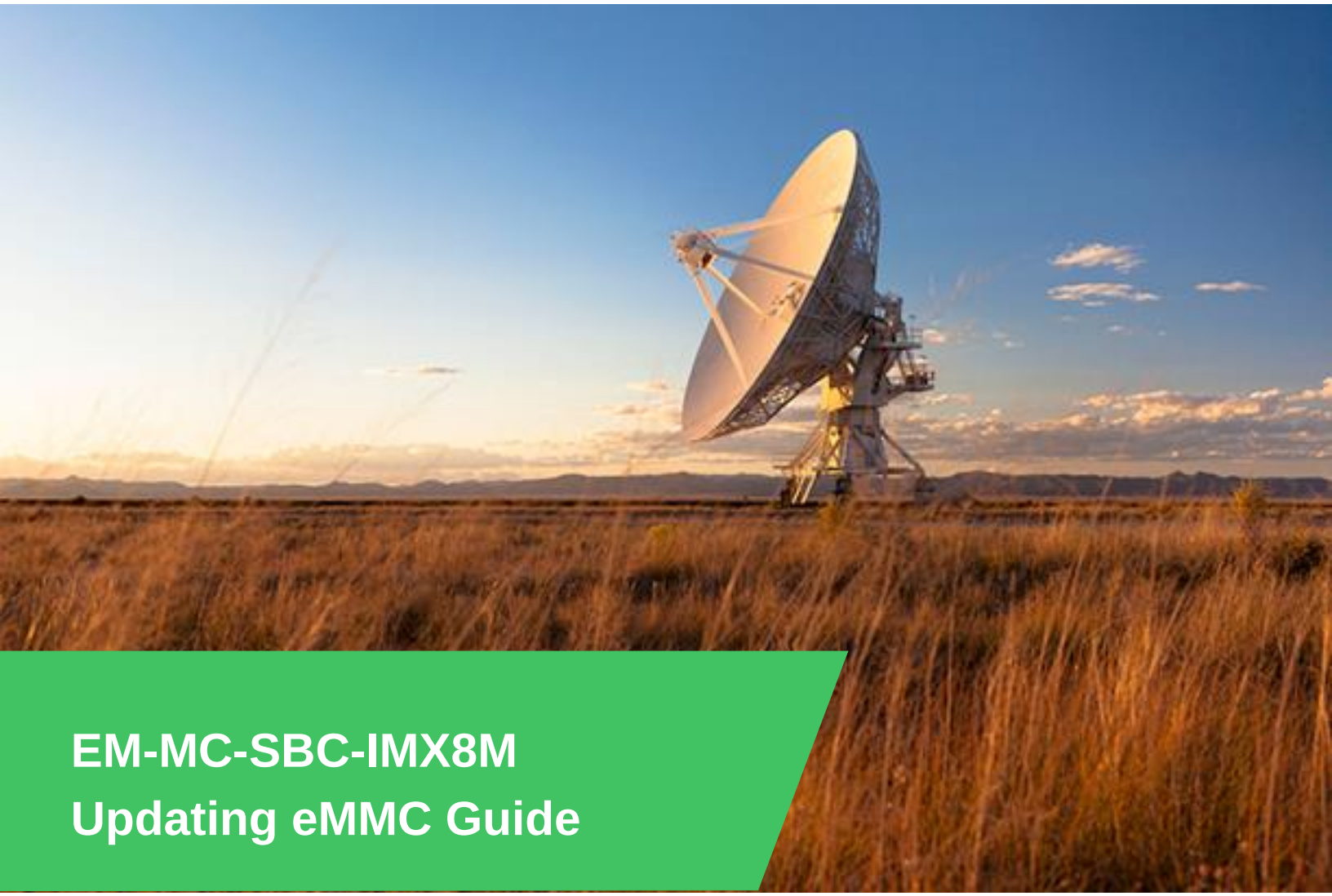


EM-MC-SBC-IMX8M
Updating eMMC Guide

Directory

Directory 2
1 Introduction 3
1.1 Running Environment 3
2 Burning Linux Image 4
2.1 Preparation 4
2.2 Download Script 4
2.3 Burn the Image 4
3 Burning Android Image 7
3.1 Preparation 7
3.1.1 Burn Entire System Image 7
3.1.2 Burn Android Compile Output 7
3.2 Download Script 7
3.3 Burn the Image 9
4 Announcement 11
5 Appendix 11
5.1 Hardware 11
6 Revision History 12

1 Introduction

The default version of EM-MC-SBC-IMX8M supported SD Card. Avnet also provide eMMC version for users to customize. This document will introduce how to burn the system image to the eMMC.

1.1 Running Environment

Burning Tool: Universal Update Utility (Short as UUU)

Download Link: https://github.com/NXPmicro/mfgtools/releases

Software:

- Win10 64 bit OS
- Ubuntu 64 bit OS, 16.14 or higher
- Win7 64bit OS is support, but need to install USB driver according to Win7 User Guide

2 Burning Linux Image

2.1 Preparation

Put the following files and uuu tool into the same directory

- u-boot-imx8m-uuu.imx // The Bootloader to burn the eMMC
- u-boot.imx //The U-boot image file compiled with the system Image
- EM-MC-SBC-IMX8M Linux system image file, e.g.: EM-MC-SBC-IMX8M-Yocto-Image-SDcard-V1.0.2b03.img
- uuu_linux.lst //The script file using in download

2.2 Download Script

uuu_linux.lst file content as below:

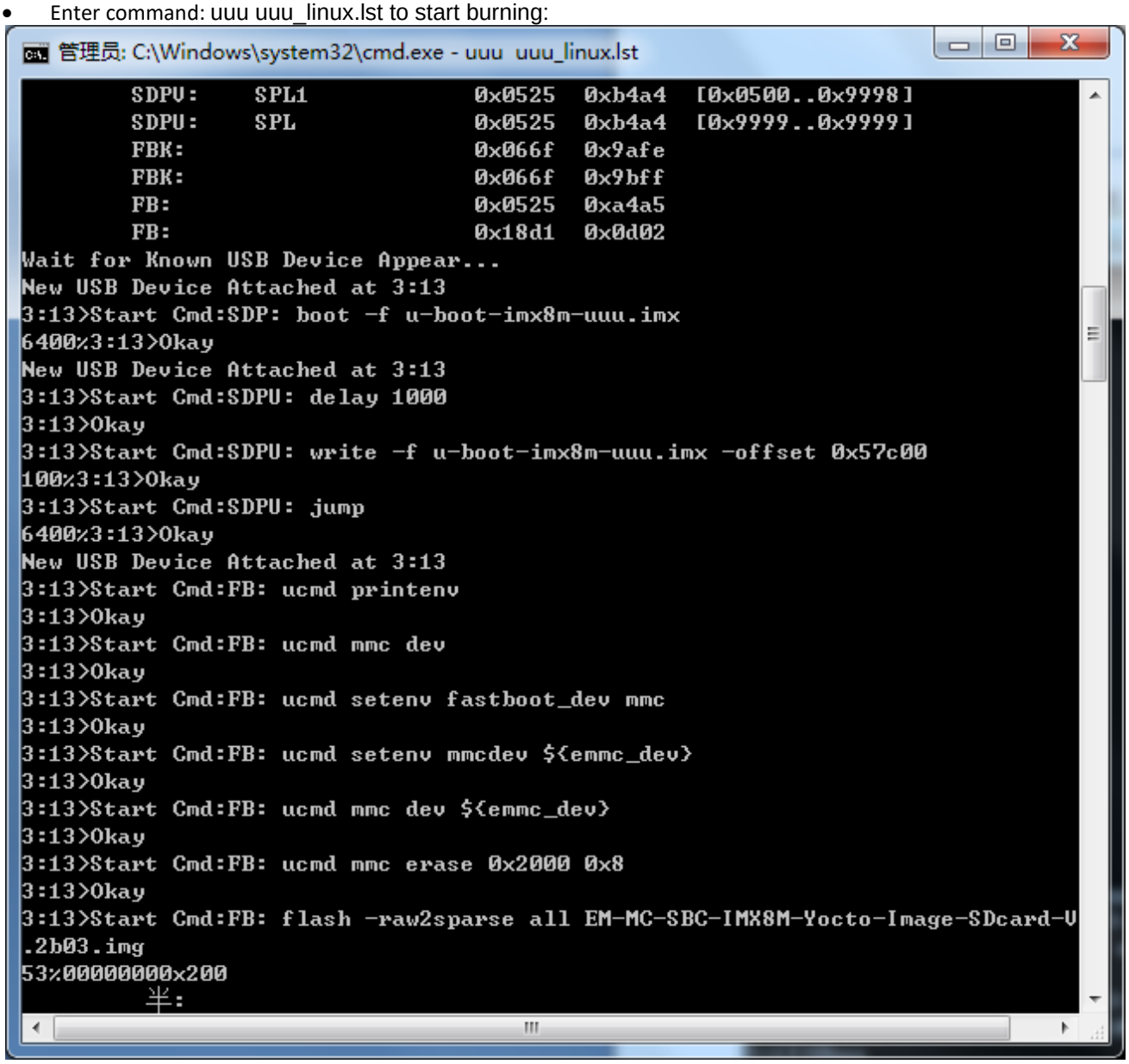
EM-MC-SBC-IMX8M-Yocto-Image-SDcard-V1.0.2b03.img should be replaced by the actual file name:

```
uuu_version 1.2.91

SDP: boot -f u-boot-imx8m-uuu.imx
# This command will be run when use SPL
SDPU: delay 1000
SDPU: write -f u-boot-imx8m-uuu.imx -offset 0x57c00
SDPU: jump
# This command will be run when ROM support stream mode
SDPS: boot -f u-boot.imx

FB: ucmd printenv
FB: ucmd mmc dev
FB: ucmd setenv fastboot_dev mmc
FB: ucmd setenv mmcdev ${emmc_dev}
FB: ucmd mmc dev ${emmc_dev}

# erase environment variables of uboot
FB: ucmd mmc erase 0x2000 0x8

FB: flash -raw2sparse all EM-MC-SBC-IMX8M-Yocto-Image-SDcard-V1.0.2b03.img
FB: flash bootloader u-boot.imx
FB: done
```

2.3 Burn the Image

- Connect USB0 (the lower one in USB 3.0 interface J5) and PC using USB type A cable.
- Connect the debug interface to PC with USB to TTL converter. Pin 6, 8 and 10 of J10 to the GND, RXD and TXD pin of the USB to TTL converter.
- Powered the board with a 5V, 2A, Type-C interface power (to J4).
- Enter the directory of UUU in command line:

- Enter command: uuu uuu_linux.lst to start burning:



Command line will show the rate of process, while the serial terminal will print similar information until the burning finished.

```
........ wrote 16776192 bytes to 'all'
request 00000000becdf000 was not queued to ep1in-bulk
Starting download of 16776244 bytes
request 00000000becdf000 was not queued to ep1in-bulk
.....................................................................
.....................................................request 00000000becdf000
1in-bulk

downloading of 16776244 bytes finished
writing to partition 'all'
sparse flash target is mmc:0
writing to partition 'all' for sparse, buffer size 16776244
Flashing sparse image at offset 0
Flashing Sparse Image
```

- Burning finished. Command line will print:

```
100%3:13>Okay00
3:13>Start Cmd:FB: flash bootloader u-boot.imx
0x400000000x2003:13>Okay
3:13>Start Cmd:FB: ucmd mmc partconf ${emmc_dev} 0 1 0
3:13>Okay
3:13>Start Cmd:FB: done
3:13>Okay
←[?25h



D:\uuu>_
```

- Shut down the power of EM-MC-SBC-IMX8M, disconnect the USB cable, power on the board again, then the board will boot from eMMC.

3 Burning Android Image

3.1 Preparation

3.1.1 Burn Entire System Image

Put the following files and uuu tool into the same directory

- u-boot-imx8m-uuu.imx // The Bootloader to burn the eMMC
- u-boot-imx8mq.imx //The U-boot image file compiled with the system Image
- EM-MC-SBC-IMX8M Andriod system image file, e.g.:
  android_rel_imx8m_emmc_20190510.img
- uuu_android.lst //The script file using in download

3.1.2 Burn Android Compile Output

Put the following files and uuu tool into the same directory

- u-boot-imx8m-uuu.imx // The Bootloader to burn the eMMC
- Compile Output:
  - partition-table-7GB.img*
  - u-boot-imx8mq.imx
  - dtbo-imx8mq.img
  - boot.img
  - vendor.img
  - vbmeta-imx8mq.img
  - system.img
- uuu_android.lst //The script file using in download

Note: When your eMMC storage is 8GB, partition-table should choose partition-table-7GB.img, 16GB eMMC storage should choose partition-table-default.img, 32GB eMMC storage should choose partition-table-28GB.img

3.2 Download Script

1. Download script for entire android system image:

android_rel_imx8m_emmc_20190510.img should be replaced by the actual file name:

```
uuu_version 1.2.91
# uuu scripts for imx8mq Android imx_pi9.0 eMMC
SDP: boot -f u-boot-imx8m-uuu.imx
# This command will be run when use SPL
SDPU: delay 1000
SDPU: write -f u-boot-imx8m-uuu.imx -offset 0x57c00
SDPU: jump
# This command will be run when ROM support stream mode
SDPS: boot -f u-boot-imx8mq.imx
FB: ucmd setenv fastboot_dev mmc
FB: ucmd setenv mmcdev ${emmc_dev}
FB: ucmd mmc dev ${emmc_dev}
# erase environment variables of uboot
```

```
FB: ucmd mmc erase 0x2000 0x8
FB: flash -raw2sparse all android_rel_imx8m_emmc_20190510.img
FB: flash bootloader u-boot-imx8mq.imx
FB: ucmd mmc partconf ${emmc_dev} 0 1 0
FB: done
```

2. Download script for Android compile output:

partition-table should be modified according to the actual storage size:

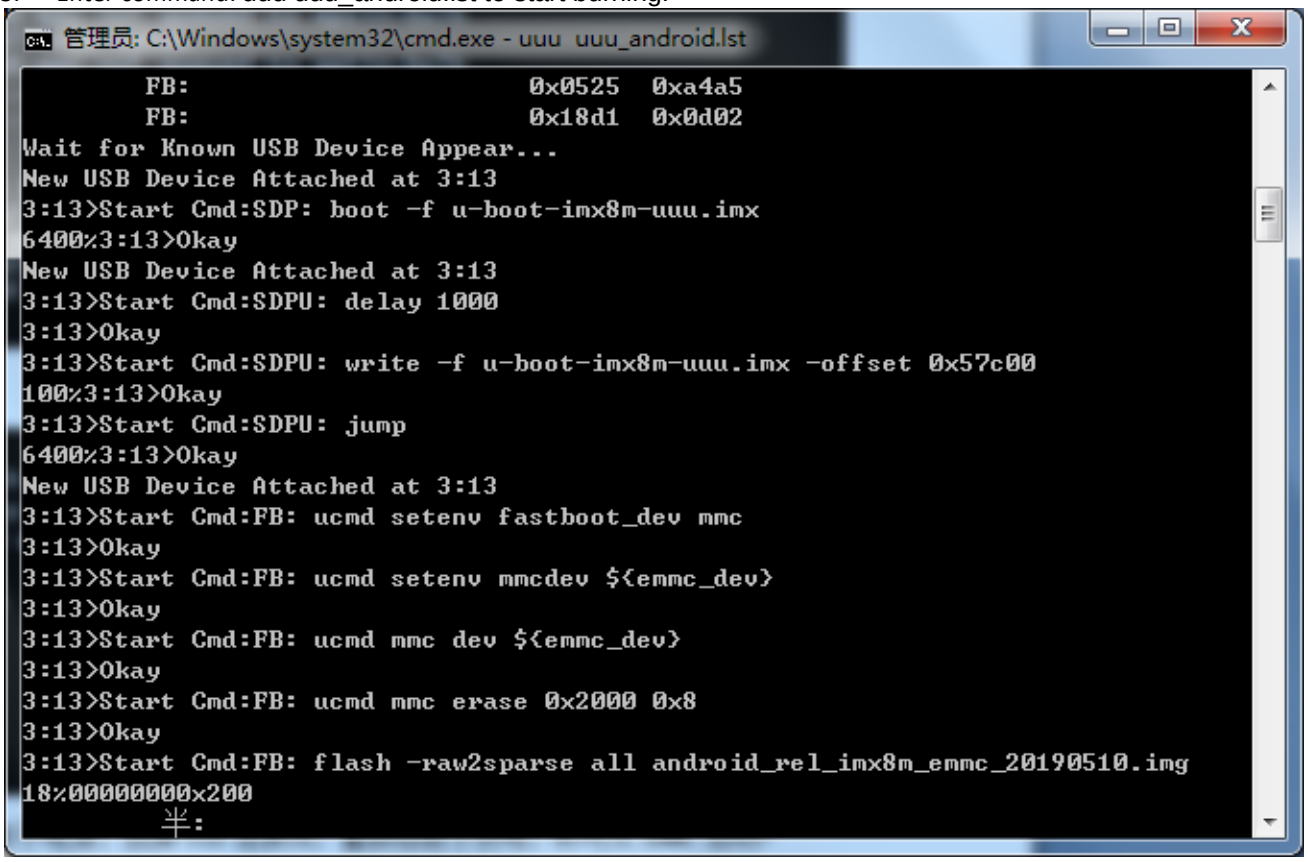
```
uuu_version 1.2.91
# uuu scripts for imx8mq Android imx_pi9.0 eMMC
SDP: boot -f u-boot-imx8m-uuu.imx
# This command will be run when use SPL
SDPU: delay 1000
SDPU: write -f u-boot-imx8m-uuu.imx -offset 0x57c00
SDPU: jump
# This command will be run when ROM support stream mode
SDPS: boot -f u-boot-imx8mq.imx

FB: ucmd setenv fastboot_dev mmc
FB: ucmd setenv mmcdev 0
FB: ucmd mmc dev 0
FB: flash bootloader u-boot-imx8mq.imx
FB[-t 600000]: flash gpt partition-table-7GB.img

# erase environment variables of uboot
FB: ucmd mmc dev 0 0
FB: ucmd mmc erase 0x2000 8

FB: ucmd mmc partconf 0 0 1 0

FB: flash boot_a boot.img
FB: flash boot_b boot.img
FB[-t 100000]: flash system_a system.img
FB[-t 100000]: flash system_b system.img
FB: flash vbmeta_a vbmeta-imx8mq-emmc.img
FB: flash vbmeta_b vbmeta-imx8mq-emmc.img
FB: flash vendor_a vendor.img
FB: flash vendor_b vendor.img
FB: flash dtbo_a dtbo-imx8mq-emmc.img
FB: flash dtbo_b dtbo-imx8mq-emmc.img

# erase userdata and misc partition
FB[-t 600000]: ERASE userdata
FB: ERASE misc
FB[-t 100000]: ERASE presistdata
FB[-t 100000]: ERASE fbmisc
```

```
FB: done
```

3.3 Burn the Image

1. Connect USB0 (the lower one in USB 3.0 interface J5) and PC using USB type A cable.
2. Connect the debug interface to PC with USB to TTL converter. Pin 6, 8 and 10 of J10 to the GND, RXD and TXD pin of the USB to TTL converter.
3. Powered the board with a 5V, 2A, Type-C interface power (to J4).
4. Enter the directory of UUU in command line:

5. Enter command: uuu uuu_android.lst to start burning:



Command line will show the rate of process, while the serial terminal will print similar information until the burning finished.

```
........ wrote 16776192 bytes to 'all'
request 00000000becdf000 was not queued to ep1in-bulk
Starting download of 16776244 bytes
request 00000000becdf000 was not queued to ep1in-bulk
.....................................................................
.....................................................request 00000000becdf000
1in-bulk

downloading of 16776244 bytes finished
writing to partition 'all'
sparse flash target is mmc:0
writing to partition 'all' for sparse, buffer size 16776244
Flashing sparse image at offset 0
Flashing Sparse Image
```

6. Burning finished. Command line will print:

```
3:13>Start Cmd:FB: flash -raw2sparse all android_rel_imx8m_emmc_20190510.img
100%3:13>Okay00
3:13>Start Cmd:FB: flash bootloader u-boot-imx8mq.imx
0x400000000x2003:13>Okay
3:13>Start Cmd:FB: ucmd mmc partconf ${emmc_dev} 0 1 0
3:13>Okay
3:13>Start Cmd:FB: done
3:13>Okay
←[?25h



D:\uuu>
```

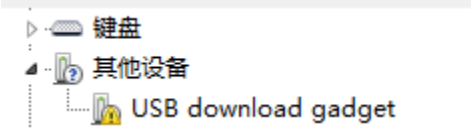
7. Shut down the power of EM-MC-SBC-IMX8M, disconnect the USB cable, power on the board again, then the board will boot from eMMC.

4 Announcement

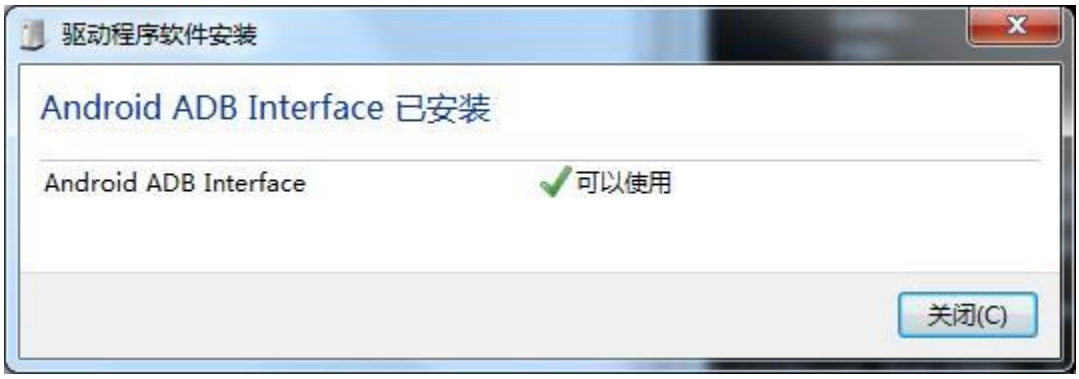
The first attempt to burn the board, you may get the following note:

```
Wait for Known USB Device Appear...
New USB Device Attached at 3:13
3:13>Start Cmd:SDP: boot -f u-boot-imx8m-uuu.imx
6400%3:13>Okay
New USB Device Attached at 3:13
3:13>Fail Failure open usb device
←[?25h
```

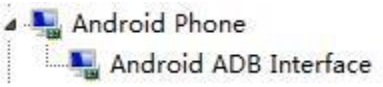
Check the device manager, find that USB download gadget is not installed:



System will install the driver automatically, when install finished, you will get following notification



Device manager will show:



If the auto installation failed, install it by yourself, refer to Win7 User Guide

Execute the Burning command again after the driver is installed correctly.

5 Appendix

5.1 Hardware

For the detail hardware introduction, please refer to EM-MC-SBC-IMX8M Hardware user manual.

6 Revision History

Date	Version	Revision
03 Oct 19	01	Initial Release