

|  |
| --- |
| Linux网络测试程序开发v1.0 |
| 基于TI AM335x核心平台 |
|  |

、

免责声明

本文档是作者对GOEMBED 产品进行实际操作和测试后，自我心得总结。建议读者具备一定的计算机基础和基本软件操作能力，如在操作过程中，遇到疑问和错误，欢迎加QQ群(462424566)交流，或发厂商技术支持邮箱进行咨询: support@goembed.com

操作环境配套说明：

|  |  |
| --- | --- |
| 硬件 | 详细介绍链接 |
| SBC3358-B1A单板机 | [c:\users\administrator\appdata\roaming\360se6\User Data\temp\1411389502416719.jpg](http://www.goembed.com/index.php/Products/detail/tpid/25) |
| 串口调试器：COM10U | [c:\users\administrator\appdata\roaming\360se6\User Data\temp\1408786855283579.jpg](http://goembed.com/index.php/Products/detail/tpid/16) |

|  |  |
| --- | --- |
| 软件 | 详细介绍链接 |
| Ubuntu版本：12.04 LTS（64bit） | [http://www.ubuntu.org.cn/download/desktop](http://www.ubuntu.org.cn/download/desktop%20) |
| Linux版本：3.11.0-15-generic |
| gcc版本：4.6.3 |

### SBC3358-B1A单板机软件特性

#### 1、BootLoader版本：u-boot-2013.01.01

#### 2、内核版本：linux-3.2.0

* LCD驱动
* LCD背光驱动
* 电阻式触摸屏驱动
* VGA驱动
* HSMMC/SD/MMC/SDIO驱动
* IIC驱动
* SPI驱动
* 音频驱动
* DMA驱动
* RTC实时时钟驱动
* 电源管理
* USB HOST/DEVICE驱动
* USB OTG驱动
* DEBUG驱动
* 以太网驱动
* TF卡驱动
* CAN驱动
* 串口驱动
* WG驱动

#### 3、交叉工具链：arm-linux-gnueabihf-gcc

### SBC3358-B1A单板机资源分配特性

#### emmc空间分配

|  |  |  |
| --- | --- | --- |
| Partition | Size | Description |
| BootLoader | **200MB** | **FAT32格式分区** |
| rootfs | **约1500MB** | **EXT3格式分区** |

### 一、准备工作

1、参考《TI AM335x 搭建Linux开发环境 v1.0.docx》和《TI AM335x Linux系统编译 v1.0.docx》把开发环境搭建好。

2、为了方便阅读和修改代码，在这里我使用的是Source Insight(一个代码编辑工具)对代码进行修改。用户可以直接在终端使用VI编辑器编辑代码，结果是一样的，这里是为了阅读方便。

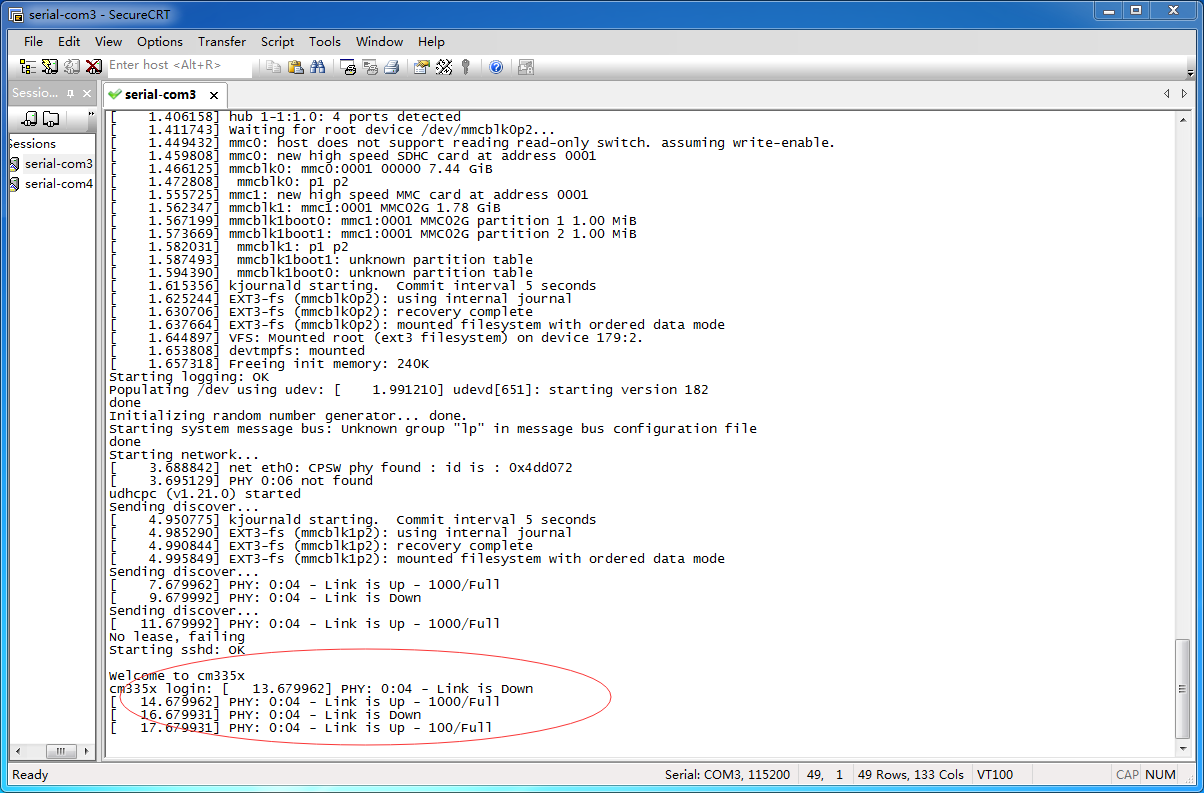
注：应用程序编译要采用arm-2009q1交叉编译。（后续补上）

### 二、硬件连接

1、将SBC3358-B1A的千兆网口和路由器的其中一个LAN口用网线连接起来。

2、将电脑网口和路由器的其中一个LAN口用网线连接起来。

3、连上网线开机后会有以下提示：



### 三、IP设置

#### 1、设置SBC3358-B1A的IP地址

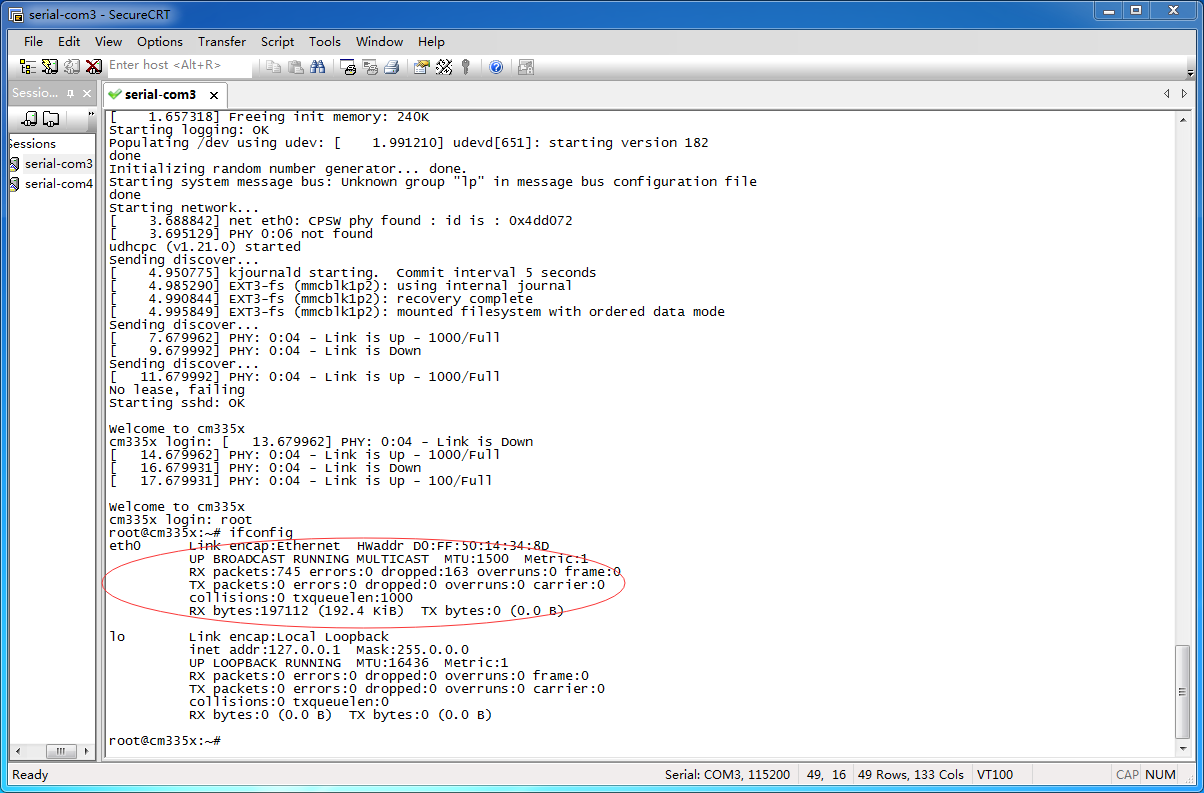
（1）输入”root”登录（不用密码）:

先查看一下自己的IP地址：

* 命令：

ifconfig

开发板打印以下信息：

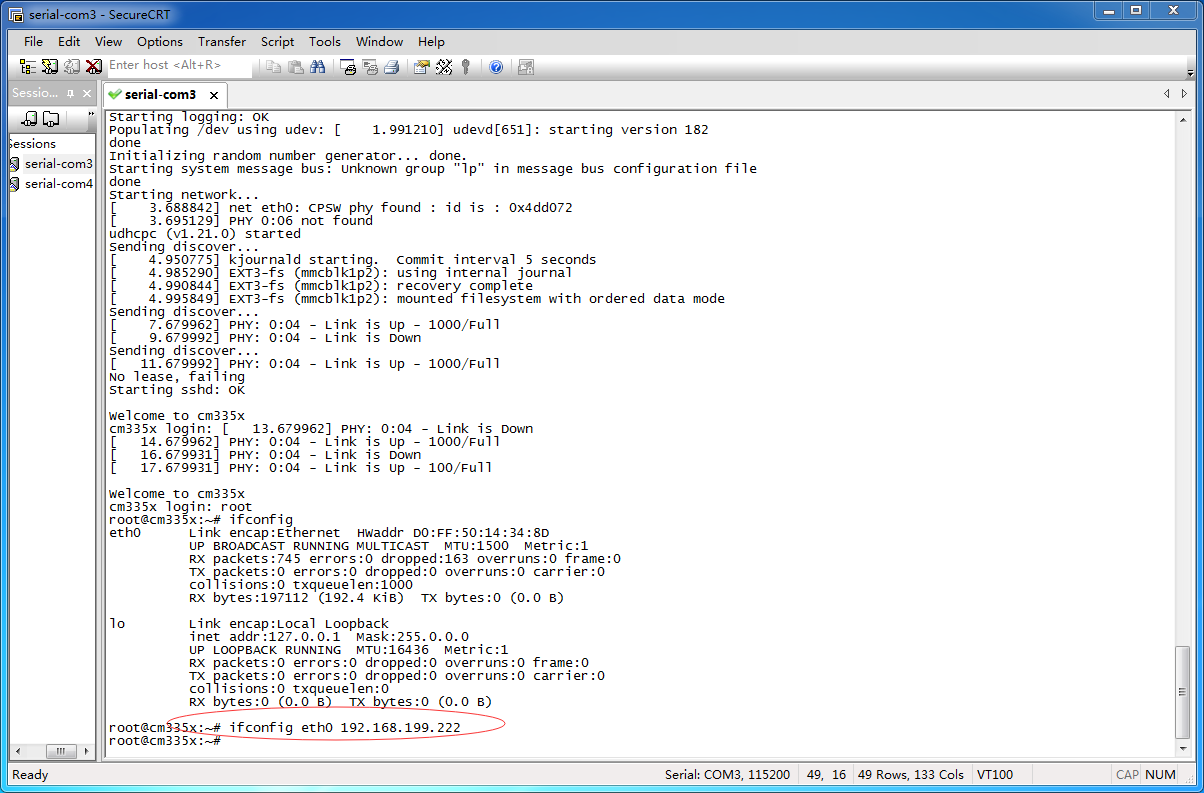


可以看到无默认静态IP地址，下面我们设置一个静态IP：

* 命令：

ifconfig eth0 192.168.199.222

如下：

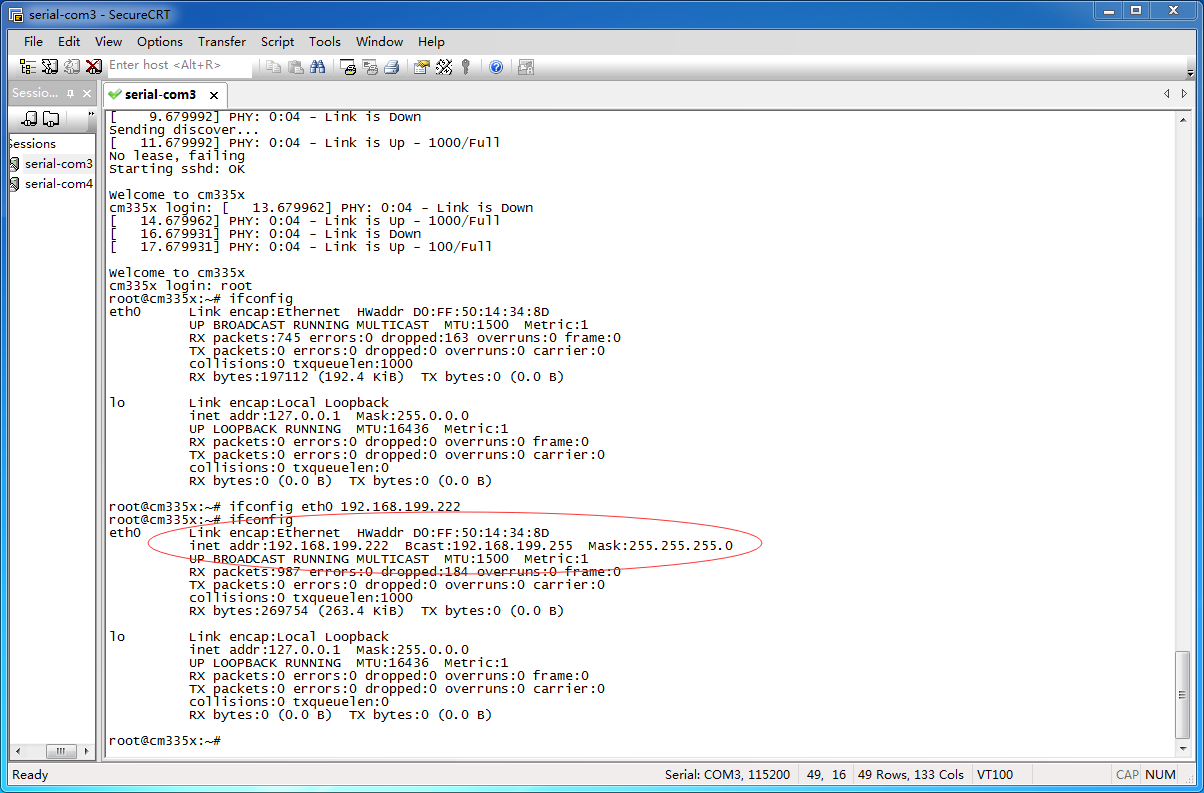


然后

* 命令：

ifconfig

查看一下目前已配置的IP：

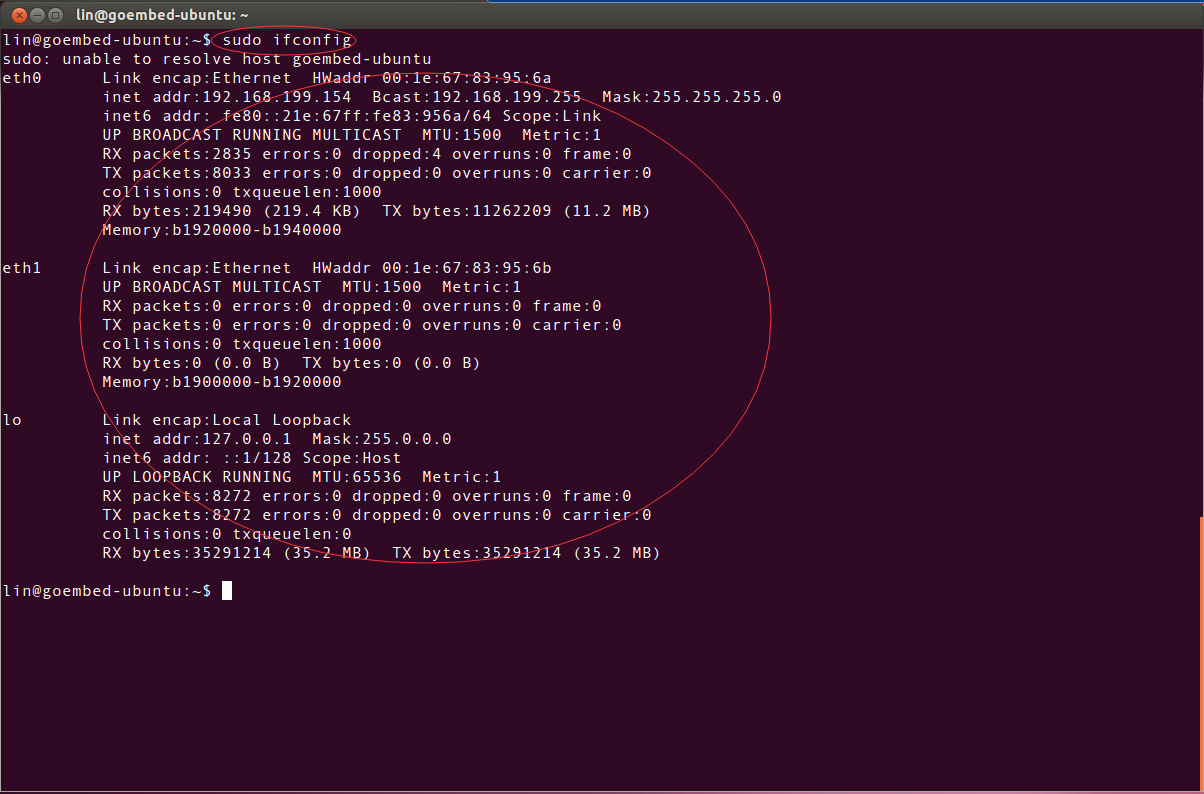


可以看到，开发板此时的静态IP为：192.168.199.154，广播地址为：192.168.199.255，子网掩码为255.255.255.0，为了使我们的开发板和服务器能正常通讯，我们需要将服务器的IP设置在同一网段中。

注：开发板上的IP设置将在测试代码中自动完成，不需要手动设置，这里只是为了在执行测试程序前先测试网络情况以保证实验的正确性。

#### 2、设置服务器的IP地址

（1）、查看服务器的IP:

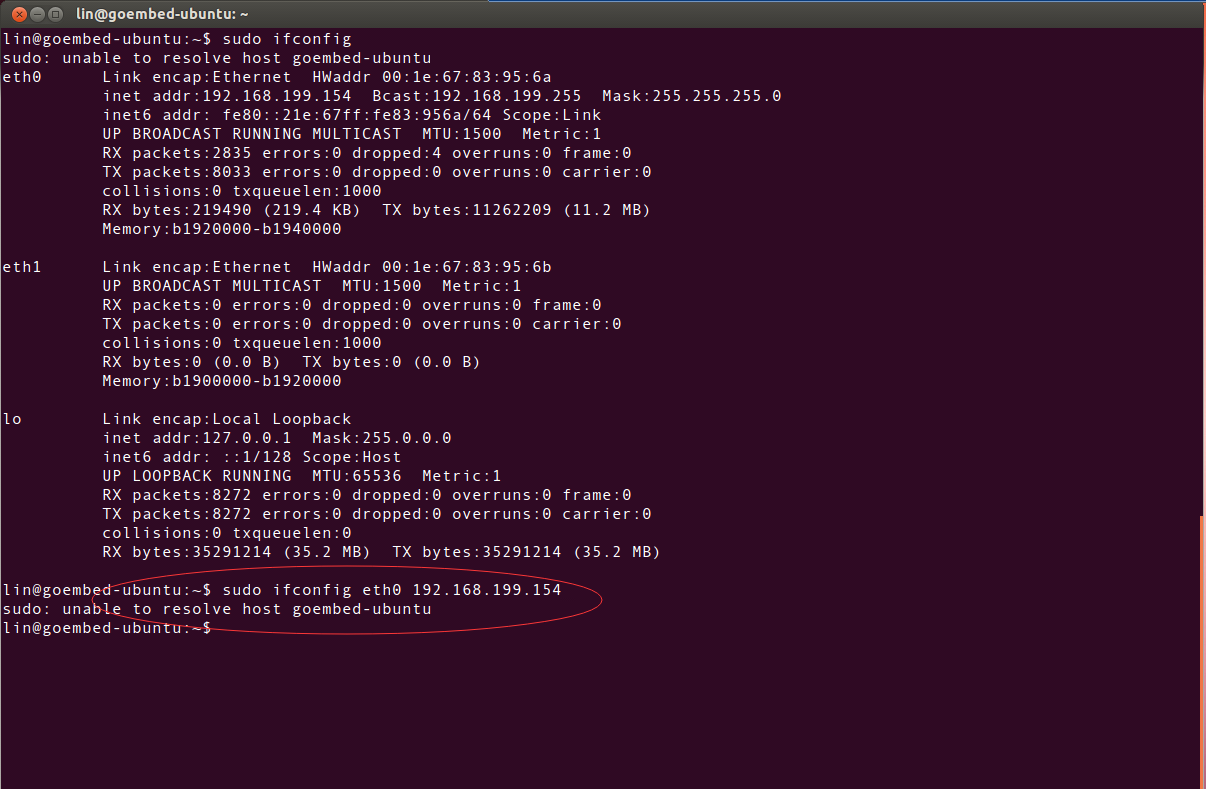


可见服务器已经设置好了IP，为了演示我再设置一遍。

（2）、设置服务器的IP：

* 命令：

sudo ifconfig eth0 192.168.199.154



查看设置结果：

* 命令：

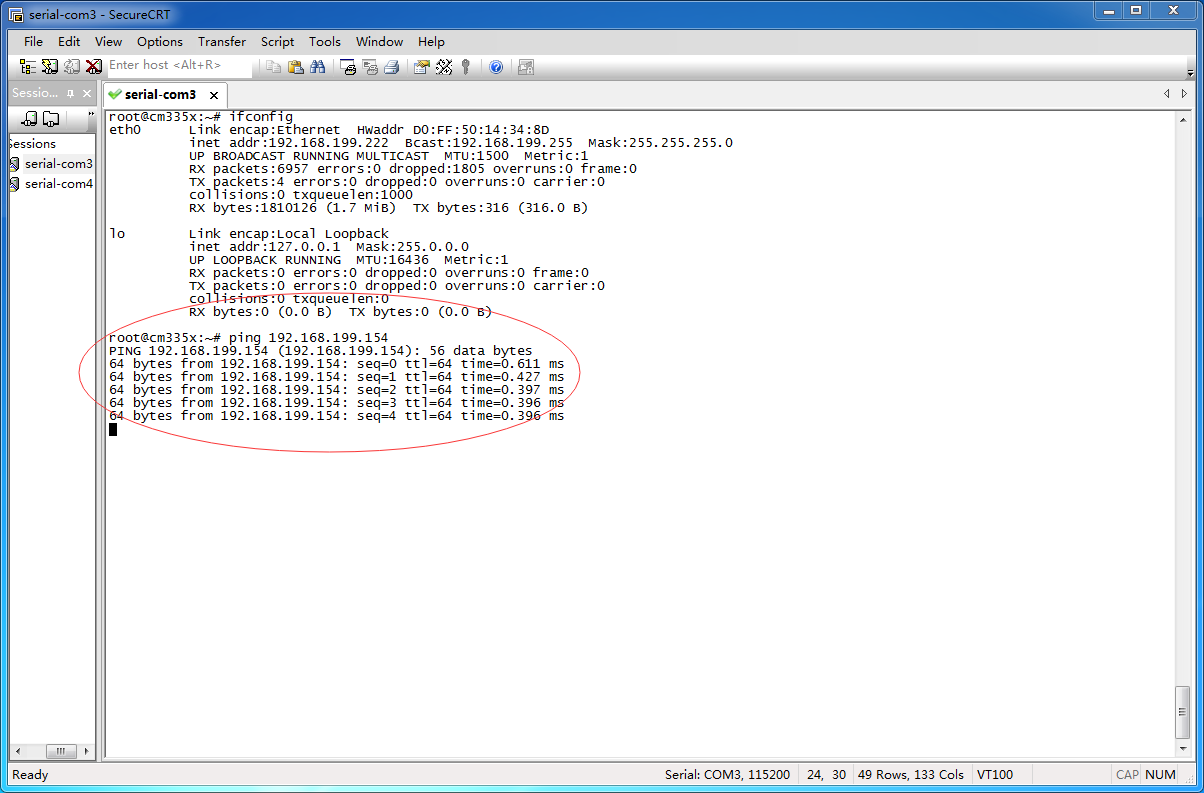
sudo ifconfig



#### 3测试网络

* 在开发板上输入：
* 命令：

ping 192.168.199.154

测试结果如下：

* 在服务器上输入：
* 命令：

ping 192.168.199.222

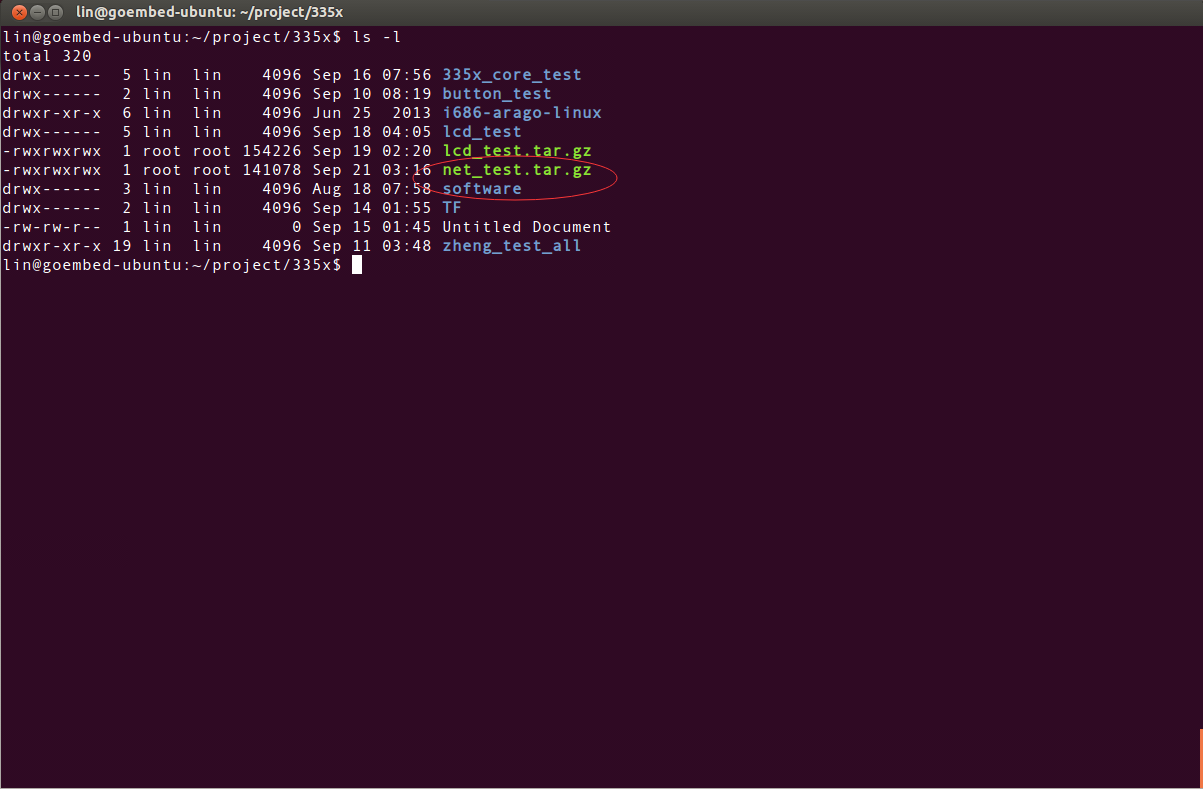
测试结果如下图：

从以上结果来看，开发板和服务器的网络已通。

### 四、分析测试程序

#### 1、解压源码包

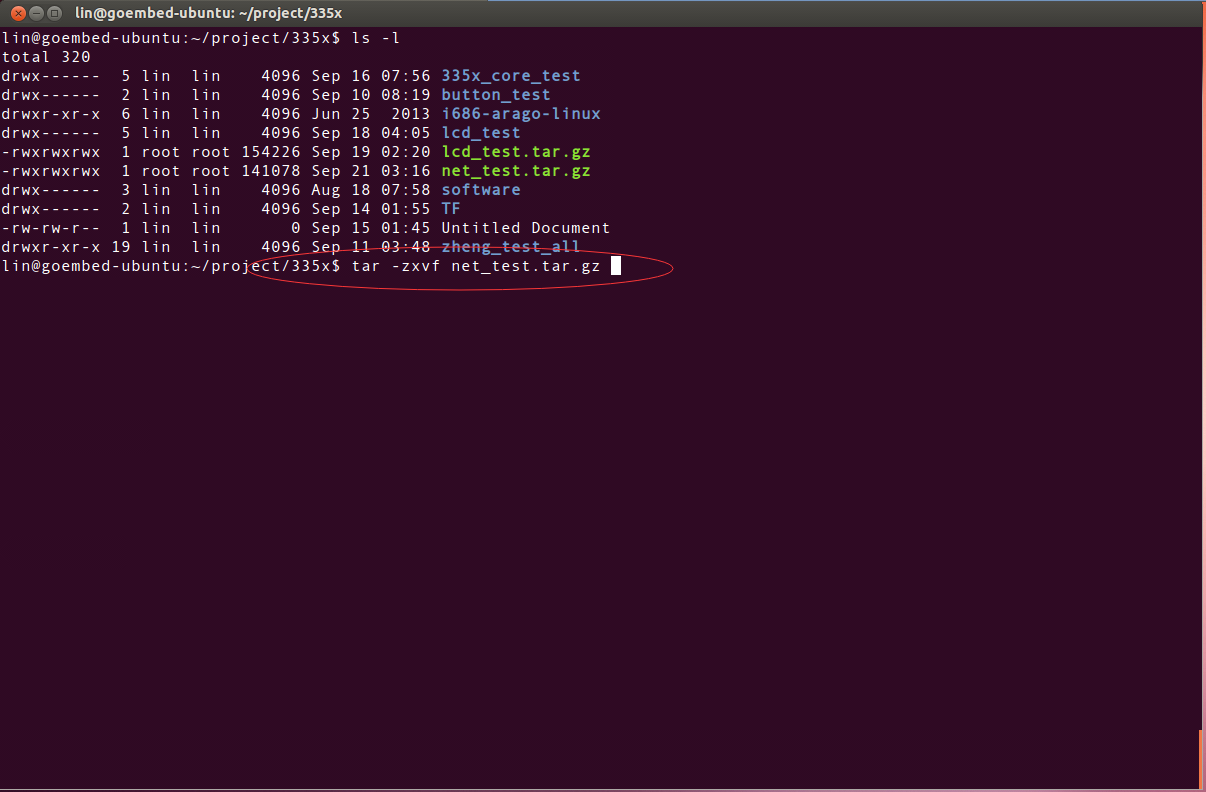
（1）、找到源码包：



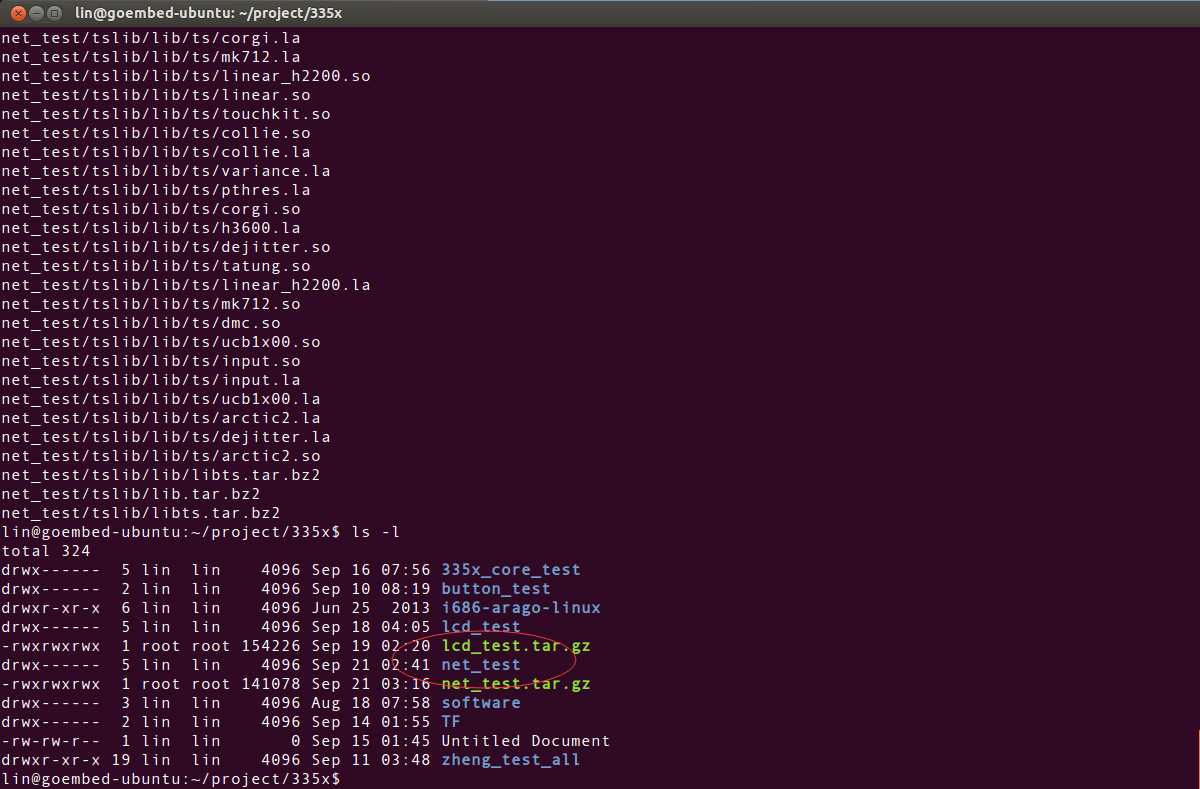
（2）、用以下命令解压：

* 命令：

tar –zxvf net\_test.tar.gz



解压完成后出现net\_test源码包文件夹：



#### 2、分析Makefile文件

最重要的几个参数如下：

* “NAME：net\_test\_out”：指的是生成的可执行测试文件名
* “CROSS= /usr/local/toolchain/arm-2009q1/bin/arm-none-linux-gnueabi-”：指使用的是arm-2009ql交叉编译器
* “SRCDIRS= net\_test”指的是要编译的文件夹，这里指的就是源码包解压出来的net\_test文件夹，也是最重要的文件夹。

#### 3、分析net\_test.c文件

我编写的这个代码比较短，直接贴上：

**#include <ctype.h>**

**#include <time.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <sys/types.h>**

**#include <sys/stat.h>**

**#include <fcntl.h>**

**#include <sys/mman.h>**

**#include <linux/fb.h>**

**#include <unistd.h>**

**#include <termios.h>**

**#include <string.h>**

**#include <linux/kd.h>**

**#include <sys/ioctl.h>**

**#include<sys/stat.h>**

**#define CMD\_IFCONFIG "ifconfig eth0 192.168.199.222 1>/tmp/net 2>/tmp/net"**

**#define CMD\_PING "ping -c 2 192.168.199.154 1>/tmp/net 2>/tmp/net"**

**#define CMD\_GREP "cat /tmp/net | grep '2 packets transmitted, 2 packets received, 0% packet loss' 1>/tmp/net\_info 2>/tmp/net\_info"**

**#define ERRO\_INFO "/tmp/net\_info"**

**#define CMD\_NET\_DOWN "ifconfig eth0 down"**

**#define CMD\_NET\_UP "ifconfig eth0 up"**

**int main(void)**

**{**

**struct stat stat\_buf;**

**system(CMD\_IFCONFIG);**

**printf("CMD\_IFCONFIG OK!\n\r");**

**system(CMD\_PING);**

**printf("CMD\_PING OK!\n\r");**

**system(CMD\_GREP);**

**printf("CMD\_GREP OK!\n\r");**

**if (stat(ERRO\_INFO,&stat\_buf)<0)**

**{**

**printf("Failed:buf<0\n\r");**

**return -1;**

**}**

**if (stat\_buf.st\_size <= 0)**

**{**

**printf("Failed:size<=0\n\r");**

**return -1;**

**}**

**printf("net is ok!\n\r");**

**return 0;**

**}**

这个测试程序看起来简单，其实还是运用了很多东西的。下面我给大家一一分析。

（1）分析宏定义

* #define CMD\_IFCONFIG "ifconfig eth0 192.168.199.222 1>/tmp/net 2>/tmp/net"

这个宏定义其实定义的是一条命令，“”中的内容就是具体的命令。这条命令的意思是在开发板上配置eth0网络设备的IP地址为192.168.199.222，并把标准正确输出到/tmp/net文件中，如果文件存在则覆盖，如果文件不存在则创建；把标准错误输出到/tmp/net文件中，如果文件存在则覆盖，如果文件不存在则创建。

* #define CMD\_PING "ping -c 2 192.168.199.154 1>/tmp/net 2>/tmp/net"

这个宏定义也是定义一条命令，“”中的内容就是具体的命令。这条命令的意思是在开发板上尝试ping 192.168.199.154这个节点，-c表示ping这个节点两次。并把标准正确输出到/tmp/net文件中，如果文件存在则覆盖，如果文件不存在则创建；把标准错误输出到/tmp/net文件中，如果文件存在则覆盖，如果文件不存在则创建。

* #define CMD\_GREP "cat /tmp/net | grep '2 packets transmitted, 2 packets received, 0% packet loss' 1>/tmp/net\_info 2>/tmp/net\_info"

这个宏定义也是定义一条命令，“”中的内容就是具体的命令。这条命令的意思是在开发板上搜索/tmp/net文件，如果该文件中包含“2 packets transmitted, 2 packets received, 0% packet loss”则将该内容打印到/tmp/net\_info文件中。如果没有则将/tmp/net\_info文件清空。

（2）分析main函数

其中最重要的函数是int system(const char \*string)函数，这个函数会调用fork()产生子进程， 由子进程来调用/bin/sh-c string来执行参数string字符串所代表的命令，这也是我上面定义那些宏命令的原因。此命令执行完后随即返回原调用的进程。

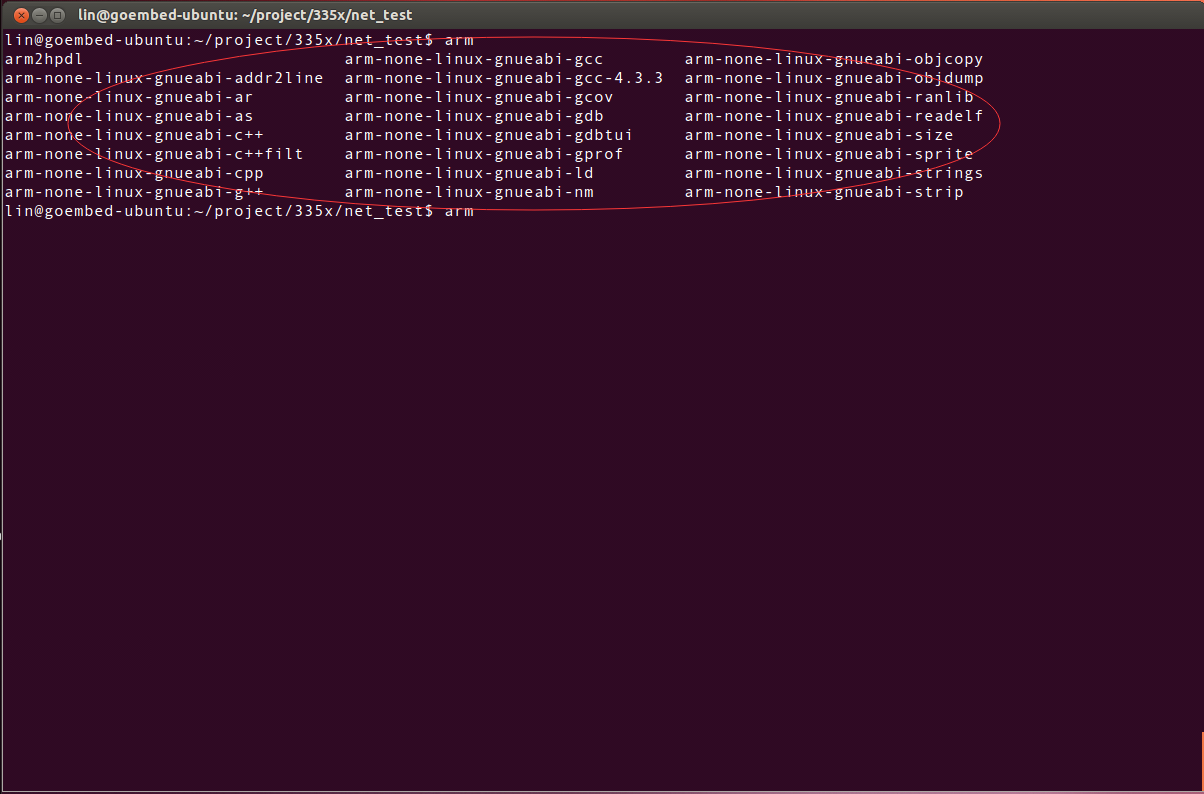
* 第一：我在system()函数中传入CMD\_IFCONFIG配置一下开发板的IP，并将输出信息记录到/tmp/net文件中，如果输出正确信息则说明开发板的网卡能访问和配置。
* 第二：我在system()函数中传入CMD\_PING测试一下当前网络连接情况，并将输出信息记录在/tmp/net文件中。最后我们在system()函数中传入CMD\_GREP分析前面两个测试步骤的结果。
* 第三：我调用stat()函数判断/tmp/net\_info文件文件的信息，这个函数会将net\_info文件的信息保存在stat\_buf结构体中，如果该函数操作不成功会返回-1，所以我加入if语句判断函数返回值，返回-1则认为整个测试失败并退出函数。如果该步骤操作成功，则说明/tmp/net\_info文件无异常。接下来我判断stat\_buf结构体中的st\_size成员，如果该成员小于等于0则说明/tmp/net\_info文件没有存放任何东西，结合前面执行的system(CMD\_GREP)我们可以知道如果执行成功则st\_size成员大于0，失败反之。

### 五、编译测试程序

注：应用程序编译要采用arm-2009q1交叉编译

前面我已经说过，我们的源码里面有一个Makefile文件，里面把交叉编译工具、链接库、输入文件夹、输出文件名等都设置好了，我们直接在当前目录下（前提已经安装arm-2009ql交叉编译工具）输入make即可。

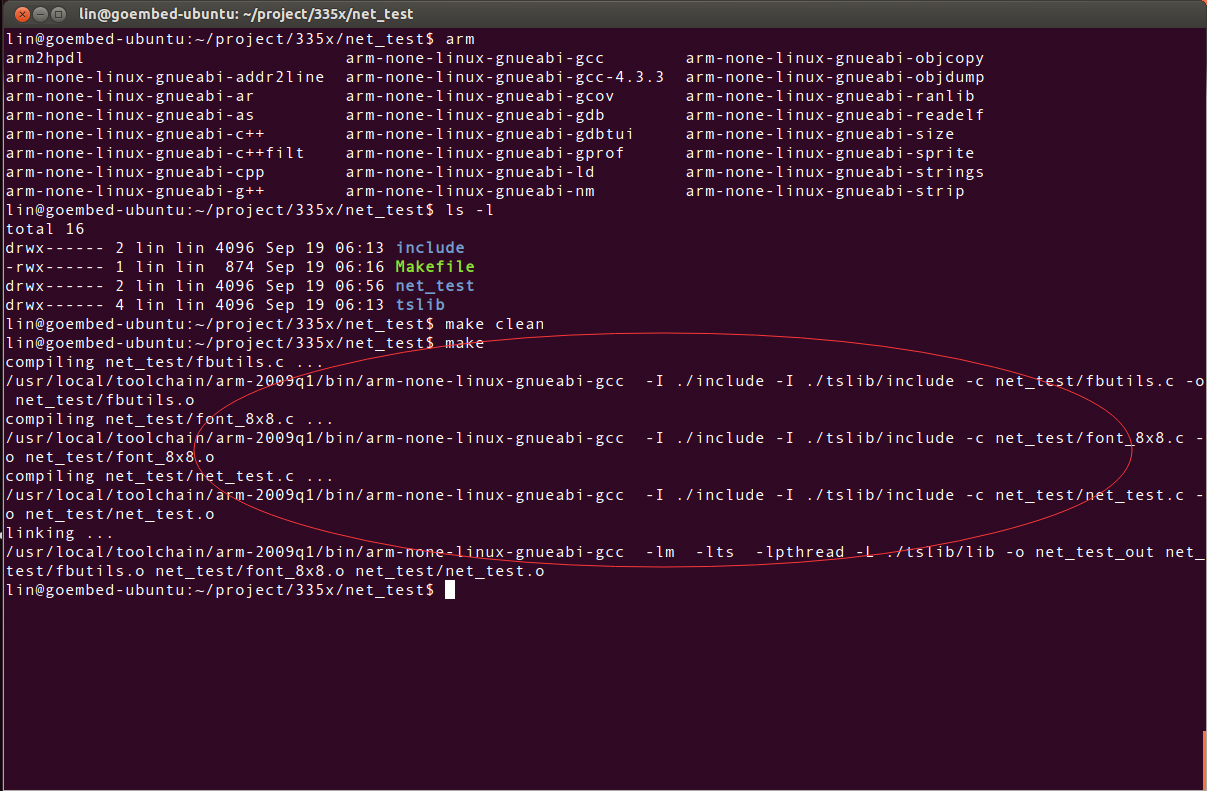
先导出交叉编译工具，如下图：



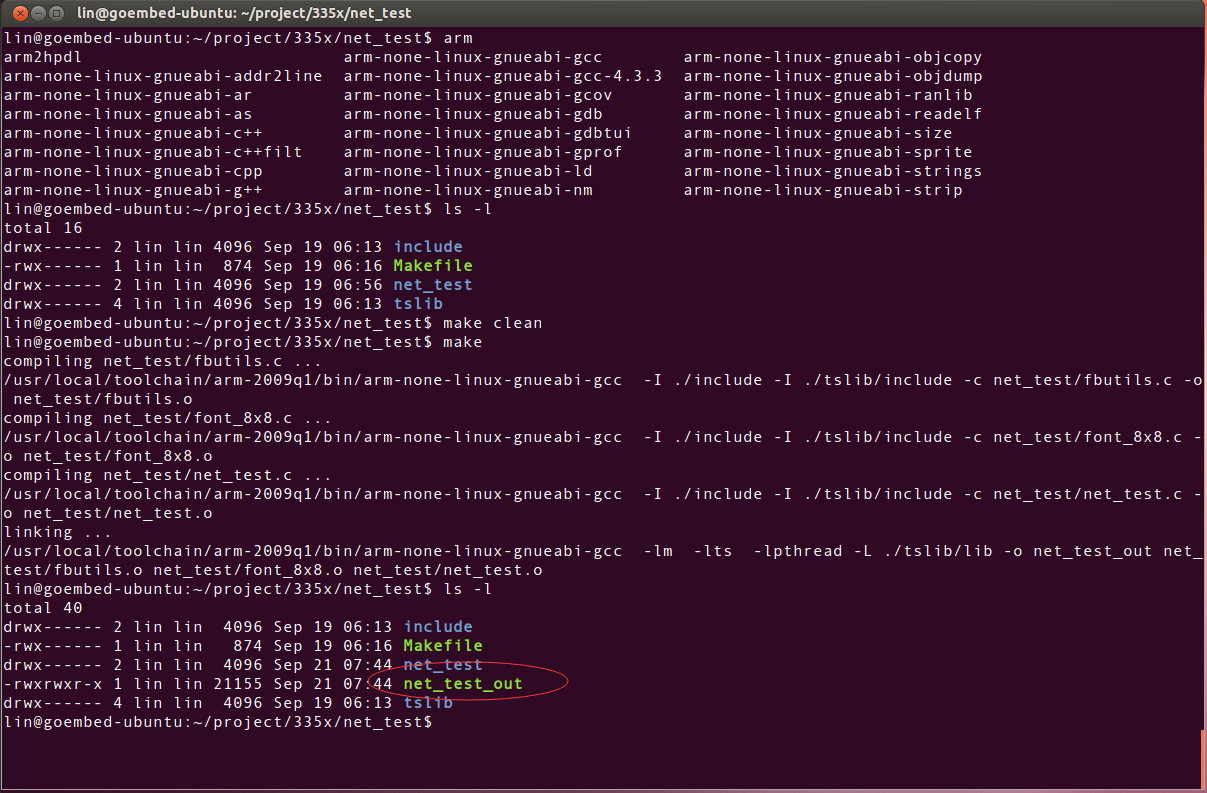
进入到测试文件夹：



执行make（若有残留旧文件可先执行“make clean”）：

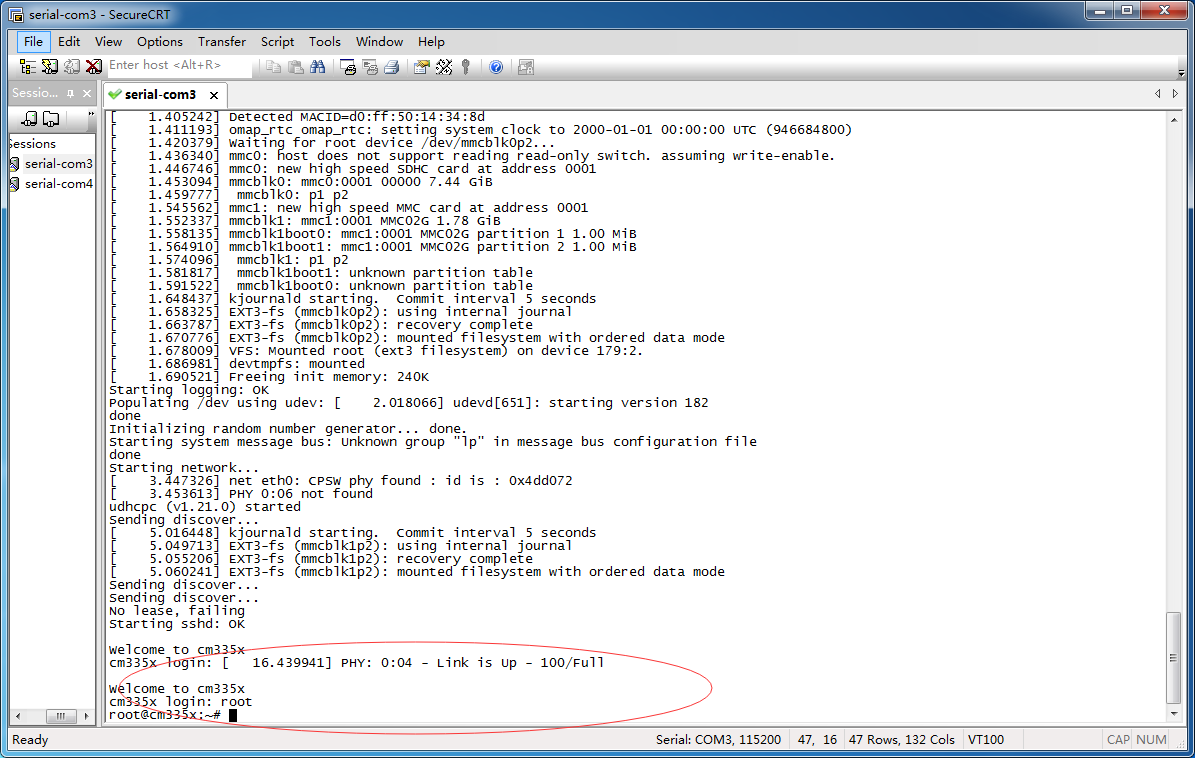


生成可执行文件net\_test\_out：

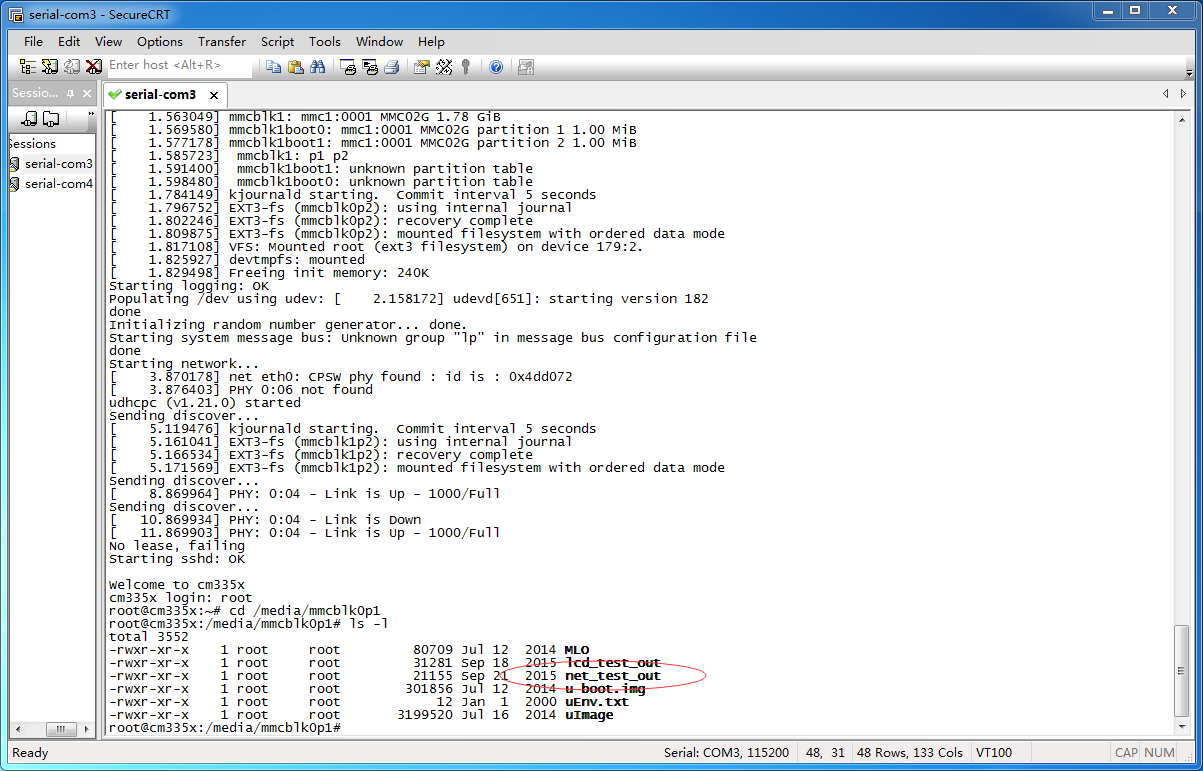


### 五、执行测试程序

上电开机后输入root登入SBC3358-B1A：



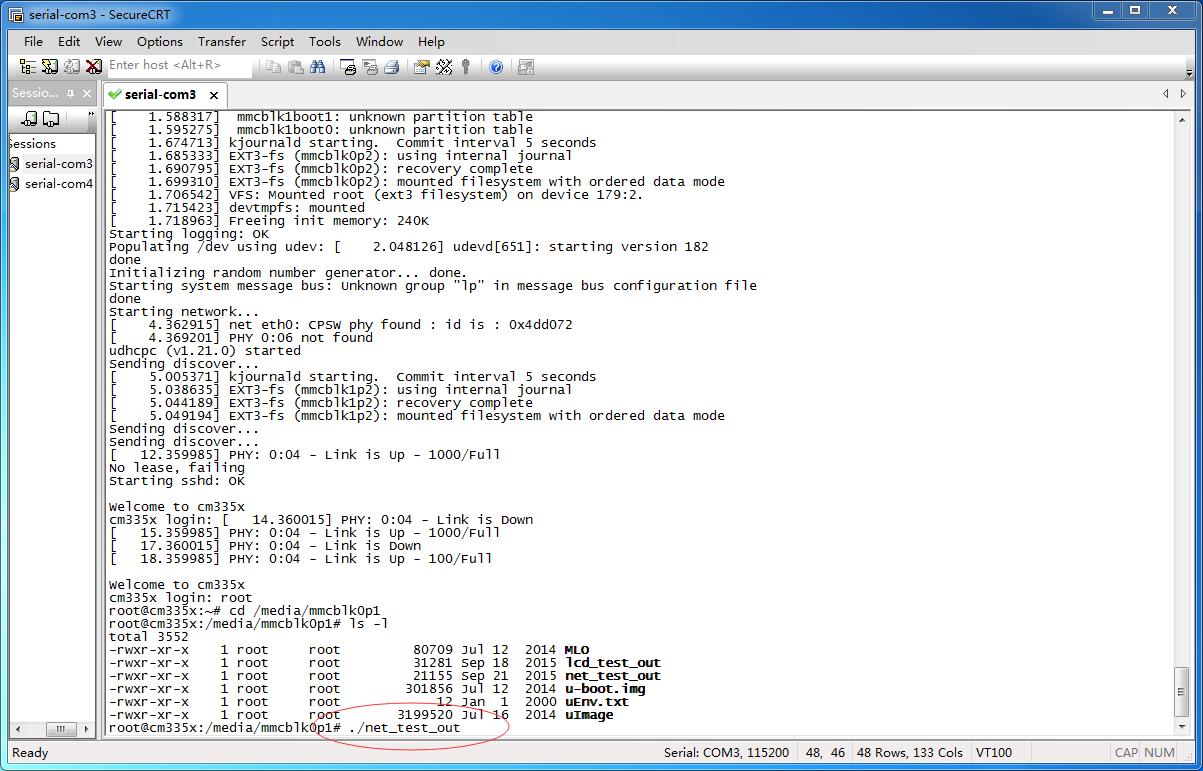
找到该文件：（事先从电脑上复制到开发板,该文件的位置根据用户启动系统的实际情况决定，在这里不赘述）



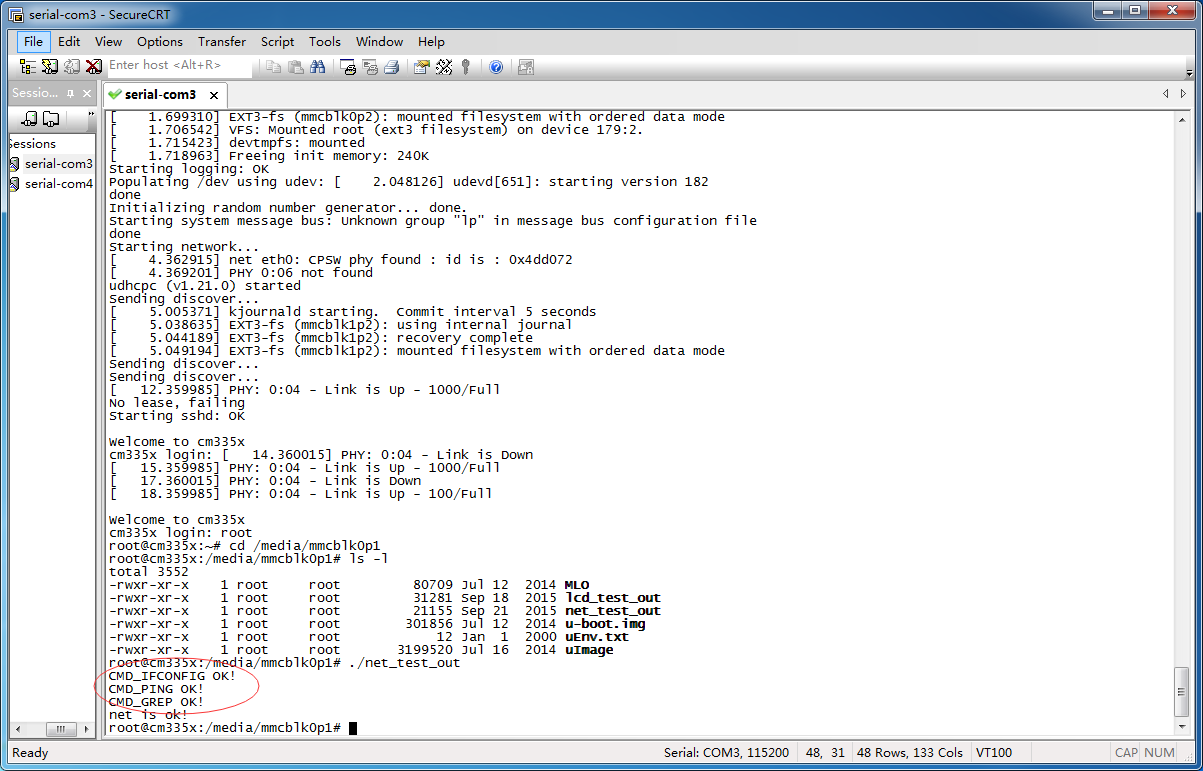
* 命令：

**./net\_test\_out**

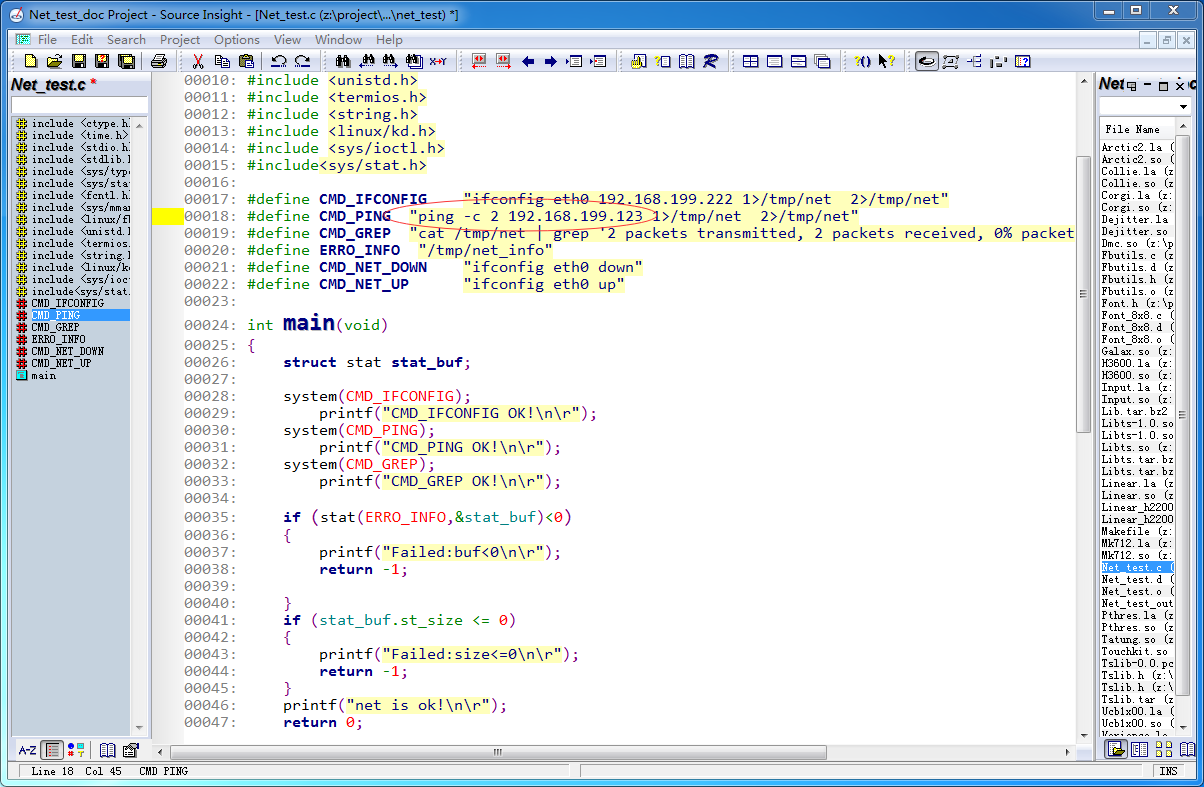
输入以上命令执行该应用程序：



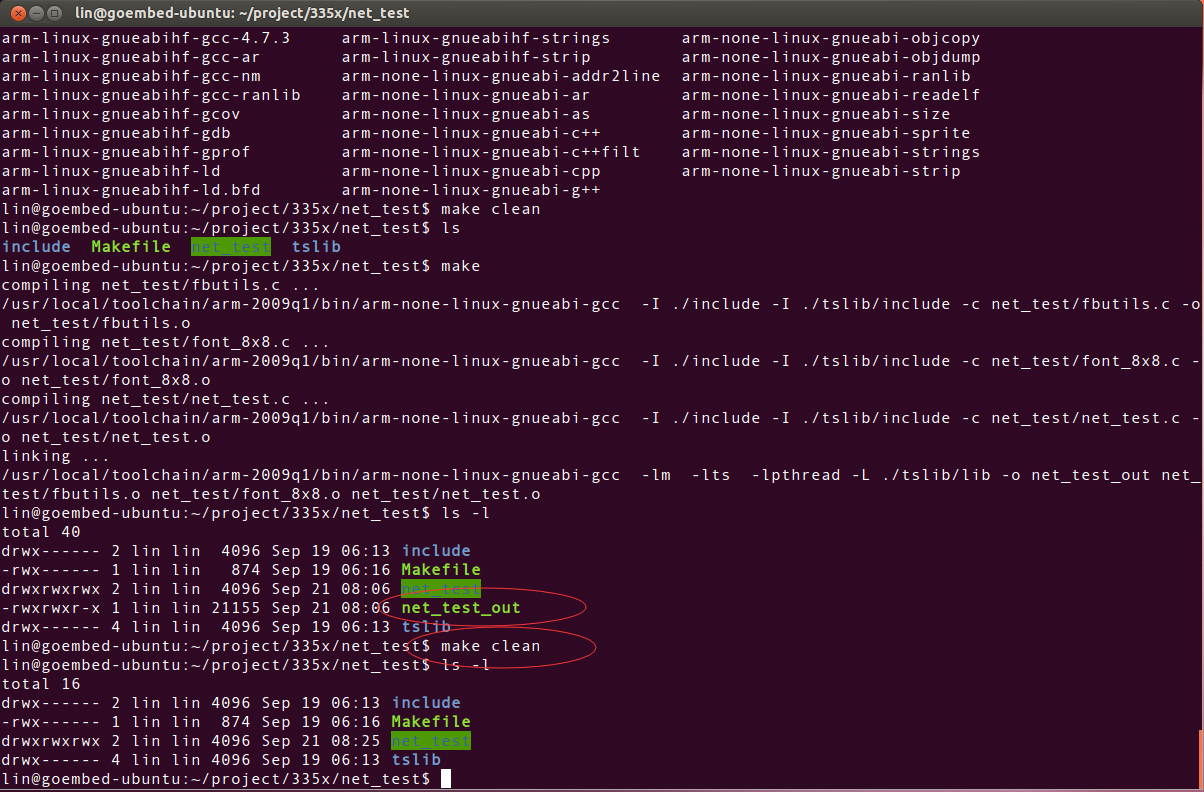
测试结束后程序自动安全退出,此次测试结果如下：



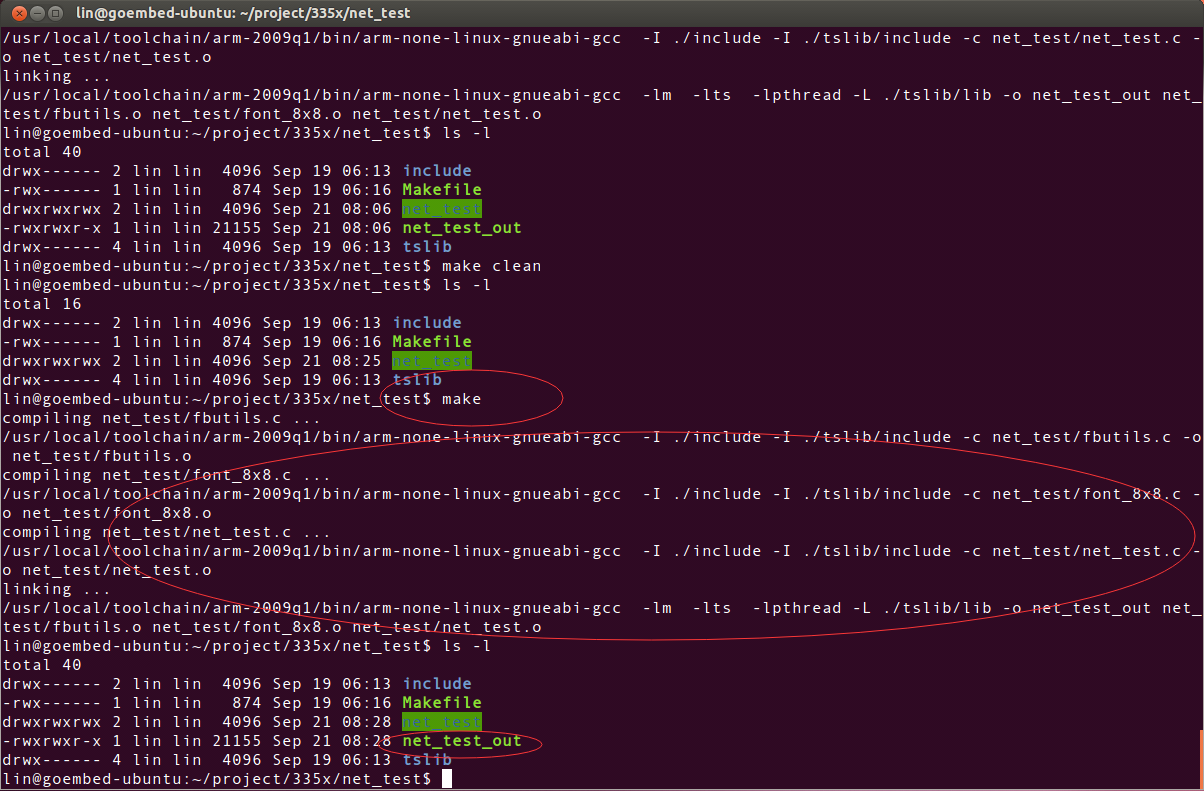
为了演示网络测试失败的情况，我们改一下代码，我们把用于测试的服务器IP改掉：



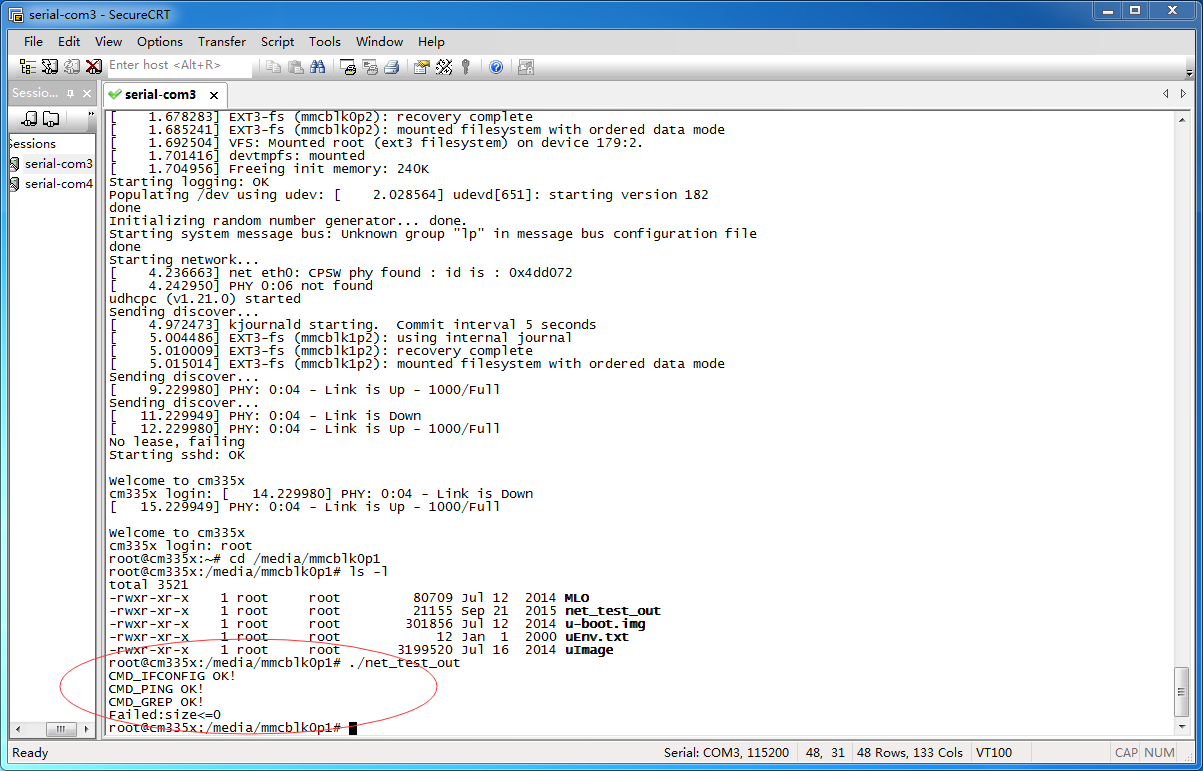
保存代码，重新编译一下，先执行make clean（原来的net\_test\_out和一些中间文件会被清除）：



再执行make编译新的代码，可见新的net\_test\_out生成：



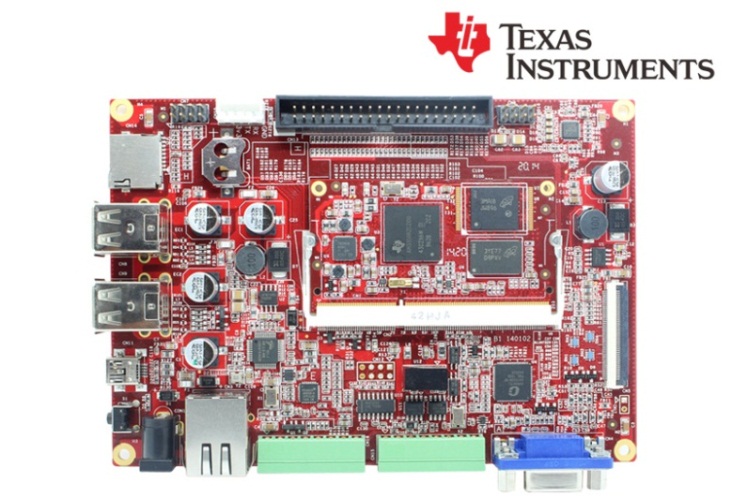
把新的net\_test\_out放到开发板上，重新开机测试结果如下：



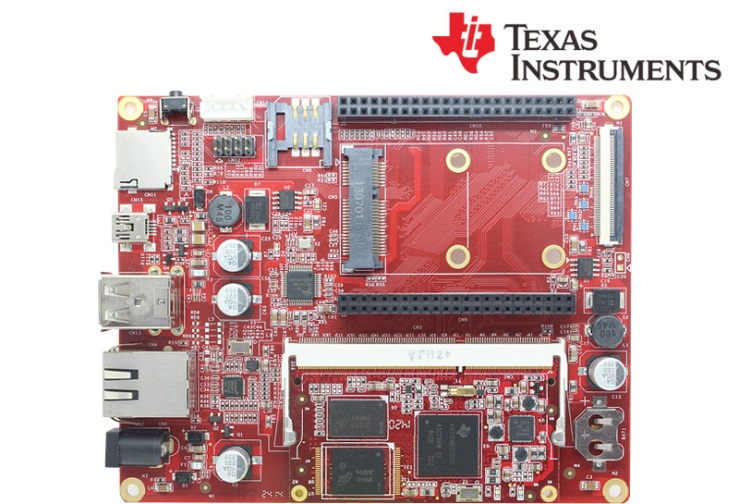
由于此时网络是不通的（我的局域网同个网段中没有192.168.199.123这个服务器），也就是说执行ping命令返回的信息写入/tmp/net的内容中没有“2 packets transmitted, 2 packets received, 0% packet loss”，那么网络就肯定还没通，所以返回的“Failed：size<0”代表网络不通。

到此，网络测试程序成功编写并分析完毕。

### 附 相关GOEMBED产品介绍

[](http://www.goembed.com/index.php/Products/detail/tpid/25)

*SBC335x – B1A*

[](http://www.goembed.com/index.php/Products/detail/tpid/12)

*SBC335x – B2A*

The single board computer SBC335x-B1A/B2A which has an expansion board to carry the CM335X is one of our design of the base plate . The flexible design allows the fast and easy way of realizing and upgrading the controller’s capabilities. In additional to those features offered by CM335X.

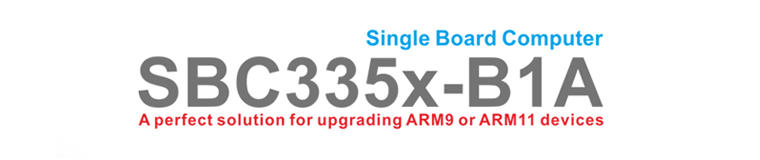
The B1A features 4 serial ports (including 2 RS232 and 2 TTL), 4 USB Host and 1 USB OTG, 1 Ethernet ports, CAN, RS485, Wiegand, VGA, LCD, Touch screen, Audio, ADC and more other peripherals.

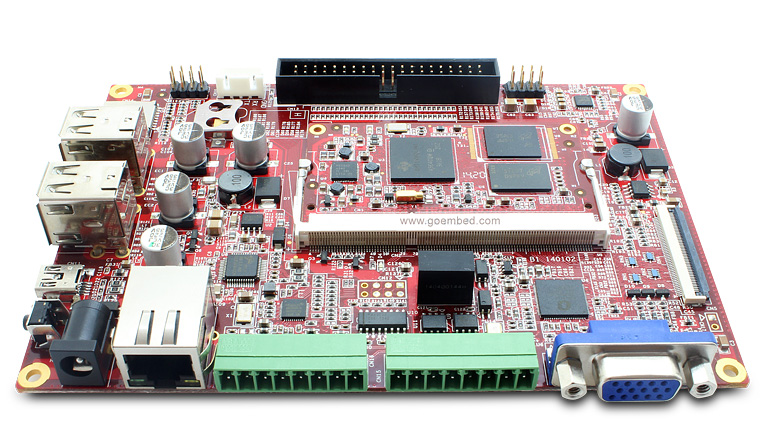
The B2A features 4 USB Host and 1 USB OTG, 1 Ethernet ports, LCD, Touch screen,RTC, and more other peripherals.

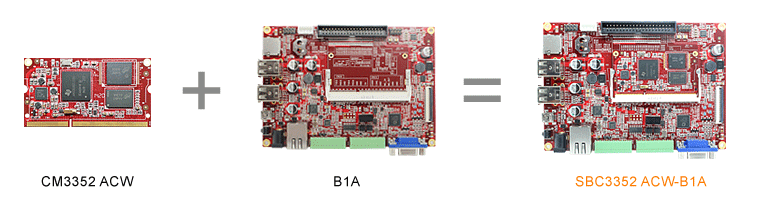
The SBC board targets a wide range of applications, including: HMIs, Digital Signage, POS, Data Terminal, Medical Devices, Navigation, Industrial Automation, Entertainment system, Thin Clients, Robotics, Game Console and much more.

The SBC335x-B1A/B2A are ready-to-run platform to support Linux 3.x, Android 4.x and WinCE 7.0/6.0 operating systems.

If you want to support other Operating System, For more information to contact us.







**SBC335x-B1A boards Description of part code:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Series** | **B1** | **B1** | **B1** | **B1** |
| **Part Code** | SBC3352 ACW-B1A | SBC3352 BCW-B1A | SBC3358 ACW-B1A | SBC3358 BCW-B1A |
| **Order Code** | - | - | - | - |
| **Core Module** | [CM3352 ACW](http://goembed.com/index.php/Products/detail/tpid/14)  [-M51E20/08](http://goembed.com/index.php/Products/detail/tpid/14) | [CM3352 BCW](http://goembed.com/index.php/Products/detail/tpid/29)  [-M51E40/08](http://goembed.com/index.php/Products/detail/tpid/29) | [CM3358 ACW](http://www.goembed.com/index.php/Products/detail/tpid/44)  [-M51E20/10](http://www.goembed.com/index.php/Products/detail/tpid/44) | [CM3358 BCW](http://goembed.com/index.php/Products/detail/tpid/22)  [-M51E40/10](http://goembed.com/index.php/Products/detail/tpid/22) |
| **CPU Type** | ARM Cortex™-A8 | | | |
| **CPU Cores** | 1x | | | |
| **CPU Clock** | 800MHz | 800MHz | 1.0GHz | 1.0GHz |
| **RAM DDR3** | Micron 512MB@16bit\*1 | | | |
| **eMMC Flash** | 2GB@8bit\*1 | 4GB@8bit\*1 | 2GB@8bit\*1 | 4GB@8bit\*1 |
| **PMU** | TI TPS65910A3 | | | |
| Supply Voltage | DC 9-14V | | | |
| Optimal Input | DC 12V,1.5A | | | |
| **Size(L\*W)** | 146 x 102 mm | | | |
| **Temperature** | 0° to 70° C | | | |
| **Support OS** | Linux 3.x/ Android 4.x/ Ubuntu/ Angstrom/ Debian/ QT/ WinCE 6.0/7.0 | | | |
| **Inventory status** | In Stock | **Out of Stock**  [**Contact us**](mailto:%20sales@goembed.com) | In Stock | **Out of Stock**  [**Contact us**](mailto:%20sales@goembed.com) |
| **Minimum Availability** | 2022 | | | |

**SBC335x-B1A Block Diagram**

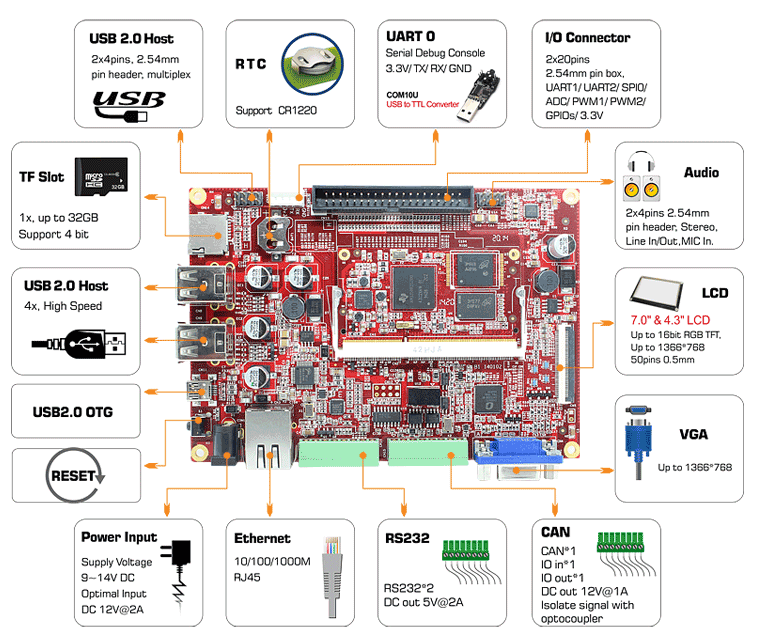
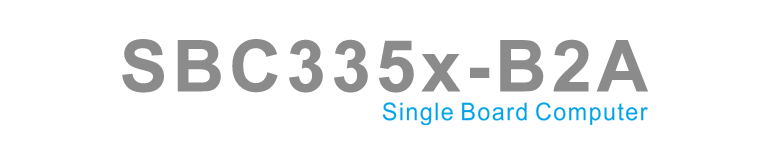
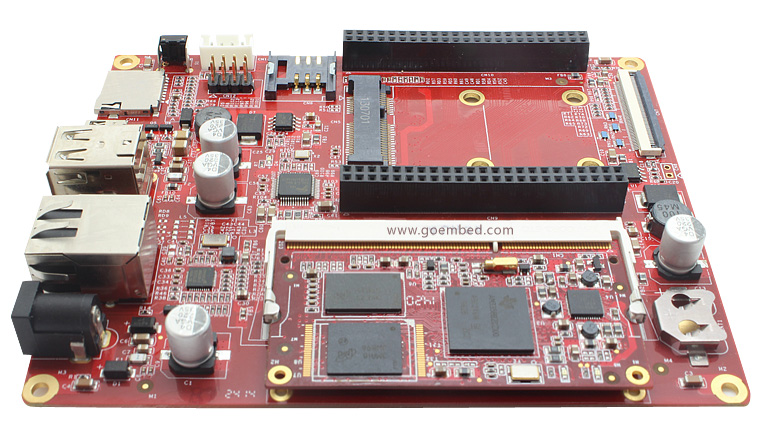
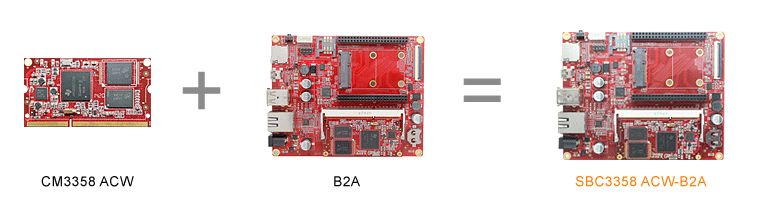


Figure 1 B1 Block Diagram







**SBC335x-B2A boards Description of part code:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Series** | **B2A** | **B2A** | **B2A** | **B2A** |
| **Part Code** | SBC3352 ACW-B2A | SBC3352 BCW-B2A | SBC3358 ACW-B2A | SBC3358 BCW-B2A |
| **Order Code** | - | - | - | - |
| **Core Module** | [CM3352 ACW](http://goembed.com/index.php/Products/detail/tpid/14)  [-M51E20/08](http://goembed.com/index.php/Products/detail/tpid/14) | [CM3352 BCW](http://goembed.com/index.php/Products/detail/tpid/29)  [-M51E40/08](http://goembed.com/index.php/Products/detail/tpid/29) | [CM3358 ACW](http://www.goembed.com/index.php/Products/detail/tpid/44)  [-M51E20/10](http://www.goembed.com/index.php/Products/detail/tpid/44) | [CM3358 BCW](http://goembed.com/index.php/Products/detail/tpid/22)  [-M51E40/10](http://goembed.com/index.php/Products/detail/tpid/22) |
| **CPU Type** | ARM Cortex™-A8 | | | |
| **CPU Cores** | 1x | | | |
| **CPU Clock** | 800MHz | 800MHz | 1.0GHz | 1.0GHz |
| **RAM DDR3** | Micron 512MB@16bit\*1 | | | |
| **eMMC Flash** | 2GB@8bit\*1 | 4GB@8bit\*1 | 2GB@8bit\*1 | 4GB@8bit\*1 |
| **PMU** | TI TPS65910A3 | | | |
| Supply Voltage | DC 9-14V | | | |
| Optimal Input | DC 12V,1.5A | | | |
| **Size(L\*W)** | 130 x 103.5 mm | | | |
| **Temperature** | 0° to 70° C | | | |
| **Support OS** | Linux 3.x/ Android 4.x/ Ubuntu/ Angstrom/ Debian/ QT/ WinCE 6.0/7.0 | | | |
| **Inventory status** | In Stock | **Out of Stock**  [**Contact us**](mailto:%20sales@goembed.com) | In Stock | **Out of Stock**  [**Contact us**](mailto:%20sales@goembed.com) |
| **Minimum Availability** | 2022 | | | |

**SBC335x-B2A Block Diagram**

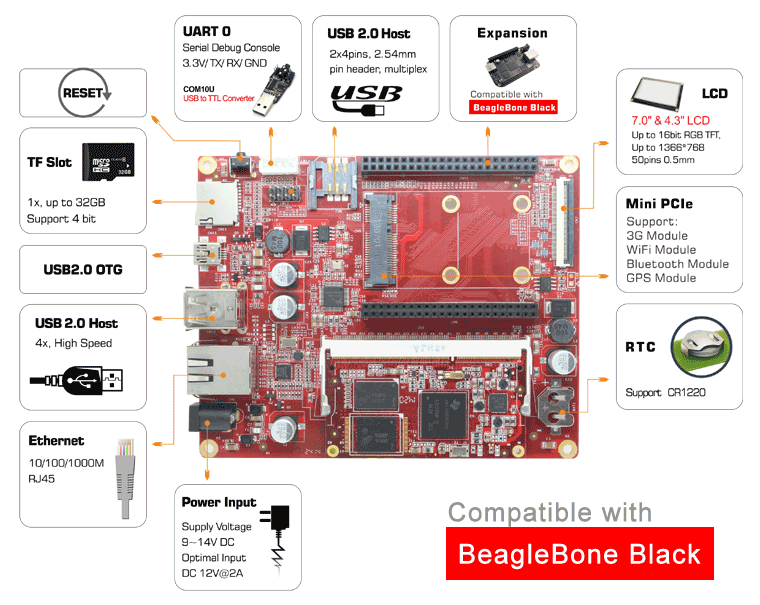


Figure 1 B2A Block Diagram

**ABOUT GOEMBED**

GOEMBED team with experienced embedded engineers who have been engaged in ARM hardware and software design for 10+ years.

Our products include single board computers and CPU core modules based on TI ® Sitara and Freescale ® i.MX Applications Processors based on ARM® Cores. Supported by Linux / Android / Debian / Ubuntu / QT / Angstrom / WinCE 7.0 & 6.0 / uCOS. We can redesign carrier boards and SBC as your idea quickly.

GOEMBED focus on Embedded Board Solutions, provide a complete new board for your specified requirement or even a turnkey solution to accelerate your new products to market.

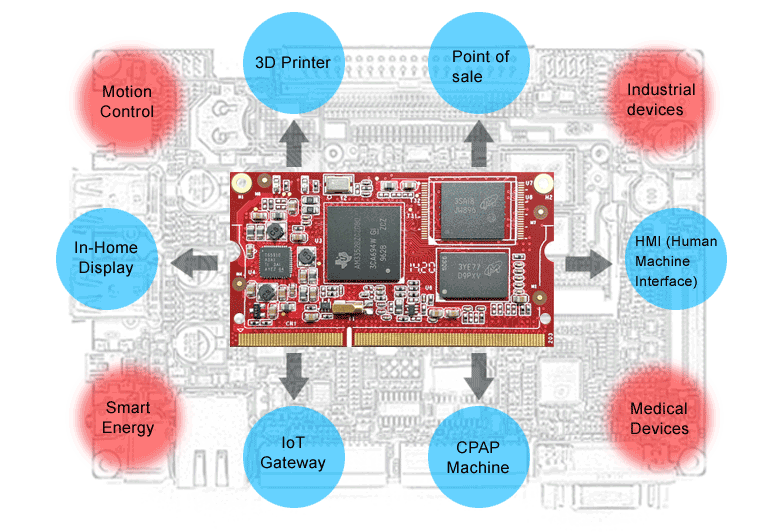
We are your trust worthy partner on ARM embedded design services and solutions.

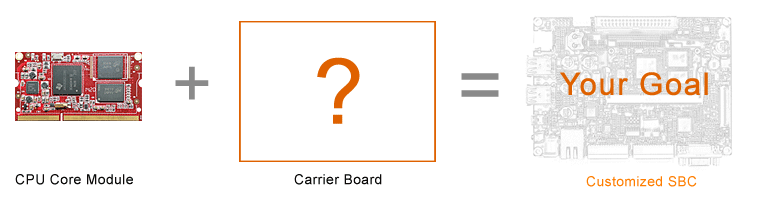
**More Carrier Boards**

Customized based on your needs!

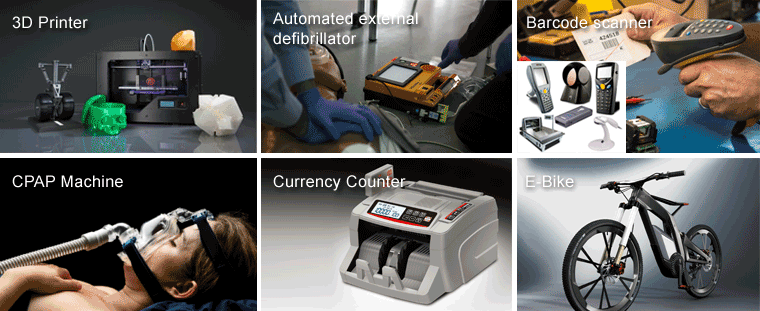
**ODM / OEM Services**

Bring your new products to market quickly





**Related end equipment**





 Learn more applications please click <http://www.ti.com/lsds/ti/apps/appshomepage.page>

