RAPID PROTOTYPING AND LINUX KERNEL DEVELOPMENT WITH THE POCKETBEAGLE®

SuperCon 2017

WiFi Access Point & Local File Server

SSID: DigiKey-PocketBeagle-2.4 or DigiKey-PocketBeagle-5

Pass: digikey2017

http://192.168.8.10/

OS - Windows/Mac Users: (Use VirtualBox)

Download/Install from: http://192.168.8.10/VirtualBox/ (or wait for the USB flash drive)

VirtualBox Installer and debian based image:

- VirtualBox-5.1.x-x-<OSX/Win>.<filename>
- Oracle VM VirtualBox*.extpack
- Debian-PocketBeagle*.ova

The extpack need to be installed: File -> Preferences ->Extensions

OS - linux

```
git clone https://github.com/RobertCNelson/Supercon-2017-PocketBeagle
```

Pull down Updates (VirtualBox users)

```
cd ./Supercon-2017-PocketBeagle/

git pull
```

Download large project files:

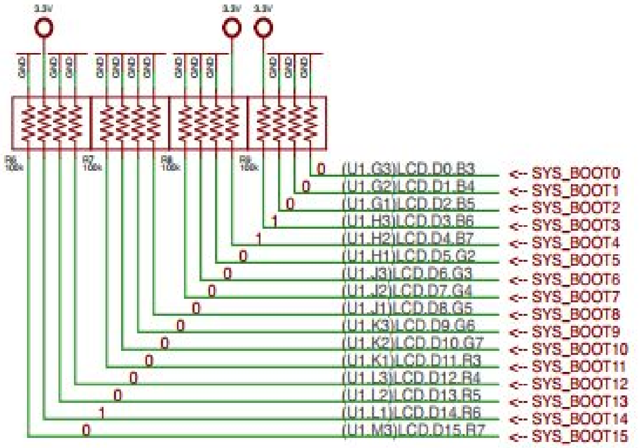
Run: “./scripts/get_all.sh”

- GCC Toolchain: gcc-linaro-6.4.1-2017.08
- Linux: 4.14-rc8
- U-Boot: 2017.11-rc4
- Base Rootfs: debian-9.2-iot-armhf-2017-11-08

TI AM335x: bootrom

From: (AM335x and AMIC110 Sitara™ Processors Technical Reference Manual (Rev. P))

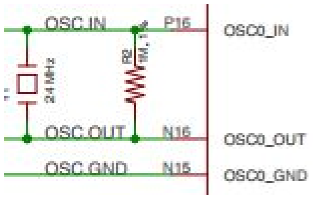
- http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (page 5032)



3.3V
3.3V
3.3V
GND
R6 100k
R7 100k
R8 100k
R9 100k
0 (U1.G3)LCD.D0.B3 <-- SYS_BOOT0
0 (U1.G2)LCD.D1.B4 <-- SYS_BOOT1
0 (U1.G1)LCD.D2.B5 <-- SYS_BOOT2
1 (U1.H3)LCD.D3.B6 <-- SYS_BOOT3
1 (U1.H2)LCD.D4.B7 <-- SYS_BOOT4
0 (U1.H1)LCD.D5.G2 <-- SYS_BOOT5
0 (U1.J3)LCD.D6.G3 <-- SYS_BOOT6
0 (U1.J2)LCD.D7.G4 <-- SYS_BOOT7
0 (U1.J1)LCD.D8.G5 <-- SYS_BOOT8
0 (U1.K3)LCD.D9.G6 <-- SYS_BOOT9
0 (U1.K2)LCD.D10.G7 <-- SYS_BOOT10
0 (U1.K1)LCD.D11.R3 <-- SYS_BOOT11
0 (U1.L3)LCD.D12.R4 <-- SYS_BOOT12
0 (U1.L2)LCD.D13.R5 <-- SYS_BOOT13
1 (U1.L1)LCD.D14.R6 <-- SYS_BOOT14
0 (U1.M3)LCD.D15.R7 <-- SYS_BOOT15

SYSBOOT[15:14] = 01 = 24Mhz
SYSBOOT[4:0] = 11000

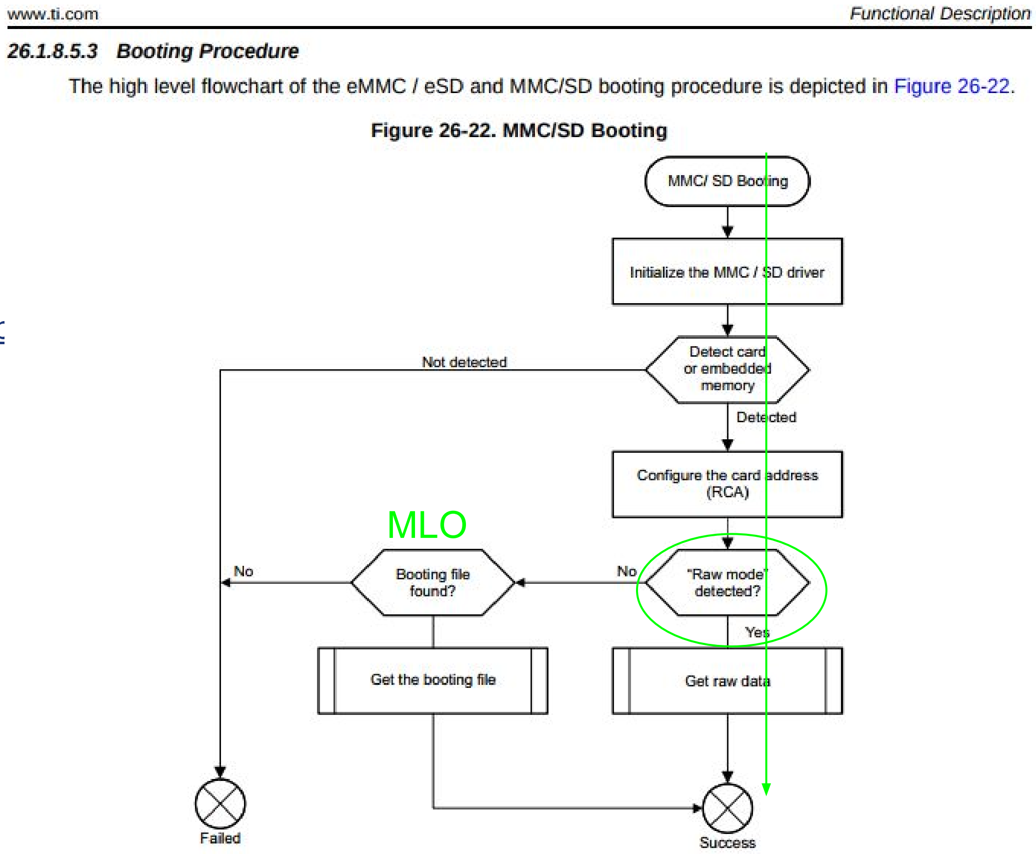
1. SPI0
2. MMC0 - going to use today
3. USB0 - (node-beagle-boot)
4. UART0



TI AM335x: bootrom

http://www.ti.com/lit/ug/spruh73p/spruh73p

Page: 5053



www.ti.com Functional Description

26.1.8.5.3 Booting Procedure

The high level flowchart of the eMMC / eSD and MMC/SD booting procedure is depicted in Figure 26-22.

Figure 26-22. MMC/SD Booting

TI AM335x: bootrom: raw mode:

http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (Page: 5054)

1. 0x0 <- (FAT Boot Sector, let's leave it blank...)
2. 0x20000 (128KB) <- We are going to use this location
3. 0x40000 (256KB) <- (2nd “backup” location)
4. 0x60000 (384KB) <- (3rd “backup” location)

Only 128KB in size… (hint, only 128KB of SRAM)

Das U-Boot (the Universal Boot Loader) U-Boot

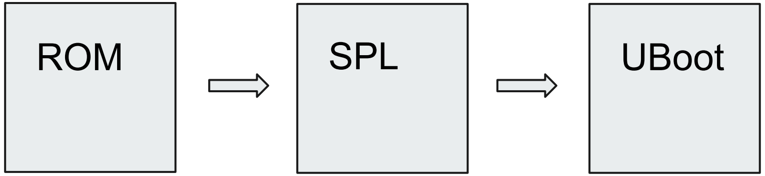
Original Author: Wolfgang Denk, now maintained by Tom Rini

- https://www.denx.de/wiki/U-Boot
- http://git.denx.de/?p=u-boot.git;a=summary
- https://en.wikipedia.org/wiki/Das_U-Boot

U-Boot: AM335x

Outputs two files for TI am335x targets:

- MLO = SPL (or Secondary Program Loader)
- u-boot.img (or u-boot-dtb.img) (U-Boot)



U-Boot: SPL

1. Initializes main memory (DDRx for am335x)
2. Loads full (U-Boot) into DDR memory

Or:

1. Initializes main memory (DDRx for am335x)
2. Loads Linux Kernel into DDR memory (aka: Falcon mode, faster boot mode/etc)

U-Boot:

- Network
- USB
- MMC
- File System (fat/extX)
- Shell

Sometimes you don’t need a full OS, have U-Boot init and then have U-Boot load/run your application.

U-Boot:

```
CPU  : AM335X-GP rev 2.1
I2C:   ready
DRAM:  512 MiB
Some drivers were not found
Reset Source: Power-on reset has occurred.
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment

Board: BeagleBone Black
<ethaddr> not set. Validating first E-fuse MAC
BeagleBone Black:
Model: SeeedStudio BeagleBone Green:
```

U-Boot: (SPL for reference)

```
U-Boot SPL 2017.11-rc4-00002-g58430b9263 (Nov 07 2017 - 16:38:38)
Trying to boot from MMC1

U-Boot 2017.11-rc4-00002-g58430b9263 (Nov 07 2017 - 16:38:38 -0600)
```

U-Boot: Step 1: untar & patch

(Don’t worry about typing this)

```
tar xf u-boot-2017.11-rc4.tar.bz2
cd ./u-boot-2017.11-rc4
patch -p1 < ../0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
patch -p1 < ../0002-U-Boot-BeagleBone-Cape-Manager.patch
```

0001: This is our main “Beagle” patchset, most of it is supporting things we did WRONG, thus non-mainline-able.
0002: This is our new “U-Boot Overlays” patchset, still under development. Needs to be cleaned up and pushed mainline.

U-Boot: Step 2: build commands

(Don’t worry about typing this)

- make ARCH=arm CROSS_COMPILE=${CC} distclean
- make ARCH=arm CROSS_COMPILE=${CC} am335x_evm_defconfig
- make ARCH=arm CROSS_COMPILE=${CC}

Type this:

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/build_u-boot.sh
```

U-Boot: Step 3: (waiting...)

U-Boot: Step 4: Everything Built right?

```
 MKIMAGE MLO
  MKIMAGE MLO.byteswap
  CFGCHK  u-boot.cfg
'./MLO' -> '../../deploy/MLO'
'./u-boot.img' -> '../../deploy/u-boot.img'
total 452K
-rw-r--r-- 1 voodoo voodoo  73K Nov  9 17:57 MLO
-rw-r--r-- 1 voodoo voodoo 373K Nov  9 17:57 u-boot.img
```

- MLO < 128KB (thus it'll fit in SRAM)
- u-boot.img > 128KB (thus we need main DDR up and running)

U-Boot: microSD

Insert USB-microSD adapter, and type “lsblk”

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ lsblk
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda         8:0    0 465.8G  0 disk
└─sda1      8:1    0 465.8G  0 part /
sde         8:64   1   7.4G  0 disk
└─sde1      8:65   1   7.4G  0 part
```

That’s our 8GB USB Flash Drive

Open: system.sh change: MMC=/dev/sde

U-Boot: Format microSD

```
sudo dd if=/dev/zero of=${MMC} bs=1M count=10

sudo sfdisk ${MMC} <<-__EOF__
4M,,L,*
__EOF__

sudo mkfs.ext4 -L rootfs ${MMC}1
```

U-Boot: Format microSD

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/format_drive.sh
```

U-Boot: (refresh for your memory)

http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (Page: 5054)

1. 0x0
2. 0x20000 (128KB) <- We are going to use this location
3. 0x40000 (256KB)
4. 0x60000 (384KB)

```
sudo dd if=./deploy/MLO of=${MMC} count=1 seek=1 bs=128k
sudo dd if=./deploy/u-boot.img of=${MMC} count=2 seek=1 bs=384k
```

U-Boot: Install U-Boot

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/install_u-boot.sh
0+1 records in
0+1 records out
74744 bytes (75 kB, 73 KiB) copied, 0.0134723 s, 5.5 MB/s
0+1 records in
0+1 records out
381180 bytes (381 kB, 372 KiB) copied, 0.000987122 s, 386 MB/s
```

U-Boot: Verify... (Grab your board)

Favorite serial terminal: tio/screen/gtkterm/cuteterm/etc.

Baud rate (default: 115200)
Data bits (default: 8)
Flow control (default: none)

U-Boot: Verify... (Grab your board)

```
U-Boot SPL 2017.11-rc4 (Nov 09 2017 - 18:28:18)
Trying to boot from MMC1

U-Boot 2017.11-rc4 (Nov 09 2017 - 18:28:18 -0600)

CPU  : AM335X-GP rev 2.1
I2C:   ready
DRAM:  512 MiB
No match for driver 'omap_hsmmc'
No match for driver 'omap_hsmmc'
Some drivers were not found
Reset Source: Power-on reset has occurred.
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment

Model: BeagleBoard.org PocketBeagle
```

U-Boot: Verify... (Grab your board)

```
Checking for: /uEnv.txt ...
Checking for: /boot.scr ...
Checking for: /boot/boot.scr ...
Checking for: /boot/uEnv.txt ...
** Invalid partition 2 **
** Invalid partition 3 **
** Invalid partition 4 **
** Invalid partition 5 **
** Invalid partition 6 **
** Invalid partition 7 **
Card did not respond to voltage select!
```

We need to install the rootfs & kernel...

Base Rootfs: Debian 9.x (Stretch)

Maintainer: me (with lots of help from all the Debian Developers and 1000’s of other users)

- https://elinux.org/Beagleboard:BeagleBoneBlack_Debian#2017-11-05_-_Debian_9_.28Stretch.29_-_Weekly
- https://www.debian.org/
- https://github.com/beagleboard/image-builder

Base Rootfs: Install

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/install_rootfs.sh
```

Linux Kernel

Maintainer: Linus Torvalds

- https://www.kernel.org/
- https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/

Linux Kernel: Step 1: untar & patch

(Don't worry about typing this)

- tar xf linux-4.14-rc8.tar.gz
- cd ./linux-4.14-rc8/
- patch -p1 < ../patch-4.14-rc8-bone3.diff

0001: This is our main "Beagle" patchset, a big chunk of it is Kernel Overlays which are EOL (U-Boot Overlays are replacing them), the rest are things being staged for mainline...

Linux Kernel: Step 2: build commands

(Don’t worry about typing this)

- `make ARCH=arm CROSS_COMPILE=${CC} distclean`
- `cp -v ../defconfig ./.config`
- `make ARCH=arm CROSS_COMPILE=${CC} menuconfig`
- `make -j${CORES} ARCH=arm CROSS_COMPILE=${CC} zImage modules`
- `make ARCH=arm CROSS_COMPILE=${CC} dtbs`

Type this:

- `voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/build_linux.sh`

Linux Kernel: Step 3: (waiting...)

Tony Lindgren (omap maintainer)

- https://git.kernel.org/pub/scm/linux/kernel/git/tmlind/linux-omap.git/

ARM Tree: Arnd Bergmann, Olof Johansson, and others..

- https://git.kernel.org/pub/scm/linux/kernel/git/arm/arm-soc.git/

Linux Kernel: Step 4: Everything Built right?

```
'arch/arm/boot/zImage' -> '../../deploy/zImage'
'arch/arm/boot/dts/am335x-pocketbeagle.dtb' -> '../../deploy/am335x-pocketbeagle.dtb'
total 3.7M
-rw-r--r-- 1 voodoo voodoo 116K Nov  9 19:02 am335x-pocketbeagle.dtb
-rw-r--r-- 1 voodoo voodoo  73K Nov  9 18:28 MLO
-rw-r--r-- 1 voodoo voodoo 498K Nov  9 19:02 modules.tar.gz
-rw-r--r-- 1 voodoo voodoo 373K Nov  9 18:28 u-boot.img
-rwxr-xr-x 1 voodoo voodoo 2.7M Nov  9 19:02 zImage
```

Linux Kernel:

zImage:

- Linux Kernel

am335x-pocketbeagle.dtb:

- Device Tree Binary - https://elinux.org/index.php?title=Device_Tree_Reference

modules.tar.gz:

- Linux Kernel Modules

Linux Kernel: U-Boot Load address

zImage:

- 0x82000000

am335x-pocketbeagle.dtb:

- 0x88000000

Linux Kernel:

```
loading /boot/vmlinuz-4.14.0-rc8-bone3 ...
2733840 bytes read in 211 ms (12.4 MiB/s)
loading /boot/dtbs/4.14.0-rc8-bone3/am335x-pocketbeagle.dtb ...
118611 bytes read in 52 ms (2.2 MiB/s)
uboot_overlays: add [enable_uboot_overlays=1] to /boot/uEnv.txt to enable...
debug: [console=ttyO0,115200n8 root=/dev/mmcblk0p1 ro rootfstype=ext4 rootwait coherent_pool=1M net.ifnames=0 quiet] ...
debug: [bootz 0x82000000 - 0x88000000] ...
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Device Tree to 8ffe0000, end 8fffff52 ... OK

Starting kernel ...
```

Debian!

```
Debian GNU/Linux 9 beaglebone ttyS0

BeagleBoard.org Debian Image 2017-11-08

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

beaglebone login:
```

Cheat Sheets and Led’s

LED:

- Not color coordinated
- - = GND, + = GPIO

P2_02: Classic GPIO export “./01_gpio.sh”

P2_02: GPIO 59

```
debian@beaglebone:~$ cd /sys/class/gpio/gpio59/
debian@beaglebone:/sys/class/gpio/gpio59$ cat direction
in
debian@beaglebone:/sys/class/gpio/gpio59$ echo in > direction
debian@beaglebone:/sys/class/gpio/gpio59$ echo out > direction
debian@beaglebone:/sys/class/gpio/gpio59$ echo 1 > value
```

P2_02: config-pin: gpio “./02_gpio.sh”

P2_02: GPIO 59

```
debian@beaglebone:~$ config-pin -q P2_02
P2_02 Mode: default Direction: in Value: 0
debian@beaglebone:~$ config-pin -l P2_02
default gpio gpio_pu gpio_pd
debian@beaglebone:~$ config-pin P2_02 gpio_pu
```

P2_01: gpio & pwm! “./03_gpio_pwm.sh”

P2_01: GPIO 50

```
debian@beaglebone:~$ config-pin -q P2_01
P2_01 Mode: default Direction: in Value: 0
debian@beaglebone:~$ config-pin -l P2_01
default gpio gpio_pu gpio_pd pwm

debian@beaglebone:~$ config-pin P2_01 pwm
debian@beaglebone:~$ config-pin -q P2_01
P2_01 Mode: pwm
```

P2_01: gpio & pwm! “./04_gpio_pwm.sh”

P2_01: GPIO 50

```
cd /sys/devices/platform/ocp/48302000.epwmss/48302200.pwm/pwm/pwmchip*/
```

```
drwxrwxr-x 3 root pwm     0 Nov  8 18:10 .
drwxr-xr-x 3 root root    0 Nov  3  2016 ..
lrwxrwxrwx 1 root pwm     0 Nov  8 18:10 device -> ../../../48302200.pwm
-rw-rw---- 1 root pwm  4.0K Nov  8 18:10 export
-rw-rw-r-- 1 root pwm  4.0K Nov  8 18:10 npwm
drwxrwxr-x 2 root pwm     0 Nov  8 18:10 power
lrwxrwxrwx 1 root pwm     0 Nov  8 18:10 subsystem -> ../../../../../../../class/pwm
-rw-rw-r-- 1 root pwm  4.0K Nov  8 18:10 uevent
-rw-rw---- 1 root pwm  4.0K Nov  8 18:10 unexport
```

P2_01: gpio & pwm! “./05_gpio_pwm.sh”

P2_01: GPIO 50

```
echo 0 > export
drwxrwxr-x 3 root pwm    0 Nov  8 18:42 pwm-2:0
```

```
echo 20000 > pwm-2\:0/period
```

```
echo 10000 > pwm-2\:0/duty_cycle
```

```
echo 1 > pwm-2\:0/enable
```

Special Thanks to Chris For helping!

- 37 boards all boot/usb tested
- 68 * 37 headers pins to solder
- 40 * 37 long headers to cut (I picked out the wrong header)
- 37 * 2 headers to sandpaper to fit
- 5 * 2 * 37 wires to solder to led’s (and then trim)
- 5 * 37 usb headers to solder
- 1 * 37 header pins to short for USB host to work

Took Chris and I about two 8 hour days...