

|  |
| --- |
| Linux下脚本编写教程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，自我操作总结。由于作者水平有限，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)交流和建议，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

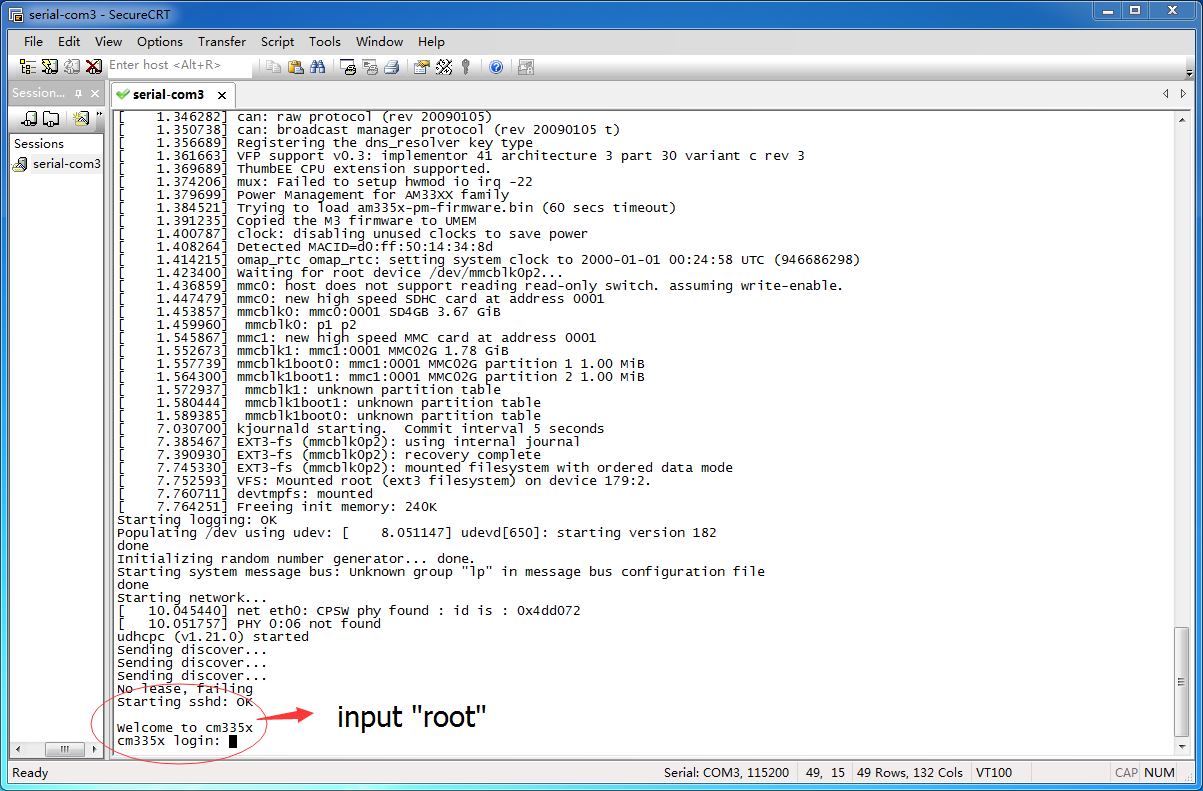
|  |  |  |
| --- | --- | --- |
| Partition | Size | Description |
| BootLoader | **200MB** | **FAT32格式分区** |
| rootfs | **约1500MB** | **EXT3格式分区** |

### 准备工作

1、准备好已经烧好Linux系统的TF卡，且TF卡FAT分区中必须有：MLO、u-boot.img、uEnv.txt、uImage和rootfs.tar.bz2这几个文件，再把卡插到开发板中。

2、连接好USB转TTL串口模块，打开串口调试软件SecureCRT.exe。

3、开发板接上12V电源适配器，开机后串口调试软件打印如下信息：



4、输入root登录Linux系统。

5、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。

### 脚本简介

脚本文件类似于[DOS操作系统](http://baike.baidu.com/view/341022.htm)中的批处理文件，它可以将不同的命令组合起来，并按确定的顺序自动连续地执行。脚本文件是文本文件，用户可使用任意文本编辑器来创建脚本文件。

脚本是[批处理文件](http://baike.baidu.com/view/295769.htm)的延伸，是一种纯文本保存的程序，一般来说的计算机[脚本程序](http://baike.baidu.com/view/69991.htm)是确定的一系列控制计算机进行运算操作动作的组合，在其中可以实现一定的逻辑分支等。

脚本程序相对一般程序开发来说比较接近[自然语言](http://baike.baidu.com/view/229451.htm)，可以不经编译而是[解释执行](http://baike.baidu.com/view/1182922.htm)，利于快速开发或一些轻量的控制。

参考：<http://baike.baidu.com/link?url=LVyAZxXcvaDLALY-HK5ZzfEQPul-aotTo_qQXcEHuYymos6_2FbNfwVFYgFGsR1w9LWa5Tj5E4SZ1xsydc-tgq>

### 脚本的主要特点

1、脚本语法比较简单，比较容易掌握。

2、脚本与应用程序密切相关，比较容易掌握。

3、脚本一般不具备通用性。所能处理的问题范围有限。

### 编写思路

在这个教程中，我们会教大家如何编写一个脚本来自动烧写Linux系统到eMMC中。思路是这样的：第一，要将系统烧写进eMMC，我们需要先从TF卡启动，也就是说我们首先要能够启动系统，并且登陆进去；第二，成功进入系统之后，我们要找到我们编写的脚本，然后执行它；第三，执行完脚本之后，我们除了看输出信息，还要拔掉TF卡，尝试从eMMC启动系统。如果能正常启动系统，我们就认为该脚本编写成功。那么，我们编写时该参考什么呢？正如我上面所说，脚本的执行可以简单地认为是在执行一条条命令，我们可以根据《TI AM335x 烧写Linux系统到eMMCv 1.0》这篇文章来编写我们的脚本。

### 脚本代码

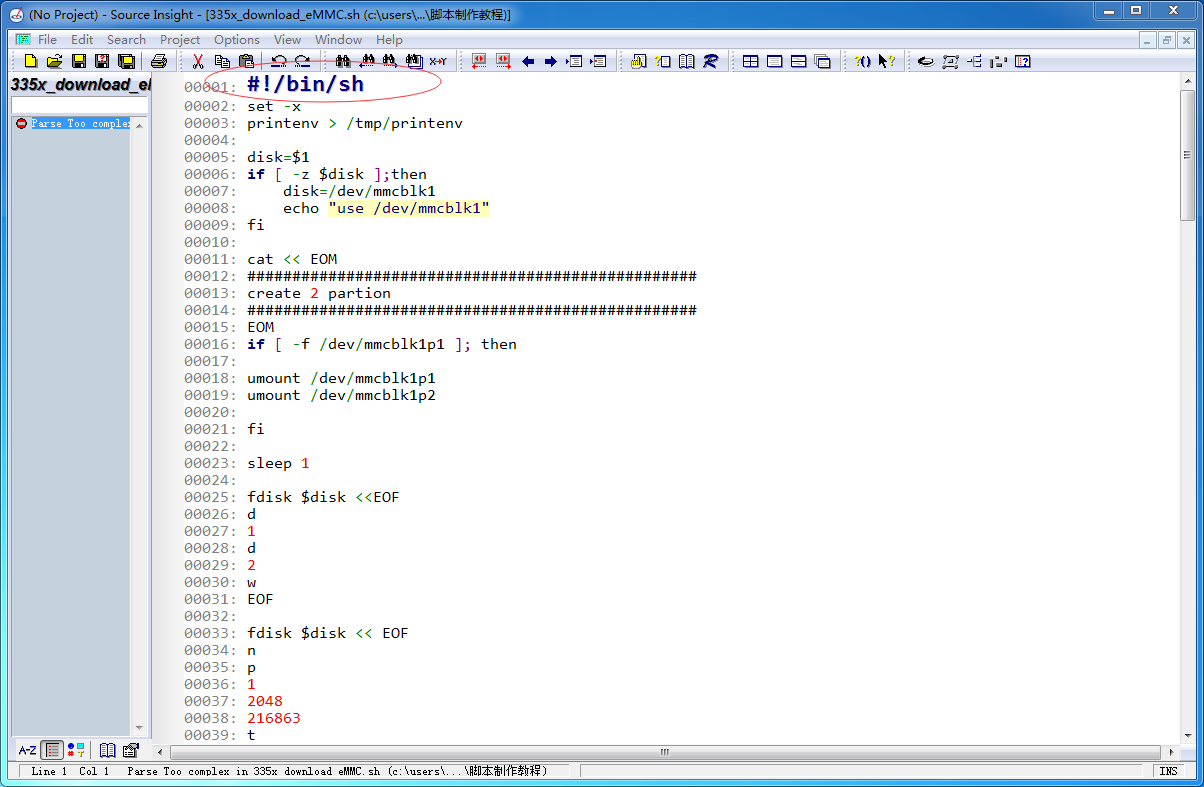
#### 1、完整脚本代码

|  |  |
| --- | --- |
| 前半部分 | 后半部分 |
| #!/bin/sh  set -x  printenv > /tmp/printenv  disk=$1  if [ -z $disk ];then  disk=/dev/mmcblk1  echo "use /dev/mmcblk1"  fi  cat << EOM  ####################################  create 2 partion  ####################################  EOM  if [ -f /dev/mmcblk1p1 ]; then  umount /dev/mmcblk1p1  umount /dev/mmcblk1p2  fi  sleep 1  fdisk $disk <<EOF  d  1  d  2  w  EOF  fdisk $disk << EOF  n  p  1  2048  216863  t  b  a  1  w  EOF  fdisk $disk << EOF  n  p  2  216864  3751935  t  2  83  w  EOF  sleep 1  umount /dev/mmcblk1p1  sleep 1  umount /dev/mmcblk1p2  sleep 1  cat << EOM  ####################################  ge shi hua boot partion  ####################################  EOM  mkdosfs -F 32 /dev/mmcblk1p1 | cat << EOM  ####################################  ge shi hua rootfs partion  ####################################  EOM  mkfs.ext3 /dev/mmcblk1p2  cat << EOM  #############################################  mount /dev/mmcblk1p1 to /media/mmcblk1p1  #############################################  EOM  mkdir -p /media/mmcblk1p1  mount /dev/mmcblk1p1 /media/mmcblk1p1  cat << EOM  #############################################  mount /dev/mmcblk1p2 to /media/mmcblk1p2  #############################################  EOM  mkdir -p /media/mmcblk1p2  mount /dev/mmcblk1p2 /media/mmcblk1p2  cat << EOM  #############################################  cd /media/mmcb0k0p1 CARD 1 partion  #############################################  EOM  cd /media/mmcblk0p1  cat << EOM  #############################################  cp /media/mmcblk0p1/MLO /media/mmcblk1p1  #############################################  EOM  cp /media/mmcblk0p1/MLO /media/mmcblk1p1  cat << EOM  #############################################  cp /media/mmcblk0p1/u-boot.img /media/mmcblk1p1  #############################################  EOM  cp /media/mmcblk0p1/u-boot.img /media/mmcblk1p1  cat << EOM  #############################################  cp /media/mmcblk0p1/uImage /media/mmcblk1p1  #############################################  EOM  cp /media/mmcblk0p1/uImage /media/mmcblk1p1  cat << EOM  #############################################  cp /media/mmcblk0p1/uEnv.txt /media/mmcblk1p1  #############################################  EOM  cp /media/mmcblk0p1/uEnv.txt /media/mmcblk1p1  sync  tar -mjxvf /media/mmcblk0p1/rootfs.tar.bz2 -C /media/mmcblk1p2  sync  umount /dev/mmcblk1p1  sleep 1  umount /dev/mmcblk1p2  sleep 1 |

#### 2、代码分析

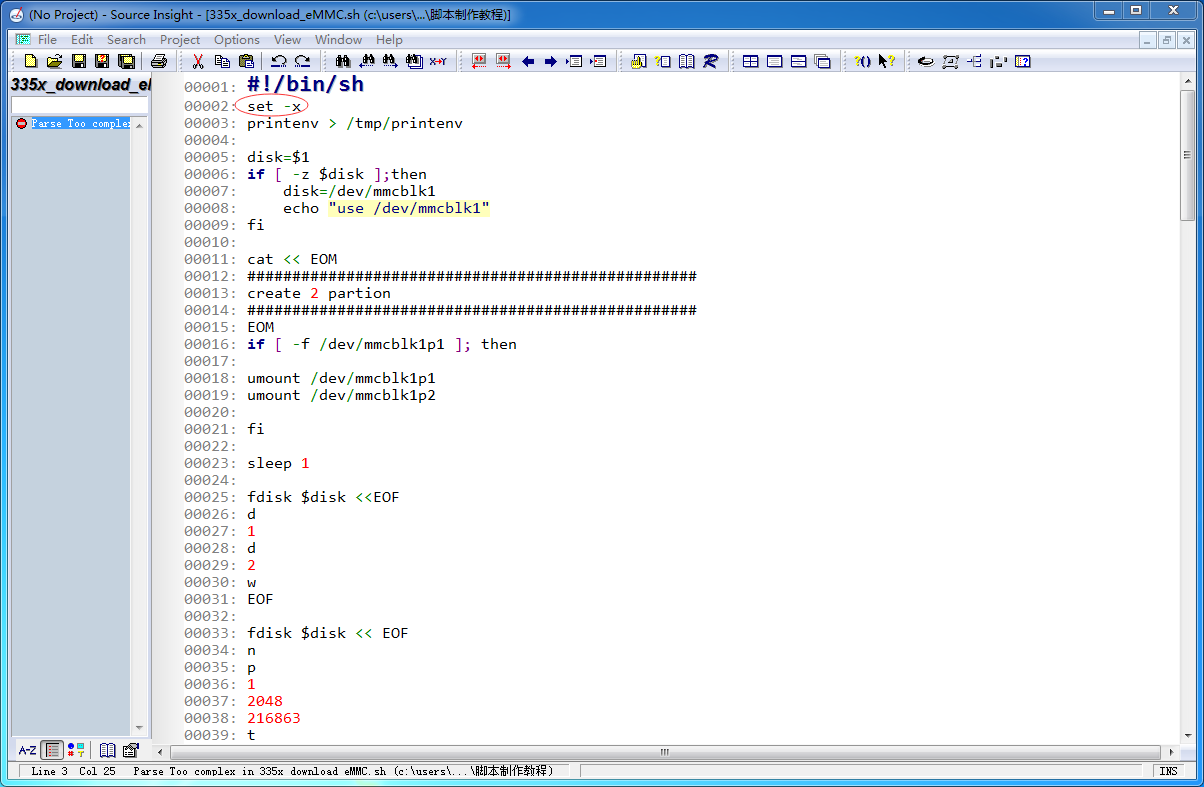
接下来我们参考《TI AM335x 烧写Linux系统到eMMCv 1.0》来分析代码。

（1）、尽量添加声明：“#!/bin/sh”



它不是注释，而是对shell的声明，说明了我们所用的是哪一种类型的shell和它存放的路径。如果没有声明，则该脚本将在默认的shell中执行，默认shell是由用户所在的系统定义为执行shell脚本的shell。

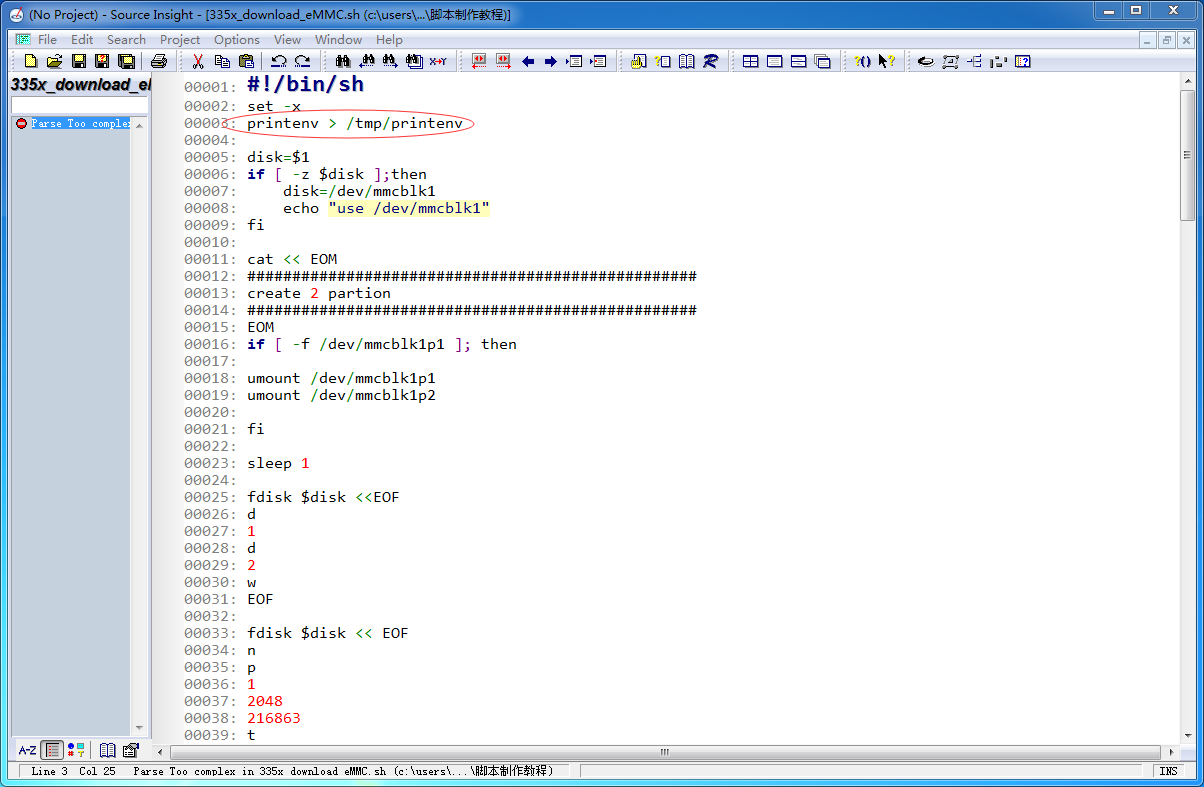
（2）、set –x



“set –x”和“set +x”指令用于脚本调试。set是指把它下面的命令打印到屏幕。

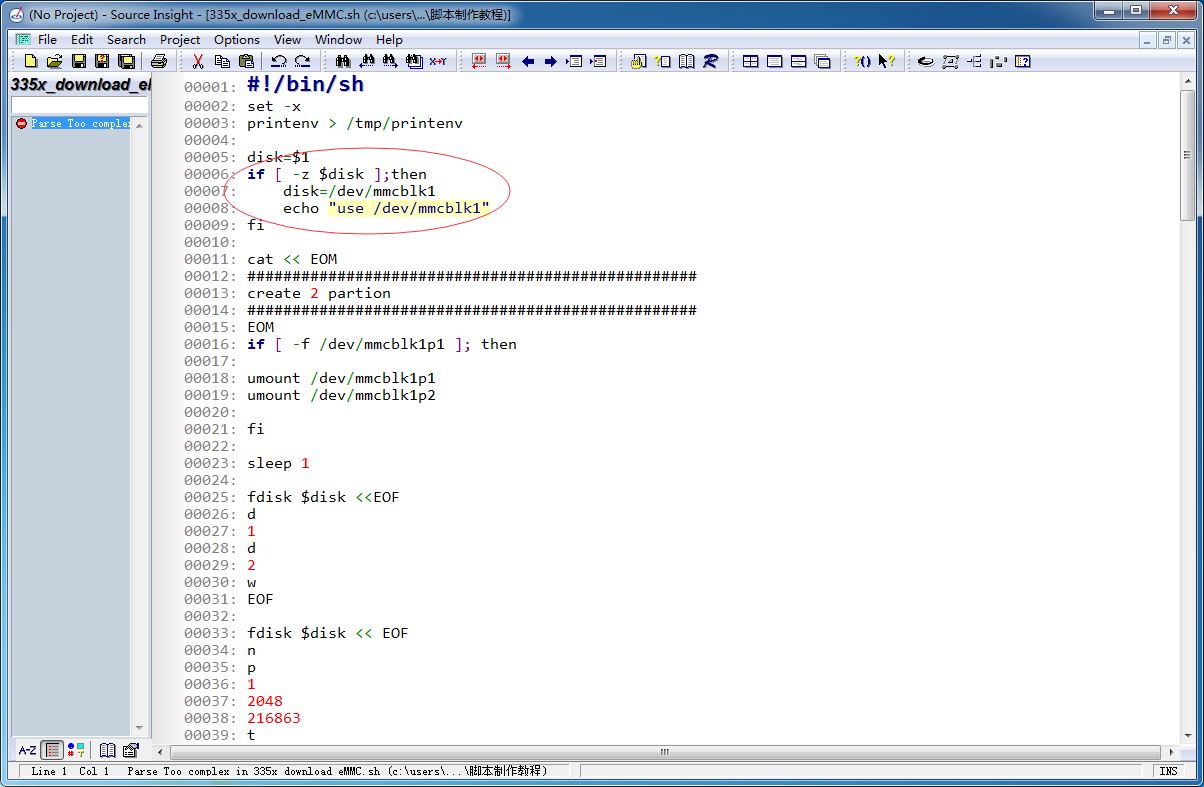
“set -x”是开启，“set +x”是关闭。“set -o”是查看。执行set –x之后对整个脚本都有效。

（3）、printenv > /tmp/printenv



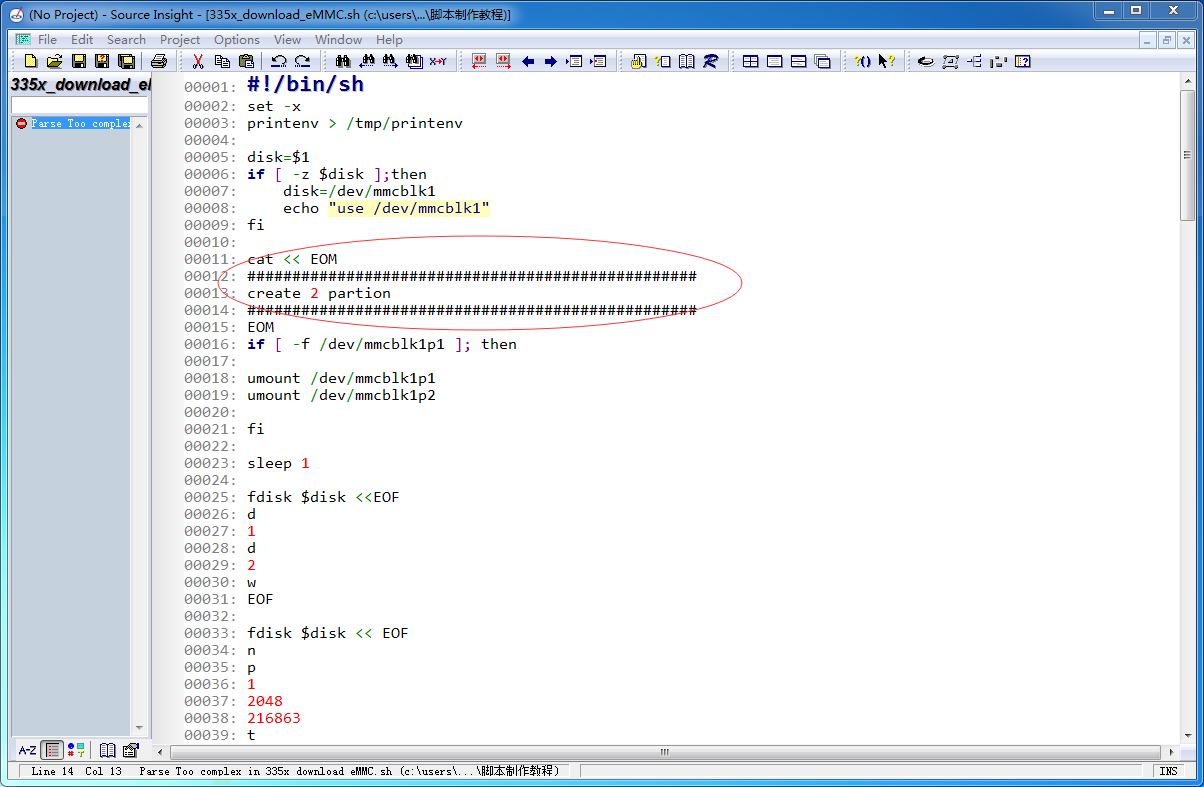
“printenv > /tmp/printenv”这句话的意思是执行“printenv”这个命令，并且把输出的结果重定向到/tmp/printenv文件中，可以简单理解为把输出的内容保存到该文件中。

（4）、



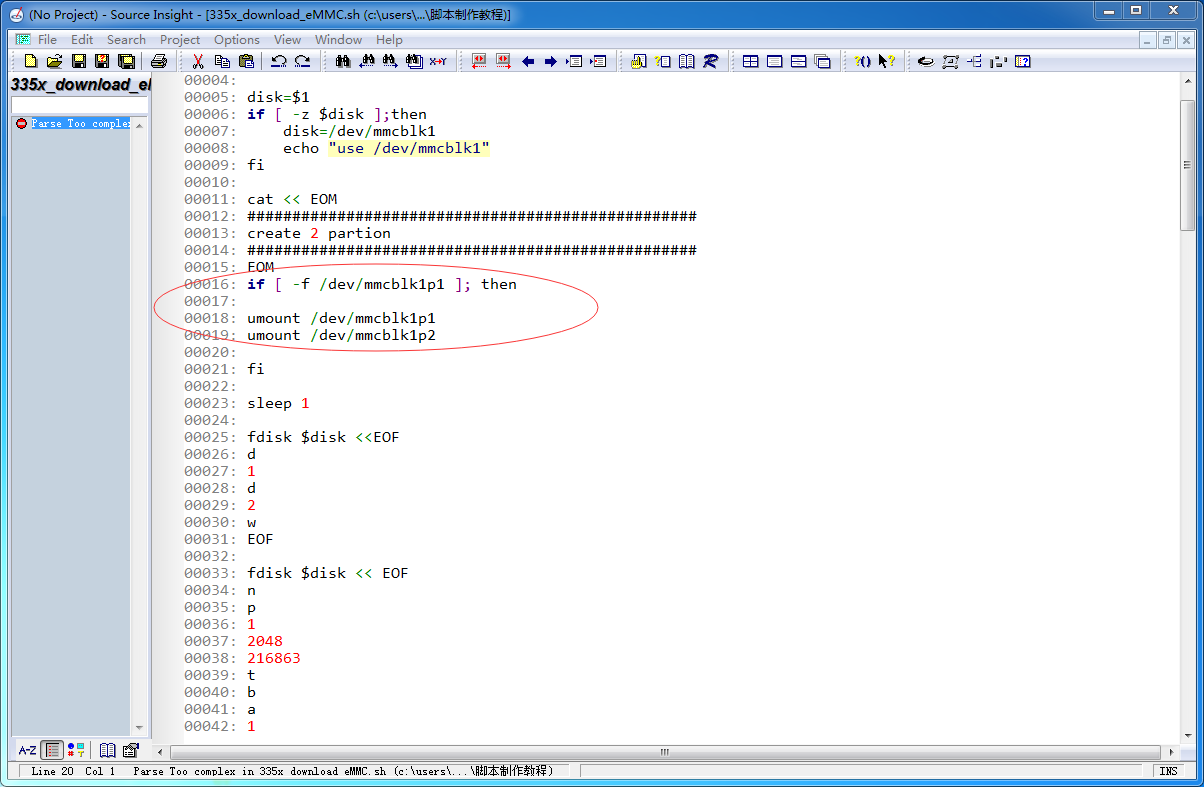
首先我们得先明白以下几点：第一，我们执行脚本时是可以可以传递参数的，比如执行我们正编写的这个脚本，可以是“./ 335x\_download\_eMMC.sh”,也可以是“./335x\_download\_eMMC.sh /dev/sdb”。如果不带参数，则脚本中的$1为空，则执行then到fi之间的语句。也就是默认把disk赋值为“/dev/mmcblk1”。如果带参数，则直接把参数赋值给disk。在这个脚本中disk 代表了将烧写的分区。第二，我们从TF卡启动系统后，TF卡在系统中则对应文件“/dev/mmcblk0”，eMMC则对应文件“/dev/mmcblk1”。现在我们的目的是烧写系统进eMMC，则应为“/dev/mmcblk1”。echo用于将“use /dev/mmcblk1”这个字符串送往标准输出。“if...fi”是控制符号。

（5）、



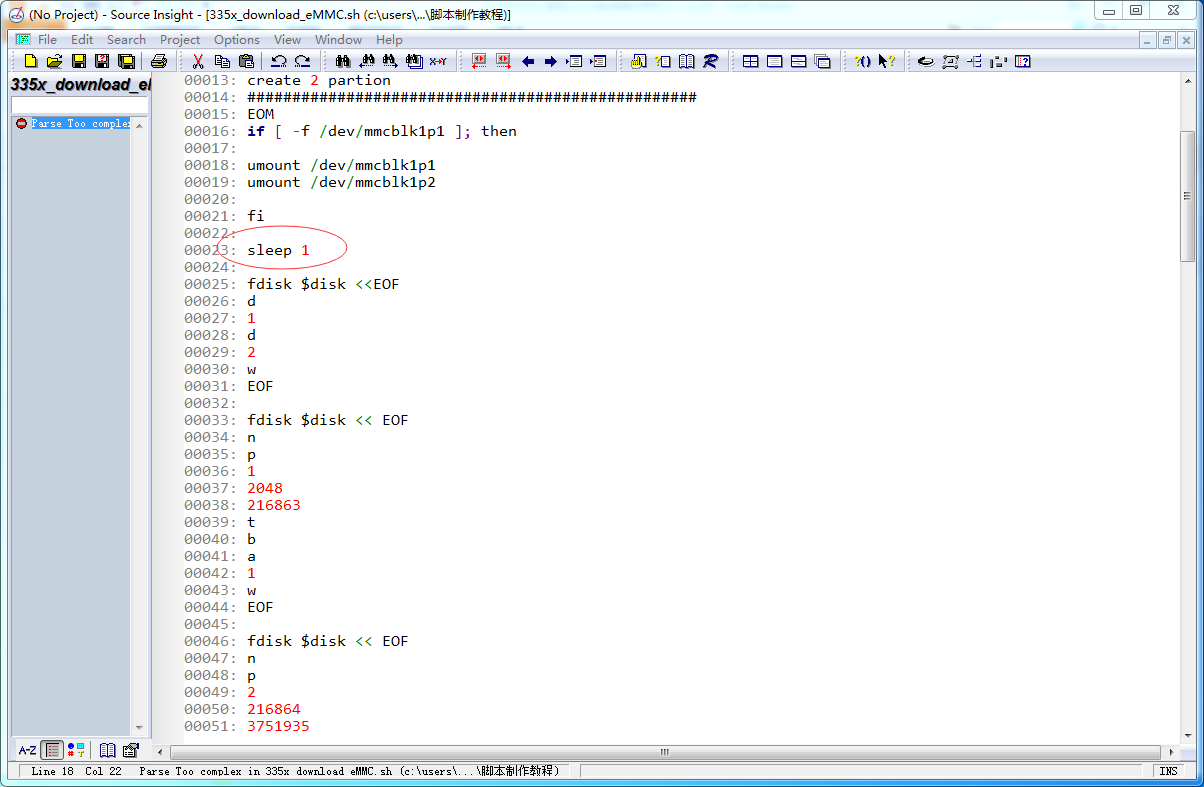
cat 用于打印一段文字，cat << EOM...EOM用于打印多行信息，也就是说EOM包含的字符都会打印出来。这个指令的用于打印提示信息。

（6）、



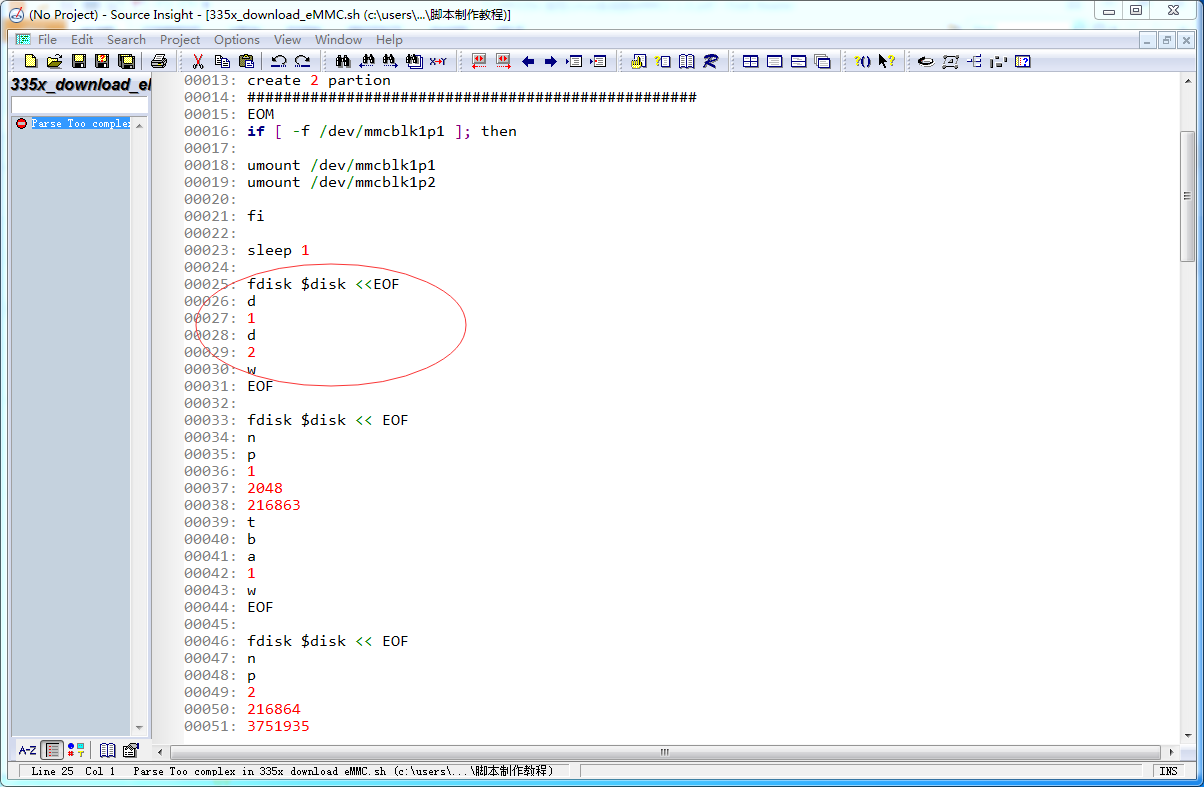
这句话的作用是判断有无/dev/mmcblk1p1这个文件，如果有，则表示mmcblk1（也就是eMMC）已经有一个分区，为了避免这个分区已经被自动挂载影响后面的操作，我们将两个分区都卸载。也就是执行接下来的两条umount指令。

（7）、



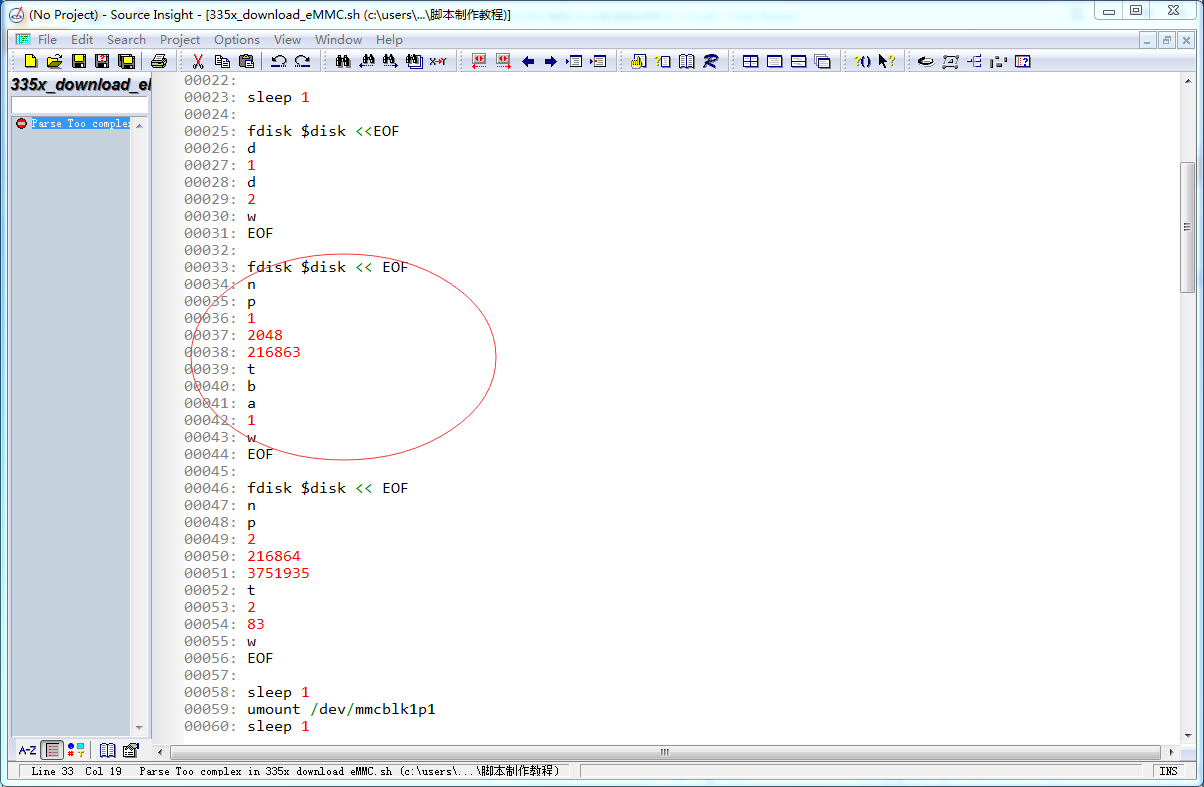
sleep表示休眠1s。

（8）、

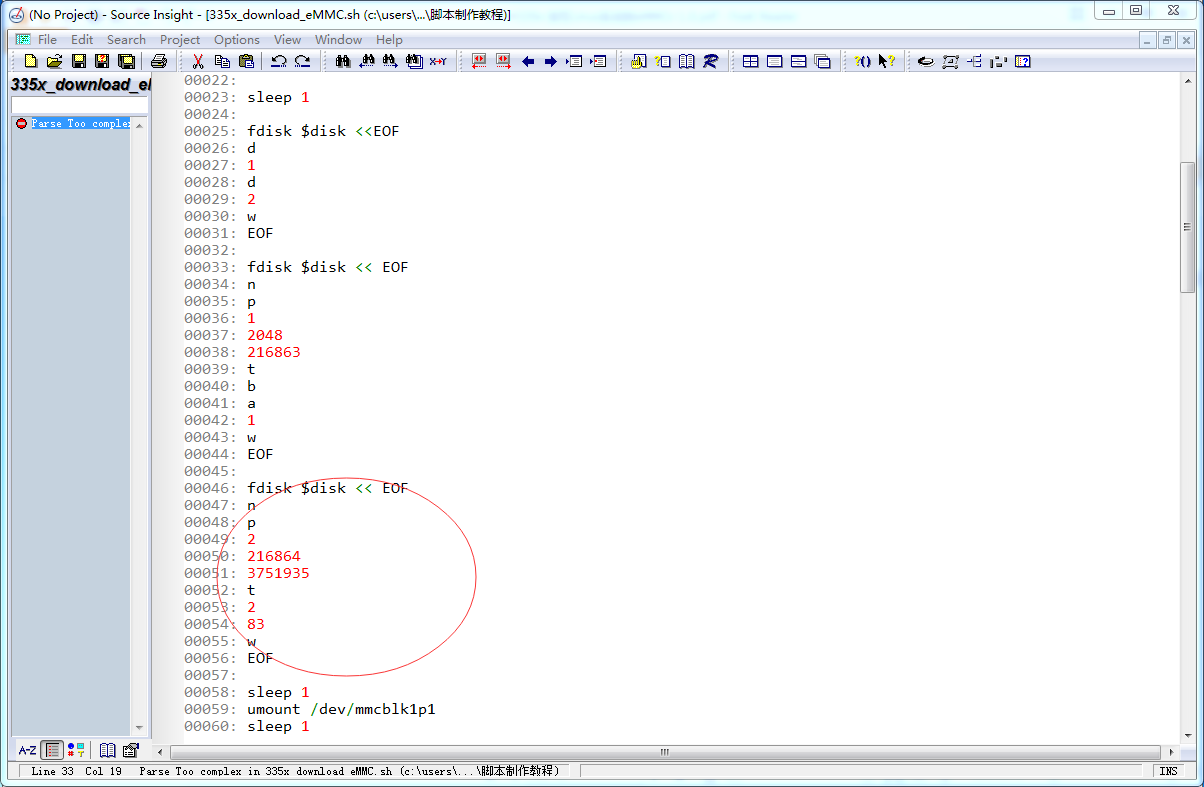


这个是删除第1、2个分区。有的板子之前可能存在好几个分区，则需要客户对应增加。比如还有第3个分区，则在“2”后面增加“d 3”。以此类推。补充一句，上面卸载分区也是同样的道理。这时我们发现已经开始和《TI AM335x 烧写Linux系统到eMMCv 1.0》对应上了。其中，“fdisk $disk <<EOF...EOF”中的“...”表示用户需要输入的内容，根据《TI AM335x 烧写Linux系统到eMMCv 1.0》可知，我们执行完“fdisk”命令之后需要输入多个参数，制作脚本的目的就是想尽量减少人工操作，通过EOF终止输入，也就是结束输入。

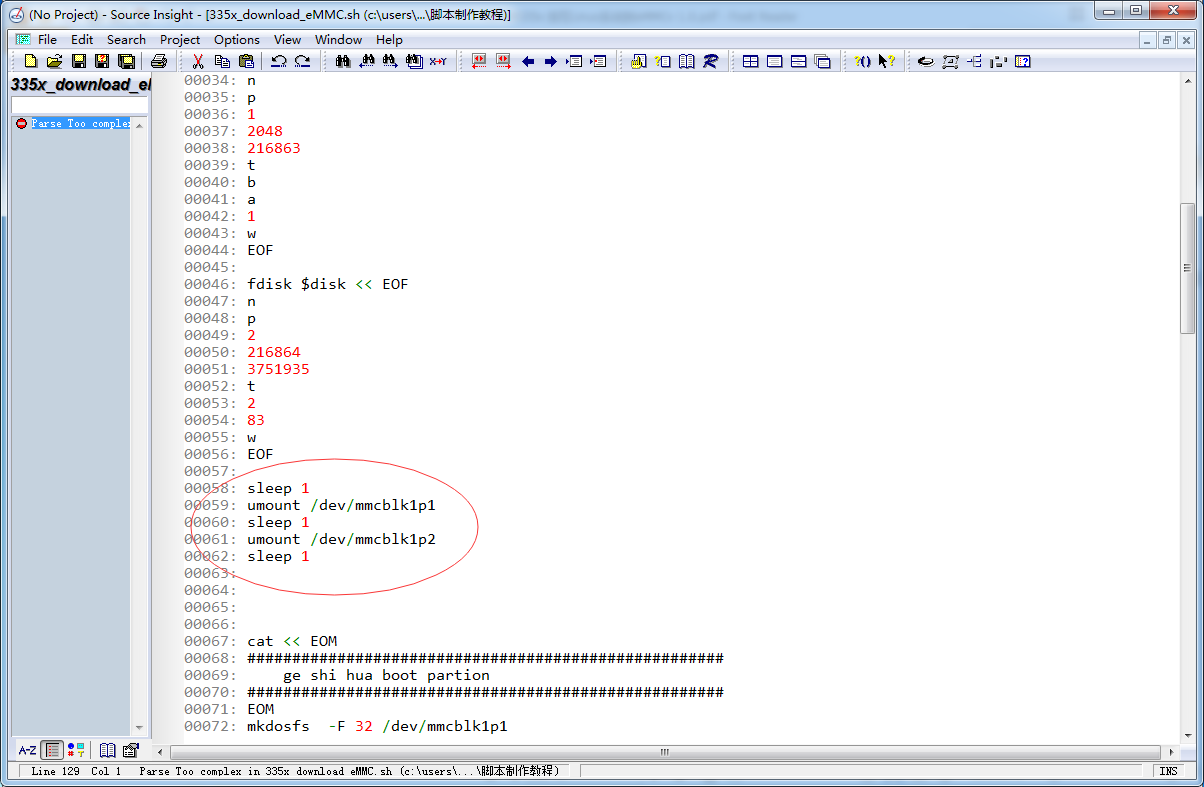
以下截图对应分区1：



以下截图对应分区2：



分区完之后卸载（跟《TI AM335x 烧写Linux系统到eMMCv 1.0》中的一模一样）：

。

剩余的代码只要照着《TI AM335x 烧写Linux系统到eMMCv 1.0》来理解就可以了，完全是一模一样的，我在这里不再赘述。

到这里脚本代码分析完毕。然后把脚本文件复制到TF卡中进行测试。

### 脚本测试

注：SBC3358-B1A默认先从eMMC启动，如果eMMC中没有镜像，则会从TF卡启动，如果TF卡中也没有镜像，则串口终端会一直打印“CCCC”。当eMMC中有镜像且TF卡也有镜像时，如果这时想从TF卡启动，只要先将板子上的CN17引脚短接再上电即可从TF卡启动。

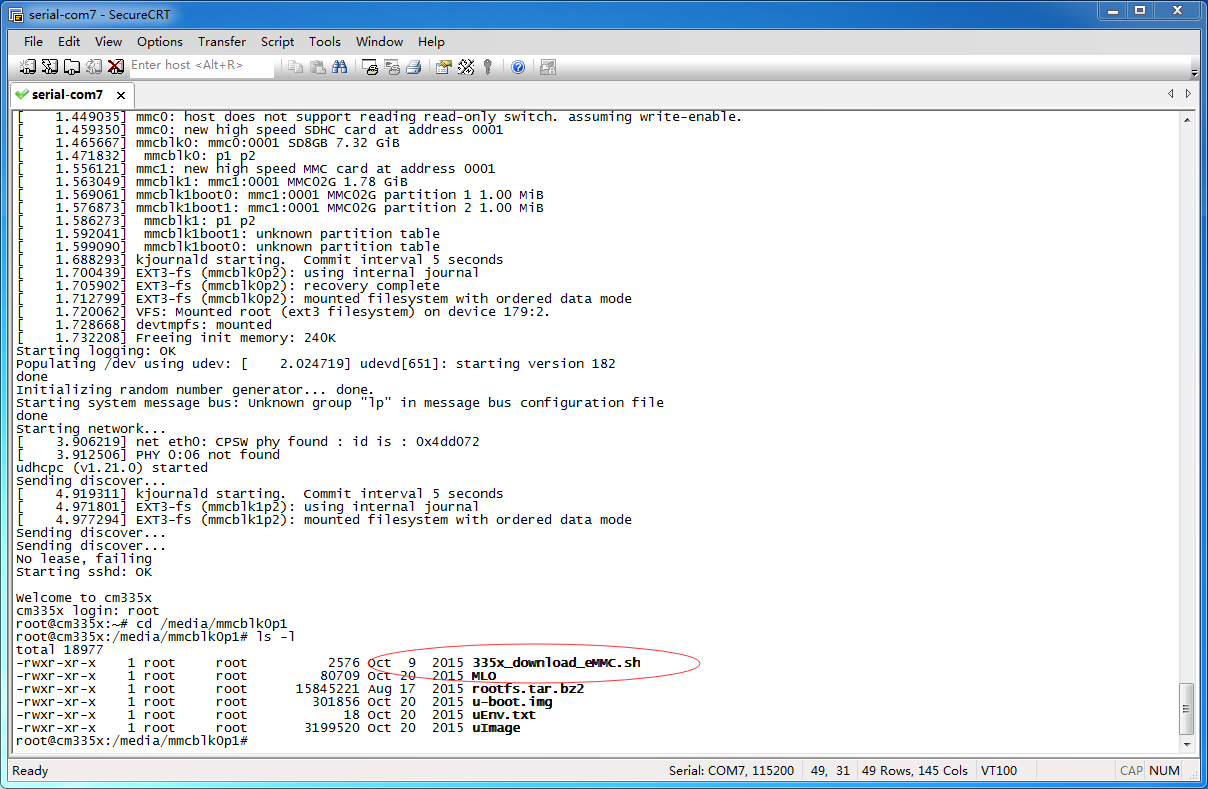
（1）、插上卡，接好串口线，打开串口调试软件。

（2）、软件设置为：波特率115200,8 bit数据位，无校验位，1bit停止位，无流控。

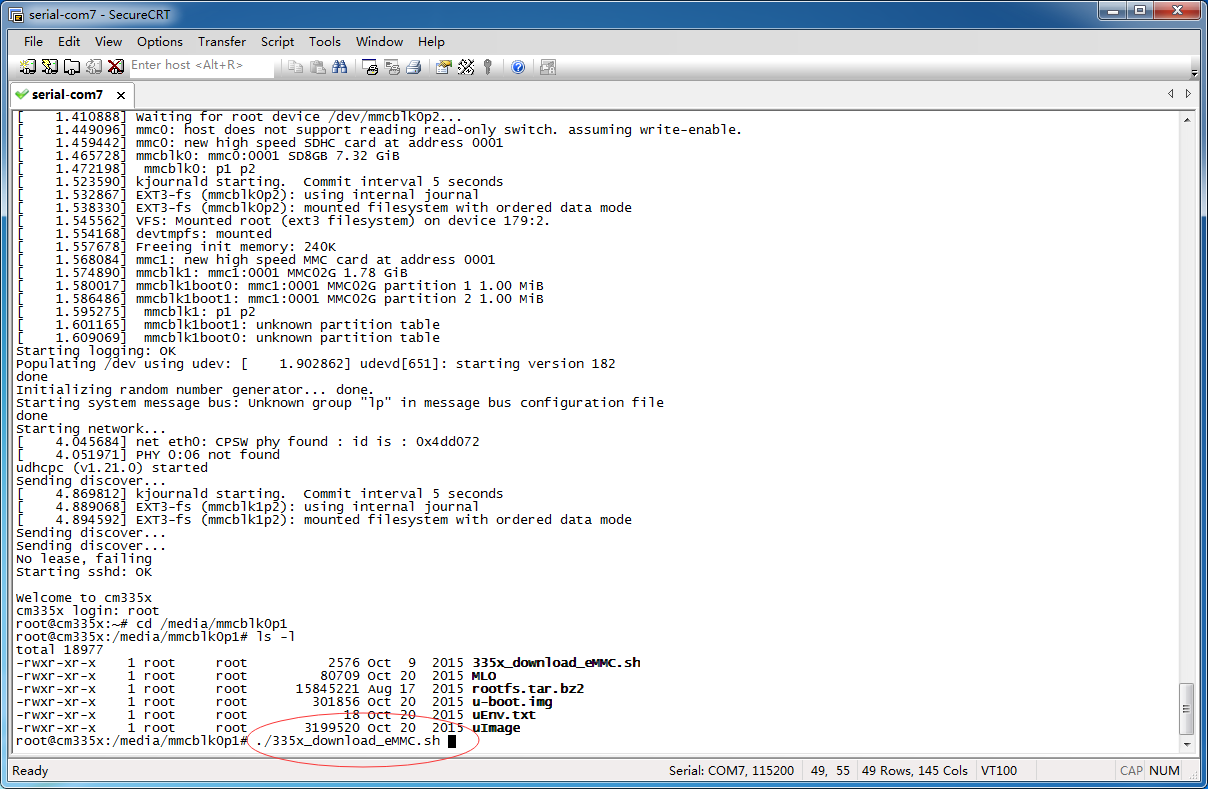
（3）、准备335x\_download\_eMMC.sh和rootfs.tar.bz2这两个文件，并确保这两文件在同一个目录下。

（4）、打开电源后输入“root”登录系统。

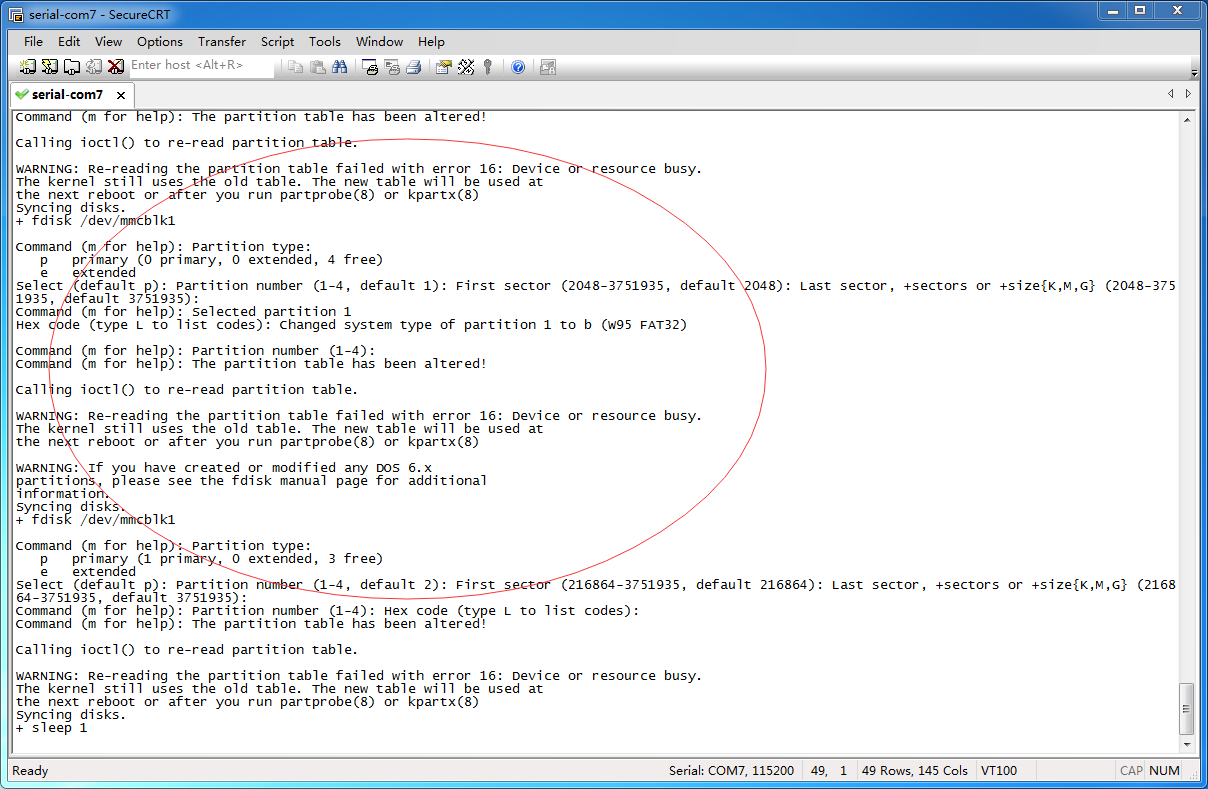
（5）、找到“335x\_download\_eMMC.sh”这个文件：（从TF卡启动时系统默认把TF卡的第一分区挂载到/media/mmcblk0p1这个目录下）



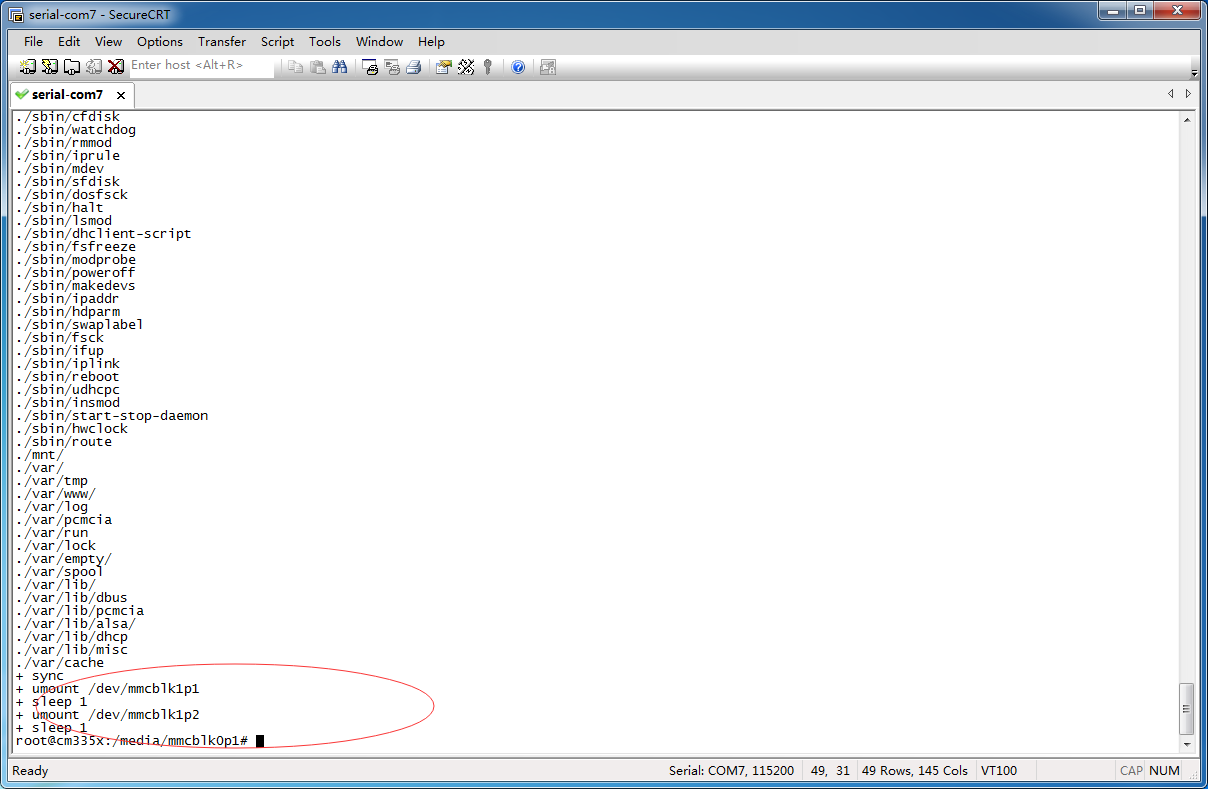
（6）、运行脚本文件：



（7）运行过程如下：



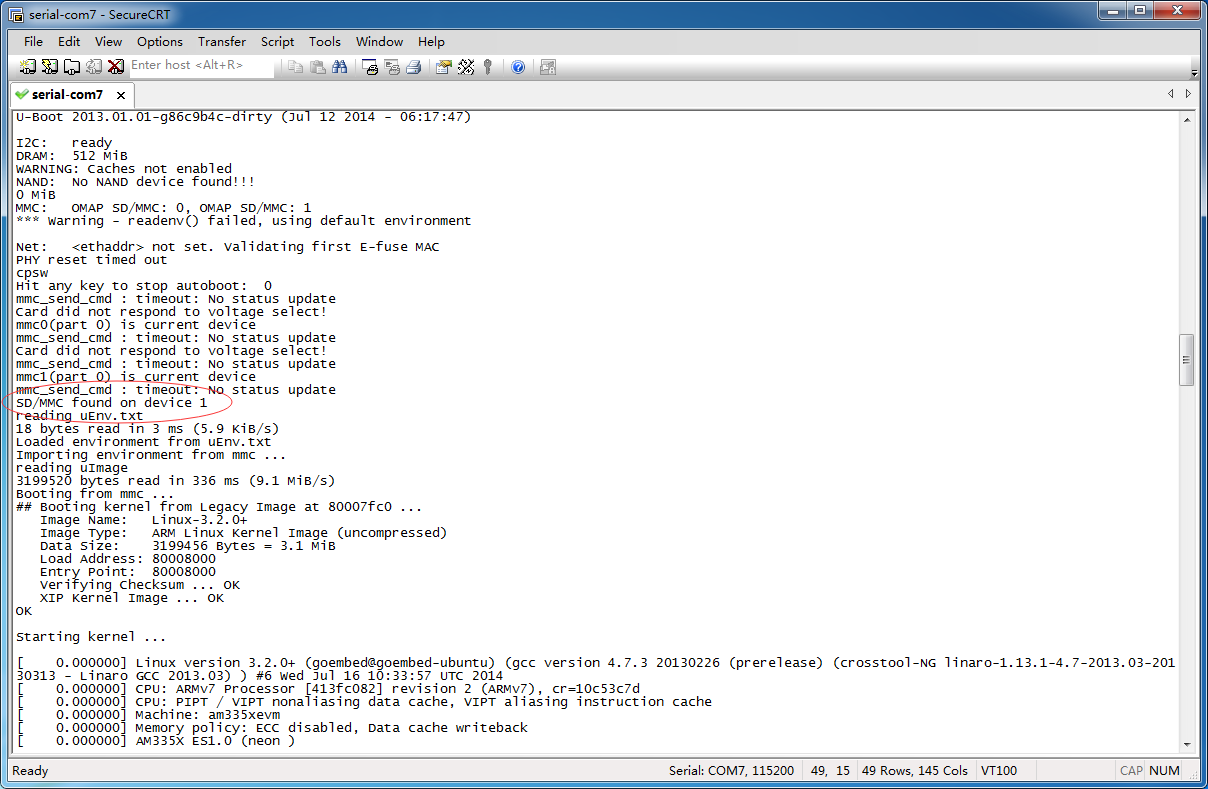
（8）、脚本自动执行结束后：



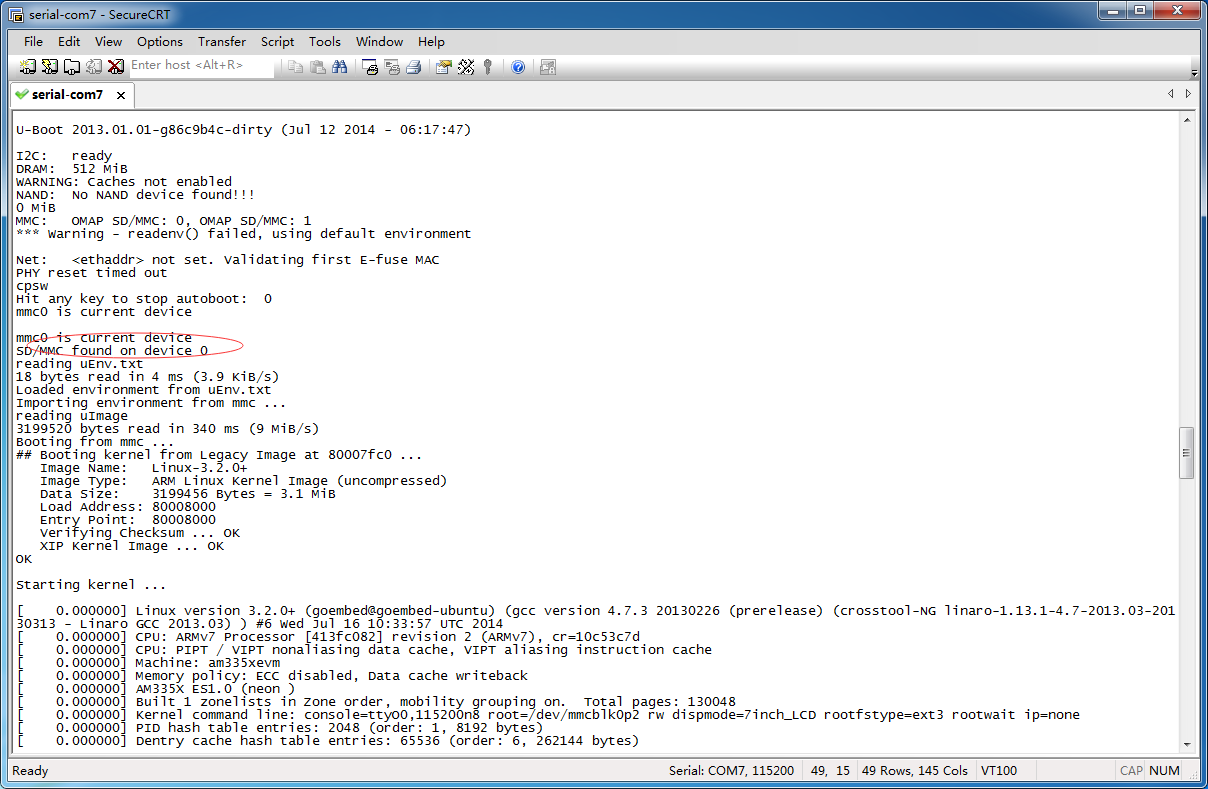
到这里我们已经通过脚本，把Linux系统自动烧写到eMMC了。

为了检查结果是否是对的。我们把TF卡拔掉，直接上电启动。如果系统能正常启动，我们则认为操作成功。

拔掉TF卡后启动截图如下：

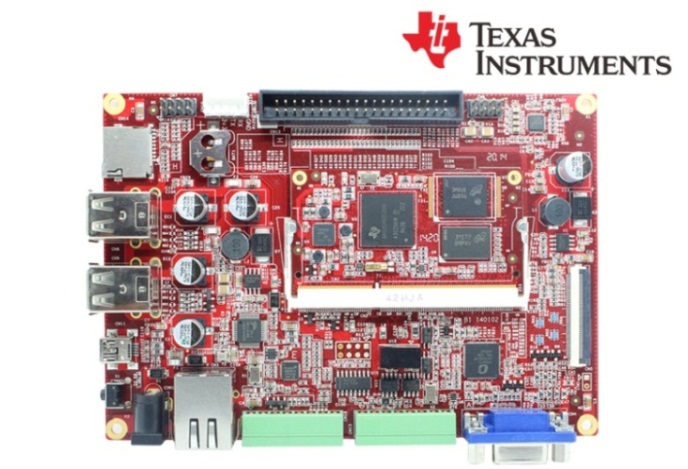


我们发现系统可以正常从eMMC中启动，操作成功。请注意图中圈出来的文字，此时发现的是device 1.也就是指eMMC。如果是从TF卡启动，则应该是打印device 0。我们再次插上TF卡启动看看（TF卡和eMMC中都有系统的话会优先从TF卡启动）：

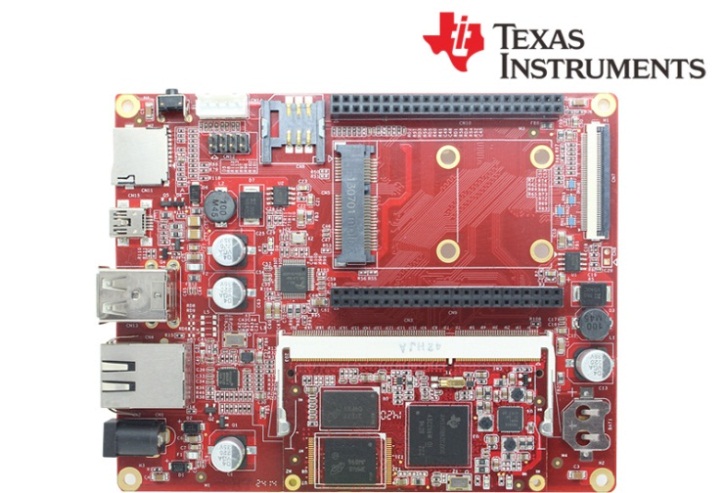


通过上面的截图我们发现确实打印device 0，这样一样我们就很容易辨别出系统启动的位置了。

到这里，Linux下简单的脚本编写教程完成。

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.

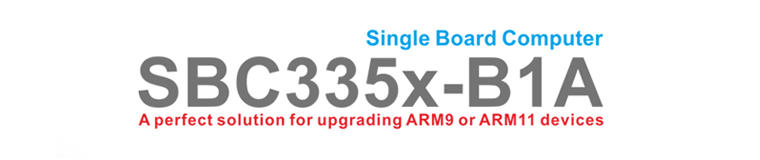
The B1A features 4 serial ports (including 2 RS232 and 2 TTL), 4 USB Host and 1 USB OTG, 1 Ethernet ports, CAN, RS485, Wiegand, VGA, LCD, Touch screen, Audio, ADC and more other peripherals.

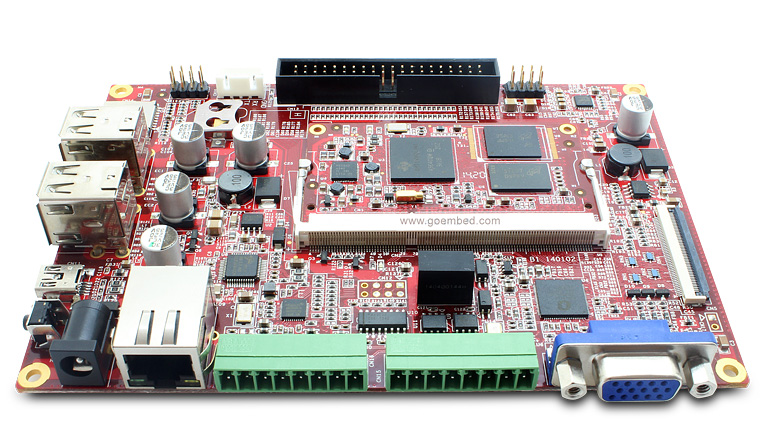
The B2A features 4 USB Host and 1 USB OTG, 1 Ethernet ports, LCD, Touch screen,RTC, and more other peripherals.

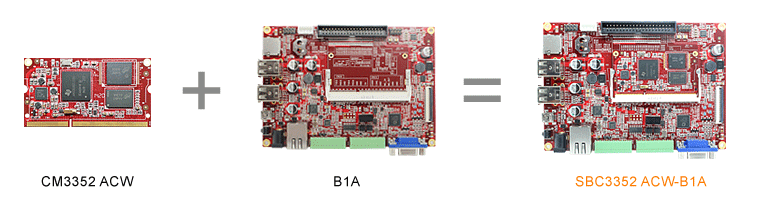
The SBC board targets a wide range of applications, including: HMIs, Digital Signage, POS, Data Terminal, Medical Devices, Navigation, Industrial Automation, Entertainment system, Thin Clients, Robotics, Game Console and much more.

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**

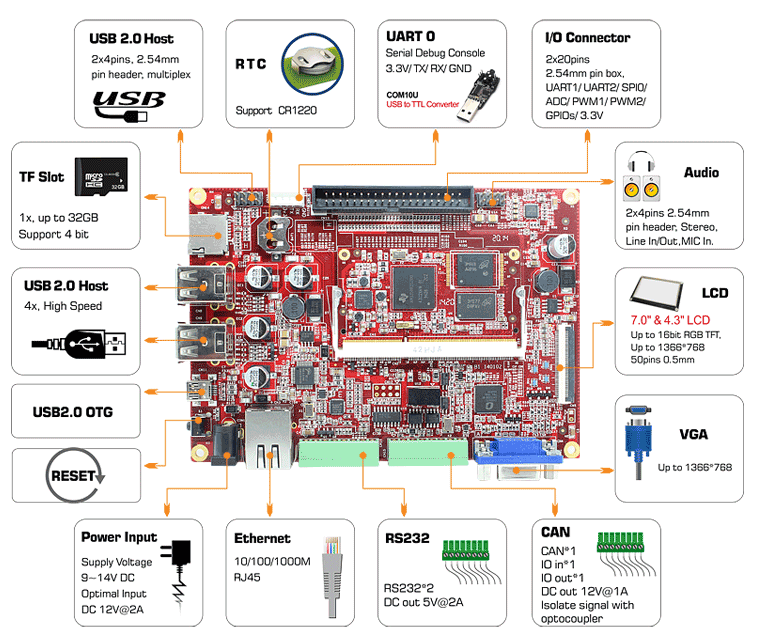
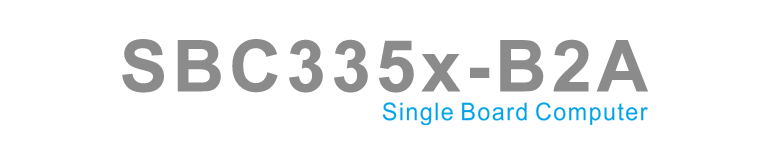
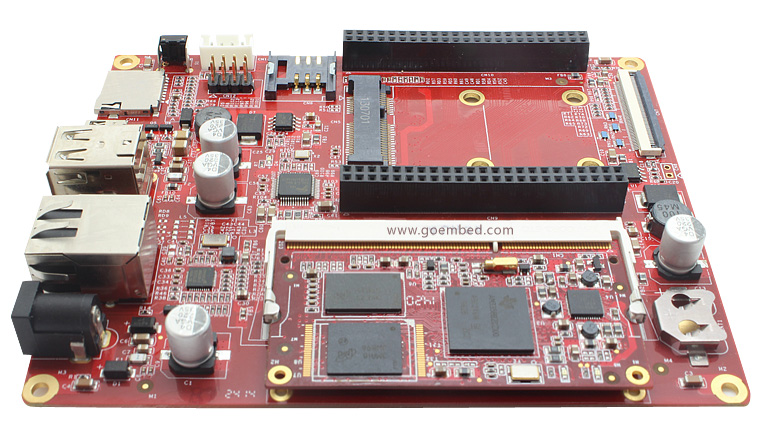
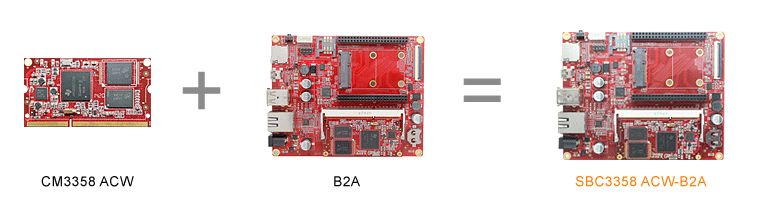


Figure 1 B1 Block Diagram







**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**

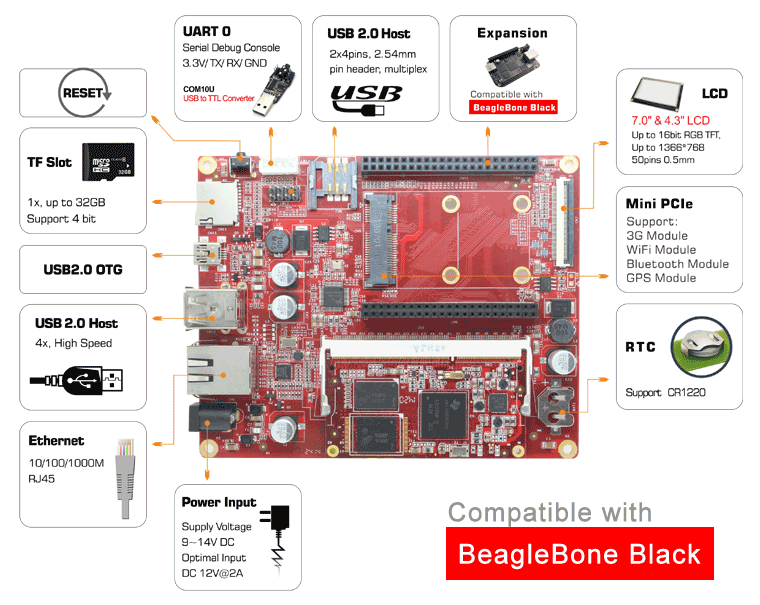


Figure 1 B2A Block Diagram

**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.

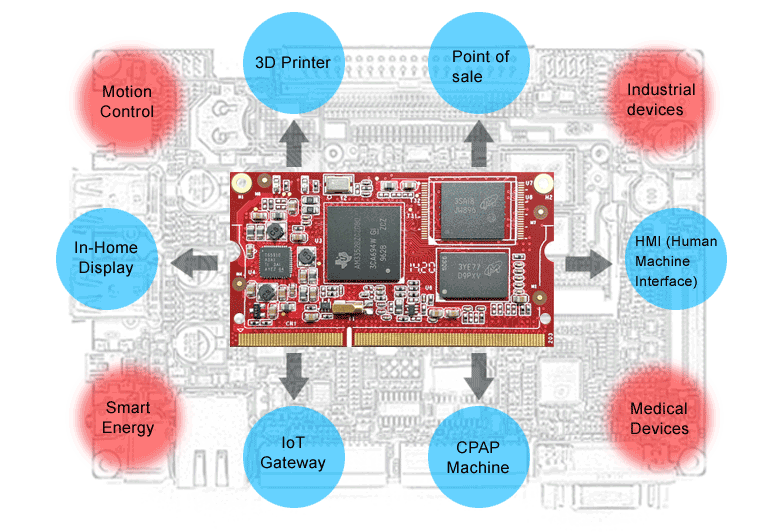
We are your trust worthy partner on ARM embedded design services and solutions.

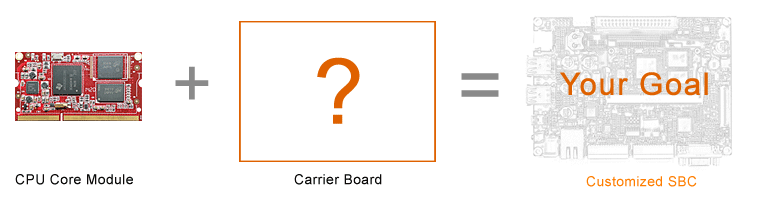
**More Carrier Boards**

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>

