RSC-IMX51

Freescale™ i.MX515 Cortex-A8 800MHz Micro Module with 4GB eMMC, Optional HDMI & 512MB DDR2

User's Manual

3rd Ed – 11 October 2013

Part No. E2047A41802R

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 INSTALLED 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.

Notice

This guide is designed for experienced users to setup the system within the shortest time. For detailed information, please always refer to the electronic user's manual.

Copyright Notice

Trademark Acknowledgement

Brand and product names are trademarks or registered trademarks of their respective owners.

Disclaimer

Avalue Technology Inc. reserves the right to make changes, without notice, to any product, including circuits and/or software described or contained in this manual in order to improve design and/or performance. Avalue Technology assumes no responsibility or liability for the use of the described product(s), conveys no license or title under any patent, copyright, or masks work rights to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless

otherwise specified. Applications that are described in this manual are for illustration purposes only. Avalue Technology Inc. makes no representation or warranty that such application will be suitable for the specified use without further testing or modification.

Life Support Policy

Avalue Technology's PRODUCTS ARE NOT FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE PRIOR WRITTEN APPROVAL OF Avalue Technology Inc.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into body, or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

A Message to the Customer

Avalue Customer Services

Each and every Avalue's 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 Avalue 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 for which the name Avalue has come to be known.

Your satisfaction is our primary concern. Here is a guide to Avalue's 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.

To receive the latest version of the user's manual; please visit our Web site at:
http://www.avalue.com.tw/

Product Warranty

Avalue warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Avalue, or which have been subject to misuse, abuse, accident or improper installation. Avalue assumes no liability under the terms of this warranty as a consequence of such events. Because of Avalue's high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If any of Avalue's products is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time, and freight. Please consult your dealer for more details. If you think you have a defective product, follow these steps:

1. Collect all the information about the problem encountered. (For example, CPU type and speed, Avalue's products model name, hardware & BIOS revision number, other hardware and software used, etc.) Note anything abnormal and list any on-screen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information available.
3. If your product is diagnosed as defective, obtain an RMA (return material authorization) number from your dealer. This allows us to process your good return more quickly.
4. Carefully pack the defective product, a complete Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.
5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

Contents

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

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

- 1 x RSC-IMX51 Micro Module
- 1 x Quick Installation Guide for RSC-IMX51

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

1.3 Document Amendment History

Revision	Date	Comment
1st	Jan. 2012	Initial Release
2nd	Oct. 2012	Update for RSC-IMX51 B1 Version
3rd	Oct. 2013	Update Linux User Guide

1.4 Manual Objectives

This manual describes in detail the Avalue Technology RSC-IMX51 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 interface with RSC-IMX51 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).

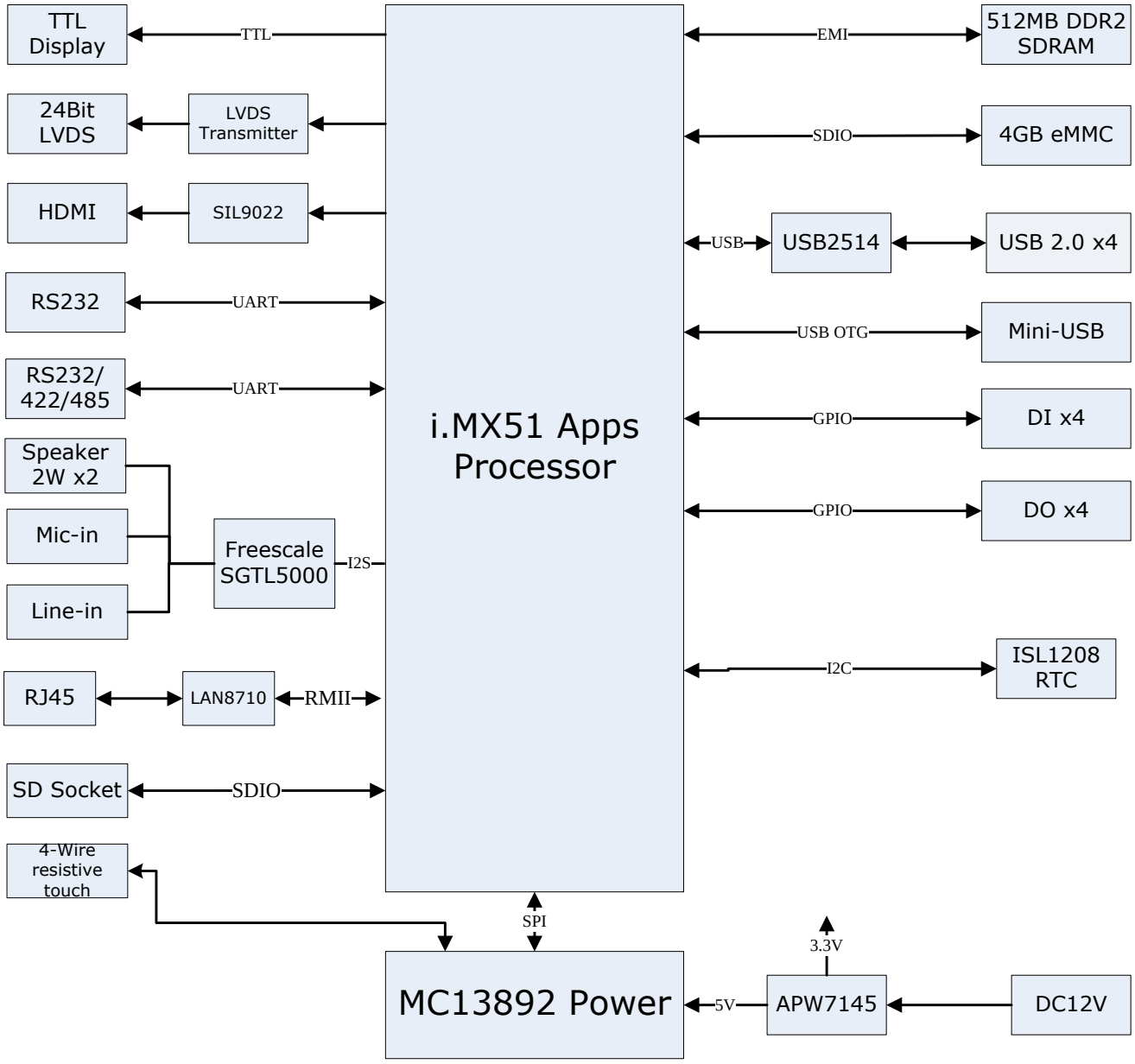
If you have any suggestions or find any errors concerning this manual and want to inform us of these, please contact our Customer Service department with the relevant details.

1.5 System Specifications

System	
CPU	Freescale i.MX515 ARM Cortex-A8 800MHz
PMIC	Freescale MC13892AJVL
Flash	Onboard 4GB eMMC
Memory	512MB DDR2 memory onboard
SSD	SD socket support SDHC up to 32GB
I/O	
MIO	1 x RS-232, 1 x RS-232/ 422/ 485,
USB	4 x USB 2.0
DIO	8-bit General Purpose I/O (DI x4, DO x4)
Display	
Video Processing Unit	IPUv3EX Encoding/ Decoding of MPEG-4, H.263, H.264 Standards
Video interface	18 ~ 24-bits TTL/ LVDS TFT Up to 1280 x 720
TV-Out	HDMI interface (Optional)
Audio	
Audio Codec	Freescale SGTL5000
Audio Interface	Line-in/MIC-in/Speaker-out(2W)
Ethernet	
LAN	Onchip MAC + PHY (SMSC LAN8710)
Ethernet Interface	10 /100 Base-Tx Ethernet Compatible
Mechanical & Environmental	
Power Requirement	+12V DC
Operation Temperature	0 ~ 60°C (32 ~ 140°F)
Operating Humidity	0% ~ 90% Relative Humidity, Non-condensing
Size (L x W)	120mm x 80mm
Weight	0.18lbs (0.08kg)

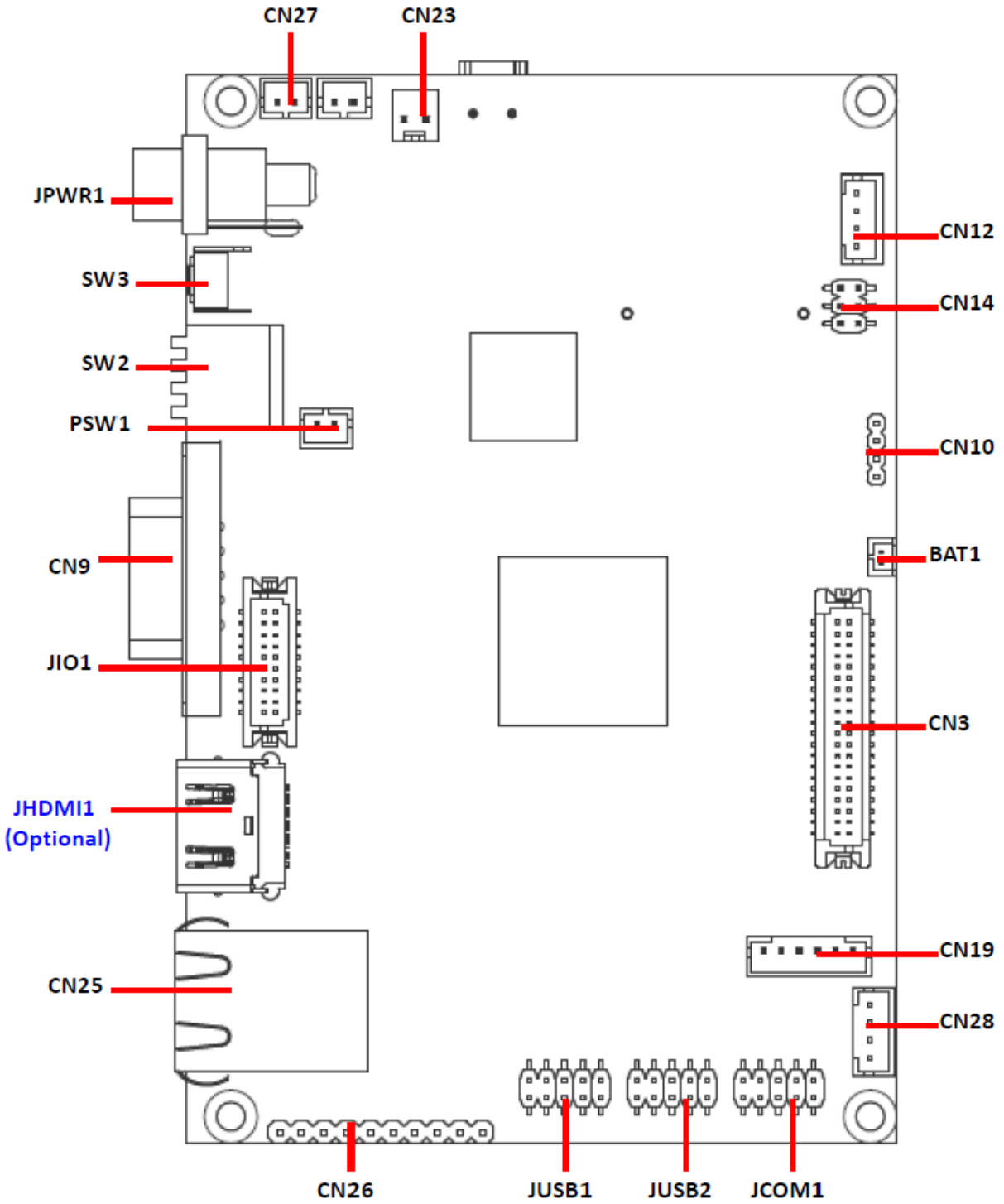
1.6 Architecture Overview – Block Diagram

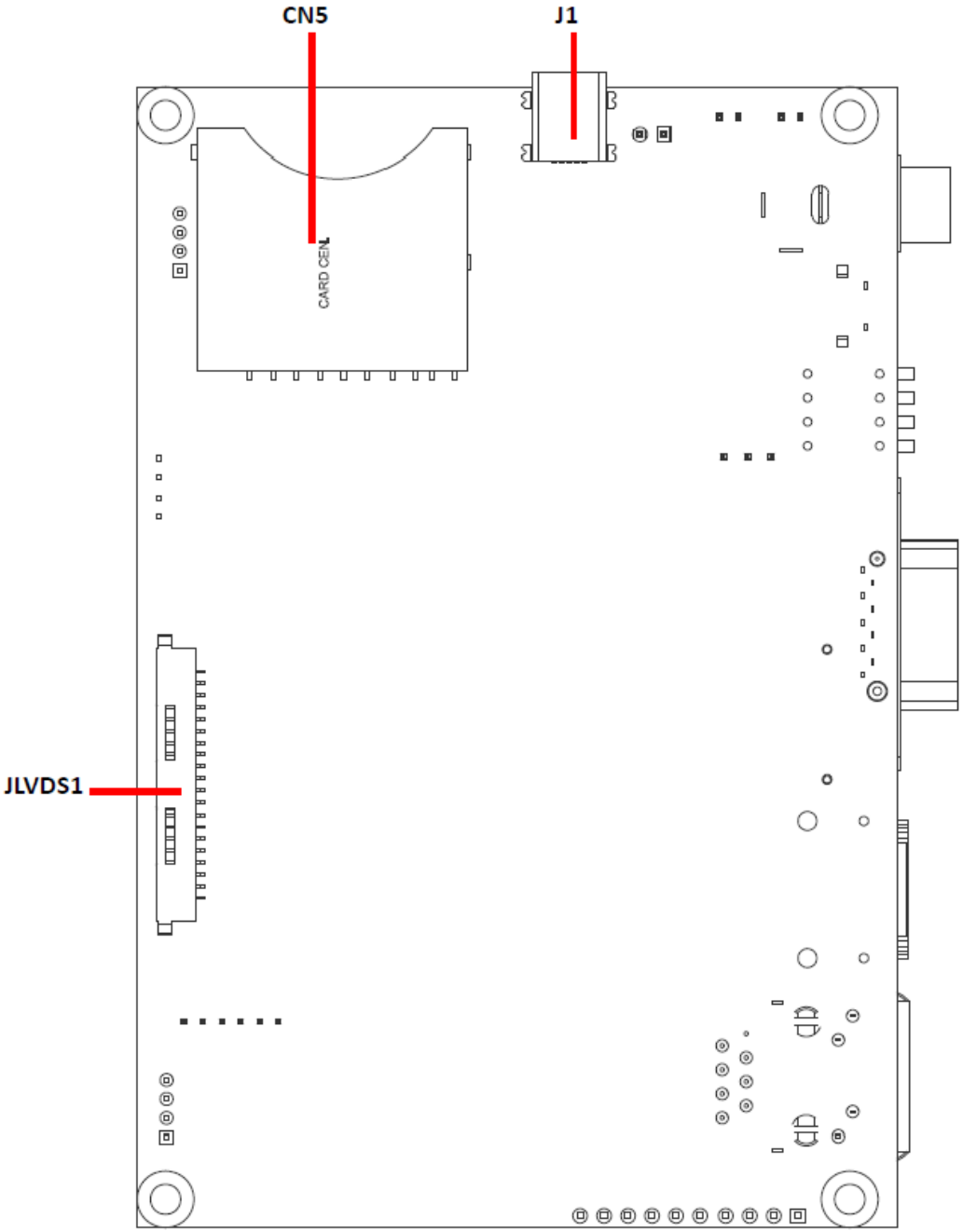
The following block diagram shows the architecture and main components of RSC-IMX51.



2. Hardware Configuration

2.1 Product Overview

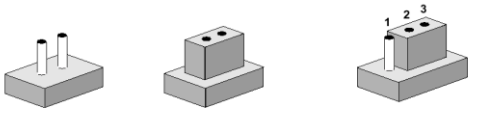



CN5
J1
JLVDS1
CARD CEN

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.



Open Closed Closed 2-3

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

Open Closed Closed 2-3

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

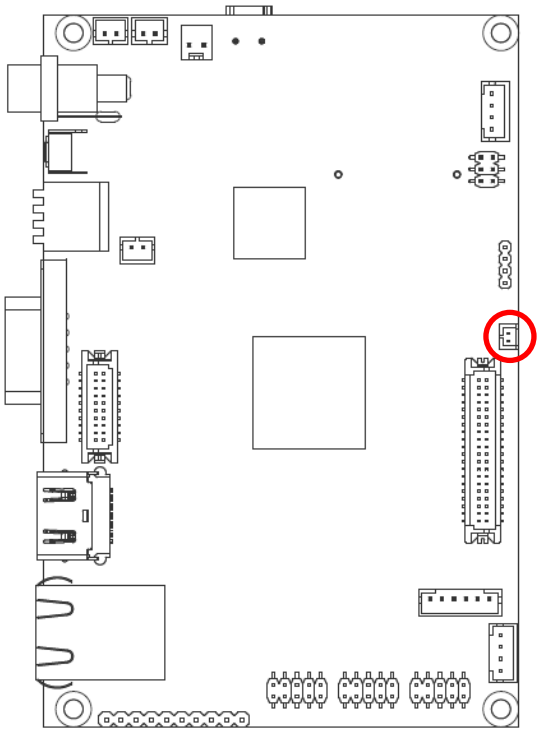
Label	Function	Note
SW2	Boot Mode selector	DIP Switch 4P

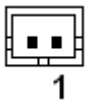
Connectors

Label	Function	Note
BAT1	Battery connector	2 x 1 wafer, pitch 1.25mm
CN3	TTL Panel connector	BB_20x2V_S, pitch 1.25mm
CN5	SD Memory Card Socket	SDCARD_9H, Push/Push Type
CN9	Serial Port 1 connector	DSUB 9-pin, Male
CN10	Touch Panel connector	4 x 1 header, pitch 2.00mm
CN12	Speaker_L/R connector	4 x 1 wafer, pitch 2.00mm
CN14	Line_In, MIC connector	3 x 2 header, pitch 2.00mm
CN19	Inverter Power connector & BL_PWM	6 x 1 wafer, pitch 2.00mm
CN23	12V Input connector	2 x 1 wafer, pitch 2.54mm
CN25	10/100 Mpbs Ethernet connector	RJ45_w/XFMR&LED
CN26	POE Module connector	10 x 1 header, pitch 2.54mm
CN27	Power on LED connector	2 x 1 wafer, pitch 2.00mm
CN28	I2C connector	4 x 1 wafer, pitch 2.00mm
J1	Mini USB connector for Boot/Debug	MINI USB-MAB_5P
JCOM1	Serial Port 2 connector	5 x 2 header, pitch 2.0mm
JHDMI1	HDMI connector	HDMI_19P
JIO1	GPIO Interface connector	10 x 2 wafer, pitch 1.25mm
JLVDS1	LVDS Interface connector	20H, pitch 1.25mm
JPWR1	12V DC-in Jack	DC JACK, 3P, 2.5mm
JUSB1	USB 0&1 connector	5 x 2 header, pitch 2.0mm
JUSB2	USB 2&3 connector	5 x 2 header, pitch 2.0mm
PSW1	Power on/off Switch connector	2 x 1 wafer, pitch 2.00mm
SW3	Reset Switch	TACT Switch 4P 90D

2.3 Setting Jumpers & Connectors

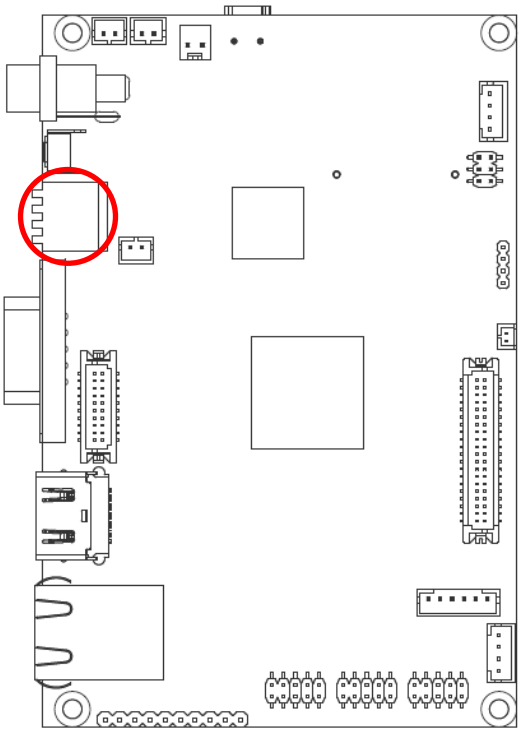
2.3.1 Battery connector (BAT1)



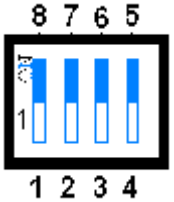


PIN	Signal
1	LICELL
2	GND

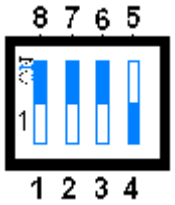
2.3.2 Boot Mode selector (SW2)



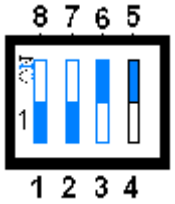
Boot from onboard SD



Boot from SD socket



USB OTG mode (Reflash onboard SD only)



Signal	PIN	PIN	Signal
BMOD1	1	5	+V2D775_BOOT
BMOD0	2	6	
BT_SRC[1]	3	7	+V1D8_DIG1
BT_SRC[0]	4	8	

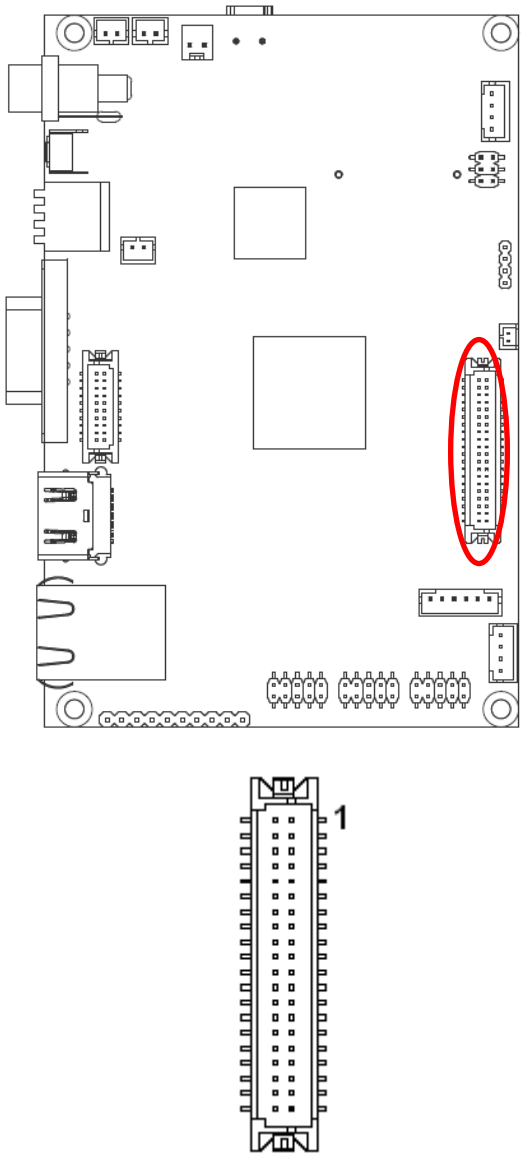
Please note:

DIP Switch setting:

0=Off, 1=On

When **Position4** is switched On, the system is forced to power On as soon as power is applied. Switch to Off mode for normal operation.

2.3.3 TTL Panel connector (CN3)



Signal	PIN	PIN	Signal
+5V	2	1	+5V
GND	4	3	GND
+3.3V	6	5	+3.3V
GND	8	7	N/C
B1	10	9	B0
B3	12	11	B2
B5	14	13	B4
B7	16	15	B6
G1	18	17	G0
G3	20	19	G2
G5	22	21	G4
G7	24	23	G6
R1	26	25	R0
R3	28	27	R2
R5	30	29	R4
R7	32	31	R6
GND	34	33	GND
VSYNC	36	35	SHCLK
HSYNC	38	37	LDEMOD
N/C	40	39	ENBLK

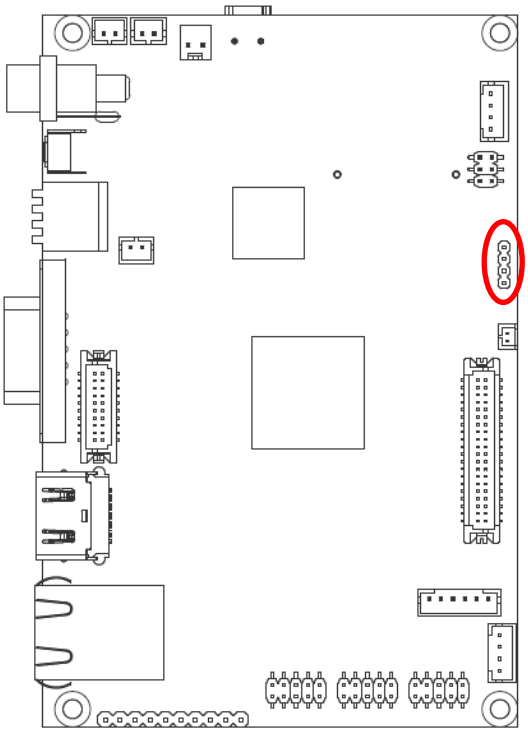
2.3.3.1 Signal Description – TFT Panel Connector (CN3)

Signal	Description
B [0:7]G[0:7]R[0:7]	Flat panel data output for 24 bit TFT flat panels. The flat panel data and control outputs are all on-board controlled for secure power-on/off sequencing
SHCLK	Shift Clock. Pixel clock for flat panel data
HSYNC	Flat panel equivalent of horizontal synchronization
VSYNC	Flat panel equivalent of vertical synchronization
LDEMOD	Multipurpose signal, function depends on panel type. May be used as AC drive control signal or as BLANK# or Display Enable signal
ENBKL	Enable backlight signal. This signal is controlled as a part of the panel power sequencing

2.3.3.2 TTL cable wiring table

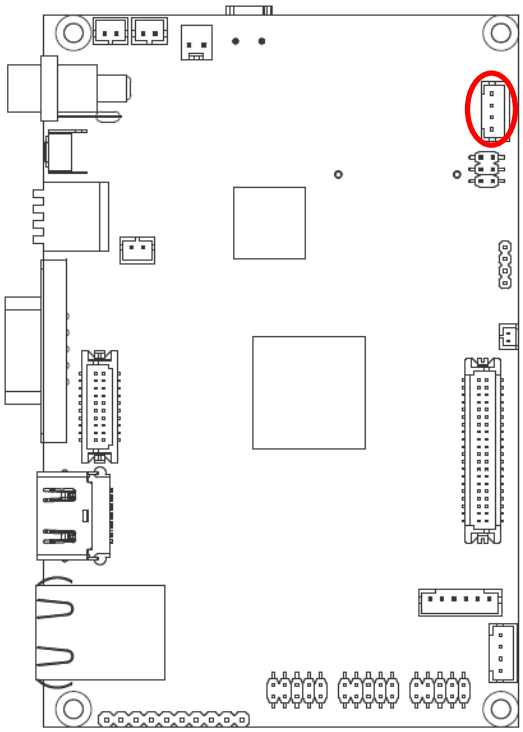
Signal	18-bit TFT	24bit TFT
B0		B0
B1		B1
B2	B0	B2
B3	B1	B3
B4	B2	B4
B5	B3	B5
B6	B4	B6
B7	B5	B7
G0		G0
G1		G1
G2	G0	G2
G3	G1	G3
G4	G2	G4
G5	G3	G5
G6	G4	G6
G7	G5	G7
R0		R0
R1		R1
R2	R0	R2
R3	R1	R3
R4	R2	R4
R5	R3	R5
R6	R4	R6
R7	R5	R7

2.3.4 Touch Panel connector (CN10)



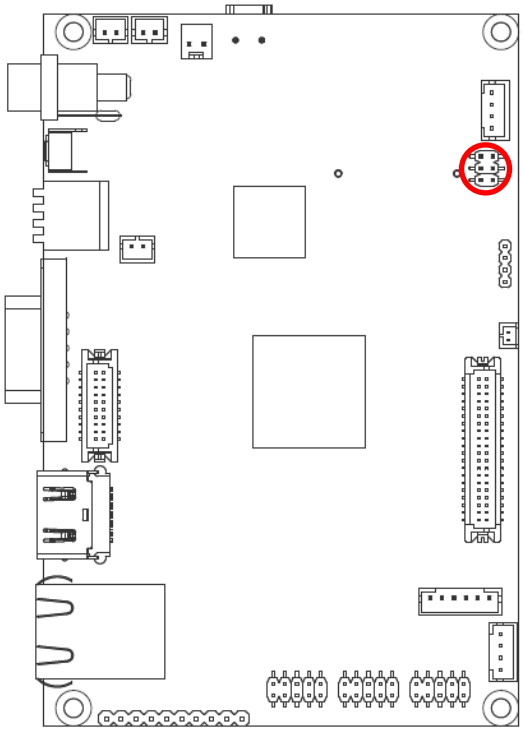
PIN	Signal
4	BOT
3	LEFT
2	TOP
1	RIGHT

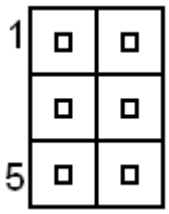
2.3.5 Speaker L/R connector (CN12)



Signal	PIN
SPEAKER_LEFT_P	4
SPEAKER_LEFT_N	3
SPEAKER_RIGHT_P	2
SPEAKER_RIGHT_N	1

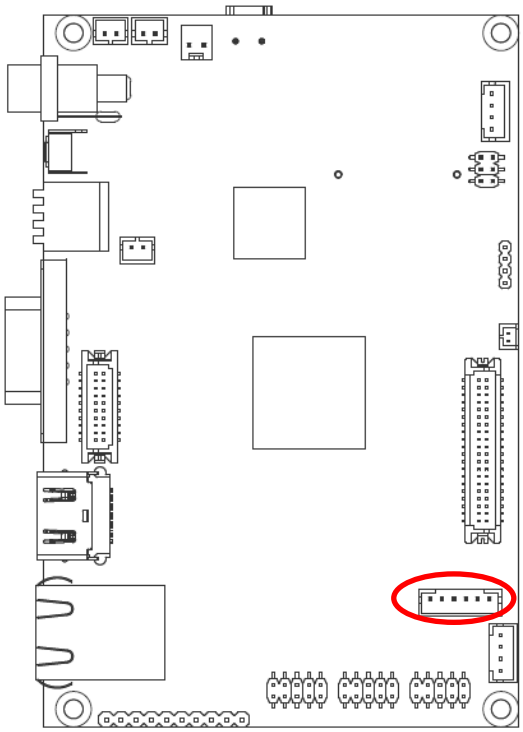
2.3.6 Line_In, MIC connector (CN14)

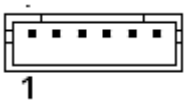




Signal	PIN	PIN	Signal
GND	1	2	GND
LINEIN_R	3	4	LINEIN_L
MIC IN	5	6	MIC IN

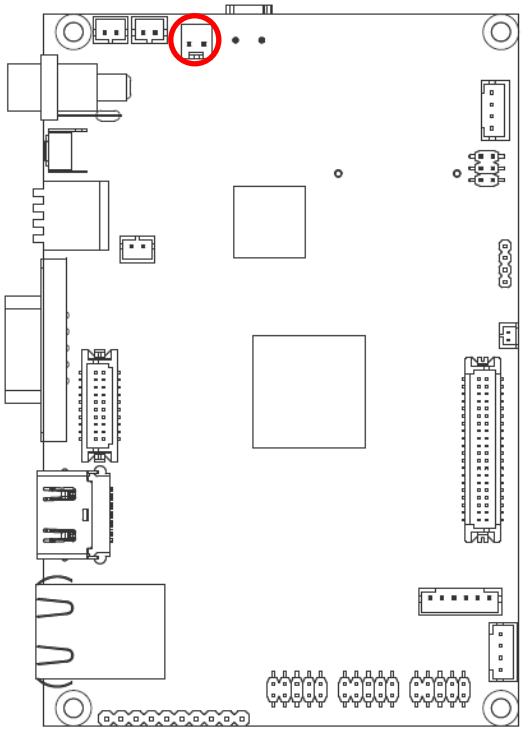
2.3.7 Inverter Power connector & BL_PWM (CN19)

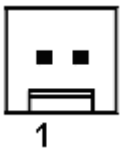




PIN	Signal
1	+12V
2	GND
3	DISPEN
4	W
5	V5BL
6	PWM BL

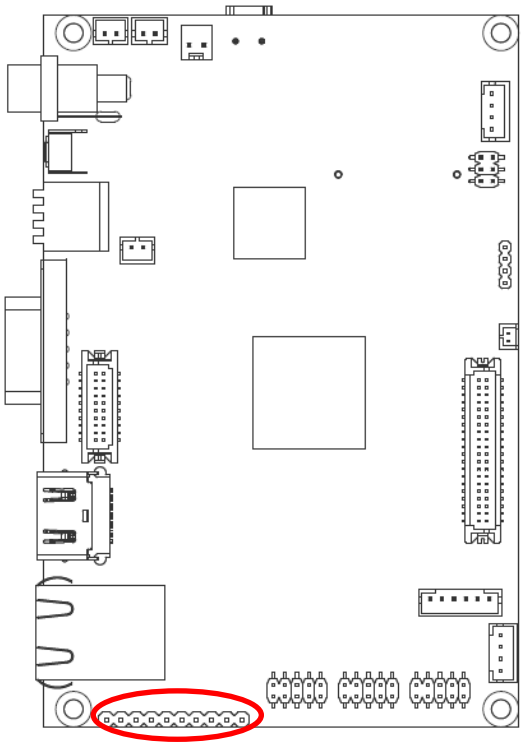
2.3.8 12V Input connector (CN23)

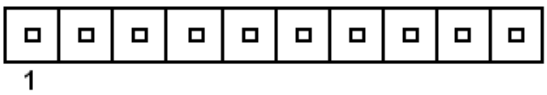




PIN	Signal
1	PWR_IN
2	GND

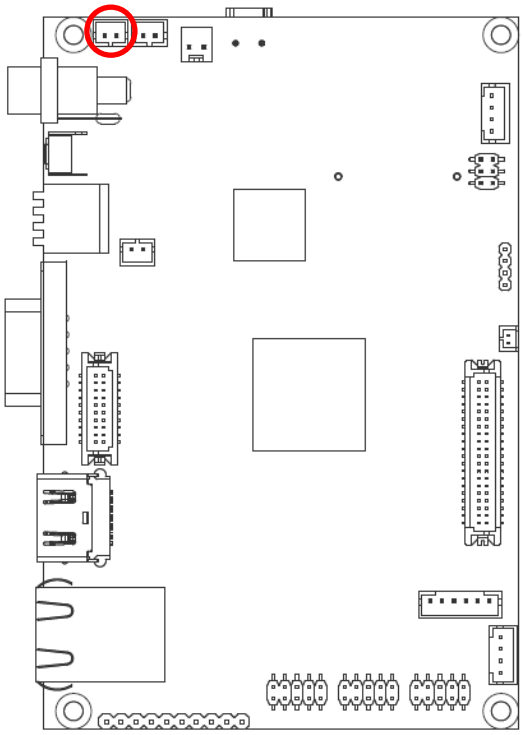
2.3.9 POE Module connector (CN26)

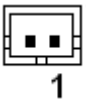




PIN	Signal
1	LED2_ANODE
2	LED2_CATHODE
3	LED1_CATHODE
4	LED1_ANODE
5	CENTER
6	8710TXP
7	8710TXN
8	GND
9	8710RXP
10	8710RXN

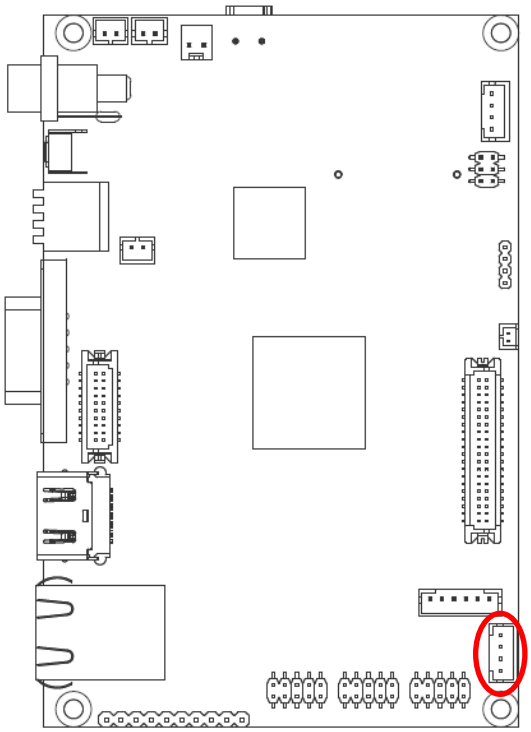
2.3.10 Power on LED connector (CN27)





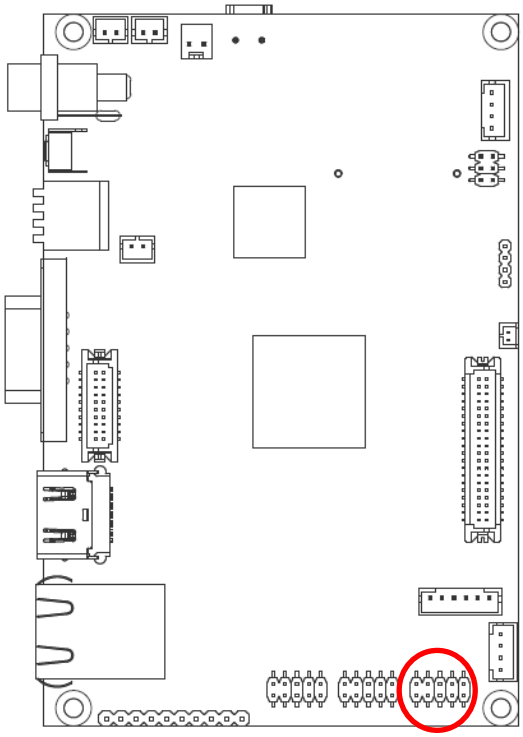
PIN	Signal
1	+3V
2	GND

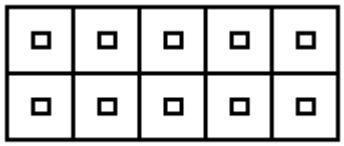
2.3.11 I2C connector (CN28)



PIN	Signal
4	GND
3	I2C SDA
2	I2C_SCL
1	+3V

2.3.12 Serial Port 2 connector (JCOM1)





1

Signal	PIN	PIN	Signal
485TX-	1	2	NRX3
NTXS3	3	4	422RX-
GND	5	6	NC
NRTS3	7	8	NCTS3
NC	9	10	GND

RS232 mode

Signal	Pin	Pin	Signal
NC	1	2	RX
TX	3	4	NC
GND	5	6	NC
RTS	7	8	CTS
NC	9	10	GND

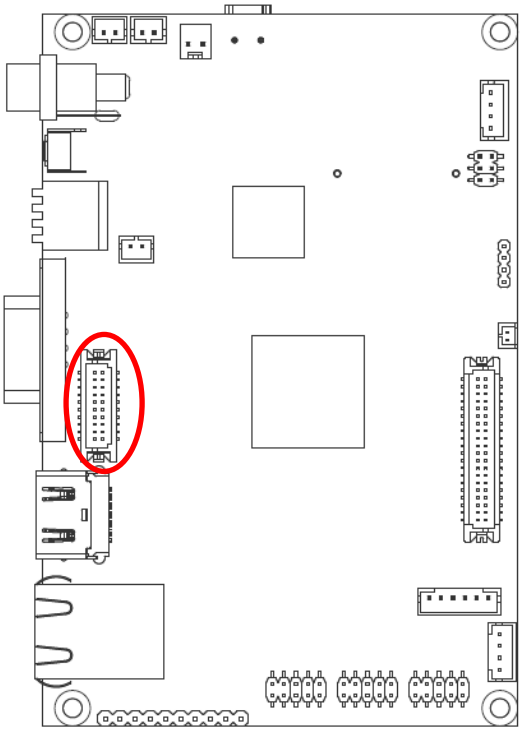
RS422 mode

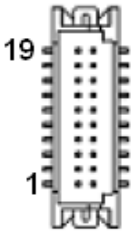
Signal	Pin	Pin	Signal
TX-	1	2	RX+
TX+	3	4	RX-
GND	5	6	NC
NC	7	8	NC
NC	9	10	GND

RS485 mode

Signal	Pin	Pin	Signal
DATA-	1	2	NC
DATA+	3	4	NC
GND	5	6	NC
NC	7	8	NC
NC	9	10	GND

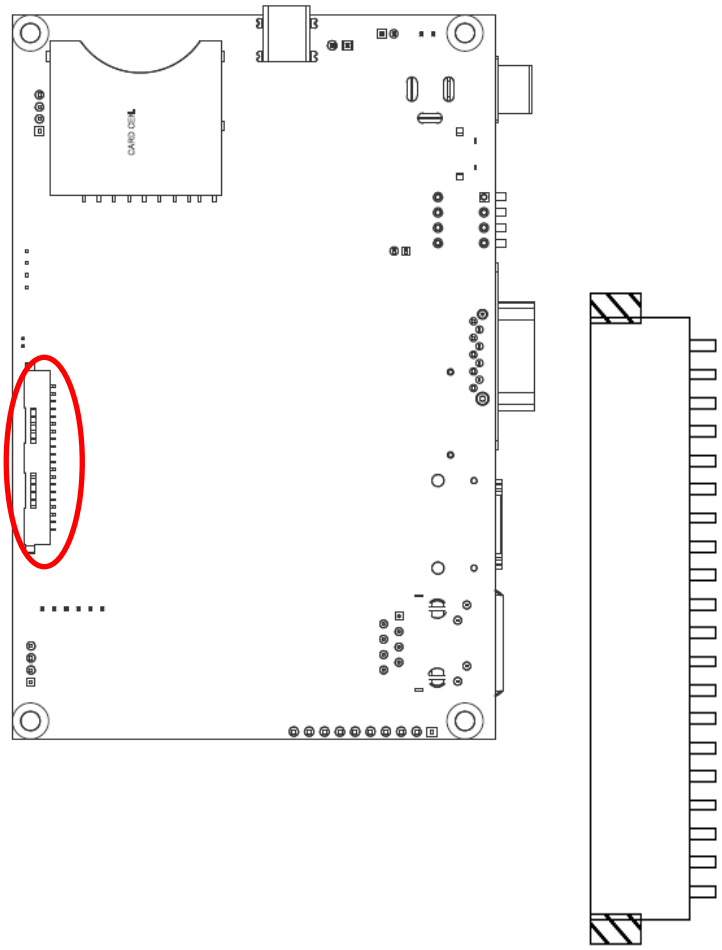
2.3.13 GPIO Interface connector (JIO1)





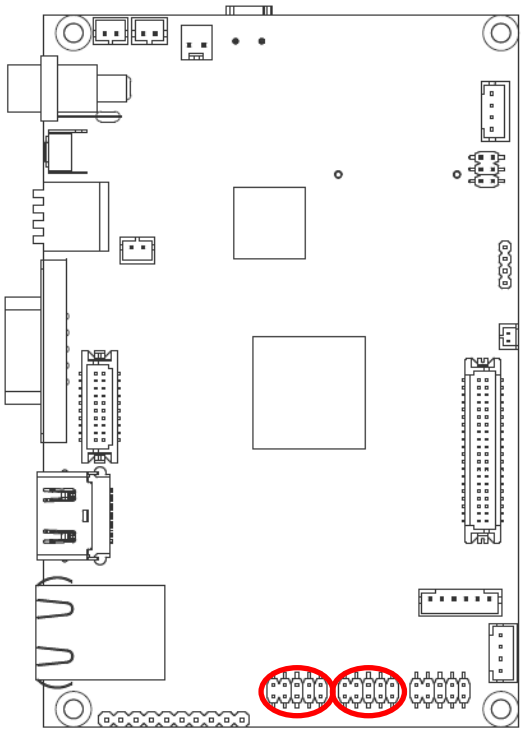
Signal	PIN	PIN	Signal
GND	19	20	GND
GPO2_10_R	17	18	SPI1_MOSI
GPO2_11_R	15	16	SPI1_MISO
GPO2_12_R	13	14	SPI1_SCLK
GPO2_13_R	11	12	SPI1_SS1_B
+3V	9	10	NC
GPI1_31	7	8	NC
GPI3_15	5	6	NC
GPI3_6	3	4	GND
GPI3_14	1	2	+V2D775_BOOT

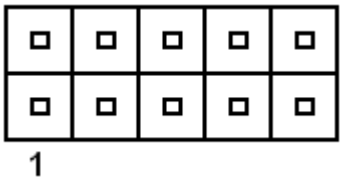
2.3.14 LVDS Interface connector (JLVDS1)



PIN	Signal
1	+3.3V
2	+3.3V
3	EDID_CLK
4	EDID_DAT
5	GND
6	TxOUT-0N
7	TxOUT-0P
8	GND
9	TxOUT-1N
10	TxOUT-1P
11	GND
12	TxOUT-2N
13	TxOUT-2P
14	GND
15	TxOUT-CLKN
16	TxOUT-CLKP
17	TxOUT-3N
18	TxOUT-3P
19	+12V
20	PWM_BL

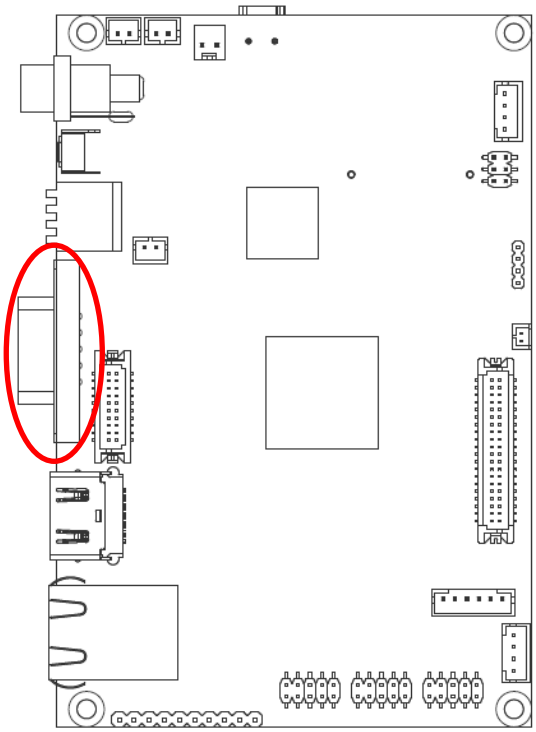
2.3.15 USB 0&1 / 2&3 connector (JUSB1/JUSB2)

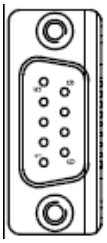




Signal	PIN	PIN	Signal
+5V	1	2	GND
D0-	3	4	GND
D0+	5	6	D1+
GND	7	8	D1-
GND	9	10	+5V

2.3.16 Serial Port 1 connector (CN9)





Signal	PIN	PIN	Signal
NC	1	2	RX1
TX1	3	4	NC
GND	5	6	NC
RTS	7	8	CTS
NC	9		

3. WinCE User Guide

3.1 Installing Windows Embedded Tools and RSC-IMX51 BSP

To create the complete RSC-IMX51 development environment, install the Microsoft Windows Embedded CE 6.0 development tools and the RSC-IMX51 BSP as follows:

1) Install Visual Studio 2005 and the Windows Embedded CE 6.0 Platform Builder plugin on Windows XP from the installation discs. The following must be installed in order to create a Windows Embedded CE 6.0 development environment for RSC-IMX51 WinCE 6.0 BSP:
   - Visual Studio 2005
   - Visual Studio 2005 SP1
   - Visual Studio 2005 SP1 Update for Vista (if applicable)
   - Windows Embedded CE 6.0 Platform Builder
   - Windows Embedded CE 6.0 SP1 (required if PB 6.0 Tools have been installed)
   - Windows Embedded CE 6.0 R2
   - Windows Embedded CE 6.0 R3
   - Windows Embedded CE 6.0 R3 Update Rollup
   - Windows Embedded CE 6.0 Cumulative Product Update Rollup Package (through 12/31/2010)
   - Windows Embedded CE 6.0 Monthly Update January 2011
   - Windows Embedded CE 6.0 Monthly Update February 2011

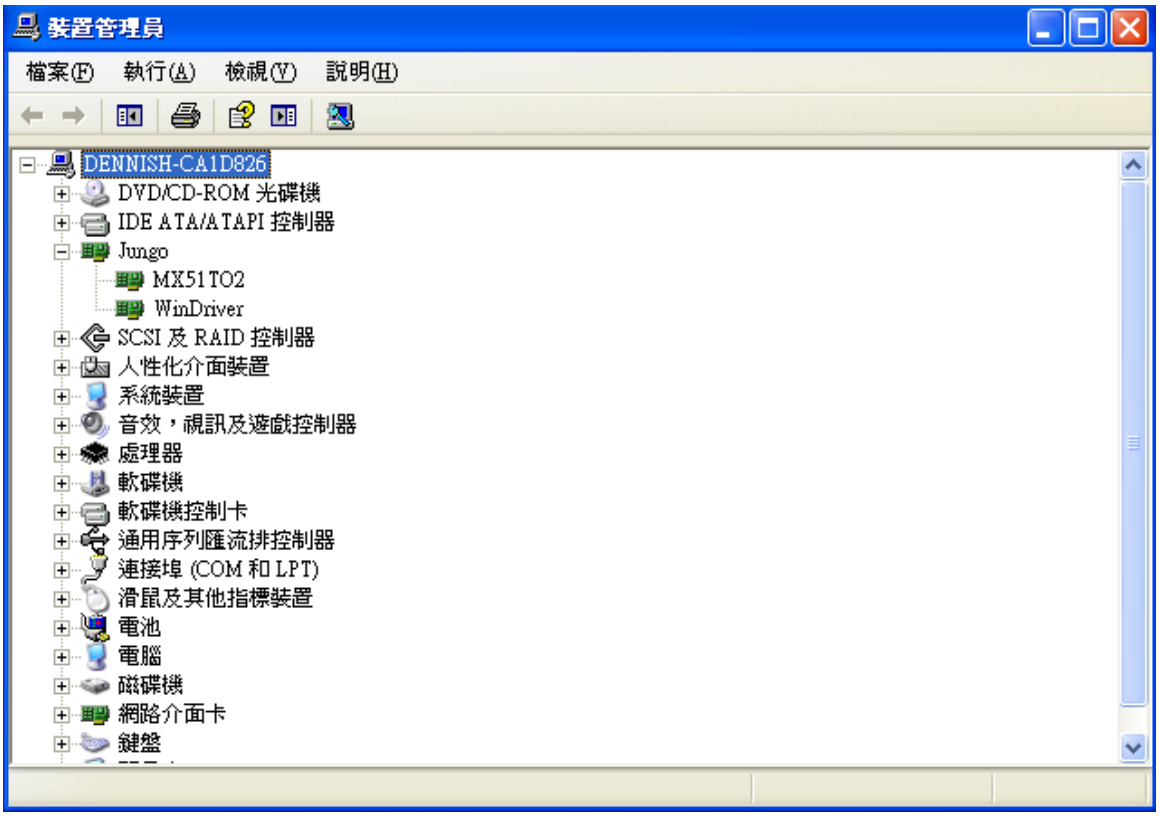
2) Follow steps below to install RSC-IMX51 binary BSP and generate WinCE image
   - Unzip RSC-iMX51_BinaryBSP.zip
   - Copy BSP folder RSC_iMX51_BIN to C:\WINCE600\PLATFORM\
   - Copy folder SUPPORT to C:\WINCE600\
   - Copy OS design project folder RSC_iMX51_BIN_Demo to C:\WINCE600\OSDesigns\
   - In Platform Builder 6.0 which is bundled with VS2005, implement File => Open => Project/Solution to open OS design project file RSC_iMX51_BIN_Demo.sln
   - Confirm the demo project is built with release mode first. If not, please change it.
   - Click Project => Properties to go to Property Pages.
   - In Build Options, uncheck the checkbox Item "Enable KITL" and check the item "Enable kernel debugger"
   - After finishing settings, implement Build => Build Solution to make a new CE image NK.bin

3.2 ATK Tool Installation & Setup

The Advanced ToolKit (ATK) is a graphical user interface application for use in an i.MX platform for development and validation.

To install the standard ATK tool, follow these steps:

- Extract the ATK package and place it in a temporary folder.
- From the top level folder, execute FSL_ATK_TOOL_WINS_STD_INSTALL_1_70.exe
  You do not need to reboot the PC after executing this step
- From the Start Menu, point to Programs, then to Advanced ToolKit, and then click Advanced ToolKit.
  The Advanced ToolKit configuration screen is displayed. At this point the USB Drivers are installed. Please set SW2 of RSC-IMX51 in USB OTG mode and power on it. To verify that the USB driver is ready to use, navigate to the Device Manager. An item labeled Jungo should appear in the devices list, similar to the one shown below.



3.3 Preparing for Downloading and Debugging

The target and development workstation must be properly configured and initialized before OS images can be downloaded and executed. This section discusses the steps required to prepare the target and development workstation so that the Platform Builder can be used to download and debug images on the target.

Serial debug messages are used by the boot loader and OS images to report status and error information. Additionally, the boot loader uses serial input to allow for user interaction. This section describes the configuration of the desktop workstation and Freescale BSP to support serial debug messages.

1) Desktop Workstation Serial Debug Port

Any terminal emulation application can be used to display messages sent from the serial port of the target. Configure the terminal application with the following communications parameters:

- Bits per second—115200
- Data bits—8
- Parity—None
- Stop bits—1
- Flow control—None

2) Target Serial Debug Port

Connect a Cross-Over serial cable between CN9 of RSC-IMX51 and RS232 interface of the development workstation.

3.4 Program XLDR and EBOOT in on-board flash Using ATK Tool

Follow these steps to program XLDR and EBOOT in onboard SD card using the ATK Tool:

- Connect USB cable from mini-USB connector(J1) of RSC-IMX51 to your development workstation.
- Set SW2 of RSC-IMX51 in USB OTG mode and power on it.
- Launch the ATK Tool on development workstation.
- Select i.MX51_TO2 in i.MX CPU.
- Select DDR in Device Memory Initial.
- Select USB in Communication Channel.
- Click Next > Flash Tool > Go.
- Select Program in Operation type.
- Select MMC/SD in Flash Model.
- Set Operation Settings as 0x00000400.
- Specify the XLDR.nb0 in Image.
- Press the Program button to program XLDR in SD/MMC card
- Set Operation Settings as 0x00020000.
- Specify the EBOOT.nb0 in Image.
- Press the Program button to program EBOOT in SD/MMC card.
- Power off RSC-IMX51
- Set SW2 to boot from onboard flash.
- Power on RSC-IMX51. The EBOOT menu should be seen on the serial terminal.
- Exit the ATK Tool.

3.5 Initialize EBOOT Network Configuration

To initialize EBOOT network configuration, follow these steps:

- Follow the steps in Section 3.4 “Program XLDR and EBOOT in on-board flash Using ATK Tool,” to execute EBOOT on the target.
- Quickly switch over to the terminal emulation application and wait for the debug message, Press [ENTER] to download now or [SPACE] to cancel to appear.
- Press the space bar to bring up the EBOOT configuration menu.
- Press K to disable KITL Enable mode
- Press 5 to select Boot device to “NK from SD/MMC”
- Press 6 to specify an Ethernet MAC address by selecting the MAC Address menu option and then entering the 12-digit hexadecimal MAC address delimited by periods.
- Press enter to return to the EBOOT configuration menu. The specified MAC address should now be displayed in the EBOOT menu.
- By default, DHCP is enabled and there is no need to specify a static IP or static subnet mask. If static networking parameters are needed, use the IP Address and Subnet Mask menu options.
- After the desired network configuration appears in the EBOOT menu, select the Save configuration menu option to save the configuration to nonvolatile memory.

3.6 Configuring Ethernet Connection for Downloading and Debugging

To configure an Ethernet connection that can be used for downloading and debugging images, follow these steps:

- From the Platform Builder Target menu, choose Connectivity Options.
- Choose Kernel Service Map.
- In the Target Device box, choose a target device.
- In the Download box, choose Ethernet as the download service.
- Launch EBOOT on the target. After EBOOT initialization completes, BOOTME messages begin to appear on the serial debug output. Observe the device name created by EBOOT on the serial debug output.
- To the right of the Download box, choose Settings. The device name of the target should appear in the Active Devices box.
- Select the target from the Active Devices box, and then choose OK.
- In the Transport box, choose Ethernet as a kernel transport.
- To the right of the Transport box, choose Settings.
- Check the box next to Use device name from bootloader and then choose OK.
- If the run-time image includes support for the kernel debugger stub, KdStub, from the Debugger box, choose KdStub. If the run-time image does not include support for a debugger, from the Debugger box, choose None.
- Choose Core Service Settings. To instruct the Platform Builder to download a run-time image each time the Platform Builder connects with the target device, under Download Image, choose Always.
- Select Enable KITL on device boot.
- Select Clear memory on soft reset.
- Select Enable access to desktop files.
- Choose Apply.
- Choose Close.
- Click Target => Attach Device, then RSC-IMX51 will start to download NK.bin from WinCE6.0 platform builder automatically.

3.7 Using cfimager Utility to Flash SD/MMC Card

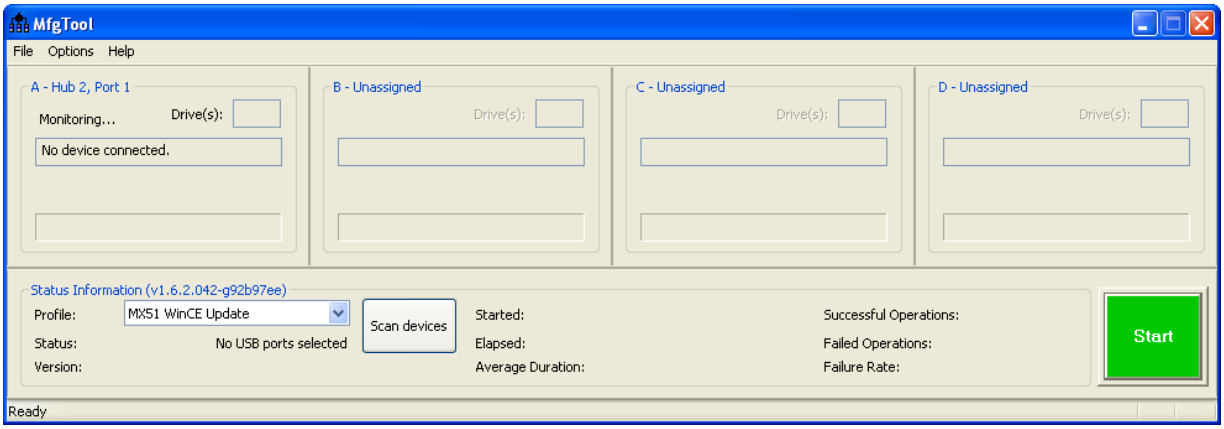
cfimager.exe is a host PC tool used to flash boot images (xldr.nb0, eboot.nb0, nk.nb0) and create a FAT partition on SD/MMC cards for i.MX51. Users can avoid programming the card on the device. The tool for i.MX51 can only flash *.nb0 files. A card reader on the PC is required. The utility, accompanied by a readme file, is provided under <%WINCEROOT%>\SUPPORT\TOOL\COMMON\CFIMAGER. Any data on the card may be lost by flashing it with cfimager. Follow these steps to flash the card:

- To flash the XLDR, use the following command in a build window: cfimager -f xldr.nb0 -d <card reader drive letter without colon> -imx51
- To flash the bootloader, use the following command: cfimager -f eboot.nb0 -d <card reader drive letter without colon> -imx51
- To flash the OS image, use the following command: cfimager -f nk.nb0 -d <card reader drive letter without colon> -imx51 Flashing a debug nk.nb0 image does not work because it is too big.
- To add the FAT partition, add -a option to any of the above command lines. When creating a FAT partition, any previous data in any existing FAT partition is erased.

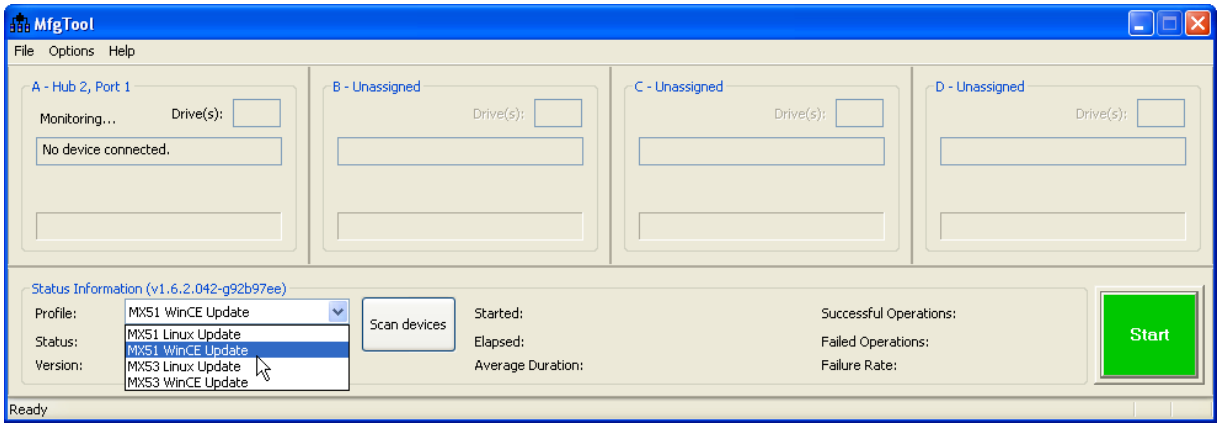
3.8 Using mfgtool to Flash eMMC onboard

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.

1) Unzip the tool package to your local directory, say: D:\ Mfgtools-Rel-1.6.2.042\, and enter it, you will find an .exe file named MfgTool.exe. Without any installation step, you can run it directly. So, just run it. You should be able to see below user interface.

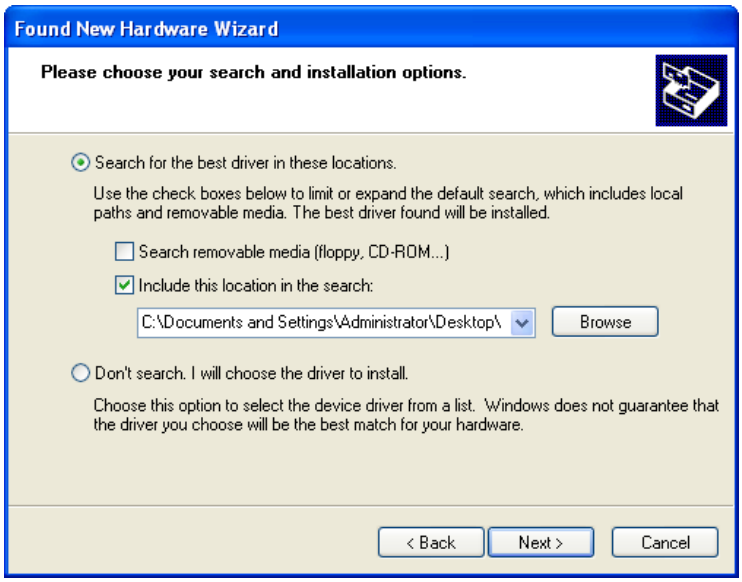


2) The user interface appears to have nothing to do since there is nothing is selected, so let's configure something necessary.
Click Profile drop menu, you will find a number of platforms, choose the right one according to your platform



3) Now you need to connect your board to one USB port of your PC. Please make sure you have set RSC-IMX51 SW2 to USB OTG mode before powering on the board. Please refer to platform specific document to see how to set the board to bootstrap mode.
If your device is in below device list, then you will find a HID mode device appears on your PC when you connect your board to one USB port and power on it. Please jump to Step 4.
You will find below popup window when you connect your board to one USB port and power on it.

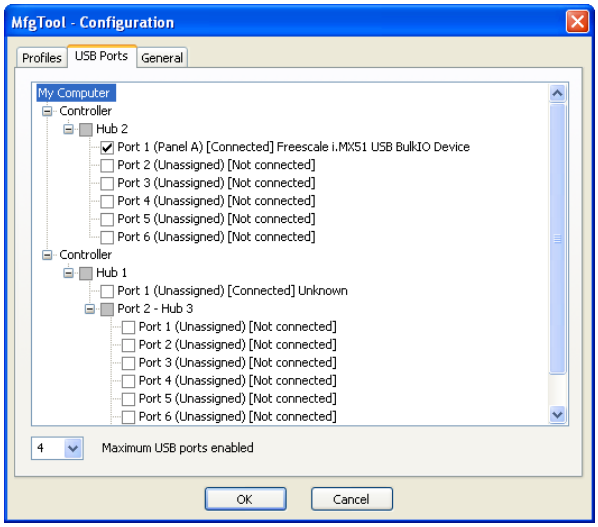
Choose “Install from a list or specific location” option and click “Next”.



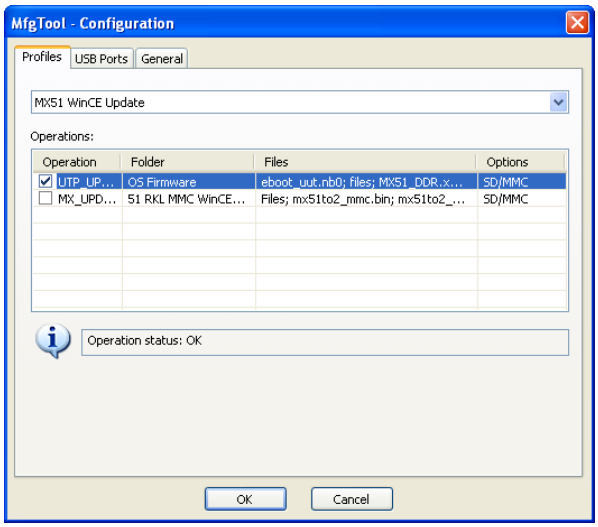
Input your driver location:
YourDiskVolume:\mfgtools-rel\Drivers\iMX_BulkIO_Driver and click “Next”.

Click “Finish” to finish driver installation.

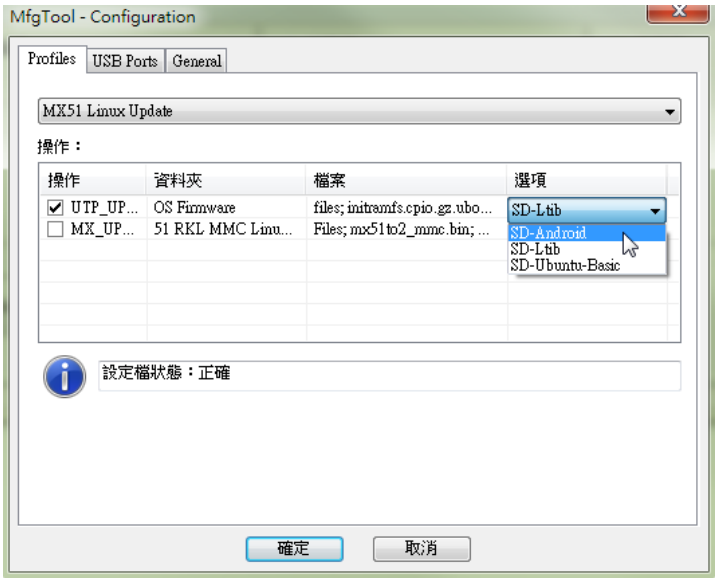
4) Copy update WinCE image (nk.nb0) to "Mfgtools-Rel-1.6.2.042\Profiles\MX51 WinCE Update\OS Firmware\files"
   Copy update Android image to "Mfgtools-Rel-1.6.2.042\Profiles\MX51 Linux Update\OS Firmware\files\android"
   Copy update LTIB image to "Mfgtools-Rel-1.6.2.042\Profiles\MX51 Linux Update\OS Firmware\files\ltib"
   Copy update Ubuntu image to "Mfgtools-Rel-1.6.2.042\Profiles\MX51 Linux Update\OS Firmware\files\ubuntu-basic"
5) Click the "Scan" button on the main dialog to auto scan the devices connected to your PC.
   The other way is to click "Options" menu at upper-right corner, and further click configuration item, you can find below pop-up window. Click "USB Ports" tab, you can view all the USB port. Choose the one to which your device is connected and click "OK" to close the window.



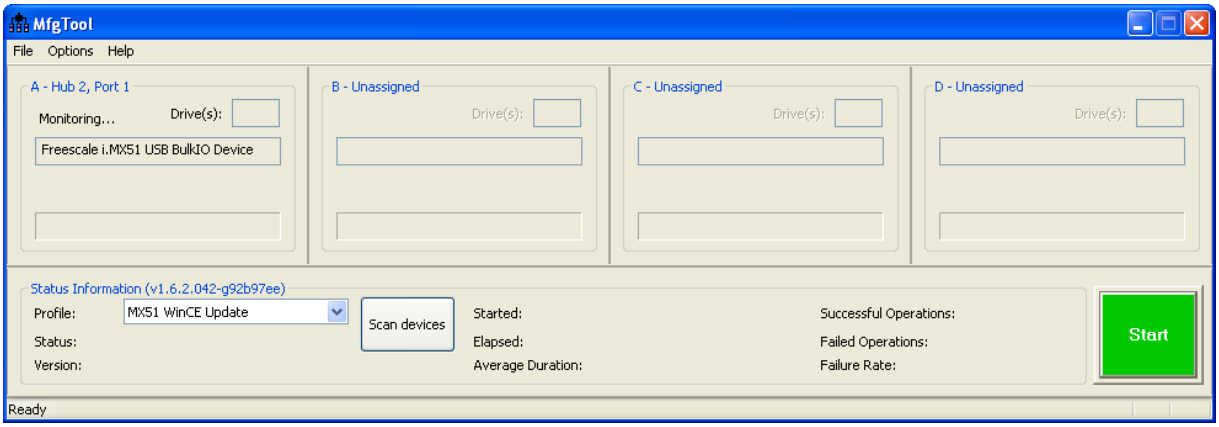
Also you will have to choose which OS you want to flash into i.MX51 platform.
Here is setting for WinCE6 update



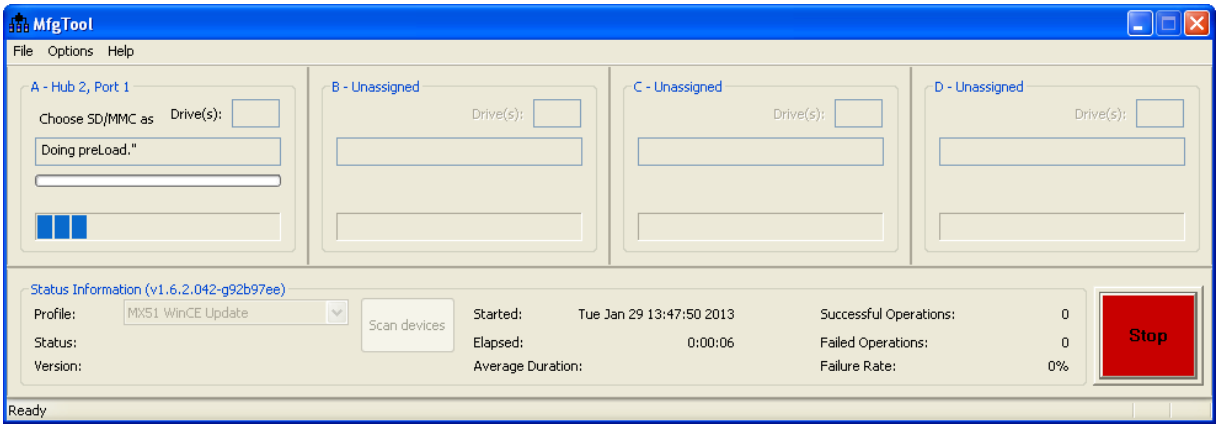
Here is setting for Linux (LTIB/Android/Ubuntu) update



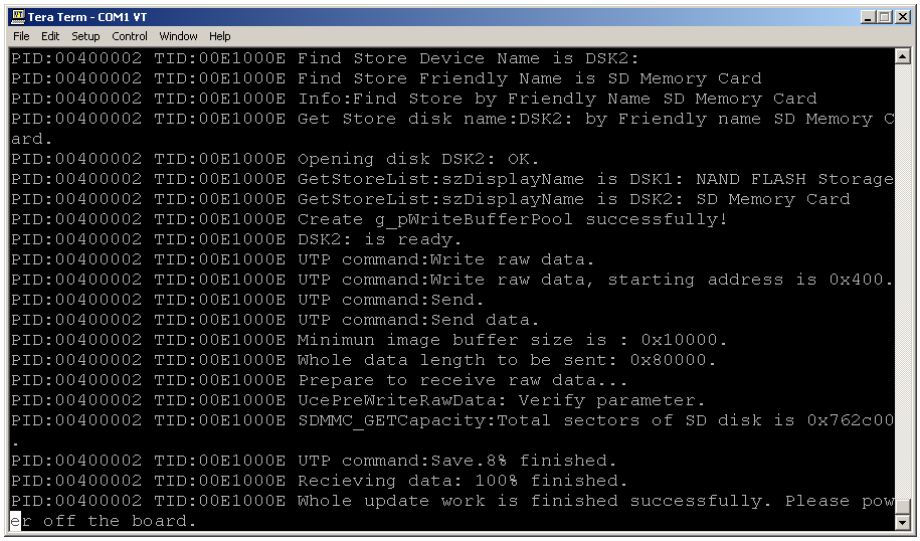
Now you can find the tool is ready to do a demo work. Click “Start” button. If you have a terminal tool to monitor the debug serial port of your board, it is suggested to open it. You can get more information from it.



6) The process is ongoing.



You can find some information from the terminal.



3.9 Programming Guide for GPIO Driver

The RSC-iMX51 Windows CE image had included the onboard GPIO and watchdog timer driver API for application programming. The function that API will provide are list as below, refer each function's description to get more detail information.

1) **Digital I/O Port**

- **API_LibInit()**

BOOL API_LibInit();

Return Value :

TRUE indicates success, FALSE indicates failure. The error numbers are listed in Appendix A.

Remarks :

This function assigns JIO1 pin 1(GPIO3_14), 3(GPIO3_6), 5(GPIO3_15) and 7(GPIO1_31) of RSC-iMX51 as input pins, pin 11(GPIO2_13), 13(GPIO2_12), 15(GPIO2_11) and 17(GPIO2_10) as output pins and then returns TRUE if success.

- **GPIO_BIT_OUTPUT**

BOOL GPIO_BIT_OUTPUT(unsigned int dwport, unsigned int dwbit, unsigned int dwval);

Parameters:

dwport: [in] Port number of GPIO.

dwbit: [in] Bit number of the selected port dwport.

dwval: [in] Control the current state(high/low) of selected pin on selected output GPIO bit.

0: Low, 1: High.

Return Value :

TRUE indicates success, FALSE indicates failure.

Remarks :

This function will control all 4 pins of selected output port turning on and off.

For example:

GPIO_BIT_OUTPUT(2, 10, 1) is called to turn on GPIO2_10 which is the output pin 17 of JIO1 on RSC-iMX51.

- **GPIO_BIT_INPUT**

int GPIO_BIT_INPUT(unsigned int dwport, unsigned int dwbit);

Parameters:

dwport: [in] Port number of GPIO.

dwbit: [in] Bit number of the selected port #dwport.

Return Value :

Current status (high/low) of the selected pin is passed to. 0: Low, 1: High

Remarks:

The current state of a selected pin is obtained by calling this function.

For example:

int dwVal;

dwVal=GPIO_BIT_INPUT(1, 31) is called to get the current state of GPIO1_31 which is the input pin 7 of JIO1 on RSC-iMX51.

Watchdog Function:

Watchdog timer of RSC-iMX51 consists a time-out counter with time-out periods from 0.5 seconds up to 128 seconds. The APIs are used to start the watchdog timer (WDT) with pre-defined timing (mini-seconds after).

● **WDT_START**

void WDT_START(DWORD timemsec);

Parameters:

timemsec: [in] WDT time-out value range from 500 msec to 128000 msec with time resolution of 500 msec

Return Value:

None

Remarks:

This function enables WDT to start counting down from non-zero timemsec and then time-out occurs when WDT counts to zero which causes system reboot. Calling this function before time-out happening causes non-zero timemsec to be reloaded to the watchdog counter and count down resumes.

For example:

WDT_START(500) means watchdog time-out happens after 500 mini-seconds.

Software reset:

void IMX51_REBOOT(void);

Parameters:

None

Return Value:

None

Remarks:

This function enables software reset in Windows CE.

4. Linux User Guide

Note: Installation procedures and screen shots in this section are for your reference and may not be exactly the same as shown on your screen.

4.1 Setting up a Linux host for building U-boot & Kernel Image file

We support building using Ubuntu 10.04 installed from the 64 bit Desktop Ubuntu install cd. Other versions of Ubuntu are 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 aptitude -y install gettext libgtk2.0-dev rpm bison m4 libfreetype6-dev
sudo aptitude -y install libdbus-glib-1-dev liborbit2-dev intltool
sudo aptitude -y install ccache libncurses-dev zlib1g zlib1g-dev gcc g++ libtool
sudo aptitude -y install uuid-dev liblzo2-dev
sudo aptitude -y install tcl
sudo aptitude -y install ia32-libs libc6-dev-i386 lib32z1
sudo aptitude -y install gparted emacs22-nox openssh-server
sudo aptitude -y install nfs-common nfs-kernel-server lintian
sudo aptitude -y install git-core git-doc git-email git-gui gitk
sudo aptitude -y install diffstat indent tofrodos fakeroot doxygen uboot-mkimage
sudo aptitude -y install sendmail mailutils meld atftpd sharutils
sudo aptitude -y install manpages-dev manpages-posix manpages-posix-dev linux-doc
sudo aptitude -y install vnc4server xvnc4viewer
```

4.2 Building up U-boot Image (u-boot.bin)

1)Depress folder "core" form the path "CDROM→OS→Linux→Source Code→uboot_kernel" on User`s CD-ROM

2) $ cd uboot-imx

3) $ ./run.sh linux –j4 (for the j parameter, you can refer your PC`s CPU performance to insert it, for this example, mine is "Quad Core CPU" , so insert "j4")

4) Then you can find "u-boot.bin" under folder "core/uboot-imx"

4.3 Building up Linux Kernel Image (uImage)

1)Depress folder "core" form the path "CDROM→OS→Linux→Source Code→uboot_kernel" on User`s CD-ROM

2) $ cd kernel_linux

3) $ ./run.sh linux –j4 (for the j parameter, you can refer your PC`s CPU performance to insert it, for this example, mine is “Quad Core CPU” , so insert “j4”)

4) Then you can find “uImage” from the path “core/kernel_linux/arch/arm/boot”

4.4 Configure TFTP Server

1) Install tftp and required packages

sudo apt-get install xinetd tftpd tftp

2) Create /etc/xinetd.d/tftp and put the following inside that text file

service tftp
{
protocol = udp
port = 69
socket_type = dgram
wait = yes
user = nobody
server = /usr/sbin/in.tftpd
server_args = /home/user/Documents/tftpboot
(Please note this is path of your TFTP folder) Do not copy this Line!!
disable = no
}

3) Make /tftpboot directory

$ cd ~
$ mkdir tftpboot
$ sudo chmod -R 777 /tftpboot
$ sudo chown -R nobody /tftpboot

4) Stop and start xinetd to apply changes

$ sudo /etc/init.d/xinetd stop
$ sudo /etc/init.d/xinetd start
You should now be able to use /home/user/tftpboot as your TFTP root and start serving TFTP!

4.5 Configure NFS Server

1) Install NFS required packages

$ sudo apt-get install nfs-common
$ sudo apt-get install nfs-kernel-server

2) Edit /etc/exports:

- *$ sudo vi /etc/exports*

- Add following line in the file

*/home/user/nfs *(rw,sync,no_root_squash)*

home/user/nfs is the path of your nfs folder.

- Reset nfs server

  $ sudo /etc/init.d/nfs-kernel-server start

- Check nfs status

  $ showmount -e localhost

If the setting works, then you show be to see the message below:

Export list for localhost:
/home/user/nfs

4.6 Create a bootable SD card with Ubuntu 10.04 file system

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

1) Check device node of your SD card by command below.
2) $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$
```

3) Install bootloader on SD card by command below.
   $ sudo dd if=u-boot.bin of=/dev/sdd bs=512 && sync && sync
4) Install Linux kernel image on SD card by command below.
   $ sudo dd if=uImage of=/dev/sdd seek=2048 bs=512 && sync && sync
5) Create EXT3 partition for SD card
   $sudo fdisk /dev/sdd
   Type the following parameters (each followed by <ENTER>):
   u [switch the unit to sectors instead of cylinders]
   d [repeat this until no partition is reported by the ‘p’ command]
   n [create a new partition]
   p [create a primary partition]
   1 [the first partition]
   8192 [starting at offset sector #8192, i.e. 4MB, 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]
6) Format new partition in EXT3 format
   $sudo mkfs.ext3 /dev/sdd1

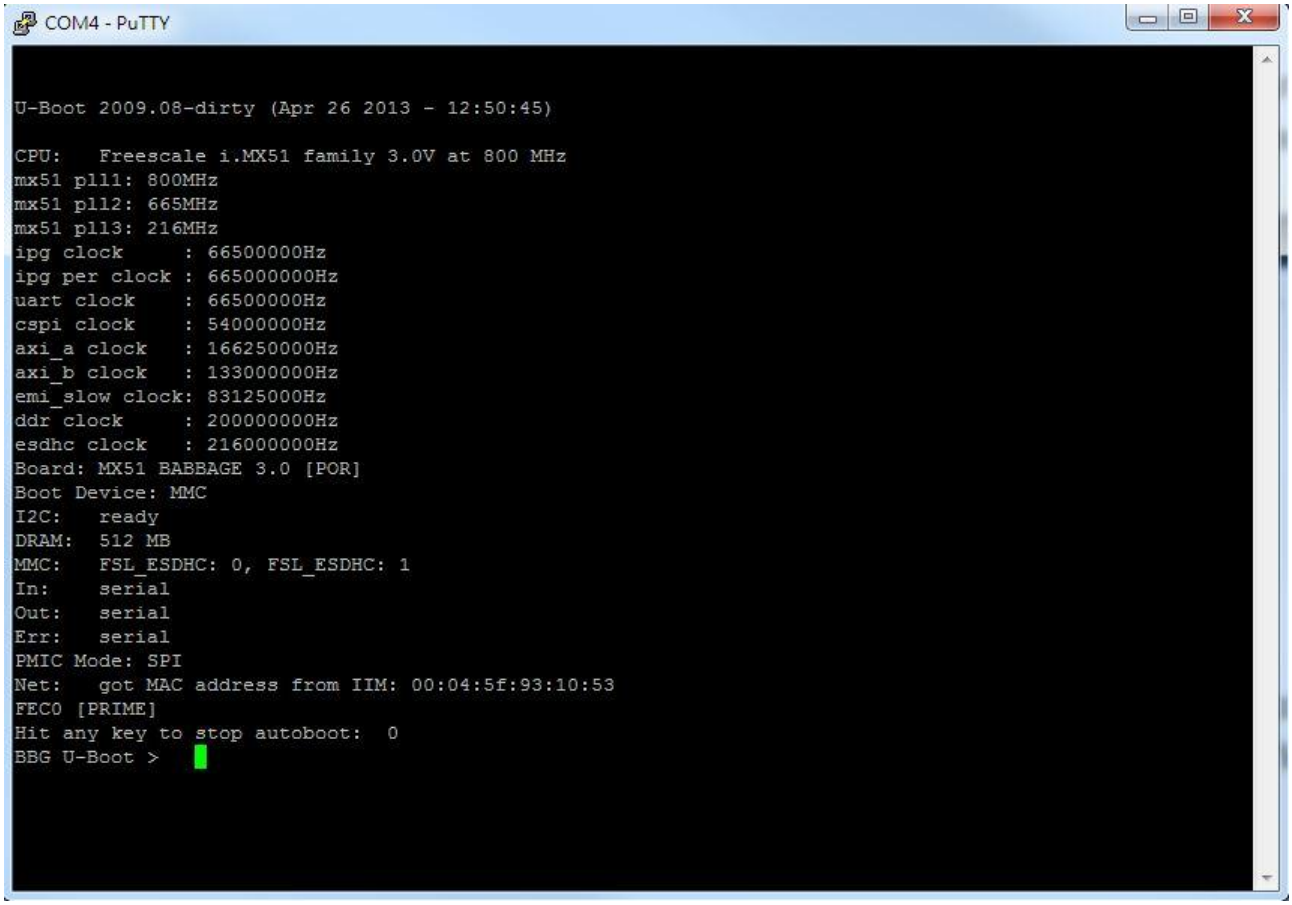
7) Please find the Ubuntu file system ”lucid_imx51.tar.gz” from the path ”CDROM→OS→Linux→file system→Ubuntu_1004_notebook” on User`s CD-ROM then copy it to the partition
   $ sudo tar –xzf lucid_imx51.tar.gz
   $ cd lucid
   $ sudo mount /dev/sdd1 /mnt
   $ sudo cp –dpR * /mnt

$ sudo umount /mnt

The Ubuntu file system content is now on the SD card. Please use user account "lucid" to log into system on serial terminal and password of the account is "lucid" Password of root account is "lucid".

4.7 Bootloader settings for booting from SD card

1) Please refer to section 2.3.2 and set SW2 to boot from SD socket.
2) Insert SD card on SD socket. Connect RS232 cross over cable from CN9 of RSC-IMX51 to COM port of Host PC.
3) Run hyper terminal program on Host PC (teraterm on Windows or minicom on Linux)
4) Power on RSC-IMX51 and press "space" key to get into bootloader menu.



5) Setup boot device
   BBG U-Boot > setenv bootcmd 'run bootcmd_sd2'
6) Save bootloader setting
   BBG U-Boot > saveenv
7) Press reset button and boot RSC-IMX51 again

4.8 Bootloader settings for booting from onboard SD

1) Please refer to section 2.3.2 and set SW2 to boot from onboard SD.
2) Follow section 4.7.3~4.7.4 to setup bootloader
3) Setup boot device
   BBG U-Boot > setenv bootcmd 'run bootcmd_sd1'
4) Save bootloader setting
   BBG U-Boot > saveenv
5) Press reset button and boot RSC-IMX51 again

4.9 Bootloader settings for booting from NFS

To boot from NFS server, you will have to finish settings on section 4.4 & 4.5 for your Linux Host PC

Please follow steps below to setup bootloader and boot from NFS

1) Please refer to section 2.3.2 and set SW2 to boot from SD socket.
2) Follow section 4.7.3~4.7.4 to setup bootloader
3) Setup boot device
   BBG U-Boot > setenv bootcmd 'run bootcmd_nfs'
4) Setup server IP (IP address of your NFS server, for example, IP address of my Linux Host PC is 192.168.1.2)
   BBG U-Boot > setenv serverip 192.168.1.2
5) Setup IP address of RSC-IMX51 ((IP address of RSC-IMX51. You can use command "DHCP" to get IP address in U-Boot)
   BBG U-Boot > setenv ipaddr 192.168.1.3
6) Setup nfsroot (path of your NFS folder on NFS server, fro example, mine is /home/user/nfs)
   BBG U-Boot > setenv nfsroot /home/user/nfs
7) Save bootloader setting
   BBG U-Boot > saveenv
8) Press reset button and boot RSC-IMX51 again

5. Android User Guide

Note: Installation procedures and screen shots in this section are for your reference and may not be exactly the same as shown on your screen.

5.1 How to setup PC (Windows) to support ADB

- ADB function
  - Download the Android SDK.
  - Update the adb configuration to scan for freescale's pid:
    - Run the SDK's tools to generate a configure file: android-sdk-windows\tools\android.bat update adb
    - Modify the files: X:\Profile\<your account>\.android\adb_usb.ini, to add freescale vendor id:

```
# ANDROID 3RD PARTY USB VENDOR ID LIST -- DO NOT EDIT.
# USE 'android update adb' TO GENERATE.
# 1 USB VENDOR ID PER LINE.
0x15a2
```

  - Unpack the freescale Android USB win driver in the release package tool\android_usb_fsl.zip.
  - Connect the Android Device into PC, uninstall your old driver named "Android Phone" in the device manager, then reinstall driver by scanning .inf file under the directory you unpack the android_usb_fsl.zip.
  - Restart the adb server
    - adb kill-server
    - adb start-server

5.2 How to setup PC (Linux) to support ADB

- ADB function
  - Download the Android SDK.
  - Update the adb configuration to scan for freescale's pid:
    - Run the SDK's tools to generate a configure file: android-sdk-linux_86/tools/android update adb
    - Modify the files:~/.android/adb_usb.ini, to add freescale vendor id:

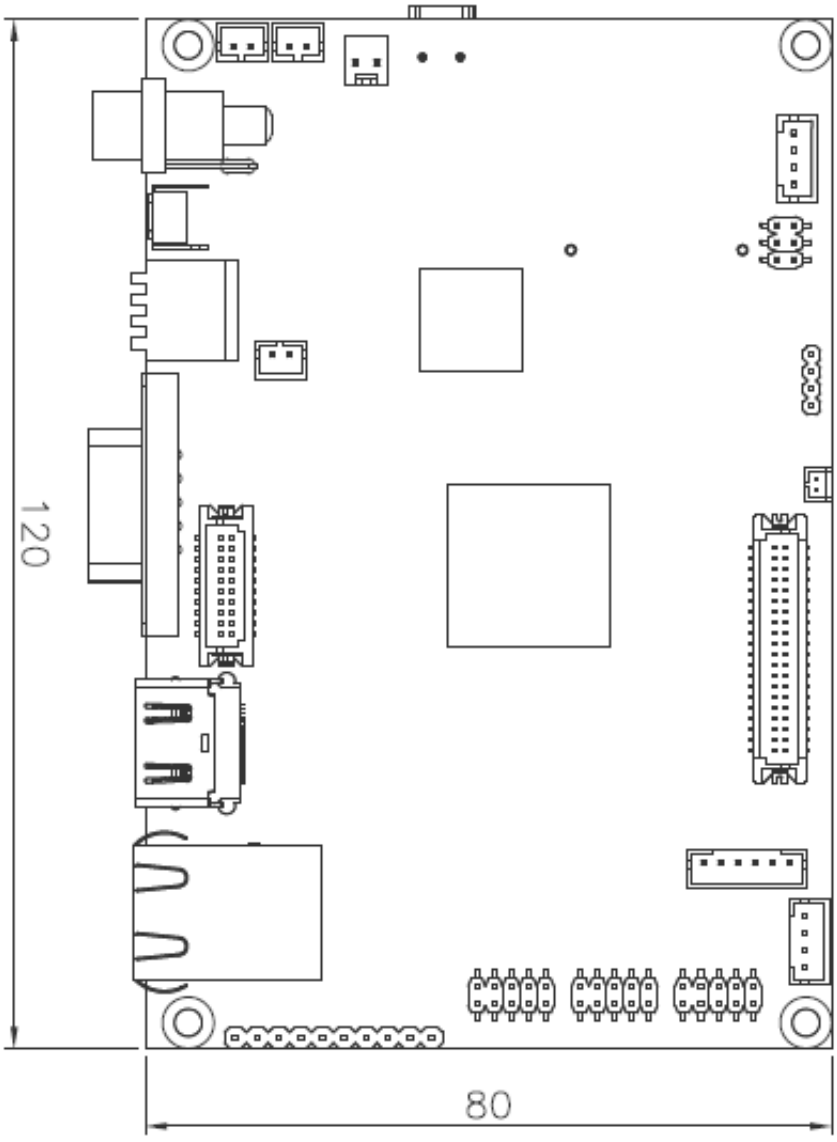
```
# ANDROID 3RD PARTY USB VENDOR ID LIST -- DO NOT EDIT.
# USE 'android update adb' TO GENERATE.
# 1 USB VENDOR ID PER LINE.
0x15a2
```

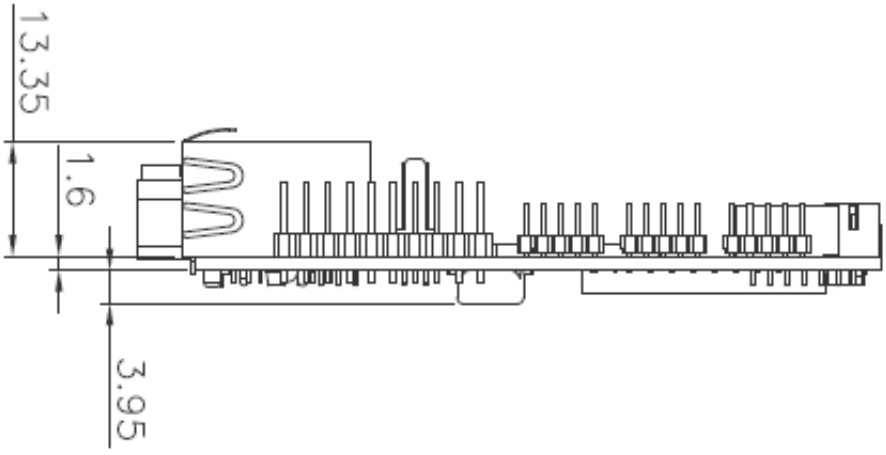
- Create a new udev rule file under the PC's /etc/udev/rules.d/ named: imx-android.rules. And fill in the following line into the file:

```
SUBSYSTEM=="usb", SYSFS{idVendor}=="15a2", MODE="0666"
```

- Change the new udev rule file's permission:
  chmod a+r /etc/udev/rules.d/imx-android.rules
- Connect the Android Device
- Restart the adb server
  - adb kill-server
  - adb start-server

6. Mechanical Drawing





Unit: mm