

DFU function on AM1808/OMAPL138 with USB

Concept - What is DFU

- USB Device Firmware Upgrade (DFU) is an official USB device class specification of the USB Implementers Forum
- The current version is 1.1, it can be found for free download at http://www.usb.org/developers/devclass_docs/DFU_1.1.pdf

Concept – Why we need DFU

- On AM1808, we cannot support USB boot mode
- Customer need to upgrade the images/firmware (bootloader, kernel and FS) realtime
- After product release / sale, end user need to upgrade the images at home
- A easy tool and easy connection is important for end users

AM1808 firmware/images upgrade

- For AM1808/OMAPL138, we can upgrade the Linux images with Ethernet, USB host, and DFU (as a USB slave)
- Ethernet – tftp
- USB host – USB2.0 OTG works as a host to read out the data from external mass storage
- All up 2 methods need the end user to operate in the terminal application. It is unsafe and difficult for them.
- DFU – connect the AM1808 board to a WinXP PC as a slave device
 - Users can operate on PC side only

AM1808 USB DFU

- DFU is done at UBoot phase – so, need ensure the uboot is working well
- Request a UBoot patch to enable the DFU function on target side
 - Standard Denx's u-boot tree doesn't support DFU but Openmoko has DFU support in it's own uboot tree [http:// git.openmoko. org/ ?p=u-boot.git;a=summary](http://git.openmoko.org/?p=u-boot.git;a=summary)
 - TI has ported DFU driver from above tree to Denx's u-boot-ti tree.
- Need a PC side utility to connect to board and handle the upgrade operation
 - using Openmoko's host side dfu-utils which can be downloaded at [http:// wiki.openmoko.org/wiki/Dfu-util-windows](http://wiki.openmoko.org/wiki/Dfu-util-windows)
- Need to upgrade the whole images (firmware)
 - unfortunately, we can only upgrade the kernel and FS. and the UBL and UBoot cannot be supported as the NAND ECC difference(Romcode/UBL and Uboot ECC different).
- Only NAND supported

Examples on AM1808/OMAPL138

- Apply the patch and recompile UBoot
- Download UBoot to board with serial port flash utility
- In UBoot terminal, select the default mtdparts
AM1808> mtdparts default
- Connect the board to WinXP PC with a USB cable from the USB2.0 OTG port
- A USB device will be enumerated on PC
- In DOS command window,

DOS> dfu-utils.exe -l

DOS> dfu-util.exe -d 0x0451:0xa4a7 -a altN -D path/to/file/to/be/flashed

here, altN is the Altsetting of the DFU, for example “kernel” or “4”

Left issues

- How to enable the UBL and UBOOT upgrade with DFU (ECC map coherent for RBL/UBL and Uboot)
- After the kernel upgrade, the UBoot is destroyed – need to check where is the UBoot location in NAND flash
- Customer need to develop a PC side application based on the openmoko utility. Can they get the full source code?
- How many customer want this upgrade approach on their device (with NAND flash on it and allow end user to upgrade firmware based on USB)