SolidRun

eFuses for i.MX6 SOM (Developers page)

Contents

The intention of this page is to provide developers the utilities to boot SolidRun based i.MX6 boards (by default HummingBoard2) that can boot from Micro SD, eMMC and M.2 SSD.

The difference between this page and the older page Setting the eFuses is that the later demonstrates how to blow fuses of i.MX6 to boot in a production line, rather a developer friendly environment.

In this tutorial you will learn how to boot a standalone minimal kernel with buildroot based initial ramdisk that is completely booted off the USB OTG port and boots into a prompt.

Once the prompt is there you can blow fuses, download files (wget) and perform other tasks for the sake of evaluating the boot process.

Setting up the environment

1. HummingBoard Gate/Edge
2. PC that runs Linux
3. Terminal emulation (putty, minicom etc..) that are connected to HummingBoard2 serial terminal 115200bps 8N1 (J25 pin header where pin 1-gnd,2-i.MX6 TX, 3-i.MX6 RX) – newer PCB layout: J2 pins (6,8,10)
4. Download and build – imx_usb_loader (credit to BoundaryDevices for the great tool) – https://github.com/boundarydevices/imx_usb_loader

5. RECOMMENDED – For ease of use we already built imx_usb_loader as static binary with libusb-1.0 and all required configuration files. It can be downloaded from Here-Rev1.1
6. USB host to host cable – refer to Setting the eFuses USB host to host cable preparation

Booting through USB OTG

As a reminder, a fresh i.MX6 device (unfused) will boot off the USB OTG port. SolidRun USB OTG port is actually a USB type A host connector but carries the same USB OTG signals. (please make sure to have the latest u-boot-tools installed)

1. Connect HummingBoard2 USB OTG Host port to PC host and power it up.
2. While running 'lsusb' on the Linux PC you should be seeing either one of the following IDs where the first is for the dual/quad and the second is for the solo / dual-lite devices –

```
Bus 002 Device 047: ID 15a2:0054 Freescale Semiconductor, Inc. i.MX6Q
SystemOnChip in RecoveryMode
Bus 002 Device 059: ID 15a2:0061 Freescale Semiconductor, Inc.
```

3. Open up the serial console, as a reminder this is a null modem (i.e. hw/sw flow controls are disabled)
4. Run './runme.sh' which will mkimage the boot.txt to boot.scr (boot scriptr) and then run 'sudo ./imx_usb -c .' which will transfer u-boot.imx DCD (DDR intialization), zImage to 0x10a00000, device tree to 0x18000000 and a boot.scr to 0x17f00000. Once u-boot.imx runs it will source boot.scr
5. You should see on the terminal u-boot prompt with it's boot count down. Press any key to stop it.
6. You can now use 'fuse' utility to modify fuses, or you can boot Linux and modify fuses there. To boot Linux run –

```
'setenv bootargs console=ttymxc0,115200; bootz 0x10a00000 - 0x18000000'
```

7. The root filesytstem is embedded in the kernel image (initramfs) and root password is –

```
123456
```

Blowing fuses to program the unit MAC address

Blowing fuses is an irreversible act. If you set a bit from '0' to '1' you can not set it back to '0'.

echo <high 16 bit of the MAC address> > /sys/fsl_otp/HW_OCOTP_MAC1 echo <lower 32bit of the

MAC address> > /sys/fsl_otp/HW_OCOTP_MAC0 **For example –**

```
echo 0xd063 > /sys/fsl_otp/HW_OCOTP_MAC1
echo 0x12345678 > /sys/fsl_otp/HW_OCOTP_MAC0
```

In order to blow high 16 bit of MAC address in u-boot; run –

```
fuse prog -y 4 3 0xd063
```

and for the low 32bit run –

```
fuse prog -y 4 2 0x12345678
```

In order to read the low 32bit; run –

```
fuse read 4 2
```

and for the high 16bit run –

```
fuse read 4 3
```

Blowing fuses to boot from Micro SD

For this we blow two fuse sets that the first marks the boot device as SD2 by setting the value 0x2840 (micro SD in HummingBoard), and the second set (value 0x10) instructs i.MX6 to ignore the boot from GPIO and use the eFuses for the boot device settings.
Under Linux –

```
echo 0x2840 > /sys/fsl_otp/HW_OCOTP_CFG4
echo 0x10 > /sys/fsl_otp/HW_OCOTP_CFG5
```

Under U-Boot –

```
fuse prog -y 0 5 0x2840
fuse prog -y 0 6 0x10
```

Blowing fuses to boot from SATA (m.2 on HummingBoard, mSata on HummingBoard-1 and eSata on CuBox-i

For this we blow the boot device, then tell i.MX6 that next boot it should use the selected boot device in the efuses –

```
echo 0x0020 > /sys/fsl_otp/HW_OCOTP_CFG4
echo 0x10 > /sys/fsl_otp/HW_OCOTP_CFG5
```

Blowing fuses to from eMMC (HummingBoard2 eMMC or MicroSOM rev 1.5 on-SOM eMMC)

For this we blow the boot device, then tell i.MX6 that next boot it should use the selected boot device in the efuses.

> **Booting from eMMC is trickier than MicroSD or SATA since the device is soldered on the board and can't be modified by simply taking it out and flashing it on a PC. So when flashing u-boot to the eMMC device, make sure you use a u-boot image that you can get into it's console in case you need to modify it afterwards. MicroSOM rev 1.5 has a 0402 resistor pads that can be short to keep the eMMC device in reset. Refer to** Setting the eFuses **Forcing eMMC reset. If you are using the older version of the SOM (rev 1.3) that doesn't have this force eMMC then the only option is to swap the SOM with another one that is set to boot from Micro SD, boot it up, erase the eMMC and flash the eMMC with a working u-boot/OS etc…**

Under Linux –

```
echo 0x1060 > /sys/fsl_otp/HW_OCOTP_CFG4
echo 0x10 > /sys/fsl_otp/HW_OCOTP_CFG5
```

Under U-Boot –

```
fuse prog -y 0 5 0x1060
fuse prog -y 0 6 0x10
```

Notice that the above settings is for booting with a single bit data width eMMC; which is not critical performance wise since SPL and u-boot are small and once they are up and running eMMC is set to full width (8 bits). If you insist on booting SPL/u-boot in 8 bit then the above value 0x1060 can be changed to 0x5060. This also provides some flexibility to carrier board designers in case they don't want to use all 8 bits of eMMC for the sake of using the the unused to other functions.

As an example of flashing u-boot on eMMC –

```
wget <URL>/SPL
wget <URL>/u-boot.img
dd if=SPL of=/dev/mmcblk2 bs=1K seek=1
dd if=u-boot.img of=/dev/mmcblk2 bs=1K seek=42
```

When finished and want to restart the system, you need to either power cycle the unit or click the reset button. Running the 'reboot' command will not work since new fuses values will not be activated.

> When finished and want to restart the system, you need to either power cycle the unit or click the reset button. Running the 'reboot' command will not work since new fuses values will not be activated.

Updated on December 16, 2018

Tagged: boot efuses faq fuses imx6 otp