

# i.MX 8 Universal Update Utility Tool

Bryan Thomas

Field Applications Engineer

---

October 2019 | Session #AMF-AUT-T3893



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# Agenda

---

- UUU Tool Introduction
- Install
- Usage



# UUU

- Tool to flash storage media on i.MX products
- UUU replaces the MFG Tool
- Less work to create flashing scripts
- Less host and target dependencies
- More host OS supported
- Source code available on Github

# System Requirements

- Windows 10 64bit
- Linux
  - Above Ubuntu 16.04
- MacOS
  - In development

# Known Issues

- **Windows:** Open device failure
  - libusb bug, need to retry
- **Linux:** Open device open failure
  - Use **sudo** or add user permissions for device
- **Issue reporting:** FAE or through github
  - <https://github.com/NXPmicro/mfgtools/issues>

# Supported SOC Platforms

- Currently Supported
  - MX8QXP
  - MX8QM
  - MX8M
  - All MX6 and MX7D, ULP
  - i.MX28

# Technology Detail

- Standard C++ library, zlib, libusb
- U-boot 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 mechanism
- UI part and core library are separate. Customers can design a GUI easily
- FB/FBK use android fastboot protocol

# OS Support for UUU Compared with MFG Tools

UUU



Mac OS

(Not test yet)

Mfgtools





# UUU Compared with MFG Tools

## UUU

```
2:1 1/ 1 [HID Write failure] 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

^Ccuess 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

^Ccuess 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 Only.

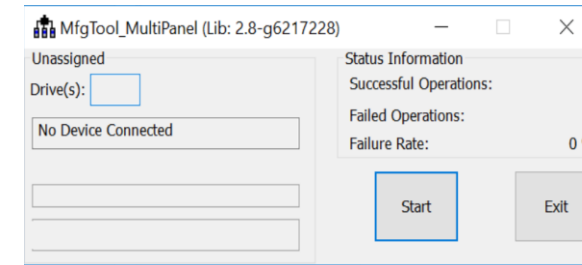
Multi Device Supports

Not limited by uuu.

Only limited by number of ports

Simple pure text script

## Mfgtools



GUI

Supports max of 4 devices

XML File

# How uuu Helps Development

## UUU

1. [sudo] uuu -d flash.bin
1. modify u-boot code
2. Build u-boot & make flash
3. Press reset button

## SD Card

1. Change uboot code
2. Build uboot & make flash
3. Remove card from board
4. dd flash.bin to sd card
5. Plug sd card
6. Reboot board

# Downloading UUU and UUU Resources

- Source code:
  - <https://github.com/codeauroraforum/mfgtools>
- Prebuilt Binaries
  - <https://github.com/codeauroraforum/mfgtools/releases>
- WIKI
  - <https://github.com/NXPmicro/mfgtools/wiki>
- Sample Scripts
  - <https://github.com/NXPmicro/mfgtools/wiki/Sample-script>

# Building from Source

## Linux

- git clone <https://github.com/NXPmicro/mfgtools.git>
- cd mfgtools
- sudo apt-get install libusb-1.0-0-dev libzip-dev libbz2-dev pkg-config cmake
- cmake .
- make
- copy to your local bin! ( cp uuu/uuu ~/bin/; chmod a+x uuu)

# Building from Source

## Windows

- git clone <https://github.com/NXPmicro/mfgtools.git>
- cd mfgtools
- git submodule init
- git submodule update
- open msvs/uuu.sln with Visual Studio 2017

# UUU udev support

## Linux

- run `$ uuu -udev`
- This will show a command that you can run to add a set of rules to your udev directory. Then restart udev!

# Quick User Guide

- **Run u-boot only**
  - `uuu flash.bin` (for i.MX8)
  - `uuu u-boot.imx` (for i.MX6,i.MX7, and ULP)
- **Run QNX SPL only**
  - `uuu -f ipl-imx8qm-cpu-mek_b0.imx` (for i.mx8)
- **Burn Images into emmc**
  - `uuu android.zip` (Android release package)
  - `uuu L4.14.98_2.0.0_ga_images_MX8QXPMEK.zip` (Linux release package)
- **For other storage you can write a script.**
  - Examples - <https://github.com/NXPmicro/mfgtools/wiki/Sample-script>

# How to Flash Multiple Boards

- **If boards are all the same**
  - `uuu -d flash.bin.` (uuu will monitor all ports for all known type boards)
- **If boards are different**
  - `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 communicate with u-boot**
  - Default u-boot will auto run fastboot cmd if booting 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 with uuu

- `uuu uuu.lst`
- **Script is text file**
  - First line must be `uuu_version` of your release or later
  - Add any commands after first required line
  - Done command is required at end of script. Which tells uuu to exit in normal mode or increase success number in daemon mode
- **`uuu -s` enters shell mode**
  - You can input commands and run them at a shell prompt
  - `uuu.inputlog` will record all input commands
  - You can copy it another file
  - Next time you use `uuu <your new file>` to run all commands

## How to Download Kernel and DTB Files

- FB: ucmd setenv fastboot\_buffer \${loadaddr}
- FB: download -f Image
- FB: ucmd setenv fastboot\_buffer \${fdt\_addr}
- FB: download -f imx8qxp\_mek.dtb
- FB: acmd booti \${loadaddr} - \${fdt\_addr}

# How to Flash an 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 to Flash an Image with uuu Autoscripts

- Just run below command
- **uuu imx8qxp\_mek\_android.zip**
  - uuu will search auto.uuu in root directory of imx8qxp\_mek\_android.zip and run it.

# How to Flash a 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-imx8qxpmeek.rootfs.sdcard
- FB: flash bootloader flash.bin
- FB: ucmd mmc partconf 0 1 1 0
- FB: Done

# How to Flash a Yocto Image to eMMC

- **uuu fsl-image-validation-imx-imx8qxpmeek-20180516162233.rootfs.sdcard**
- **uuu <release package>**
- **Note. This have not deployed in current release, Maybe change method in finial GA.**
- Need put a uuu.auto in fat parttion
  - 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

# Replicating the MFG Tools Default Script

uuu\_version 1.0.1

SDPS: boot -f flash\_mfg.bin

FBK: ucp mksdcard.sh t:/tmp

FBK: ucmd chmod 777 /tmp/mksdcard.sh

FBK: ucmd /tmp/mksdcard.sh /dev/mmcblk0

FBK: ucmd dd if=/dev/zero of=/dev/mmcblk0 bs=1k  
seek=4096 count=1

FBK: ucmd sync

FBK: ucmd echo 0 > /sys/block/mmcblk0boot0/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/mmcblk0boot0/force\_ro

FBK: ucmd while [ ! -e /dev/mmcblk0p1 ]; do sleep 1; done

FBK: ucmd mkfs.vfat /dev/mmcblk0p1

FBK: ucmd mkdir -p /mnt/mmcblk0p1

FBK: ucmd mount -t vfat /dev/mmcblk0p1 /mnt/mmcblk0p1

FBK: ucp Image t:/mnt/mmcblk0p1

FBK: ucp fsl-imx8qxp-mek.dtb t:/mnt/mmcblk0p1

FBK: ucmd umount /mnt/mmcblk0p1

FBK: ucmd mkfs.ext3 -F -E nodiscard /dev/mmcblk0p2

FBK: ucmd mkdir -p /mnt/ext3

FBK: ucmd mount /dev/mmcblk0p2 /mnt/ext3

FBK: acmd tar -jxv -C /mnt/ext3

FBK: ucp rootfs.tar.bz2 t:-

FBK: Sync

FBK: ucmd umount /mnt/ext3

FBK: DONE





**SECURE CONNECTIONS  
FOR A SMARTER WORLD**