# SMA-IMX6

**Freescale i.MX 6 ARM Cortex-A9 RISC Module**

# User's Manual

**1st Ed – 17 October 2014**

## FCC Statement

THIS DEVICE COMPLIES WITH PART 15 FCC RULES. OPERATION IS SUBJECT TO THE FOLLOWING TWO CONDITIONS:

(1) THIS DEVICE MAY NOT CAUSE HARMFUL INTERFERENCE.

(2) THIS DEVICE MUST ACCEPT ANY INTERFERENCE RECEIVED INCLUDING INTERFERENCE THAT MAY CAUSE UNDESIRED OPERATION.

THIS EQUIPMENT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS "A" DIGITAL DEVICE, PURSUANT TO PART 15 OF THE FCC RULES.

THESE LIMITS ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST HARMFUL INTERFERENCE WHEN THE EQUIPMENT IS OPERATED IN A COMMERCIAL ENVIRONMENT. THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND, IF NOT INSTATLLED AND USED IN ACCORDANCE WITH THE INSTRUCTION MANUAL, MAY CAUSE HARMFUL INTERFERENCE TO RADIO COMMUNICATIONS.

OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE HARMFUL INTERFERENCE IN WHICH CASE THE USER WILL BE REQUIRED TO CORRECT THE INTERFERENCE AT HIS OWN EXPENSE.

## A Message to the Customer

### Customer Services

Each and every product is built to the most exacting specifications to ensure reliable performance in the harsh and demanding conditions typical of industrial environments. Whether your new device is destined for the laboratory or the factory floor, you can be assured that your product will provide the reliability and ease of operation.

Your satisfaction is our primary concern. Here is a guide to our customer services. To ensure you get the full benefit of our services, please follow the instructions below carefully.

### Technical Support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For the most frequently asked questions, you can easily find answers in your product documentation. These answers are normally a lot more detailed than the ones we can give over the phone. So please consult the user's manual first.

# Content

# 1. Getting Started

## 1.1 Safety Precautions

**Warning!**

Always completely disconnect the power cord from your chassis whenever you work with the hardware. Do not make connections while the power is on. Sensitive electronic components can be damaged by sudden power surges. Only experienced electronics personnel should open the PC chassis.

**Caution!**

Always ground yourself to remove any static charge before touching the CPU card. Modern electronic devices are very sensitive to static electric charges. As a safety precaution, use a grounding wrist strap at all times. Place all electronic components in a static-dissipative surface or static-shielded bag when they are not in the chassis.

**Always note** that improper disassembling action could cause damage to the motherboard. We suggest not removing the heatsink without correct instructions in any circumstance. If you really have to do this, please contact us for further support.

## 1.2 Packing List

Before you begin installing your single board, please make sure that the following materials have been shipped:

- 1 x SMA-IMX6 Risc Module
- 1 x Quick Installation Guide for SMA-IMX6

If any of the above items is damaged or missing, contact your retailer.

## 1.3 Document Amendment History

| Revision | Date | By | Comment |
|---|---|---|---|
| 1st | April 2014 | | Initial Release |

## 1.4 Manual Objectives

This manual describes in details SMA-IMX6 Single Board.

We have tried to include as much information as possible but we have not duplicated information that is provided in the standard IBM Technical References, unless it proved to be necessary to aid in the understanding of this board.

We strongly recommend that you study this manual carefully before attempting to set up SMA-IMX6 series or change the standard configurations. Whilst all the necessary information is available in this manual we would recommend that unless you are confident, you contact your supplier for guidance.

Please be aware that it is possible to create configurations within the CMOS RAM that make booting impossible. If this should happen, clear the CMOS settings, (see the description of the Jumper Settings for details).

If you have any suggestions or find any errors regarding this manual and want to inform us of these, please contact our Customer Service department with the relevant details.

# 1.5 System Specifications

| System | |
|---|---|
| **Processor** | Freescale i.MX 6<br>Quad Core ARM Cortex-A9 Up to 1.2GHz |
| **Graphics** | Dual Display<br>HD 1080p Encode and Decode<br>2D and 3D Acceleration |
| **Memory** | DDR3L 512MB ~ 2GB |
| **Flash** | 4GB (Up to 64GB) eMMC On-module |
| **Ethernet** | 10/100/1000 Mbit/sec |
| **USB** | 2 x USB 2.0 Port (One OTG) |
| **Display** | Parallel LCD 18/24-bit<br>LVDS Single Channel 18/24-bit<br>HDMI |
| **Image Capture Interfaces** | 2 Interfaces (PCAM, CSI) |
| **Serial** | 2 x RX/TX (Ser1/3); 2 x UART (Ser0/2) |
| **Additional Interfaces** | Up to 3 PCIex1, MLB150, 12 x GPIOs, SDIO,<br>SATA eMMC, 2 x SPI, 5 x I2C, 1 x I2S, SPDIF, WDT,<br>2 x CAN, JTAG |
| **FRAM** | 128Kb I2C Interface FRAM (Optional) |
| **Mechanical & Environmental** | |
| **SW Support** | Android<br>Linux |
| **Thermal** | Industrial Temperature: -40°C to 85°C (800MHz versions) |
| **Power Supply** | 3V to 5.25V - Operates Directly from Single Level<br>Lithium Ion Cells, or Fixed 3.3V or 5V Power Supplies |
| **Compliance** | Form Factor: 82mm x 50mm<br>Compliance: SMARC (Smart Mobility Architecture)<br>Specification by SGET |

## 1.6 Architecture Overview—Block Diagram

The following block diagram shows the architecture and main components of SMA-IMX6.
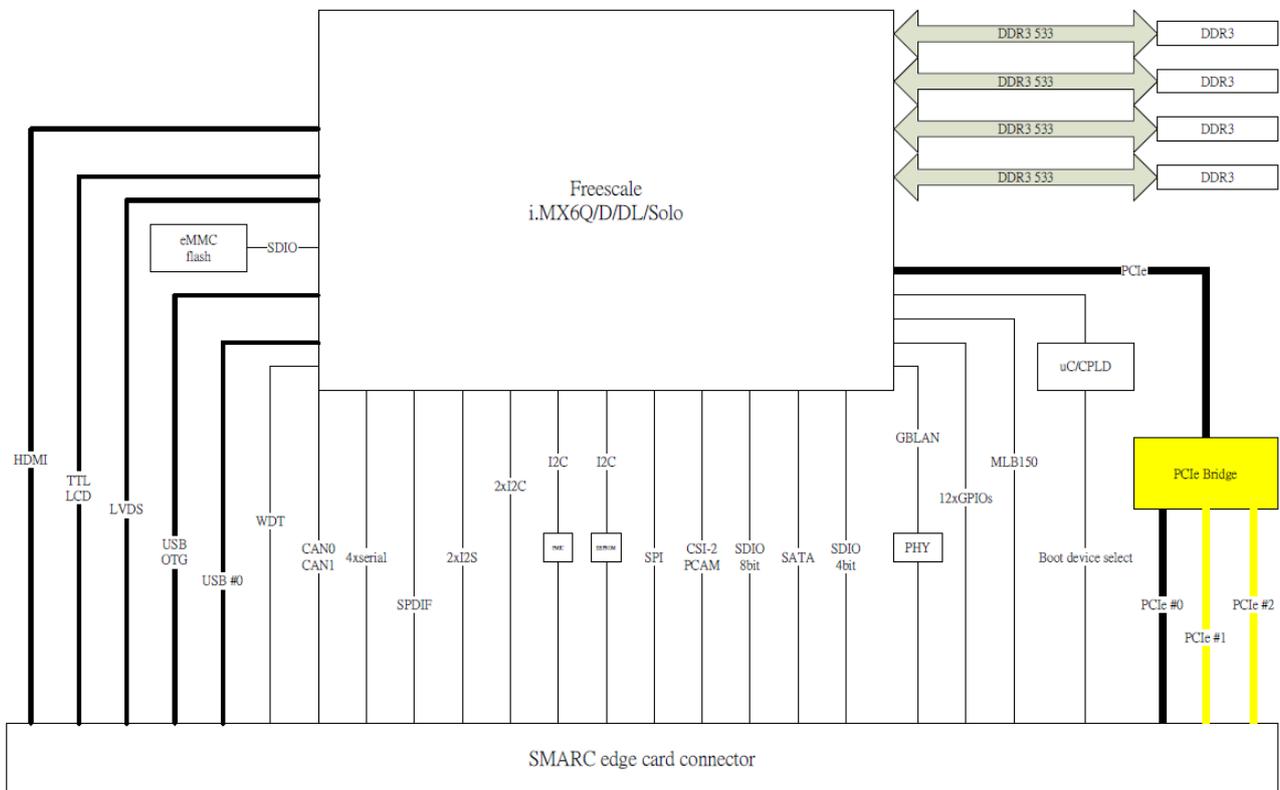
# 2. Hardware Configuration

## 2.1 Product Overview
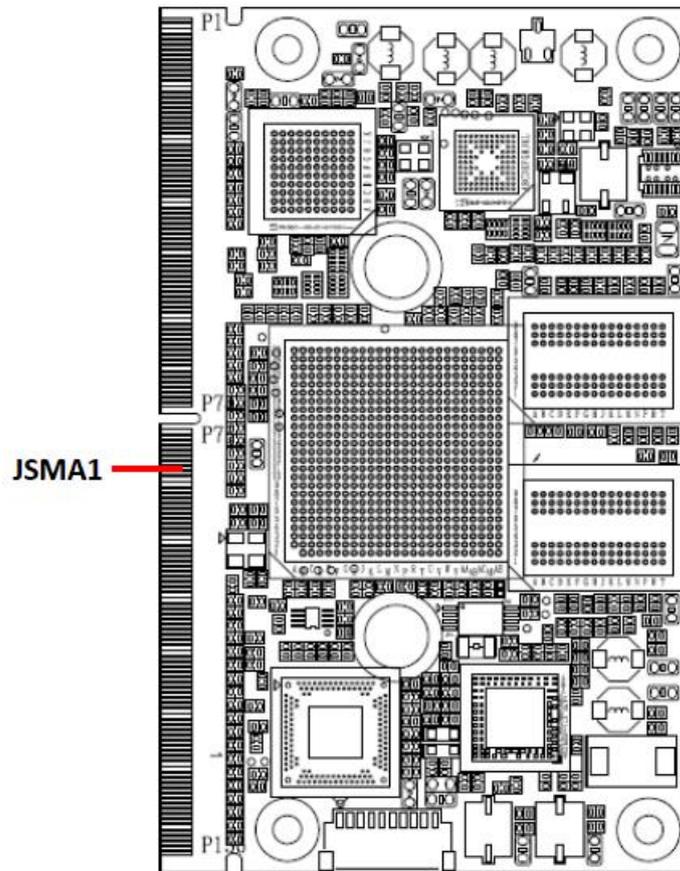


## 2.2 Connector List

The following tables list the function of each of the board's connectors.

### Connectors

| Label | Function | Note |
|-------|----------|------|
| JSMA1 | iMX6 connector | |

## 2.3 Setting Connectors

### 2.3.1   iMX6 connector (JSMA1)
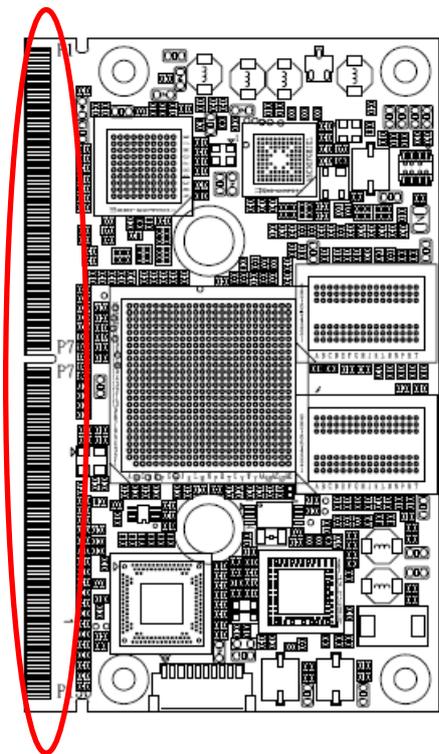


***Default**

| Signal | PIN | PIN | Signal |
|---|---|---|---|
| | | S1 | PCAM_VSYNC |
| PCAM_PXL_CK1 | P1 | S2 | PCAM_HSYNC |
| GND | P2 | S3 | GND |
| PCAM_D0 | P3 | S4 | PCAM_PXL_CK0 |
| PCAM_D1 | P4 | S5 | I2C_CAM_CK |
| PCAM_DE | P5 | S6 | CAM_MCK |
| PCAM_MCK | P6 | S7 | I2C_CAM_DAT |
| PCAM_D2 | P7 | S8 | CSI0_CK+ |
| PCAM_D3 | P8 | S9 | CSI0_CK- |
| GND | P9 | S10 | GND |
| PCAM_D4 | P10 | S11 | CSI0_D0+ |
| PCAM_D5 | P11 | S12 | CSI0_D0- |
| GND | P12 | S13 | GND |
| PCAM_D6 | P13 | S14 | CSI0_D1+ |
| PCAM_D7 | P14 | S15 | CSI0_D1- |
| GND | P15 | S16 | GND |
| PCAM_D8 | P16 | S17 | AFB0_OUT / LED1 |
| PCAM_D9 | P17 | S18 | AFB1_OUT / LED2 |
| GND | P18 | S19 | AFB2_OUT |

| Signal | PIN | PIN | Signal |
|---|---|---|---|
| GBE_MDI3- | P19 | S20 | AFB3_IN / VDD_IN MNTR |
| GBE_MDI3+ | P20 | S21 | AFB4_IN / ADC_IN |
| GBE_LINK100# | P21 | S22 | AFB5_IN |
| GBE_LINK1000# | P22 | S23 | AFB6_PTIO |
| GBE_MDI2- | P23 | S24 | AFB7_PTIO |
| GBE_MDI2+ | P24 | S25 | GND |
| GBE_LINK_ACT# | P25 | S26 | SDMMC_D0 |
| GBE_MDI1- | P26 | S27 | SDMMC_D1 |
| GBE_MDI1+ | P27 | S28 | SDMMC_D2 |
| GBE_CTREF | P28 | S29 | SDMMC_D3 |
| GBE_MDI0- | P29 | S30 | SDMMC_D4 |
| GBE_MDI0+ | P30 | S31 | SDMMC_D5 |
| SPI0_CS1# | P31 | S32 | SDMMC_D6 |
| GND | P32 | S33 | SDMMC_D7 |
| SDIO_WP | P33 | S34 | GND |
| SDIO_CMD | P34 | S35 | SDMMC_CK |
| SDIO_CD# | P35 | S36 | SDMMC_CMD |
| SDIO_CK | P36 | S37 | SDMMC_RST# |
| SDIO_PWR_EN | P37 | S38 | AUDIO_MCK |
| GND | P38 | S39 | I2S0_LRCK |
| SDIO_D0 | P39 | S40 | I2S0_SDOUT |
| SDIO_D1 | P40 | S41 | I2S0_SDIN |
| SDIO_D2 | P41 | S42 | I2S0_CK |
| SDIO_D3 | P42 | S43 | I2S1_LRCK |
| SPI0_CS0# | P43 | S44 | I2S1_SDOUT |
| SPI0_CK | P44 | S45 | I2S1_SDIN |
| SPI0_DIN | P45 | S46 | I2S1_CK |
| SPI0_DO | P46 | S47 | GND |
| GND | P47 | S48 | I2C_GP_CK |
| SATA_TX+ | P48 | S49 | I2C_GP_DAT |
| SATA_TX- | P49 | S50 | I2S2_LRCK |
| GND | P50 | S51 | I2S2_SDOUT |
| SATA_RX+ | P51 | S52 | I2S2_SDIN |

| Signal | PIN | PIN | Signal | Signal | PIN | PIN | Signal |
|---|---|---|---|---|---|---|---|
| SATA_RX- | P52 | S53 | I2S2_CK | PCIE_A_REFCK+ | P83 | S84 | PCIE_B_REFCK+ |
| GND | P53 | S54 | SATA_ACT# | PCIE_A_REFCK- | P84 | S85 | PCIE_B_REFCK- |
| SPI1_CS0# | P54 | S55 | AFB8_PTIO | GND | P85 | S86 | GND |
| SPI1_CS1# | P55 | S56 | AFB9_PTIO | PCIE_A_RX+ | P86 | S87 | PCIE_B_RX+ |
| SPI1_CK | P56 | S57 | PCAM_ON_CSI0# | PCIE_A_RX- | P87 | S88 | PCIE_B_RX- |
| SPI1_DIN | P57 | S58 | PCAM_ON_CSI1# | GND | P88 | S89 | GND |
| SPI1_DO | P58 | S59 | SPDIF_OUT | PCIE_A_TX+ | P89 | S90 | PCIE_B_TX+ |
| GND | P59 | S60 | SPDIF_IN | PCIE_A_TX- | P90 | S91 | PCIE_B_TX- |
| USB0+ | P60 | S61 | GND | GND | P91 | S92 | GND |
| USB0- | P61 | S62 | AFB_DIFF0+ | HDMI_D2+ | P92 | S93 | LCD_D0 |
| USB0_EN_OC# | P62 | S63 | AFB_DIFF0- | HDMI_D2- | P93 | S94 | LCD_D1 |
| USB0_VBUS_DET | P63 | S64 | GND | GND | P94 | S95 | LCD_D2 |
| USB0_OTG_ID | P64 | S65 | AFB_DIFF1+ | HDMI_D1+ | P95 | S96 | LCD_D3 |
| USB1+ | P65 | S66 | AFB_DIFF1- | HDMI_D1- | P96 | S97 | LCD_D4 |
| USB1- | P66 | S67 | GND | GND | P97 | S98 | LCD_D5 |
| USB1_EN_OC# | P67 | S68 | AFB_DIFF2+ | HDMI_D0+ | P98 | S99 | LCD_D6 |
| GND | P68 | S69 | AFB_DIFF2- | HDMI_D0- | P99 | S100 | LCD_D7 |
| USB2+ | P69 | S70 | GND | GND | P100 | S101 | GND |
| USB2- | P70 | S71 | AFB_DIFF3+ | HDMI_CK+ | P101 | S102 | LCD_D8 |
| USB2_EN_OC# | P71 | S72 | AFB_DIFF3- | HDMI_CK- | P102 | S103 | LCD_D9 |
| PCIE_C_PRSNT# | P72 | S73 | GND | GND | P103 | S104 | LCD_D10 |
| PCIE_B_PRSNT# | P73 | S74 | AFB_DIFF4+ | HDMI_HPD | P104 | S105 | LCD_D11 |
| PCIE_A_PRSNT# | P74 | S75 | AFB_DIFF4- | HDMI_CTRL_CK | P105 | S106 | LCD_D12 |
| <Key> | | | <Key> | HDMI_CTRL_DAT | P106 | S107 | LCD_D13 |
| PCIE_A_RST# | P75 | S76 | PCIE_B_RST# | HDMI_CEC | P107 | S108 | LCD_D14 |
| PCIE_C_CKREQ# | P76 | S77 | PCIE_C_RST# | GPIO0 / CAM0_PWR# | P108 | S109 | LCD_D15 |
| PCIE_B_CKREQ# | P77 | S78 | PCIE_C_RX+ | GPIO1 / CAM1_PWR# | P109 | S110 | GND |
| PCIE_A_CKREQ# | P78 | S79 | PCIE_C_RX- | GPIO2 / CAM0_RST# | P110 | S111 | LCD_D16 |
| GND | P79 | S80 | GND | GPIO3 / CAM1_RST# | P111 | S112 | LCD_D17 |
| PCIE_C_REFCK+ | P80 | S81 | PCIE_C_TX+ | GPIO4 / HDA_RST# | P112 | S113 | LCD_D18 |
| PCIE_C_REFCK- | P81 | S82 | PCIE_C_TX- | GPIO5 / PWM_OUT | P113 | S114 | LCD_D19 |
| GND | P82 | S83 | GND | | | | |

| Signal | PIN | PIN | Signal |
|---|---|---|---|
| GPIO6 / TACHIN | P114 | S115 | LCD_D20 |
| GPIO7 / PCAM_FLD | P115 | S116 | LCD_D21 |
| GPIO8 / CAN0_ERR# | P116 | S117 | LCD_D22 |
| GPIO9 / CAN1_ERR# | P117 | S118 | LCD_D23 |
| GPIO10 | P118 | S119 | GND |
| GPIO11 | P119 | S120 | LCD_DE |
| GND | P120 | S121 | LCD_VS |
| I2C_PM_CK | P121 | S122 | LCD_HS |
| I2C_PM_DAT | P122 | S123 | LCD_PCK |
| BOOT_SEL0# | P123 | S124 | GND |
| BOOT_SEL1# | P124 | S125 | LVDS0+ |
| BOOT_SEL2# | P125 | S126 | LVDS0- |
| RESET_OUT# | P126 | S127 | LCD_BKLT_EN |
| RESET_IN# | P127 | S128 | LVDS1+ |
| POWER_BTN# | P128 | S129 | LVDS1- |
| SER0_TX | P129 | S130 | GND |
| SER0_RX | P130 | S131 | LVDS2+ |
| SER0_RTS# | P131 | S132 | LVDS2- |
| SER0_CTS# | P132 | S133 | LCD_VDD_EN |
| GND | P133 | S134 | LVDS_CK+ |
| SER1_TX | P134 | S135 | LVDS_CK- |
| SER1_RX | P135 | S136 | GND |
| SER2_TX | P136 | S137 | LVDS3+ |
| SER2_RX | P137 | S138 | LVDS3- |
| SER2_RTS# | P138 | S139 | I2C_LCD_CK |
| SER2_CTS# | P139 | S140 | I2C_LCD_DAT |
| SER3_TX | P140 | S141 | LCD_BKLT_PWM |
| SER3_RX | P141 | S142 | LCD_DUAL_PCK |
| GND | P142 | S143 | GND |

| Signal | PIN | PIN | Signal |
|---|---|---|---|
| CAN0_TX | P143 | S144 | RSVD / EDP_HPD |
| CAN0_RX | P144 | S145 | WDT_TIME_OUT# |
| CAN1_TX | P145 | S146 | PCIE_WAKE# |
| CAN1_RX | P146 | S147 | VDD_RTC |
| VDD_IN | P147 | S148 | LID# |
| VDD_IN | P148 | S149 | SLEEP# |
| VDD_IN | P149 | S150 | VIN_PWR_BAD# |
| VDD_IN | P150 | S151 | CHARGING# |
| VDD_IN | P151 | S152 | CHARGER_PRSNT# |
| VDD_IN | P152 | S153 | CARRIER_STBY# |
| VDD_IN | P153 | S154 | CARRIER_PWR_ON |
| VDD_IN | P154 | S155 | FORCE_RECOV# |
| VDD_IN | P155 | S156 | BATLOW# |
| VDD_IN | P156 | S157 | TEST# |
| | | S158 | VDD_IO_SEL# |

# 3. Linux User Guide

(Using with carrier board REV-SA01)

## 3.1 Download Source code for building Ubuntu image file

Please make a folder for storing the source code first then typing the command below to get started for the source code download.


$ Sudo apt-get install git

$ git clone guest@202.55.227.57:freescale/core.git -b SMARC

About password, please check with BCM Sales or PM to get it.


## 3.2 Set up a Linux host for building U-boot & Kernel Image

We support to compile u-boot & Kernel on Ubuntu 12.04 (64bit version), other version of Ubuntu is not currently supported and may have built issues.


Install host packages needed by building code. This document assumes you are using Ubuntu. Not a requirement, but the packages may be named differently and the method of installing them may be different.


```
sudo apt-get install ia32-libs
sudo apt-get install uboot-mkimage
```


## 3.3 Building up U-boot & Kernel image

You can follow up the steps below to compile the u-boot & Kernel after downloading the source code.

1.Please move to folder "core" then start to compile both the u-boot & Kernel.


■■~/$ cd core/

```
leo@ubuntu:~/imx6_core$ ls
core
leo@ubuntu:~/imx6_core$ cd core/
leo@ubuntu:~/imx6_core/core$
```


2.Type the command to compile both u-boot & Kernel.

$ make rev-sa01 –j number

(**-j number** means multi jobs for more efficiant building, you can add it according to your CPU performance of PC, e.g. mine is "–j16" as below )

```
leo@ubuntu:~/imx6_core/core$ make rev-sa01 -j16
```

**SMA-IMX6 User's Manual**

3. You can find the u-boot(u-boot.bin) & Kernel(uImage) under folder "core" as below after the compiling is finish.

```
leo@ubuntu:~/imx6_core/core$ ls
fsl  kernel  Makefile  u-boot  u-boot-quad.bin  u-boot-solo.bin  uImage
```

PS: If you would like to use Mfgtool for flashing image file, you must put the file u-boot.bin and uImage under "~\Image\smarc" for right detected path.
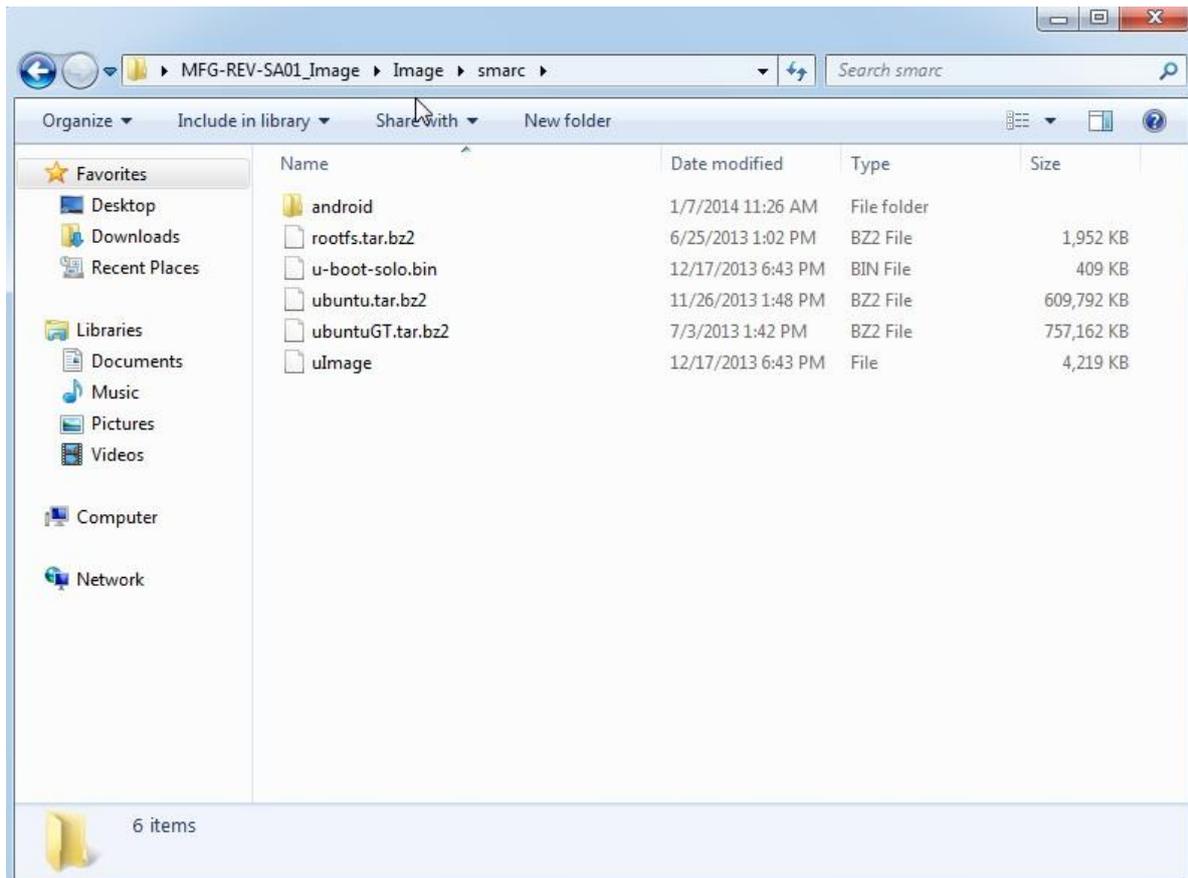
## 3.4 Use MfgTool to flash Ubuntu into onboard eMMC

Manufacturing tool, a successor of ATK, provides a series of new features to power your mass production work. The features like windows style GUI, multiple devices support, explicit status monitoring, versatile functionalities and highly flexible architecture make it a best choice to meet your critical timing, cost and customization requirements.

For using Mfgtool to flash image file into onboard eMMC, please follow up the steps below

1. Please turn on the Pin4 of the DIP switch as below into burning mode of Mfgtool.
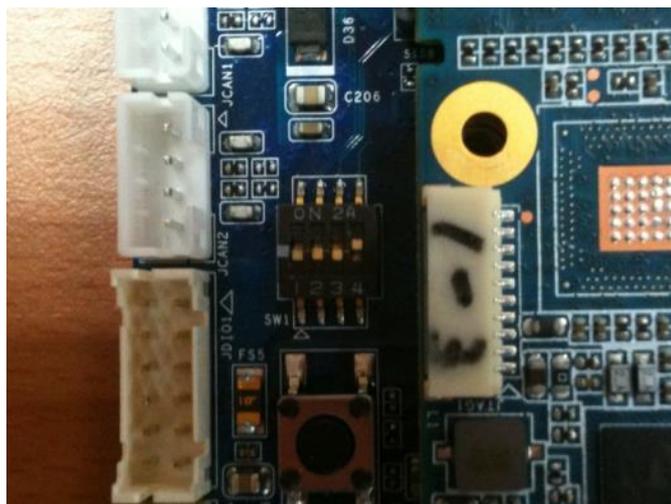


2. Power on the mainboard then plug the cable from OTG socket to PC.
3. Click the folder "**~\MFG-Tools**", e.g. mine is
   D:\ MFG-REV-SA01_Image\**MFG-Tools**"



4. Click the "**MfgLoader.exe**".

5.  Select the MCU option by name, if the MCU of module board is "i.MX6 Solo", please click "**MX6DL Linux Update**", and click "**Linux-ubuntu**" (Ubuntu GUI version) for the OS of flashing, then click "**Run MFG Tool**".
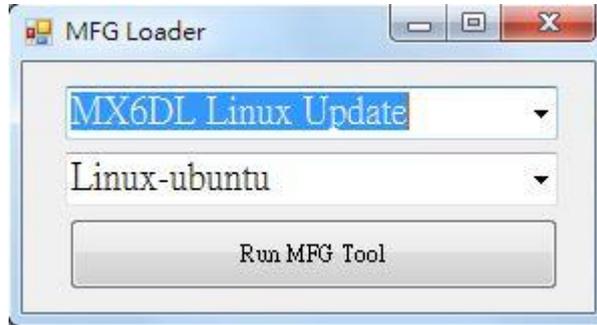


Or the if the MCU of module board is "i.MX6 Quad core", please click "**MX6Q Linux Update**", and click "**Linux-ubuntu**" (Ubuntu GUI version) for      the OS of flashing, then click "**Run MFG Tool**".
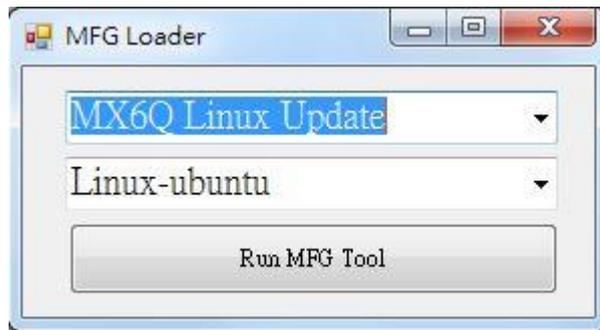


6.  The second screen will show up after clicking "Run MFG Tool", and please check whether it shows "**HID-compliant device**" as below, if not, please re-check the cable connection and DIP switch setting between mainboard and PC.

7.  Click "**Start**" to flash image file.



8.  It will show "Done" after flashing is finish, then click "Stop" and "Exit" to close the screen.



9.  You can also get the information from Terminal (debug port→COM1) after flashing is finish.
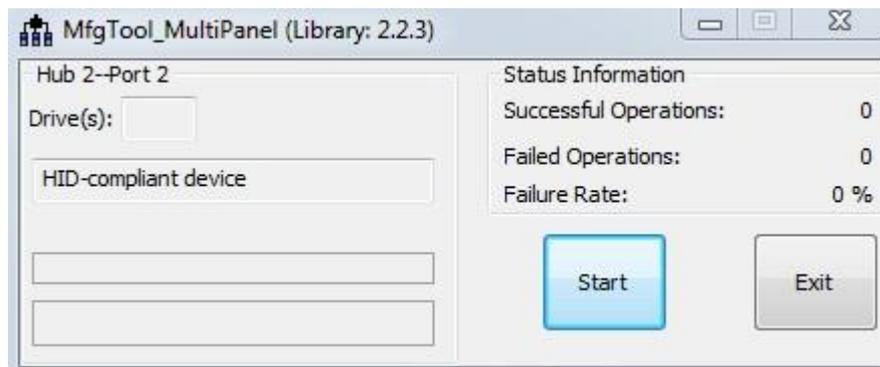
## 3.5 Create a bootable SD card with Ubuntu 12.04 file system

Please insert a SD card in the card reader on your Linux host PC

1)  Check device node of your SD card by command below.

    $cat /proc/partitions (for example, mine is /dev/sdd as below)

```
avalue@ubuntu:~/test/ltib$ cat /proc/partitions
major minor  #blocks  name

   8        0  104857600 sda
   8        1  100548608 sda1
   8        2          1 sda2
   8        5    4305920 sda5
   8       48    1000448 sdd
   8       49     996352 sdd1
avalue@ubuntu:~/test/ltib$
```

Create EXT3 partition for SD card

$ sudo fdisk /dev/sdd

Type the following parameters (each followed by <ENTER>):

d           [delete the previous partition]

n           [create a new partition]

p           [create a primary partition]

1           [the first partition]

20480      [20480x512bytes=10MB, which leaves enough space for the kernel, the
            boot loader and its configuration data]

<enter>     [using the default value will create a partition that spans to the last
            sector of the medium]

 w          [ this writes the partition table to the medium and fdisk exits].

2)  Format new partition in EXT3 format

    $sudo umount /dev/sdd1

    $sudo mkfs.ext3 /dev/sdd1

3)  Install bootloader on SD card by command below.

    $ sudo dd if=u-boot-solo.bin(u-boot-quad.bin) of=/dev/sdd   bs=**1k** seek=1 skip=1
    conv=fsync

4)  Install Linux kernel image on SD card by command below.

    $ sudo dd if=uImage of=/dev/sdd   bs=**1M** seek=1 conv=fsync

5)  Please find the Ubuntu file system from the path "CDROM\REV
    Image\MFG-REV-SA01_Image\Image\smarc\ ubuntu.tar.bz2" on User`s CD-ROM and
    copy it to the partition then follow up the command below.

$ sudo umount /dev/sdd1

$ sudo mount /dev/sdd1 /mnt

$ cd /mnt

$ sudo tar jxvpf ~/ubuntu.tar.bz2

$ cd

$ sudo umount /dev/sdd1

The Ubuntu file system content is now on the SD card. You can insert it to mainboard then turn on the DIP switch pin2&3 as below for booting.

## 3.6 Bootloader settings for booting from SD card

1) Please turn on the Pin 2&3 of the DIP switch as below for booting from SD card .



2) Insert SD card on SD socket. Connect RS232 cross over cable from **COM1** of mianboard to COM port of Host PC.

3) Run hyper terminal program on Host PC (teraterm on Windows or minicom on Linux)

4) Power on mainboard and press "space" key to get into bootloader menu.



```
U-Boot 2009.08 (Dec 11 2013 - 17:57:30)

CPU: Freescale i.MX6 family TO1.1 at 792 MHz
Thermal sensor with ratio = 178
Temperature:    38 C, calibration data 0x5694e069
mx6q pll1: 792MHz
mx6q pll2: 528MHz
mx6q pll3: 480MHz
mx6q pll8: 50MHz
ipg clock       : 66000000Hz
ipg per clock : 66000000Hz
uart clock      : 80000000Hz
cspi clock      : 60000000Hz
ahb clock       : 132000000Hz
axi clock    : 198000000Hz
emi_slow clock: 99000000Hz
ddr clock       : 396000000Hz
usdhc1 clock  : 198000000Hz
usdhc2 clock  : 198000000Hz
usdhc3 clock  : 198000000Hz
usdhc4 clock  : 198000000Hz
nfc clock       : 24000000Hz
Board: i.MX6DL/Solo: SMARC Board: 0x61011 [POR ]
Boot Device: SD
DRAM:   1 GB
MMC:    FSL_USDHC: 0,FSL_USDHC: 1,FSL_USDHC: 2,FSL_USDHC: 3
In:     serial
Out:    serial
Err:    serial
Net:    got MAC address from IIM: 00:00:00:00:00:00
FEC0 [PRIME]
Hit any key to stop autoboot:   0
MX6 Solo SMARC U-Boot >
```

5) Setup boot device
   SMARC U-Boot > print

6) Set boot device as below

→ SMARC U-Boot >setenv linux_cmd    'setenv bootargs ${linux_bootargs};mmc dev
              1;mmc read ${loadaddr} 0x800 0x3000;bootm'


→ SMARC U-Boot > setenv linux_bootargs 'console=tty0 console=ttymxc0,115200
              root=/dev/mmcblk1p1 rootwait rw'

→SMARC U-Boot> saveenv

→SMARC U-Boot> boot

```
SMARC U-Boot > setenv linux_cmd 'setenv bootargs ${linux_bootargs};mmc dev 1;mmc read ${loadaddr} 0x800 0x3000;bootm'
SMARC U-Boot > setenv linux_bootargs 'console=tty0 console=ttymxc0,115200 root=/dev/mmcblk1p1 rootwait rw'
SMARC U-Boot > saveenv
```

## 3.7 Bootloader settings for booting from onboard eMMC

1) Please turn on the Pin 1 of the DIP switch as below for booting from onboard eMMC.



2) Insert SD card on SD socket. Connect RS232 cross over cable from **COM1** of mianboard to COM port of Host PC.

3) Run hyper terminal program on Host PC (teraterm on Windows or minicom on Linux)

4) Power on mainboard and press "space" key to get into bootloader menu.

```
U-Boot 2009.08 (Dec 11 2013 - 17:57:30)

CPU: Freescale i.MX6 family TO1.1 at 792 MHz
Thermal sensor with ratio = 178
Temperature:   38 C, calibration data 0x5694e069
mx6q pll1: 792MHz
mx6q pll2: 528MHz
mx6q pll3: 480MHz
mx6q pll8: 50MHz
ipg clock      : 66000000Hz
ipg per clock  : 66000000Hz
uart clock     : 80000000Hz
cspi clock     : 60000000Hz
ahb clock      : 132000000Hz
axi clock      : 198000000Hz
emi_slow clock: 99000000Hz
ddr clock      : 396000000Hz
usdhc1 clock  : 198000000Hz
usdhc2 clock  : 198000000Hz
usdhc3 clock  : 198000000Hz
usdhc4 clock  : 198000000Hz
nfc clock      : 24000000Hz
Board: i.MX6DL/Solo: SMARC Board: 0x61011 [POR ]
Boot Device: SD
DRAM:   1 GB
MMC:   FSL_USDHC: 0,FSL_USDHC: 1,FSL_USDHC: 2,FSL_USDHC: 3
In:    serial
Out:   serial
Err:   serial
Net:   got MAC address from IIM: 00:00:00:00:00:00
FEC0 [PRIME]
Hit any key to stop autoboot:  0
MX6 Solo SMARC U-Boot >
```

5) Setup boot device

   SMARC U-Boot > print

6) Set boot device as below

→ SMARC U-Boot >setenv linux_cmd    'setenv bootargs ${linux_bootargs};mmc dev
              3;mmc read ${loadaddr} 0x800 0x3000;bootm'

→ SMARC U-Boot > setenv linux_bootargs 'console=tty0 console=ttymxc0,115200

root=/dev/mmcblk0p1 rootwait rw'

→SMARC U-Boot> saveenv

→SMARC U-Boot> boot

```
U-Boot > setenv linux_cmd 'setenv bootargs ${linux_bootargs};mmc dev 3;mmc read ${loadaddr} 0x800 0x3000;bootm'
U-Boot > setenv linux_bootargs 'console=tty0 console=ttymxc0,115200 root=/dev/mmcblk0p1 rootwait rw'
U-Boot > saveenv
```

## 3.8 Display output application of IMX6

This section describes how to setup the display output for LVDS, HDMI, VGA of IMX6 module.

1. You can find the file of resolution setup of LVDS&VGA under the directory "**..~/core/kernel/drivers/video/mxc**", for LVDS is "**ldb.c**", and for VGA is " **mxc_lcdif.c**".



"ldb.c"                                    "mxc_lcdif.c"

2. You need to fill the resolution parameter(.mode_str) on the file "**rev_sa01.c**" under "**~/core/kernel/arch/arm/mach-mx6/smarc/**", and the code of First Display is alway in the upper block.

3. Finally, you should fill the parameter for booting on the file "**mx6_smarc.h**" under the directory "~/core/u-boot/include/configs/".

   Please add "video=mxcfb0:dev=**display name**" on the column 132

   "rootwait rw \0" to enable display output function when booting.



4. Please refer ch1.3 to re-build the u-boot &Kernel binary file for booting.



**Note:** If you need to use double display output in Ubuntu, you should setup the "**rev_sa01.c**" file first then add the content "video=mxcfb0:dev=**first display name**

video=mxcfb1:dev=**second display neme"** to **mx6_smarc.h,** but for this application, you also need to write a program for controling the second diplay first or the second display will not enable after you follow up all the setting above.

## 3.9 Download Android Source Code for building image file

Please make a folder for storing the source code first then typing the command below to get started for the source code download.

$ sudo apt-get install git

$ **git clone guest@202.55.227.57:freescale/imx6/Android.git -b 4.4.2-SMARC**   About password, please check with BCM Sales or PM to get it.

## 3.10  Set up for building Android image file

We support to compile u-boot & Kernel on Ubuntu 12.04 (64bit version), other version of Ubuntu is not currently supported and may have built issues.

Install host packages needed by building code. This document assumes you are using Ubuntu. Not a requirement, but the packages may be named differently and the method of installing them may be different.

1)  Please follow up the commands below to install "Oracle JDK6.0" first for building up Android image file.

$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java6-installer

2)  Please follow up the commands below to install the necessary package for build image file.

$ sudo apt-get install git-core gnupg flex bison gperf build-essential \
    zip curl libc6-dev libncurses5-dev x11proto-core-dev \
    libx11-dev:i386 libreadline6-dev:i386 \
    libgl1-mesa-dev g++-multilib mingw32 openjdk-6-jdk tofrodos \
    python-markdown libxml2-utils xsltproc zlib1g-dev:i386 \
    ia32-libs u-boot-tools minicom lib32ncurses5-dev \

## 3.11 Building up Android image file

You can follow up the steps below to compile Android image file after download the source code.

1. Please move to the folder "Android" then start to compile image file.

```
leo@ubuntu:~/imx6_Android$ ls
Android
leo@ubuntu:~/imx6_Android$ cd Android/
leo@ubuntu:~/imx6_Android/Android$
```

2. Type the command to compile image file.

   $ **./run.sh –j16**

   (**-j number** means multi jobs for more efficiant building, you can add it according to your CPU performance of PC, e.g. mine is "–j16" as below ).

```
~/imx6_Android/Android$ ./run.sh -j16
```

3. You can find the finished image file(**u-boot-6q.bin, u-boot-6solo.bin, system.img, recover.img, boot.img**) as below after compiling on the directory **~/Android/out/target/product/smarc.**

```
leo@ubuntu:~/imx6_Android/Android/out/target/product/smarc$ ls -al
total 314160
drwxrwxr-x  9 leo leo      4096 Jan  2 23:44 .
drwxrwxr-x  3 leo leo      4096 Jan  2 22:09 ..
-rw-rw-r--  1 leo leo        12 Jan  2 22:47 android-info.txt
-rw-r--r--  1 leo leo   4784128 Jan  2 23:34 boot.img
-rw-rw-r--  1 leo leo     29030 Jan  2 22:45 clean_steps.mk
drwxrwxr-x  4 leo leo      4096 Jan  2 23:36 data
-rw-rw-r--  1 leo leo     53886 Jan  2 23:44 installed-files.txt
-rwxr-xr-x  1 leo leo   4593124 Jan  2 22:52 kernel
drwxrwxr-x 14 leo leo      4096 Jan  2 23:44 obj
-rw-rw-r--  1 leo leo       373 Jan  2 22:45 previous_build_config.mk
-rw-rw-r--  1 leo leo    186407 Jan  2 23:34 ramdisk.img
-rw-rw-r--  1 leo leo    701522 Jan  2 23:34 ramdisk-recovery.img
drwxrwxr-x  3 leo leo      4096 Jan  2 23:34 recovery
-rw-r--r--  1 leo leo   5298176 Jan  2 23:34 recovery.img
drwxrwxr-x  8 leo leo      4096 Jan  2 23:33 root
drwxrwxr-x  5 leo leo      4096 Jan  2 23:33 symbols
drwxrwxr-x 12 leo leo      4096 Jan  2 23:33 system
-rw-r--r--  1 leo leo 293601280 Jan  2 23:44 system.img
drwxrwxr-x  3 leo leo      4096 Jan  2 23:19 test
-rwxr-xr-x  1 leo leo    445804 Jan  2 22:47 u-boot-6q.bin
-rwxr-xr-x  1 leo leo    445556 Jan  2 22:46 u-boot-6solo.bin
-rwxr-xr-x  1 leo leo    445804 Jan  2 22:47 u-boot.bin
-rwxr-xr-x  1 leo leo   4593188 Jan  2 22:52 uImage
-rw-r--r--  1 leo leo 134217728 Jan  2 23:37 userdata.img
```

PS: If you would like to use **Mfgtool** for flashing image file, you must put all the files **u-boot-6q.bin, u-boot-6solo.bin, system.img, recover.img, boot.img** under "~\Image\smarc\android" for right detected path.
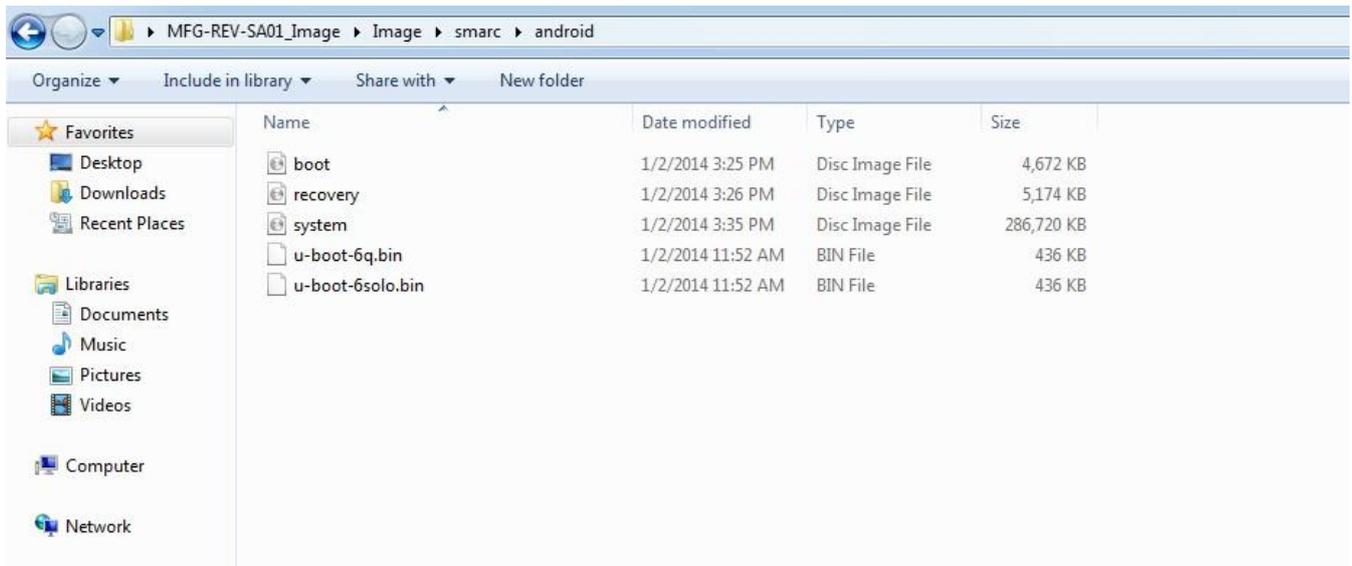
## 3.12  Use MfgTool to flash Android into onboard eMMC

Manufacturing tool, a successor of ATK, provides a series of new features to power your mass production work. The features like windows style GUI, multiple devices support, explicit status monitoring, versatile functionalities and highly flexible architecture make it a best choice to meet your critical timing, cost and customization requirements.

For using Mfgtool to flash image file into onboard eMMC, please follow up the steps below

1) Please turn on the **Pin4** of the DIP switch as below into burning mode of Mfgtool.
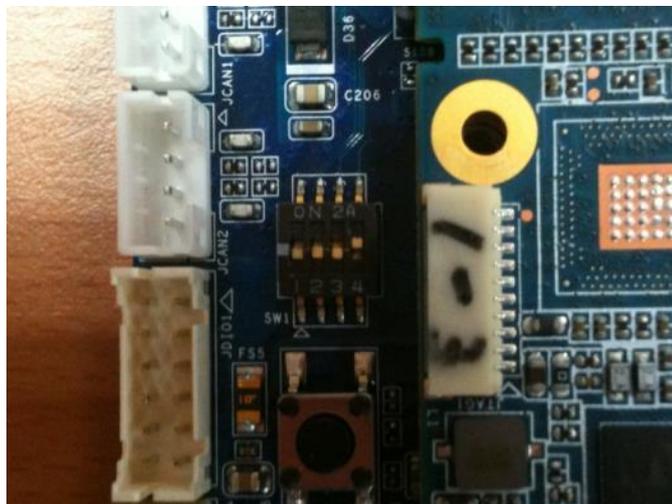


2) Power on the mainboard then plug the cable from OTG socket to PC.

3) Select the right folder by MCU name, if the MCU of module board is "**i.MX6 Solo**", please click the folder "~\ **MX6DL-IMX6**" to flash image file, e.g. mine is D:\ MFG-REV-SA01_Image\**MX6DL-IMX6**"
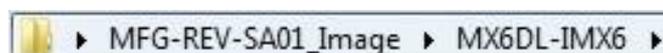
**SMA-IMX6 User's Manual**

On the other hand, if the MCU of module board is "**i.MX6 Quad core**", please click the folder "~\ **MX6Q-IMX6**" to flash image file, e.g. mine is
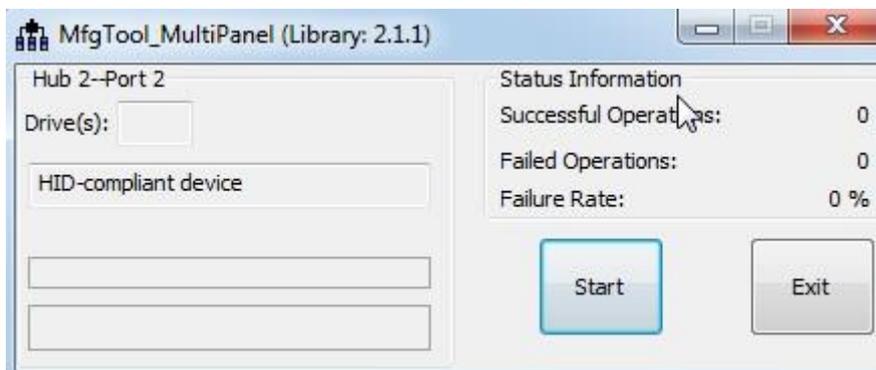
D:\ MFG-REV-SA01_Image\ **MX6Q-IMX6**"



4) Click "MfgTool2.exe" to flash image file into smarc module.



| Name | Date modified | Type | Size |
|---|---|---|---|
| Document | 1/7/2014 11:27 AM | File folder | |
| Profiles | 1/7/2014 11:27 AM | File folder | |
| Utils | 1/7/2014 11:27 AM | File folder | |
| cfg | 9/18/2013 3:54 PM | Configuration sett... | 1 KB |
| i.MX 6 DQ Profiles Included readme | 1/17/2013 2:42 AM | Text Document | 2 KB |
| InternalParam | 1/6/2014 4:52 PM | Configuration sett... | 1 KB |
| MfgTool | 1/6/2014 4:53 PM | Text Document | 2 KB |
| MfgTool2 release notes | 11/22/2012 6:27 AM | Text Document | 3 KB |
| MfgTool2 user guide | 11/22/2012 6:27 AM | Text Document | 4 KB |
| MfgTool2 | 12/18/2012 8:39 PM | Application | 1,661 KB |
| MfgToolLib.dll | 12/18/2012 8:39 PM | Application extens... | 600 KB |
| UICfg | 12/18/2012 8:39 PM | Configuration sett... | 1 KB |

5) Click "**Start**" to flash image file.



6) It will show "Done" after flashing is finish, then click "Stop" and "Exit" to close the screen.

7) You can also get the information from Terminal (debug port→COM1) after flashing is finish.