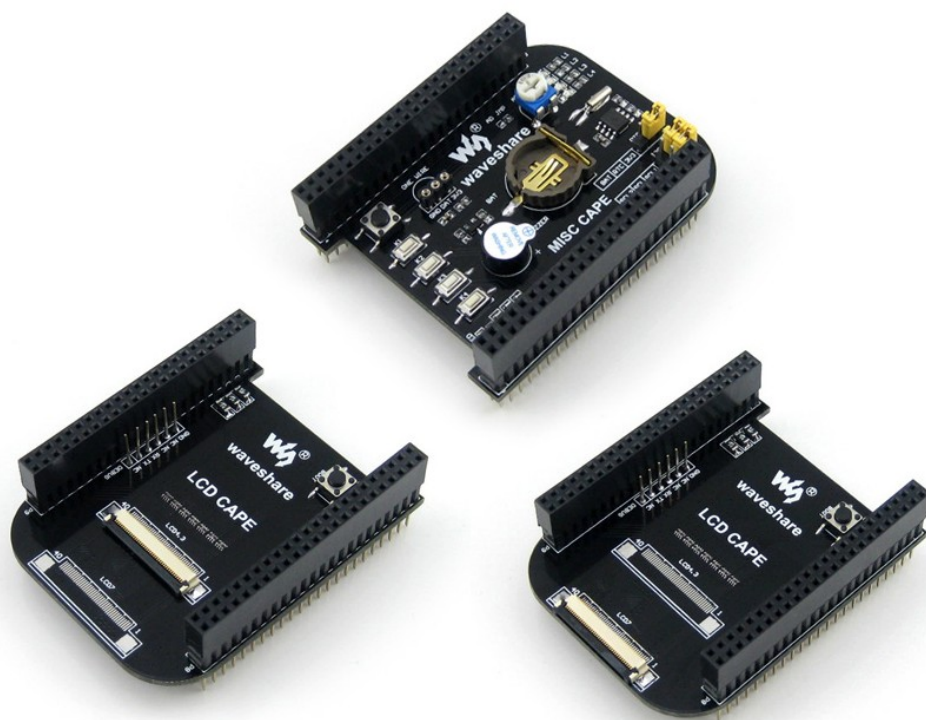


BB Black 扩展板

驱动移植手册

2014.02.07 V1.0



版权声明

本手册所有权由深圳市微雪电子有限公司独家持有。未经本公司的书面许可，不得以任何方式或形式进行修改、分发或复制本文档的任何部分，否则一切后果由违者自负。

版本更新记录

版本	日期	说明
V1.0	2014.02.07	初始发布

目录

版权声明.....	I
版本更新记录.....	I
特殊说明.....	1
第一章 Device Tree	1
1.1. Device Tree 简介.....	1
1.2. 编写 Device Tree.....	1
1.3. 编写 Device Tree Overlay.....	2
第二章 PWM 驱动移植.....	3
2.1. 编写.dts 文件.....	3
2.2. 编译.dts 文件.....	3
2.3. 加载.dtbo 文件.....	4
第三章 LED 驱动移植.....	4
3.1. 编写.dts 文件.....	4
3.2. 编译.dts 文件.....	5
第四章 按键驱动移植	6
4.1. 编写.dts 文件.....	6
4.2. 编译.dts 文件.....	6
4.3. 加载.dtbo 文件.....	6
第五章 Dallas 1-wires 驱动移植.....	6
5.1. 编写.dts 文件.....	7
5.2. 编译.dts 文件.....	7

5.3.	加载.dtbo 文件.....	7
第六章	I2C 驱动移植.....	7
6.1.	编写.dts 文件.....	7
6.2.	编译.dts 文件.....	8
6.3.	加载.dtbo 文件.....	8
第七章	RTC 驱动移植.....	8
7.1.	编写.dts 文件.....	8
7.2.	编译.dts 文件.....	9
7.3.	加载.dtbo 文件.....	9
第八章	SPI 驱动移植.....	9
8.1.	编写.dts 文件.....	9
8.2.	编译.dts 文件.....	10
8.3.	加载.dtbo 文件.....	11
第九章	CAN 驱动移植.....	11
9.1.	编写.dts 文件.....	11
9.2.	编译.dts 文件.....	11
9.3.	加载.dtbo 文件.....	11
第十章	RS485 驱动移植.....	11
10.1.	编写.dts 文件	12
10.2.	编译.dts 文件	12
10.3.	加载.dtbo 文件	12
第十一章	LCD 和 ADC 驱动移植	12

11.1.	编写.dts 文件	12
11.2.	编译.dts 文件	13
11.3.	加载.dtbo 文件	13
第十二章	USB Camera 驱动移植	13
第十三章	USB WIFI 驱动移植	14
第十四章	系统更新.....	14
第十五章	文件拷贝.....	16
第十六章	关于卸载.dtbo 文件的说明	16

特殊说明

PC 机的 ubuntu 终端输入的命令前加“#”，并且是 root 用户权限；
开发板终端输入的命令前加“\$”。

第一章 Device Tree

1.1. Device Tree 简介

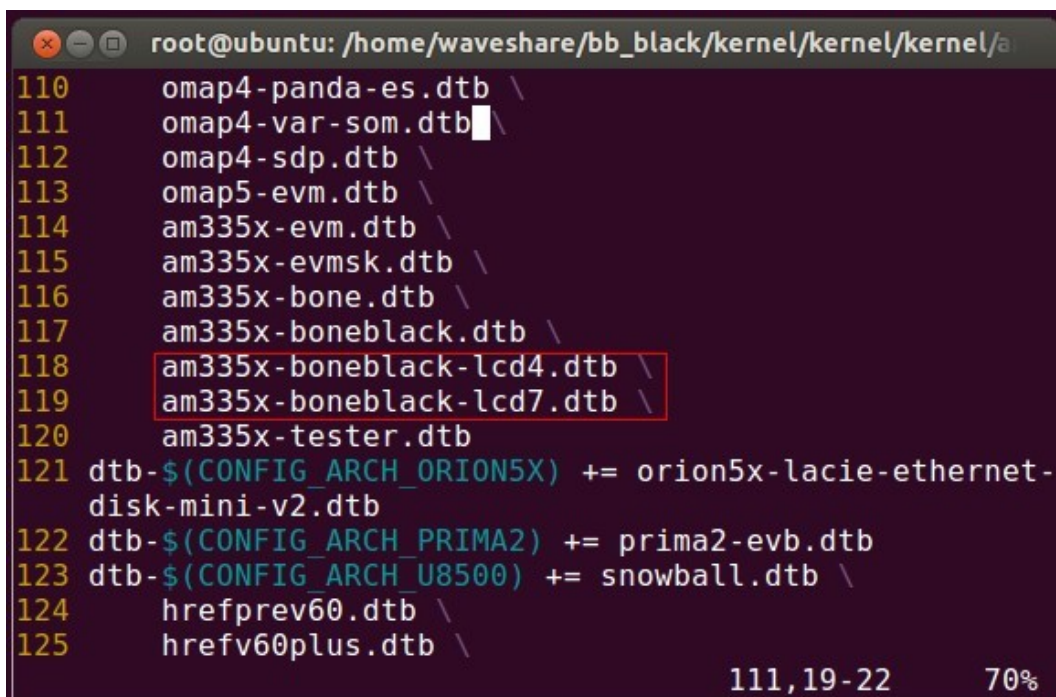
在 过 去 的 ARM Linux 中，arch/arm/plat-xxx 和 arch/arm/mach-xxx 中充斥着大量的垃圾代码，相当多数的代码只是在描述板级细节，而这些板级细节对于内核来讲，不过是垃圾。采用 Device Tree 后，许多硬件细节可以直接透过它传递给 Linux，而不再需要在 kernel 中进行大量的冗余编码。

Device Tree 的主要作用是将某个硬件外设与相应的驱动程序直接进行绑定，系统启动时就会加载这个 Device Tree 并配置各种硬件资源。有时，Device Tree 只负责缺省的硬件配置，因此我们可以使用 Device Tree Overlay 来动态重定义某些功能。Device Tree Overlay 与 Device Tree 类似，都是.dts 文件，它可以在系统启动后动态加载。

1.2. 编写 Device Tree

在 内 核 /arch/arm/boot/dts 目 录 下 编 写

am335x-boneblack-lcd4.dts 、 am335x-boneblack-lcd7.dts 、
am335x-bone-common-lcd4.dtsi 和 am335x-bone-common-lcd7.dtsi
文件，并修改该目录的 Makefile。详细文件在源码\dtb 目录下。



```
root@ubuntu: /home/waveshare/bb_black/kernel/kernel/kernel/a
110 omap4-panda-es.dtb \
111 omap4-var-som.dtb \
112 omap4-sdp.dtb \
113 omap5-evm.dtb \
114 am335x-evm.dtb \
115 am335x-evmsk.dtb \
116 am335x-bone.dtb \
117 am335x-boneblack.dtb \
118 am335x-boneblack-lcd4.dtb \
119 am335x-boneblack-lcd7.dtb \
120 am335x-tester.dtb
121 dtb-$(CONFIG_ARCH_ORION5X) += orion5x-lacie-ethernet-
disk-mini-v2.dtb
122 dtb-$(CONFIG_ARCH_PRIMA2) += prima2-evb.dtb
123 dtb-$(CONFIG_ARCH_U8500) += snowball.dtb \
124 hrefprev60.dtb \
125 hrefv60plus.dtb \
111,19-22 70%
```

在内核顶层目录下执行命令：

#make ulmage dtbs

在/arch/arm/boot/dts 目录下即可产生 am335x-boneblack-lcd4.dtb
和 am335x-boneblack-lcd7.dtb 文件。

1.3. 编写 Device Tree Overlay

在内核/firmware/capes 目录下有各种.dts 文件，用户可以参考
编写自己的.dts 文件。并修改/firmware 目录下的 Makefile。下面
所提到的驱动移植就是以 Device Tree Overlay 的方式，在系统启动
后动态加载。

```
root@ubuntu: /home/waveshare/bb_black/kernel/kernel/kerne
154 BB-BONE-LCD4-01-00A1.dtbo \
155 BB-BONE-LCD7-01-00A2.dtbo \
156 BB-BONE-LCD7-01-00A3.dtbo \
157 BB-BONE-LCD7-01-00A4.dtbo \
158 DVK530-LCD4-01-00A0.dtbo \
159 DVK530-LCD7-01-00A0.dtbo \
160 DVK531-I2C1-01-00A0.dtbo \
161 DVK531-I2C2-01-00A0.dtbo \
162 DVK531-SPI1-01-00A0.dtbo \
163 DVK531-UART2-01-00A0.dtbo \
164 DVK531-CAN1-01-00A0.dtbo \
165 DVK531-RTC-01-00A0.dtbo \
166 DVK531-KEYS-01-00A0.dtbo \
167 DVK531-PWM-01-00A0.dtbo \
168 DVK531-W1-01-00A0.dtbo \
169 BB-BONE-eMMC1-01-00A0.dtbo \
170 BB-BONE-GPEVT-00A0.dtbo \

154,2-5 42%
```

第二章 PWM 驱动移植

2.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-PWM-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

2.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-PWM-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新

到系统，详细操作看“第十四章 系统更新”章节。

2.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-PWM-01 > /sys/devices/bone_capemgr.* /slots
```

注意：bone_capemgr.*中的*号实际是一个数字，但是每次系统启动后这个数字可能会变化，所以用*代替（以下同理）。

第三章 LED 驱动移植

3.1. 编写.dts 文件

不需要重新编写.dts 文件，在 am335x-bone-common-lcd4.dtsi 或 am335x-bone-common-lcd7.dtsi 文件中添加相应代码即可：

```
/*LED4 FOR DVK531*/  
  
board_led1 {  
    label = "led1";  
    gpios = <&gpio3 2 1>;  
    default-state = "off";  
    linux,default-trigger = "none";  
};  
  
board_led2 {  
    label = "led2";
```

```

        gpios = <&gpio3 3 1>;
        default-state = "off";
        linux,default-trigger = "none";
    };

    board_led3 {
        label = "led3";
        gpios = <&gpio3 4 1>;
        default-state = "off";
        linux,default-trigger = "none";
    };

    board_led4 {
        label = "led4";
        gpios = <&gpio3 5 1>;
        default-state = "off";
        linux,default-trigger = "none";
    };
};

```

3.2. 编译.dts 文件

在内核顶层目录下执行命令：

#make ulmage dtbs

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新

到系统，详细操作看“第十四章 系统更新”章节。

第四章 按键驱动移植

4.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-KEYS-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码 \dts 目录下。

4.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-KEYS-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

4.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-KEYS-01 > /sys/devices/bone_capemgr.*/slots
```

第五章 Dallas 1-wires 驱动移植

5.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-W1-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

5.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-W1-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

5.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-W1-01 > /sys/devices/bone_capemgr.*/slots
```

第六章 I2C 驱动移植

6.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-I2C1-01-00A0.dts 和 DVK531-I2C2-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

6.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在 /firmware 目录下即可产生 DVK531-I2C1-01-00A0.dtbo 和 DVK531-I2C2-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

6.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-I2C1-01 > /sys/devices/bone_capemgr.*/slots
```

```
$ echo DVK531-I2C2-01 > /sys/devices/bone_capemgr.*/slots
```

第七章 RTC 驱动移植

7.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-RTC-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

7.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-RTC-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

7.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-RTC-01 > /sys/devices/bone_capemgr.* /slots
```

注意：确保 I2C 驱动已经移植好！

第八章 SPI 驱动移植

8.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-SPI1-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

SPI 的 CS 管脚的控制驱动文件 ctl_io.c 在源码\driver 下，把它拷贝到内核/drivers/char/目录下，同时修改该目录下的 Makefile 和 Kconfig 文件：

Makefile:

```
obj-$(CONFIG_CTL_IO) += ctl_io.o
```

Kconfig:

```
config CTL_IO
```

```
    tristate "ctl_io control driver support"
```

```
    depends on ARCH_OMAP
```

```
    help
```

```
        io control driver
```

配置内核:

#make menuconfig

Device Drivers --->

Character devices --->

<*> ctl_io control driver support

保存后退出。

8.2. 编译.dts 文件

在内核顶层目录下执行命令:

#make ulmage dtbs

在/firmware 目录下即可产生 DVK531-SPI1-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统, 详细操作看“第十四章 系统更新”章节。

8.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-SPI1-01 > /sys/devices/bone_capemgr.*/slots
```

第九章 CAN 驱动移植

9.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-CAN1-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dtb 目录下。

9.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-CAN1-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

9.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-CAN1-01 > /sys/devices/bone_capemgr.*/slots
```

第十章 RS485 驱动移植

10.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK531-UART2-01-00A0.dts 文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dts 目录下。

RS485 用到驱动文件 ctl_io.c，请确保该文件已经移植好，详细看“第八章 SPI 驱动移植”章节。

10.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在/firmware 目录下即可产生 DVK531-UART2-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

10.3. 加载.dtbo 文件

在开发板上执行如下命令加载：

```
$ echo DVK531-UART2-01 > /sys/devices/bone_capemgr.*/slots
```

第十一章 LCD 和 ADC 驱动移植

11.1. 编写.dts 文件

在内核/firmware/capes 目录下编写 DVK530-LCD4-01-00A0.dts (4.3 寸)和 DVK530-LCD7-01-00A0.dts(7 寸)文件，并修改/firmware 目录下的 Makefile。详细.dts 文件在源码\dts 目录下。

11.2. 编译.dts 文件

在内核顶层目录下执行命令：

```
#make ulmage dtbs
```

在 /firmware 目录下即可产生 DVK530-LCD4-01-00A0.dtbo 和 DVK530-LCD7-01-00A0.dtbo 文件。

把/arch/arm/boot 目录下的 ulmage 和/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 更新到系统，详细操作看“第十四章 系统更新”章节。

11.3. 加载.dtbo 文件

DVK530-LCD4-01-00A0.dtbo 或者 DVK530-LCD7-01-00A0.dtbo 的加载是在开发板启动的时候由系统自动加载的。

系统启动后，源码 \tslib 目录下的 50-tslib.conf 和 xf86-input-tslib_0.0.6-r17.1_armv7a-vfp-neon.ipk 两个文件拷贝到开发板（假如拷贝到/mything 目录下），详细拷贝操作看“第十五章 文件拷贝”章节。

执行相应命令：

```
$opkg install  
/mything/xf86-input-tslib_0.0.6-r17.1_armv7a-vfp-neon.ipk  
$cp /mything/50-tslib.conf /usr/share/X11/xorg.conf.d/  
$sync
```

第十二章 USB Camera 驱动移植

内核已经有驱动的支持，用户可以直接使用。通过网络进行视频的捕抓，详细操作参考《mjpg-streamer 移植手册》。

第十三章 USB WIFI 驱动移植

内核已经有驱动的支持，只需要在开发板上安装相应工具即可。

1. 更新下载源：

```
$ opkg update
```

2. 安装无线上网工具：

```
$ opkg install wireless-tools
```

3. 编辑 wpa_supplicant.service 文件：

```
$ vi /lib/systemd/system/wpa_supplicant.service
```

修改

```
ExecStart=/usr/sbin/wpa_supplicant -u
```

为：

```
#ExecStart=/usr/sbin/wpa_supplicant -u
```

4. 拷贝 rtl8192cufw.bin 文件：

在/lib/firmware/目录下新建 rtlwifi 文件夹，把源码\wifi 目录下的 rtl8192cufw.bin 文件拷贝到/lib/firmware/rtlwifi 目录下，详细的拷贝参考“第十五章 文件拷贝”章节。

第十四章 系统更新

这里以更新从 TF 卡启动的系统为例。首先确保 TF 卡中已经烧写有 Angstrom 操作系统。详细烧写过程参考《用户手册》。系统更新参考如下步骤（在 PC 机的 Ubuntu 上操作）：

- 1、 将 TF 卡通过 USB 读卡器连接到 PC 机 ubuntu 中，系统会识别到两个分区 sdb1 和 sdb2（根据你的实际情况）。

- 2、 在/mnt 目录下创建一个文件夹 sdcard。

```
#mkdir /mnt/sdcard
```

- 3、 将/dev/sdb2 挂载到/mnt/sdcard。

```
#mount /dev/sdb2 /mnt/sdcard/
```

- 4、 把/arch/arm/boot 目录下的 ulmage、/arch/arm/boot/dts 目录下的 am335x-boneblack-lcd4.dtb 和 am335x-boneblack-lcd7.dtb 拷贝到/mnt/sdcard/boot 目录下。

```
#cp
```

```
/home/waveshare/bb_black/kernel/kernel/kernel/arch/arm/boot/ulmage /mnt/sdcard/boot/
```

```
#cp
```

```
/home/waveshare/bb_black/kernel/kernel/kernel/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb /mnt/sdcard/boot/
```

```
#cp
```

```
/home/waveshare/bb_black/kernel/kernel/kernel/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb /mnt/sdcard/boot/
```

- 5、 卸载/mnt/sdcard。

```
#umount /mnt/sdcard/
```

- 6、取下 TF 卡即完成更新。

第十五章 文件拷贝

这里以从 TF 卡启动后，往 BeagleBone Black 进行文件的拷贝。这里以 U 盘传输为例，参考如下步骤（在 BeagleBone Black 上操作）：

- 1、将需要拷贝的文件复制到 U 盘，比如：hello.c
- 2、把 U 盘插入 BeagleBone Black 的 USB Host 接口，系统会识别到分区 sda1（根据你的实际情况）。
- 3、挂载 U 盘，在终端输入命令：

```
$mount /dev/sda1 /mnt/card/
```

- 4、在根目录下创建文件夹 mything

```
$mkdir /mything
```

- 5、拷贝 U 盘文件（hello.c）到/mything

```
$cp /mnt/card/hello.c /mything/
```

- 6、卸载 U 盘

```
$umount /mnt/card
```

- 7、同步

```
$sync
```

- 8、拔下 U 盘即完成文件的拷贝。

第十六章 关于卸载.dtbo 文件的说明

- 1、 目前的 Angstrom 系统，卸载.dtbo 文件只能通过重启系统。
- 2、 我们提供基于 TF 卡的 Angstrom 系统镜像
Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.05.24-waveshare.img 在系统上电的时候自动加载相应的.dtbo 文件。即执行/home/xuser/waveshare_demo/目录下的 capes 脚本，在/etc/profile 文件中添加。用户可以通过修改/etc/profile，进行动态加载。