

|  |
| --- |
| Linux下LCD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。

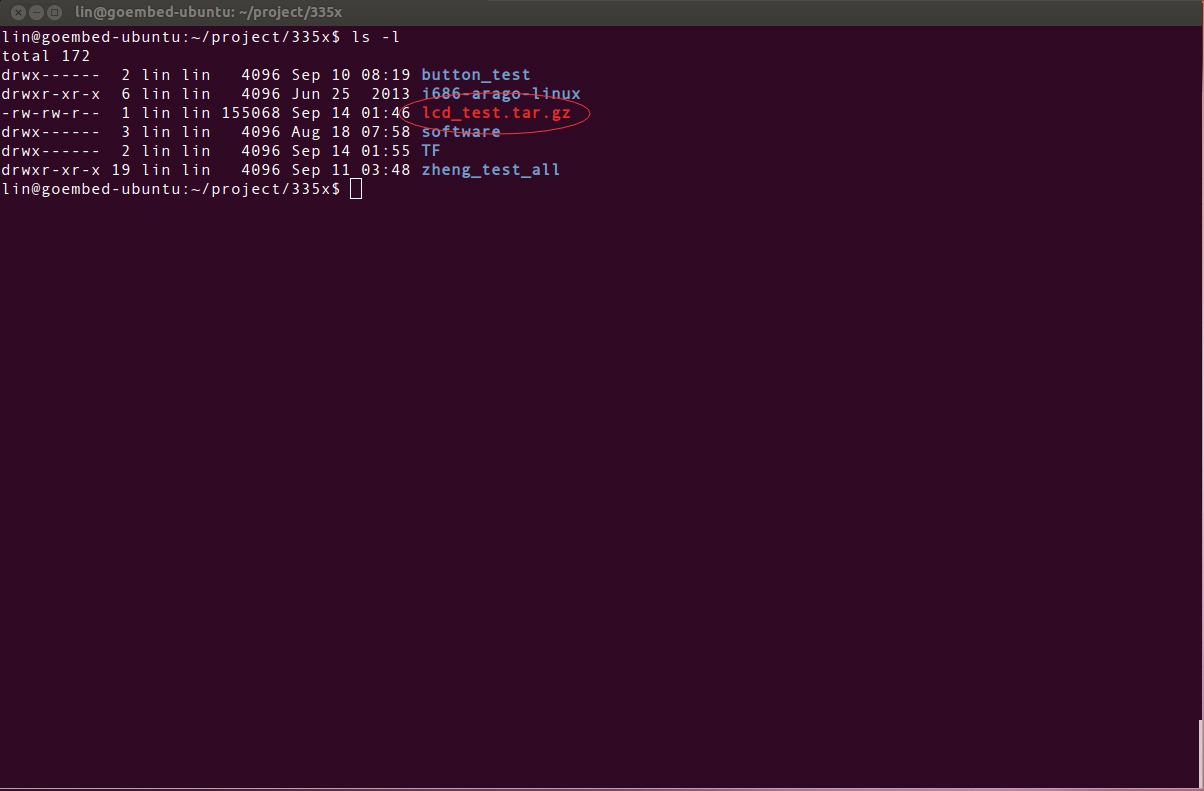
2、为了方便阅读和修改代码，在这里我使用的是Source Insight(一个代码编辑工具)对代码进行修改。用户可以直接在终端使用VI编辑器编辑代码，结果是一样的，这里是为了阅读方便。

注：1、应用程序编译要采用arm-2009q1交叉编译。（后续补上）

2、以下使用的是4.3寸TFT屏进行测试

### 二、准备代码

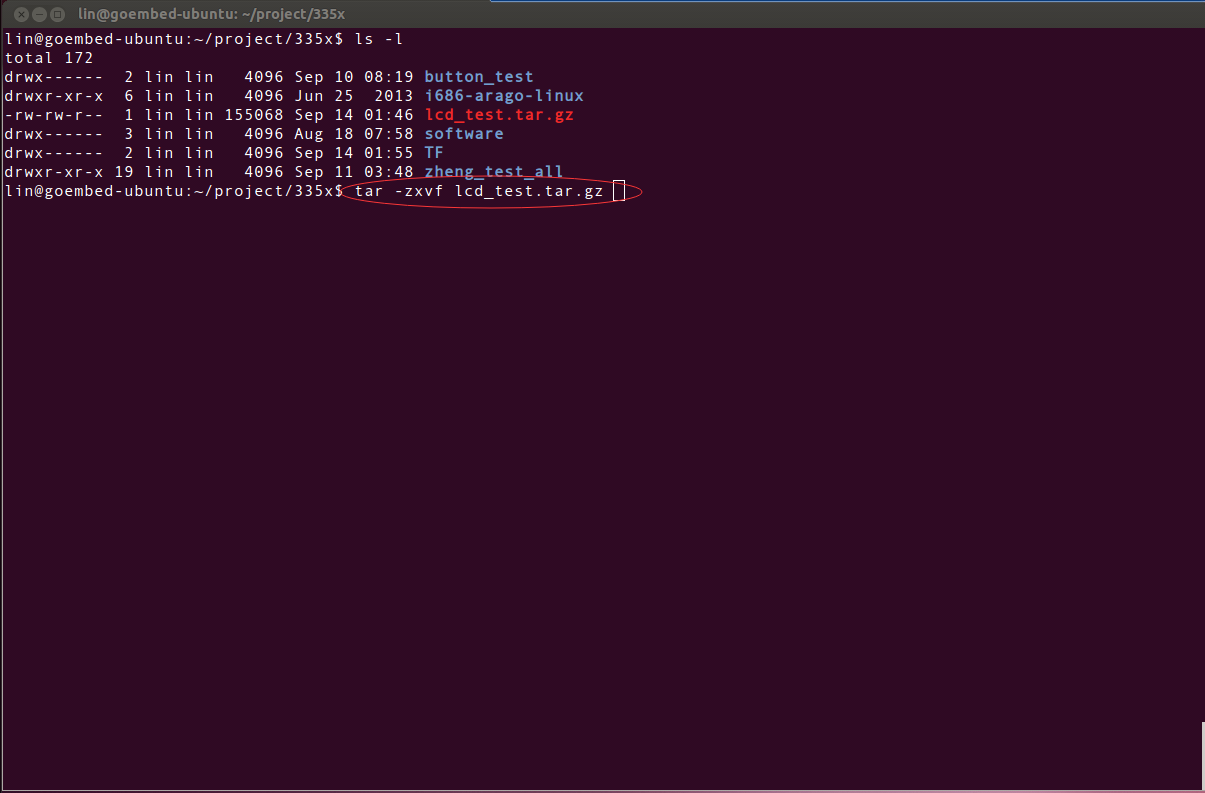
1、找到lcd\_test的源代码压缩包，名为“lcd\_test.tar.gz”。



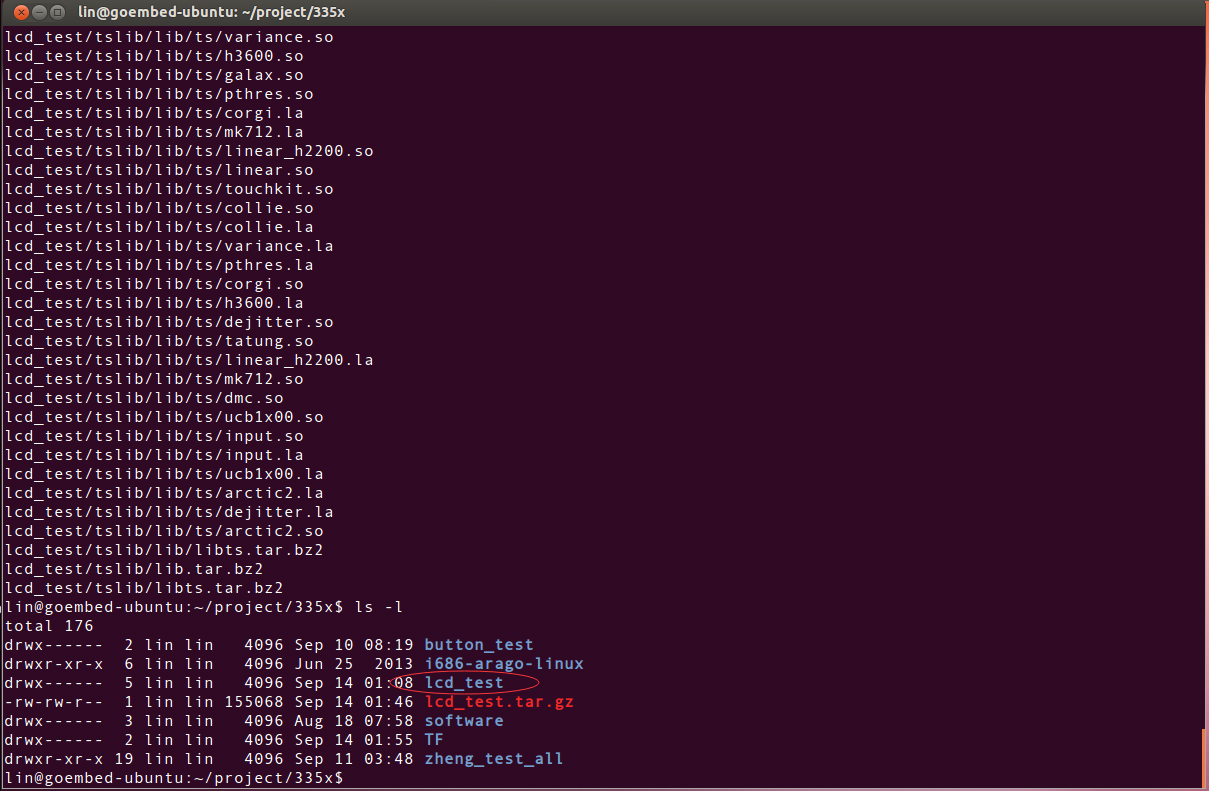
用以下命令进行解压：

* 命令：

tar –zxvf lcd\_test.tar.gz



这时在当前文件夹下会生成lcd\_test文件夹，里面的代码就是测试的源码：



### 三、分析代码

#### 1、frame buffer 设备驱动原理简介

（1）通过framebuffer，应用程序用mmap将显存映射到应用程序的虚拟地址空间，只要把用于显示的数据写入该内存空间就可以在屏幕上显示出来。

（2）驱动程序需要分配系统的一部分内存作为显存，通过实现fb\_ops结构中的接口，控制和操作LCD内部的控制器。

（3）驱动程序需要将显存的起始地址和长度传递给LCD控制器的寄存器，LCD控制器就会自动地将显存中的数据显示在LCD屏幕上。

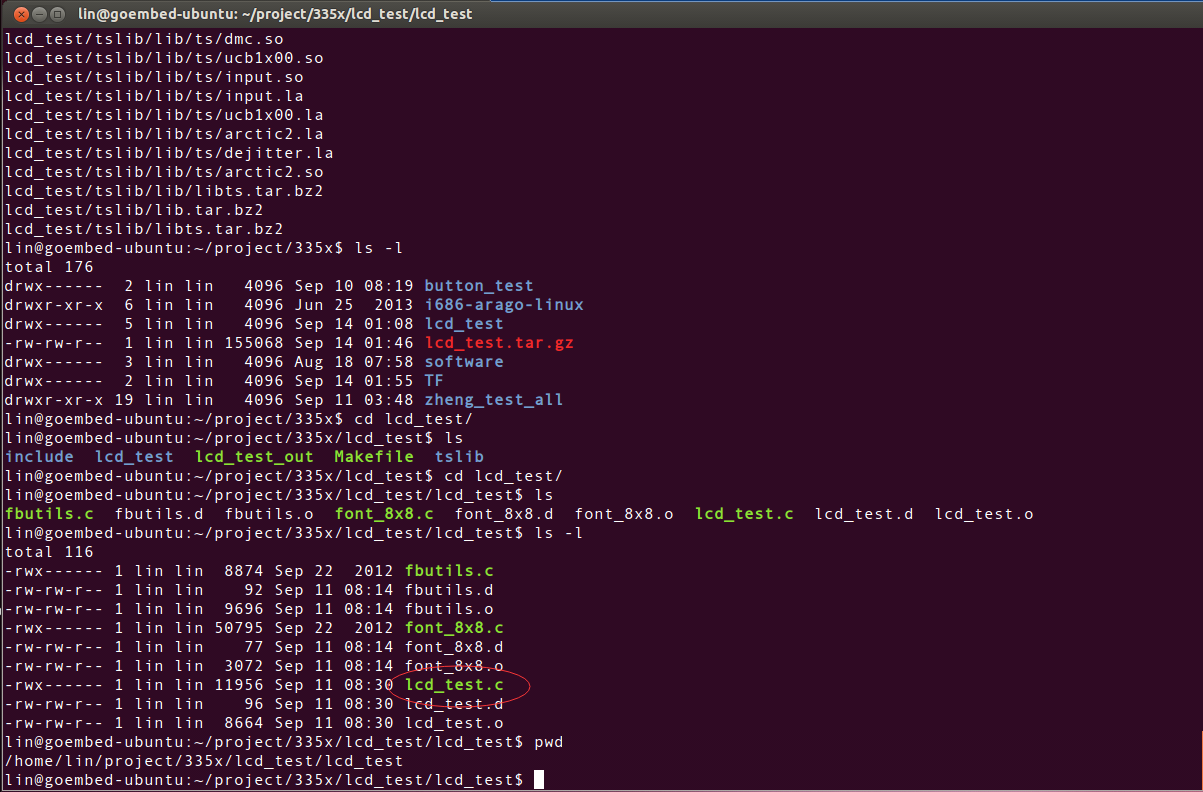
#### 2、Makefile文件

最重要的几个参数如下：

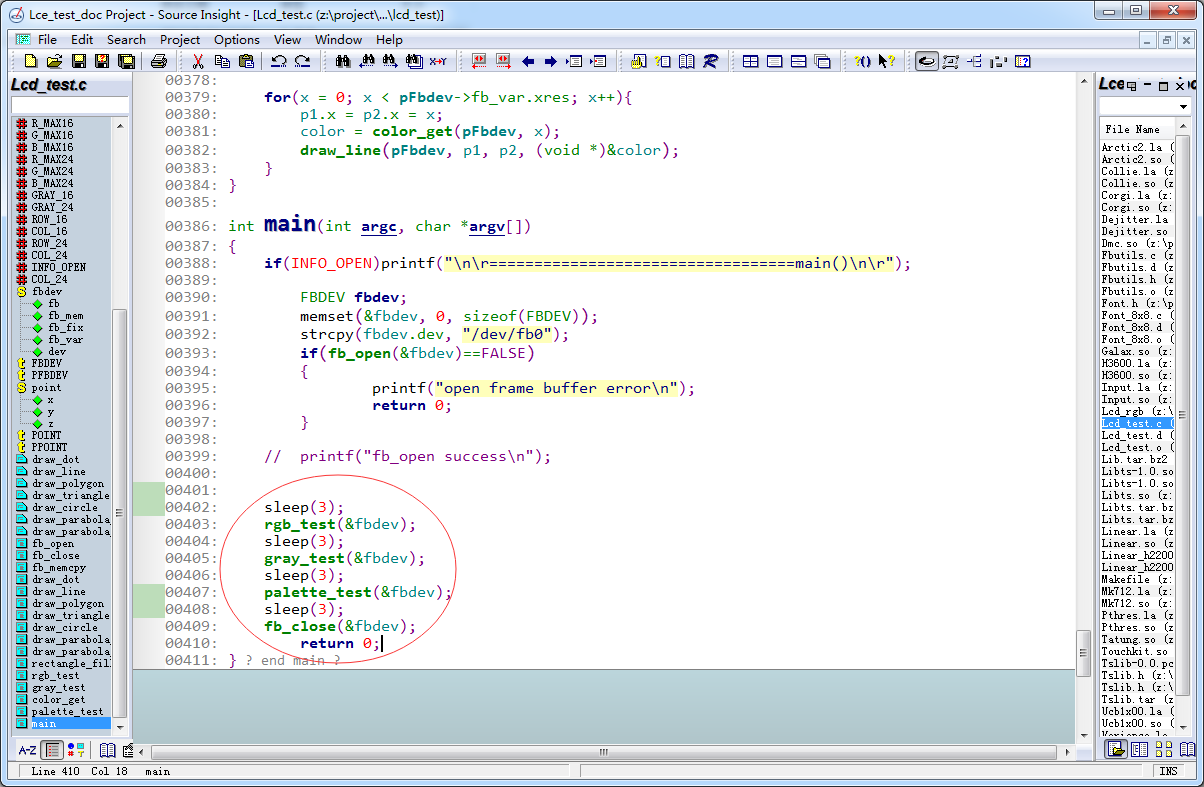
* “NAME：lcd\_test\_out”：指的是生成的可执行测试文件名
* “CROSS= /usr/local/toolchain/arm-2009q1/bin/arm-none-linux-gnueabi-”：指使用的是arm-2009ql交叉编译器
* “SRCDIRS= lcd\_test”指的是要编译的文件夹，这里指的就是源码包解压出来的lcd\_test文件夹，也是最重要的文件夹。

#### 3、分析“lcd\_test.c”

在lcd\_test文件夹中找到它：

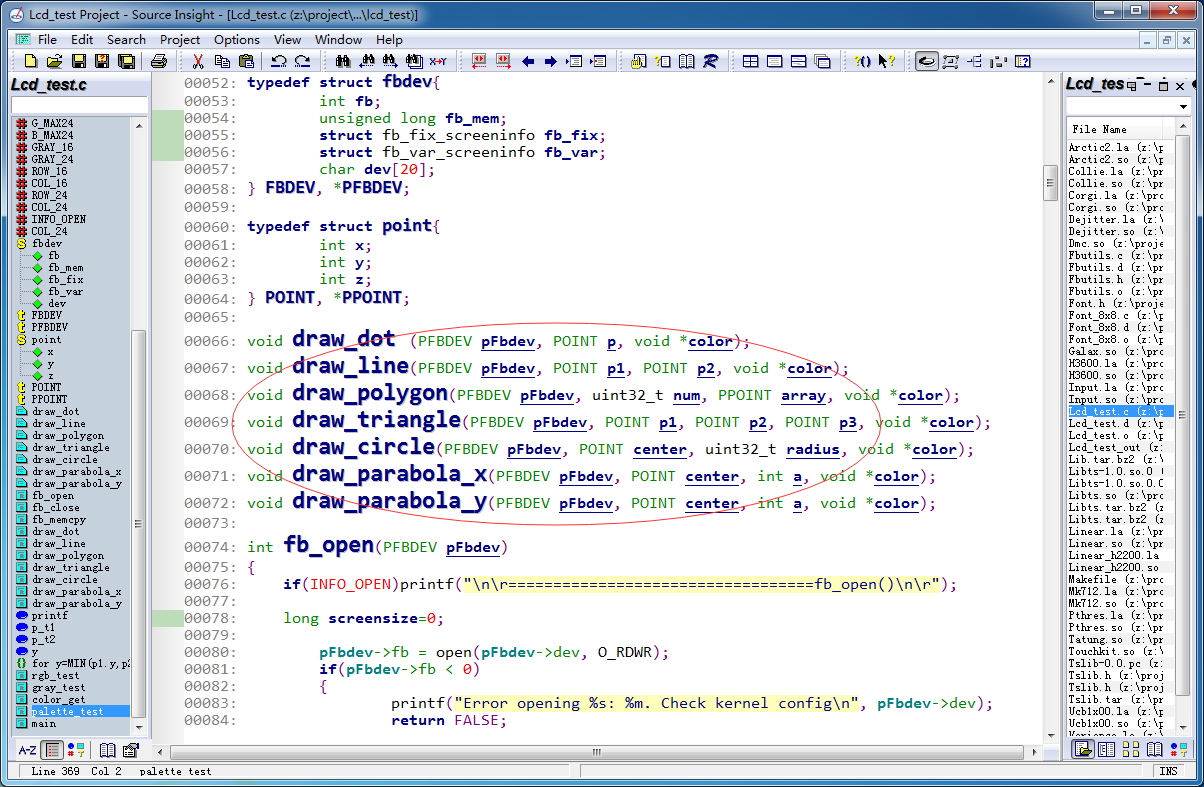


在main函数中打开的是“/dev/fb0”,由于我们只有一块LCD屏，默认为fb0。当我们打开fb0这个设备之后，底层驱动注册了一个framebuffer驱动，完成内存映射，该应用对应的用户空间与内核空间使用的是同一块内存，在此时该内存已映射为LCD显存，应用程序可以直接读写该内存，而且在LCD屏上会马上得到显示。比如将一张图片写入该内存，则LCD会马上显示该图片。也即是说，只要该内存写入数据，该数据就会立即在LCD屏上显示出来。根据这个原理，我们的应用程序就容易很多了。我们可以看到，在main函数中其实就是调用了几个测试函数，如下：



这几个函数的作用测试就是分别在LCD屏上进行三原色测试、灰度测试和调色板测试，这个我后面进行介绍。我们先来看一下测试程序中有哪些可调用函数比较重要：

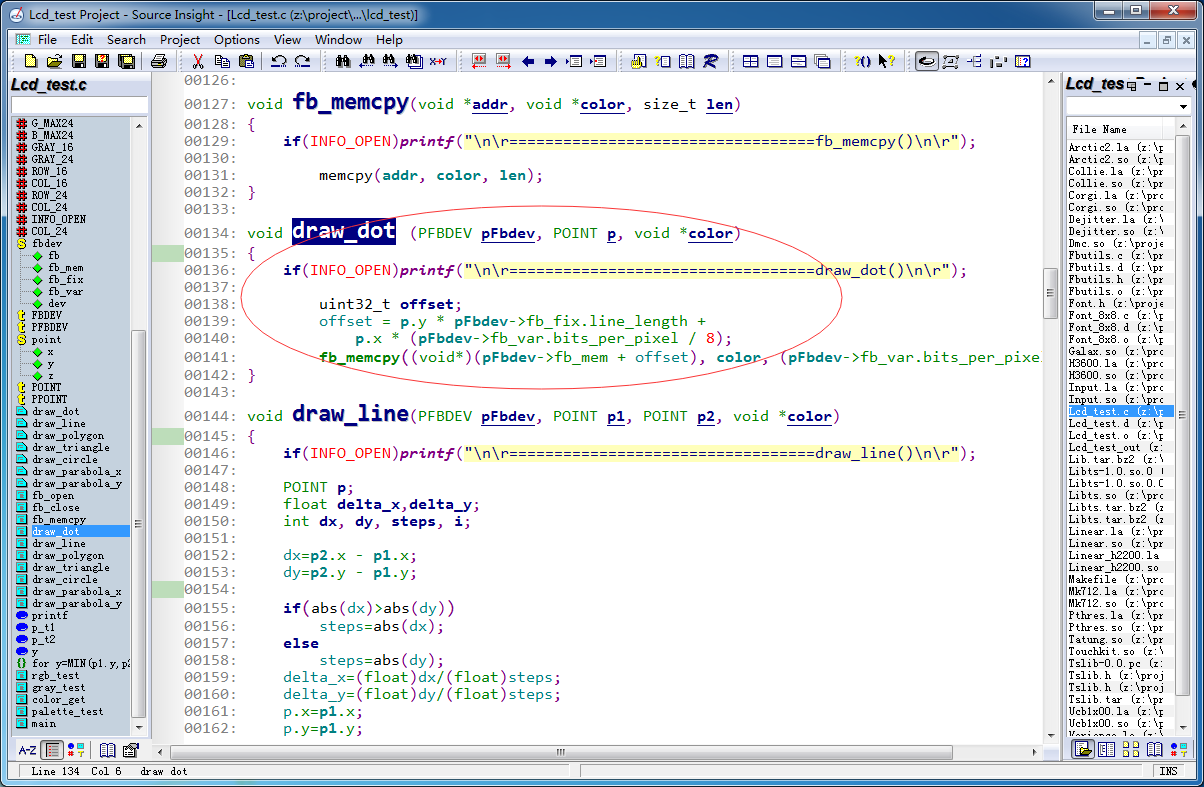
在文件开头我定义了以下函数：



我们逐个分析。

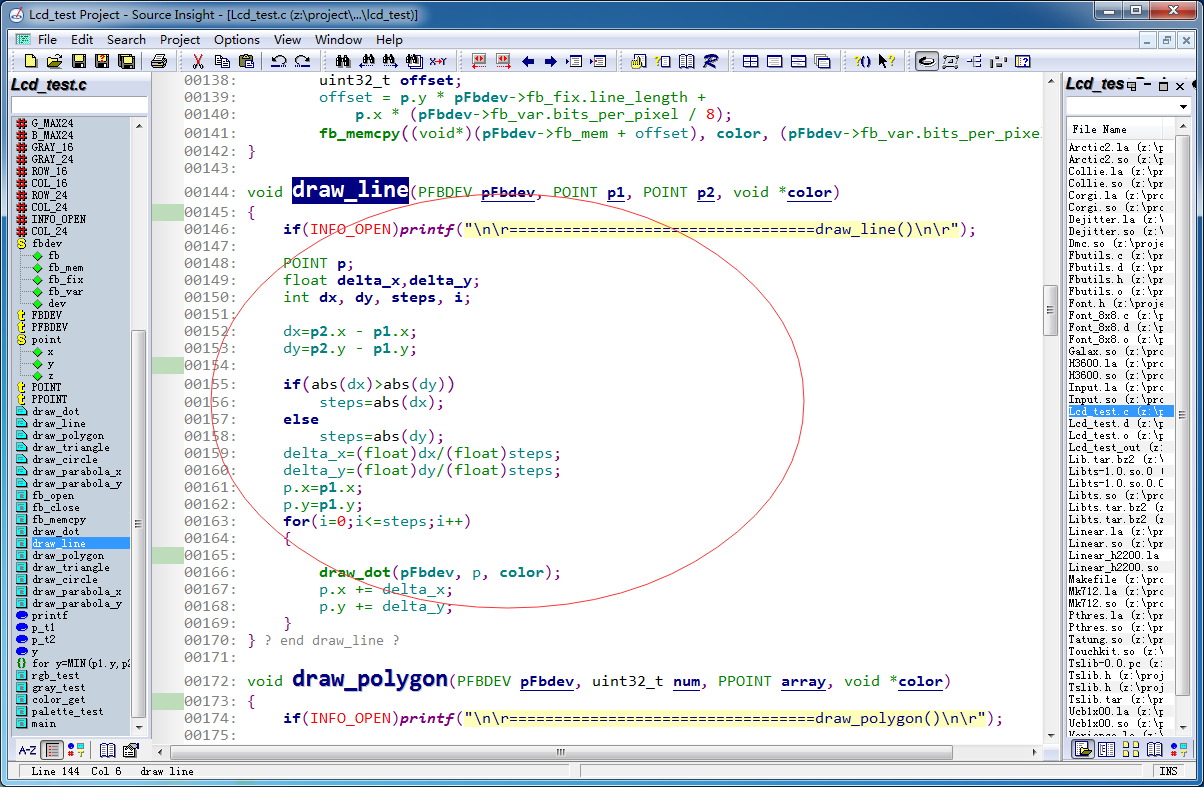
第一个是void draw\_dot (PFBDEV pFbdev, POINT p, void \*color);，

如下：



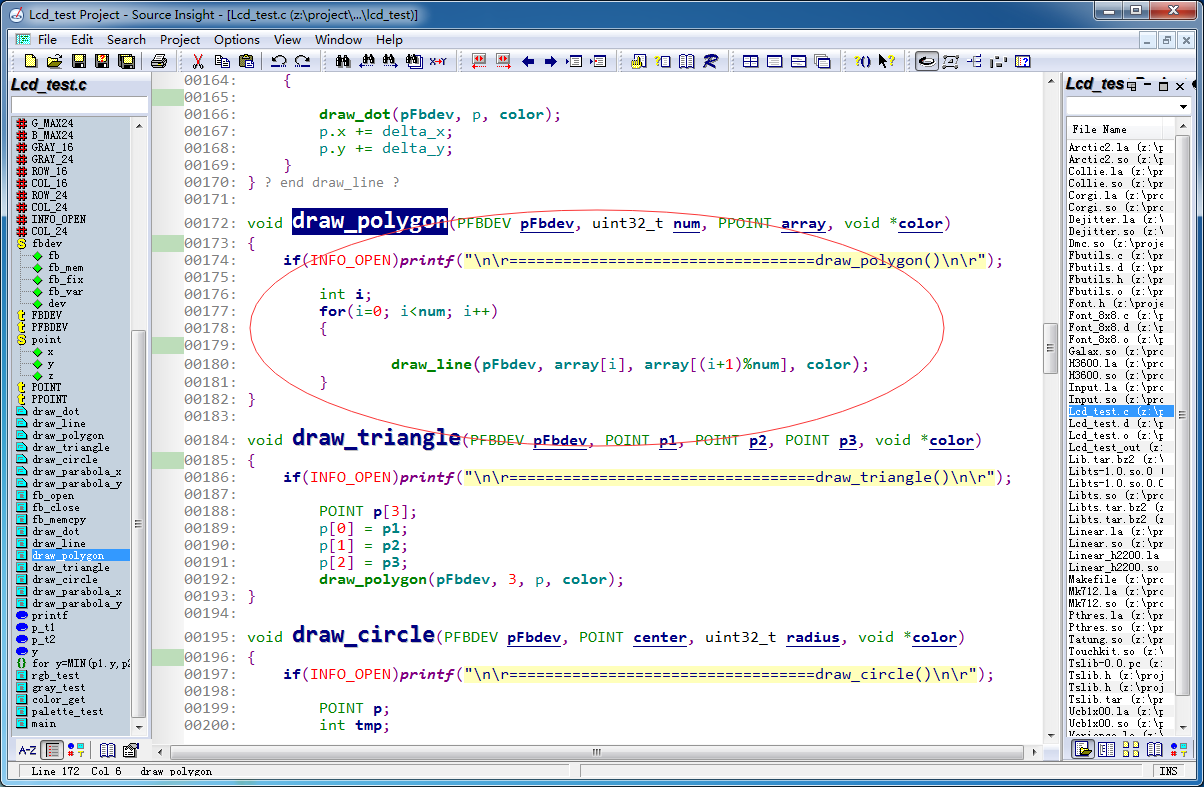
这个函数的作用是在LCD屏上画一个点，这个是最基本的函数，后面我们可以发现，很多函数都调用它。里面最重要的就是fb\_memcpy()函数，这个函数的作用是将某一个点的像素信息直接写到显存里，通过这个步骤可以在LCD屏上显示一个特定颜色，特定位置的点。

第二个是void draw\_line(PFBDEV pFbdev, POINT p1, POINT p2, void \*color)，如下：



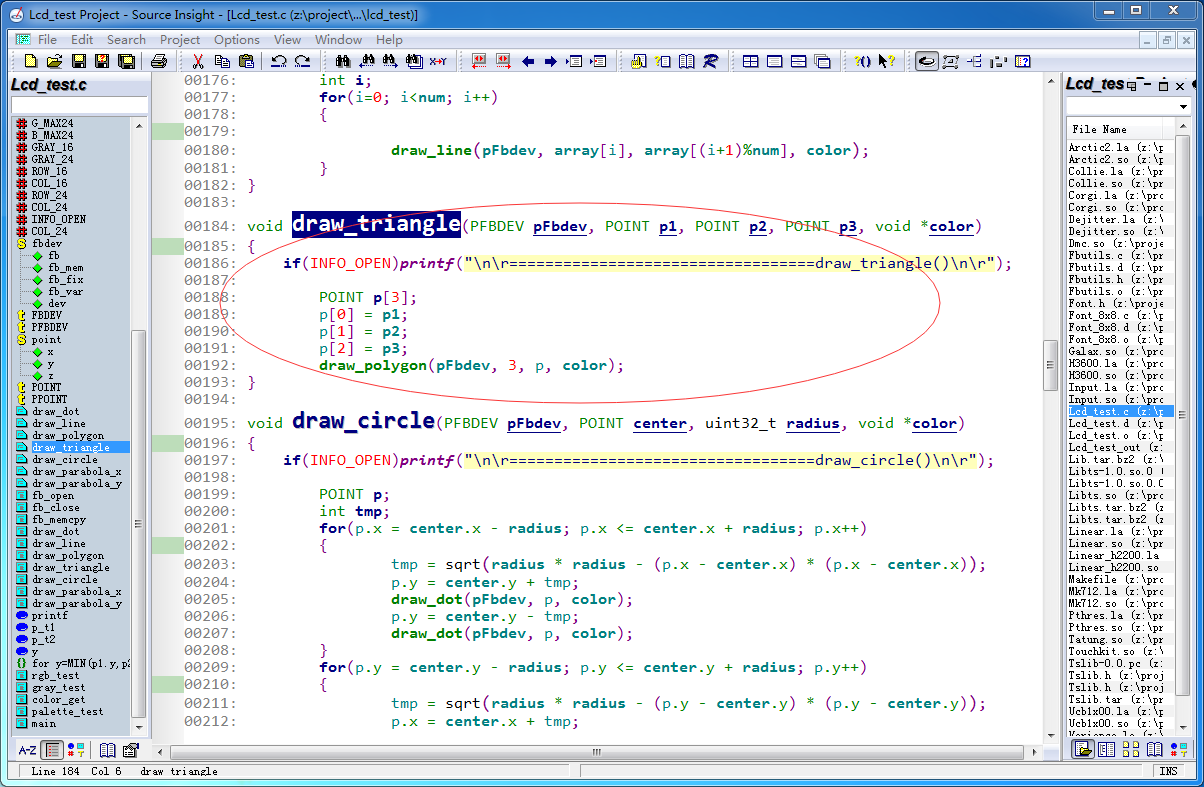
这个函数的作用是在LCD屏上画一条直线，我们可以看到，除了计算坐标的函数外，在for语句中调用的是draw\_dot(pFbdev, p, color);也就是说，其实画直线就是在画一个个连续的点，每个连续的点连起来就是一条线。

第三个是void draw\_polygon(PFBDEV pFbdev, uint32\_t num, PPOINT array, void \*color)，如下：



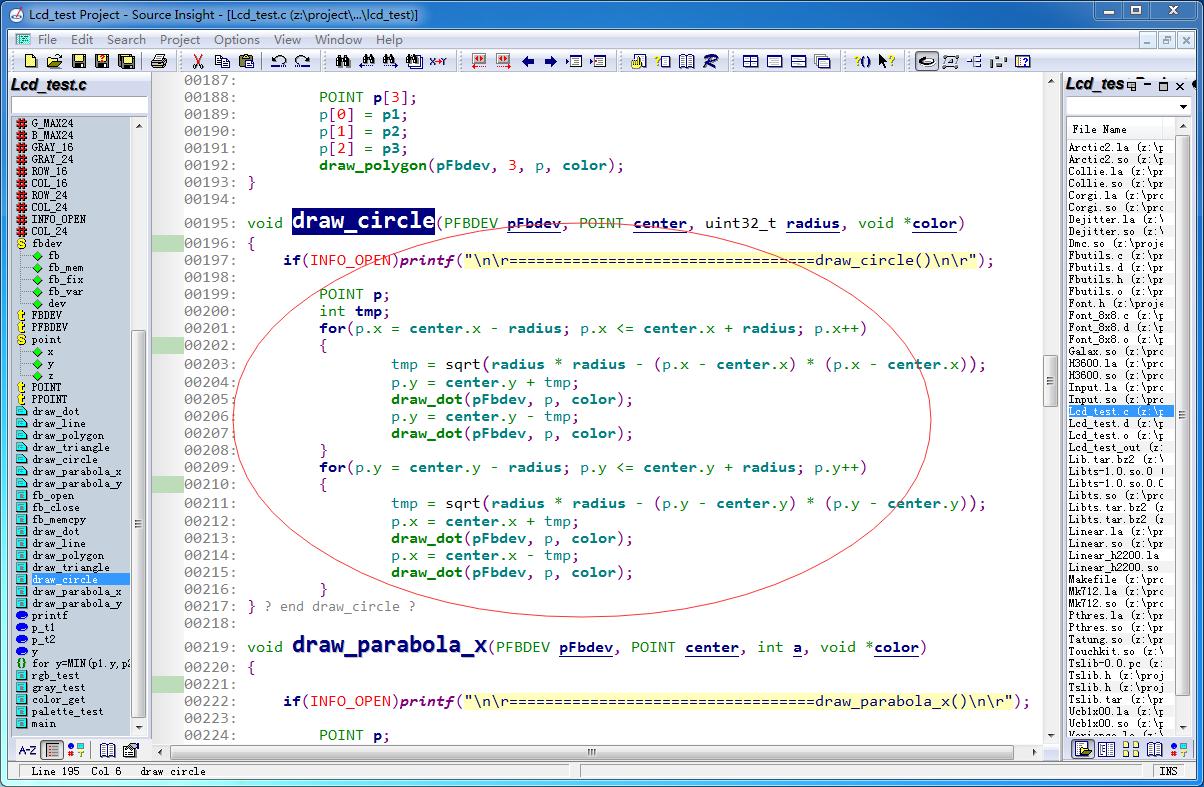
这个函数的作用是在LCD屏上画一个多边形。我们可以看到，函数里for语句中调用的其实就是draw\_line()函数，众所周知，多边形就是由三条或三条以上的线组成的，而每画一条线，在draw\_line()函数中又会调用draw\_dot()函数，其实说到底还是在画点。由点构成线，由线构成多边形。

第四个是void draw\_triangle(PFBDEV pFbdev, POINT p1, POINT p2, POINT p3, void \*color)，如下：



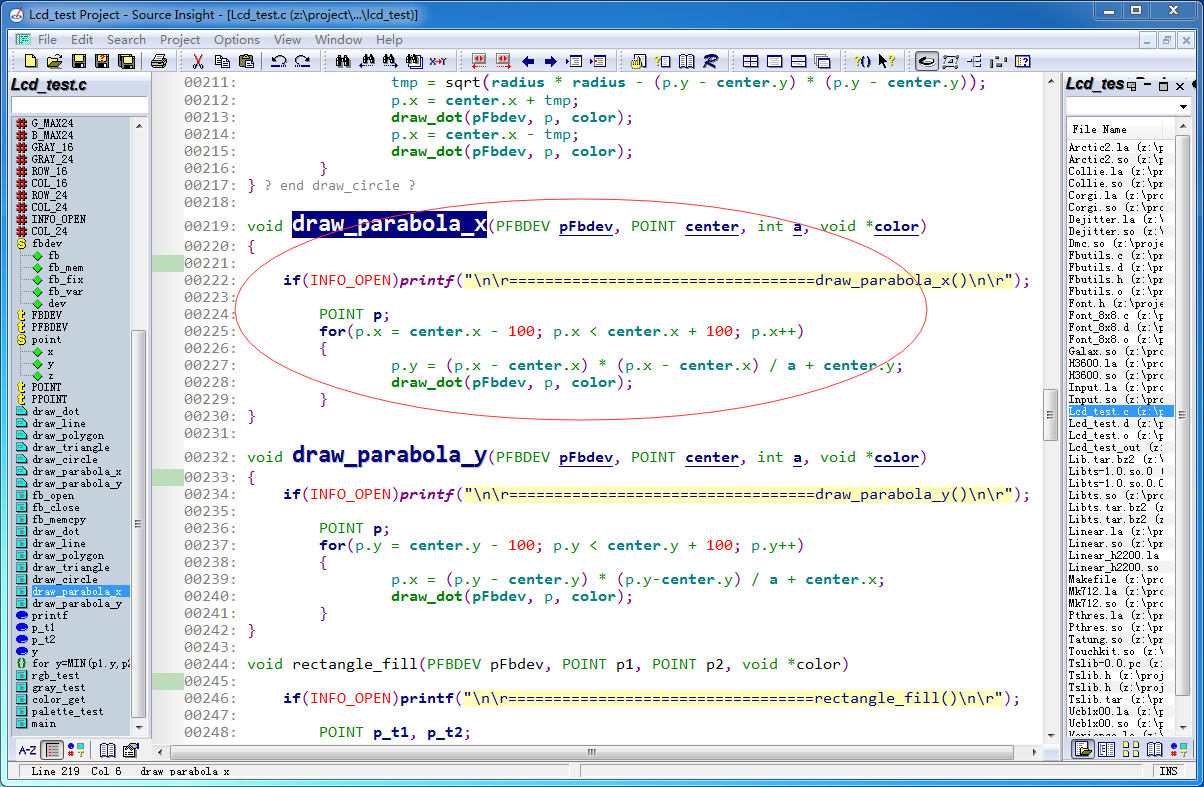
这个函数的作用是在LCD屏上画一个三角形，我们可以看到，该函数就只调用了一个画多边形函数draw\_polygon(pFbdev, 3, p, color)，里面的num参数直接传入3，这个不用说，三角形边数固定为3。然后draw\_polygon()函数又调用画线函数draw\_line(),draw\_line()又调用画点函数draw\_dot()，最终也就是画点。

第五个是void draw\_circle(PFBDEV pFbdev, POINT center, uint32\_t radius, void \*color)，如下：



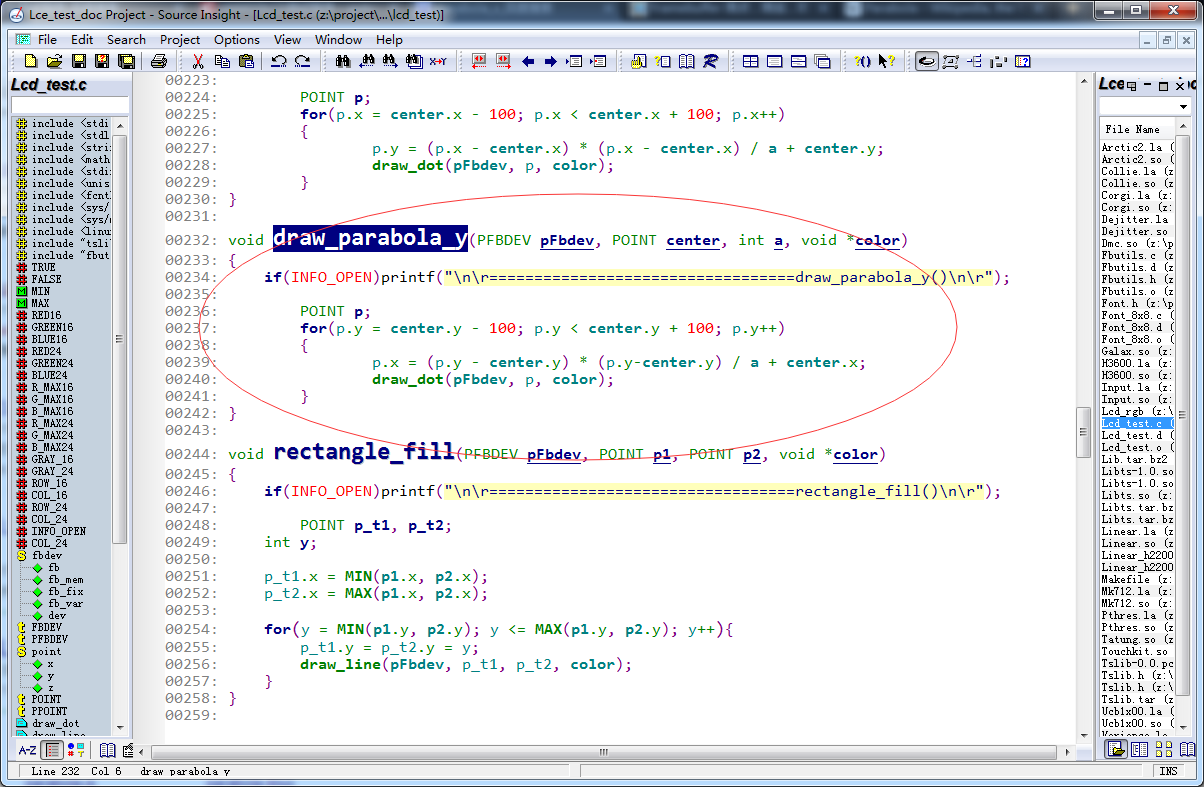
这个函数的作用是在LCD屏上画一个圆圈，我们可以看到，画圆圈分为两步进行，先画上半部分，再画下半部分。调用关系这里不再赘述。

第六个是void draw\_parabola\_x(PFBDEV pFbdev, POINT center, int a, void \*color)，如下：



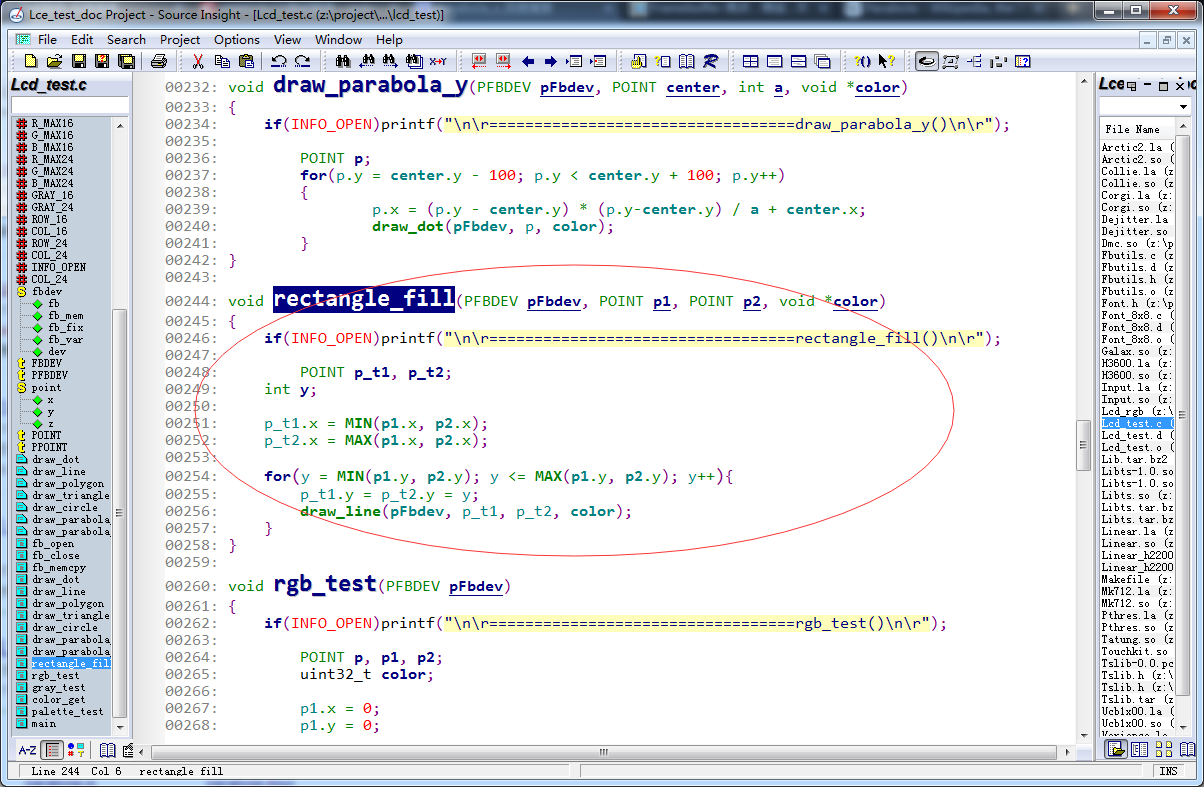
这个函数的作用是在LCD屏幕上画一条关于x轴对称的抛物线。

第七个是void draw\_parabola\_y(PFBDEV pFbdev, POINT center, int a, void \*color)，如下：



这个函数的作用是在LCD屏幕上画一条关于y轴对称的抛物线。

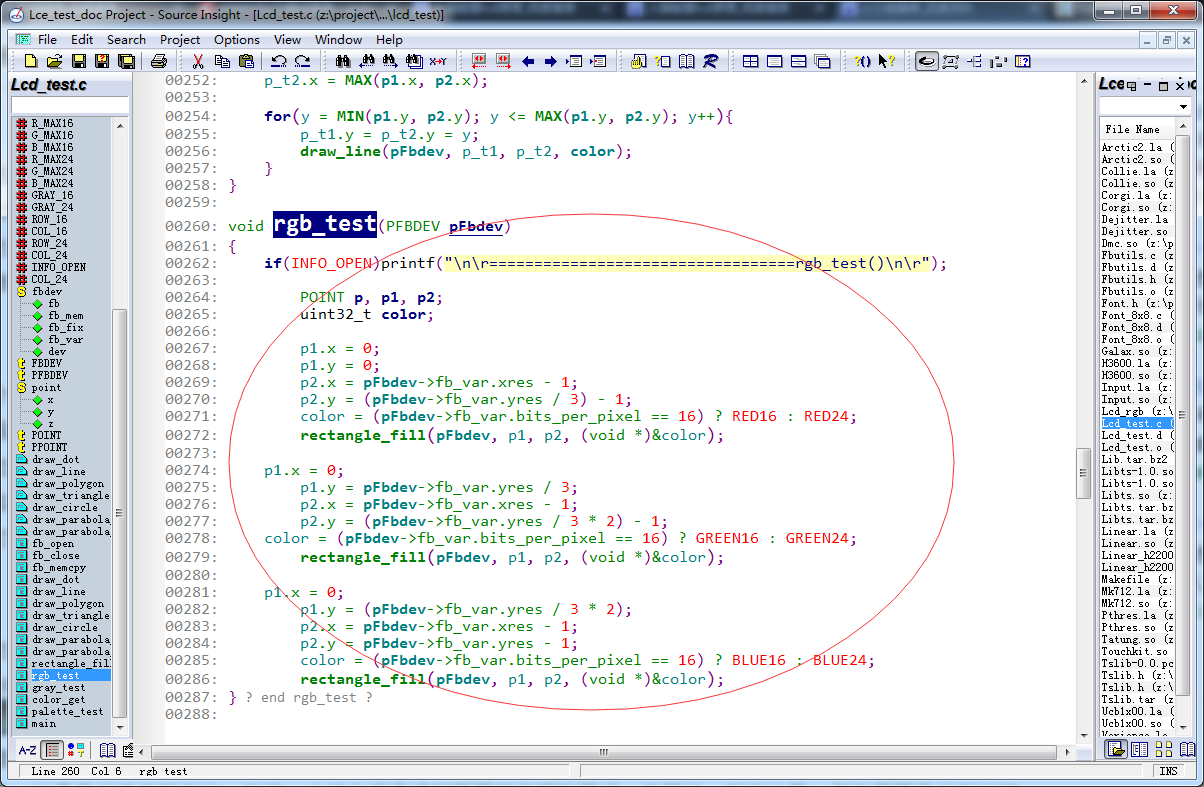
除了以上七个最基本的函数之外，我还在测试程序中给出了填充矩形的函数，如下：



这个函数的作用是根据传递的矩形的坐标和颜色参数在该矩形中以填充线条的方式填充颜色，说到底还是调用画点函数，这个用点填充的过程非常快，由于人眼存在“视觉残留”的现象，高速运动的画面在人脑中会形成短暂的印象，因此，我们是感受不到填充点的过程的。

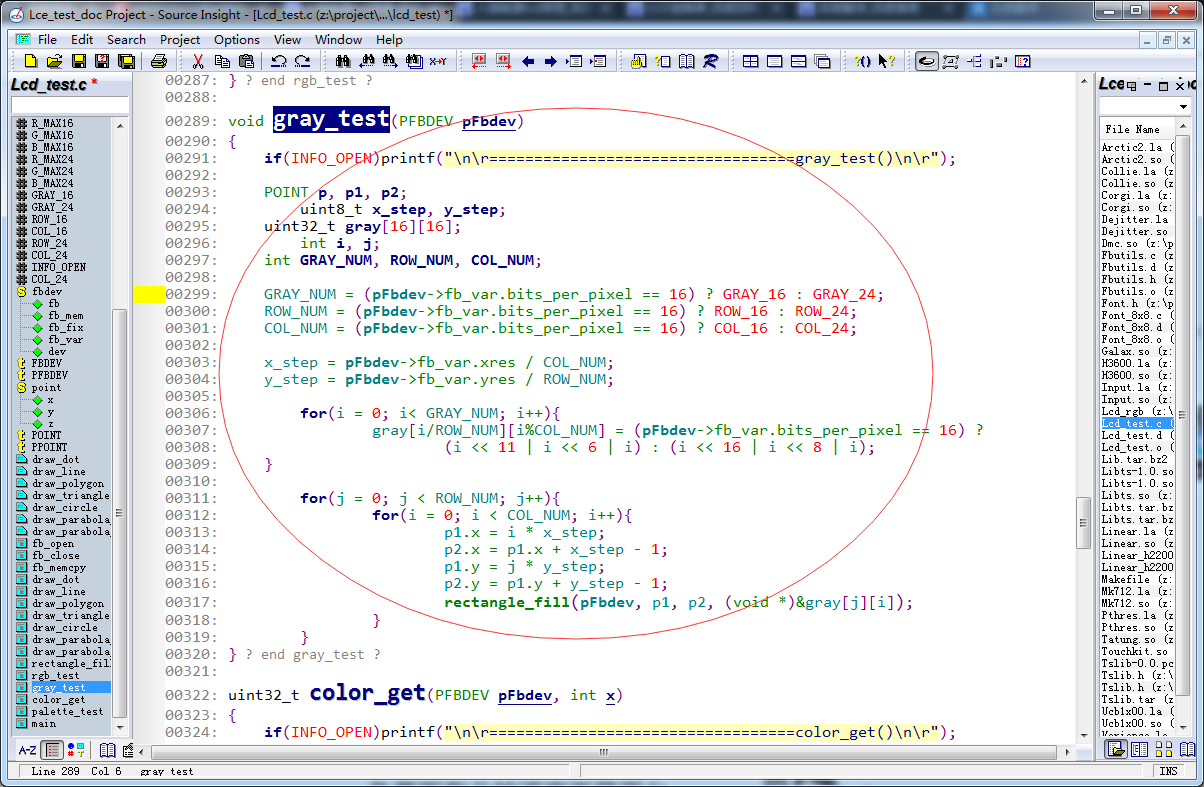
接下来我写了几个测试函数供用户调用，有：rgb\_test()、gray\_test()和palette\_test()。

第一个是void rgb\_test(PFBDEV pFbdev)，如下：



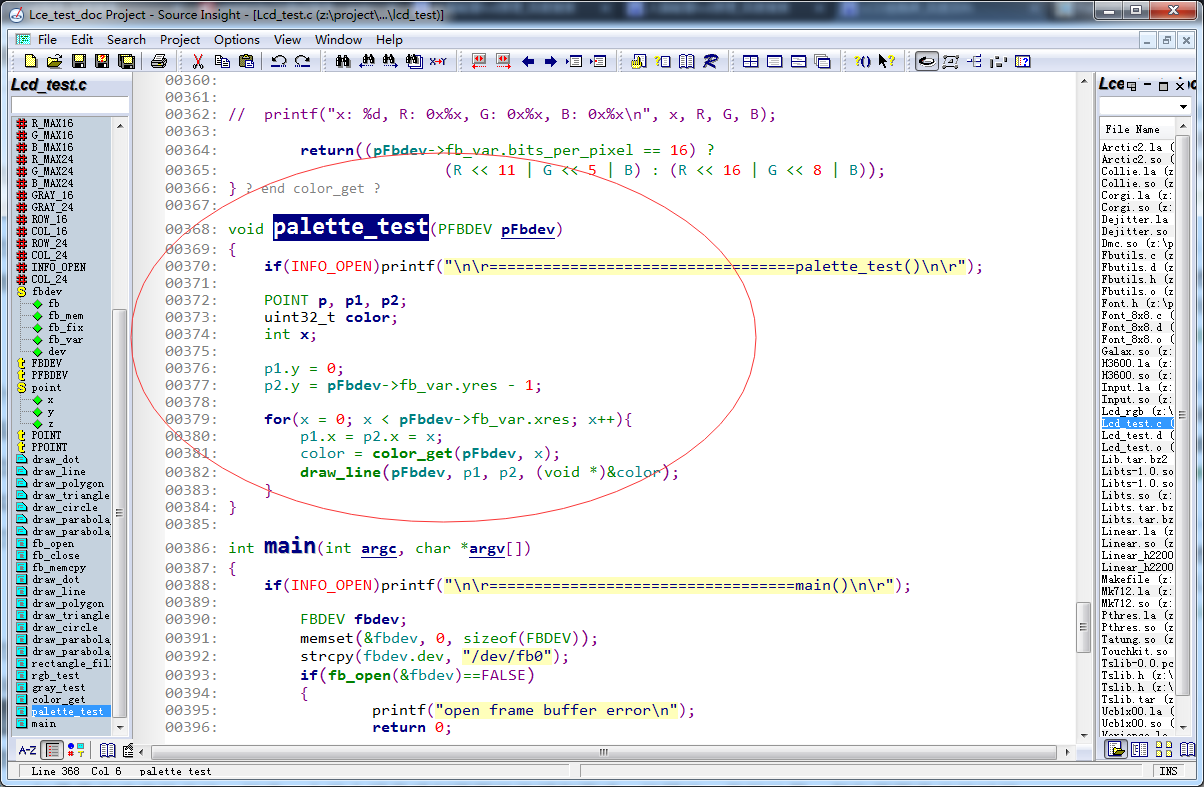
这个函数的作用是先将屏幕从左往右平均划分为竖着排列的三个矩形，第一个矩形填充纯红色，第二个矩形填充纯绿色，第三个矩形填充纯蓝色。

第二个是void gray\_test(PFBDEV pFbdev)，如下：



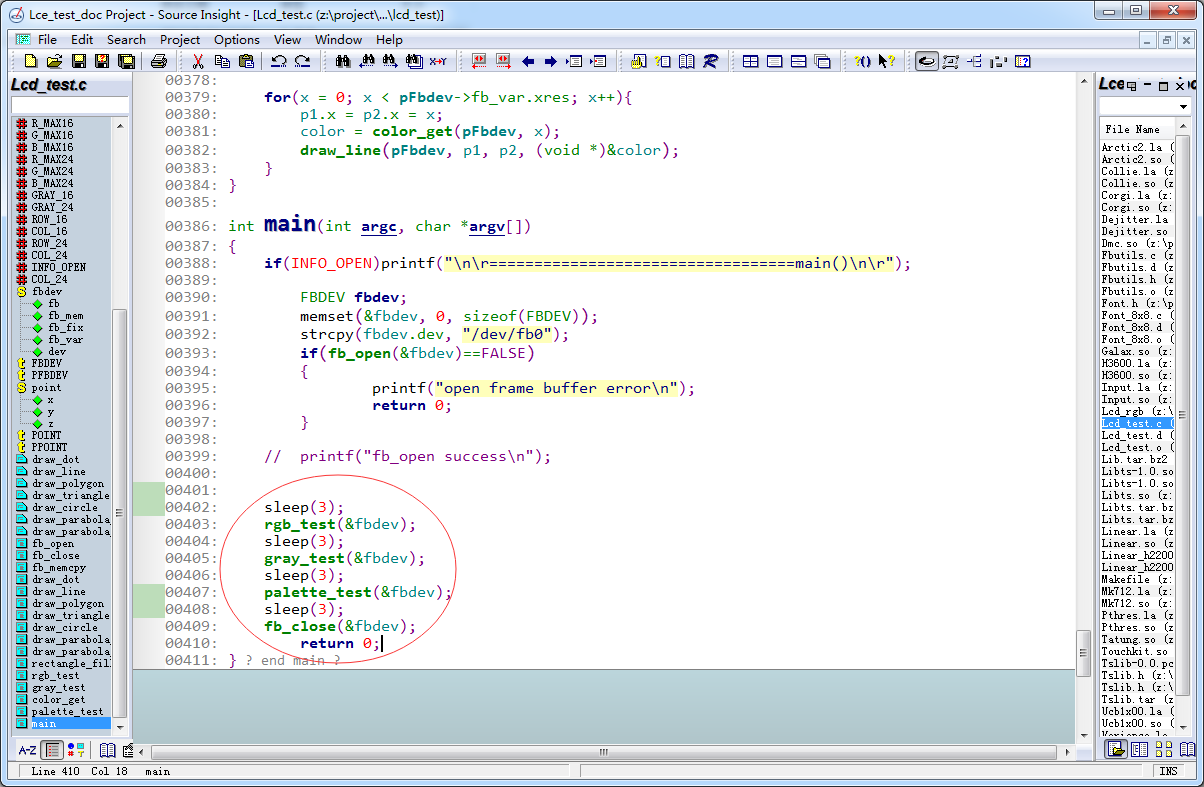
这个函数的作用是进行灰度测试。函数首先判断像素的位数，在我这里我们是16位，然后我把屏幕分成4行8列，这样的话一共有32个矩形，按灰度的不同等级将矩形用线条填充。最终我们在屏幕上看到的是32个不同颜色的矩形。

第三个是void palette\_test(PFBDEV pFbdev);如下：



这个函数进行调色板测试，过程是这样的，我先获取LCD屏的长宽，在一个for循环中逐步改变获取的颜色，然后每获取一种颜色就在LCD屏上用改颜色画一条竖线，直到画满整个屏幕。最终我们会在屏幕上看到一幅调色板画面。

我们来看一下main函数，在main函数中，首先打开/dev/fb0文件，如果打开失败则打印"open frame buffer error\n"，接下来依次调用三原色测试函数、灰度测试函数和调色板测试函数。如下：



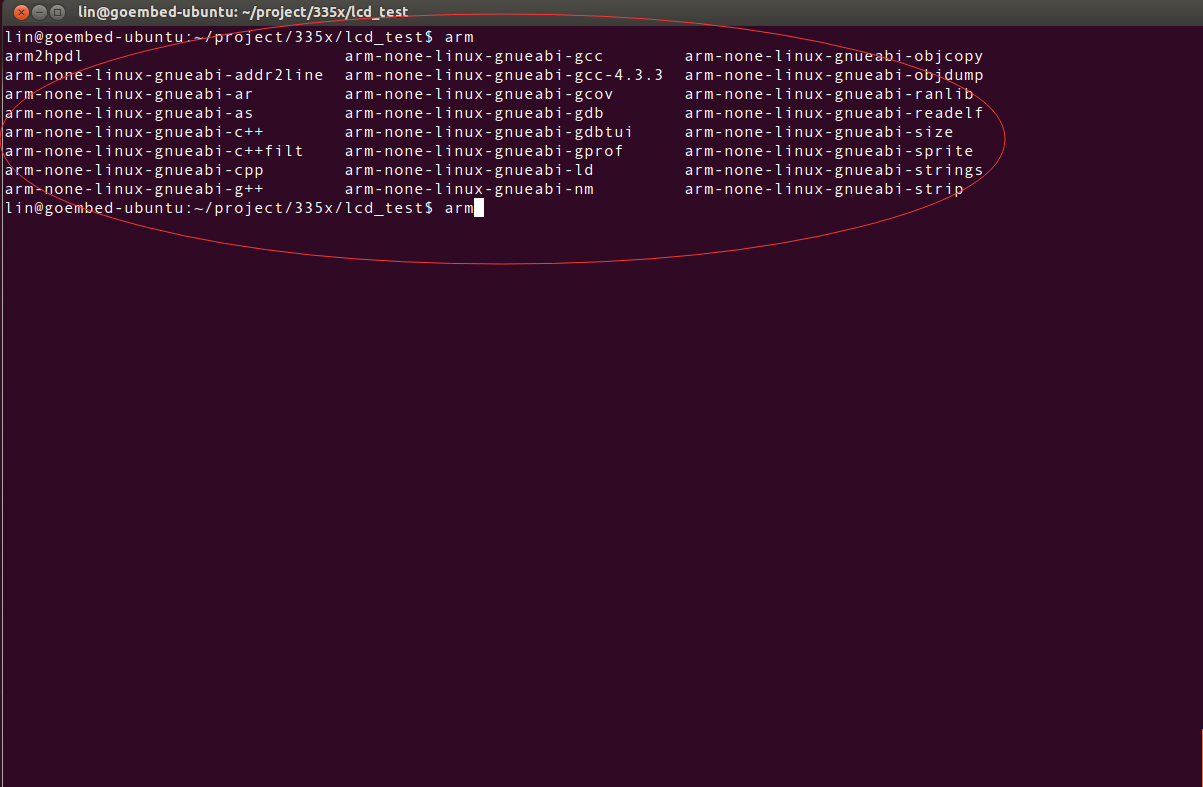
到这里，LCD屏测试程序基本分析完毕。在这里我们只是抛砖引玉，读者可以尝试写一下更高级的程序。下面我们把应用程序编译后生成的可执行文件复制到板子上测试一下。

### 四、编译测试程序

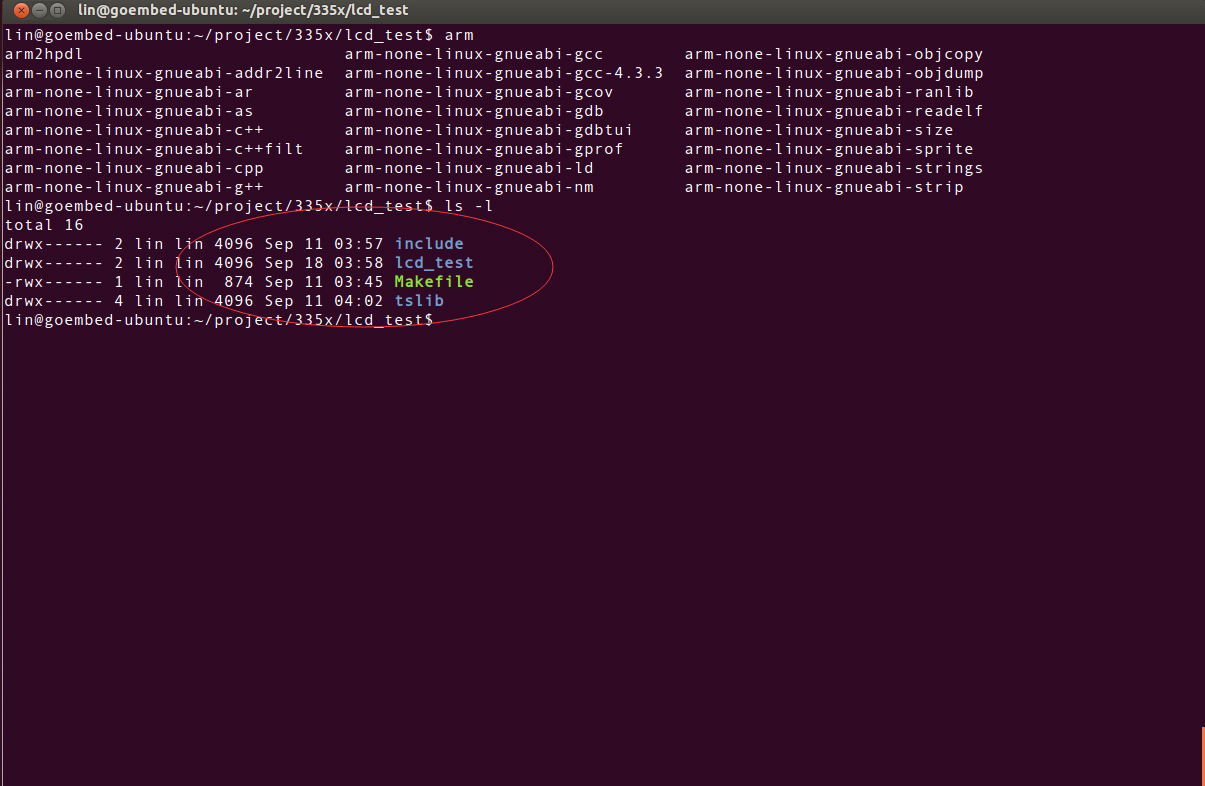
注：应用程序编译要采用arm-2009q1交叉编译

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。

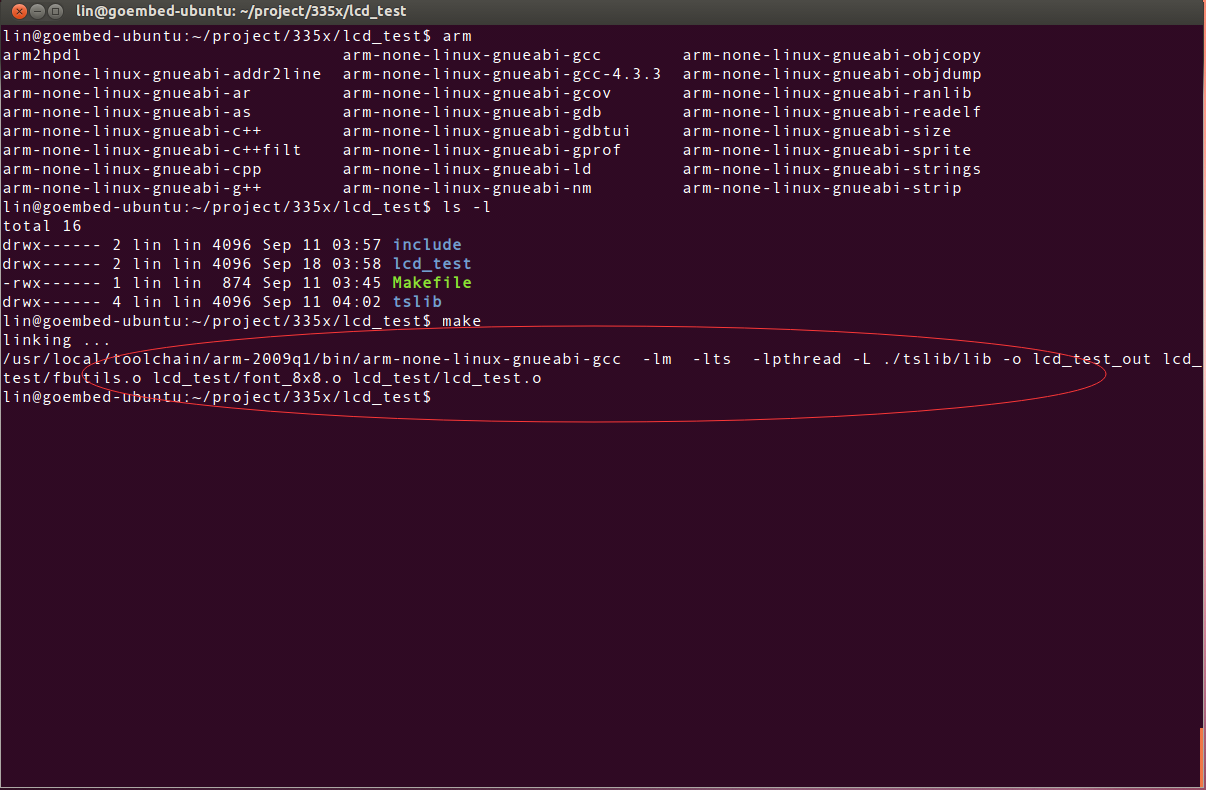
先导出交叉编译工具，如下图：



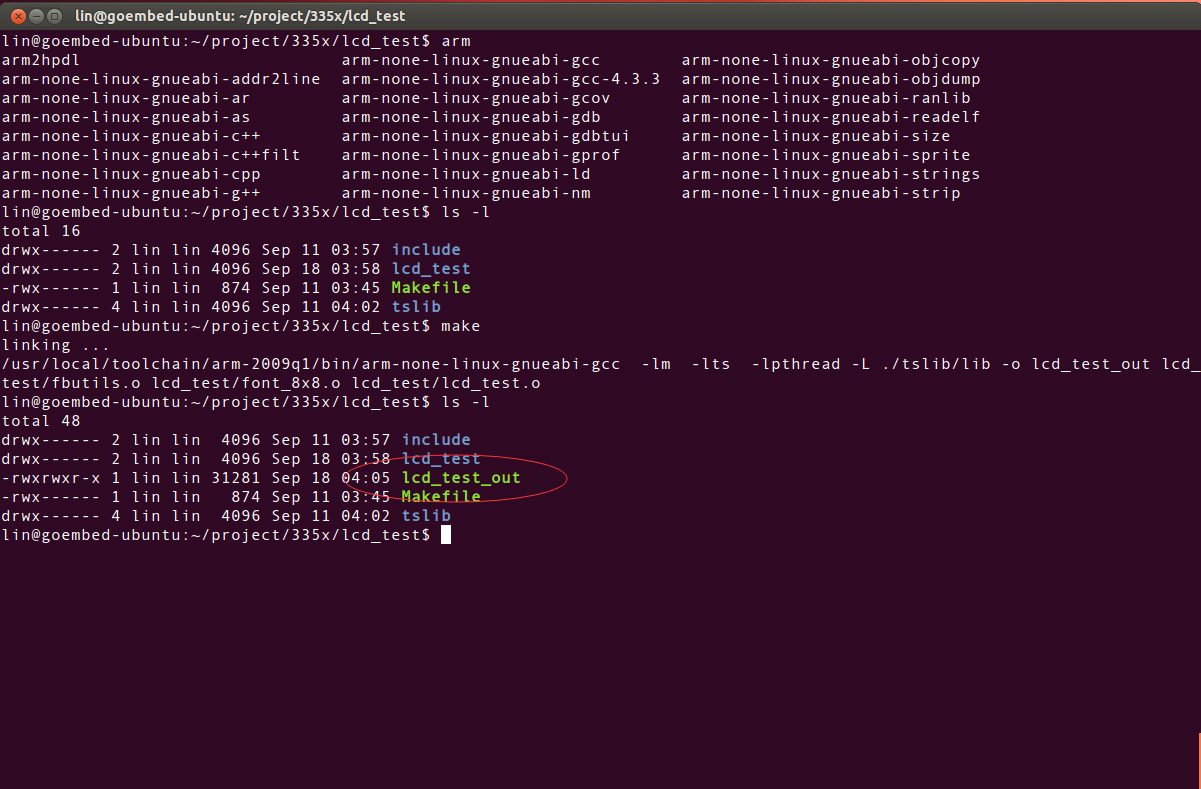
进入到测试文件夹：



执行make（若有残留旧文件可先执行“make clean”）：

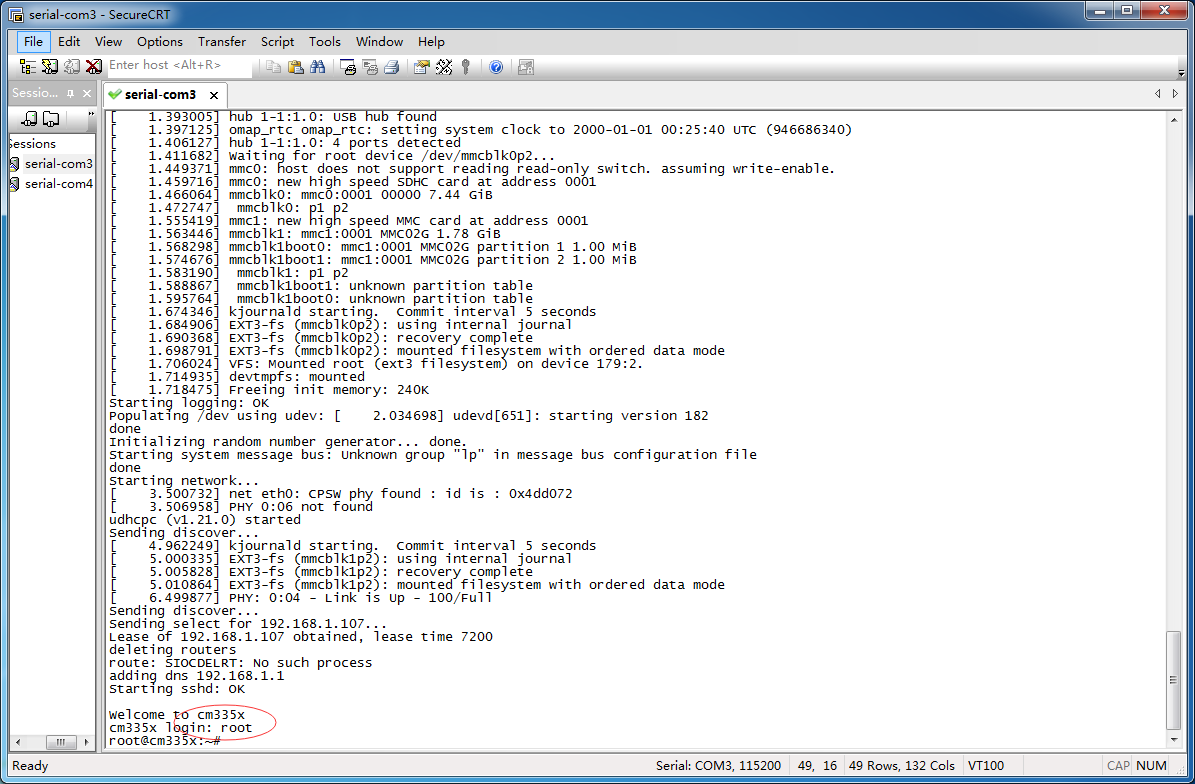


生成可执行文件lcd\_test\_out：



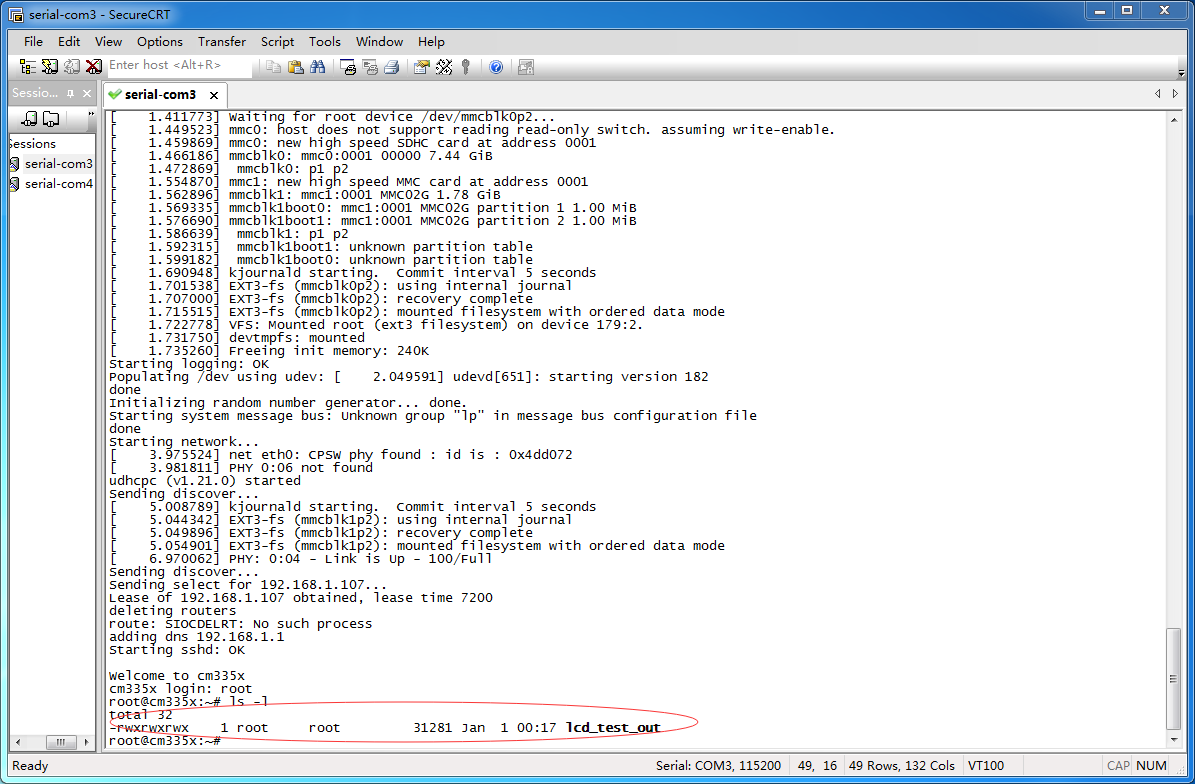
### 五、测试测试程序

上电开机后输入root登入SBC3358-B1A：



启动系统后找到该文件：

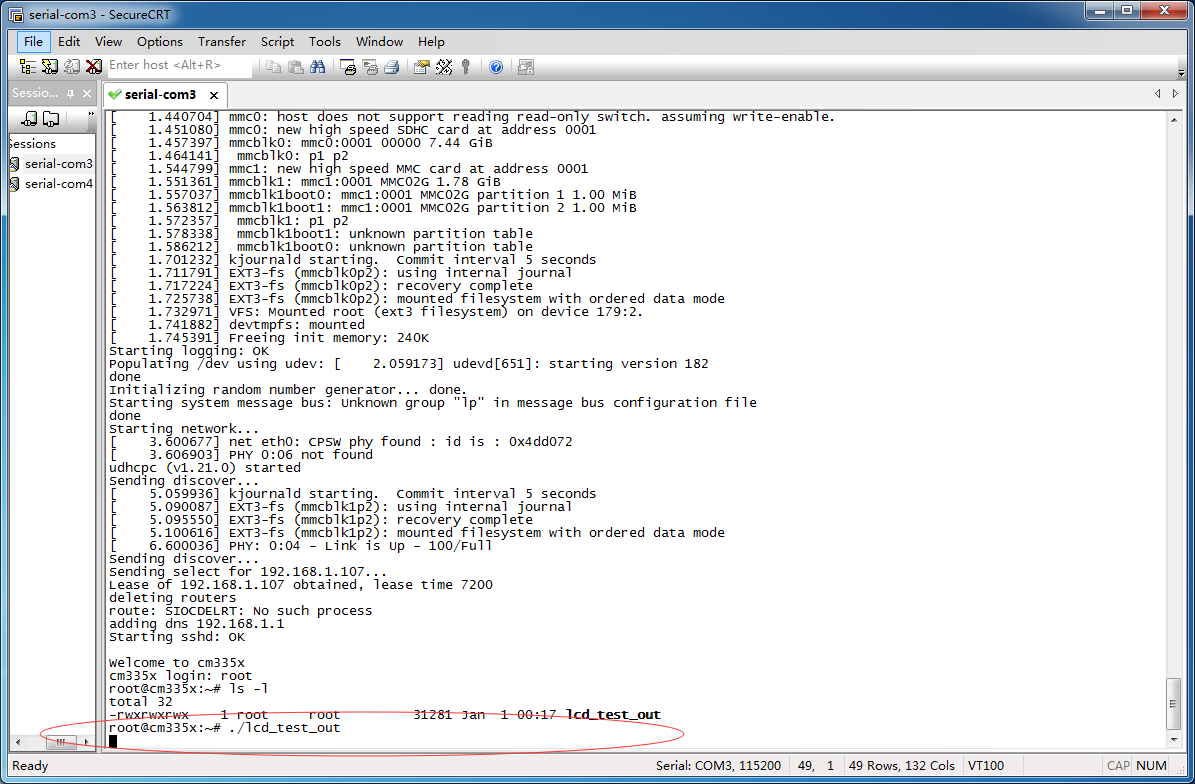
（该文件的位置根据用户启动系统的实际情况决定，在这里不赘述）



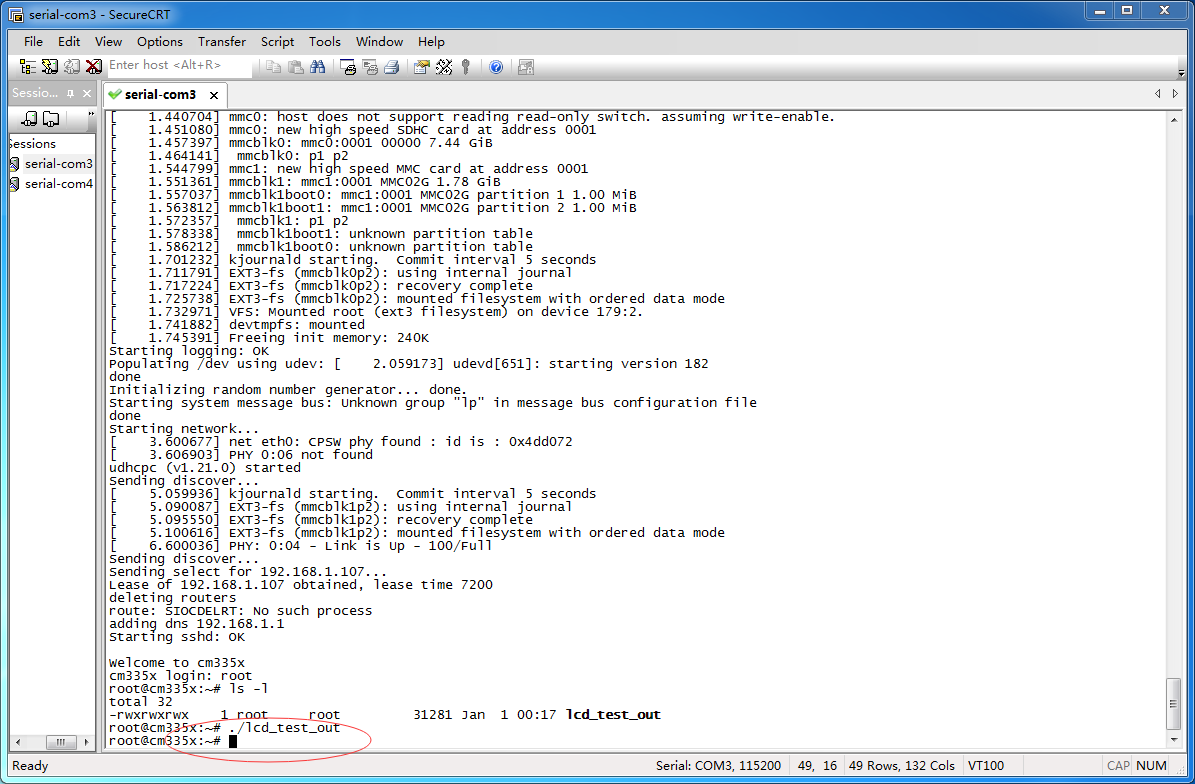
* 命令：

**./lcd\_test\_out**

输入以上命令执行该应用程序（成功执行黑色光标在闪并且LCD屏幕上显示测试画面）：

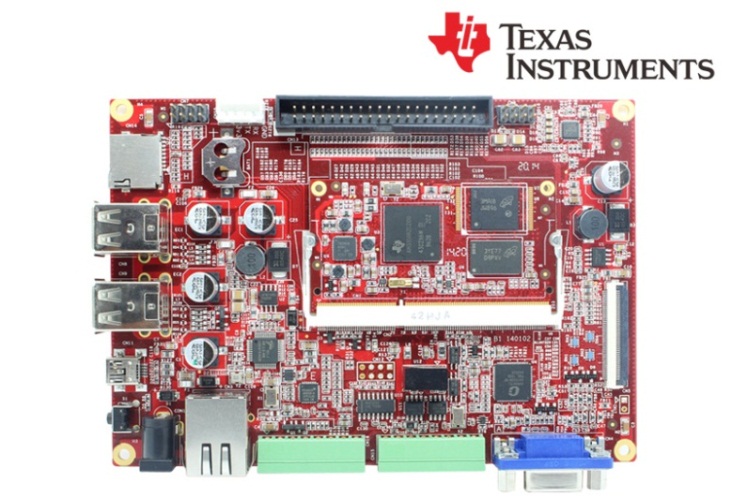


测试结束后程序自动安全退出：

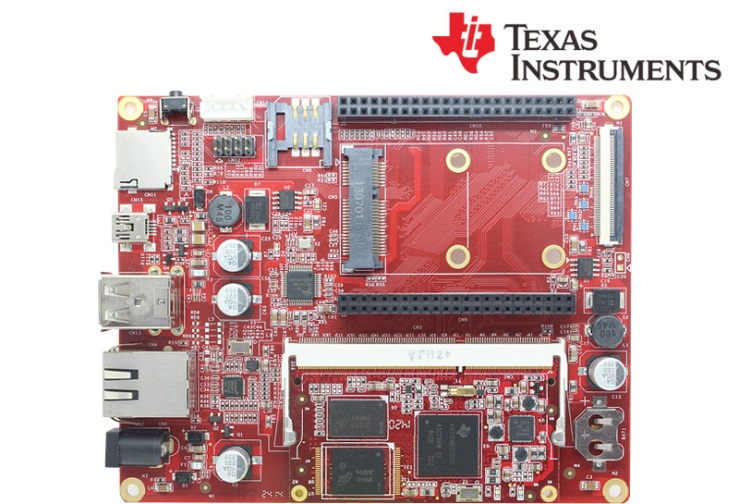


到这里，LCD屏幕测试已成功完成。

### 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.

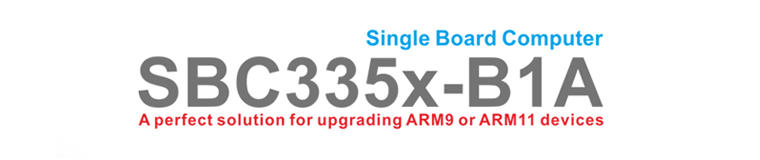
The B1A features 4 serial ports (including 2 RS232 and 2 TTL), 4 USB Host and 1 USB OTG, 1 Ethernet ports, CAN, RS485, Wiegand, VGA, LCD, Touch screen, Audio, ADC and more other peripherals.

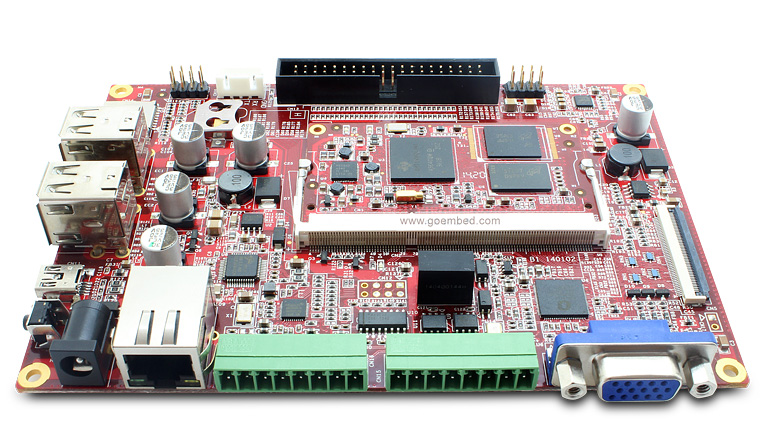
The B2A features 4 USB Host and 1 USB OTG, 1 Ethernet ports, LCD, Touch screen,RTC, and more other peripherals.

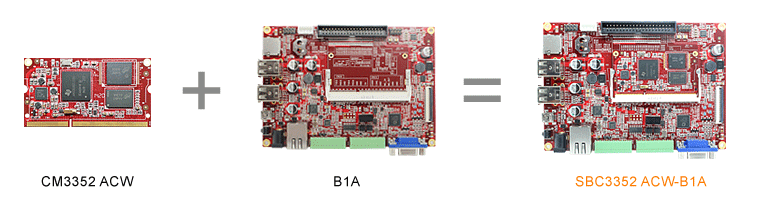
The SBC board targets a wide range of applications, including: HMIs, Digital Signage, POS, Data Terminal, Medical Devices, Navigation, Industrial Automation, Entertainment system, Thin Clients, Robotics, Game Console and much more.

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**

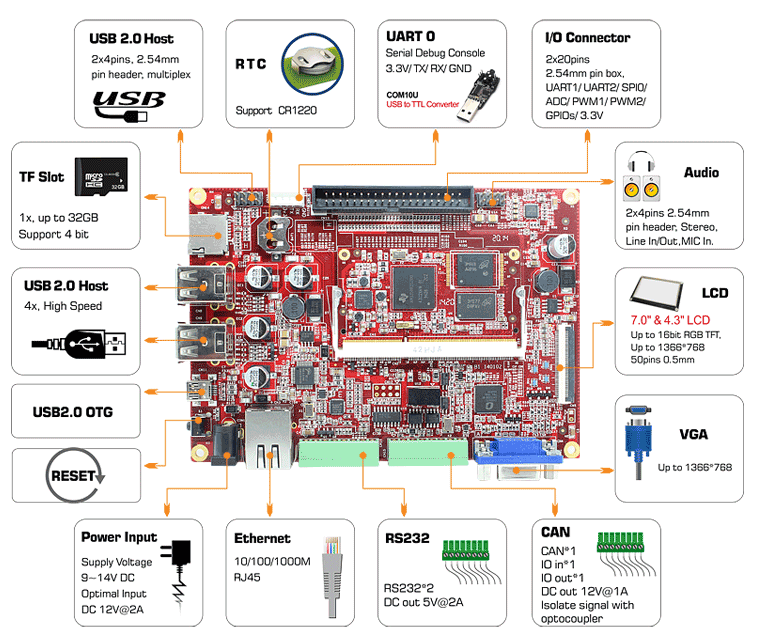
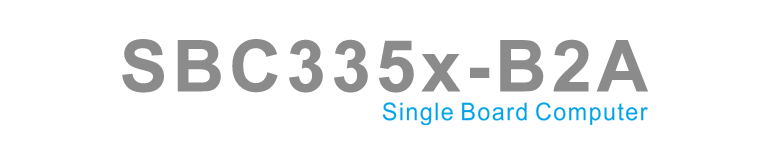
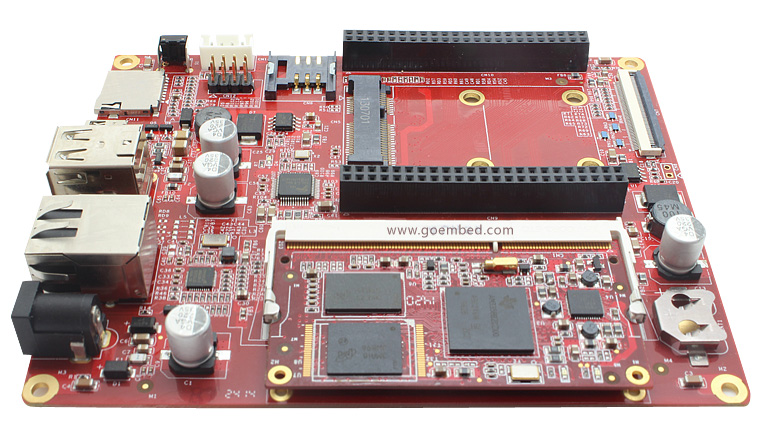
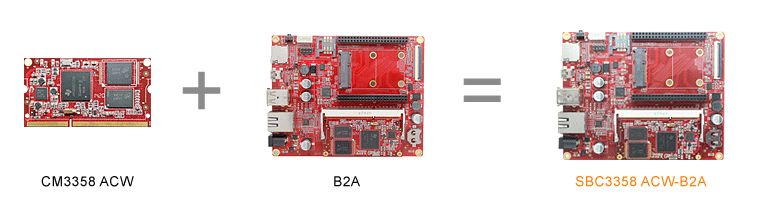


Figure 1 B1 Block Diagram







**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**

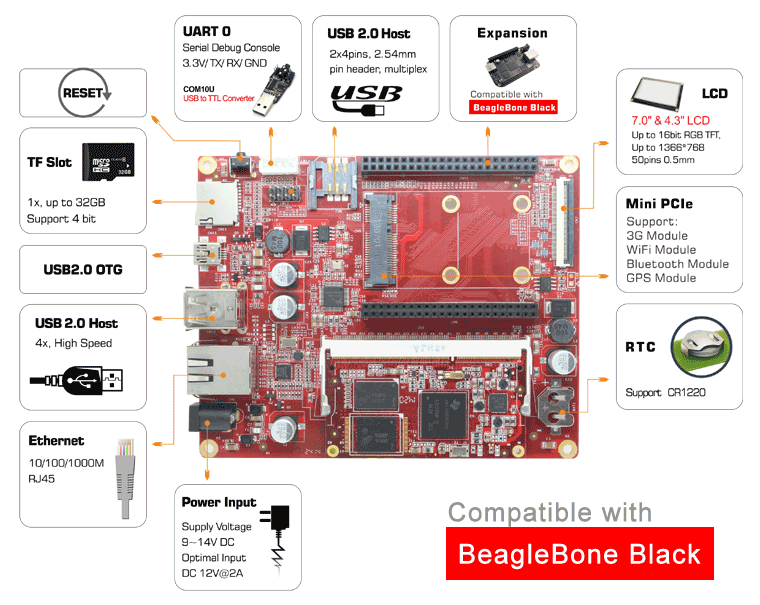


Figure 1 B2A Block Diagram

**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.

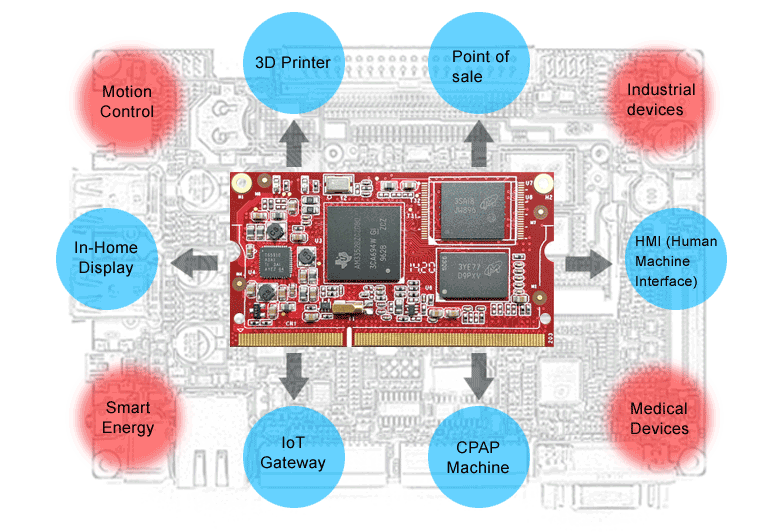
We are your trust worthy partner on ARM embedded design services and solutions.

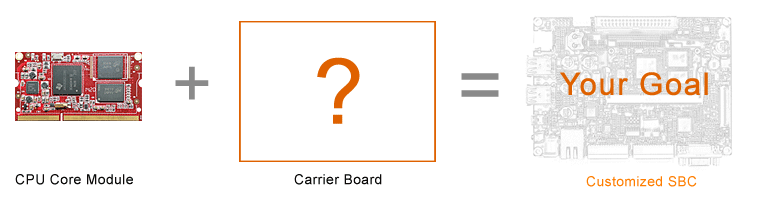
**More Carrier Boards**

Customized based on your needs!

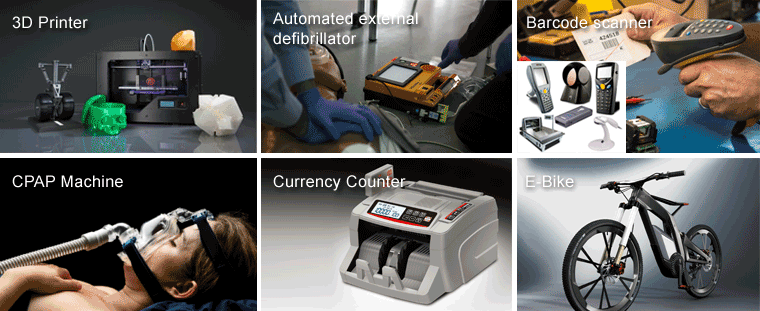
**ODM / OEM Services**

Bring your new products to market quickly





**Related end equipment**





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

