

|  |
| --- |
| Linux下按键测试程序开发 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** |

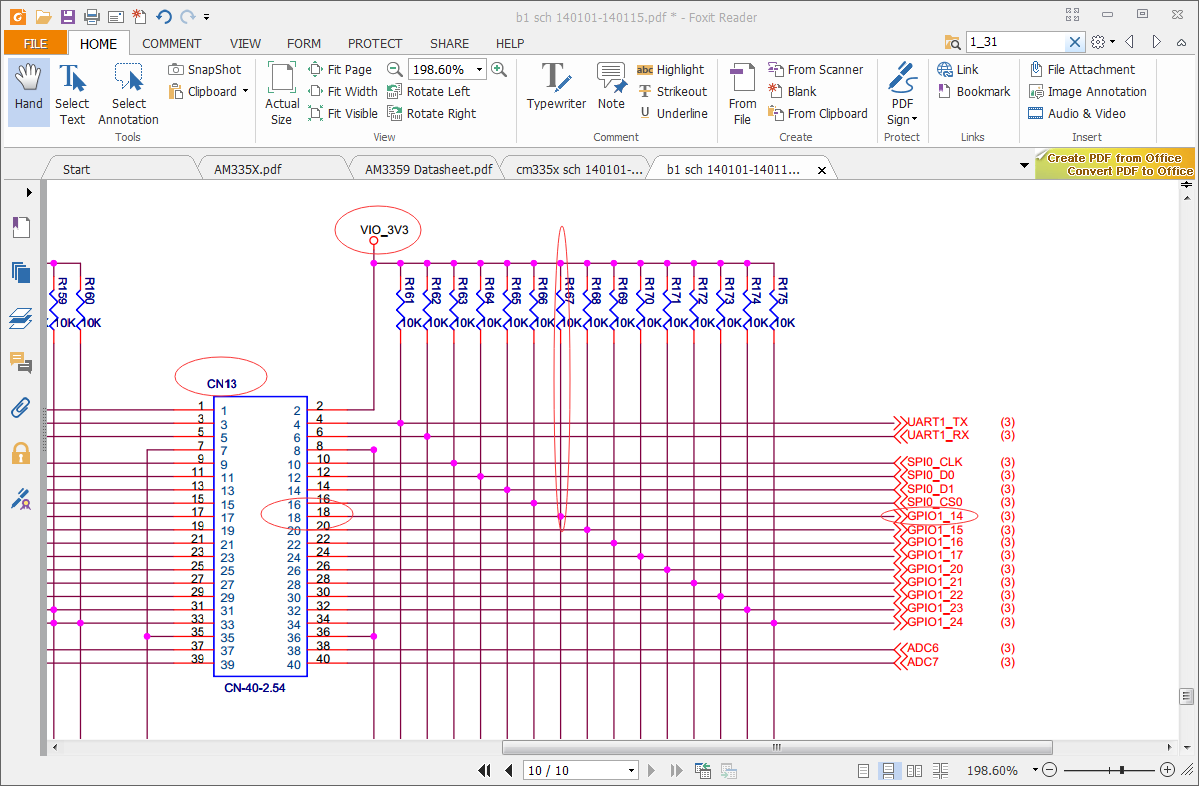
### 一、准备工作

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。

### 二、分析原理图

首先打开SBC3358-B1A的底板原理图，我们发现SBC3358-B1A的底板并没有设置普通按键（只有一个Reset按键），但是板子还是预置了非常多GPIO口的，我们随便选择其中一个来进行按键测试。先看下图：



我们可以发现，GPIO1\_14（类似的旁边还有很多，我们取一个做例子）通过CN13的pin18连接到核心板上，当GPIO1\_14引脚为高电平时pin18为高，当GPIO1\_14为低电平时pin18也为低，也就是说我们完全可以用GPIO1\_14来模拟一个按键。实际操作中我们可以使用一根导线接在pin18，当把pin18接地时模拟按键按下，当不接地时相当于未按下。沿着这个思路我们接着往下做。

### 三、修改代码

#### 1、修改“gpio\_keys\_pin\_mux”

由于SBC3358-B1A底板没有自带普通按键，因此出厂镜像也没有添加按键的平台设备，我们需要新增KEY的平台设备。该文件是：/home/goembed/335x/work/linjia/linux-3.2.0-psp04.06.00.08.sdk/arch/arm/mach-omap2/board-cm335x.c

打开board-cm335x.c，找到gpio\_keys\_pin\_mux结构体数组变量，增加要测试的GPIO1\_14，

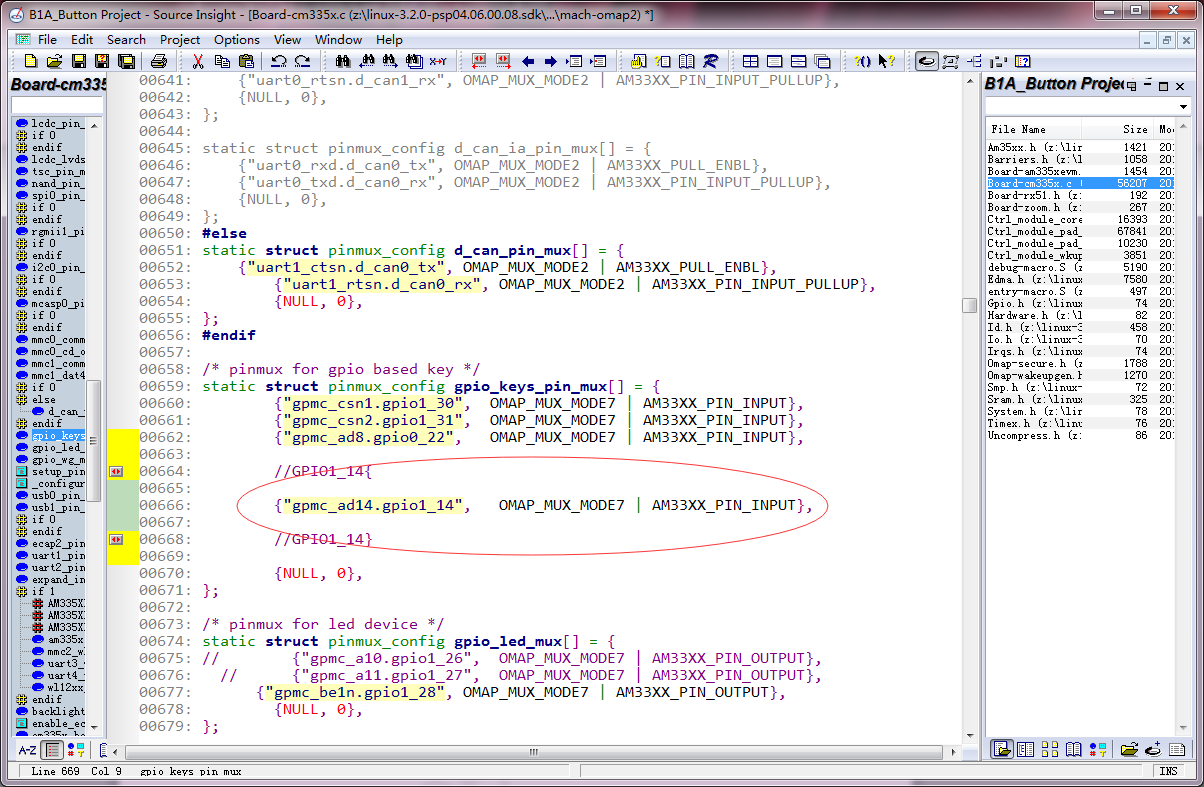
代码为：

**//GPIO1\_14{**

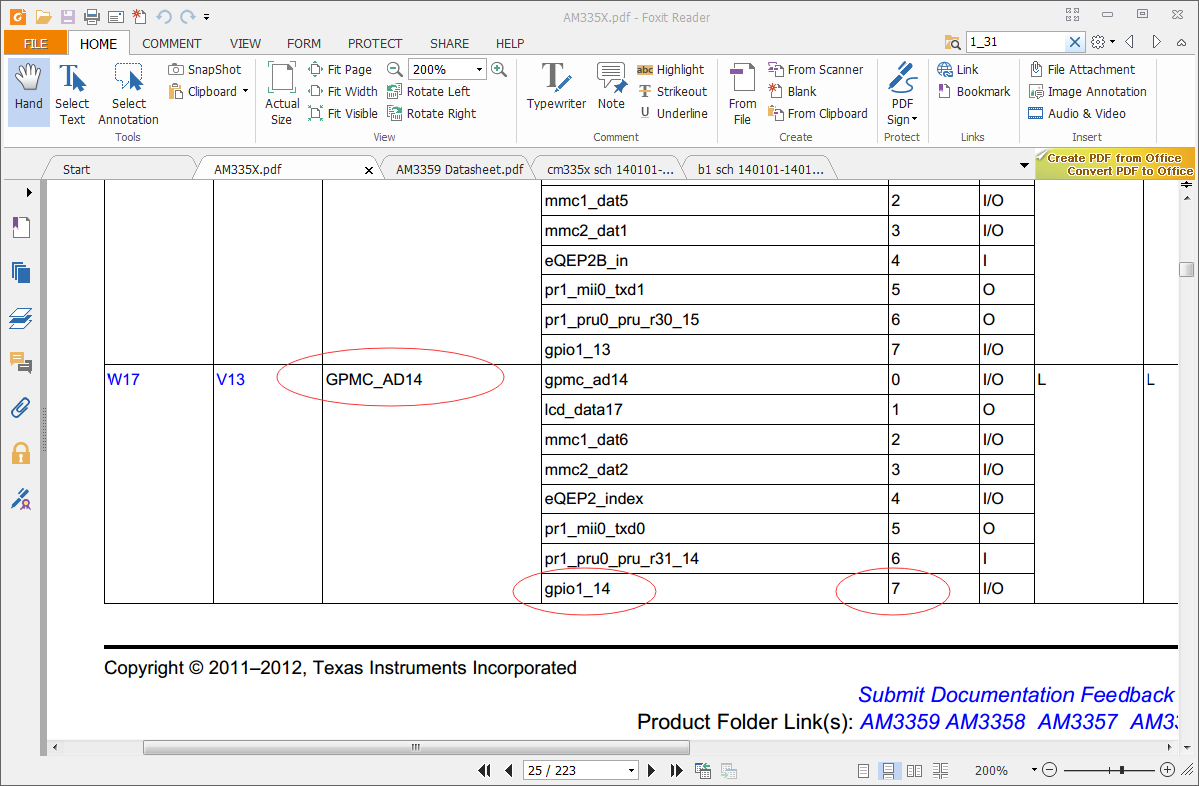
**{"gpmc\_ad14.gpio1\_14", OMAP\_MUX\_MODE7 | AM33XX\_PIN\_INPUT},**

**//GPIO1\_14}**

如下图：



其中，“gpmc\_ad14.gpio1\_14”是怎么得到的呢？我们可以查看技术手册：



从手册可以看出，由于很多GPIO口都有复用功能，“gpmc\_ad14.gpio1\_14”是指使用GPIO功能，也可以看出MODE应为MODE7,也就是“OMAP\_MUX\_MODE7 ”，然后设置为输入引脚“AM33XX\_PIN\_INPUT”。

#### 2、修改“devkit8600\_gpio\_buttons”

在“devkit8600\_gpio\_buttons”结构体数组变量中新增以下代码：

**//GPIO1\_14{**

**{**

**.code = 111,**

**.gpio = GPIO\_TO\_PIN(1, 14),**

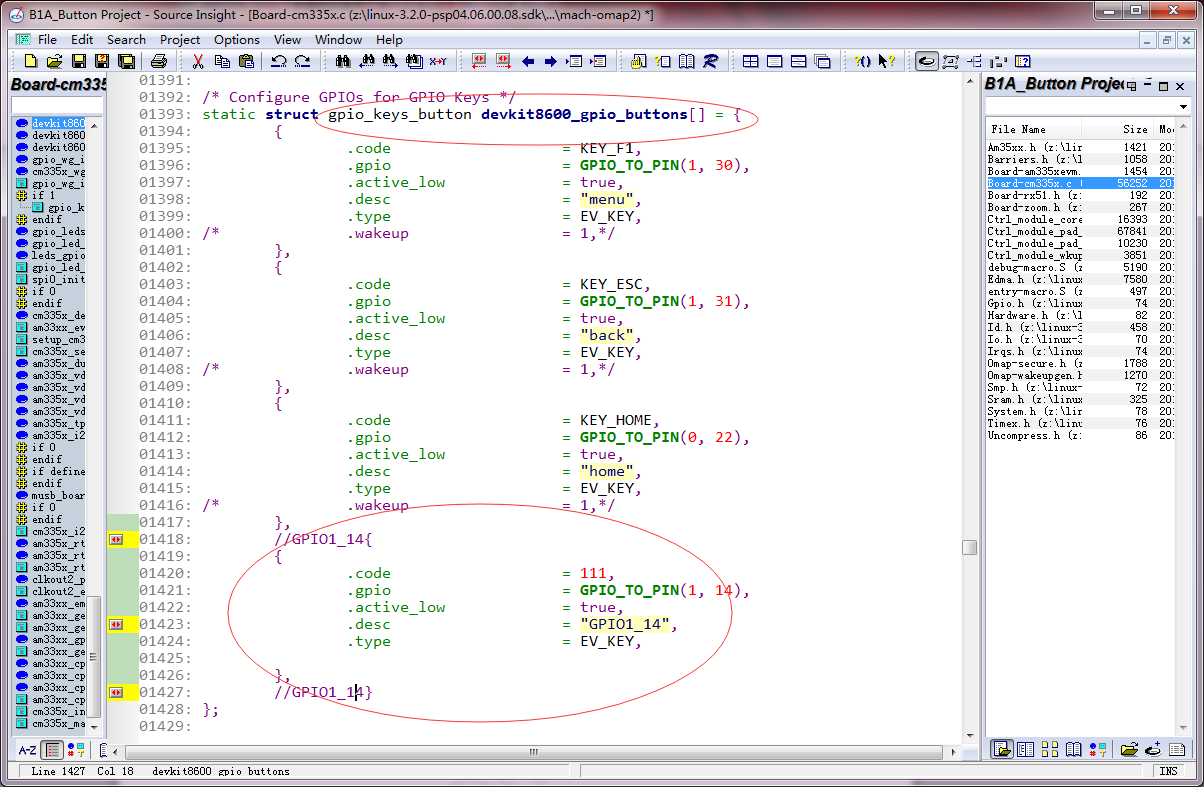
**.active\_low = true,**

**.desc = "GPIO1\_14",**

**.type = EV\_KEY,**

**},**

**//GPIO1\_14}**



其中code我们暂取为“111”，gpio则对应GPIO1\_14修改为GPIO\_TO\_PIN(1, 14)，active\_low为“true”表示低电平有效，desc暂取为“GPIO1\_14”，EV\_KEY表示按键事件。

#### 3、把gpio\_keys\_init()函数启用：（原本是#if 0）

**#if 1 //GPIO1\_14 #if 0**

**static void gpio\_keys\_init(int evm\_id, int profile)**

**{**

**int err;**

**setup\_pin\_mux(gpio\_keys\_pin\_mux);**

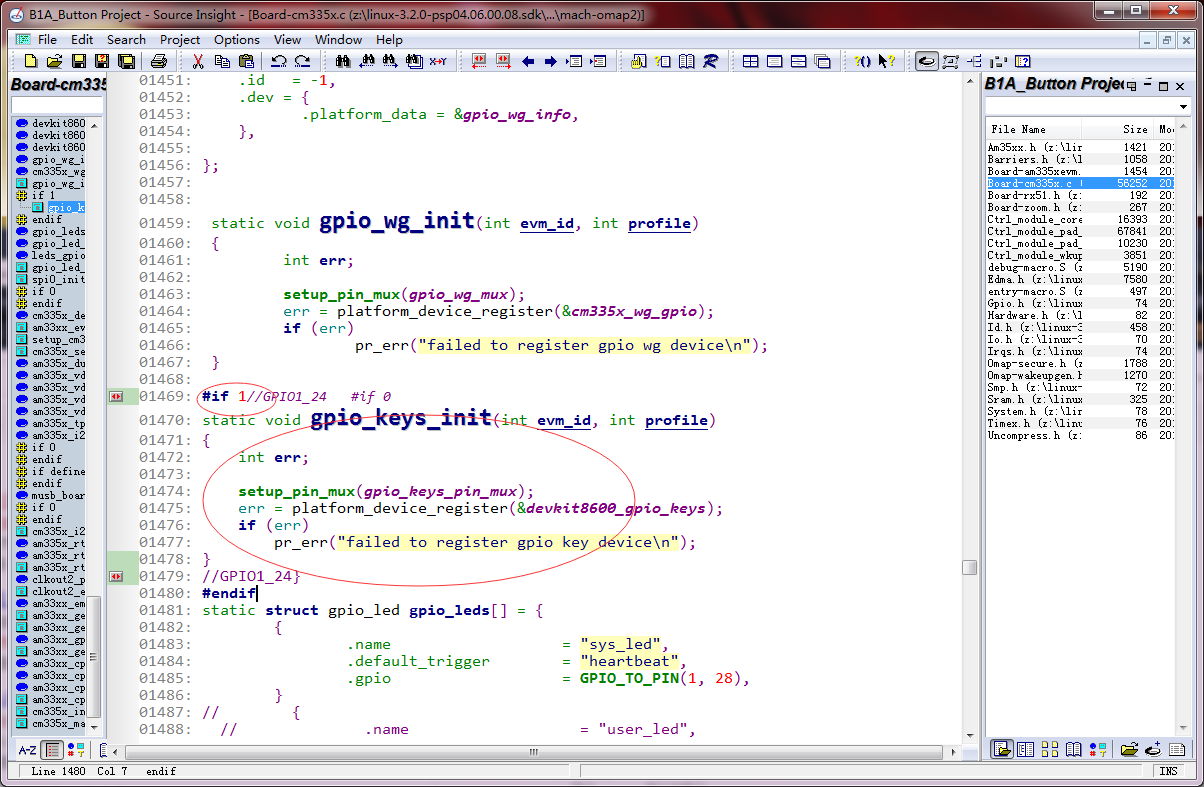
**err = platform\_device\_register(&devkit8600\_gpio\_keys);**

**if (err)**

**pr\_err("failed to register gpio key device\n");**

**}**

**//gpio1\_14}**



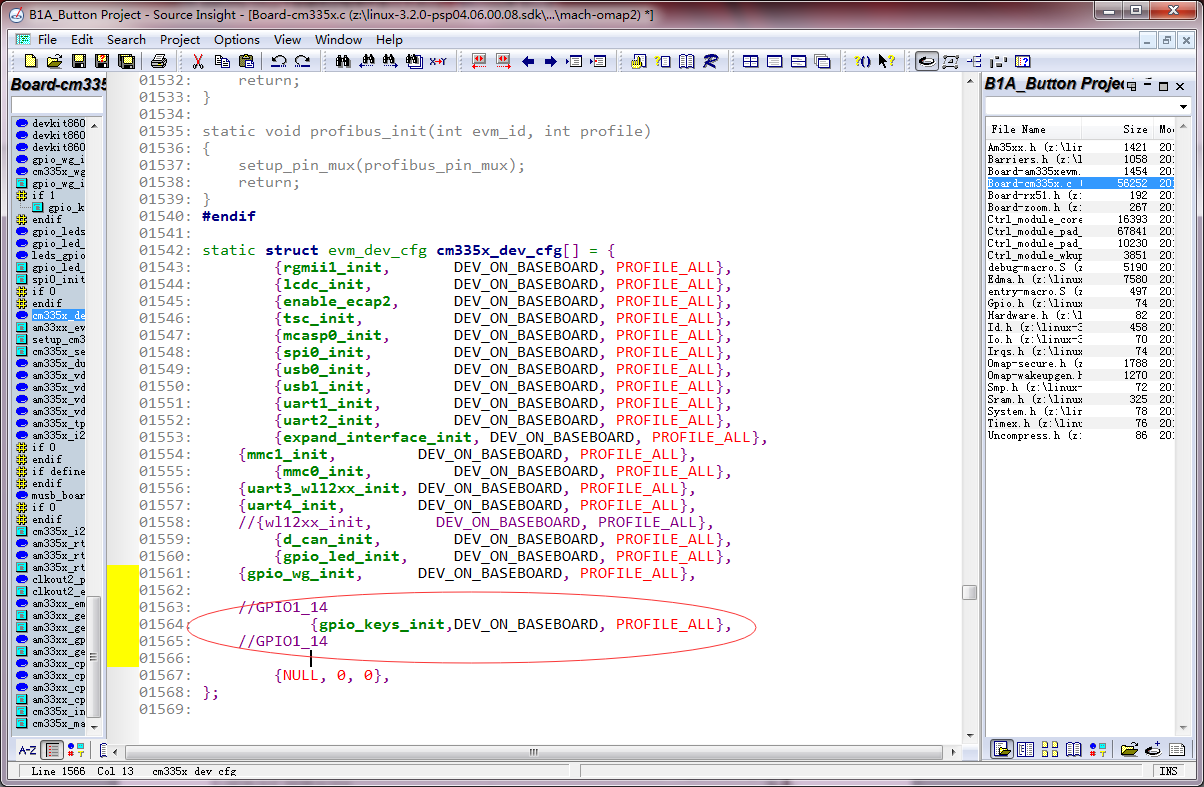
gpio\_keys\_init()函数最大的作用是注册KEY的平台驱动。我们上面定义的devkit8600\_gpio\_buttons在这里被调用，因此gpio\_keys\_init()这个函数后面必须调用。

#### 4、在cm335x\_dev\_cfg中加代码：

**//GPIO1\_14**

**{gpio\_keys\_init,DEV\_ON\_BASEBOARD, PROFILE\_ALL},**

**//GPIO1\_14**



到此，内核代码修改完毕，我们需要重新编译内核并烧写到启动卡。

### 四、编译新的内核

参考《TI AM335x Linux系统编译 v1.0.docx》重新编译内核镜像，这里只给出最重要的命令：

* **export PATH=$HOME/i686-arago-linux/usr/bin/:$PATH**
* **make O=am335x CROSS\_COMPILE=arm-linux-gnueabihf- ARCH=arm distclean**
* **make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- cm335x\_tisdk\_defconfig**
* **make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- uImage**

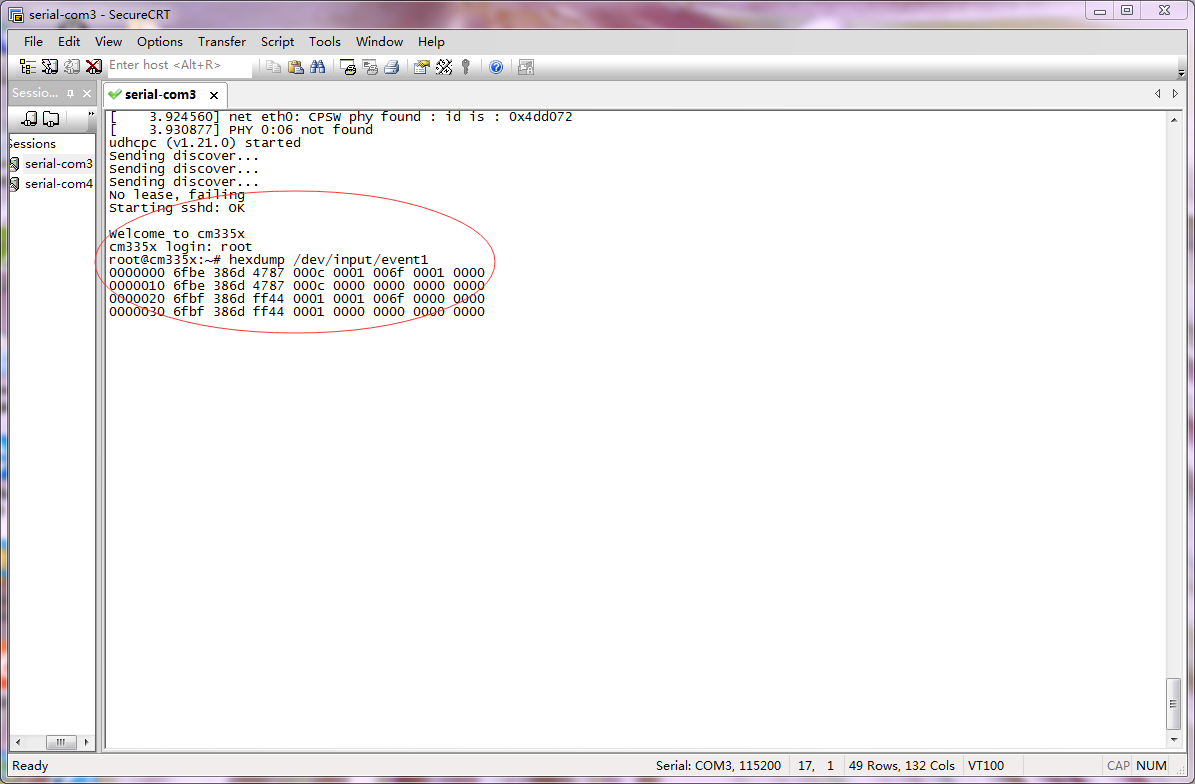
接下来用生成的新的uImage替换原来的uImage，然后重启系统。

### 五、测试新的内核

SecureCRT.exe中输入“root”登录，然后输入：

**hexdump /dev/input/event1**

当用导线将GPIO1\_14接地时可以看到如下打印信息（由于按键抖动，终端会打印出许多信息，后面由测试程序进行消抖）：



能看到以上信息说明新增的KEY没问题，接下来编写测试程序即可。

### 六、编写按键测试程序

在这里我写了一个测试程序，功能如下：当按键按下后，判断是长按（>1s）还是短按（<=1s），并且能把0.05s内产生的按键抖动过滤。

程序如下：

**#include <stdint.h>**

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

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

**#include <string.h>**

**#include <fcntl.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <signal.h>**

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

**#include <unistd.h>**

**int main(int argc, char \*\*argv)**

**{**

**int fd\_anjian;**

**struct input\_event t;**

**unsigned int backlight\_value=0x40;**

**int ret=0;**

**fd\_anjian = open("/dev/input/event1",O\_RDONLY);**

**if(fd\_anjian <=0 )**

**{**

**printf("error to open event0\r\n");**

**return 0;**

**}**

**double shijiancha=0;**

**double starttime;**

**double endtime;**

**while(1)**

**{**

**//printf("\n\r");**

**if(read(fd\_anjian,&t,sizeof(t)) == sizeof(struct input\_event))**

**{**

**if(t.type == EV\_KEY){**

**endtime=(t.time.tv\_sec+t.time.tv\_usec/1000000.0);**

**shijiancha=0;**

**if(t.code == 111 && (t.value==1) ){**

**starttime=(t.time.tv\_sec+(t.time.tv\_usec/1000000.0));**

**//printf("starttimr:%f\n\r",starttime);**

**//printf("t.time.tv\_usec:%lf\n\r",t.time.tv\_usec/1000000.0);**

**//printf("now //time:%f\n\r",((t.time.tv\_usec/1000000.0)+t.time.tv\_sec));**

**shijiancha=0;**

**}**

**if(t.code == 111 &&!t.value ){**

**shijiancha=0;**

**endtime=(t.time.tv\_sec+(t.time.tv\_usec/1000000.0));**

**//printf("endtimr:%lf\n\r",endtime);**

**}**

**shijiancha=endtime-starttime;**

**//printf("shijiancha:%lf\n\r",shijiancha);**

**if(shijiancha>1)printf("KEY:%d long %lf s\n\r",t.code,shijiancha);**

**if(shijiancha>=0.05&&shijiancha<=1)printf("KEY:%d short %lf s\n\r",t.code,shijiancha);**

**shijiancha=0;**

**}**

**}**

**}**

**close(fd\_anjian);**

**return 0;**

**}**

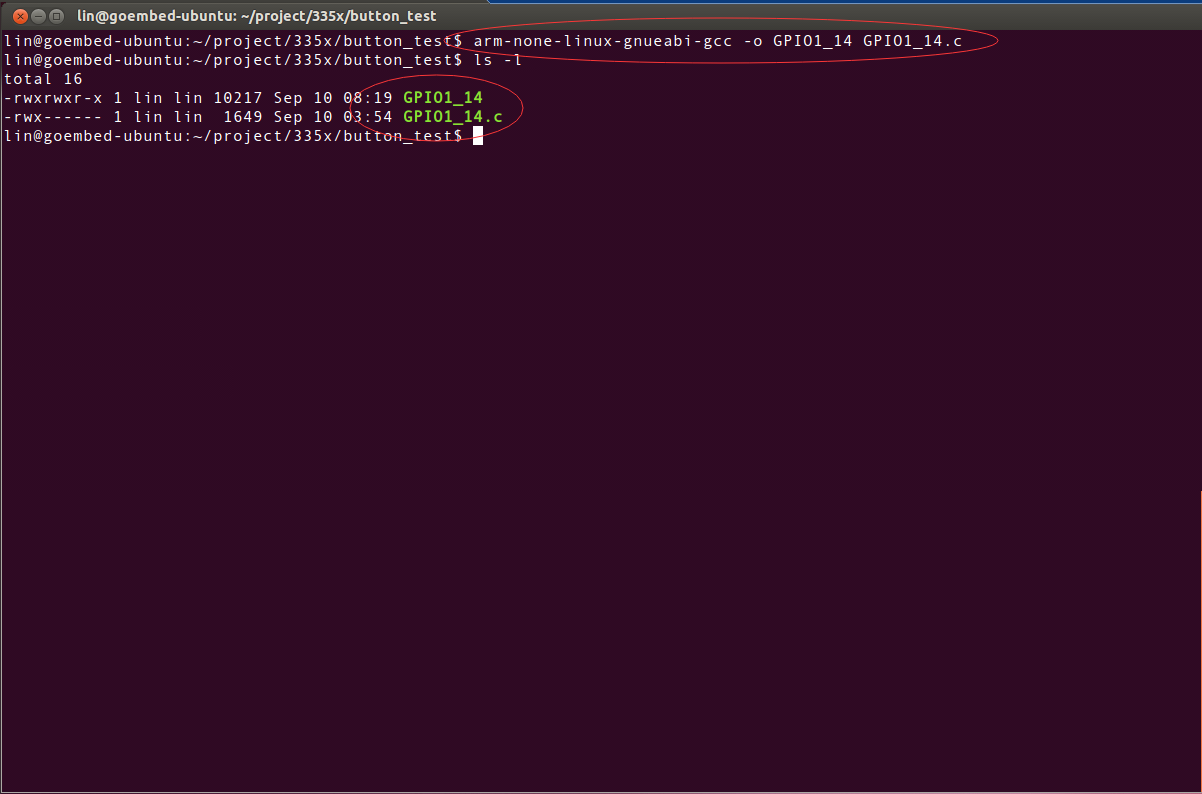
在这里我简单解释一下这个程序的流程：首先我们打开KEY对应的设备文件，也就是“/dev/input/event1”，然后进入循环等待按键按下，当有按键按下时我们先判断是不是键值为“111”，是的话记录下当前系统的时间，然后判断按键是否弹起，如果弹起则记录下弹起时的系统时间，我们用弹起时的系统时间减去按下时的系统时间得到时间差，然后判断该时间差，如果是小于0.05s的我们认为是抖动，如果是大于等于0.05秒且小于等于1s的我们认为是短按，如果是大于1s的我们认为是长按（是“长”是“短”用户可自定义，这里只是举个例子）。

注：应用程序编译要采用arm-2009q1交叉编译。（后续补上）

* 命令：

**arm-none-linux-gnueabi-gcc -o GPIO1\_14 GPIO1\_14.c**

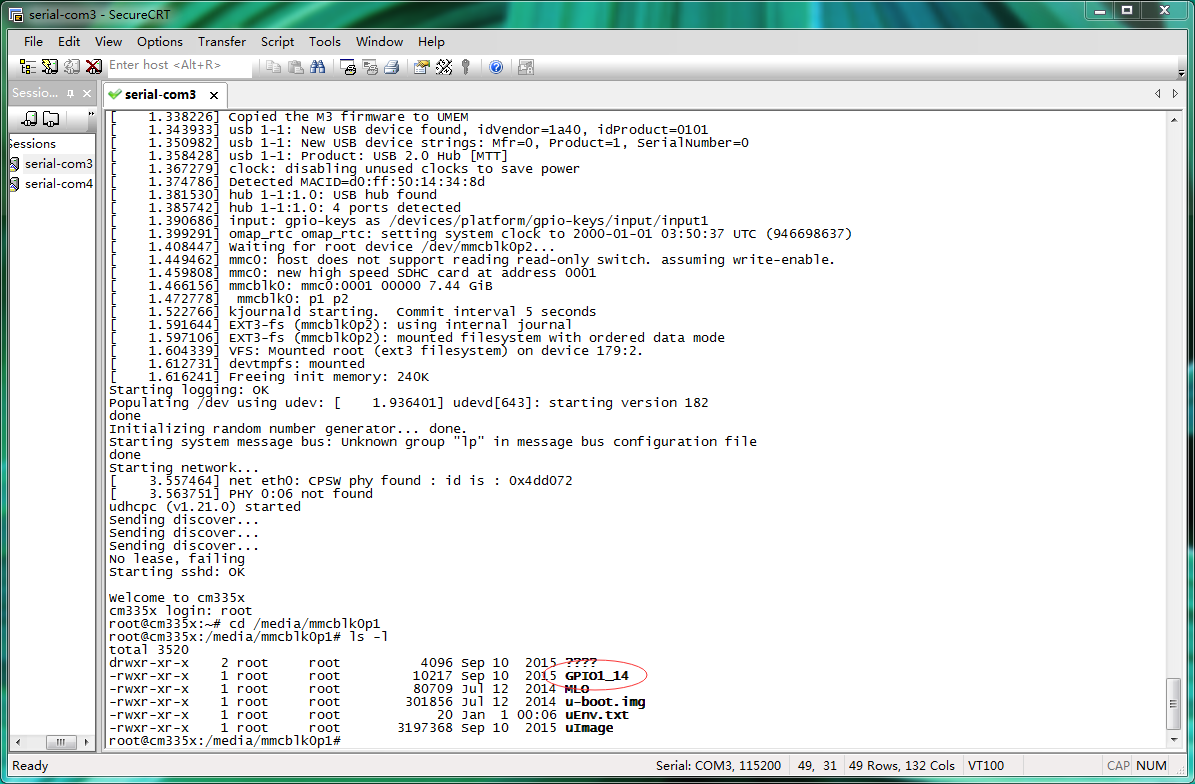
编译完成后生成可执行文件GPIO1\_14,如下图：



### 七、测试按键

将生成的可执行文件GPIO1\_14拷贝到TF卡中，启动系统后找到该文件：

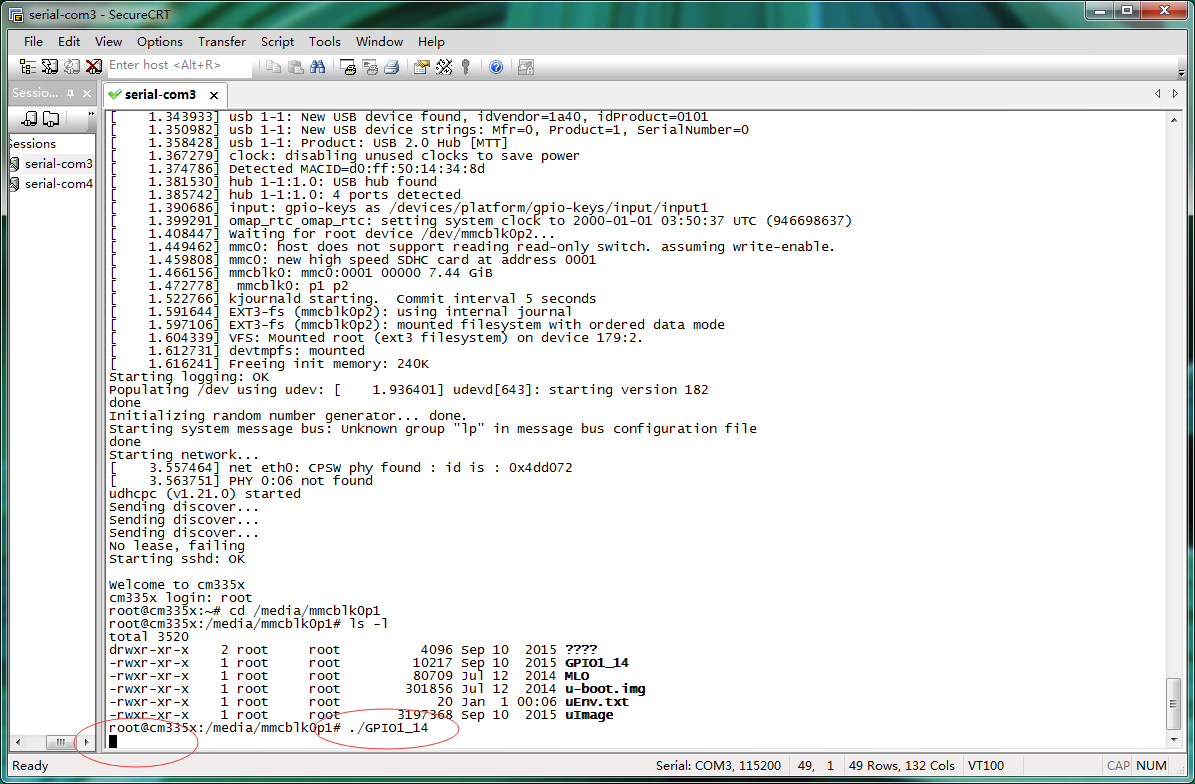
（该文件的位置根据用户启动系统的实际情况决定，在这里不赘述）



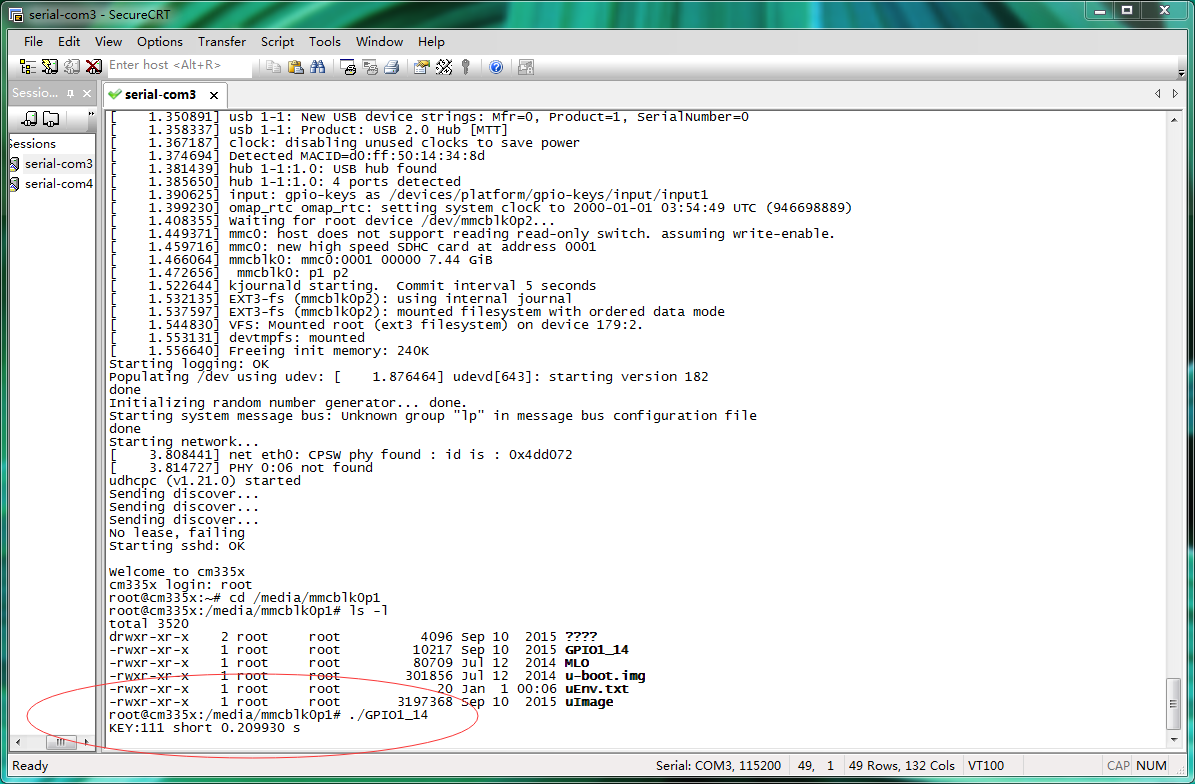
* 命令：

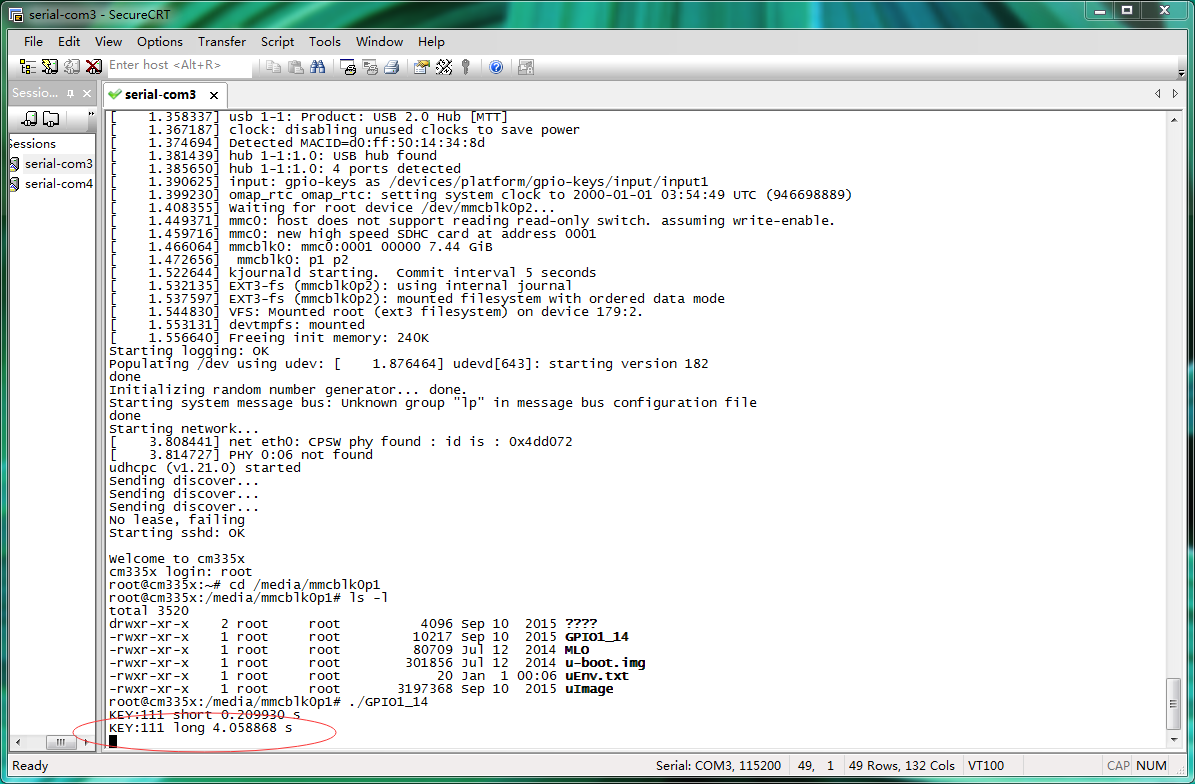
**./GPIO1\_14**

输入以上命令执行该应用程序（成功执行黑色光标在闪）：



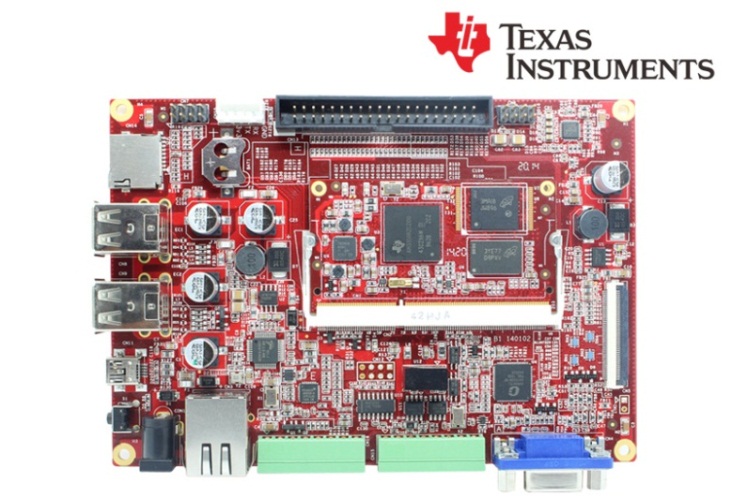
短按时如下：

长长按时如下

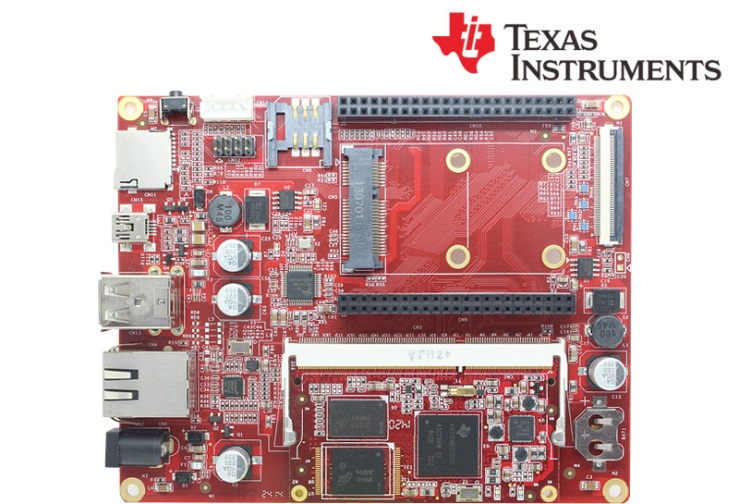


通过以上截图可以发现，按键的抖动被很好地过滤。该测试程序可以精确判断长按和短按。

### 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.

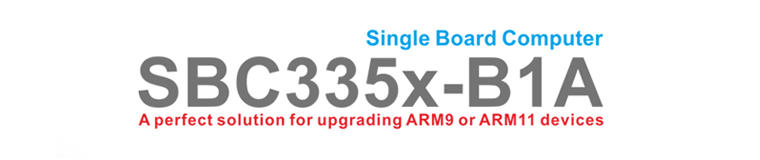
The B1A features 4 serial ports (including 2 RS232 and 2 TTL), 4 USB Host and 1 USB OTG, 1 Ethernet ports, CAN, RS485, Wiegand, VGA, LCD, Touch screen, Audio, ADC and more other peripherals.

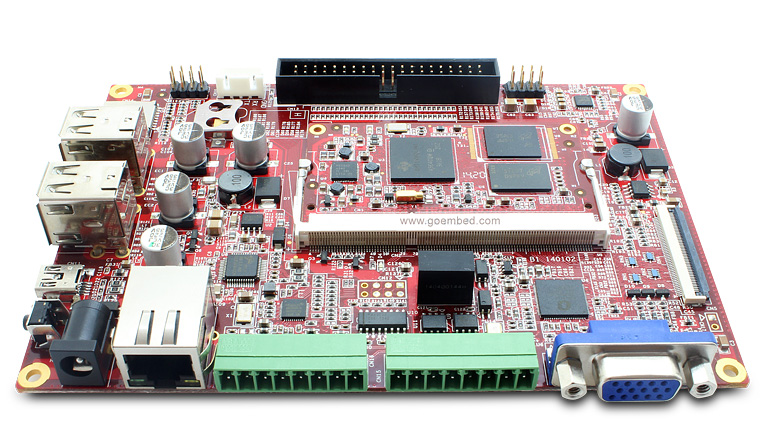
The B2A features 4 USB Host and 1 USB OTG, 1 Ethernet ports, LCD, Touch screen,RTC, and more other peripherals.

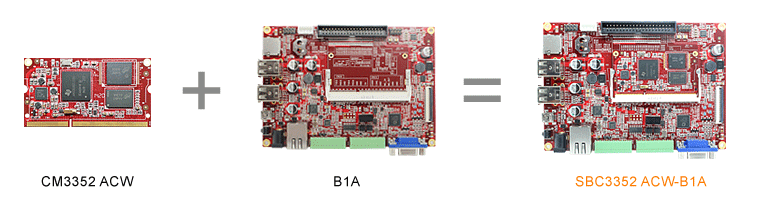
The SBC board targets a wide range of applications, including: HMIs, Digital Signage, POS, Data Terminal, Medical Devices, Navigation, Industrial Automation, Entertainment system, Thin Clients, Robotics, Game Console and much more.

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**

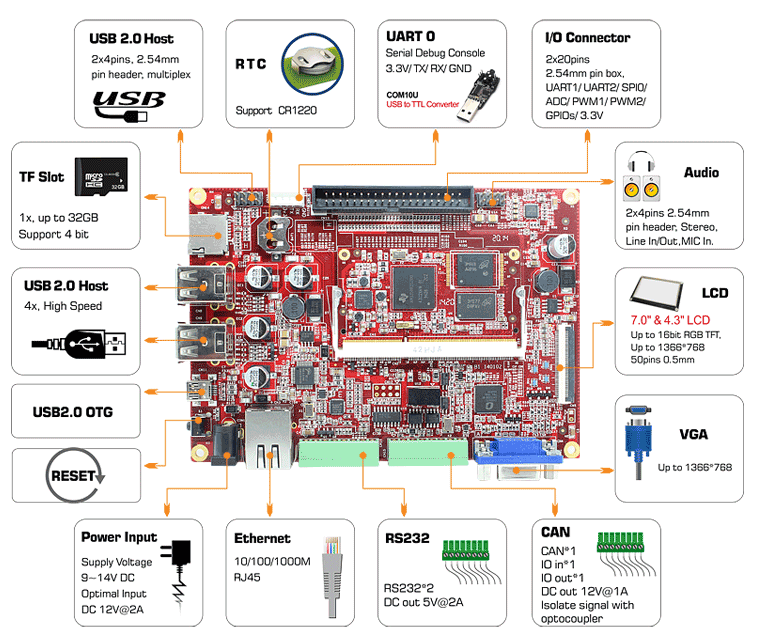
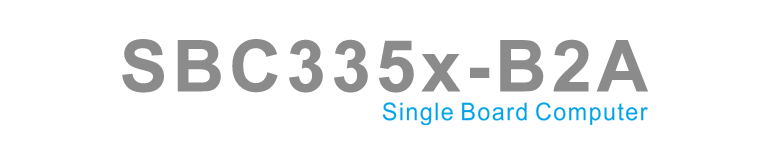
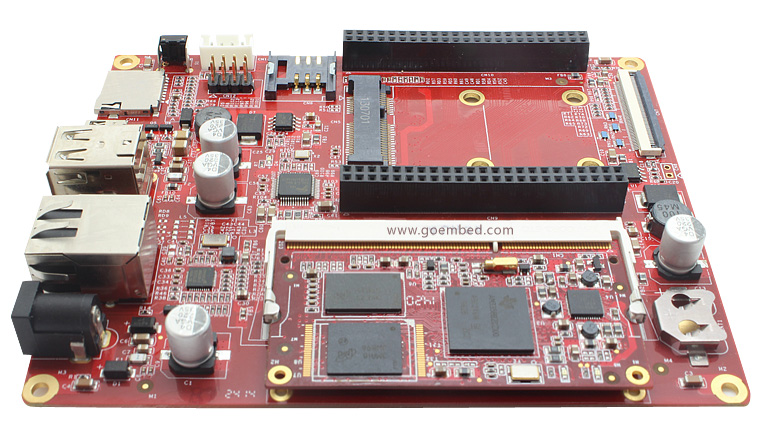
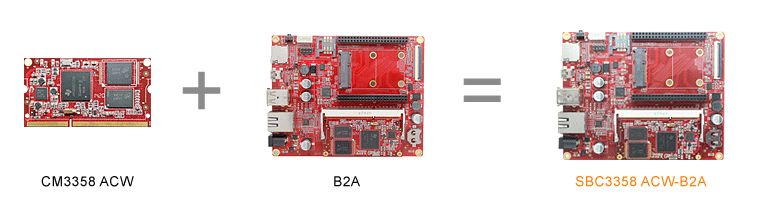


Figure 1 B1 Block Diagram







**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**

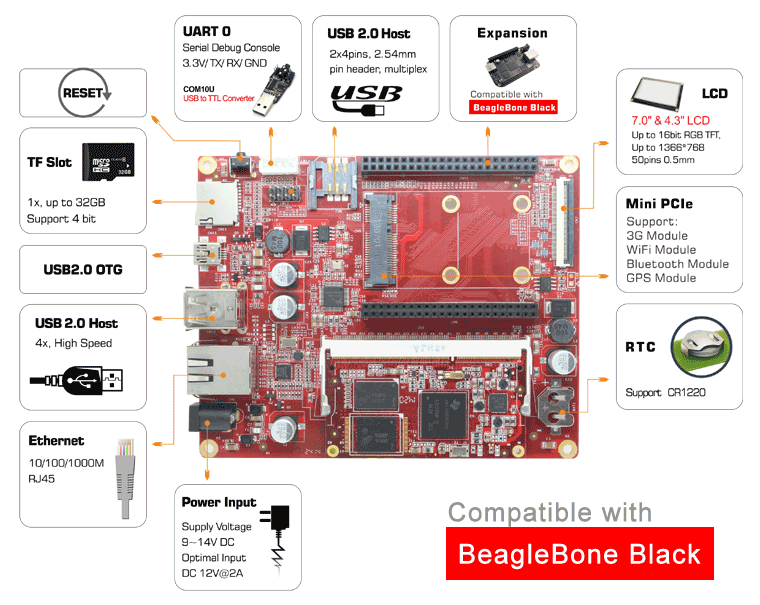


Figure 1 B2A Block Diagram

**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.

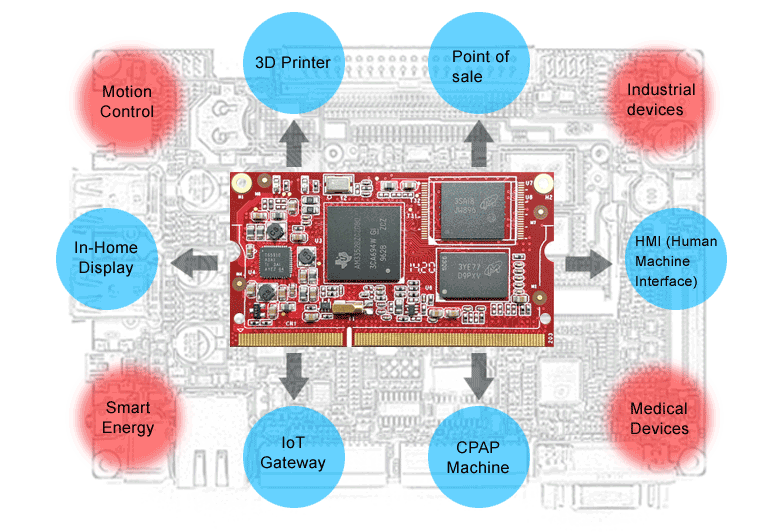
We are your trust worthy partner on ARM embedded design services and solutions.

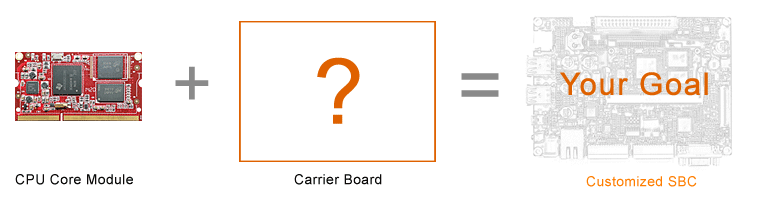
**More Carrier Boards**

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>

