

RQM-IMX6

Freescale i.MX6 Cortex-A9 Q7 Module

User's Manual

1st Ed – 30 September 2015

Copyright Notice

Copyright © 2015 Avalue Technology Inc., ALL RIGHTS RESERVED.

Part No.E2047RQM600R

Content

1. Getting Started	3
1.1 Safety Precautions	3
1.2 Packing List.....	3
1.3 Document Amendment History.....	4
1.4 Manual Objectives	5
1.5 System Specifications	6
1.6 Architecture Overview—Block Diagram.....	7
2. Hardware Configuration	8
2.1 Product Overview	9
2.2 Jumper and Connector List.....	10
2.3 Setting Jumpers & Connectors	11
2.3.1 Boot set selector (BTST1).....	11
2.3.2 Serial Port 1 connector (JCOM1)	11
2.3.3 Camera connector (JCAMERA1)	12
2.3.4 QSeven connector (Q7_GF1)	13
3. Software Setup	15
3.1 Software Setup	16
4. Mechanical Drawing	26

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.

1.2 Packing List

- 1 x RQM-IMX6 Freescale i.MX6 Cortex-A9 Q7 Module
- 4 x Screw



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

1.3 Document Amendment History

Revision	Date	By	Comment
1 st	September 2015	Avalue	Initial Release

1.4 Manual Objectives

This manual describes in details Avalue Technology RQM-IMX6 QSeven Module.

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 RQM-IMX6 QSeven Module 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

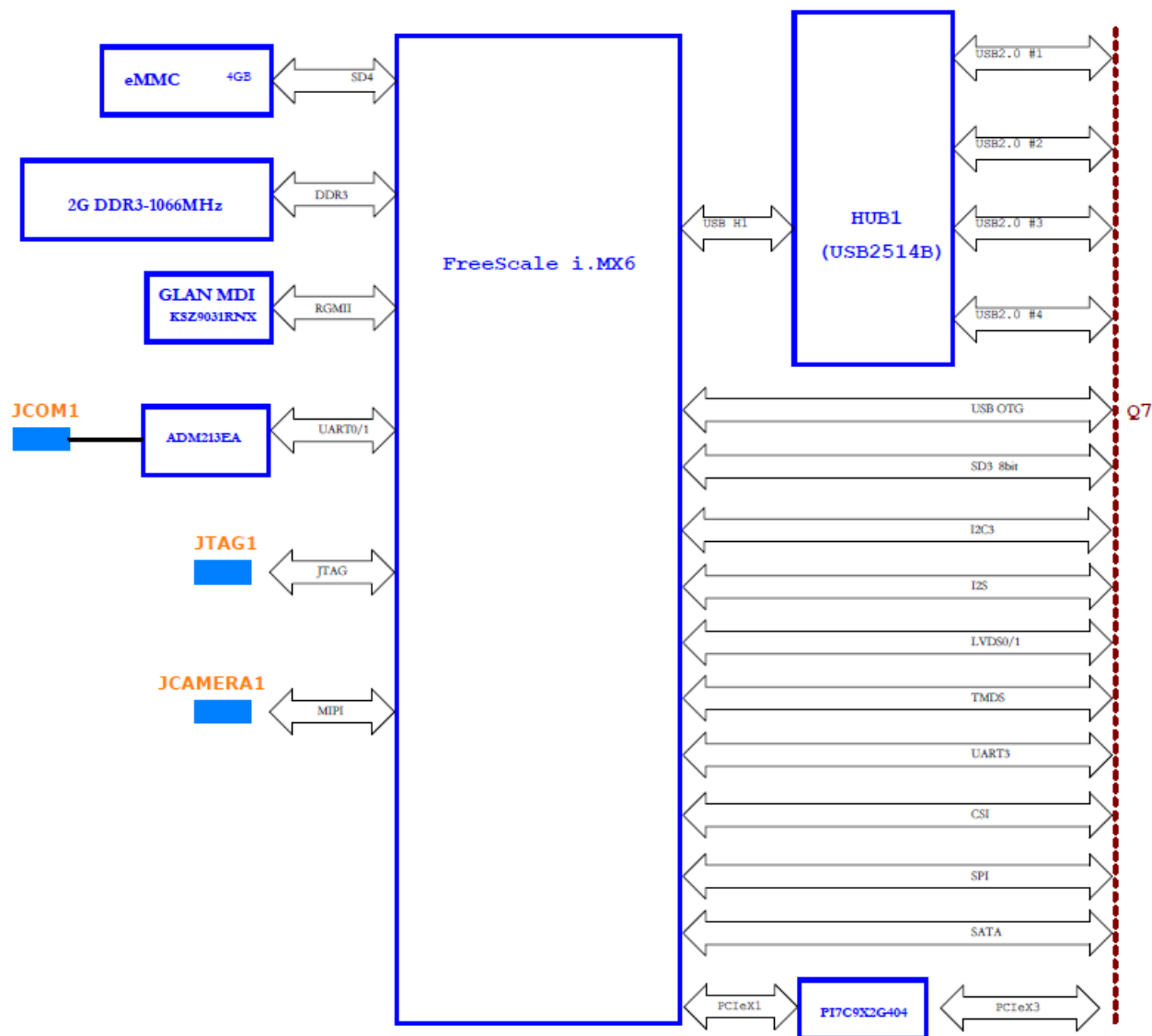
System	
CPU	Freescall i.MX 6 Solo/ Quad
System Memory	1GB, up to 2GB
Expansion	4GB eMMC, up to 32GB
I/O	
MIO	1 x SATA (Ubuntu only) 1 x PCIe x 1, up to 3 x PCIe x 1
USB	4 x USB 2.0 1 x OTG
GPIO	SDIO 8Bit, SPI, I2C, UART, CAN, GPIO, JTAG
Display	
Multiple Display	Dual Display HD 1080p Encode and Decode 2D and 3D Acceleration
Ethernet	
LAN Chip	Micrel KSZ9031
Ethernet Interface	10/100/1000 Base-Tx Gigabit Ethernet Compatible
Mechanical & Environmental	
Power Requirement	+5V
Power Type	Qseven Power Spec
Operating Temp.	0°C ~ 60°C -40°C ~ 75°C (optional)
Storage Temp.	-40°C ~ 85°C
Operating Humidity	0% ~ 90% Relative Humidity, Non-condensing
Size (L x W) (Please consult product engineers for the production feasibility if the size is larger than 410x360mm or smaller than 80x70mm)	70mm x 70mm
Weight	TBD



Note: Specifications are subject to change without notice.

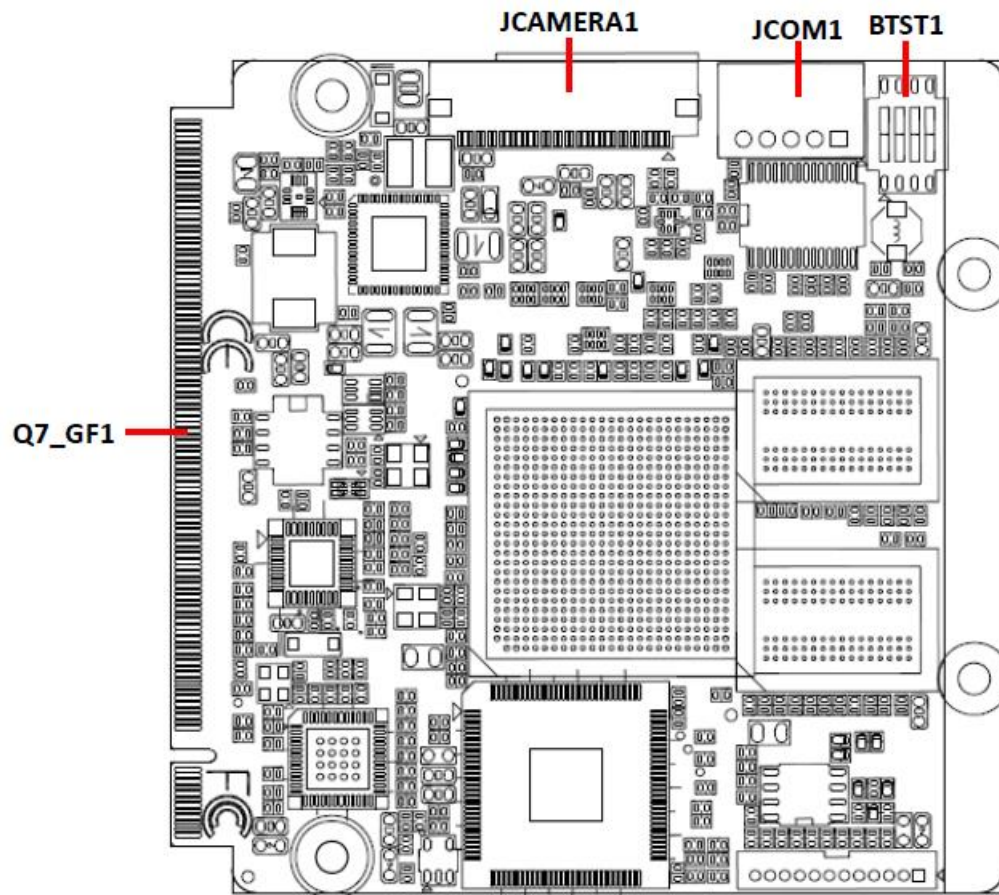
1.6 Architecture Overview—Block Diagram

The following block diagram shows the architecture and main components of RQM-IMX6 QSeven Module.



2. Hardware Configuration

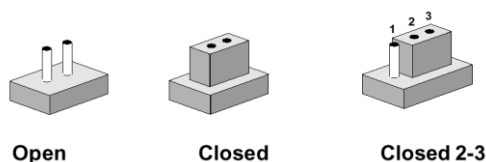
2.1 Product Overview



2.2 Jumper and Connector List

You can configure your board to match the needs of your application by setting jumpers. A jumper is the simplest kind of electric switch.

It consists of two metal pins and a small metal clip (often protected by a plastic cover) that slides over the pins to connect them. To “close” a jumper you connect the pins with the clip. To “open” a jumper you remove the clip. Sometimes a jumper will have three pins, labeled 1, 2, and 3. In this case, you would connect either two pins.



The jumper settings are schematically depicted in this manual as follows:



A pair of needle-nose pliers may be helpful when working with jumpers.

Connectors on the board are linked to external devices such as hard disk drives, a keyboard, or floppy drives. In addition, the board has a number of jumpers that allow you to configure your system to suit your application.

If you have any doubts about the best hardware configuration for your application, contact your local distributor or sales representative before you make any changes.

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

Jumpers

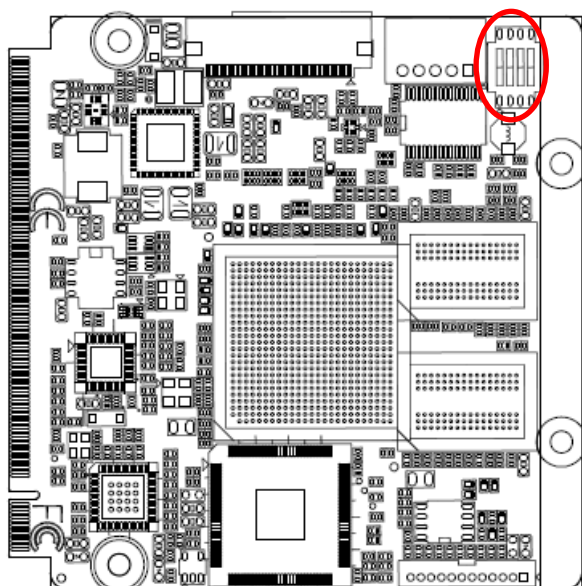
Label	Function	Note
BTST1	Boot set selector	

Connectors

Label	Function	Note
JCOM1	Serial Port 1 connector	5 x 1 wafer, pitch 2.00mm
JCAMERA1	Camera connector	
Q7_GF1	QSeven connector	

2.3 Setting Jumpers & Connectors

2.3.1 Boot set selector (BTST1)



OTG load



SD boot



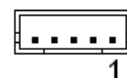
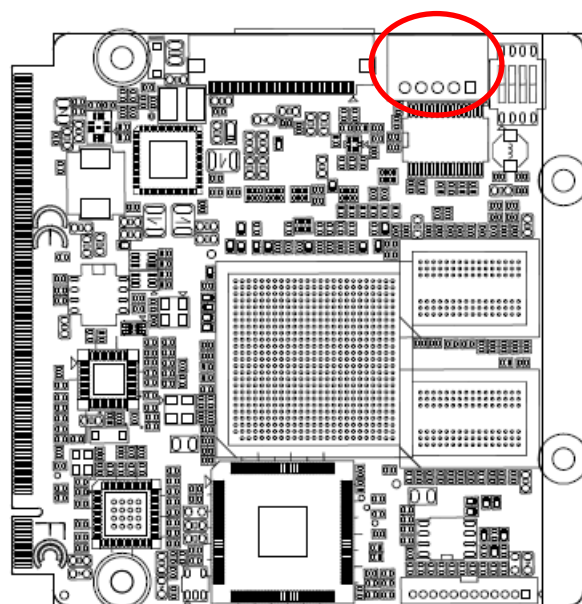
eMMC boot



SPI boot



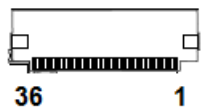
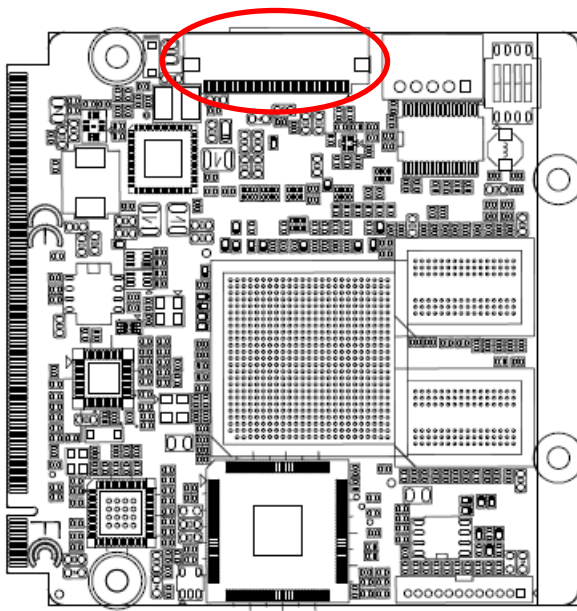
2.3.2 Serial Port 1 connector (JCOM1)



Signal	PIN
RXDD1	1
TXDD1	2
GND	3
TXDD2	4
RXDD2	3

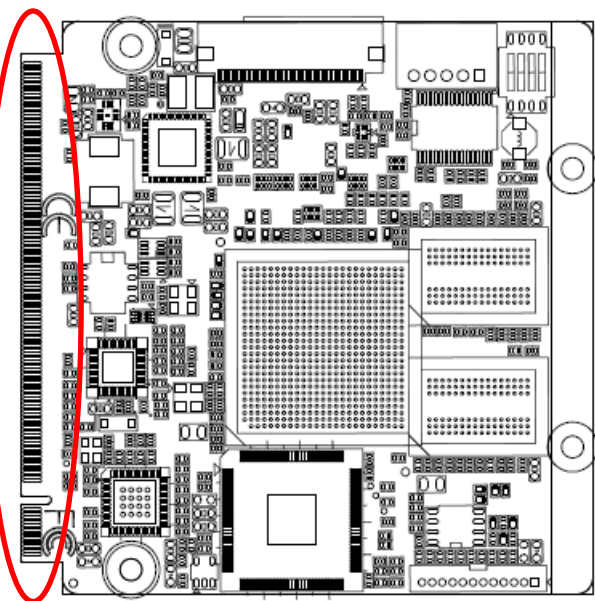
RQM-IMX6

2.3.3 Camera connector (JCAMERA1)



Signal	PIN
+3.3V	1
+3.3V	2
CSI_D0P	3
CSI_D0M	4
GND	5
CSI_D1P	6
CSI_D1M	7
GND	8
NC	9
NC	10
CSI_RST_B	11
NC	12
NC	13
GND	14
CSI_CLK0P	15
CSI_CLK0M	16
GND	17
CSI_SCL	18
CSI_SDA	19
CSI_PWRDOWN	20
CAMERA_MCLK	21
NC	22
NC	23
NC	24
GND	25
NC	26
NC	27
GND	28
NC	29
NC	30
NC	31
NC	32
NC	33
GND	34
NC	35
NC	36

2.3.4 QSeven connector (Q7_GF1)



Signal	PIN	PIN	Signal
GND	1	2	GND
GBE_MDI3-	3	4	GBE_MDI2-
GBE_MDI3+	5	6	GBE_MDI2+
GBE_LINK100#	7	8	GBE_LINK1000#
GBE_MDI1-	9	10	GBE_MDI0-
GBE_MDI1+	11	12	GBE_MDI0+
NC	13	14	GBE_ACT#
NC	15	16	NC
NC	17	18	SUS_S3#
NC	19	20	PWRBTN#
NC	21	22	NC
GND	23	24	GND
GND	25	26	PWGIN
BATLOW#	27	28	RSTBTN#
SATA0_TX+	29	30	NC
SATA0_TX-	31	32	NC
SATA_ACT#	33	34	GND
SATA0_RX+	35	36	NC
SATA0_RX-	37	38	NC

Signal	PIN	PIN	Signal
GND	39	40	GND
BIOS_DISABLE#	41	42	SDIO_CLK#
SDIO_CD#	43	44	SDIO_LED
SDIO_CMD	45	46	SDIO_WP
SDIO_PWR#	47	48	SDIO_DAT1
SDIO_DAT0	49	50	SDIO_DAT3
SDIO_DAT2	51	52	SDIO_DAT5
SDIO_DAT4	53	54	SDIO_DAT7
SDIO_DAT6	55	56	USB_DRIVE_VBUS
GND	57	58	GND
HDA_SYNC	59	60	SMB_CLK
HDA_RST#	61	62	SMB_DAT
HDA_BCLK	63	64	SMB_ALERT#
HDA_SDI	65	66	GP0_I2C_CLK
HDA_SDO	67	68	GP0_I2C_DAT
NC	69	70	WDTRIG#
NC	71	72	WDOUT
GND	73	74	GND
NC	75	76	NC
NC	77	78	NC
NC	79	80	USB_4_5_OC#
NC	81	82	USB_P4-
NC	83	84	USB_P4+
USB_2_3_OC#	85	86	USB_0_1_OC#
USB_P3-	87	88	USB_P2-
USB_P3+	89	90	USB_P2+
USB_VBUS	91	92	USB_ID
USB_P1-	93	94	USB_P0-
USB_P1+	95	96	USB_P0+
GND	97	98	GND
LVDS_A0+	99	100	LVDS_B0+
LVDS_A0-	101	102	LVDS_B0-
LVDS_A1+	103	104	LVDS_B1+

RQM-IMX6

Signal	PIN	PIN	Signal
LVDS_A1-	105	106	LVDS_B1-
LVDS_A2+	107	108	LVDS_B2+
LVDS_A2-	109	110	LVDS_B2-
LVDS_PPEN	111	112	LVDS_BLEN
LVDS_A3+	113	114	LVDS_B3+
LVDS_A3-	115	116	LVDS_B3-
GND	117	118	GND
LVDS_A_CLK+	119	120	LVDS_B_CLK+
LVDS_A_CLK-	121	122	LVDS_B_CLK-
LVDS_BLT_CTRL	123	124	GP_1-Wire_Bus
LVDS_DID_DAT	125	126	LVDS_BLC_DAT
LVDS_DID_CLK	127	128	LVDS_BLC_CLK
CAN0_TX	129	130	CAN0_RX
TMDS_CLK+	131	132	NC
TMDS_CLK-	133	134	NC
GND	135	136	GND
TMDS_LANE1+	137	138	NC
TMDS_LANE1-	139	140	NC
GND	141	142	GND
TMDS_LANE0+	143	144	NC
TMDS_LANE0-	145	146	NC
GND	147	148	GND
TMDS_LANE2+	149	150	HDMI_CTRL_DAT
TMDS_LANE2-	151	152	HDMI_CTRL_CLK
DP_HDMI_HPD#	153	154	NC
PCIE_CLK_REF+	155	156	PCIE_WAKE#
PCIE_CLK_REF-	157	158	PCIE_RST#
GND	159	160	GND
NC	161	162	NC
NC	163	164	NC
GND	165	166	GND
PCIE2_TX+	167	168	PCIE2_RX+
PCIE2_TX-	169	170	PCIE2_RX-

Signal	PIN	PIN	Signal
UART0_TX	171	172	UART0_RTS#
PCIE1_TX+	173	174	PCIE1_RX+
PCIE1_TX-	175	176	PCIE1_RX-
UART0_RX	177	178	UART0_CTS#
PCIE0_TX+	179	180	PCIE0_RX+
PCIE0_TX-	181	182	PCIE0_RX-
GND	183	184	GND
GPIO	185	186	GPIO
GPIO	187	188	GPIO
GPIO	189	190	GPIO
GPIO	191	192	GPIO
VCC_RTC	193	194	SPKR
FAN_TACHOIN	195	196	FAN_PWMOUT
GND	197	198	GND
SPI_MOSI	199	200	SPI_CS0#
SPI_MISO	201	202	SPI_CS1#
SPI_SCK	203	204	MFG_NC4
VCC_5V_SB1	205	206	VCC_5V_SB2
MFG_NC0	207	208	MFG_NC2
MFG_NC1	209	210	MFG_NC3
VCC1	211	212	VCC2
VCC3	213	214	VCC4
VCC5	215	216	VCC6
VCC7	217	218	VCC8
VCC9	219	220	VCC10
VCC11	221	222	VCC12
VCC13	223	224	VCC14
VCC15	225	226	VCC16
VCC17	227	228	VCC18
VCC19	229	230	VCC20

3. Software Setup

3.1 Software Setup

Build and install Android image for RQM-IMX6

Here you can find instruction to setup development environment for Android source code for RQM-IMX6 and the way to install it on eMMC. With this guideline, user will be able to setup the system easily and test all the functions with the system.

Setup Build Environment

Please following the command below to install Oracle JDK6.0 and other packages on Ubuntu 12.04 or 14.04.

```
#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
#sudo update-alternatives --config java
#sudo apt-get install mtools
#sudo apt-get install squashfs-tools
#sudo apt-get install mksquashfs
```

Please refer to hyperlink below to setup development environment

[Initializing a Build Environment](#)

Compile Android Source code

Please following instruction below to compile Android source code

```
#cd Android
#./run.sh -j4
```

You can find all image files in path **Android/out/target/product/smarc/**

Image File	Description
boot.img	kernel image file
recovery.img	recovery image file
system.img	system image file
u-boot-6q.bin	bootloader for IMX6-POS Quad core
u-boot-6dl.bin	bootloader for IMX6-POS Dual lite

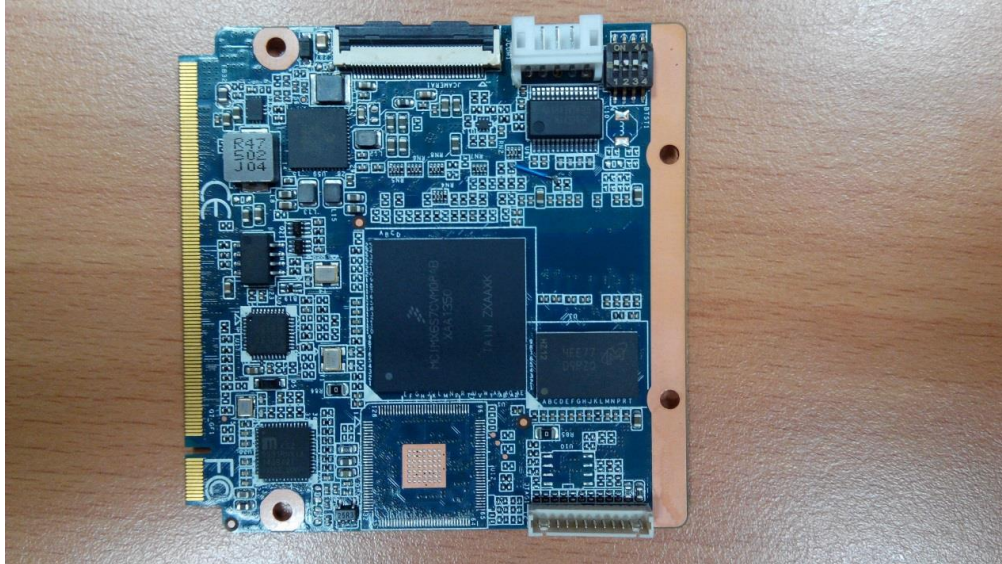
Please copy all of them to path **Android-MfgTools\Images\smarc\android** of MfgTool folder.

Install Android image into eMMC

Download MfgTool for Android from Hyperlink below

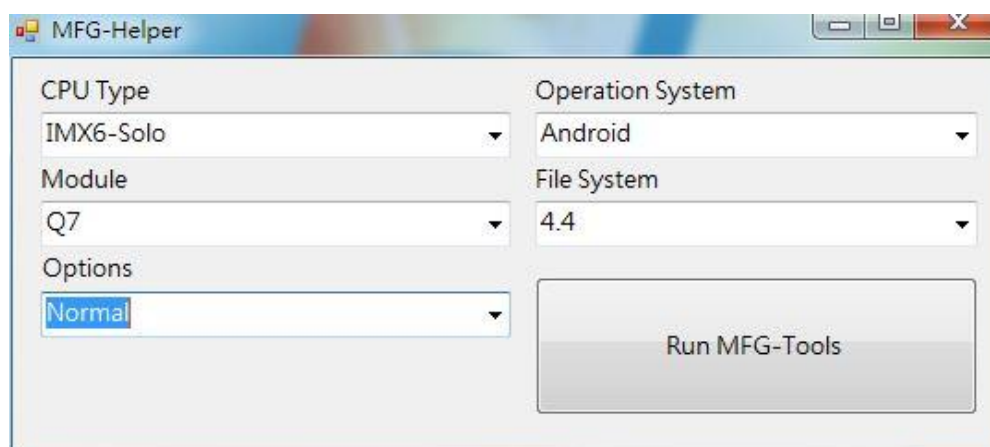
[MfgTool for Android](#)

1. Set BTST1 on RQM-IMX6(upper board) as below to serial download mode

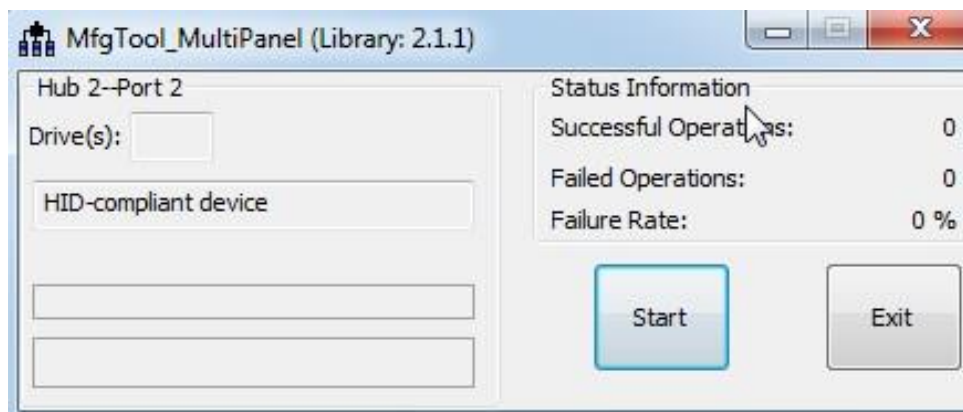
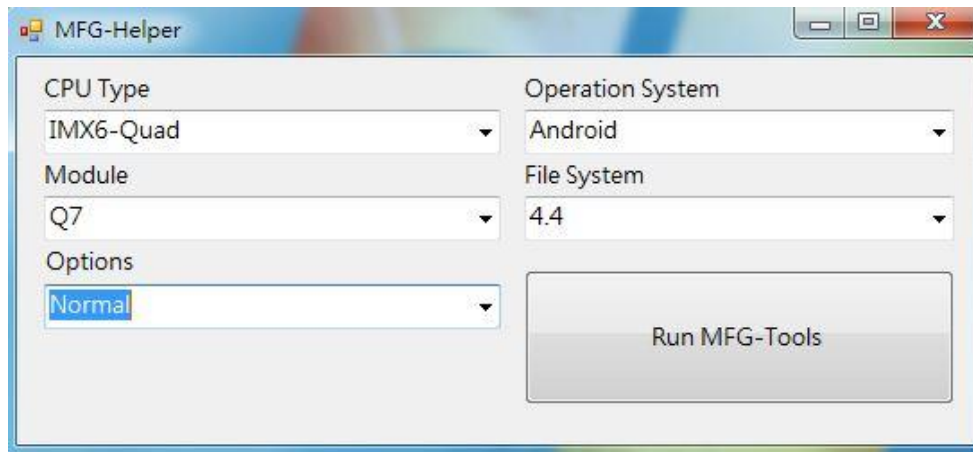


2. Use mini-USB cable to connect JMUSB1 on RQM-IMX6 to Windows 7 system.
3. Run MfgTool2.exe and click **start** to flash image file.

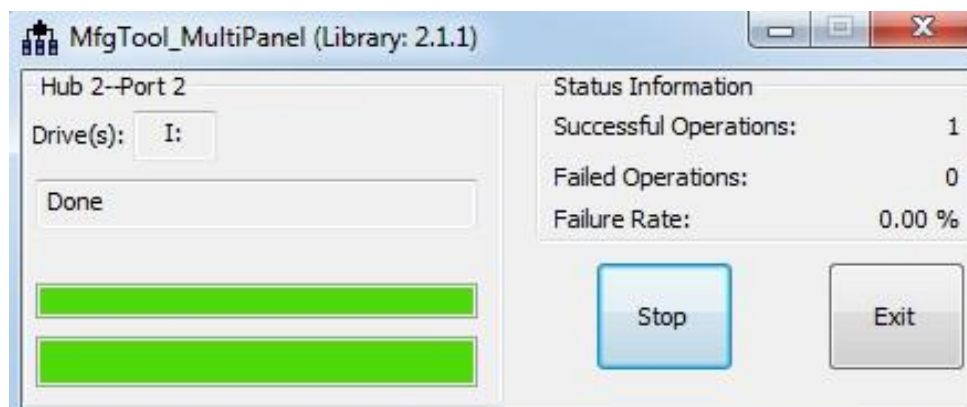
MfgTool	Description
Android-MfgTools\MX6DL-Android\MfgTool2.exe	MfgTool for i.MX6 Solo or Dual Lite
Android-MfgTools\MX6Q-Android\MfgTool2.exe	MfgTool for i.MX6 Dual & Quad



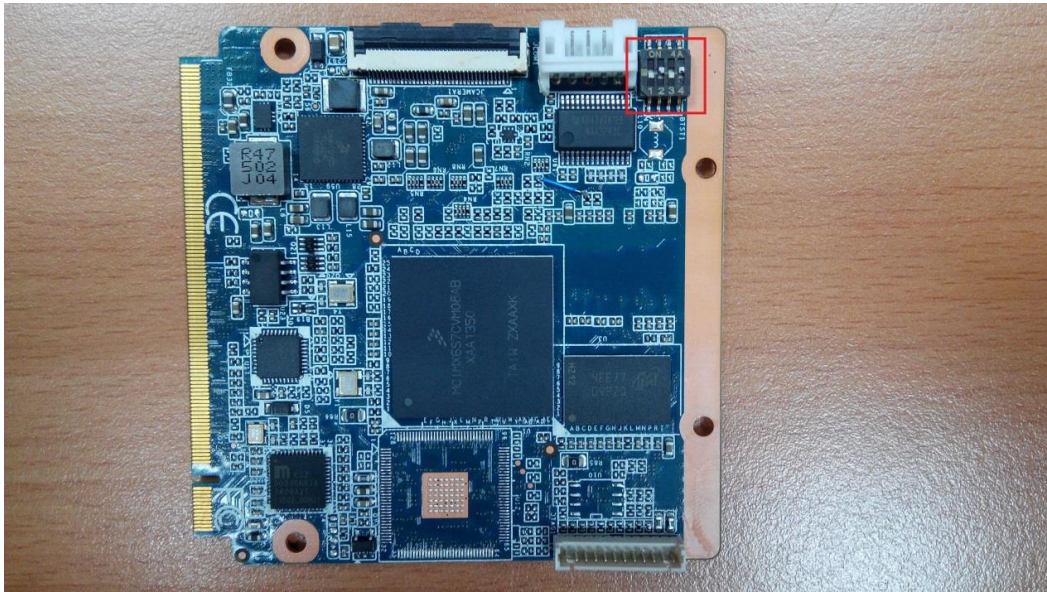
RQM-IMX6



4. It will show “Done” after flashing is finish, then click “Stop” and “Exit” to close the screen.



5. Set BTST1 as below to boot from eMMC



6. Restart the system then boot in Android

Setting for different display (LDB, LCD, HDMI, LCD+HDMI, LDB+HDMI, LDB+LCD)

1.HDMI

```
{
    .disp_dev = "hdm",
    .interface_pix_fmt = IPU_PIX_FMT_RGB24,
    .mode_str = "1920x1080M@60",
    .default_bpp = 32,
    .int_clk = false,
    .late_init = false,
},

mx6q_init_audio();
smarc_fb_init();
// set_smarc_lvds();
// imx6q_add_lcdif(&lcdif_data);
// imx6q_add_ldb(&ldb_data);
set_smarc_hdmi();
imx6q_add_vpu();
imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
imx6q_add_v4l2_output(0);
```

RQM-IMX6

```
    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();

static struct fsl_mxc_hdmi_core_platform_data hdmi_core_data = {
    .ipu_id = 0,
    .disp_id = 0,
};
```

2. LCD

```
    {
        .disp_dev = "lcd",
        .interface_pix_fmt = IPU_PIX_FMT_BGR24,
        .mode_str = "LCD_XGA",
        .default_bpp = 16,
        .int_clk = false,
    },
void __init smarc_board_init(void)
{
    mx6q_init_audio();
    smarc_fb_init();
    // set_smarc_lvds();
    imx6q_add_lcdif(&lcdif_data);
    // imx6q_add_ldb(&ldb_data);
    // set_smarc_hdmi();
    imx6q_add_vpu();
    imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
    imx6q_add_v4l2_output(0);

    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();
static struct fsl_mxc_lcd_platform_data lcdif_data = {
    .ipu_id = 0,
    .disp_id = 0,
    .default_ifmt = IPU_PIX_FMT_BGR24,
};
```

3.LDB(LVDS)

```

    {
        .disp_dev = "ldb",
        .interface_pix_fmt = IPU_PIX_FMT_RGB666,
        .mode_str = "SVGA",
        .default_bpp = 32,
        .int_clk = false,
        .late_init = false,
    },
void __init smarc_board_init(void)
{
    mx6q_init_audio();
    smarc_fb_init();
    set_smarc_lvds();
//    imx6q_add_lcdif(&lcdif_data);
    imx6q_add_ldb(&ldb_data);
//    set_smarc_hdmi();
    imx6q_add_vpu();
    imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
    imx6q_add_v4l2_output(0);

    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();
static struct fsl_mxc_ldb_platform_data ldb_data = {
    .ipu_id          = 0,
    .disp_id         = 0,
    .ext_ref         = 1,
    .mode            = LDB_SEP0,
    .sec_ipu_id      = 0,
    .sec_disp_id     = 1,
};

```

4.LCD+HDMI

```

static struct ipuv3_fb_platform_data smarc_fb_data[] = {

    {

```

```
.disp_dev = "lcd",
.interface_pix_fmt = IPU_PIX_FMT_RGB24,
.mode_str = "LCD_WVGA",
.default_bpp = 32,
.int_clk = true,
},

{
.disp_dev = "hdmi",
.interface_pix_fmt = IPU_PIX_FMT_RGB24,
.mode_str = "1920x1080M@60",
.default_bpp = 32,
.int_clk = false,
.late_init = false,
},

void __init smarc_board_init(void)
{
    mx6q_init_audio();
    smarc_fb_init();
    // set_smarc_lvds();
    imx6q_add_lcdif(&lcdif_data);
    // imx6q_add_ldb(&ldb_data);
    set_smarc_hdmi();
    imx6q_add_vpu();
    imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
    imx6q_add_v4l2_output(0);

    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();
    static struct fsl_mxc_lcd_platform_data lcdif_data = {
        .ipu_id = 0,
        .disp_id = 0,
        .default_ifmt = IPU_PIX_FMT_BGR24,
    };

    static struct fsl_mxc_hdmi_core_platform_data hdmi_core_data = {
        .ipu_id = 1,
```

```

        .disp_id = 0,
    };

```

5. LDB+HDMI

```

static struct ipuv3_fb_platform_data smarc_fb_data[] = {

    {
        .disp_dev = "ldb",
        .interface_pix_fmt = IPU_PIX_FMT_RGB666,
        .mode_str = "SVGA",
        .default_bpp = 32,
        .int_clk = false,
        .late_init = false,
    },

    {
        .disp_dev = "hdmi",
        .interface_pix_fmt = IPU_PIX_FMT_RGB24,
        .mode_str = "1920x1080M@60",
        .default_bpp = 32,
        .int_clk = false,
        .late_init = false,
    },
};

void __init smarc_board_init(void)
{
    mx6q_init_audio();
    smarc_fb_init();
    set_smarc_lvds();
    // imx6q_add_lcdif(&lcdif_data);
    imx6q_add_ldb(&ldb_data);
    set_smarc_hdmi();
    imx6q_add_vpu();
    imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
    imx6q_add_v4l2_output(0);

    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();
}

static struct fsl_mxc_ldb_platform_data ldb_data = {

```

RQM-IMX6

```
.ipu_id          = 0,
.disp_id         = 0,
.ext_ref         = 1,
.mode            = LDB_SEP0,
.sec_ipu_id      = 0,
.sec_disp_id     = 1,
};

static struct fsl_mxc_hdmi_core_platform_data hdmi_core_data = {
    .ipu_id = 1,
    .disp_id = 0,
};
```

6.LDB+LCD

```
static struct ipuv3_fb_platform_data smarc_fb_data[] = {

    {
        .disp_dev = "ldb",
        .interface_pix_fmt = IPU_PIX_FMT_RGB666,
        .mode_str = "SVGA",
        .default_bpp = 32,
        .int_clk = false,
        .late_init = false,
    },

    {
        .disp_dev = "lcd",
        .interface_pix_fmt = IPU_PIX_FMT_BGR24,
        .mode_str = "LCD_XGA",
        .default_bpp = 16,
        .int_clk = false,
    },

};

void __init smarc_board_init(void)
{
    mx6q_init_audio();
    smarc_fb_init();
    set_smarc_lvds();
}
```



```

    imx6q_add_lcdif(&lcdif_data);
    imx6q_add_ldb(&ldb_data);
//    set_smarc_hdmi();
    imx6q_add_vpu();
    imx_add_viv_gpu(&imx6_gpu_data, &imx6q_gpu_pdata);
    imx6q_add_v4l2_output(0);

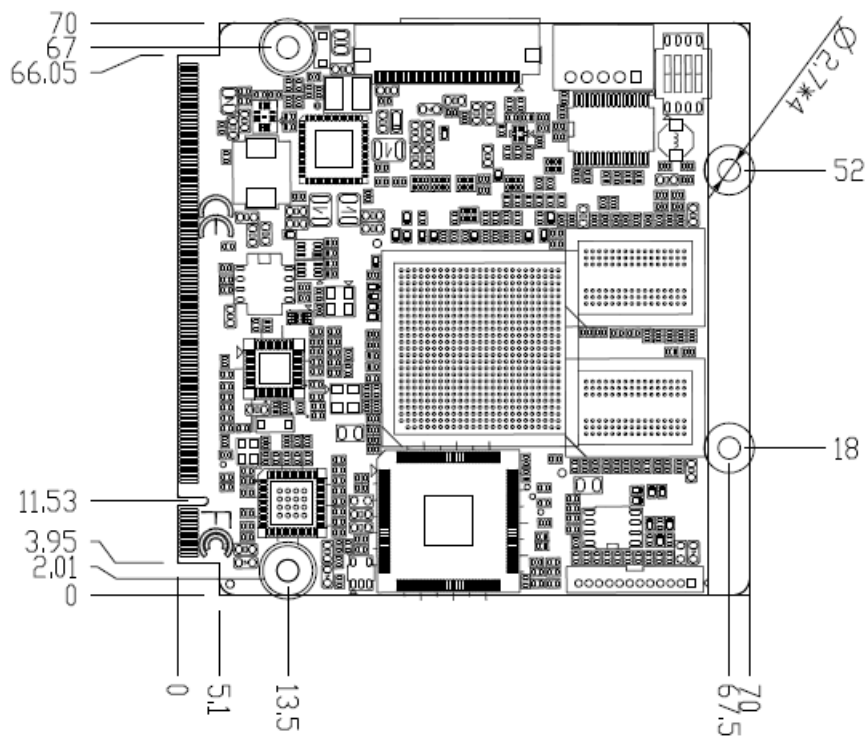
    set_smarc_gpio();
    mx6q_smarc_init_uart();
    set_smarc_sdio();
    smarc_can_init();
static struct fsl_mxc_ldb_platform_data ldb_data = {
    .ipu_id          = 0,
    .disp_id         = 0,
    .ext_ref         = 1,
    .mode            = LDB_SEP0,
    .sec_ipu_id      = 0,
    .sec_disp_id     = 1,
};

static struct fsl_mxc_lcd_platform_data lcdif_data = {
    .ipu_id = 1,
    .disp_id = 0,
    .default_ifmt = IPU_PIX_FMT_BGR24,
};

```

4. Mechanical Drawing





Unit: mm

