BeagleSoftCompile
Beagle Board Software Compilation Procedure.

Updated Aug 30, 2008 by jkridner

Beagle Board Software Compilation Options and Procedure

Related Links:

Beagle Source and Tools

Beagle Software Booting Procedure

Compiling x-loader

Compiling x-loader for NAND booting

- **In file** *include/configs/omap3530beagle.h*
- **Disable the "CFG_CMD_MMC" Macro**

```
/* For X-loader to be flashed on to NAND disable the below macro */
//#define CFG_CMD_MMC               1
```

- Compile the x-loader as shown below

```
 make CROSS_COMPILE=arm-none-linux-gnueabi- distclean
make CROSS_COMPILE=arm-none-linux-gnueabi- omap3530beagle_config
make CROSS_COMPILE=arm-none-linux-gnueabi-
```

File named "x-load.bin" will be generated

- Convert x-load.bin to x-load.bin.ift (required to FLASH x-loader to NAND)
  1. Use the "SignGP" tool to sign the x-loader image.

```
./signGP x-load.bin
```

  2. Copy x-load.bin.ift to MMC/SD card using a card reader/writer or download it through UART.

Prebuilt Image for testing (Save this as x-load.bin.ift)

Compiling x-loader for MMC booting

- **In file** *include/configs/omap3530beagle.h*
- **Enable the "CFG_CMD_MMC" Macro**

```
/* For X-loader to be flashed on to NAND disable the below macro */
#define CFG_CMD_MMC               1
```

- Compile the x-loader as shown below

```
 make CROSS_COMPILE=arm-none-linux-gnueabi- distclean
make CROSS_COMPILE=arm-none-linux-gnueabi- omap3530beagle_config
make CROSS_COMPILE=arm-none-linux-gnueabi-
```

File named "x-load.bin" will be generated

- Convert x-load.bin to MLO (required for MMC booting)
  1. Use the "SignGP" tool to sign the x-loader image.

```
./signGP x-load.bin
```

  2. Rename x-load.bin.ift to MLO

3. Copy MLO to MMC/SD card using a card reader/writer.

Prebuilt Image for testing (Save this as MLO)

Compiling u-boot

Compiling u-boot for Flashing NAND automatically

- **In file** *include/configs/omap3530beagle.h*
- **Enable the CONFIG_BOOTCOMMAND Macro as shown below**

```
Un comment the below CONFIG_BOOTCMD

#define CONFIG_BOOTCOMMAND      \
 "mmcinit;fatload mmc 0 0x80200000 x-load.bin.ift;\
 nand unlock;nand ecc hw;nand erase 0 80000;nand write.i 0x80200000 0 80000;\
 fatload mmc 0 0x80200000 flash-uboot.bin; nand unlock;\
 nand ecc sw;nand erase 80000 160000; nand write.i 0x80200000 80000 160000;\0"

Comment the below line as shown below

/* #define CONFIG_BOOTCOMMAND "\0" */
```

- Comiple the u-boot as shown below

```
 make CROSS_COMPILE=arm-none-linux-gnueabi- distclean
make CROSS_COMPILE=arm-none-linux-gnueabi- omap3530beagle_config
make CROSS_COMPILE=arm-none-linux-gnueabi-
```

File named "u-boot.bin" will be generated

Prebuilt Image for testing (Save this as u-boot.bin)

Compiling u-boot for regular Kernel Booting

- **In file** *include/configs/omap3530beagle.h*
- **Enable the CONFIG_BOOTCOMMAND Macro as shown below**

```
Comment the below CONFIG_BOOTCOMMAND macro

/*
#define CONFIG_BOOTCOMMAND      \
 "mmcinit;fatload mmc 0 0x80200000 x-load.bin.ift;\
 nand unlock;nand ecc hw;nand erase 0 80000;nand write.i 0x80200000 0 80000;\
 fatload mmc 0 0x80200000 flash-uboot.bin; nand unlock;\
 nand ecc sw;nand erase 80000 160000; nand write.i 0x80200000 80000 160000;\0"
*/

Un-comment CONFIG_BOOTCOMMAND macro as shown below
#define CONFIG_BOOTCOMMAND "\0"
```

- Comiple the u-boot as shown below

```
 make CROSS_COMPILE=arm-none-linux-gnueabi- distclean
make CROSS_COMPILE=arm-none-linux-gnueabi- omap3530beagle_config
make CROSS_COMPILE=arm-none-linux-gnueabi-
```

File named "u-boot.bin" will be generated

Prebuilt Image for testing (Save this as u-boot.bin for booting over MMC) (Save the same as flash-uboot.bin in MMC for flashing automatically to NAND)

Compiling Kernel

- Compile the Kernel as shown below

```
 make CROSS_COMPILE=arm-none-linux-gnueabi- distclean
make CROSS_COMPILE=arm-none-linux-gnueabi- omap3_beagle_defconfig
make CROSS_COMPILE=arm-none-linux-gnueabi- uImage
```

File named "uImage" will be generated in arch/arm/boot directory

Prebuilt Kernel Image for testing (Save this as uImage)

Comment by phuongminh.dang, Mar 26, 2009

I'd build the ethernet gadget module (omap_udc.ko & g_ether.ko) but I can not insert these modules, what wrong ???

$ insmod omap_udc.ko $ insmod g_ether.ko $ insmod: cannot insert 'g_hid.ko': No such device

go to omap_udc.c and I found the regiter function (usb_gadget_register_driver()) break down at the following line :

/ **basic sanity tests** / if (!udc)

return -ENODEV;

I used printk function to find out this bug and found that the omap_udc_probe() can not be called here! (just only init function is called).

I want to use omap_udc.c to develop my own HID keyboard and mouse gadget devices! musb_hdrc seem to be impossible for this purpose !!!

Sign in to add a comment