

eFuses for i.MX6 SOM

Contents

[Setting the eFuses for MicroSOM and Hummingboard](#)

[USB host to host cable preparation](#)

[Run MFGTool to Blow the eFuses](#)

[Understanding the eFuse Settings](#)

[Booting from other sources](#)

[Forcing eMMC reset](#)

.

An updated version of this article can be found here:[Setting the eFuses \(developers\)](#) .

Setting the eFuses for MicroSOM and Hummingboard

If an end-user orders a **SOM** together with a **HummingBoard** carrier board, the shipping department is notified that the boot configuration needs to be set on the **SOM**. A SOM which does not undergo this process (for example, if one is ordered by itself) will need to have an appropriate configuration completed to tell the processor where to look for the start of the boot process.

This tutorial provides the information on blowing the i.MX6 eFuses on the SoC integrated on the **SOM** in order to boot from the Micro-SD cart.

Please note that this operation is irreversible.

USB host to host cable preparation

Use these instructions as a “getting started” step if you do not already have an appropriate USB OTG cable.

1. Get the usb cables

1. Get two USB cables.
2. Cut them in the middle or where ever it's convenient by leaving enough length to the USB host connectors
3. Bridge the black, white and green wires of each cable directly to each other.
4. Bridge the red (vbus) wire with a 10 ohm resistor in the middle. Notice that there were reports that 10 ohm is too high and 1 ohm would do better.
5. USB host to host cable is ready.

Notice – the reason for the 10 ohm resistor is to protect the CuBox-i/HummingBoard power supply or PC power supply from shortening each other since in some point (when u-boot is loaded) then the vbus of both ports is active (i.e. drive 5V).

Run MFGTool to Blow the eFuses

Some key notes to keep in mind:

- This is an irreversible process. The eFuses in the i.MX6 System on Chip application processors are set by running an electrical signal through them. This process can happen exactly once. By sending the signal through these paths, they are “fused” to a specific setting.
 - This is a process for advanced users and is not typically supported by Solid Run personnel. For those users ordering standalone SOM devices, this process documentation has been provided as an informational starting point.
 - If you use a USB OTG cable without a custom resistor, you are operating without a safety. Initially, both the Hummingboard USB and the host USB of your Windows PC will be carrying an electrical charge. If you are plugged into the wrong port, or are otherwise connected with this dual-power mode for too long, you may damage your device.
1. The MFGTool provided by Freescale requires a Windows based PC, preferably with an intel based chipset. (Freescale's detection routines in the MFGTool can be picky when it comes to chipsets.)
 2. Download the SolidRun-MicroSOM-Fusing.zip archive and unzip it to a known location. (Download from [IMX6 Som Documents](#) page)
 3. Choose the right directory according to the i.MX6 device (either solo/dual lite OR dual/quad).
 4. Double click on the mfgtool2.exe
 5. Connect one side of the USB host to host cable to your PC and other to the upper USB port of CuBox-i or HummingBoard carrier with the intended SOM to configure connected.
 6. Power up the CuBox-i / HummingBoard
 7. On the Windows mfgtool2.exe you would notice a second slider that indicates that u-boot is being downloaded

8. Done. If you are using CuBox-i as a carrier for blowing fuses then the front LED will blink 3 times, the number of fuses that were blown.

Understanding the eFuse Settings

There are three fuses that SolidRun blows.

1. First at address 0x23. We set the value 0xD063. This is high 16 bits of the MAC address.
2. Second at address 0x5, the value 0x2840. This indicates to boot from SD1 (the micro SD port).
3. Third at address 0x6, the value 0x10. This indicates to the i.MX6 internal bootrom to take the boot vector from the fuses instead of OTG signals.

Note – The fuse that contains the lower 32 bits of the MAC address is programmed only after passing all the manufacturing tests.

Booting from other sources

i.MX 6Dual/6Quad Applications Processor Reference Manual by NXP: [Available Here](#)

In the previous section the value 0x2840 was written to address 0x5; which states booting from SD1.

Looking at the i.MX6 solo/dual lite or i.MX6 dual/code reference manual, chapter 5 (Fusemap) Table 5-7 (SD/eSD Boot Fusemap) explains what this boot source means.

When writing the value 0x2840 then BOOT_CFG1=0x40 and BOOT_CFG2=0x28.

BOOT_CFG1[7:3] decides the boot type (Table 5-1) and BOOT_CFG1[2:0] and BOOT_CFG2[7:0] describes the details on that boot.

In this case BOOT_CFG1[7:3] = 1b01000, which means boot from SD/eSD (~~eMMC~~)

BOOT_CFG1[2:0] doesn't set anything

BOOT_CFG2[7:0] says no SD calibration steps, 4bit bus width and use USDHC2 controller (i.e. SD1 when counting from 0) to boot.

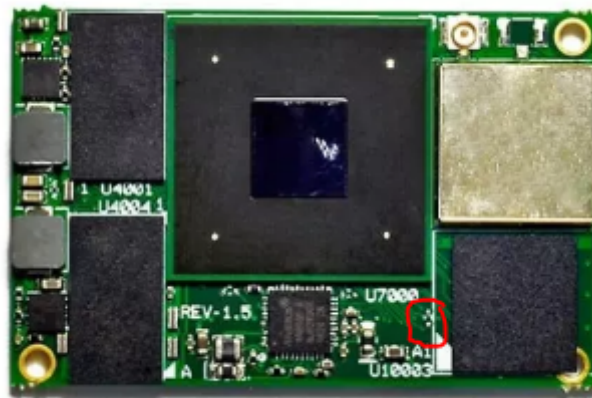
In order to boot from SATA port 0x2840 becomes 0x0020. Look for BOOT_CFG2 in this case (i.e. table 5-5) if more bits need to be set.

Notes:

1. Setting a bit in the fusemap will physically modify the processor and this can't be modified.
2. Look at BOOT_CFG3 and BOOT_CFG4 where the user can deploy a SPI flash as secondary boot in case the primary boot (i.e. what is set in BOOT_CFG1) fails.

Forcing eMMC reset

In case you use i.MX6 uSOM rev 1.5 with eMMC and you set fuses to boot from eMMC and want to fallback to USB OTG; there is an option (in development mode) to keep eMMC in reset by putting a tweaser on this –



That's a 0402 resistor that is not assembled that when the upper pad is short to the lower pad which is the GND signal then it keeps eMMC in reset.

Updated on December 16, 2018

Tagged: [efuses](#) [faq](#) [fuses](#) [imx6](#) [otp](#) [software](#)