

Porting Drivers

Copyright

All rights reserved by Waveshare Electronics Co., Ltd.
Not allow to modify, distribute, or copy without permission.

Contents

Copyright	1
1. Push Button Driver	3
1.1 Device Tree Overview	3
1.2 Coding the Device Tree	3
1.3 Coding the Device Tree Overlay	4
2. PWM Driver	5
2.1 Coding the .dts File	5
2.2 Compile the .dts File	5
2.3 Load the .dtbo File	5
3. LED Driver	6
3.1 Coding the .dts File	6
3.2 Compile the .dts File	6
4. Push Button Driver	7
4.1 Coding the .dts File	7
4.2 Compile the .dts File	7
4.3 Load the .dtbo File	7
5. Dallas 1-wires Driver	7
5.1 Coding the .dts File	7
5.2 Compile the .dts File	7
5.3 Load the .dtbo File	8
6. I2C Driver	8
6.1 Coding the .dts File	8
6.2 Compile the .dts File	8
6.3 Load the .dtbo File	8
7. RTC Driver	9
7.1 Coding the .dts File	9
7.2 Compile the .dts File	9
7.3 Load the .dtbo File	9
8. SPI Driver	9
8.1 Coding the .dts File	9
8.2 Compile the .dts File	10
	1

8.3	Load the .dtbo File	10
9.	CAN Driver.....	10
9.1	Coding the .dts File.....	10
9.2	Compile the .dts File.....	10
9.3	Load the .dtbo File	11
10.	RS485 Driver.....	11
10.1	Coding the .dts File.....	11
10.2	Compile the .dts File.....	11
10.3	Load the .dtbo File	11
11.	LCD & ADC Driver	11
11.1	Coding the .dts File.....	11
11.2	Compile the .dts File.....	12
11.3	Load the .dtbo File	12
12.	USB Camera Driver	12
13.	USB WIFI Driver	12
14.	System Update	13
15.	Copying Files.....	14
16.	About Unloading .dtbo Files.....	14

1. Push Button Driver

1.1 Device Tree Overview

The Device Tree can be used to pass hardware details to the Linux, eliminating plenty of redundant code in kernel.

Device Tree binds hardware device to its related driver. While the system booting, the Device Tree will be loaded for configuring the hardware devices. Device Tree provides default hardware configuration only, therefore, we can use Device Tree Overlay to re-define custom functions dynamically.

Device Tree Overlay is similar with Device Tree, both of them are .dts files, which can be loaded dynamically after the system booted.

1.2 Coding the Device Tree

Under the `/arch/arm/boot/dts` directory, Coding the following files:

`m335x-boneblack-lcd4.dts`

`am335x-boneblack-lcd7.dts`

`am335x-bone-common-lcd4.dtsi`

`am335x-bone-common-lcd7.dtsi`

And edit the Makefile under the same directory.

The files can be found on `Source-code\dts`.

```
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%
```

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

Two files will be generated under /arch/arm/boot/dts:

```
am335x-boneblack-lcd4.dtb
```

```
am335x-boneblack-lcd7.dtb
```

1.3 Coding the Device Tree Overlay

Users can write their own .dts file and place it in the kernel /firmware/capes directory, then edit the Makefile under /firmware directory.

In the following sections, we'll start porting drivers in Device Tree Overlay's way.

```
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%
```

2. PWM Driver

2.1 Coding the .dts File

Coding the DVK531-PWM-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on [Source-code\dts](#).

2.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-PWM-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

2.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

Note: the * in bone_capemgr.* indicates a number in practice, which keeps changing each time the system boots.

3. LED Driver

3.1 Coding the .dts File

Add the following code into the am335x-bone-common-lcd4.dtsi or am335x-bone-common-lcd7.dtsi file:

```
/*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 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

4. Push Button Driver

4.1 Coding the .dts File

Coding the DVK531-KEYS-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

4.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-KEYS-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

4.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

5. Dallas 1-wires Driver

5.1 Coding the .dts File

Coding the DVK531-W1-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

5.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-W1-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

5.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

6. I2C Driver

6.1 Coding the .dts File

Coding the DVK531-I2C1-01-00A0.dts and DVK531-I2C2-01-00A0.dts files and place them in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

6.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-I2C1-01-00A0.dtbo and DVK531-I2C2-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

6.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

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

7. RTC Driver

7.1 Coding the .dts File

Coding the DVK531-RTC-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

7.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-RTC-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

7.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

```
$ echo DVK531-RTC-01 > /sys/devices/bone_capemgr.* /slots Note: make sure that the I2C driver has been ported first.
```

8. SPI Driver

8.1 Coding the .dts File

Coding the DVK531-SPI1-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

Copy the SPI CS pin control driver file ctl_io.c (which can be found on Source-code\driver) to kernel /drivers/char directory, then edit the Makefile and Kconfig in the same directory:

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
```

Config the kernel:

```
#make menuconfig
```

```
Device Drivers --->
```

```
Character devices --->
```

```
<*> ctl_io control driver support
```

Save and exit.

8.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-SPI1-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

8.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

9. CAN Driver

9.1 Coding the .dts File

Coding the DVK531-CAN1-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

9.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-CAN1-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

9.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

10. RS485 Driver

10.1 Coding the .dts File

Coding the DVK531-UART2-01-00A0.dts file and place it in /firmware/capes, then edit the Makefile under /firmware.

The .dts file can be found on Source-code\dts.

RS485 requires driver file ctl_io.c, please make sure that it has been ported. For more info, refer to "8 SPI Driver" section.

10.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The DVK531-UART2-01-00A0.dtbo will be generated in /firmware.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

10.3 Load the .dtbo File

Execute the following command on development board terminal to load it:

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

11. LCD & ADC Driver

11.1 Coding the .dts File

Coding the DVK530-LCD4-01-00A0.dts (4.3inch) and DVK530-LCD7-01-00A0.dts (7inch)

files and place them in `/firmware/capes`, then edit the Makefile under `/firmware`.
The `.dts` file can be found on `Source-code\dts`.

11.2 Compile the .dts File

Under the kernel root directory, execute:

```
#make ulmage dtbs
```

The `DVK530-LCD4-01-00A0.dtbo` and `DVK530-LCD7-01-00A0.dtbo` will be generated in `/firmware`.

Copy the following files to the system:

```
/arch/arm/boot/ulmage
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb
```

```
/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb
```

For more info about how to copy, please refer to the "14 System Update" section.

11.3 Load the .dtbo File

`DVK530-LCD4-01-00A0.dtbo` or `DVK530-LCD7-01-00A0.dtbo` is automatically loaded when the system boots.

When the system boots successfully, copy the following files to the development board (say, copy them to `/mything`):

```
50-tslib.conf
```

```
xf86-input-tslib_0.0.6-r17.1_armv7a-vfp-neon.ipk
```

The files can be found on `Source-code\tslib`.

For more info about how to copy, refer to "15 Copying Files" section.

Execute the following commands:

```
$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
```

12. USB Camera Driver

Driver is included in the kernel by default.

Users can browse webcams through network, for more info, please refer to <<Porting mjpg-streamer >>.

13. USB WIFI Driver

Driver is included in the kernel by default, just install some required tools:

Update the source:

```
$ opkg update
```

- 1) Install wireless tools:
`$ opkg install wireless-tools`
- 2) Modify the wpa_supplicant.service file:
`$ vi /lib/systemd/system/wpa_supplicant.service`
 Edit:
`ExecStart=/usr/sbin/wpa_supplicant -u`
 As:
`#ExecStart=/usr/sbin/wpa_supplicant -u`
- 3) Copy the rtl8192cufw.bin file:
 Under the /lib/firmware directory, create a new folder rtlwifi, and copy the rtl8192cufw.bin file (which can be found on Source-code\wifi) to the /lib/firmware/rtlwifi directory.
 For more info about how to copy, refer to "15 Copying Files" section.

14. System Update

This section describes how to update a TF card system.

Make sure that the Angstrom system has been written into the TF card. For more info, refer to User Manual.

Follow the steps below to update the system (operates on the PC Ubuntu system):

- 1) Connect the TF card to the PC Ubuntu system via USB card reader, the system will detect two partitions, say, sdb1 and sdb2 (the names might be different in practice).
- 2) Create a new folder sdcard under /mnt directory:
`#mkdir /mnt/sdcard`
- 3) Mount /dev/sdb2 to /mnt/sdcard:
`#mount /dev/sdb2 /mnt/sdcard/`
- 4) Copy the following files to /mnt/sdcard/boot:
`/arch/arm/boot/ulmage`
`/arch/arm/boot/dts/am335x-boneblack-lcd4.dtb`
`/arch/arm/boot/dts/am335x-boneblack-lcd7.dtb`
`#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) Unmount /mnt/sdcard

- ```
#umount /mnt/sdcard/
```
- 6) Disconnect the TF card

## 15. Copying Files

This section describes how to copy files to the BeagleBone Black after the system boots from TF card.

Follow the steps below (operates on the BeagleBone Black):

- 1) Copy the target file to a USB flash drive, say, hello.c
- 2) Insert the USB flash drive to the BeagleBone Black USB Host connector, the system will detect a partition sda1 (the name might be different in practice).
- 3) Mount the USB flash drive:  

```
$mount /dev/sda1 /mnt/card/
```
- 4) Create a new folder /mything under root directory:  

```
$mkdir /mything
```
- 5) Copy the target file in USB flash drive (hello.c) to /mything  

```
$cp /mnt/card/hello.c /mything/
```
- 6) Unmount the USB flash drive:  

```
$umount /mnt/card
```
- 7) Synchronize  

```
$sync
```
- 8) Disconnect the USB flash drive

## 16. About Unloading .dtbo Files

- 1) In order to unload the .dtbo files, you have to reboot the Angstrom system.
- 2) We provide an Angstrom system based on TF card: Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.05.24-waveshare.img, which will automatically load related .dtbo files while the system powers up. That is, it will execute the capes scripts under /home/xuser/waveshare\_demo/ directory. Users can modify the /etc/profile for customization.