

Universal Update Utility (UUU) Tool

Andres Lopez

Presenter title goes here
Second line title goes here

September 2019 | Session #AMF-SOL-T3810



SECURE CONNECTIONS
FOR A SMARTER WORLD

UUU vs. Mfgtools

UUU



Mfgtools



UUU vs. Mfgtools

UUU

```
2:1 1/1 [Done] SDPS: boot -f flash.bin
C:\Users\nxa23210\uuu>msvc\x64\Release\uuu.exe -d msvc\imx8qxp\imx8qxp
uuu (universal update utility) for nxp imx chips -- libuuu-1.0.1-gf18a964

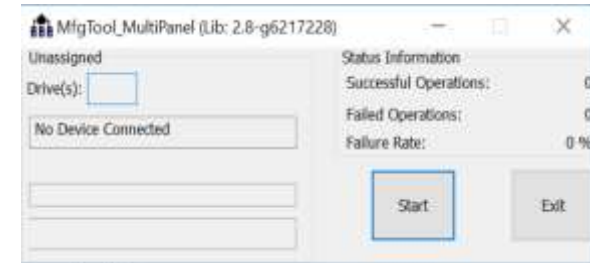
*Access 2 Failure 1
C:\Users\nxa23210\uuu>
2:1 11/11 [Done] FB: done
C:\Users\nxa23210\uuu>

C:\Users\nxa23210\uuu>msvc\x64\Release\uuu.exe -d msvc\imx8qxp\imx8qxp
uuu (universal update utility) for nxp imx chips -- libuuu-1.0.1-gf18a964

*Access 1 Failure 0
C:\Users\nxa23210\uuu>msvc\x64\Release\uuu.exe -d msvc\imx8qxp\imx8qxp
2:1 11/11 [Done] FB: done
C:\Users\nxa23210\uuu>cd ..
```

- Command line based
- Multi Device Supports
 - Device quantity not limited by UUU
 - Only limited by physical port number
- Simple pure text script

Mfgtools



- GUI Interface
- Support Max 4 devices
- XML file based

Quick User Guide

- **Run uboot only**
 - uuu flash.bin(i.MX 8/8X) \
 - uuu u-boot.imx(i.MX 6|i.MX 7/7ULP)
- **Burn android Image into eMMC**
 - uuu android.zip (android release package)
- **Burn Linux Image into eMMC**
 - uuu <yocto release package>
- **Other storage need write uuu script. See below pages**
 - <https://github.com/codeauroraforum/mfgtools/wiki>

How to Support Multi Board

- If boards is the same

- `uuu -d flash.bin.` (uuu will monitor all ports for all known type boards)

- If boards is the difference

- `uuu -d -m 1:2 -m 2:3 flash.bin` (monitor port 1:2 and 1:3 only and download flash.bin for known boards)

- `uuu -d -m 5:6 -m 5:7 u-boot.imx` (monitor port 5:6 and 5:7 only and download u-boot.imx for known boards)

How to Change u-boot Environment

- **Uuu basic command format**
 - <protocol name>: <cmd>: parameter
- **uuu using android fastboot protocol to community with uboot**
 - Default uboot will auto run fastboot cmd if boot from USB serial download
 - FB: ucmd: <any uboot command>
- **For example:**
 - FB: ucmd: setenv server ip 10.45.76.124

How to Run a Script by uuu

- `uuu uuu.lst`
- `uuu.lst` is a simple text script file
 - First line must be `uuu_version 1.0.1`
 - 1.0.1 show required uuu's minimized version
 - Followed by any commands
 - Include “done” command on the last step
- `uuu -s` enables shell mode
 - You can input command and run it
 - `Uuu.inputlog` will record all input command
 - You can save `uuu.inputlog` as a `.lst` file and run it by using `uuu <your new file> .lst`

How to Burn Android Image to eMMC

- SDPS: boot -f flash.bin
- FB: flash gpt partition-table.img
- FB: flash boot_a boot-imx8qxp.img
- FB: flash system_a system.img
- FB: flash vendor_a vendor.img
- FB: flash vbmeta_a vbmeta-imx8qxp.img
- FB: ucmd setenv fastboot_buffer \${loadaddr}
- FB: download -f u-boot-imx8qxp.imx
- FB: ucmd setexpr fastboot_blk \${fastboot_bytes}
- FB: ucmd setexpr fastboot_blk \${fastboot_blk} + 0x1FF
- FB: ucmd setexpr fastboot_blk \${fastboot_blk} / 0x200
- FB: ucmd mmc partconf 0 1 1 1
- FB: ucmd echo \${fastboot_buffer}
- FB: ucmd echo \${fastboot_blk}
- FB: ucmd mmc write \${fastboot_buffer} 0x40 \${fastboot_blk}
- FB: ucmd mmc partconf 0 1 1 0
- FB: Done

How Customer Burn Android Image by uuu (i.MX 8X Example)

- Customer just run the command below
 - **uuu imx8qxp_mek_android.zip**
- uuu will search auto.uuu in root directory of imx8qxp_mek_android.zip and run it

How to Burn Yocto Image to eMMC

- **SDPS:** boot -f flash.bin
- **FB:** ucmd setenv fastboot_dev mmc
- **FB:** ucmd setenv mmcdev \${emmc_dev}
- **FB:** flash -raw2sparse all fsl-image-validation-imx-imx8qxpmeek-20180516162233.rootfs.sdcard
- **FB:** flash bootloader flash.bin
- **FB:** ucmd mmc partconf 0 1 1 0
- **FB:** Done

How Customer Burn Yocto Image to eMMC

- **uuu fsl-image-validation-imx-imx8qxpmeek-20180516162233.rootfs.sdcard**
- **uuu <release package>**

- **Basic uuu.auto look like**

- SDPS: boot -f flash.bin
- FB: ucmd setenv fastboot_dev mmc
- FB: ucmd setenv mmcdev \${emmc_dev}
- FB: ucmd mmc dev \${emmc_dev}
- FB: flash -raw2sparse raw ..
- FB: flash bootloader flash.bin
- FB: ucmd mmc partconf 0 1 1 0
- FB: Done

Whole Scripts, Which Equal as Old mfgtools eMMC Burn

- `uuu_version 1.0.1`
- `SDPS: boot -f flash_mfg.bin`
- `FBK: ucp mkscdcard.sh t:/tmp`
- `FBK: ucmd chmod 777 /tmp/mkscdcard.sh`
- `FBK: ucmd /tmp/mkscdcard.sh /dev/mmcbk0`
- `FBK: ucmd dd if=/dev/zero of=/dev/mmcbk0 bs=1k seek=4096 count=1`
- `FBK: ucmd sync`
- `FBK: ucmd echo 0 > /sys/block/mmcbk0boot0/force_ro`
- `FBK: ucp flash.bin t:/tmp`
- `FBK: ucmd dd if=/tmp/flash.bin of=/dev/mmc0boot0 bs=1K seek=32`
- `FBK: ucmd echo 1 > /sys/block/mmcbk0boot0/force_ro`
- `FBK: ucmd while [! -e /dev/mmcbk0p1]; do sleep 1; done`
- `FBK: ucmd mkfs.vfat /dev/mmcbk0p1`
- `FBK: ucmd mkdir -p /mnt/mmcbk0p1`
- `FBK: ucmd mount -t vfat /dev/mmcbk0p1 /mnt/mmcbk0p1`
- `FBK: ucp Image t:/mnt/mmcbk0p1`
- `FBK: ucp fsl-imx8qxp-mek.dtb t:/mnt/mmcbk0p1`
- `FBK: ucmd umount /mnt/mmcbk0p1`
- `FBK: ucmd mkfs.ext3 -F -E nodiscard /dev/mmcbk0p2`
- `FBK: ucmd mkdir -p /mnt/ext3`
- `FBK: ucmd mount /dev/mmcbk0p2 /mnt/ext3`
- `FBK: acmd tar -jxv -C /mnt/ext3`
- `FBK: ucp rootfs.tar.bz2 t:-`
- `FBK: Sync`
- `FBK: ucmd umount /mnt/ext3`
- `FBK: DONE`

UUU Resource

- Source code

- github.com/codeauroraforum/mfgtools

- Prebuild binary

- <https://github.com/codeauroraforum/mfgtools/releases>

Running Environment

- **Windows 10 64bit**
 - If use windows 7, need additional update to support WCID auto install winusb driver
- **Linux**
 - Above Ubuntu 16.04

Known issues

- Linux: Open device open failure
 - Please add **sudo**

Platform

- Supported
 - All i.MX6 and i.MX 7D, i.MX 7ULP
 - i.MX 8QuadMax, i.MX 8QuadXPlus
 - i.MX 8M, i.MX 8MMini, i.MX 8MNano

Technology Detail

- Only use standard C++ library, zlib, libusb
- Uboot and kernel use WCID to auto install winusb driver in windows platform
 - About WCID: <https://github.com/pbatard/libwidi/wiki/WCID-Devices>
- Using Libusb as low level usb transfer
- UI part and core library is split. Customer can customer UI or design a GUI easily
- FB/FBK use android fastboot protocol



**SECURE CONNECTIONS
FOR A SMARTER WORLD**