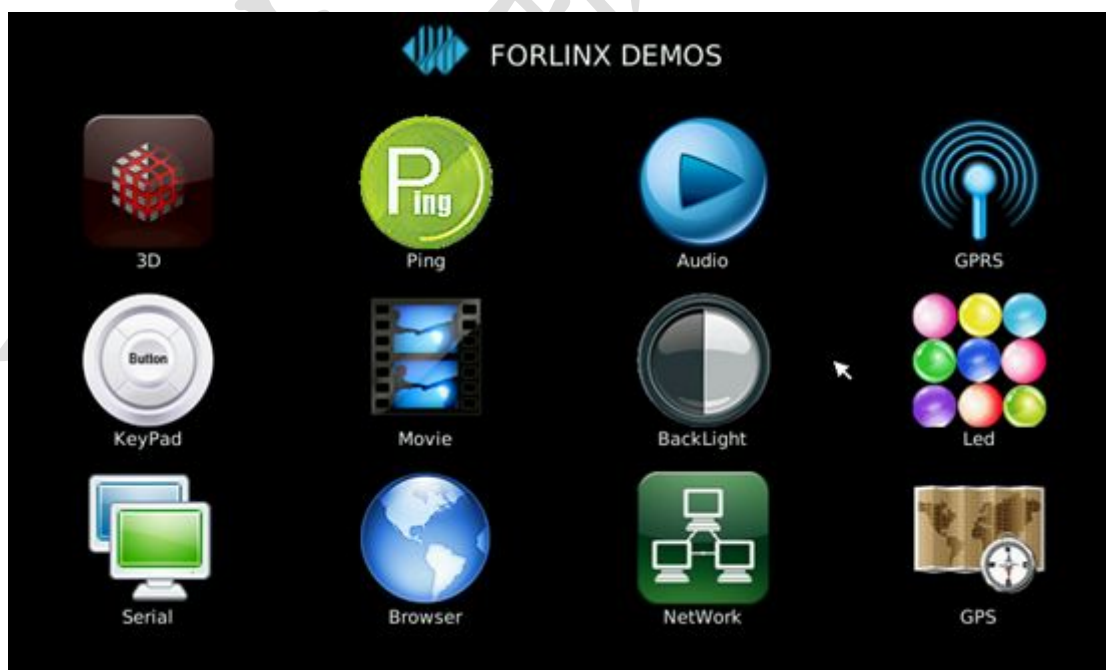
### 3.4.3 USB 测试

**注：该测试需要先将图形系统从触摸切换到鼠标模式**

把 USB 鼠标插入 USB 口，屏幕光标将可以移动，如下图（移动前）：



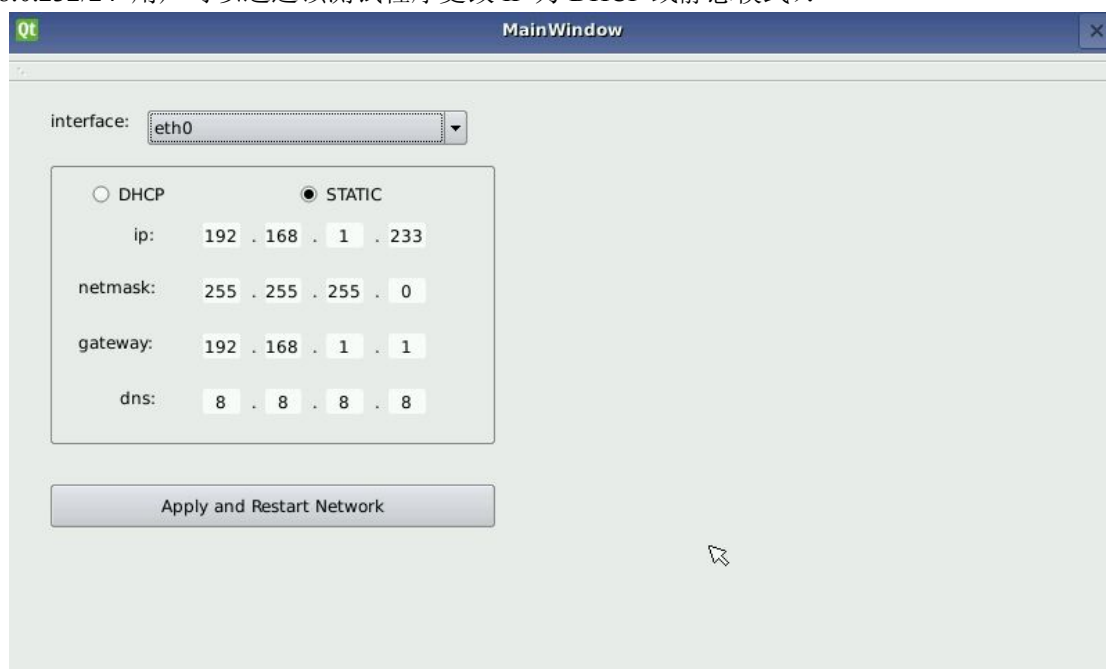
如果光标可以跟随鼠标进行移动，则说明 USB 工作正常，如下图（移动后）：



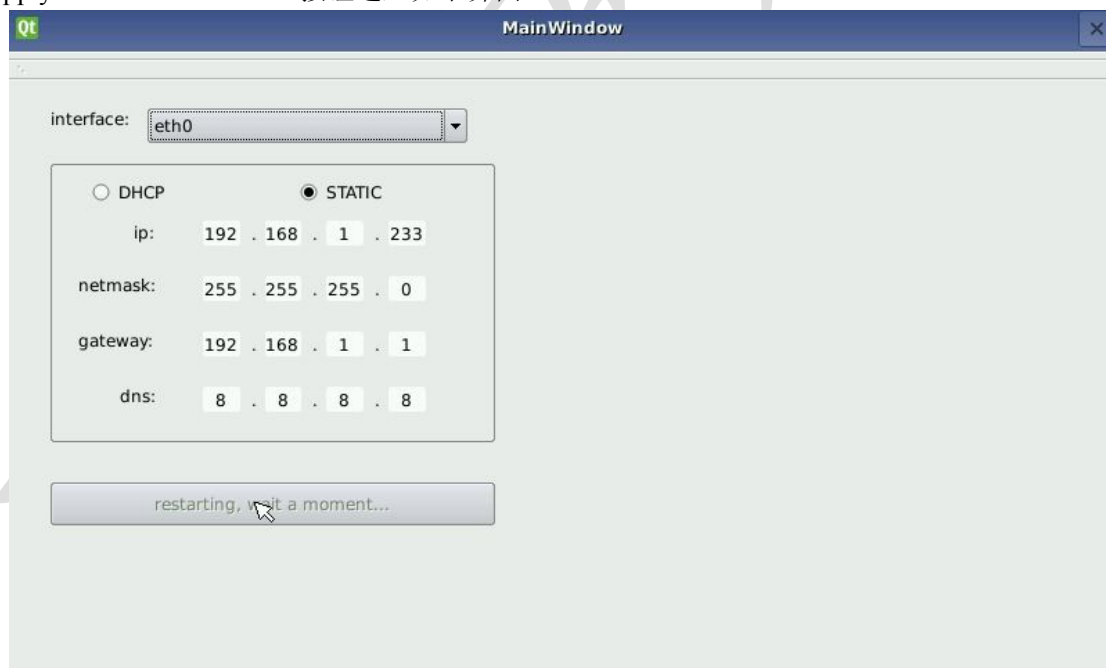


### 3.4.4 网卡配置

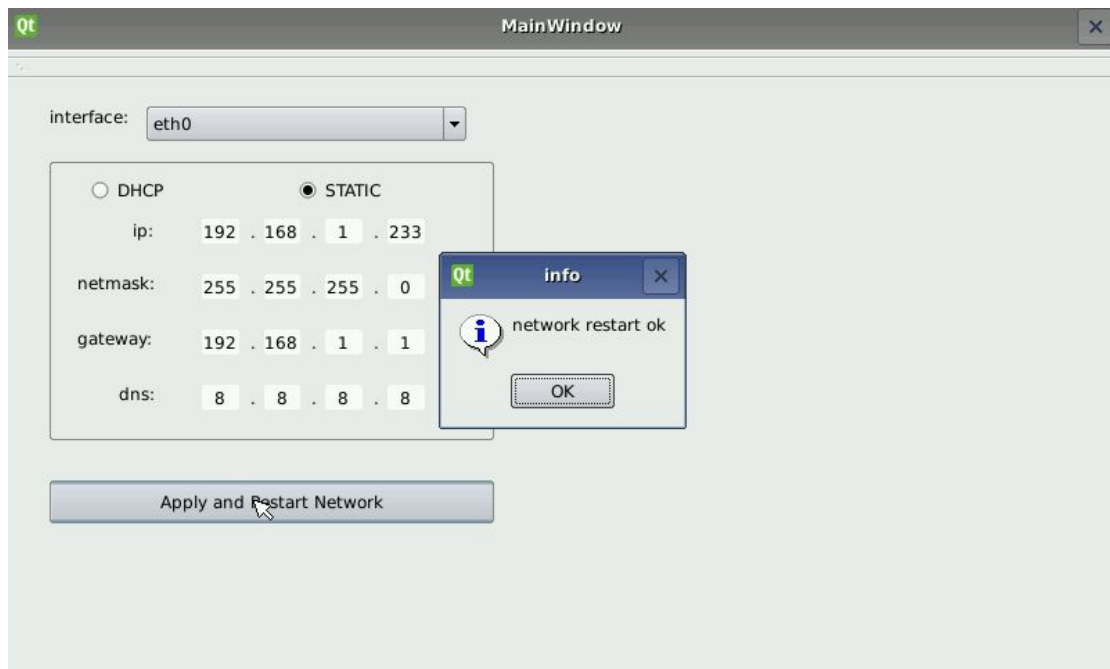
从 LCD 主菜单界面点击 Network 进入如下界面，选择对应网络接口并配置网络参数（开发板默认 IP 为 192.168.0.232/24 用户可以通过该测试程序更改 IP 为 DHCP 或静态模式）：



单击 Apply and Restart Network 按钮进入如下界面：



弹出如下界面后说明网络配置成功：

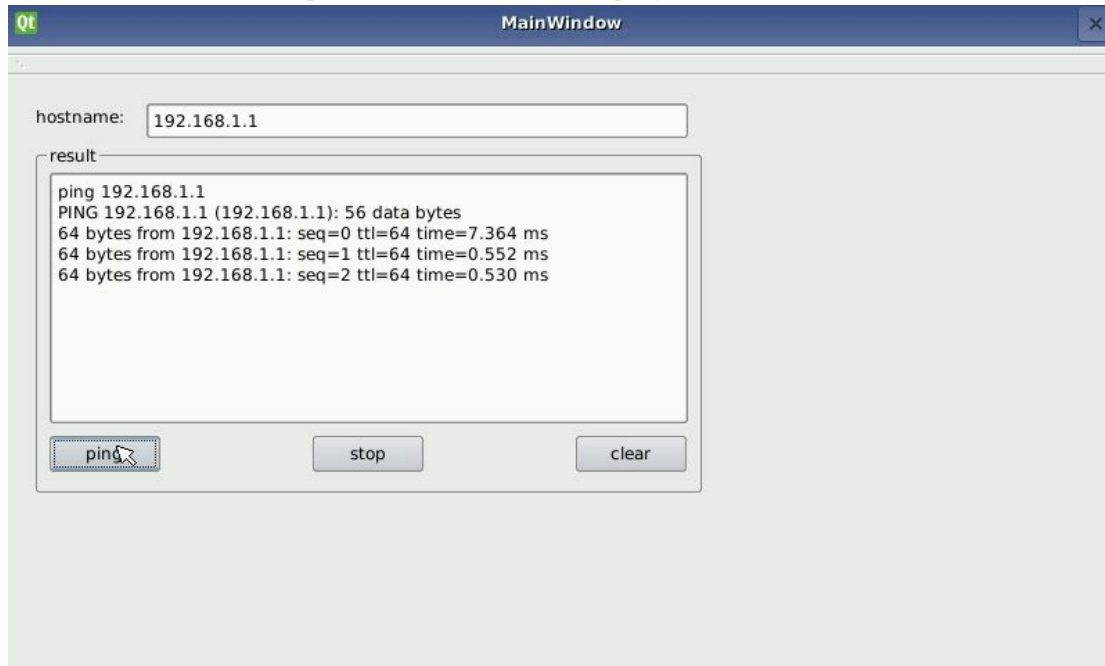


### 3.4.5 PING 测试

从 LCD 主菜单并单击 Ping 图标进入如下界面（确保已进行了网络配置）：



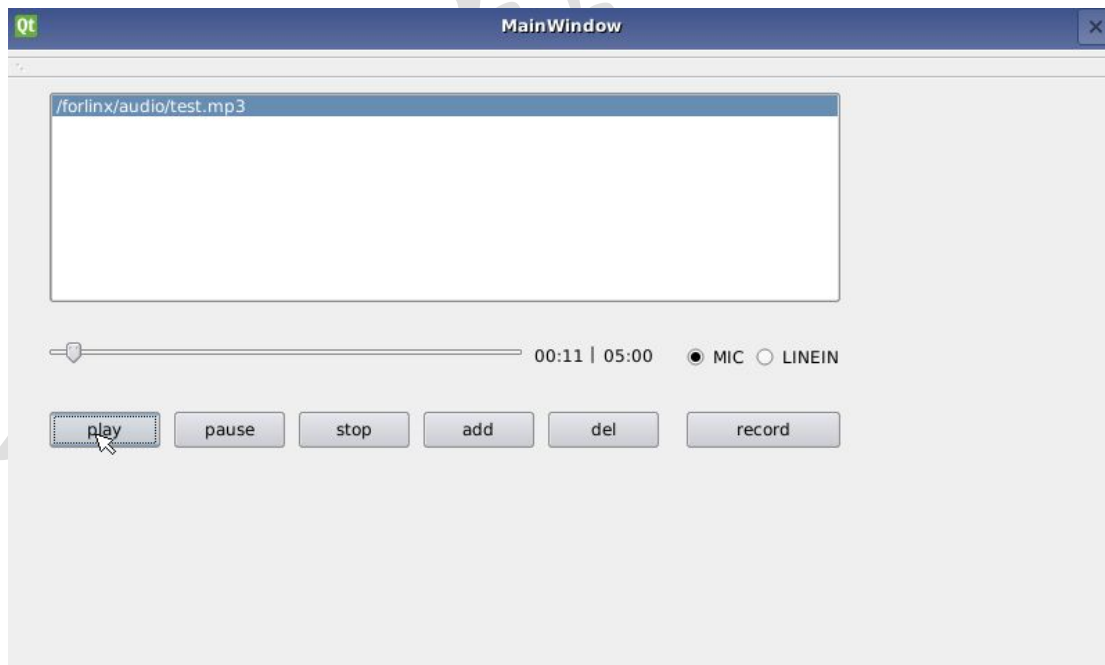
在 hostname 一栏输入要测试的 ip 地址或域名，然后单击 ping 按钮，若出现如下界面说明网络测试成功：



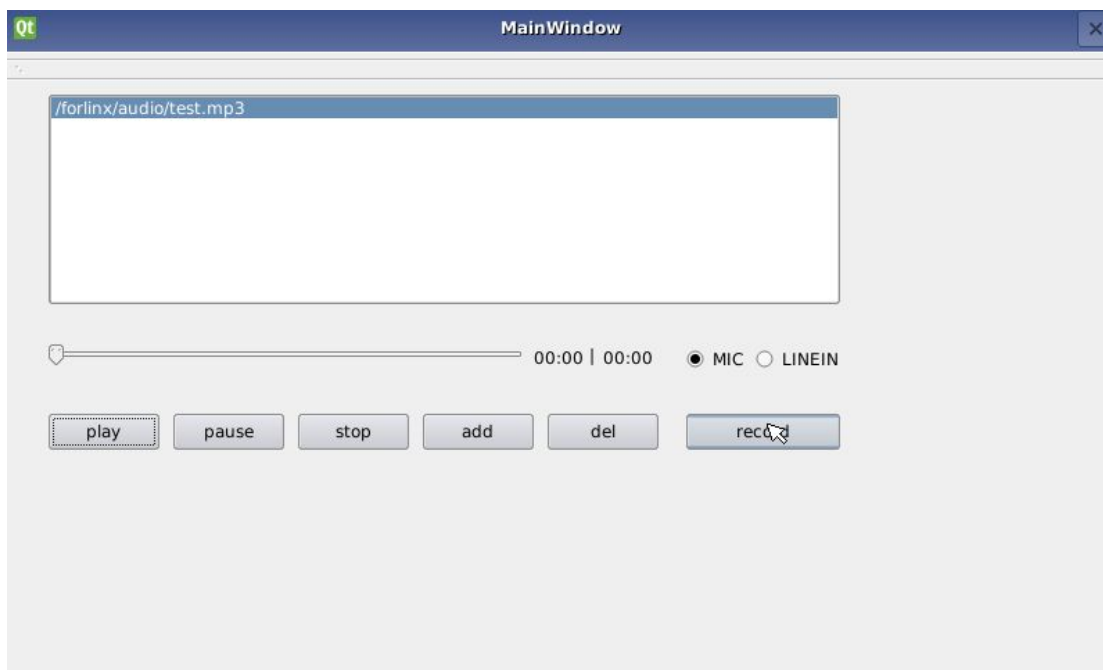
### 3.4.6 音频测试

从 LCD 主菜单界面点击 Audio 图标进入如下界面：

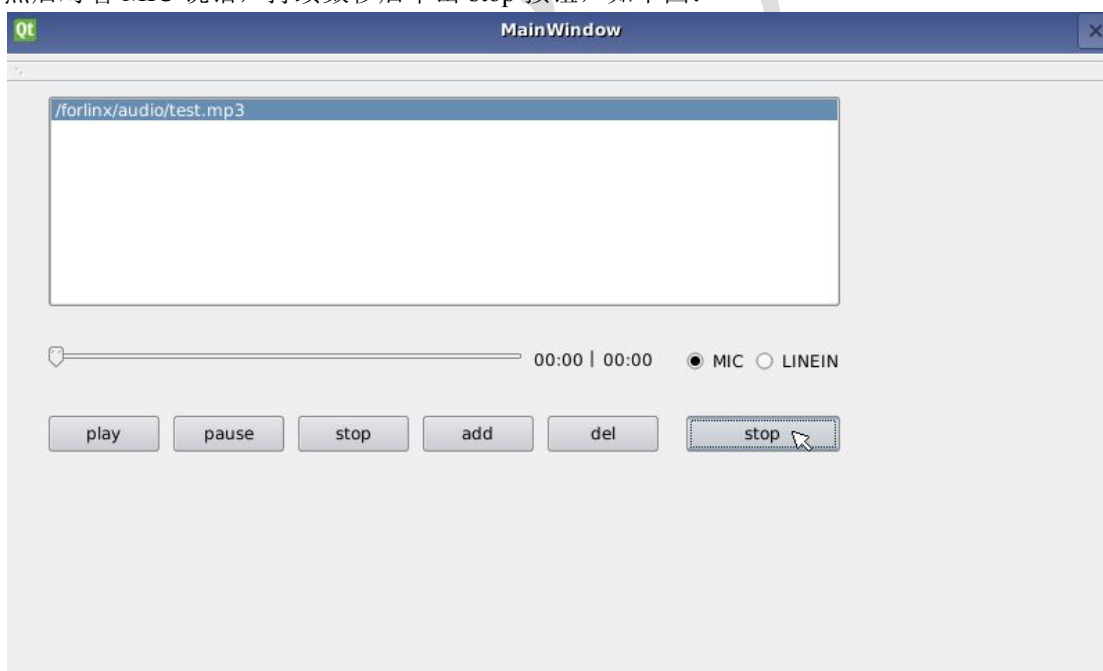
1. 放音测试，将耳机插入开发板的 PHONE 插孔，单击 play 按钮，若能听到音乐则说明放音正常：



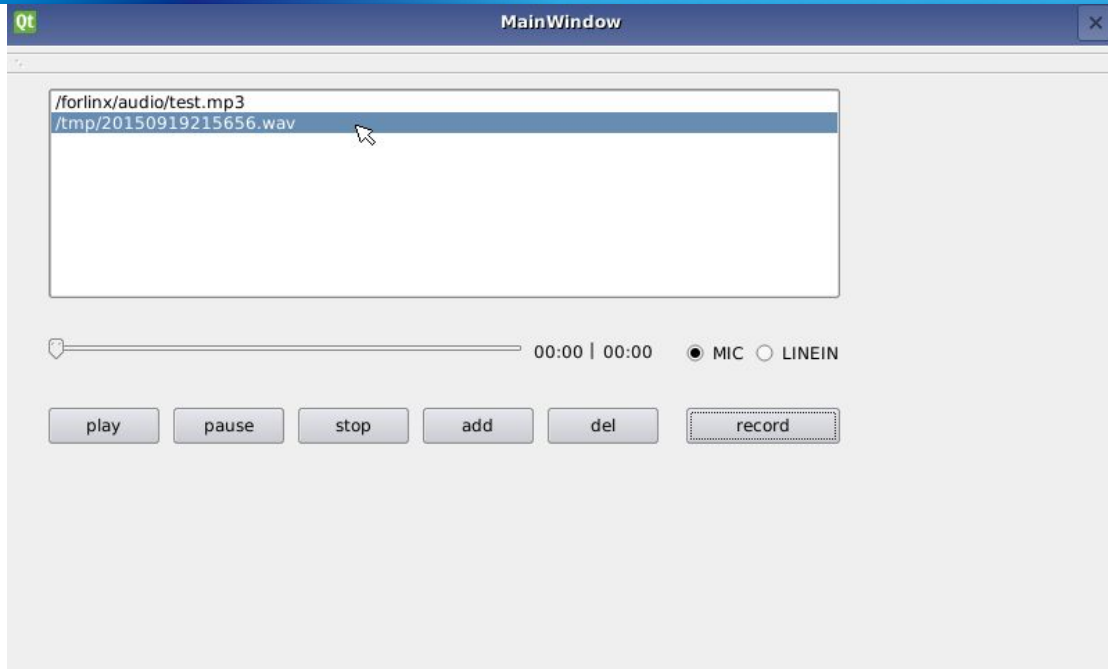
2. MIC 录音测试，将 MIC 插入开发板的 MIC 插孔，单击 record 按钮：



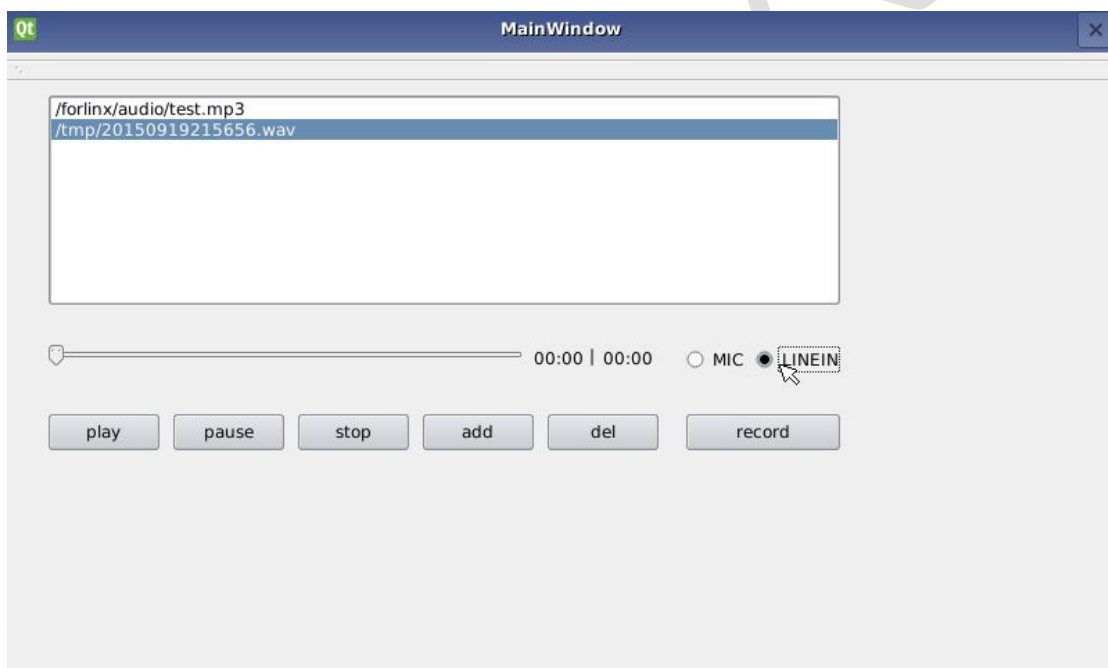
然后对着 MIC 说话，持续数秒后单击 stop 按钮，如下图：



之后会看到歌曲列表里会多出一个临时文件，单击 play 按钮可以回放录音，如下图：



3. LINEIN 录音测试，将 LINEIN 插入开发板的 LINEIN 插孔，选中 LINEIN 选钮，录音过程同 MIC：



**注：进行 Line In 录音时请将音源设备的输出音量（增益）调小，以防输入波形超出采集范围而失真**



### 3.4.7 视频测试

从 LCD 主菜单界面点击 Movie 图标，如果能播放如下视频，则说明视频播放正常：



### 3.4.8 LED 测试

从 LCD 主菜单界面点击 LED 进入如下界面，此时可看到用户 LED 灯是灭的：

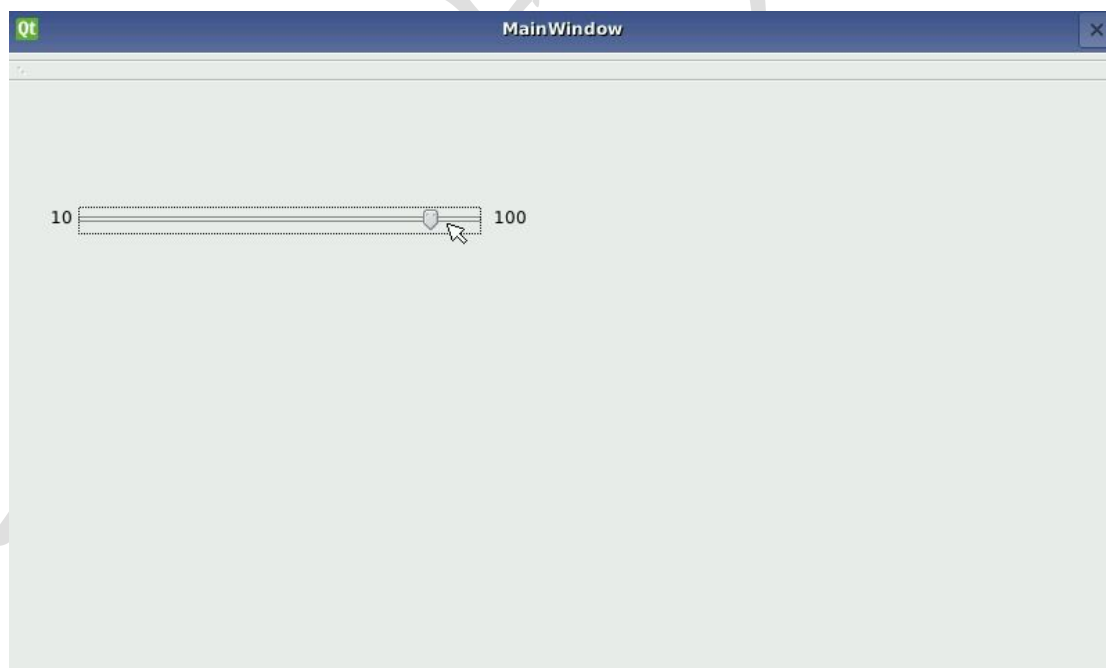


点击 LED1 按钮后，出现如下界面，此时若可看到用户 LED1 是亮的，说明测试成功：



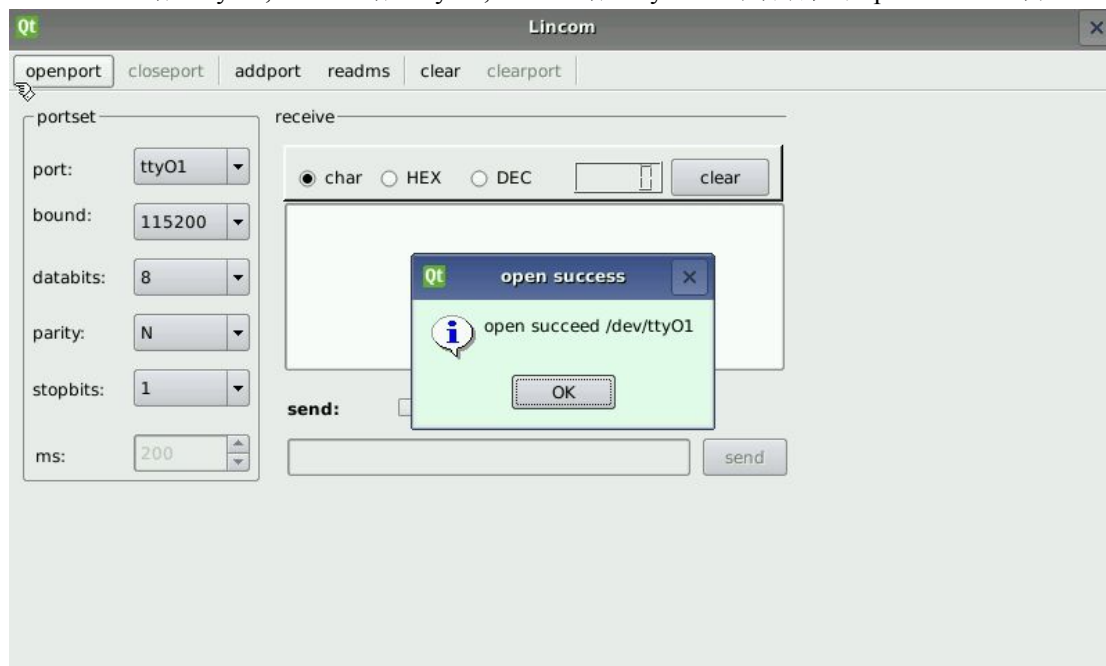
### 3.4.9 背光测试

从LCD主菜单界面点击 BackLight 图标进入如下界面，用户拖动滑条后若背光亮度发生变化说明测试成功。

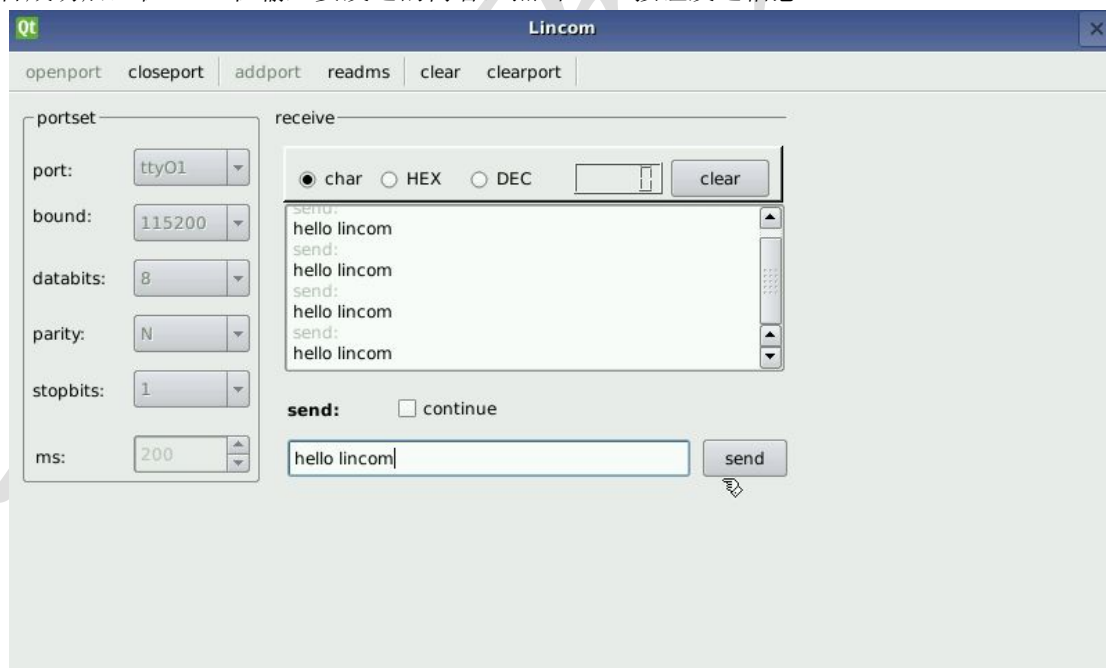


### 3.4.10 串口测试

测试前请将串口连接到 PC 并设置连接串口的端口名称、波特率、数据位、奇偶校验、停止位、硬件流控，打开串口（COM0 对应 ttyO0,COM1 对应 ttyO1,UART 对应 ttyO4），则会弹出 open success 窗口



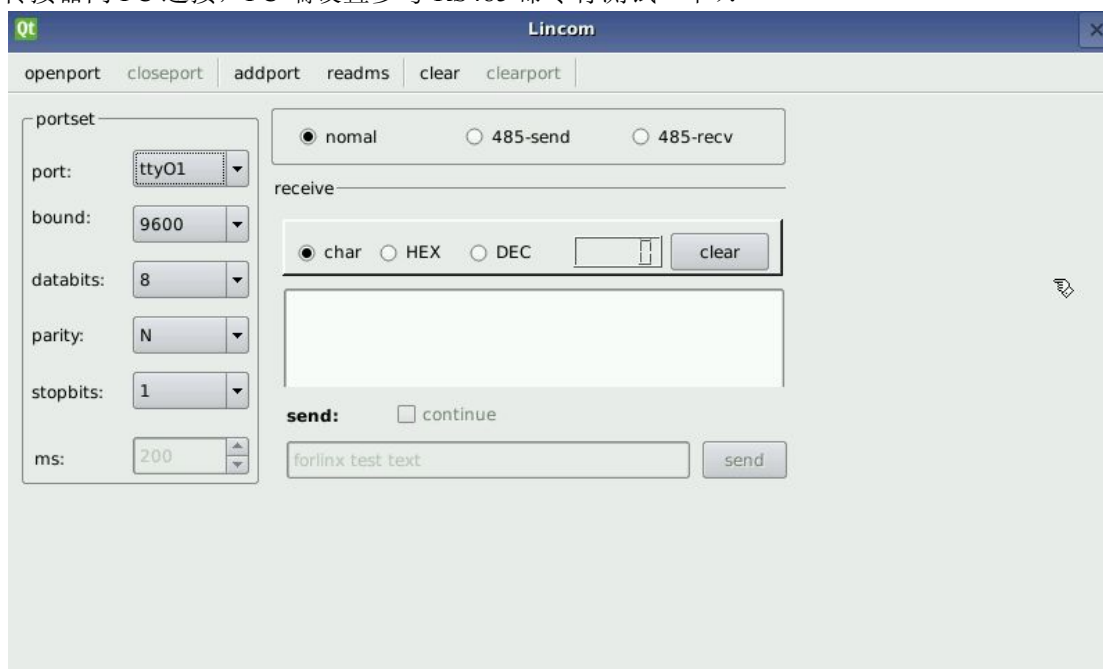
打开设备成功后，在 send 栏输入要发送的内容，点击 send 按钮发送信息。



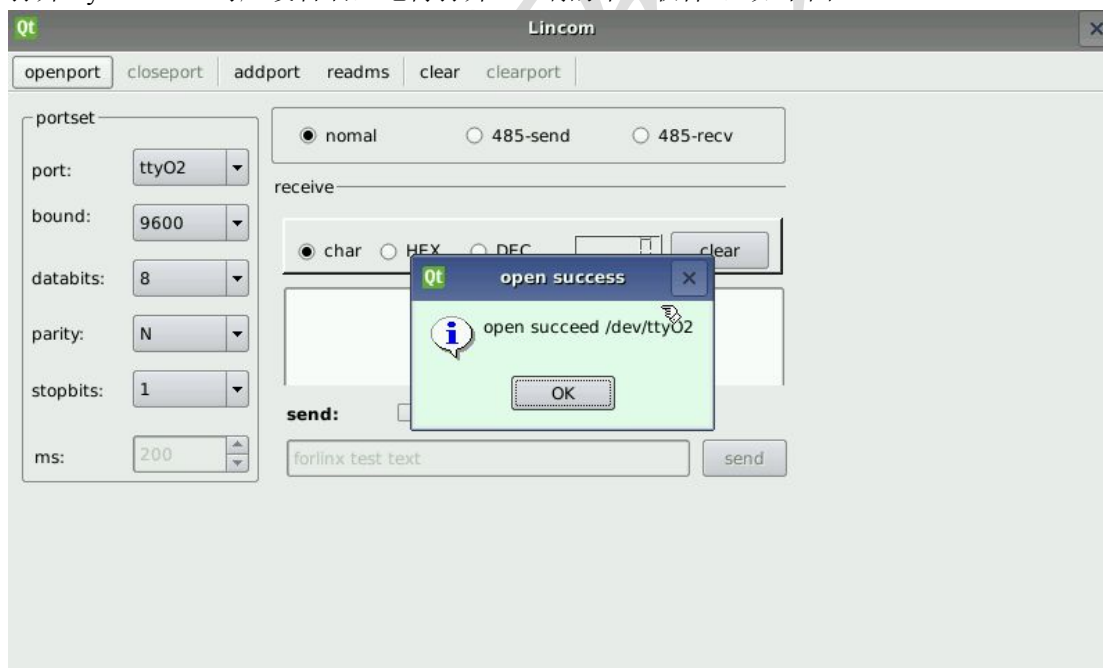
在 PC 端向开发板发送字符串可看到接收。

### 3.4.11 RS485 测试

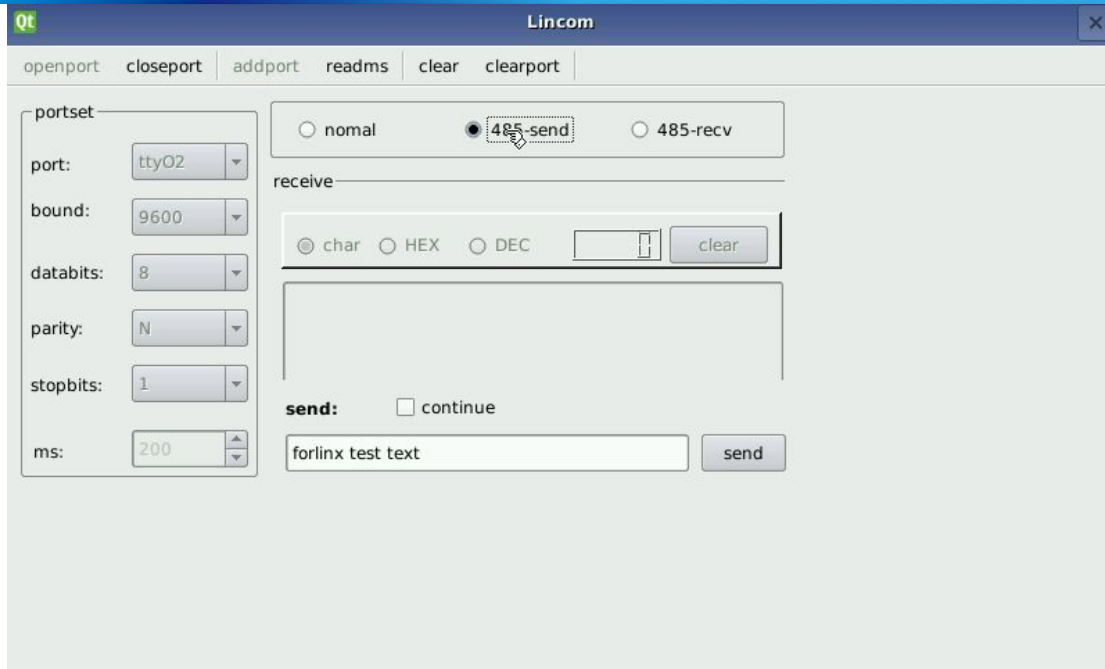
从 LCD 主菜单界面点击 Serial 图标进入如下界面（485 测试和串口测试使用同一个程序，本测试使用 485 转接器同 PC 连接，PC 端设置参考 RS485 命令行测试一节）：



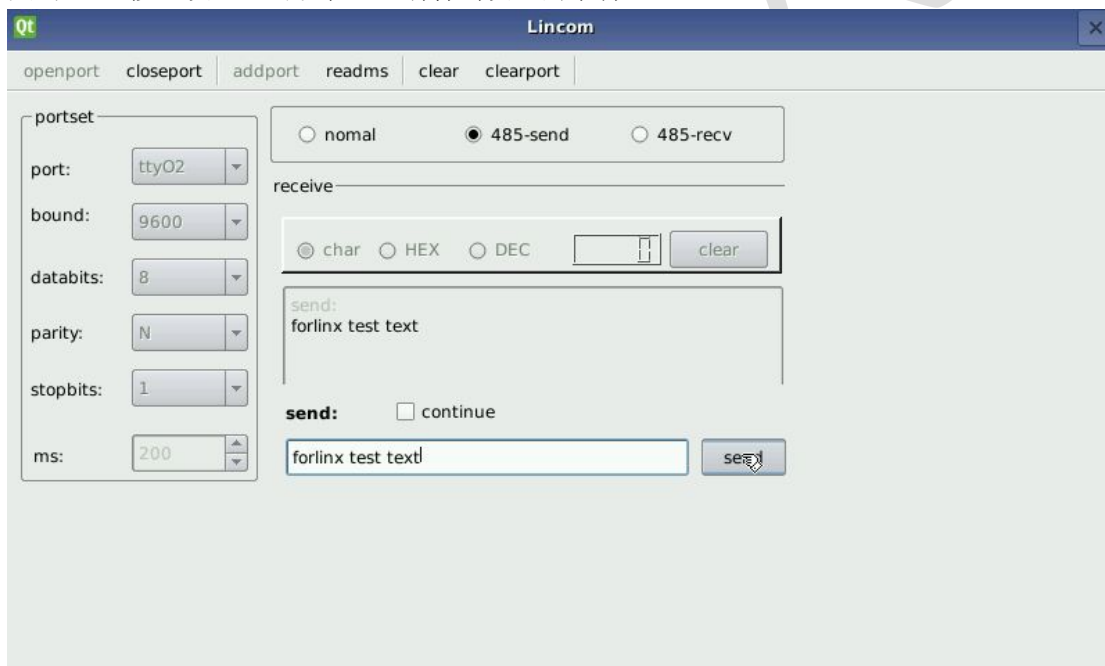
打开 ttyO2（485 对应设备名，记得打开 PC 端的串口软件），如下图：



将程序设置为 485-send 模式，如下图：

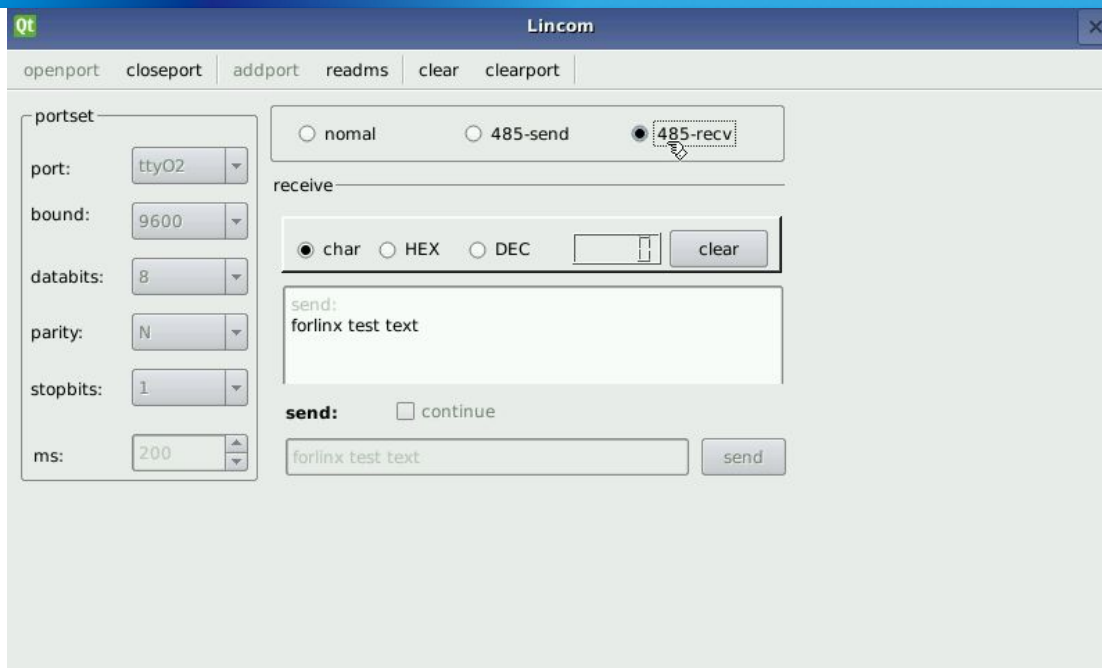


点击 send 按钮发送，可以在 PC 端看到发送的字符：

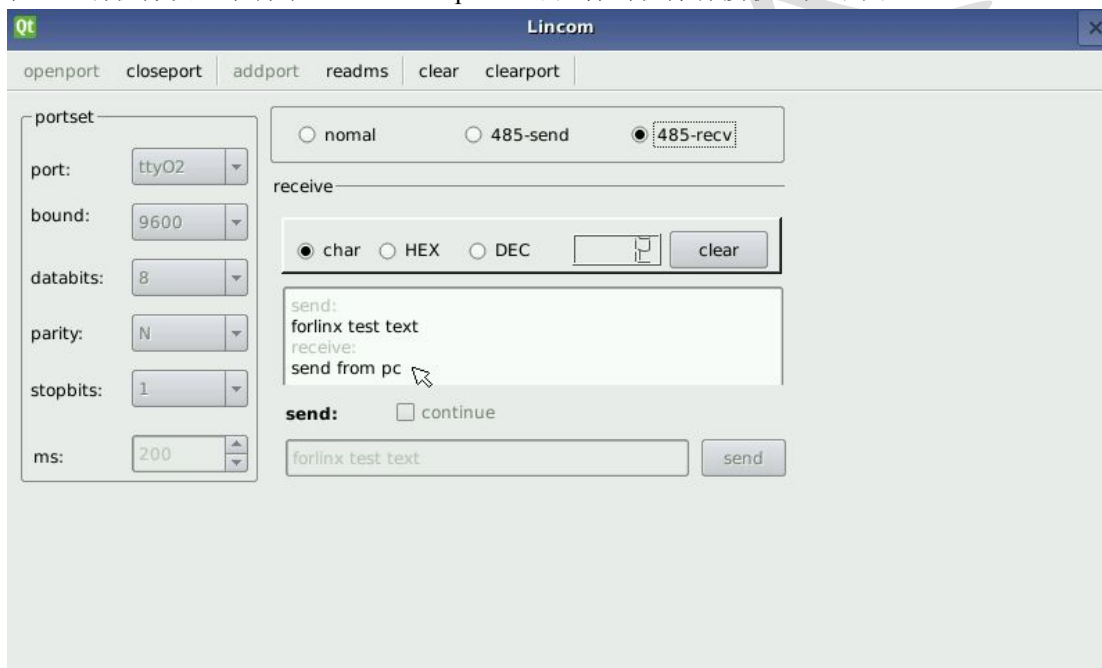


将程序设置为 485-recv 模式，如下图：





在 PC 端软件发送字符串“send from pc”，可以看到软件有接收，如下图：

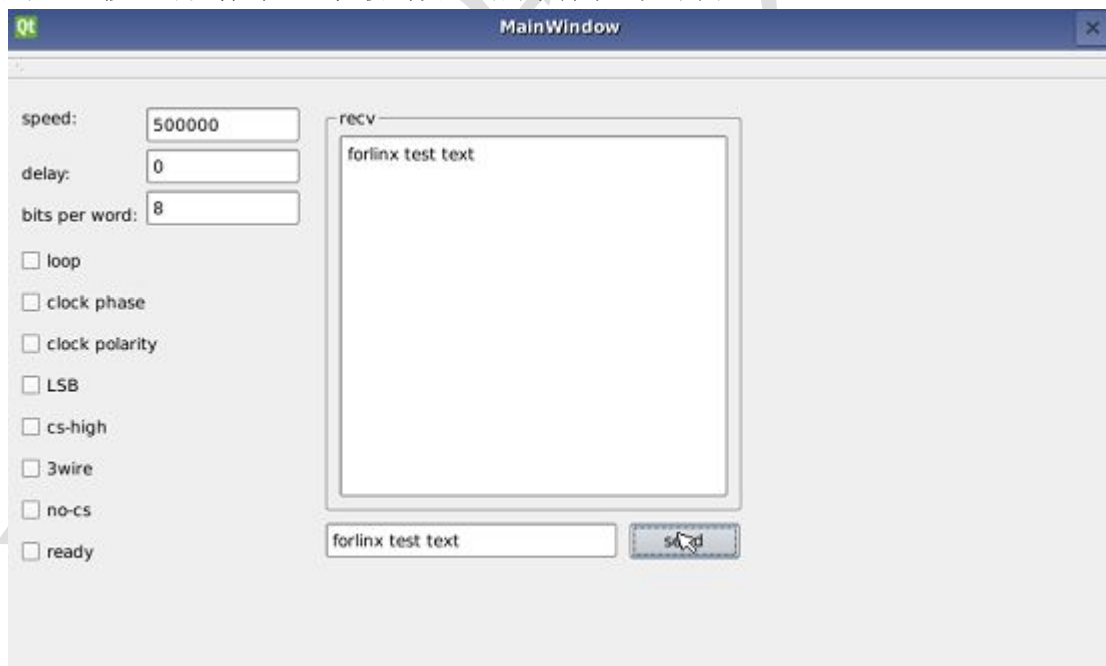


### 3.4.12 SPI 测试

从 LCD 主菜单界面点击 SPI 图标进入如下界面，将原理图中的 SPI1-D0 和 SPI1\_D1 引脚短接：



点击 send 按钮可以看到 recv 框收到发送出的字符串，如下图：



### 3.4.13 CAN 测试

从 LCD 主菜单界面点击 Can 图标进入如下界面（用户需要将两台开发板 Can 口相连，在另一台上同样运行此程序）：



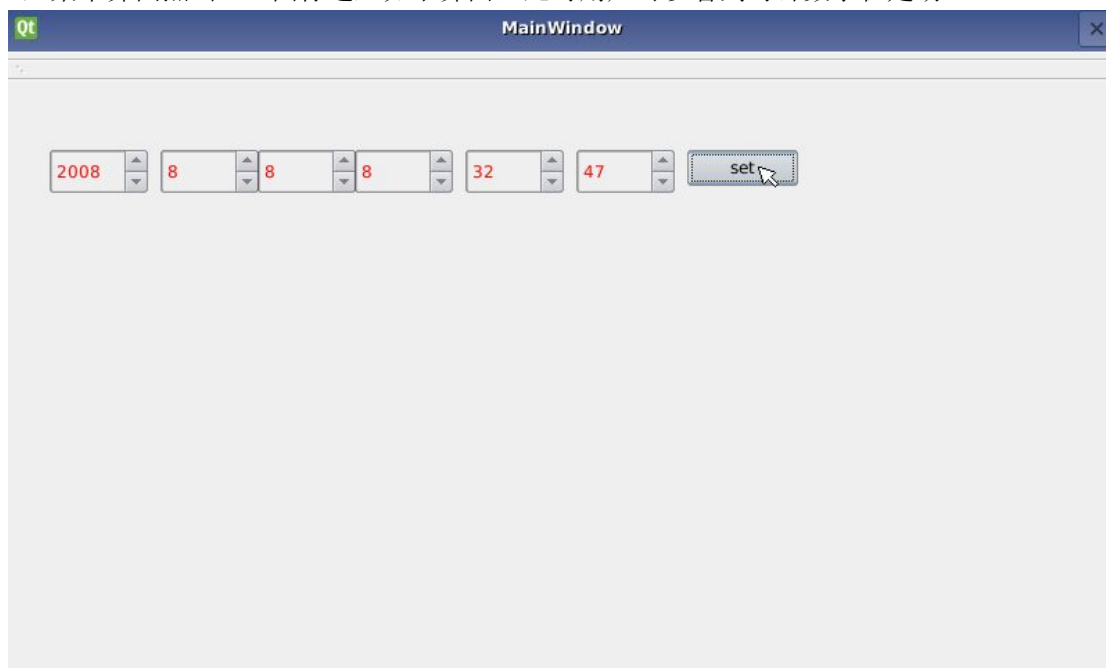
用户单击 send 按钮，在另一台可以看到有数据接收，如下图：



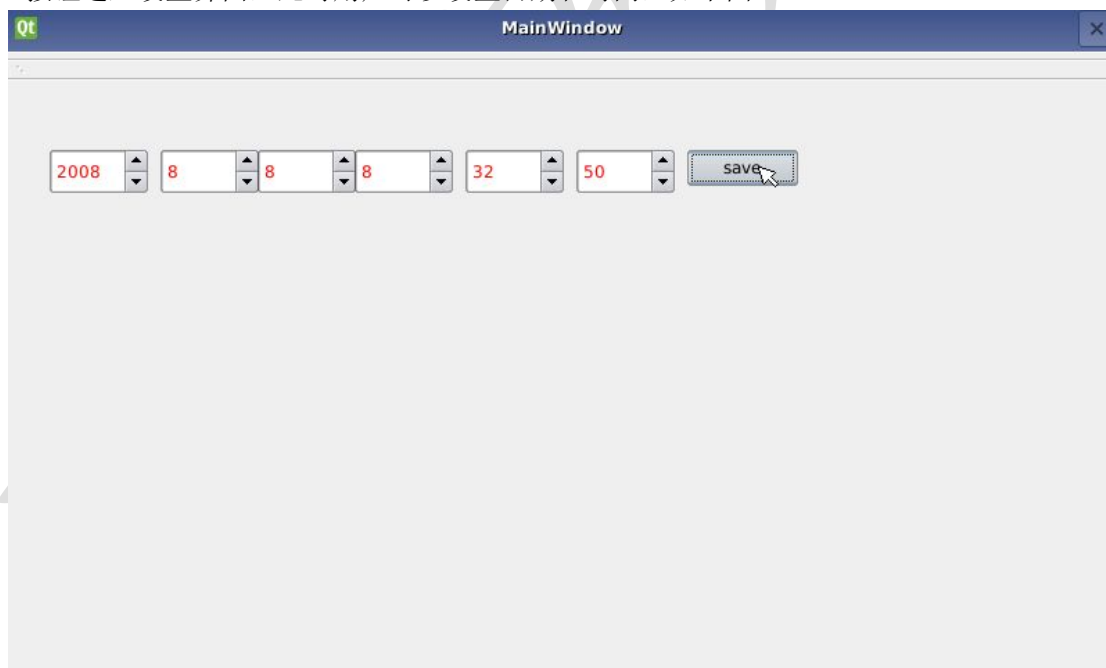
同理，在另一台上单击 send 按钮可以在测试板上看到有数据接收。

### 3.4.14 RTC 测试

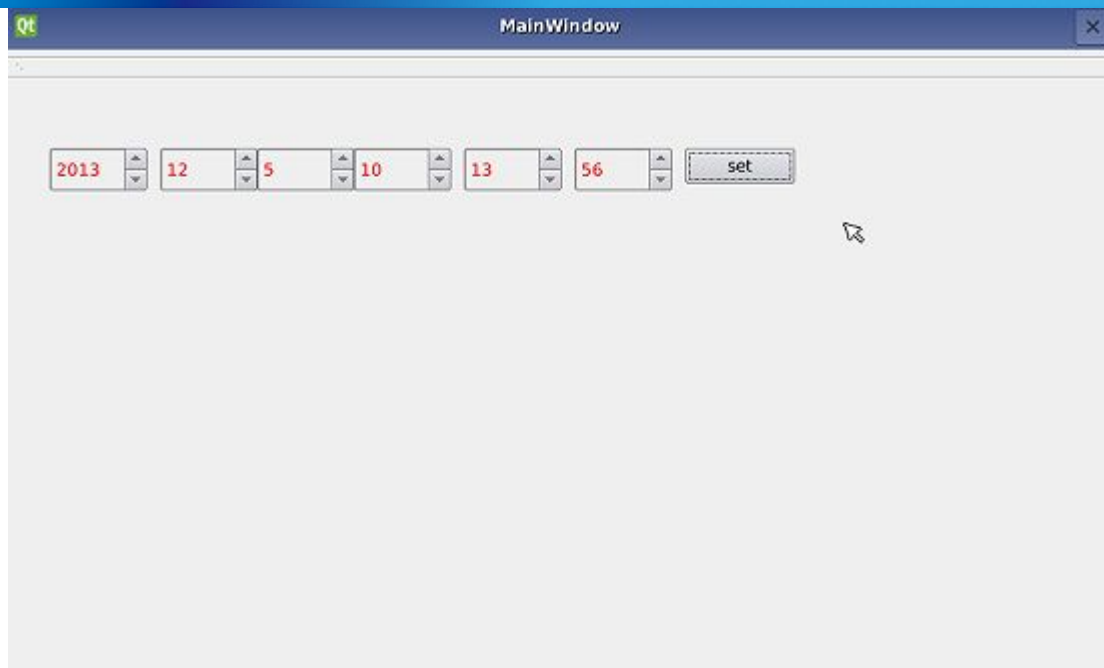
从 LCD 主菜单界面点击 Rtc 图标进入如下界面，此时用户可以看到时钟数字在走动：



点击 set 按钮进入设置界面，此时用户可以设置日期和时间，如下图：



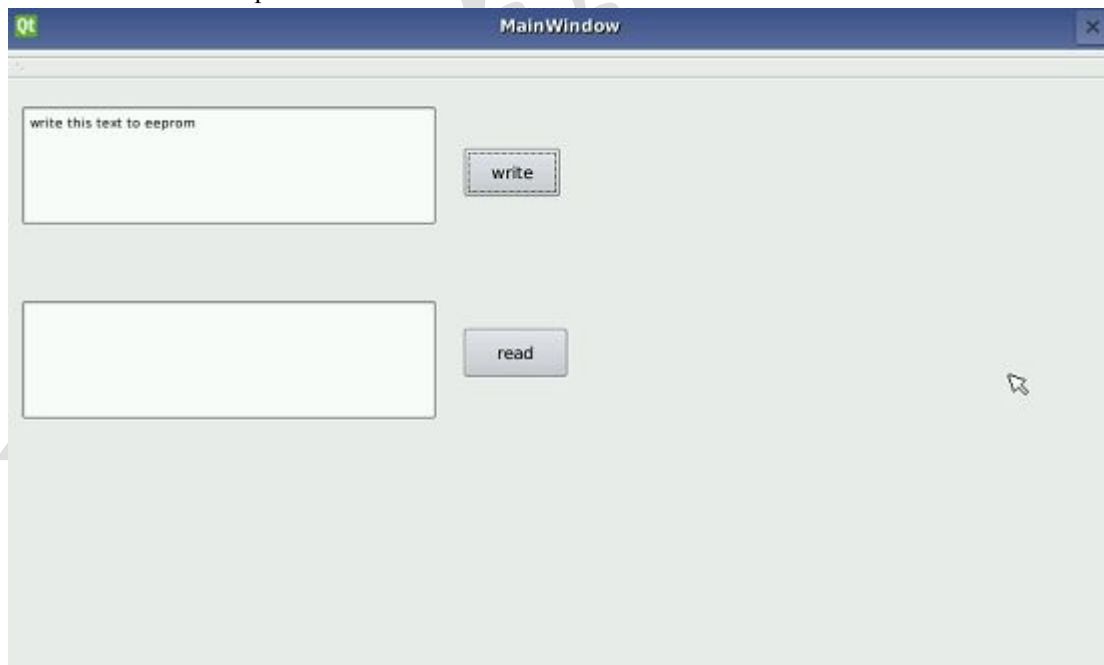
点击 save 按钮，保存日期和时间设置，如下图：



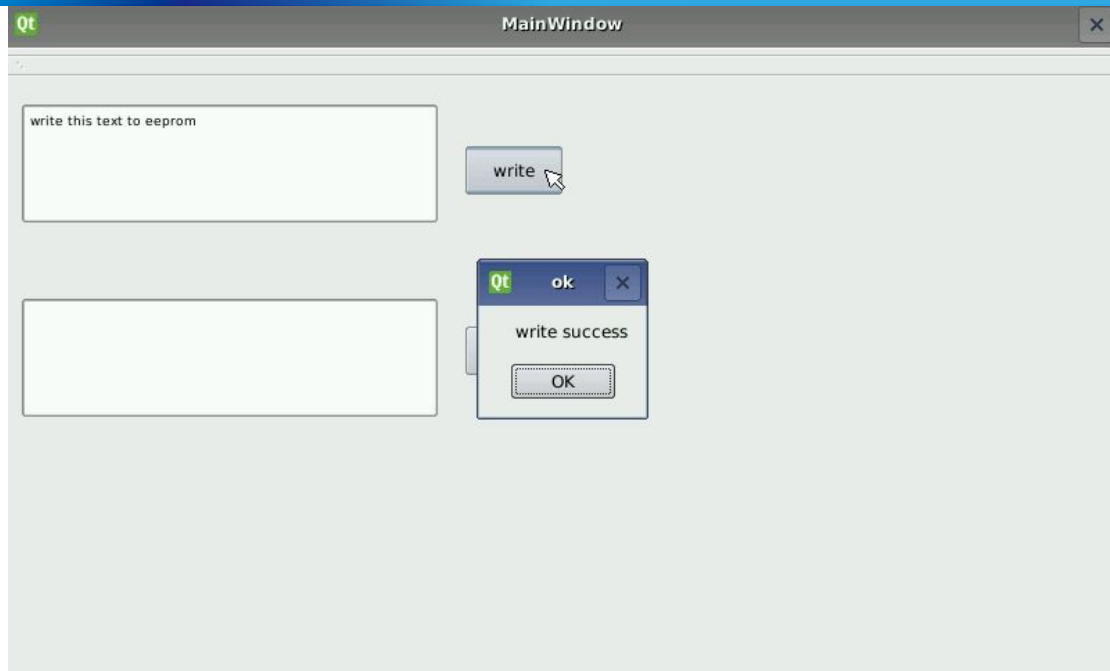
用户可以断电重启后，重新进入该程序查看刚刚设置的日期和时间是否保存成功，若成功则说明 rtc 时钟没有问题（**注意：确保板子上已经安装了纽扣电池**）。

### 3.4.15 EEPROM 测试

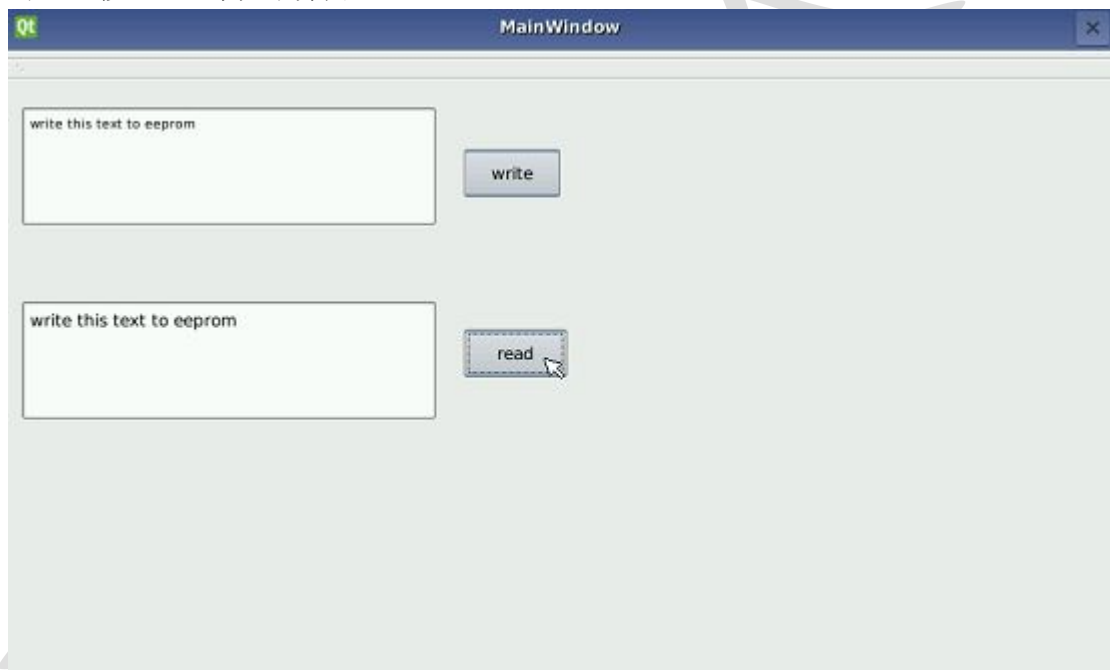
从 LCD 主菜单界面点击 Eeprom 图标进入如下界面：



单击 write 按钮，显示如下界面：



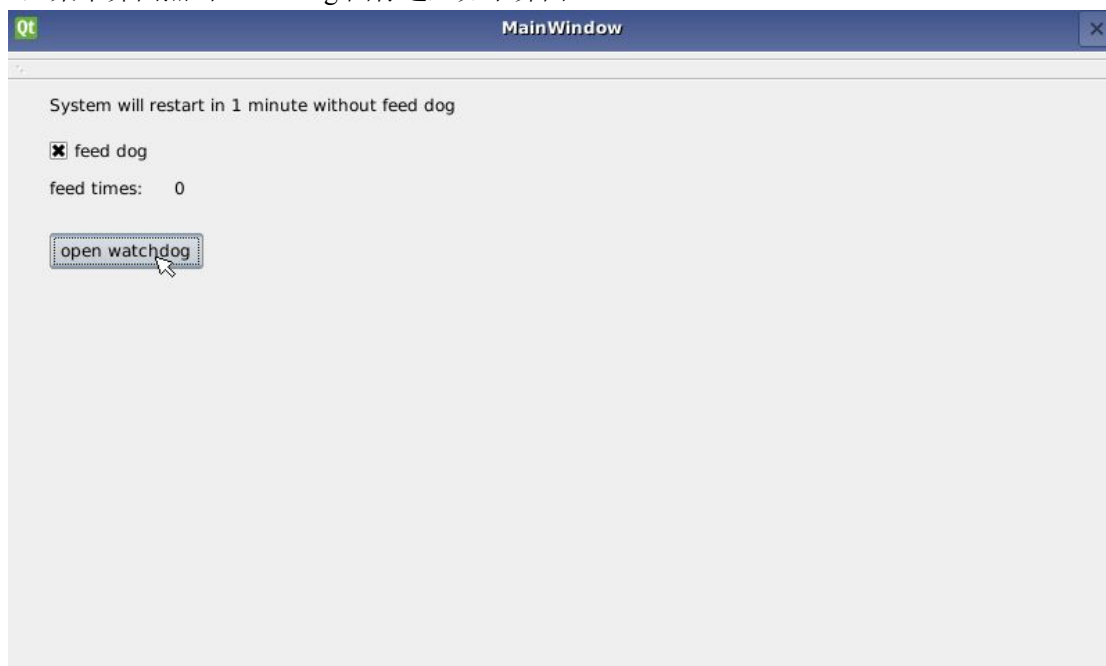
然后单击 read 按钮，显示如下界面：



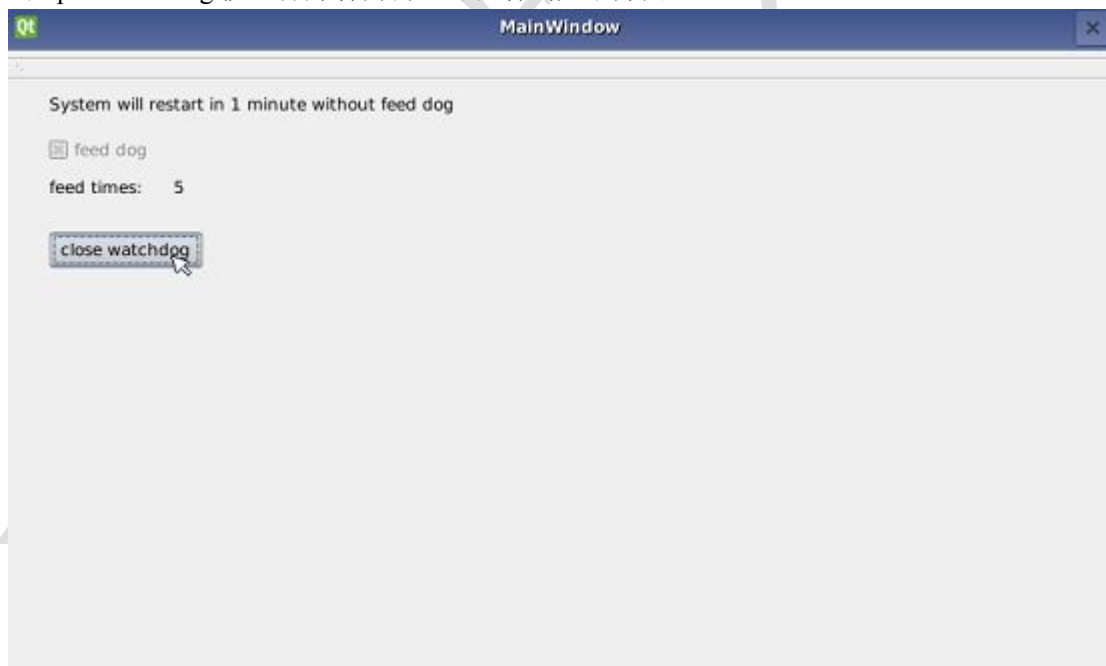


### 3.4.16 看门狗测试

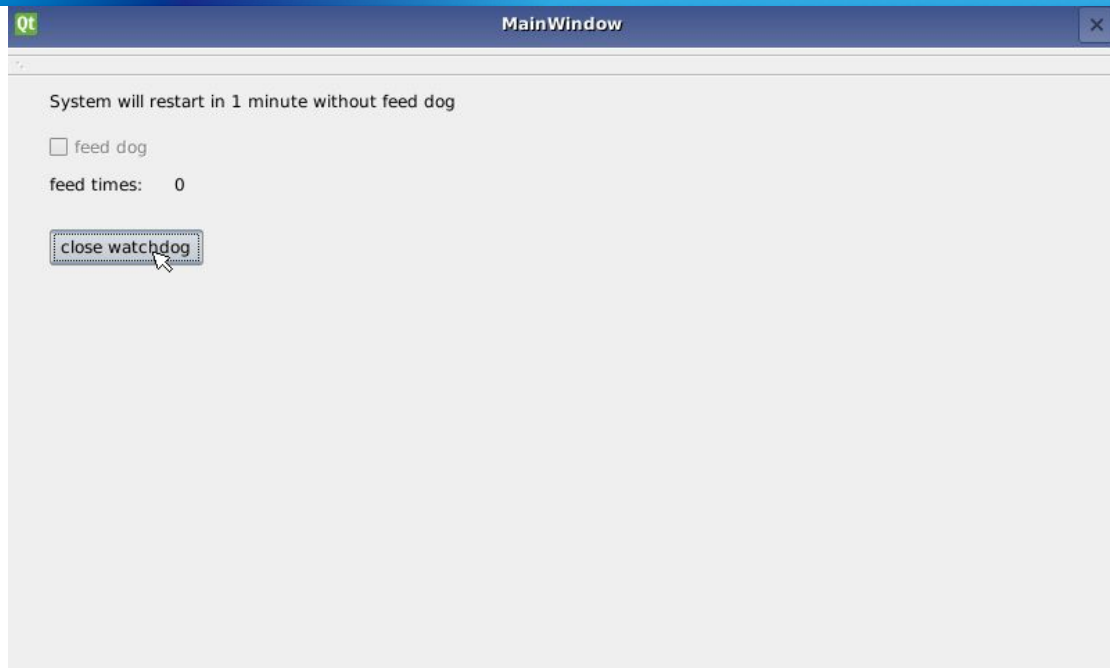
从 LCD 主菜单界面点击 Watchdog 图标进入如下界面：



用户单击 open watchdog 按钮打开看门狗，可以看到如下界面：



如果用户打开看门狗时没有勾选 feed dog 选钮，则会看到系统在一分钟重启，如下图：

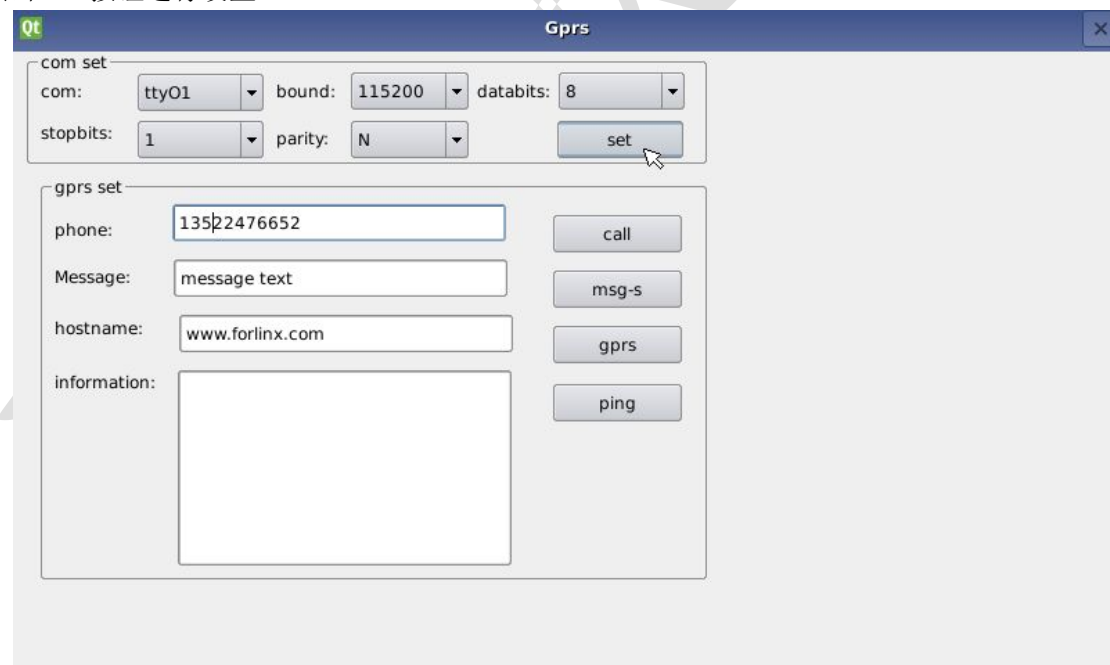


### 3.4.17 GPRS 测试

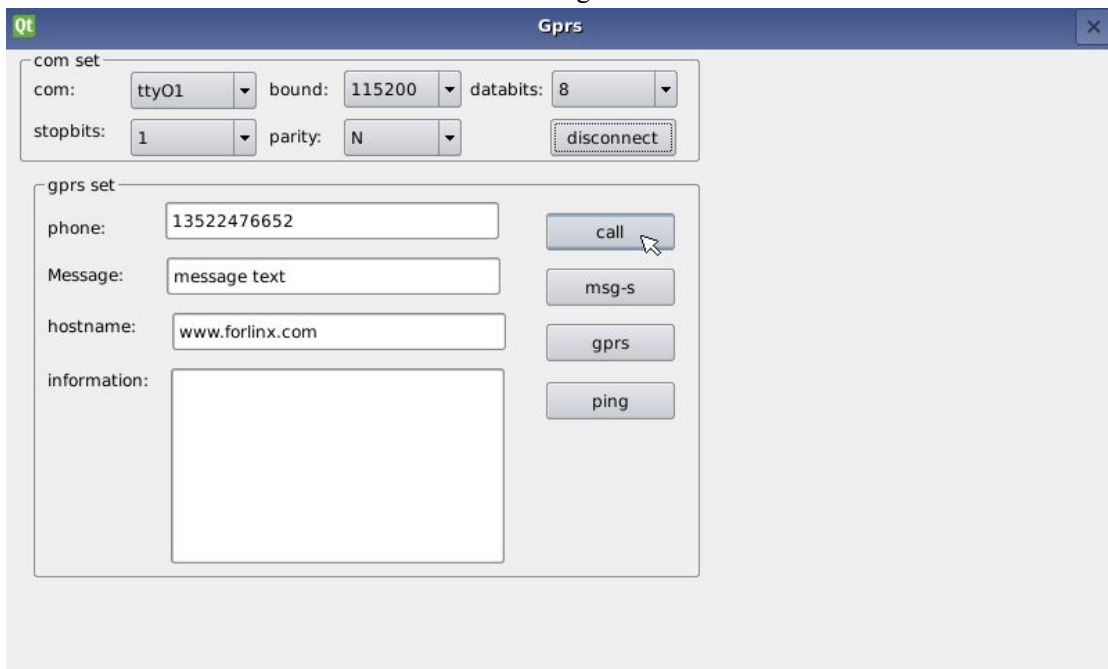
**注意：**该项功能默认情况下可以不进行测试，如果客户有 GPRS 的需求，再按照下面的步骤操作。

GPRS 模块与开发板之间采用串口连接，客户可以使用飞凌公司自产的 GPRS 模块，也可以使用自己购买的串口 GPRS 模块。

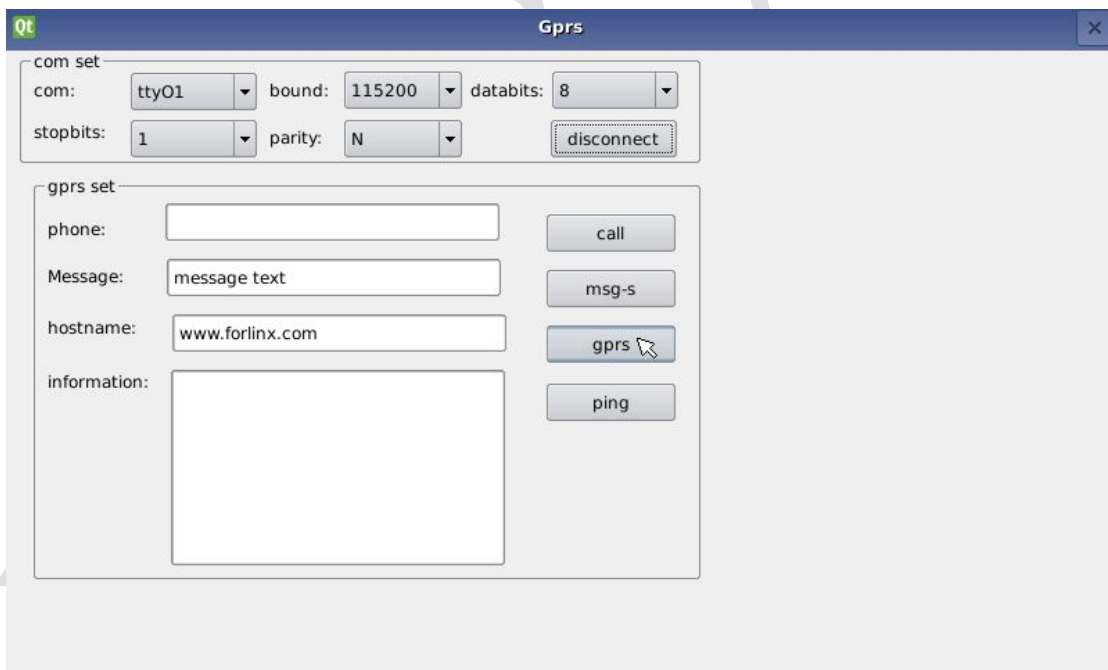
1. 选择 GPRS 模块连接的串口、设置串口波特率、数据位、奇偶校验、停止位、硬件流控，
2. 点击 set 按钮进行设置。



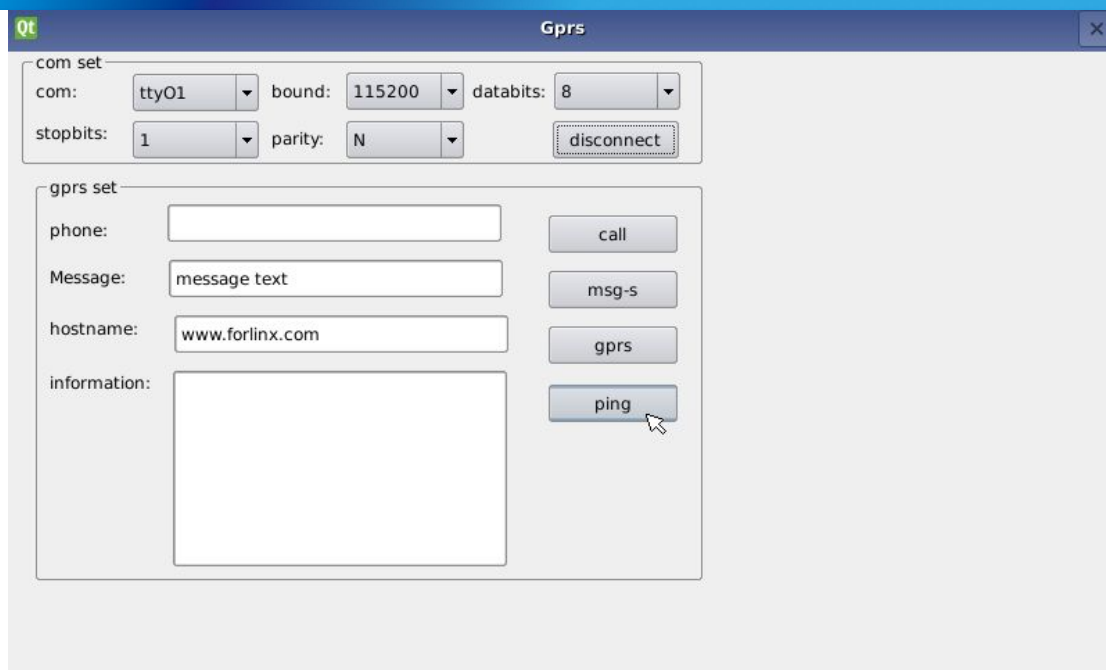
在 phone 栏添加对方的电话号码，分别点击 call 和 msg-s 按钮，进行拨打电话、发送短信息测试；



gprs 上网功能测试：单击界面上的 gprs 按钮，即可拨号上网：



用户可通过点击 ping 按钮测试 gprs 是否拨号成功：



如果拨打、接收电话，发送短信息，gprs 上网，说明 gprs 模块工作正常。

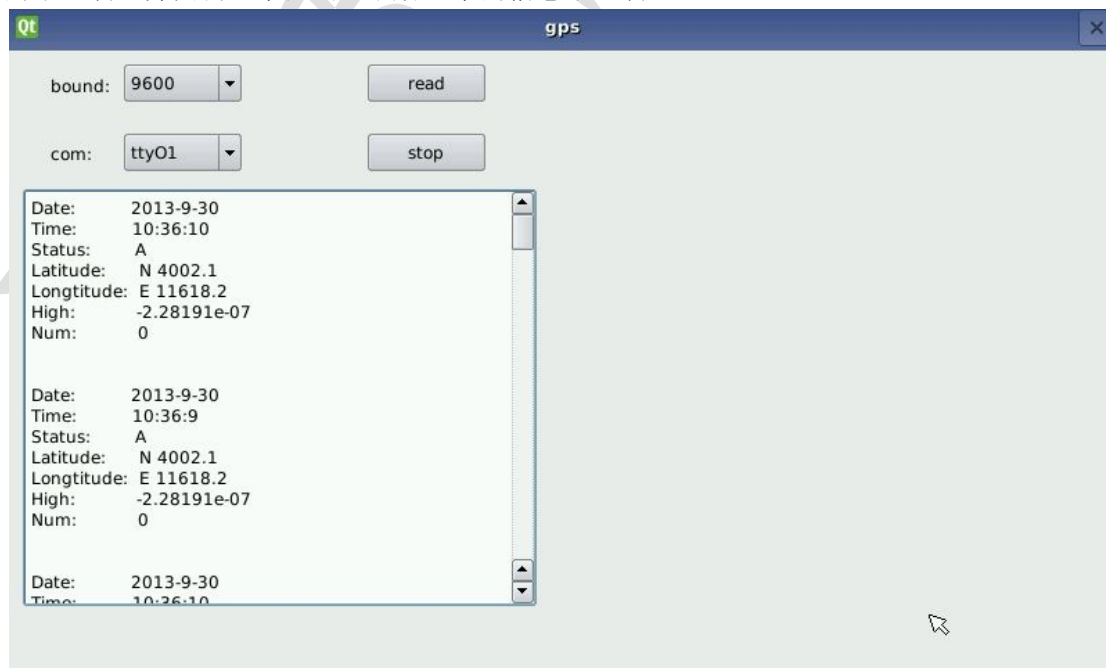
### 3.4.18 GPS 测试

**注意：**该项功能默认情况下可以不进行测试，如果客户有 GPS 的需求，再按照下面的步骤操作。

GPS 模块与开发板之间采用串口连接，客户可以使用飞凌公司自产的 GPS 模块，也可以使用自己购买的串口 GPS 模块。

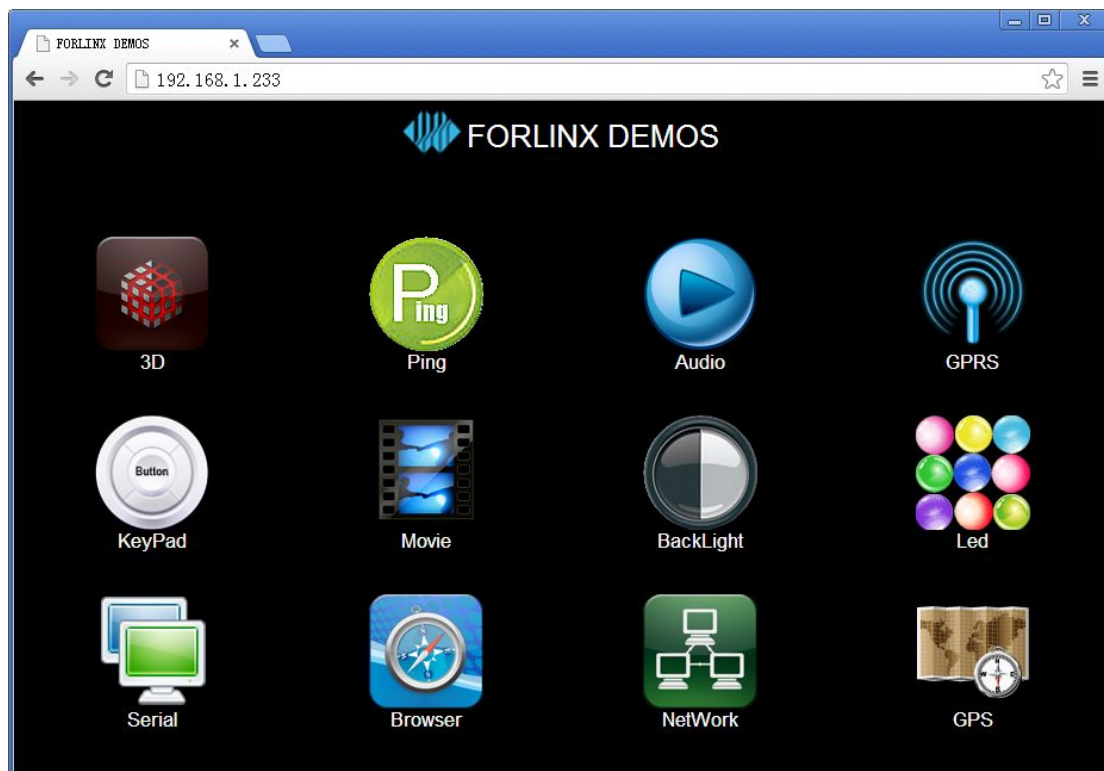
选择 GPS 模块与开发板之间连接的串口，设置串口的波特率，点击 read 按钮，串口信息会显示当前的日期时间，经纬度和海拔高度，这些信息是从卫星获取的，如果信息正确说明 GPS 模块工作正常..

注意:飞凌公司自产的 GPS 模块串口波特率是 9600，客户需要根据自己的模块灵活选择波特率，如果波特率设置的不正确，界面会显示乱码，或者显示的信息不正确。

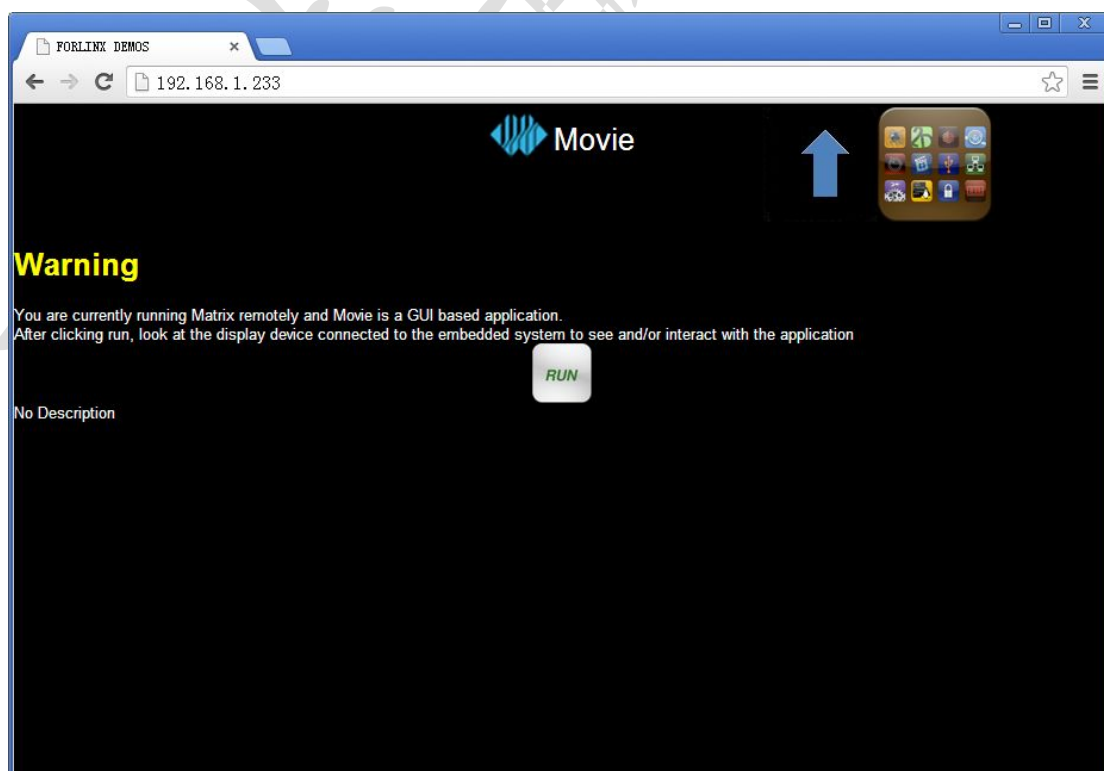


### 3.4.19 WEB 测试

用户可以通过 web 来访问开发板，开发板同 PC 通过网络连接以后，打开 PC 浏览器输入开发板地址，假如是 192.168.1.233，将出现如下图：



点击 Movie 图标将出现如下界面：



点击 RUN 按钮后开发板 LCD 屏幕上将出现电影画面，如下图：



同理其他测试程序也可以通过浏览器来启动运行。



## 第四章 Linux 系统的编译

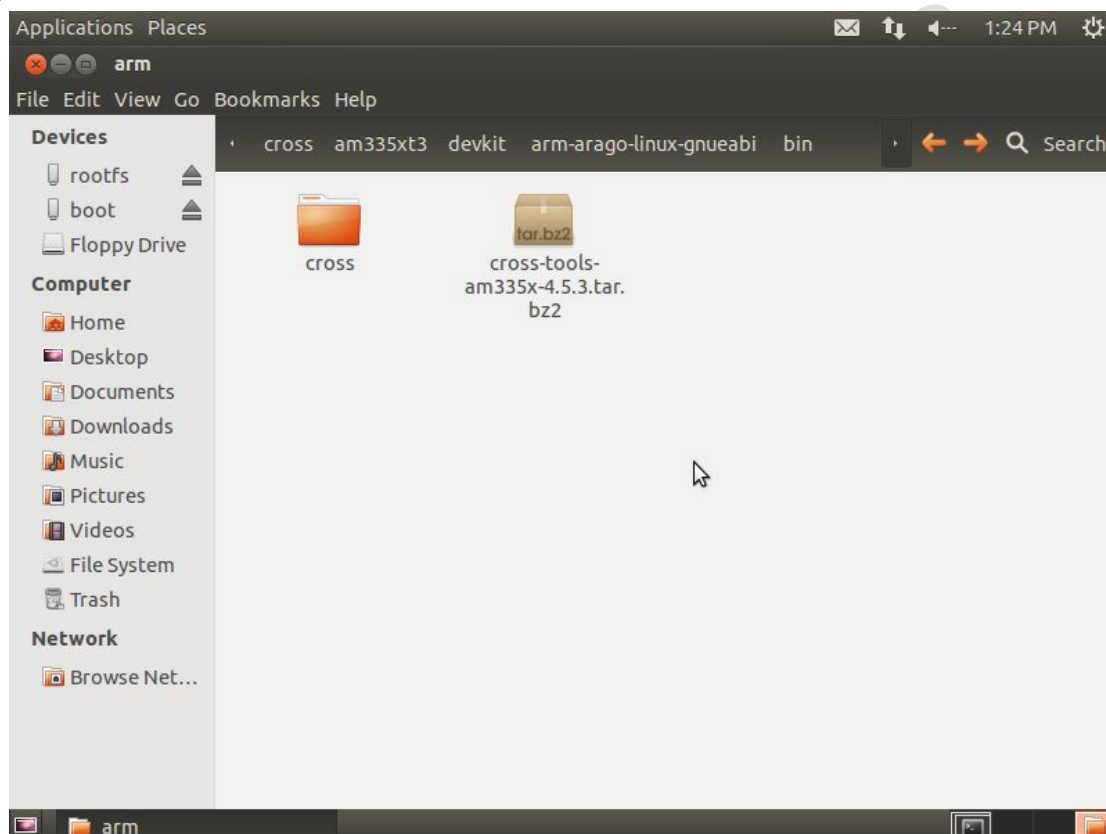
光盘 src 目录下包含 uboot 源码，内核源码，测试源码等。

### 4.1 安装编译器

1. 拷贝 tools 目录下的 cross-tools-am335x-4.5.3.tar.bz2 文件到 Ubuntu 的/usr/local/arm（推荐）目录下面：解压

2. `tar xvf cross-tools-am335x-4.5.3.tar.bz2`

如下图：



3. 增加编译器的环境变量

修改 /etc/profile 文件，增加：

```
export PATH=/usr/local/arm/cross/am335xt3/devkit/bin:$PATH
```

保存，退出，重新启动系统，这样每次进入 Ubuntu 系统，会自动增加环境变量。

## 4.2 U-boot 的编译

拷贝 src 目录下的 u-boot-2011.09-forlinx.tar.bz2 文件到 ubuntu 的 /home/forlinx/work 下面，然后执行以下命令：

### 4.2.1 普通版本 uboot 的编译

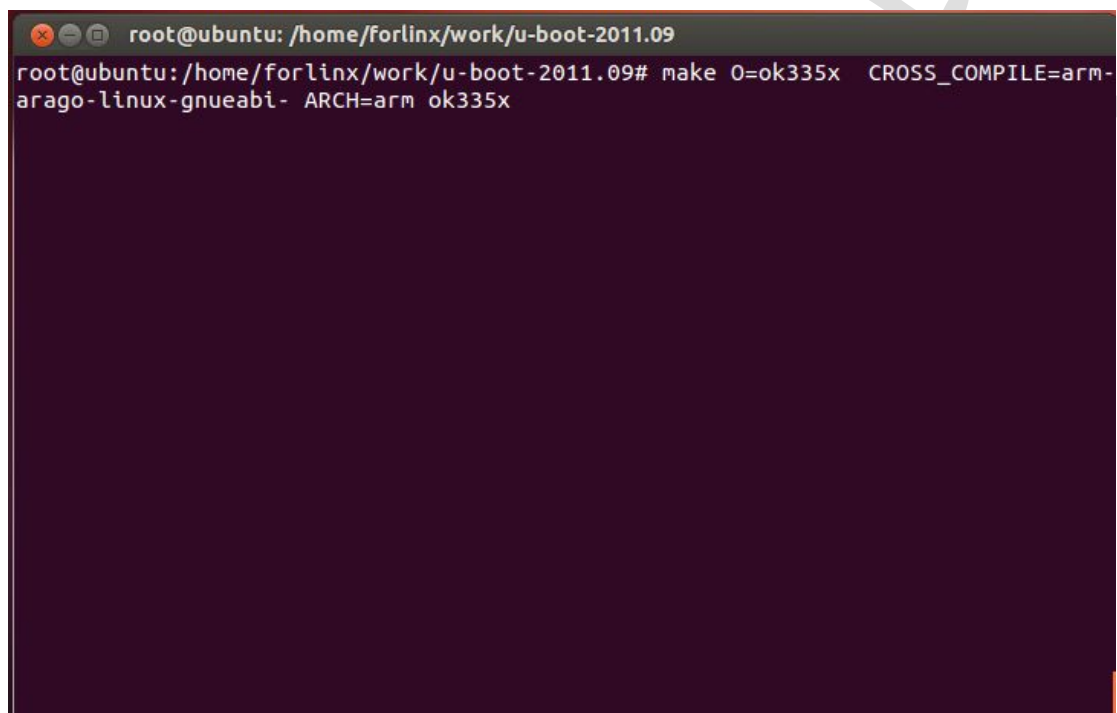
1. 解压

```
tar xvf u-boot-2011.09-forlinx.tar.bz2
```

```
cd u-boot-2011.09
```

2. 编译

```
make O=ok335x CROSS_COMPILE=arm-arago-linux-gnueabi- ARCH=arm ok335x
```



```
root@ubuntu: /home/forlinx/work/u-boot-2011.09
root@ubuntu:/home/forlinx/work/u-boot-2011.09# make O=ok335x CROSS_COMPILE=arm-
arago-linux-gnueabi- ARCH=arm ok335x
```

- 编译完成后会在 ok335x 文件目录下面生成 MLO 和 u-boot.img 文件，将其拷入 SD 卡 boot 分区或者烧写入 nand 分区后即可使用，如下图：

```

root@ubuntu: /home/forlinx/work/u-boot-2011.09/ok335x
root@ubuntu:/home/forlinx/work/u-boot-2011.09/ok335x# ls
api      common  examples include2 net      System.map u-boot.bin  u-boot.map
arch     disk    fs       lib      post    tools      u-boot.img  u-boot.srec
board    drivers include MLO      spl     u-boot     u-boot.lds
root@ubuntu:/home/forlinx/work/u-boot-2011.09/ok335x#

```

#### 4.2.2 批量烧写版本的 uboot 编译

该版本的 uboot 需要打开条件编译宏 AUTO\_UPDATESYS (include/configs/ok335x.h)，然后像编译普通 uboot 一样进行编译即可。

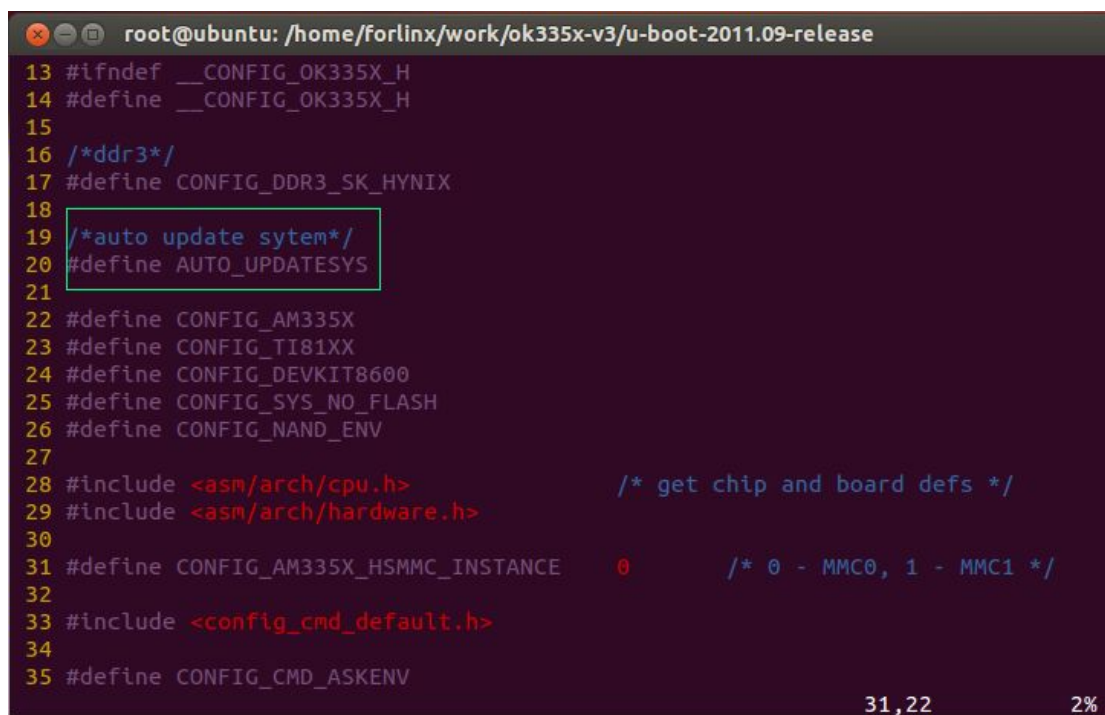
编译宏打开前如下图：

```

root@ubuntu: /home/forlinx/work/ok335x-v3/u-boot-2011.09-release
13 #ifndef __CONFIG_OK335X_H
14 #define __CONFIG_OK335X_H
15
16 /*ddr3*/
17 #define CONFIG_DDR3_SK_HYNIX
18
19 /*auto update sytem*/
20 /*#define AUTO_UPDATESYS*/
21
22 #define CONFIG_AM335X
23 #define CONFIG_TI81XX
24 #define CONFIG_DEVKIT8600
25 #define CONFIG_SYS_NO_FLASH
26 #define CONFIG_NAND_ENV
27
28 #include <asm/arch/cpu.h> /* get chip and board defs */
29 #include <asm/arch/hardware.h>
30
31 #define CONFIG_AM335X_HSMC_INSTANCE 0 /* 0 - MMC0, 1 - MMC1 */
32
33 #include <config_cmd_default.h>
34
35 #define CONFIG_CMD_ASKENV

```

编译宏打开后如下图：



```

root@ubuntu: /home/forlinx/work/ok335x-v3/u-boot-2011.09-release
13 #ifndef __CONFIG_OK335X_H
14 #define __CONFIG_OK335X_H
15
16 /*ddr3*/
17 #define CONFIG_DDR3_SK_HYNIX
18
19 /*auto update sytem*/
20 #define AUTO_UPDATESYS
21
22 #define CONFIG_AM335X
23 #define CONFIG_TI81XX
24 #define CONFIG_DEVKIT8600
25 #define CONFIG_SYS_NO_FLASH
26 #define CONFIG_NAND_ENV
27
28 #include <asm/arch/cpu.h> /* get chip and board defs */
29 #include <asm/arch/hardware.h>
30
31 #define CONFIG_AM335X_HSMC_INSTANCE 0 /* 0 - MMC0, 1 - MMC1 */
32
33 #include <config_cmd_default.h>
34
35 #define CONFIG_CMD_ASKENV
31,22 2%
```

## 4.3 Linux 的编译

拷贝 src 目录下的 linux-3.2.0-forlinx.tar.bz2 文件到 /home/forlinx/work 下面，然后执行以下命令。

### 1. 解压

```
tar xvf linux-3.2.0-forlinx.tar.bz2
cd linux-3.2.0
```

### 2. 编译

```
cp arch/arm/configs/ok335xd_evm_linux_defconfig .config
make CROSS_COMPILE=arm-arago-linux-gnueabi- ARCH=arm uImage
```

编译完成后会自动在 arch/arm/boot/ 目录形成 uImage 文件，将该文件拷入 SD 卡 boot 分区或烧写到 nand 后即可使用。

**注意：**编译内核时如果说找不到 mkimage, 请将 tools 目录下 mkimage 复制到 ubuntu /usr/bin 目录下。

通过命令 **chmod 777 mkimage** 赋予 mkimage 工具可执行权限

## 4.4 文件系统制作

对于 SD 卡，文件系统是 ext3 格式的，如需添加文件直接往里拷贝即可，对于 Nand， 文件系统是 ubi 格式的文件系统，下面说一下如何在我们提供的 ubi 文件系统中增加自己的文件。

1. 拷贝 filesystem 目录下 rootfs.tar.bz2 文件到 /home/forlinx/work 目录下，拷贝 tools 目录下的 mkfs.ubifs, , ubinize, ubinize.cfg 到 /home/forlinx/work 目录下然后执行：

2. 创建一个新文件夹

```
mkdir rootfs
```

3. 将文件系统解压到 rootfs 目录

```
tar xvf rootfs.tar.bz2 -C rootfs
```

根据您的需求修改 rootfs 文件夹里面的内容，比如增加脚本文件，设置 IP 地址等等

4. 重新生成 ubi.img

```
./mkfs.ubifs -r rootfs -m 2048 -e 126976 -c 1900 -o ubifs.img
```

```
./ubinize -o ubi.img -m 2048 -p 128KiB -s 512 -O 2048 ubinize.cfg
```

```
rm ubifs.img
```

5. 将新的 ubi.img 重新烧写到 nand 即可。

## 4.5 驱动代码路径

设备	驱动程序源代码在内核中的位置	对应的设备名
网卡驱动	drivers/net/ethernet/ti/cpsw.c	eth0
LED 驱动	drivers/misc/led_dev.c	/dev/led
LCD 背光驱动	drivers/video/backlight/pwm_bl.c	/sys/class/backlight
USB 接口 U 盘	drivers/usb/storage/	/dev/sdX
USB 鼠标	drivers/hid/usbhid/	/dev/input/mice
Flash ECC 校验	drivers/mtd/nand/nand_ecc.c	无
Nand Flash 驱动	drivers/mtd/nand/omap2.c	/dev/mtdX,mtdblockX
UBI 文件系统	fs/ubifs/	无
SD 卡驱动	drivers/mmc/card/	/dev/mmcblk0
LCD FrameBuffer	drivers/video/da8xx_fb.c	/dev/fb0
电阻触摸驱动	drivers/input/touchscreen/ti_tsc.c	/dev/input/touchscreen0
电容触摸驱动	drivers/input/touchscreen/ft5x06_ts.c	/dev/input/touchscreen0
RTC 实时时钟驱动	drivers/rtc/rtc-ds1307.c	/dev/rtc0
SPI 驱动	drivers/spi/spidev.c	/dev/spidev2.0
串口(含三个串口)	drivers/tty/serial/omap-serial.c	/dev/ttyO0,1,4
按键驱动	drivers/input/keyboard/gpio_keys.c	/dev/input/keypad
看门狗驱动	drivers/watchdog/omap_wdt.c	/dev/watchdog
RS485 驱动	drivers/tty/serial/omap-serial.c	/dev/ttyO2
EEPROM 驱动	drivers/misc/eeprom/at24.c	/dev/eeprom
CAN 驱动	drivers/net/can/d_can/d_can.c	can0
音频驱动	sound/soc/codec/tlv320aic3x.c	/dev/snd/controlC0, pcmC0D0c,pcmC0D0p, seq,timer



## 第五章 Qt 开发指南

虚拟机环境：ubuntu12.04

目标平台：ok335x 开发板

交叉编译工具链：gcc version 4.4.3

文件系统：forlinx 提供的 rootfs-mini

### 5.1 安装并设置交叉编译器环境

#### 5.1.1 安装交叉编译器

拷贝 tools 目录下的 arm-linux-4.4.3.tar.bz2 到/home/forlinx/work（推荐）目录下

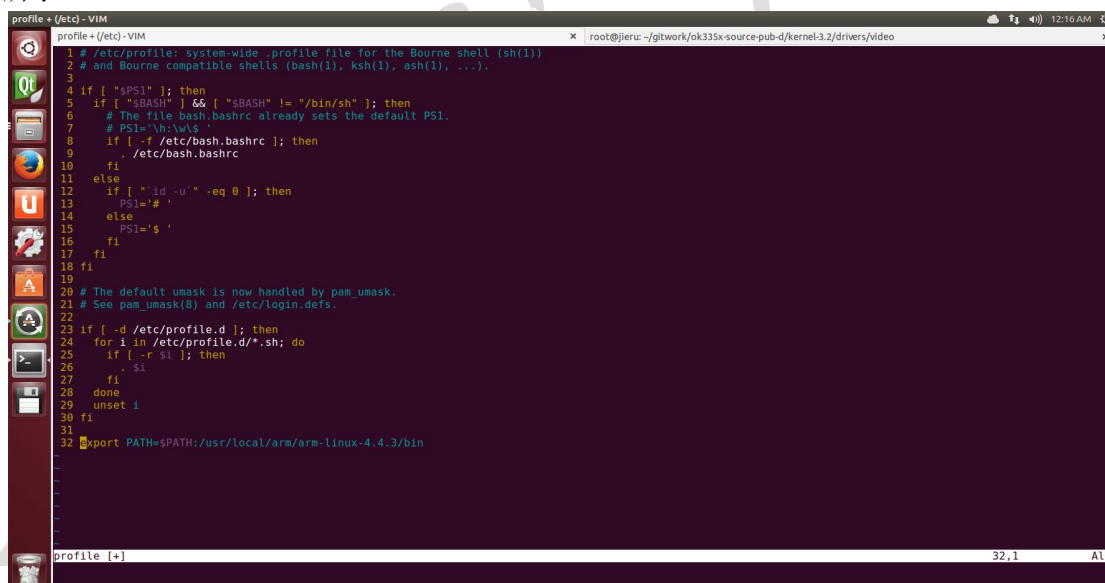
```
# tar xvf arm-linux-4.4.3.tar.bz2 -C /usr/local/arm
```

#### 5.1.2 设置交叉编译器环境变量

```
# vi /etc/profile
```

将 export PATH=\$PATH:/usr/local/arm/arm-linux-4.4.3/bin 添加到/etc/profile 文件

如下图所示：

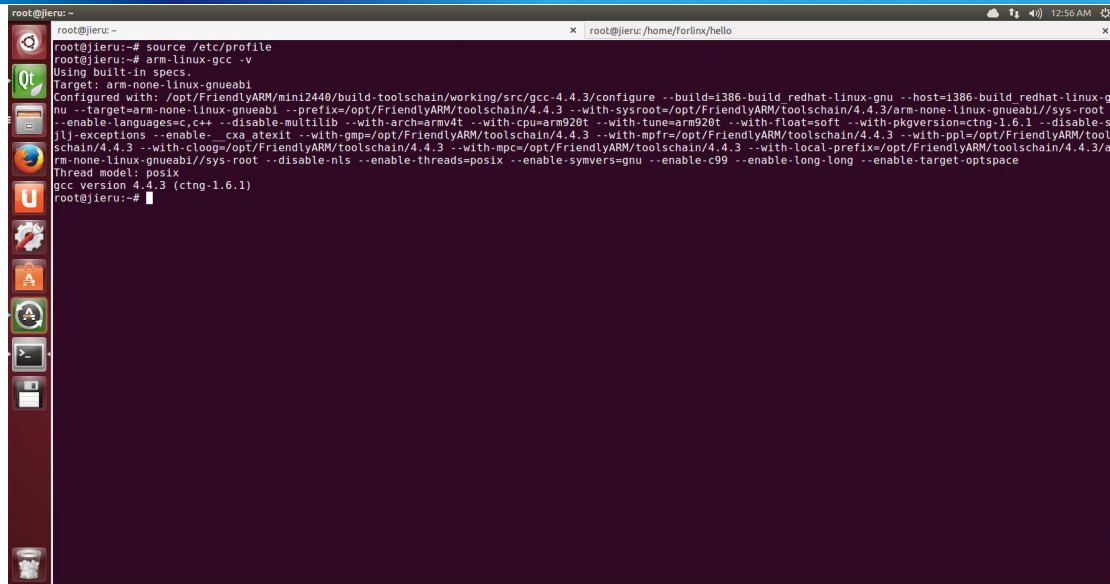


#### 5.1.3 导入环境变量

```
# source /etc/profile
```

```
# arm-linux-gcc -v
```

如下图所示则说明添加编译器成功：



## 5.2 移植 tslib

tslib 是电阻式触摸屏用于校准的一个软件库，是一个开源的程序，能够为触摸屏驱动获得的采样提供诸如滤波、去抖、校准等功能，通常作为触摸屏驱动的适配层，为上层的应用提供了一个统一的接口。因此这里先编译安装 tslib，这样在后面编译 Qt 的时候才能打包编译进去。

### 5.2.1 安装 autoconf、automake、libtool 包

```
# sudo apt-get install autoconf
# sudo apt-get install automake
# sudo apt-get install libtool
# sudo apt-get install g++
```

### 5.2.2 拷贝压缩文件

拷贝 src 目录下 tslib-1.4.tar.bz2 的到/home/forlinx/work（推荐）目录下面

```
# tar -xvf tslib-1.4.tar.bz2
# cd tslib
# ./tslib.sh #飞凌提供的编译脚本，执行完会将 tslib 默认安装到/usr/local/arm/tslib
```

### 5.2.3 移植 tslib 到 ok335x 目标板

假设用户的根文件系统路径为：/home/forlinx/work/rootfs-mini

执行以下命令：

```
# cd /usr/local/arm/tslib
# cp -ar lib/* /home/forlinx/work/rootfs-mini/usr/lib
# cp -ar bin/* /home/forlinx/work/rootfs-mini/usr/bin
# vi etc/ts.conf 去掉 module_raw input 前注释符#,一定要注意中间的空格也去掉, module 顶格
# cp -ar etc/* /home/forlinx/work/rootfs-mini/etc
```

## 5.2.4 导出 tslib 环境变量

```
# cd /home/forlinux/work/rootfs-mini
# vi etc/profile #添加以下环境变量
export TSLIB_TSDEVICE=/dev/input/touchscreen0
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_PLUGINDIR=/usr/lib/ts
export POINTERCAL_FILE=/etc/pointercal
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
```

< tslib 环境变量说明:

TSLIB_CONFFILE	//配置文件名
TSLIB_PLUGINDIR	//插件目录
POINTERCAL_FILE:	//指定包含用于校准指针设备的数据的文件
TSLIB_TSDEVICE	//触摸屏设备文件名
TSLIB_CALIBFILE	//校准的数据文件, 由 ts_calibrate 校准程序生成
TSLIB_CONSOLEDEVICE	//控制台设备文件名
TSLIB_FBDEVICE	//设备名

以上环境变量在实际开发中的实际配置可以根据实际情况决定。而这些指定的设备节点一定要和你的开发板上的/dev 目录下的设备节点相对应。

>

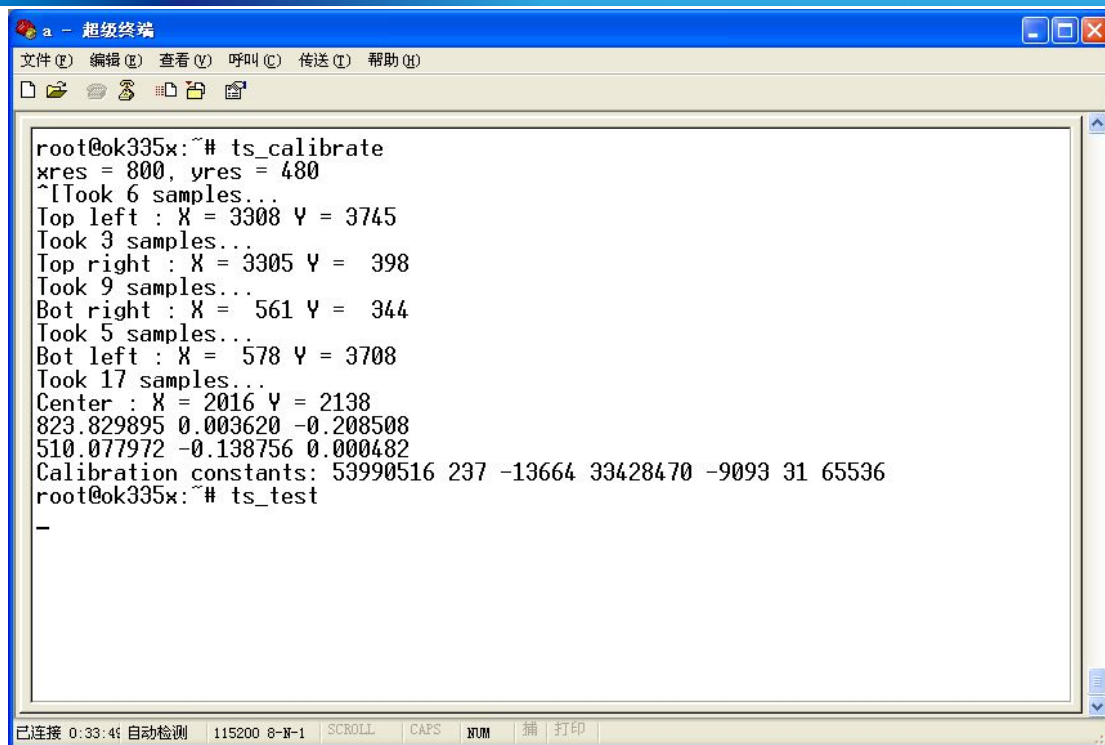
## 5.2.5 在 OK335x 目标板下测试 tslib

用新文件系统启动 OK335x 板子, 进入 shell 后执行如下命令:

```
# ts_calibrate    对电阻屏进行触摸较验
```

```
# ts_test        对电阻屏进行触摸测试
```

测试结果如下图所示:



```

root@ok335x:~# ts_calibrate
xres = 800, yres = 480
^[[ook 6 samples...
Top left : X = 3308 Y = 3745
Took 3 samples...
Top right : X = 3305 Y = 398
Took 9 samples...
Bot right : X = 561 Y = 344
Took 5 samples...
Bot left : X = 578 Y = 3708
Took 17 samples...
Center : X = 2016 Y = 2138
823.829895 0.003620 -0.208508
510.077972 -0.138756 0.000482
Calibration constants: 53990516 237 -13664 33428470 -9093 31 65536
root@ok335x:~# ts_test
-

```

## 5.3 移植 qt4.8.5

### 5.3.1 拷贝压缩文件

拷贝 src 目录下的 qt-everywhere-opensource-src-4.8.5.tar.bz2 到/home/forlinux/work（推荐）目录下面

```
# tar -xvf qt-everywhere-opensource-src-4.8.5.tar.bz2
```

```
# cd qt-everywhere-opensource-src-4.8.5
```

```
# ./qteverywhere.sh ##飞凌提供的编译脚本，执行完会将 qt 默认安装到/usr/local/arm/qt4.8.5
```

< 说明:

如果编译出现缺少链接信息错误:

warning: libts-0.0.so.0, needed by , not found (try using -rpath or -rpath-link)

undefined reference to `ts\_read\_raw'

undefined reference to `ts\_open'

undefined reference to `ts\_fd'

undefined reference to `ts\_config'

undefined reference to `ts\_close'

undefined reference to `ts\_read'

collect2: ld returned 1 exit status

解决方法如下:

修改 qt-everywhere-opensource-src-4.8.5/mkspecs/qws/linux-arm-g++/qmake.conf 文件（添加 lts 参数）:

```
QMAKE_CC = arm-linux-gcc -lts
```

```
QMAKE_CXX = arm-linux-g++ -lts
```

```
QMAKE_LINK = arm-linux-g++ -lts
```

```
QMAKE_LINK_SHLIB = arm-linux-g++ -lts
```

修改完成后保存退出，继续编译。

>

### 5.3.2 测试 arm 版 qt 是否安装并配置成功

将编译好的/usr/local/arm/qt4.8.5/bin 下 qmake 重命名为 arm-qmake，然后拷贝到/usr/bin 下

```
# cd /usr/local/arm/qt4.8.5/bin
```

```
# cp qmake /usr/bin/arm-qmake
```

```
# arm-qmake -v          #以后就可以通过 arm-qmake 来交叉编译 qt 程序了
```

```
QMake version 2.01a
```

```
Using Qt version 4.8.5 in /usr/local/arm/qt4.8.5/lib
```

### 5.3.3 移植 qt4.8.5 到 OK335x 目标板

```
# cd /usr/local/arm/qt4.8.5/
```

```
# cp -ar lib/libQt* lib/fonts/ /home/forlinux/work/rootfs-mini/forlinux/qt/lib
```

```
# cp -ar demos/embeddeddialogs/embeddeddialogs /home/forlinux/work/rootfs-mini/forlinux/qt/bin
```

注：若 forlinux/qt{lib,bin} 目录不存在可以自己创建，也可以是其他目录

### 5.3.4 导出 qt 环境变量

```
# cd /home/forlinux/work/rootfs-mini
```

```
# vi etc/profile #添加以下环境变量
```

```
export QTDIR=/forlinux/qt
```

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

```
export QT_QWS_FONTDIR=$QTDIR/lib/fonts
```

```
export QWS_SIZE=800x480
```

```
export QWS_DISPLAY=LinuxFb:/dev/fb0
```

```
export QWS_MOUSE_PROTO=Tslib:/dev/input/touchscreen0
```

< Qt 环境变量说明：

LD\_LIBRARY\_PATH：指定 qt 的共享库具体路径

QT\_QWS\_FONTDIR：指定 qt 的 font 具体路径

QWS\_SIZE：设定在屏幕尺寸

QWS\_DISPLAY：指定的显示类型和帧缓冲区

QWS\_MOUSE\_PROTO：指定触摸设备

以上环境变量在实际开发中的实际配置可以根据实际情况决定。而这些指定的设备节点一定要和你的开发板上的/dev 目录下的设备节点相对应。

>

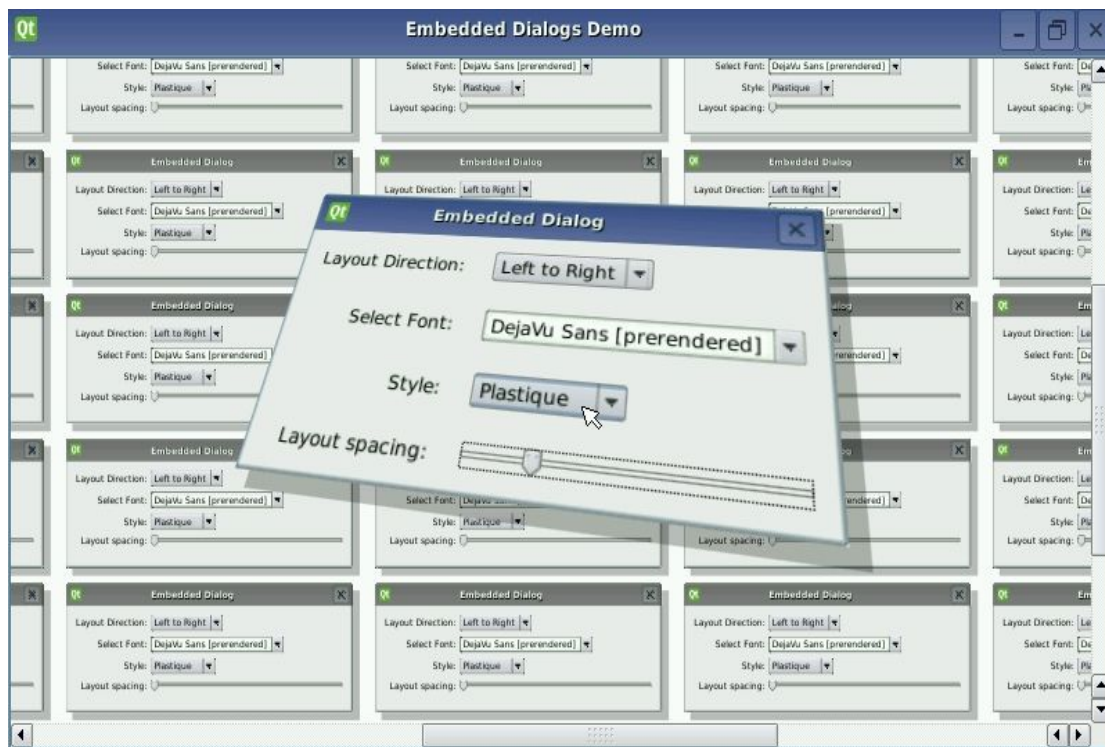
### 5.3.4 OK335x 目标板下测试 Qt4.8.5

用新移植的文件系统启动开发板，进入 shell 后执行如下命令：

```
# cd /forlinx/qt/bin
```

```
# ./embeddeddialogs -qws&
```

如下图所示：





## 5.4 Qt Creator 开发环境搭建及编程

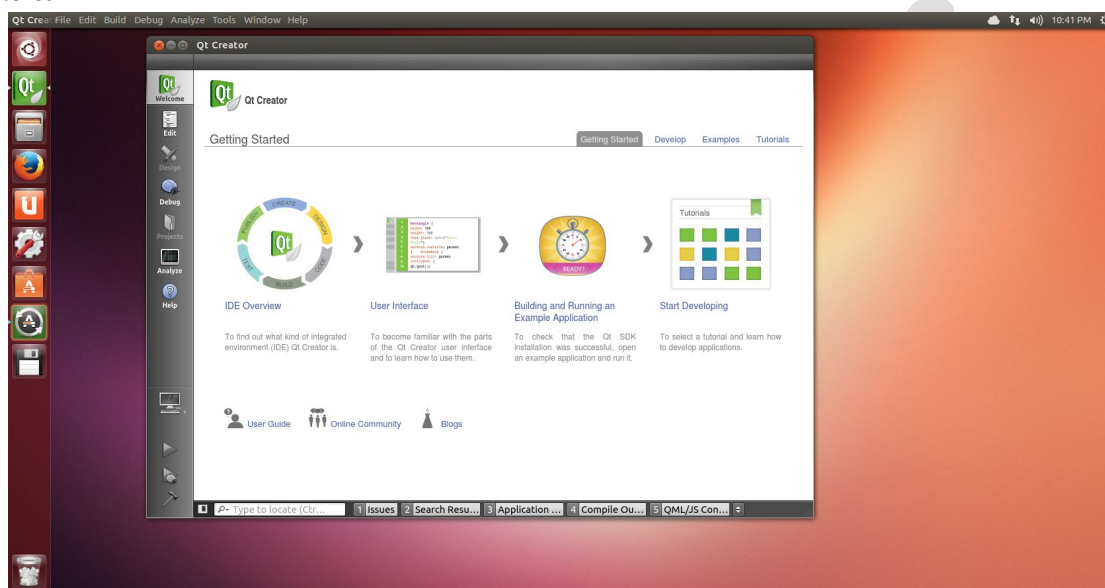
### 5.4.1 安装 Qt Creator

在 ubuntu 命令行执行如下命令

```
#apt-get install qtcreator
```

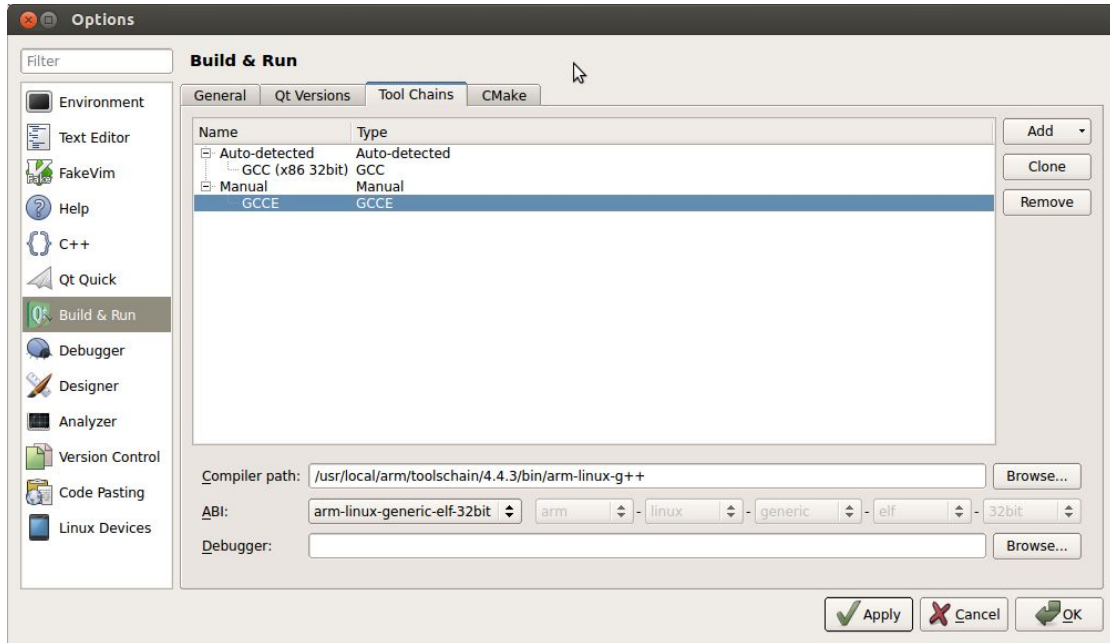
安装完成后, 在终端输入以下命令启动 Qt Creator:

```
#qtcreator&
```



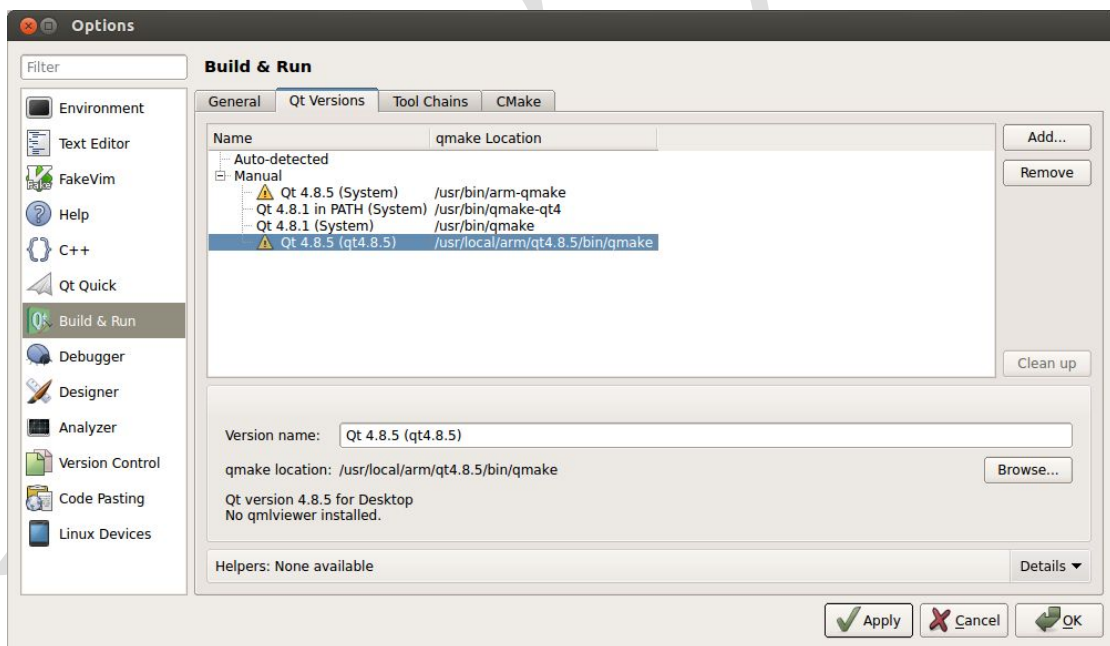
### 5.4.2 设置 Qt Creator 使其支持交叉编译

点击 Qt Creator 的 Tools 菜单下的 Options, 点击 Add 添加 GCCE, 然后点击 Compiler path Browse 添加交叉编译器(/usr/local/arm/arm-linux-4.4.3/bin/arm-linux-g++)的路径, 如下图所示:



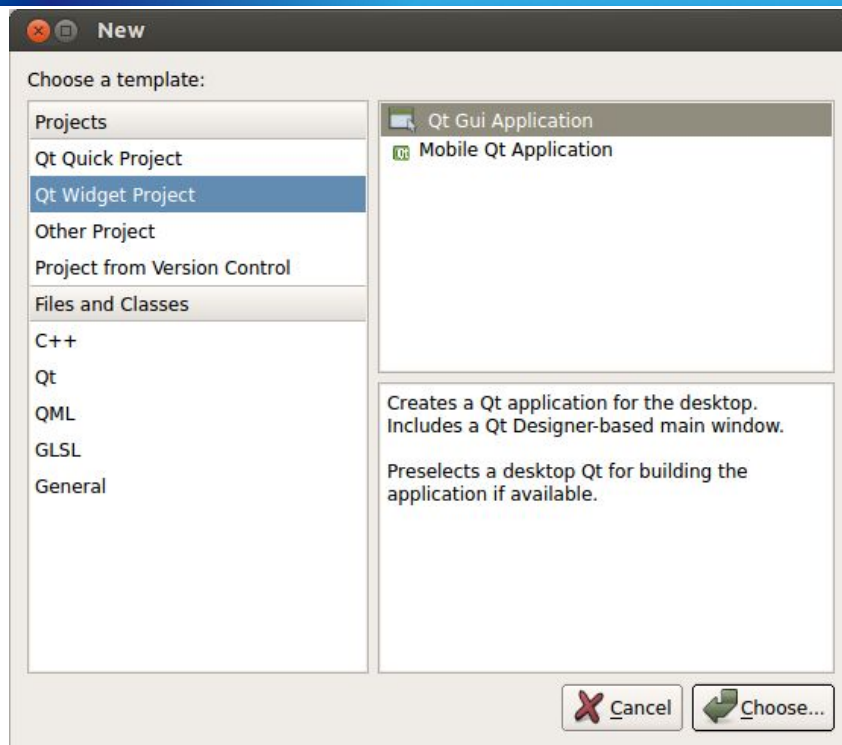
### Qt Creator Versions 的设置

点击 Add 添加/usr/local/arm/qt4.8.5/bin/qmake 文件，添加进去，之后，点击 Apply 即配置完成。如图所示：

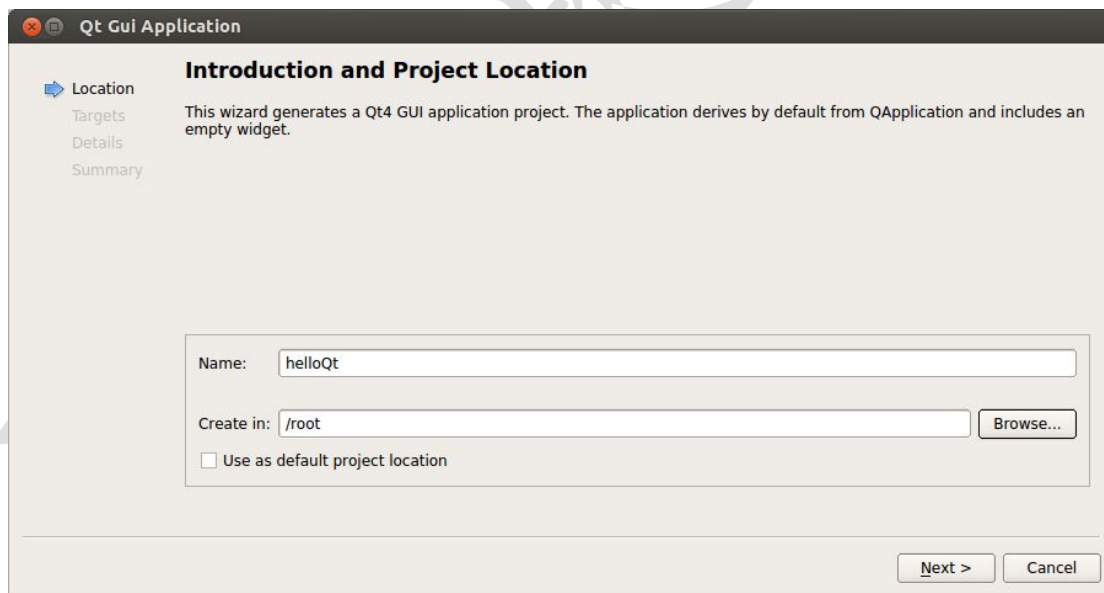


### 5.4.3 新建 hello Qt 程序进行测试

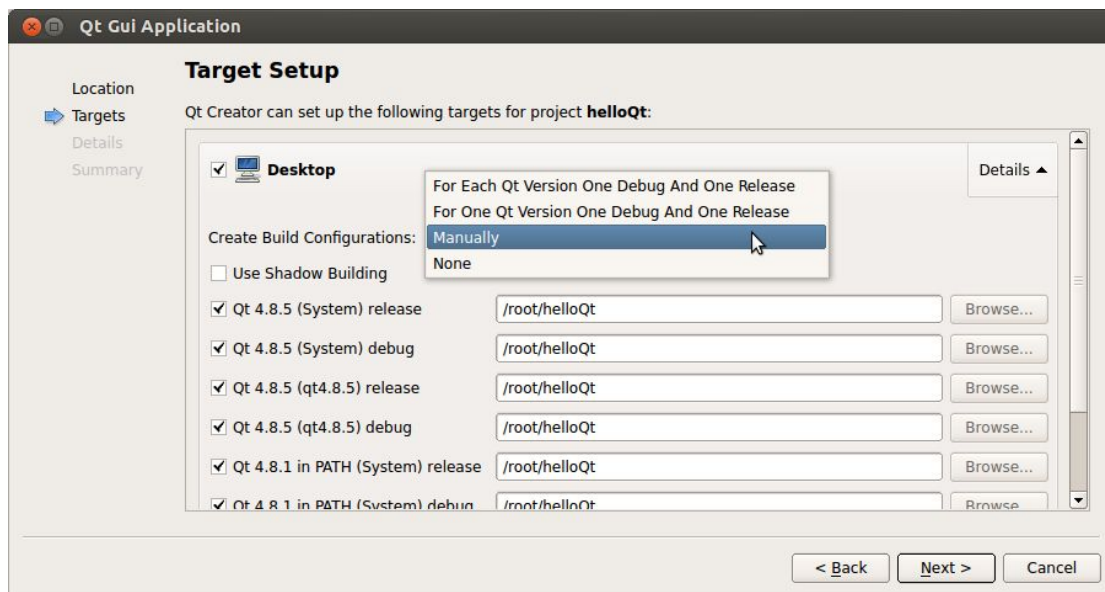
创建一个 Qt Widget Project 项目，如下图：



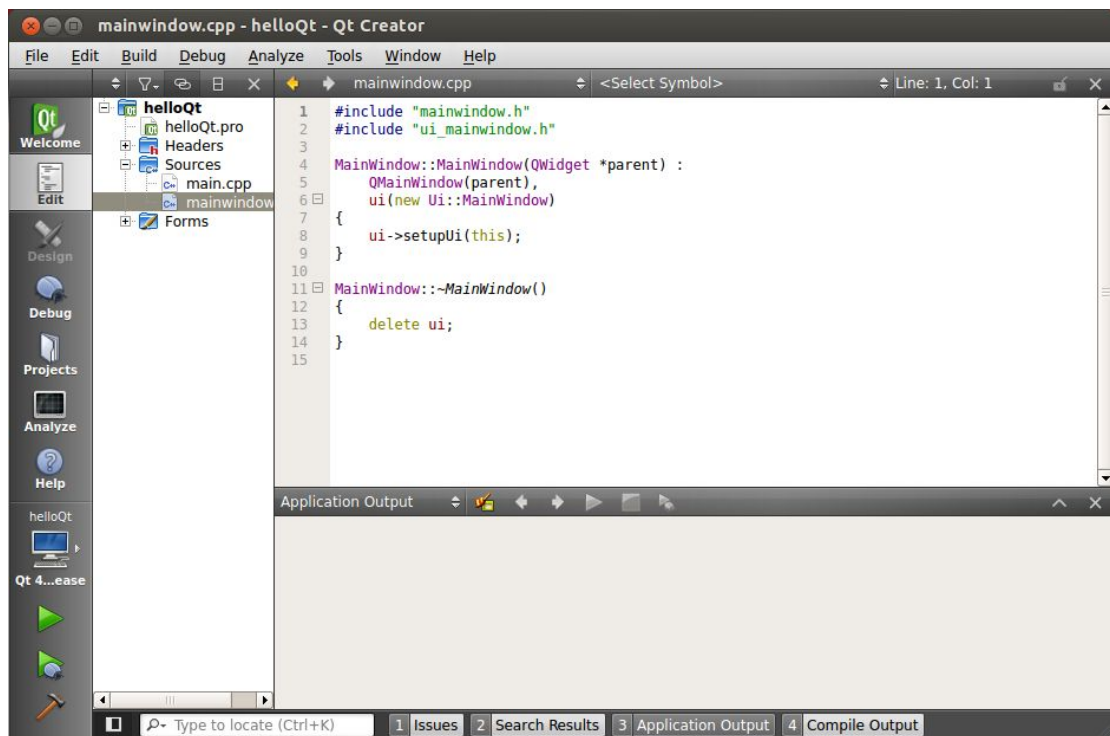
输入项目名称和项目路径，如下图：



选择 Manually 版本（就是我们刚才添加的那个交叉编译器版本，一定要选对）

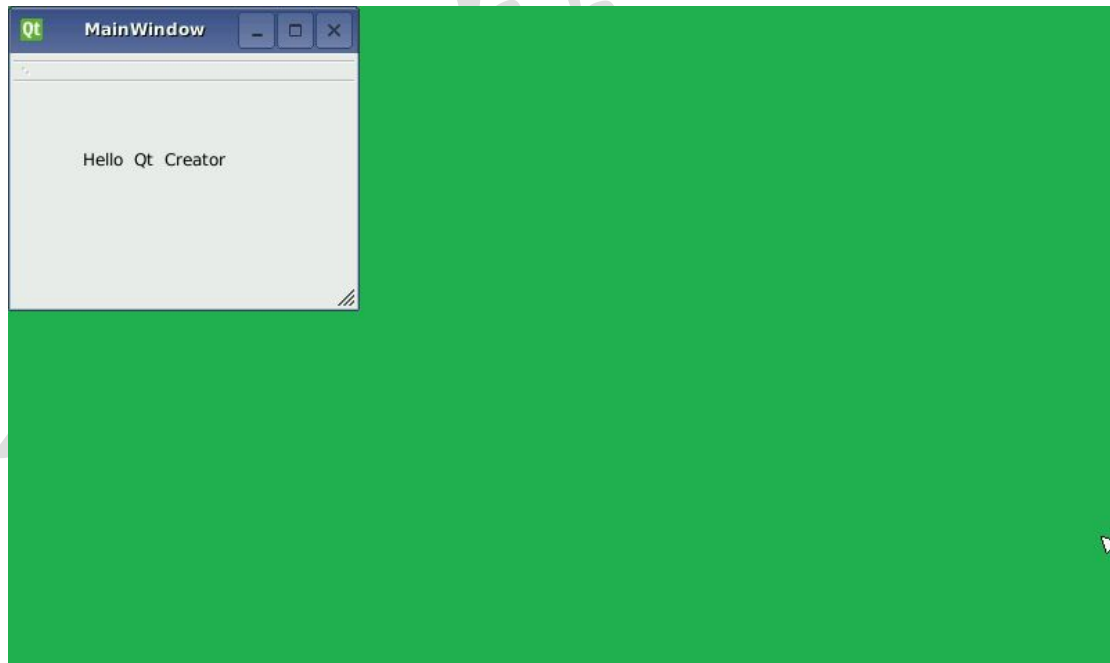


剩下的步骤默认就好，然后就可以通过 Build->Build All 进行编译，如下图：



#### 5.4.4 在开发板上测试 qt 应用程序

将编译好的应用程序移植到目标板文件系统中执行，测试效果如下图所示：



注：由于 qt 字体库里已包含一个中文字体，故直接可以在程序中使用中文，无需额外操作

## 附录 1：JTAG 调试

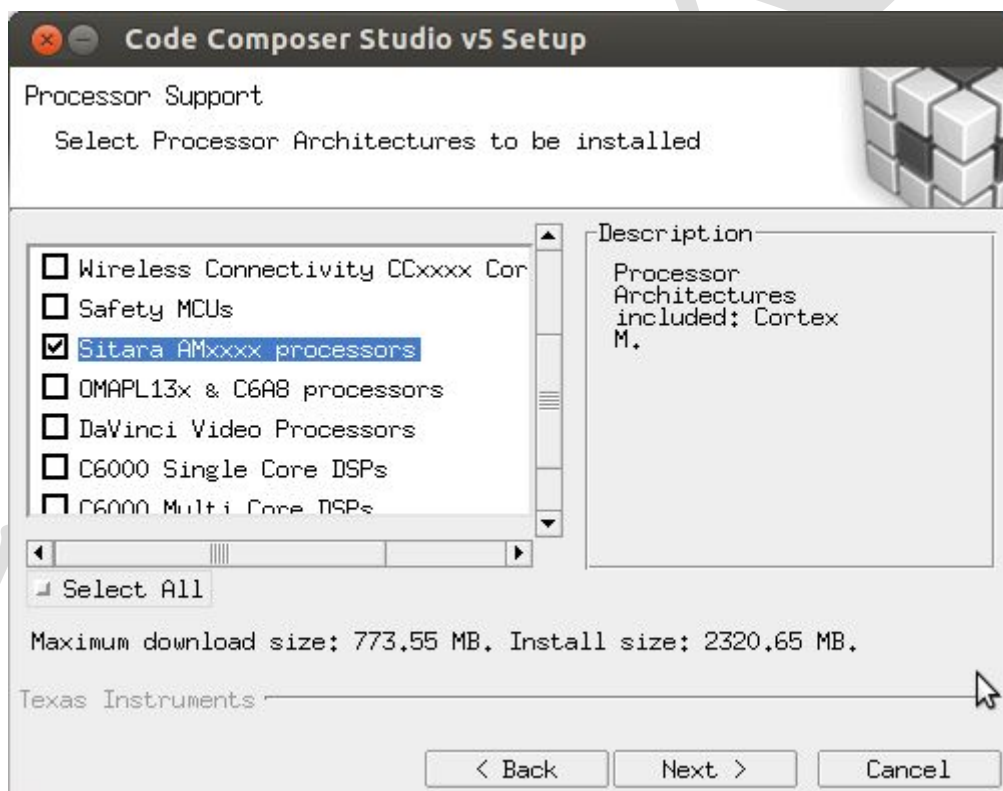
### F1.1 CCS 简介

CCS 是 TI 针对 DSP 和 ARM 系列芯片开发的基于 Eclipse 的集成调试环境,有 Windows 和 Linux 两种版本,由于 uboot 和内核的编译都是在 ubuntu 下,所以也推荐用户使用 Ubuntu 版本的 CCS,另外由于 am335x 系列芯片较新,只有 CCS5.4 及以上版本(目前最新版为 CCS5.5)的开发环境才支持,故以下教程中我们将基于 Ubuntu 版本的 CCS5.5 进行讲解,另外我们只使用 CCS 进行调试,确保已经编译过 uboot 或 linux 的源码。

下载地址 [http://processors.wiki.ti.com/index.php/Download\\_CCS](http://processors.wiki.ti.com/index.php/Download_CCS)

### F1.2 安装 CCSV5.5

从官网下载 CCS5.5 安装器后(下载时需要注册 TI 账号,且只能下载在线安装器),安装时默认路径为(/opt/ti),由于为在线安装,故安装过程十分缓慢(将持续数小时),为了加速安装过程可以在处理器选择界面只勾选 AMXXXX 芯片,如下图:



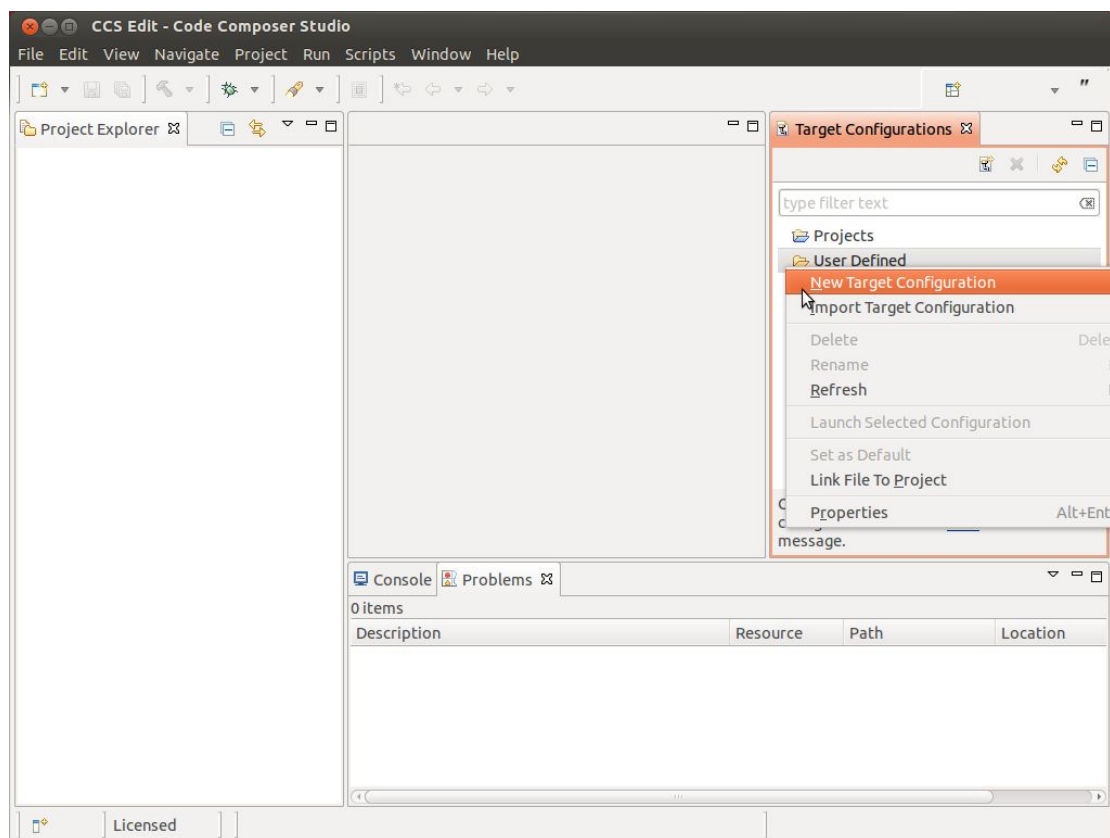
其他一路默认即可。

由于 CCSV5.5 自带 XDS100v2(本教程所使用的仿真器)的驱动,故无需安装任何额外软件,另外当 CCS 弹出 license 设置窗口时用户只需选择 Free License 即可。

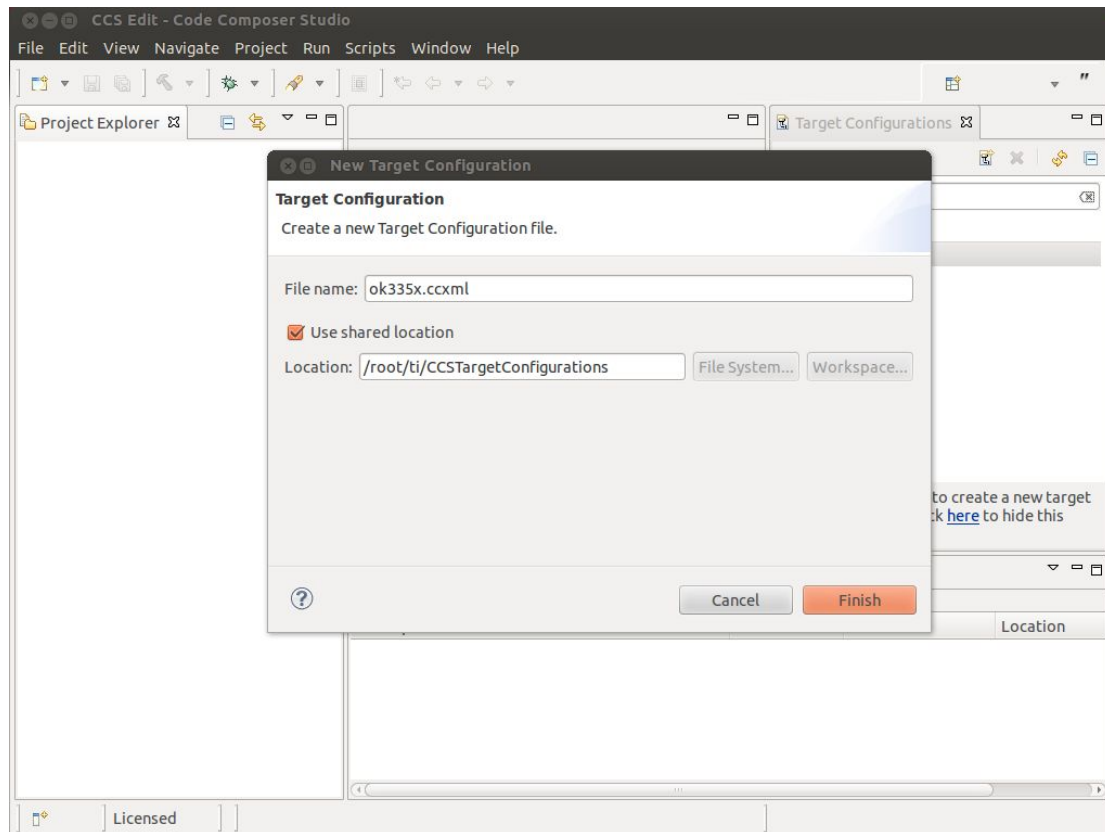


## F1.3 连接开发板进行测试

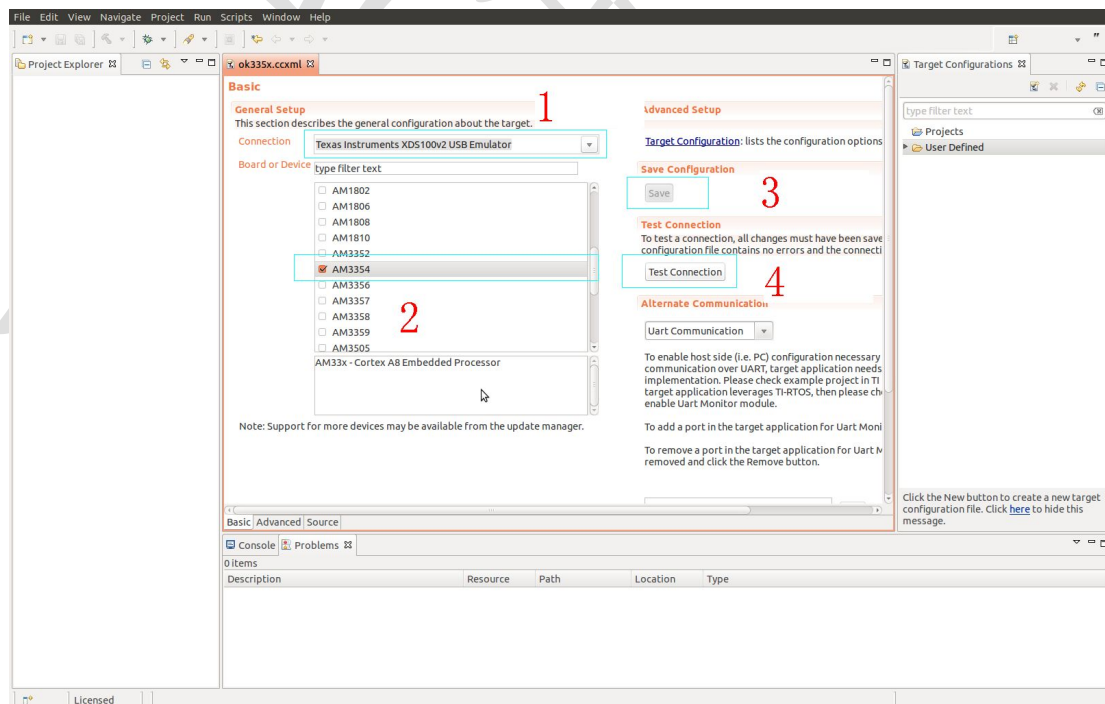
使用仿真器将开发板同 pc 连接起来并且给开发板上电，然后打开 CCS 软件，选择 View->Target Configurations 窗口，然后在该窗口内右键菜单新建一个 Target Configuration，如下图：



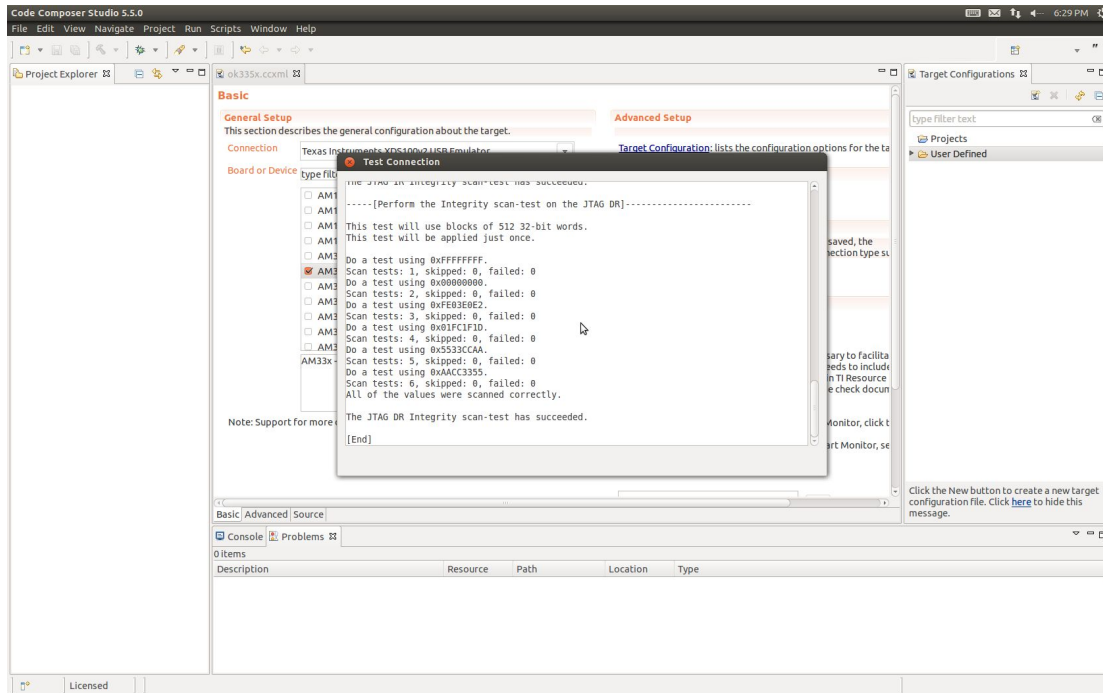
随便起个名字，比如 ok335x.ccxml，如下图：



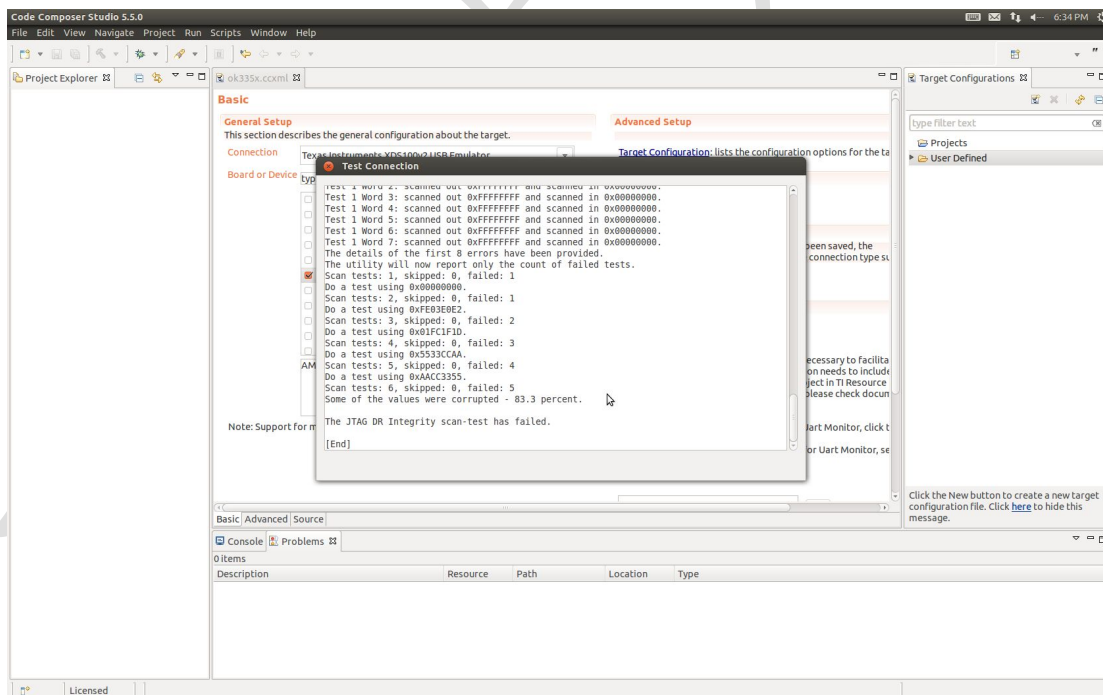
新建后将出现如下界面，在 1 处选择 XDS100V2 USB Emulator（如果此处无法选择 XDS100V2，则说明仿真器没有正确连接，使用其他仿真器的用户请参考其他说明），然后在 2 处选择 AM3354(OK335xD 使用的 CPU)，然后点击 3 处的 save 按钮进行保存：



之后点击 4 处的 Test Connection 按钮进行连通性测试，出现如下图的界面则说明测试成功，亦说明仿真器的调试环境搭建成功，后续一般不会遇到问题了。

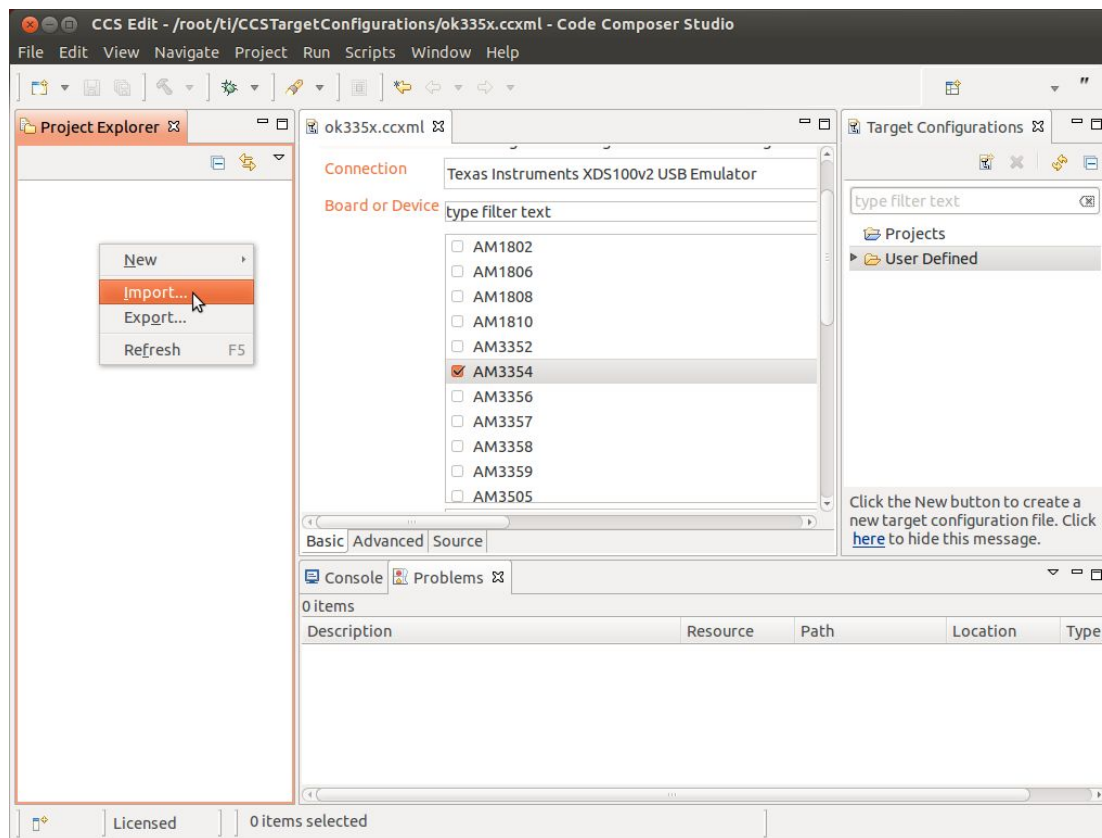


如果出现类似如下界面，或出现 failed 项则说明测试失败，请根据错误码进行检查。

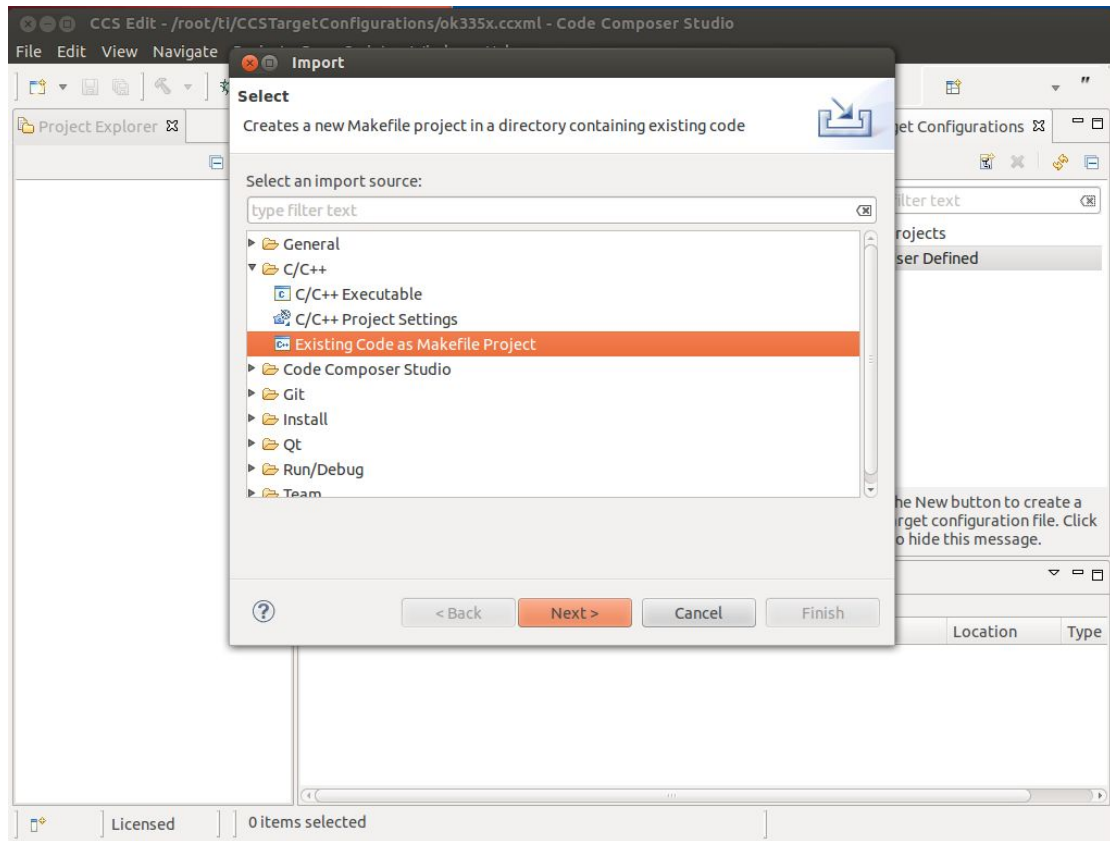


## F1.4 调试 SPL 代码

导入 uboot 工程代码如下图，在 Project Explorer 窗口内右键菜单 Import:

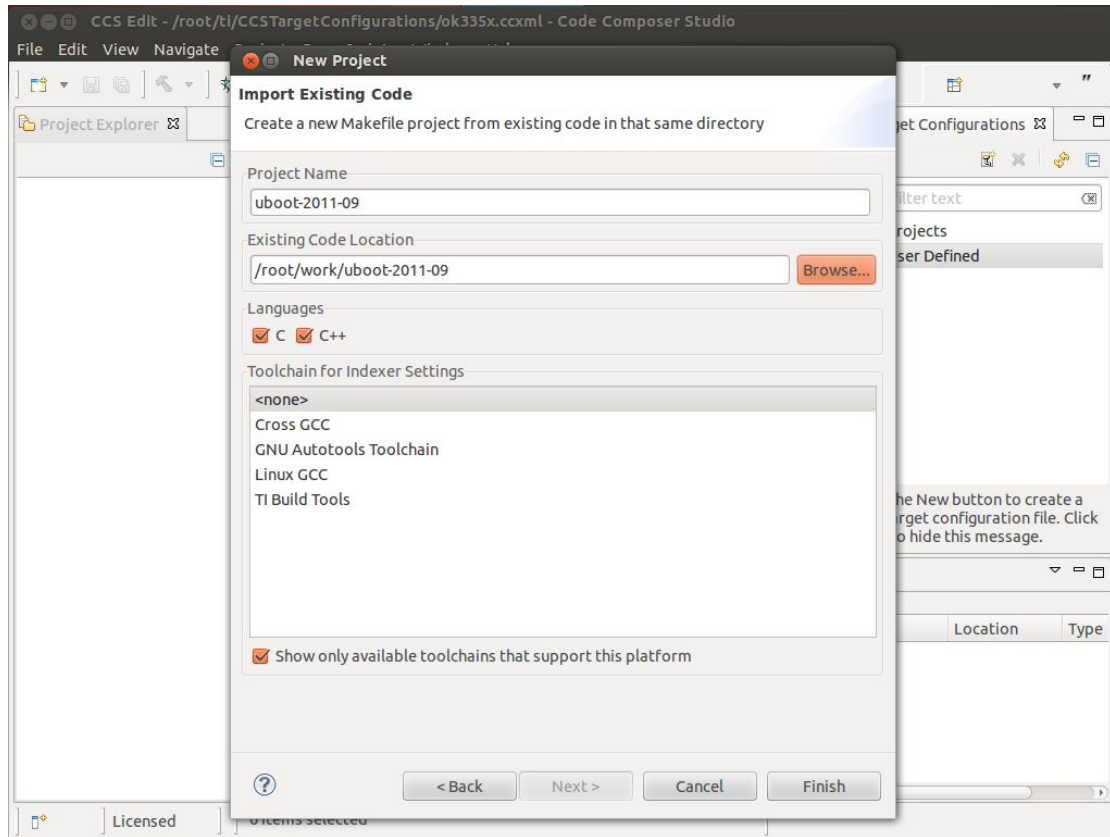


弹出如下界面，选择 Existing Code as Makefile Project:

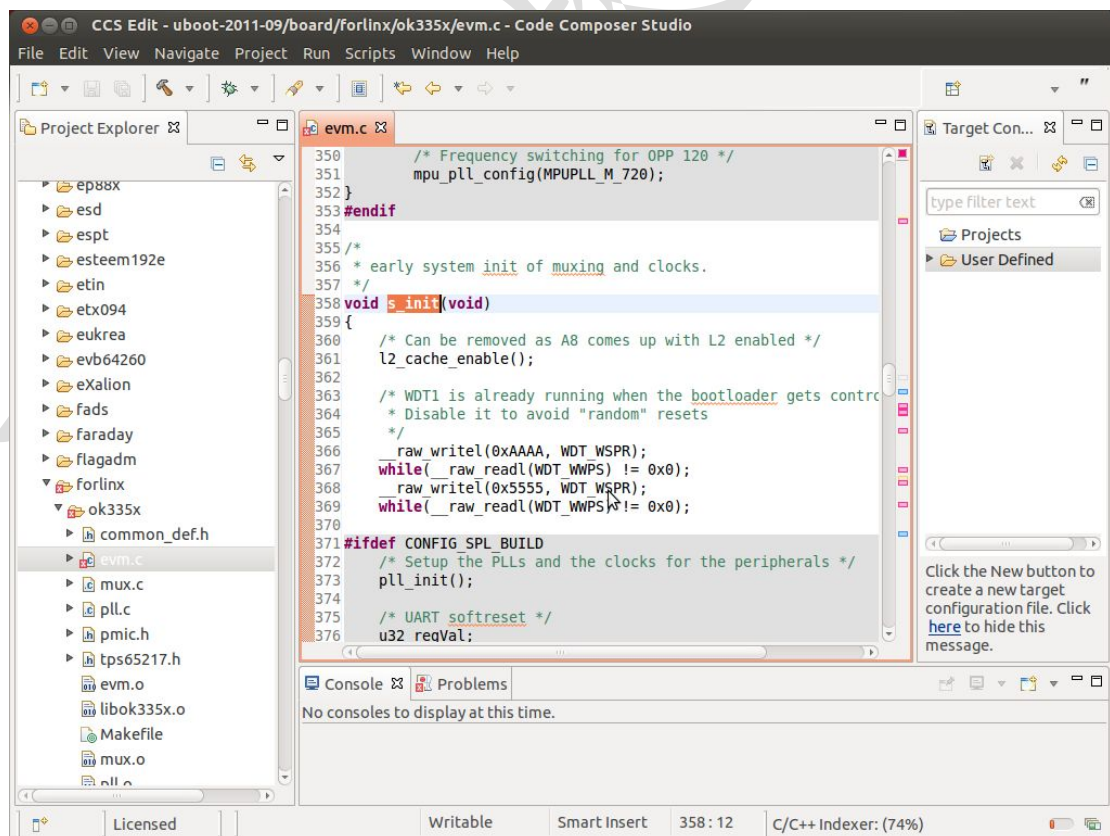


浏览选择 uboot 代码根目录路径，然后单击 Finish 如下图：



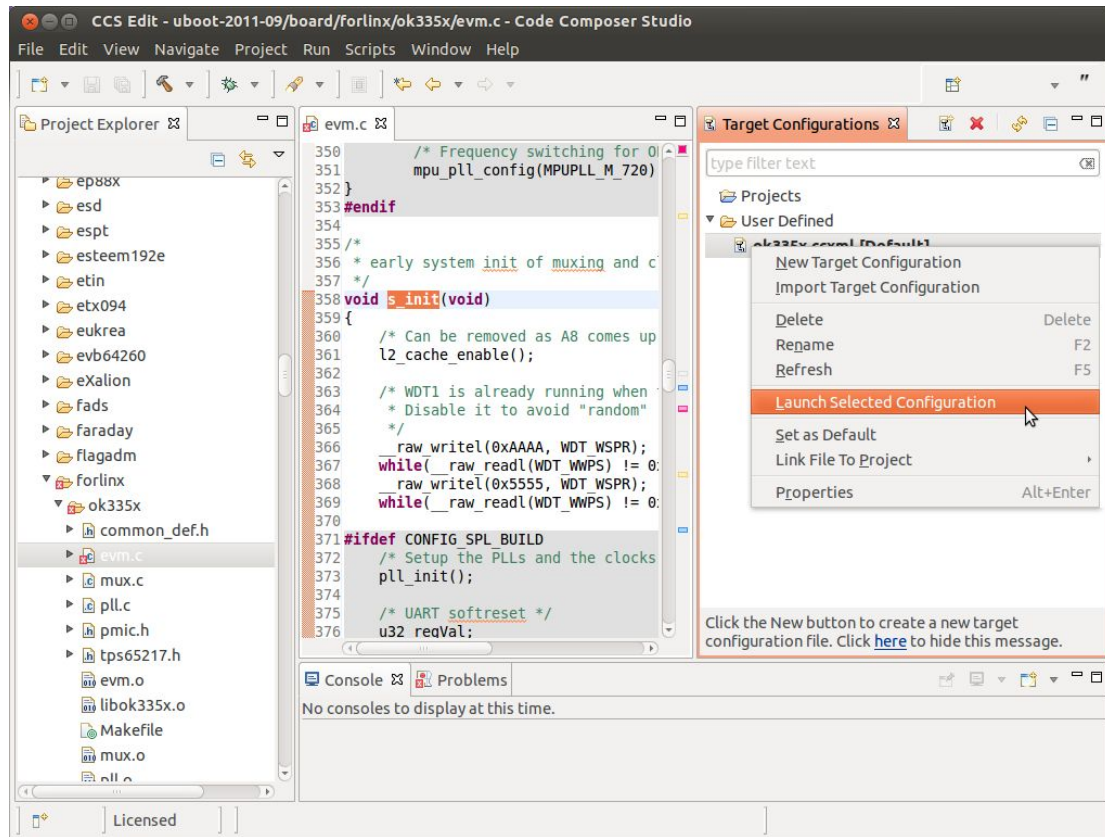


之后即可浏览 uboot 代码（对于浏览过程中代码洞察报的各种错误可以视而不见，这并不影响我们调试），如下图：

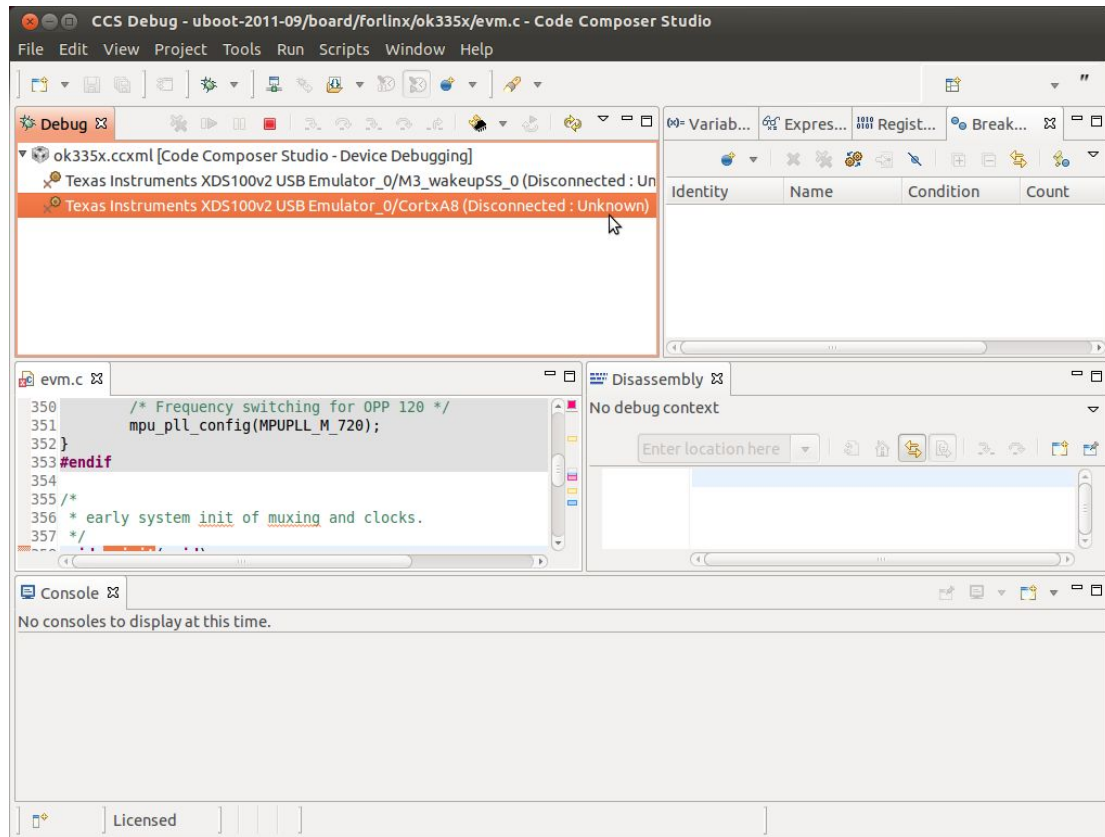




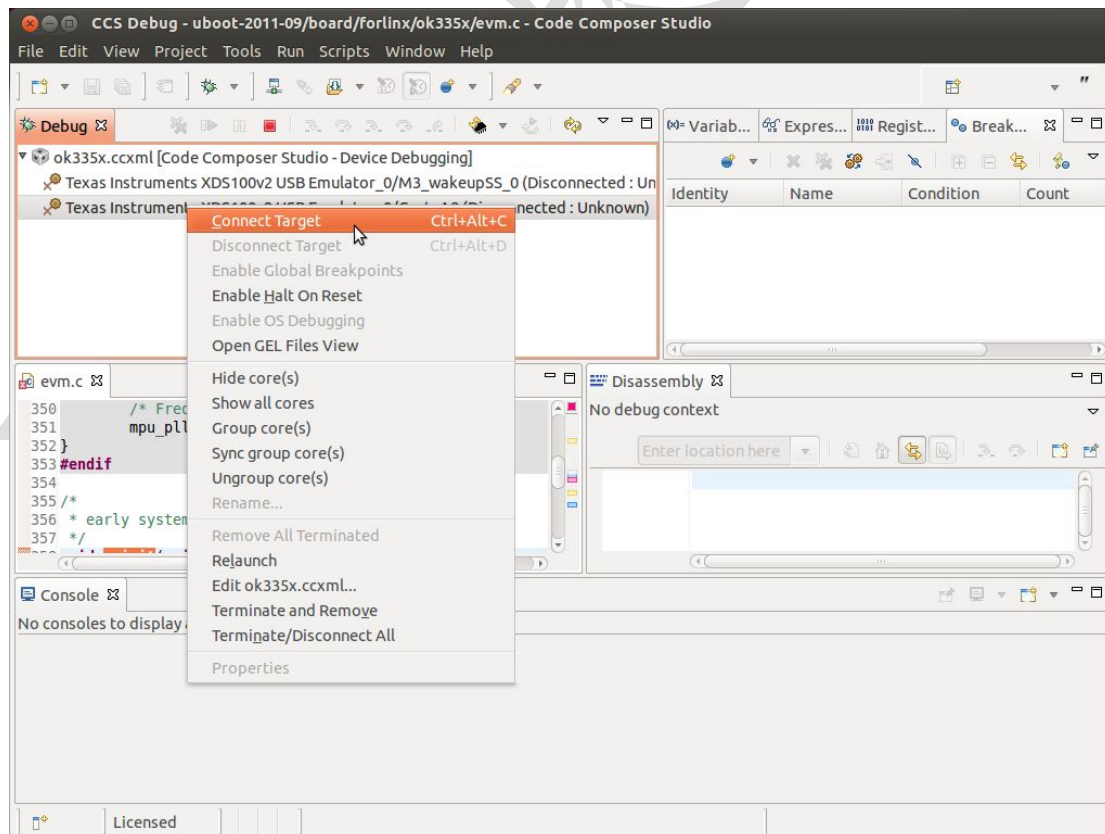
建立同开发板的连接，右键我们刚刚创建的 ok335x.ccsml，选择 Launch Selected Configuration，如下图：



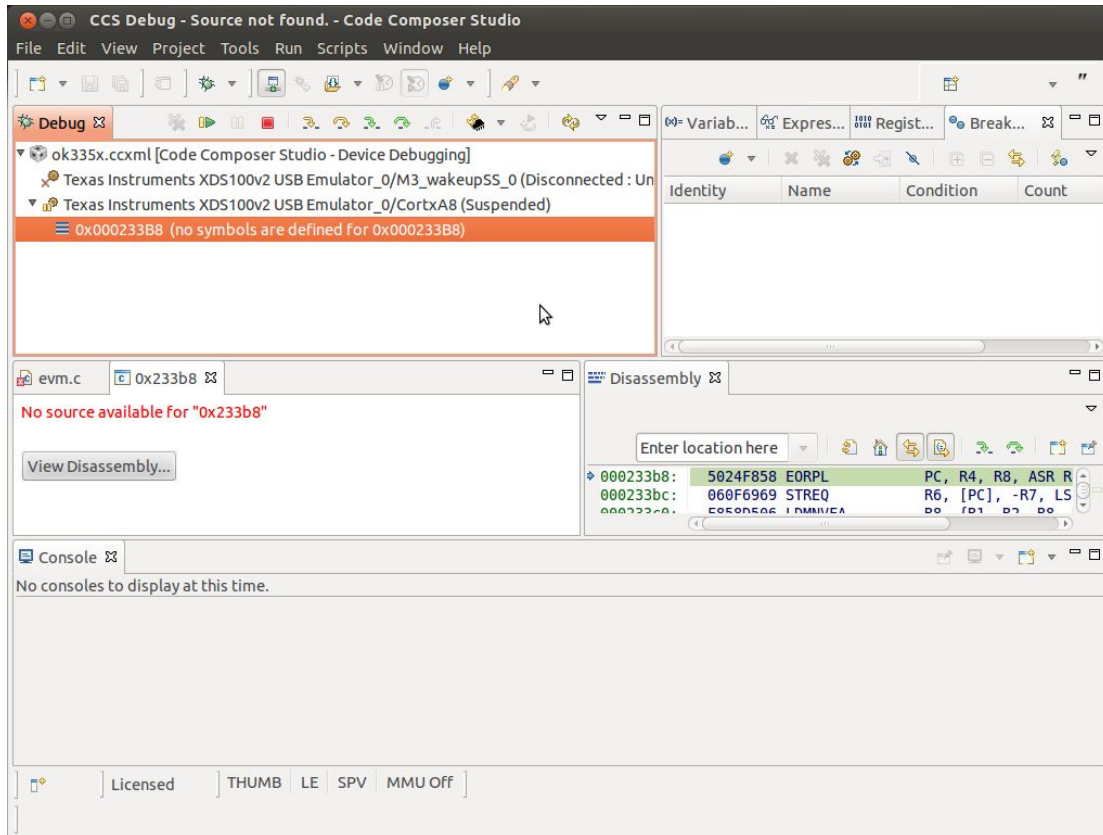
之后出现如下界面：



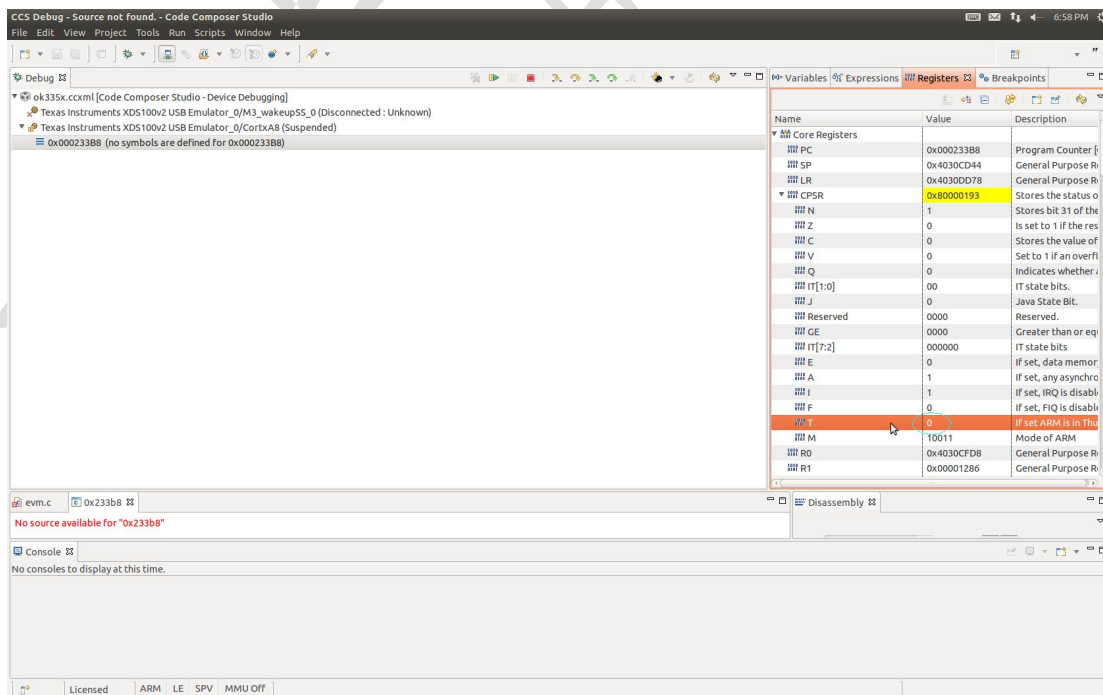
然后在 ContextA8 项上右键菜单，选择 Connect Target，如下图：



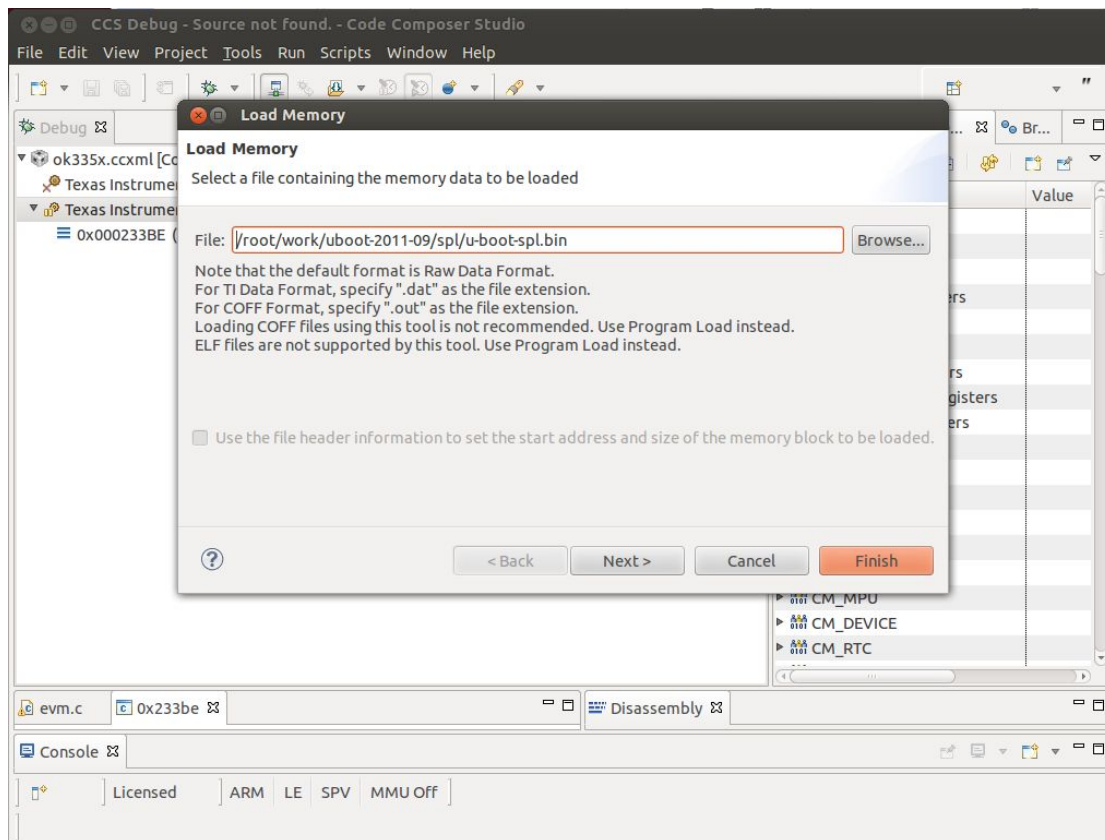
之后出现以下界面，表明已连接到开发板 CortexA8 核，此时已可以通过 Registers 窗口查看和修改寄存器的值：



修改 CPU 到 ARM 模式，修改寄存器 CPSR 的 T 位，键入 0 回车切换 CPU 到 ARM 模式：

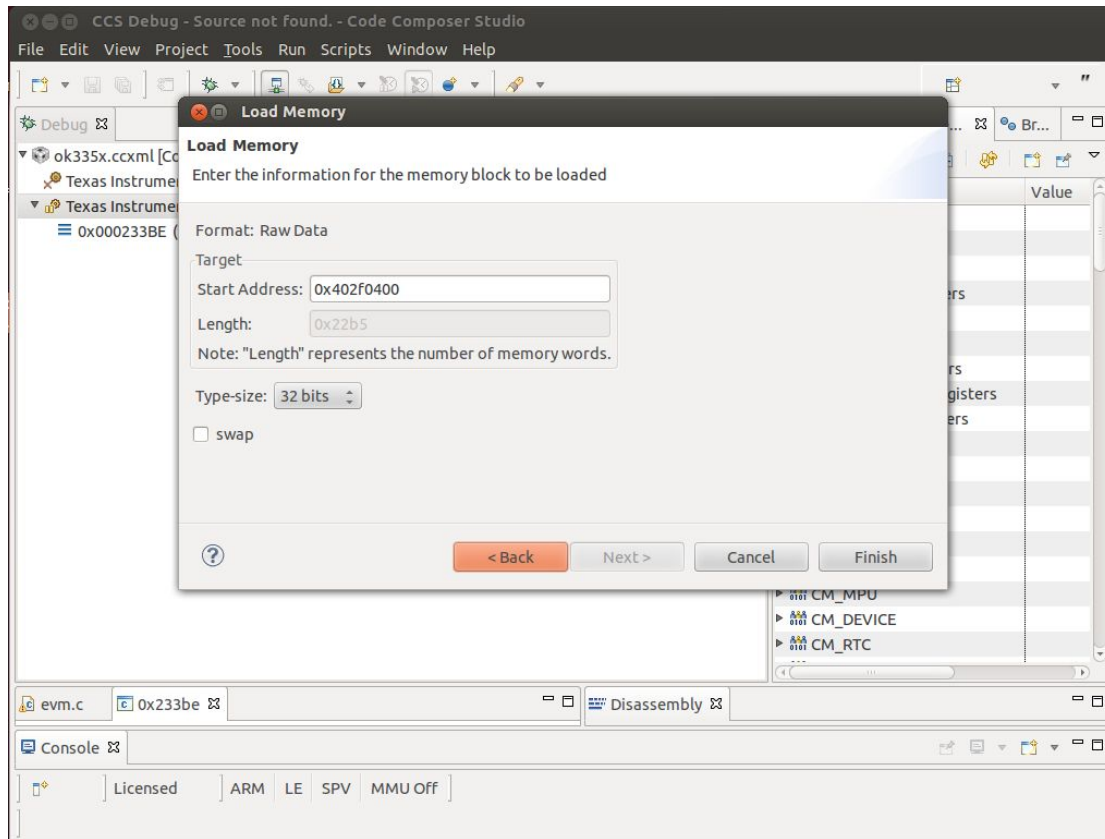


执行菜单 Tools->Load Memory 载入 u-boot-spl.bin (MLO 是该文件加了镜头头的版本) 到 CPU 内部 RAM, (注意切换文件类型到 ALL, 否则看不到 u-boot-spl.bin) 如下图:

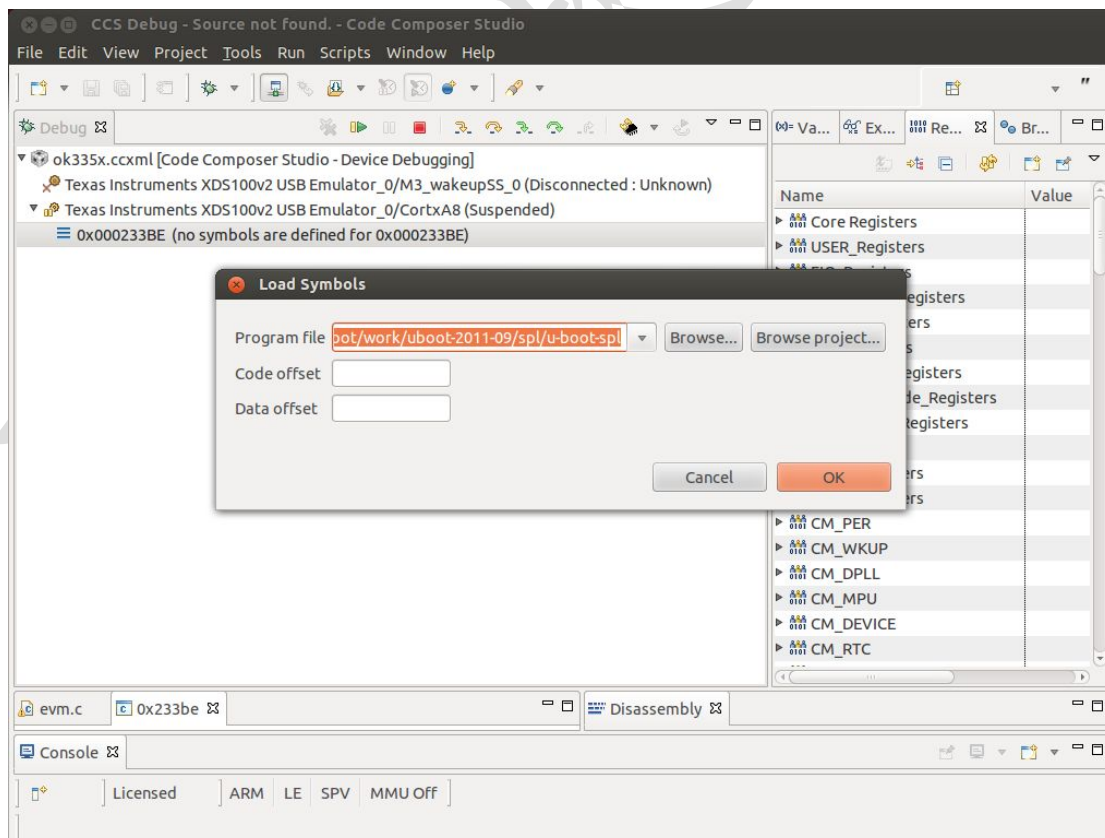


单击 next 后如下图, 填入要加载到的地址, 我们要加载到内部 RAM, 故地址为: 0x402f0400 并且切换 Type-Size 到 32 bits, 然后单击 Finish 即可。

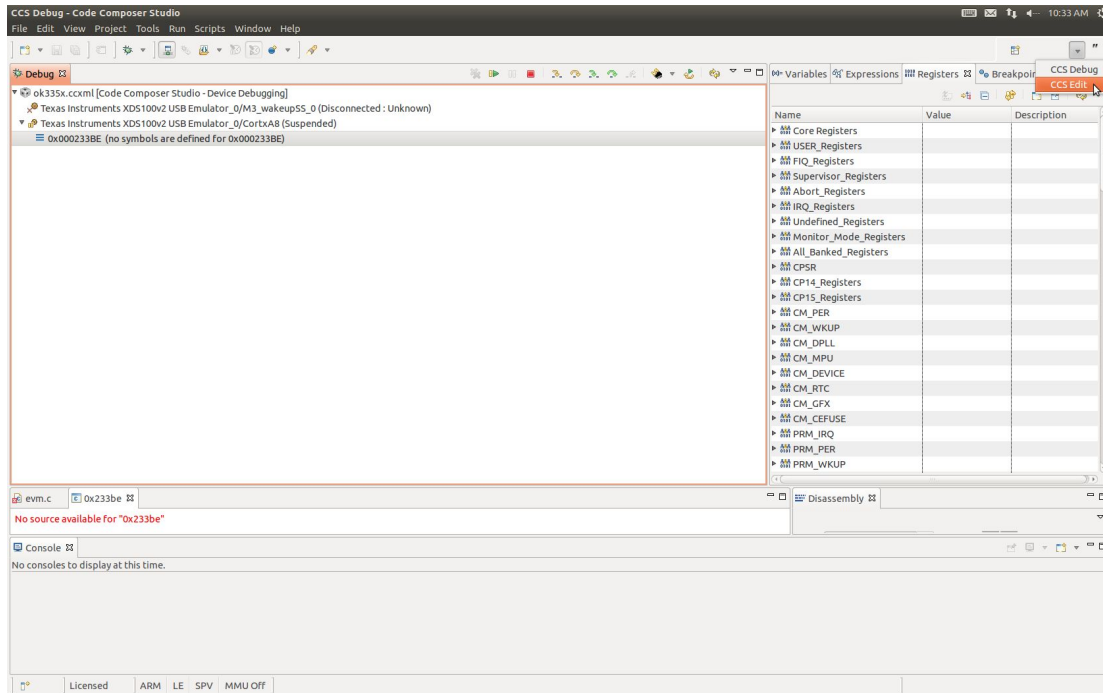




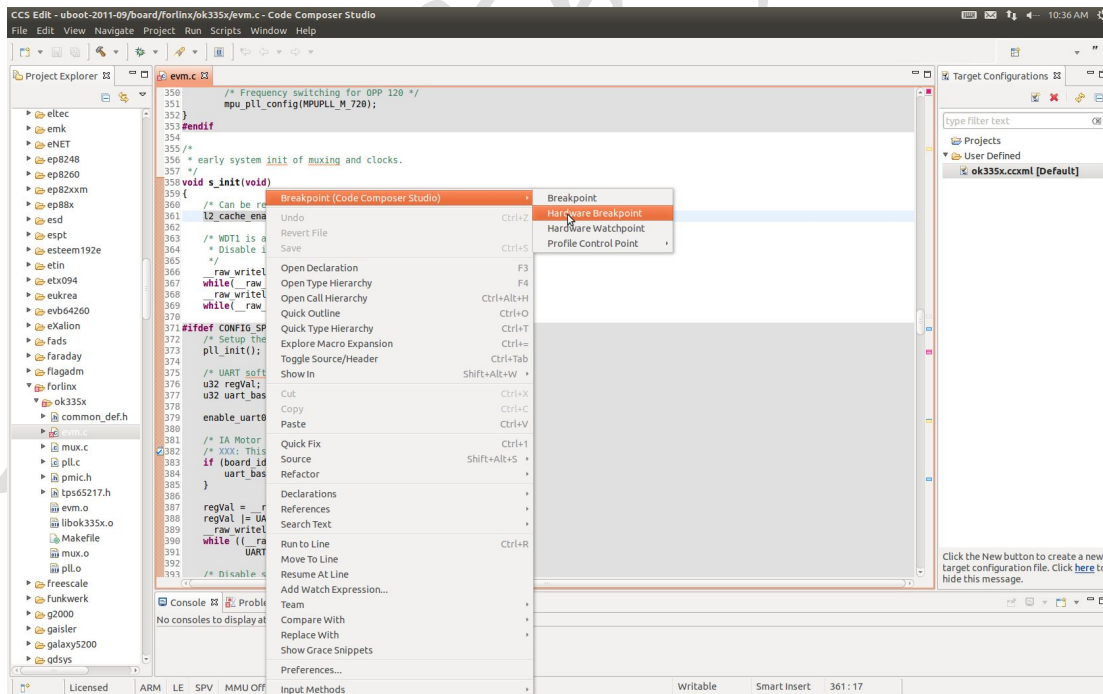
执行菜单 Run->Load->Load Symbols 加载带调试符号的 u-boot-spl(该文件为 u-boot-spl.bin 对应带调试信息版本) 到 CCS 以使其能建立源码和二进制的联系，如下图：



切换工作区到 CCS Edit，如下图：

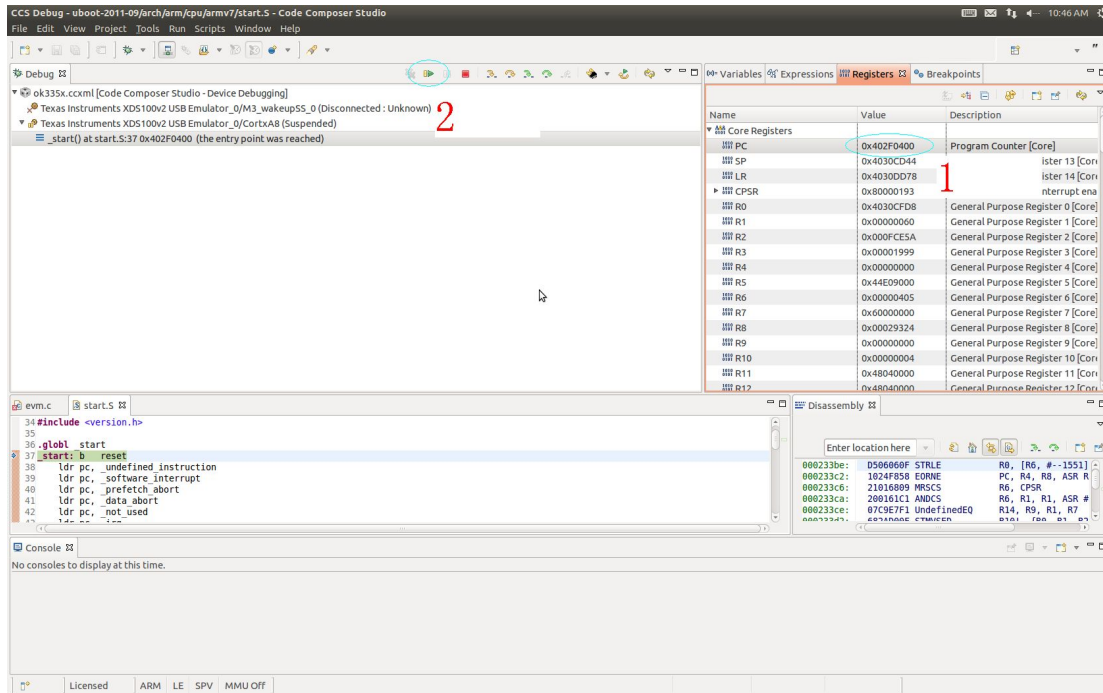


然后在源码某处右键菜单设置一个硬件断点（由于编译时优化，有些地方并不能设置断点，编译 uboot 时用户可以尝试去掉-O 选项），如下图：

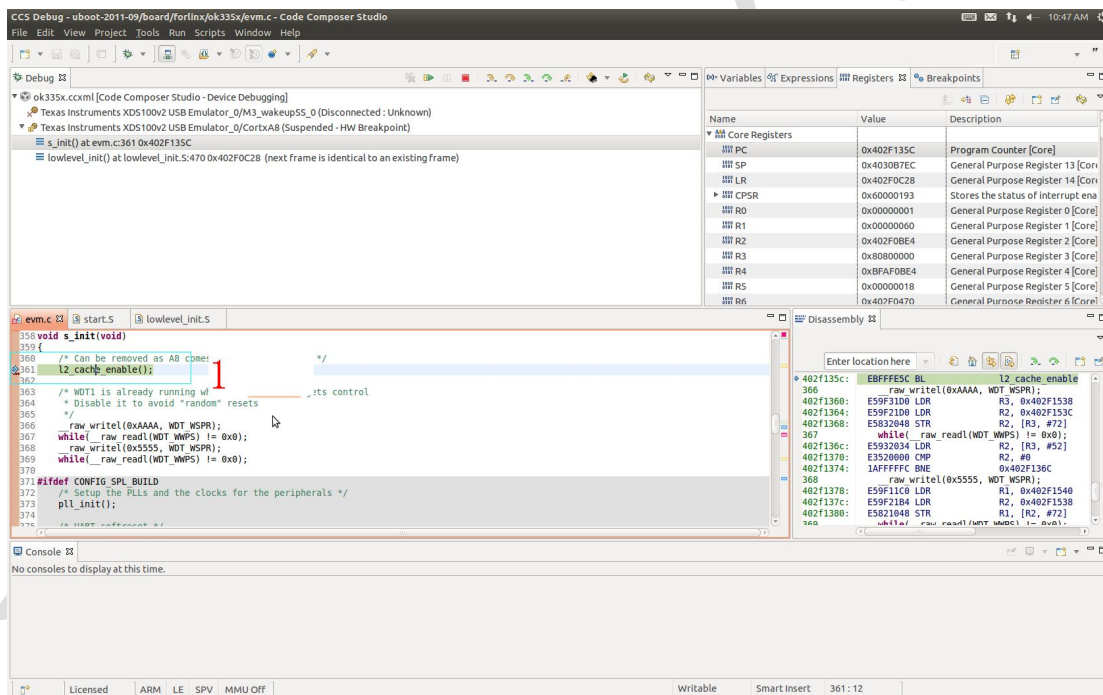


然后再切换工作区到 CCS Debug，设置 1 处 PC 寄存器指针为：0x402f0400，然后单击 2 处运行按钮，如下图：





会发现程序停留在我们设置的断点处（1 处），如下图：



之后用户可以进行内存，寄存器查看和修改，也可以进行单步调试等，Uboot 和 Linux 的调试方式与 spl 的调试方式类似，用户可以参照该文档进行调试。

## 附录 2：TFTP 服务搭建

### F2.1 安装服务器、客户端和守护进程

```
sudo apt-get install tftp-hpa tftpd-hpa xinetd
```

### F2.2 服务器配置

首先，在根目录下建一个tftpboot，并把属性改成任意用户可读写：

```
cd /  
sudo mkdir tftpboot  
sudo chmod 777 tftpboot
```

然后，进入目录 /etc/xinetd.d/，并在其中新建文件tftp，把指定的内容加入到tftp文件中：

```
cd /etc/xinetd.d/  
sudo vim tftp  
添加以下内容到tftp文件  
service tftp  
{  
    disable = no 138  
    socket_type = dgram  
    protocol = udp  
    wait = yes  
    user = root  
    server = /usr/sbin/in.tftpd  
    server_args = -s /tftpboot -c  
    per_source = 11  
    cps = 100 2  
}
```

最后，修改配置文件/etc/default/tftpd-hpa，修改为

```
TFTP_USERNAME="tftp"  
TFTP_DIRECTORY="/tftpboot"  
TFTP_ADDRESS="0.0.0.0:69"  
TFTP_OPTIONS="--secure -l -c -s"
```

注意将” TFTP\_DIRECTORY “改为新建tftpboot目录所在的路径

### F2.3 重新启动服务

```
sudo /etc/init.d/xinetd reload  
sudo /etc/init.d/xinetd restart
```

## F2.4 测试服务器

测试一下，在/tftpboot文件夹下新建立一个文件

```
touch abc
```

进入另外一个文件夹

```
tftp 192.168.1.230 (192.168.1.230为本机IP)
```

```
tftp> get abc
```

如果可以下载说明服务器已经安装成功，将开发板同 PC 通过网线进行连接后即可使用 tftp 下载文件。

## F2.5 Uboot 使用 tftp

1、用网线连接主机和开发板的网口并将编译好的内核镜像uImage复制到tftpboot目录下

2、进入到uboot阶段，修改环境变量并保存：

```
setenv serverip 192.168.1.210(192.168.1.210为tftp服务器的ip)
```

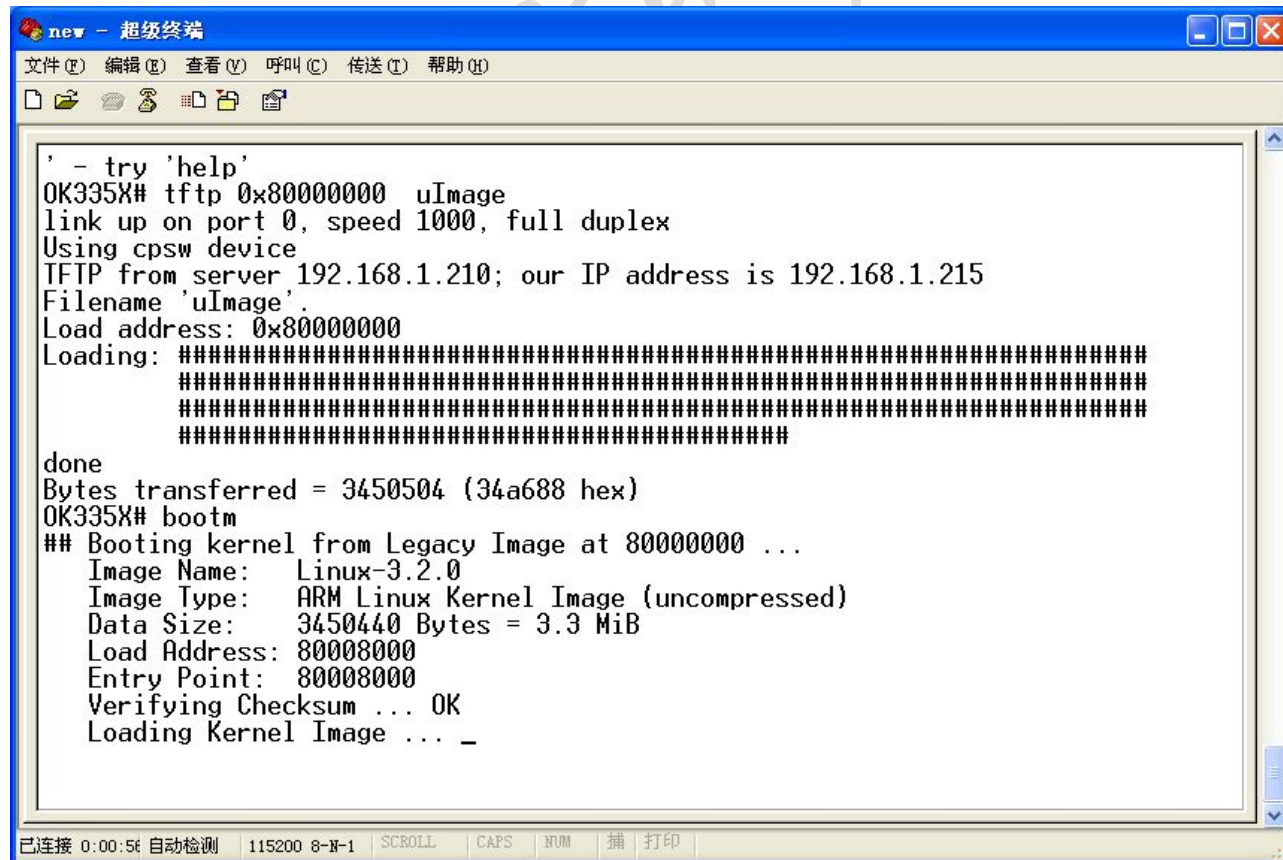
```
setenv ipaddr 192.168.1.215(192.168.1.215为开发板ip和服务器的ip应该在一个网段中)
```

```
saveenv
```

3、执行如下命令下载内核镜像并启动内核，终端中显示如下：

```
tftp 0x80000000 uImage
```

```
bootm
```



```
new - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

' - try 'help'
OK335X# tftp 0x80000000 uImage
link up on port 0, speed 1000, full duplex
Using cpsw device
TFTP from server 192.168.1.210; our IP address is 192.168.1.215
Filename 'uImage'.
Load address: 0x80000000
Loading: #####
#####
#####
#####
done
Bytes transferred = 3450504 (34a688 hex)
OK335X# bootm
## Booting kernel from Legacy Image at 80000000 ...
   Image Name:   Linux-3.2.0
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    3450440 Bytes = 3.3 MiB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
   Loading Kernel Image ... _
```

## F2.6 Linux 使用 tftp

开发板上Linux附带的tftp是精简版的，使用方式如下：

下载文件：

```
tftp xxx.xxx.xxx.xxx -r filename -g
```

上传文件：

```
tftp xxx.xxx.xxx.xxx -l filename -p
```

选项解释：

-r == remote

-g == get

-l == local

-p == put