USER MANUAL

VAB-800

Pico-ITX Freescale Cortex-A8 board

1.02-03082013-142300

Copyright

Trademarks

All trademarks are the property of their respective holders.

Disclaimer

No license is granted, implied or otherwise, under any patent or patent rights of VIA Technologies. VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right the make changes to the products described in this manual at any time without prior notice.

Regulatory Compliance

FCC-A Radio Frequency Interference Statement

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 personal expense.

Notice 1

The changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Notice 2

Shielded interface cables and A.C. power cord, if any, must be used in order to comply with the emission limits.

Tested To Comply
With FCC Standards
FOR HOME OR OFFICE USE

Battery Recycling and Disposal

- ☐ Only use the appropriate battery specified for this product.
- ☐ Do not re-use, recharge, or reheat an old battery.
- ☐ Do not attempt to force open the battery.
- ☐ Do not discard used batteries with regular trash.
- ☐ Discard used batteries according to local regulations.

Safety Precautions

- ☐ Always read the safety instructions carefully.
- ☐ Keep this User's Manual for future reference.
- ☐ All cautions and warnings on the equipment should be noted.
- ☐ Keep this equipment away from humidity.
- ☐ Lay this equipment on a reliable flat surface before setting it up.
- ☐ Make sure the voltage of the power source and adjust properly 110/220V before connecting the equipment to the power inlet.
- ☐ Place the power cord in such a way that people cannot step on it.
- ☐ Always unplug the power cord before inserting any add-on card or module.
- ☐ If any of the following situations arises, get the equipment checked by authorized service personnel:
  - The power cord or plug is damaged.
  - Liquid has penetrated into the equipment.
  - The equipment has been exposed to moisture.
  - The equipment has not worked well or you cannot get it work according to User's Manual.
  - The equipment has dropped and damaged.
  - The equipment has obvious sign of breakage.
- ☐ Do not leave this equipment in an environment unconditioned or in a storage temperature above 60°C (140°F). The equipment may be damaged.
- ☐ Do not leave this equipment in direct sunlight.
- ☐ Never pour any liquid into the opening. Liquid can cause damage or electrical shock.
- ☐ Do not place anything over the power cord.
- ☐ Do not cover the ventilation holes. The openings on the enclosure protect the equipment from overheating

Box Contents and Ordering Information

VAB-800

- ☐ 1 x VAB-800
- ☐ 1 x DC jack power cable
- ☐ 1 x Audio cable
- ☐ 1 x CAN BUS/USB cable
- ☐ 1 x COM cable

Table of Contents

Lists of Figures

Lists of Tables

1. Product Overview

Based on the ultra compact Pico-ITX form factor, measuring 10 cm x 7.2 cm, the VIA VAB-800 is the first VIA Pico-ITX board to feature an ARM SoC. With the choice of either an 800MHz or 1GHz Freescale Cortex-A8 ARM SoC, the VIA VAB-800 combines two integrated GPUs for dual displays and 3D/2D graphics acceleration to provide full HD playback support for resolutions up to 1080p.

The ultra compact VIA VAB-800 Pico-ITX is optimized for both performance and power to meet the high end demands of advanced industrial and in-vehicle applications, boasting a ruggedized design with a wide operating temperature range from -20 to 70°C, while offering extremely low power consumption of a maximum 5W TDP. In addition to the ruggedized design VIA offers up to 7 years longevity support.

The VIA VAB-800 provides an impressive selection of rear I/O in a compact form factor including VGA and Mini HDMI ports, two USB 2.0 ports and one Ethernet port. Customers can take advantage of VIA's industry leading hardware and software support to create customized designs with a quick time to market. The VIA VAB-800 is also available with board support packages (BSP) for the Android and Embedded Linux 2.6 operating systems (BSP for Windows Embedded Compact 7 is only available through third party).

On board pin headers provide support for an additional of two USB 2.0 ports, one SATA connectors, two COM pin headers, two single channel LVDS connectors, one micro SD connector and eight GPIO pin headers, etc.

1.1. Key Features

- Supports integrated graphics processing (GPU) for 3D/2D graphics acceleration and dual displays
- Supports two single-channel 18/24-bit LVDS connectors
- Supports four USB ports (two as pin headers) and one USB device port
- Supports two COM ports
- Supports mini-HDMI and VGA ports
- Small form factor and low power design

1.2. Product Specifications

- **Processor**
  - Freescale Cortex-A8 Single-Core i.MX537 @800MHz (Optional for i.MX535 @1.0GHz)
- **Flash**
  - eMMC Flash (up to 64GB)
  - SPI Serial Flash (up to 4MB) (Optional and shared with COM1 port)
- **Graphics**
  - Support two independent, integrated graphics processing units: an OpenGL® ES 2.0 3D graphics accelerator and an OpenVG™ 1.1 2D graphics accelerator
- **System Memory**
  - 1GB DDR3-800 SDRAM using 128M x16 memory devices
- **Ethernet**
  - SMSC LAN8720A 10/100 PHY Transceiver with HP Auto-MDIX support
- **Audio**
  - Freescale SGTL5000 Low Power Stereo Codec with Headphone Amp
- **HDMI**
  - Onboard HDMI Transmitter
- **CAN**
  - TI SN65HVD1050 EMC Optimized CAN transceiver
- **USB Hub**
  - SMSC USB 2.0 High Speed 4-Port Hub Controller
- **Onboard I/O**
  - 1 X SATA I connector with voltage select jumper
  - 1 X SATA DOM power supply connector
  - 1 X Micro SD connector
  - 2 X COM connectors with power supply (one supports 8-wire, the other supports 2-wire)
  - 2 X Single-channel, 18/24-bit LVDS connectors
  - 1 X RTC battery pin header
  - 1 X Front audio pin header for line-in, line-out and MIC-in

- 1 X JTAG connector
- 1 X Boot flash select pin header
- 1 X miscellaneous pin header for 2 USB2.0 ports, 1 USB device port, 2 CAN bus ports, system power-on and reset
- 1 X miscellaneous pin header for 4-wire resistive touch screen interface, 1 I^2C pair and 8 GPIOs
- 1 X 5V DC-in power connector

- **Back Panel I/O**
  - 1 X VGA port
  - 2 X USB 2.0 ports
  - 1 X Mini HDMI port
  - 1 X RJ-45 LAN port

- **WatchDog Timer**
  - Integrated WatchDog timer supports two comparison points. Each comparison point can interrupt ARM core, 2nd comparison point capable of generating external interrupts on WDOG line

- **Power**
  - 2W typical, 5W max.

- **Operating System**
  - Android 2.3
  - Embedded Linux 2.6
  - Win CE 7.0 (through 3rd party)

- **Operating Conditions**
  - **Operating Temperature**
    - 0°C ~60°C (-20°C ~ 70°C for option)
  - **Operating Humidity**
    - 0% ~ 95% (relative humidity; non-condensing)

- **Form Factor**
  - Pico-ITX (8-layers)
  - 10 cm x 7.2 cm

Note:

For the software evaluation, please contact your regional sales or Field Applications Engineer to get the software download link.

1.3. Layout Diagram

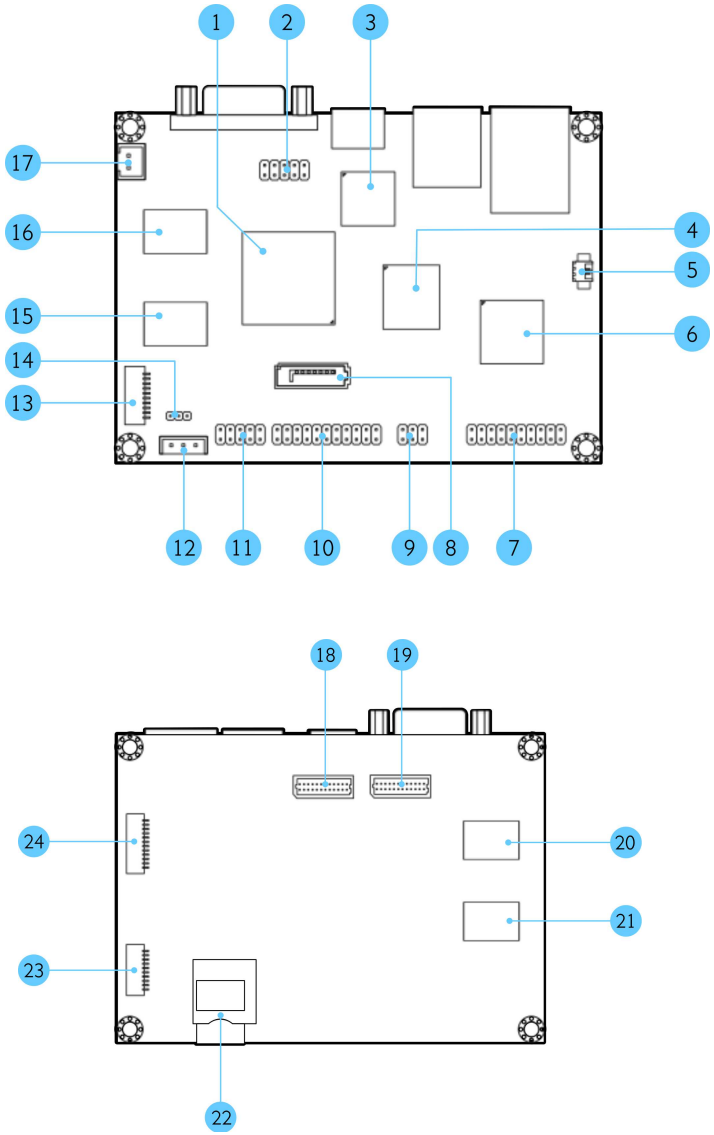


Figure 1: Layout diagram of the VAB-800 mainboard (top and bottom view)

Item	Description
1	i.MX537
2	J4: Boot Select
3	HDMI transmitter
4	eMMC Flash
5	J8: RTC Battery Connector
6	MC34708 PMIC
7	J10: DIO, Touch and I2C combination pin header
8	SATA connector
9	J2: CAN Bus Jumper
10	J9: USB, USB device, CAN, RST and PWNON combination pin header
11	F_audio
12	PWR1: External SATA Power connector
13	COM1
14	J3: SATA DOM Power select
15	DDR3
16	DDR3
17	DC-in power connector
18	LVDS2 connector
19	LVDS1 connector
20	DDR3
21	DDR3
22	SD
23	COM2
24	J_TAG

Table 1: Layout diagram description table of the VAB-800 mainboard

1.4. Product Dimensions

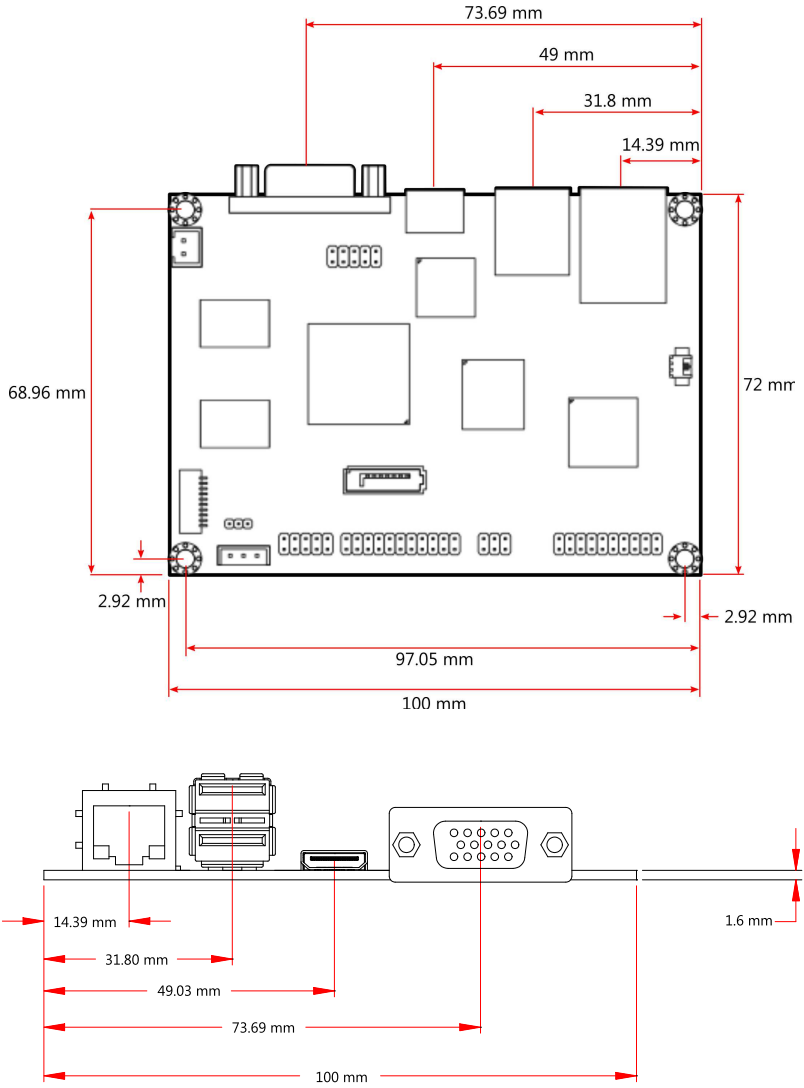


Figure 2: Mounting holes and dimensions of the VAB-800

1.5. Height Distribution

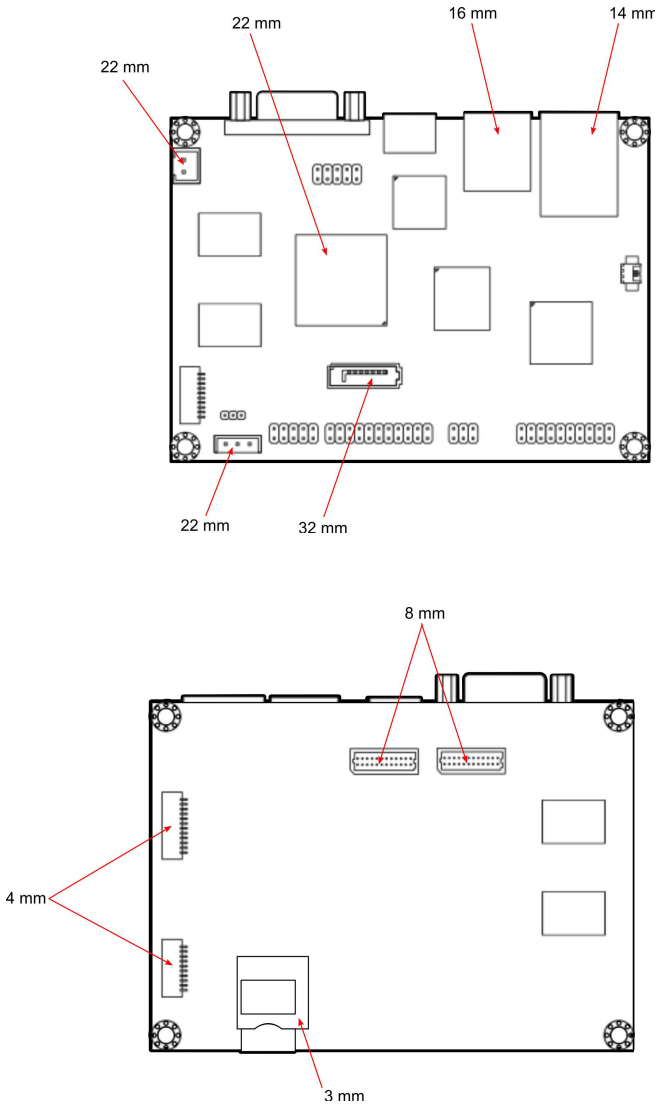


Figure 3: Height distribution of the VAB-800 mainboard

2. I/O Interface

The VAB-800 has a wide selection of interfaces. It includes a selection of frequently used ports as part of the external I/O coastline.

2.1. External I/O Ports

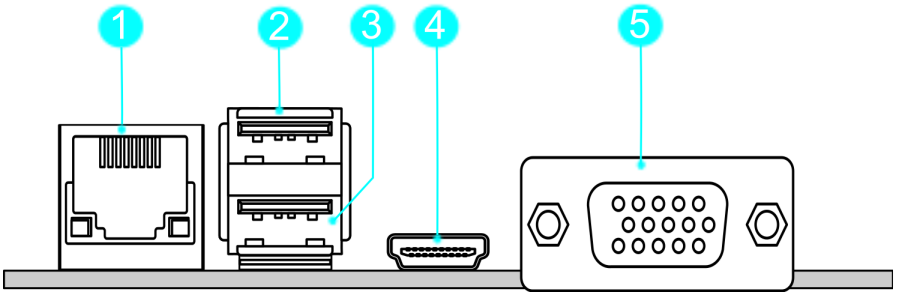


Figure 4: External I/O ports

Item	Description
1	RJ1: Fast Ethernet LAN port
2	USB1: USB 2.0 port
3	USB2: USB 2.0 port
4	HDMI1: Mini HDMI® port
5	VGA1: VGA port

Table 2: Layout diagram description table of external I/O ports

2.1.1. LAN port: Fast Ethernet

The integrated 8-pin Fast Ethernet port is using an 8 Position 8 Contact (8P8C) receptacle connector (commonly referred to as RJ45). The Fast Ethernet ports are controlled by VIA Fast Ethernet controller. The pinout of the Fast Ethernet port is shown below.

Pin	Signal
1	TD+
2	TD-
3	RD+
4	REGOUT
5	REGOUT
6	RD-
7	GND
8	GND

Table 3: Fast Ethernet port pinout

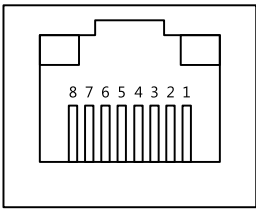


Figure 5: Fast Ethernet port pinout diagram

The RJ-45 port has two individual LED indicators located on the front side to show its Active/Link status and Speed status.

	Active LED (Left LED on RJ-45 connector)	Link LED (Right LED on RJ-45 connector)
Link Off	Off	Off
Speed_10Mbit	Flash in Green color	Off
Speed_100Mbit	Flash in Green color	The LED is always On in Yellow color

Table 4: Fast Ethernet LED color definition

2.1.2. USB 2.0 Port

The VAB-800 mainboard provides two USB 2.0 ports, Each USB port gives complete Plug and Play and hot swap capability for external devices. The USB interface complies with USB UHCI, Rev. 2.0. The pinout of the typical USB 2.0 port is shown below.

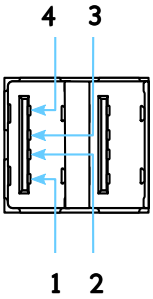


Figure 6: USB 2.0 port pinout diagram

USB1		USB2	
Pin	Signal	Pin	Signal
1	VCC	1	VCC
2	USB data -	2	USB data -
3	USB data +	3	USB data +
4	GND	4	GND

Table 5: USB 2.0 port pinout

2.1.3. Mini HDMI® Port

The integrated 19-pin HDMI® port uses an HDMI® Type C connector as defined in the HDMI® specification. The HDMI® port is for connecting to HDMI® displays. The pinout of the Mini HDMI® port is shown below.

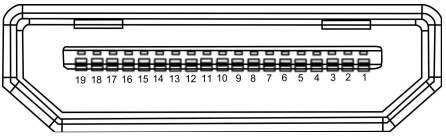


Figure 7: Mini HDMI® port pinout diagram

Pin	Signal	Pin	Signal
1	TMDS Data2 Shield	2	TMDS Data2+
3	TMDS Data2–	4	TMDS Data1 Shield
5	TMDS Data1+	6	TMDS Data1–
7	TMDS Data0 Shield	8	TMDS Data0+
9	TMDS Data0–	10	TMDS Clock Shield
11	TMDS Clock+	12	TMDS Clock–
13	DDC/CEC Ground	14	CEC
15	SCL	16	SDA
17	Reserved (N.C. on device)	18	+5V Power
19	Hot Plug Detect		

Table 6: HDMI® port pinout

2.1.4. VGA Port

The 15-pin VGA port uses a female DE-15 connector. The VGA port is for connecting to analog displays. The pinout of the VGA port is shown below.

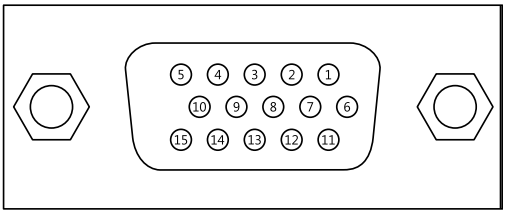


Figure 8: VGA port pinout diagram

Pin	Signal	Pin	Signal
1	VGA-R	9	+5VCRT
2	VGA-G	10	Ground
3	VGA-B	11	NC
4	NC	12	VGA-SPD
5	Ground	13	VGA_HS
6	Ground	14	VGA_VS
7	Ground	15	VGA-SPCLK
8	Ground		

Table 7: VGA port pinout

2.2. Onboard Connectors

2.2.1. LVDS1 & LVDS2 Connectors

The mainboard has two 24-pin LVDS panel connectors on the bottom side. The onboard LVDS panel connectors allow to connect the panel's LVDS cable to support the single-channel 18-bit/24-bit display. Backlight controls are integrated into the LVDS panel connector pinout. The LVDS panel connectors are labeled as "LVDS1" and "LVDS2". The pinout of the connectors is shown below.

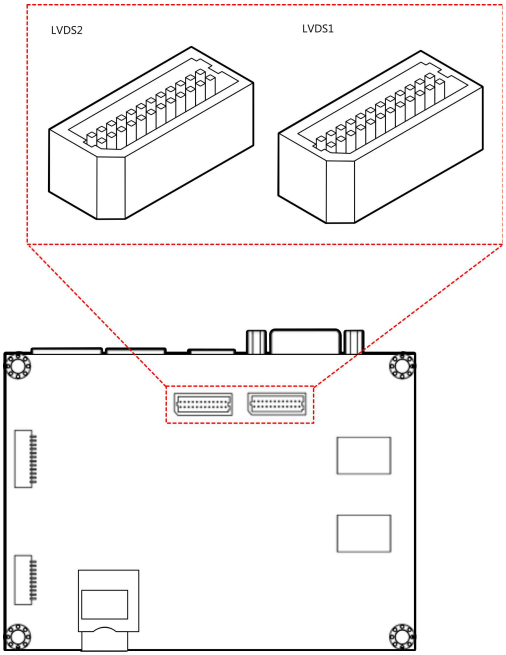


Figure 9: LVDS1 and LVDS2 connectors

Pin	Signal	Pin	Signal
1	LVDS0_TX0_N	2	12C2_SCL
3	LVDS0_TX0_P	4	12C2_SDA
5	GND	6	PVDD1 (3.3V-default; 5V-optional)
7	LVDS0_TX1_N-	8	PVDD1 (3.3V-default; 5V-optional)
9	LVDS0_TX1_P	10	IVDD1 (5V)
11	GND	12	IVDD1 (5V)
13	LVDS0_TX2_N-	14	DISP0_CONTRAST
15	LVDS0_TX2_P	16	DISP0_RD
17	GND	18	LCD0_BLT_EN
19	LVDS0_CLK_N-	20	GND
21	LVDS0_CLK_P	22	LVDS0_TX3_N
23	GND	24	LVDS0_TX3_P

Table 8: LVDS1 connector pinout

Pin	Signal	Pin	Signal
1	LVDS1_TX0_N	2	12C3_SCL
3	LVDS1_TX0_P	4	12C3_SDA
5	GND	6	PVDD2 (3.3V-default; 5V-optional)
7	LVDS1_TX1_N-	8	PVDD2 (3.3V-default; 5V-optional)
9	LVDS1_TX1_P	10	IVDD2 (5V)
11	GND	12	IVDD2 (5V)
13	LVDS1_TX2_N-	14	DISP0_CONTRAST
15	LVDS1_TX2_P	16	DISP1_RD
17	GND	18	LCD1_BLT_EN
19	LVDS1_CLK_N-	20	GND
21	LVDS1_CLK_P	22	LVDS1_TX3_N
23	GND	24	LVDS1_TX3_P

Table 9: LVDS2 connector pinout

2.2.2. SATA Connector

The SATA connector onboard can support up to 1.5 Gb/s transfer speeds. The SATA connector has a 7th pin[1] that can provide +5V power to a SATA Disk-on-Module (DOM). When a regular SATA hard drive is connected, the 7th pin will be a ground pin. The SATA connector is labeled as "SATA1". The pinout of the SATA connector is shown below.

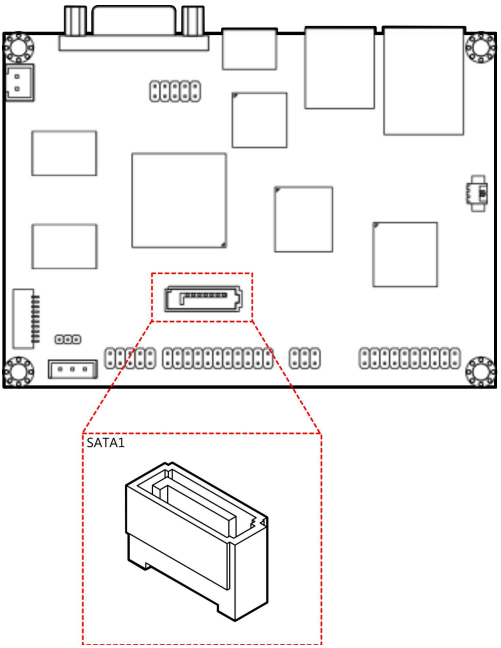


Figure 10: SATA connectors

SATA1	
Pin	Signal
1	GND
2	STXP_0
3	STXN_0
4	GND
5	SRXN_0
6	SRXP_0

7	GND/+5V

Table 10: SATA connector pinouts

Note:

The SATA connector pin 7 default setting is GND. The +5V supports is a factory option.

2.2.3. USB/USB device/CAN/RST/PWNON Combination Pin Header

The mainboard includes one USB, USB device, CAN, RST and PWNON combination pin header block labeled as "J9". The combination pin header is for connecting USB, USB device, CAN, RST and PWNON devices. The pinout of the pin header is shown below.

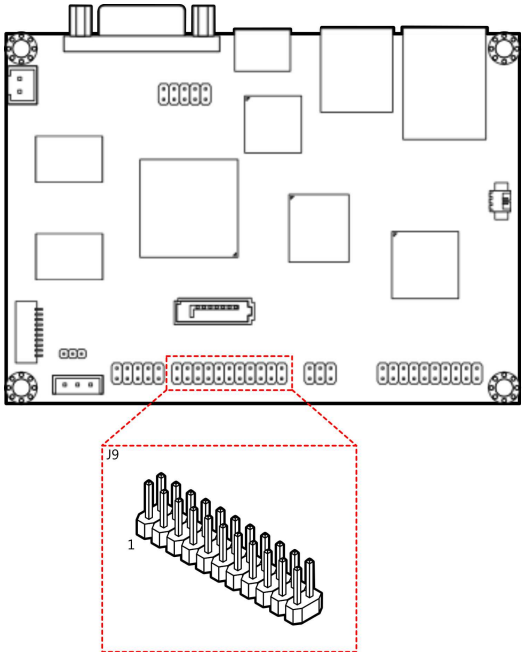


Figure 11: USB/USB device/CAN/RST/ PWNON combination pin header

Pin	Signal	Pin	Signal
1	GND	2	GND
3	USBD_T3+	4	C ANH1
5	USBD_T3-	6	C ANL1
7	USB_HOST5V	8	GND

9	USBD_T4-	10	CANH2
11	USBD_T4+	12	CANL2
13	GND	14	GND
15	PWNON1	16	USB_OTG_DP
17	GLBRST	18	USB_OTG_DN
19	GND	20	EXT_USB5V
21	P_LED	22	

Table 11: USB/USB device/CAN/RST/PWNON combination pin header pinout

2.2.4. DIO + Touch + I2C Combination Pin Header

The mainboard includes one DIO, 4-wire resistive touch and I2C combination pin header block labeled as "J10". The pinout of the pin header is shown below.

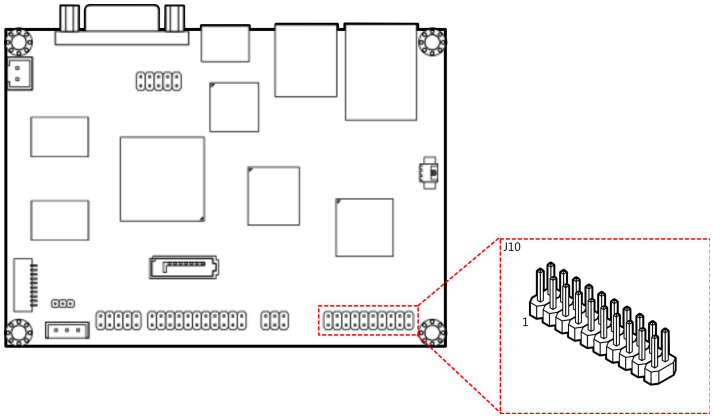


Figure 12: DIO + Touch + I2C combination pin header

Pin	Signal	Pin	Signal
1	TOPUCH_X0	2	LED+
3	TOPUCH_X1	4	LED-
5	TOPUCH_Y0	6	12C3_SCL
7	TOPUCH_Y1	8	
9	GND	10	12C3_SDA
11	GPO_10	12	GPI_2
13	GPO_11	14	GPI_16
15	GPO_12	16	GPI_18
17	GPO_13	18	GPI_19
19	GND	20	GND

Table 12: DIO + Touch + I2C combination pin header pinout

2.2.5. F_Audio Pin Header

The mainboard has a pin header for Line-out, Line-in and MIC-in. The pin header is labeled as "AUDIO". The pinout of the pin header is shown below.

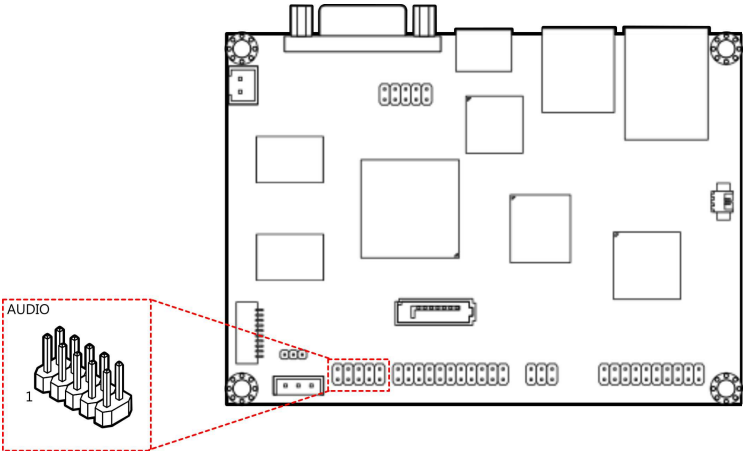


Figure 13: F_Audio pin header

Pin	Signal	Pin	Signal
1	HEAD_RIGHT	2	HEAD_LEFT
3	LINE_IN_R	4	LINE_IN_L
5	MIC_IN	6	MIC_IN
7	--	8	MIC_R_ESD
9	GND	10	GND

Table 13: F_Audio pin header pinout

2.2.6. External SATA Power Connector

The mainboard provides one built-in SATA power connector. This connector is required to power SATA hard drive. The SATA power connector is labeled as "PWR1". The pinout of the SATA power connector is shown below.

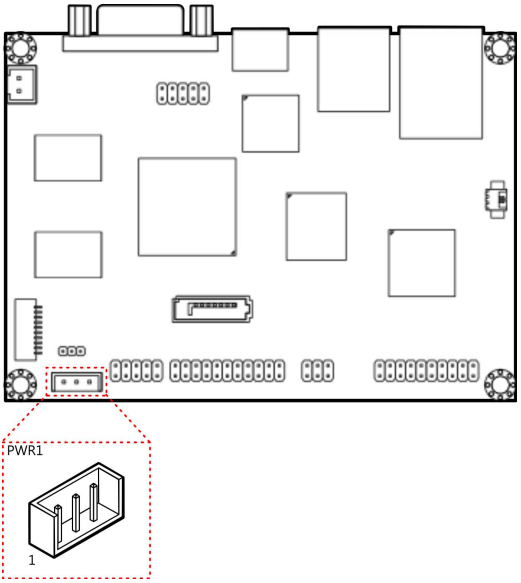


Figure 14: External SATA Power connector

Pin	Signal
1	DCDC_+5V
2	--
3	GND

Table 14: External SATA Power connector pinout

2.2.7. SATA DOM Power Select

The SATA connectors can be used to support Disk-on-Module (DOM) flash drives. When the jumpers are set, +5V will be delivered to the 7th pin of the SATA connectors. The pin jumper is labeled as "J3". The jumper settings are shown below.

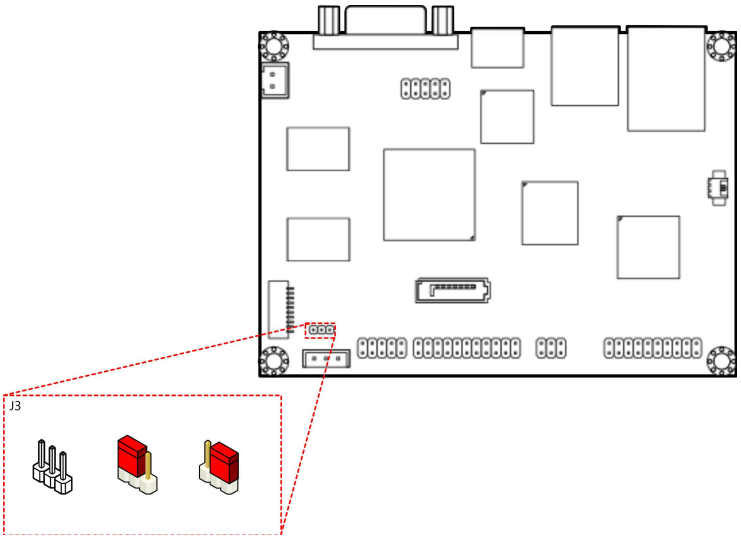


Figure 15: SATA DOM Power select

Pin	Signal
1	DCDC_+5V
2	SATA1_+5V
3	GND

Table 15: SATA DOM Power select

2.2.8. COM1 Connector

The mainboard includes one onboard COM connector on the top side of VAB-800. The onboard COM connector labeled as "COM1" is used to attach additional COM port that support RS-232 standard with DCE (data communication Equipment) type. The pinout of COM connector is as shown below.

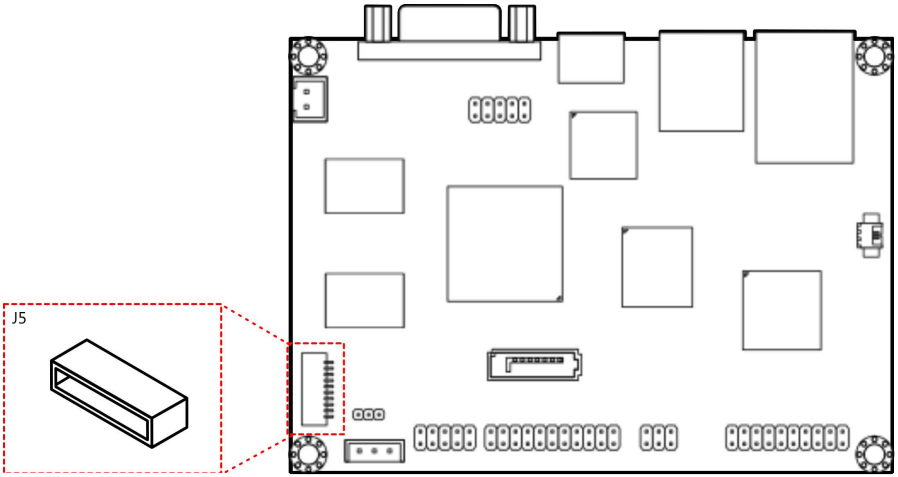


Figure 16: COM1 connector

Pin	Signal
1	--
2	COM_TXD1
3	COM_RXD1
4	COM_DCD1
5	COM_RI1
6	GND
7	COM_DTR1
8	COM_CTS1
9	COM_RTS1
10	COM_DSR1

Table 16: COM1 connector pinout

2.2.9. COM2 Connector

The mainboard includes one onboard COM connector on the bottom side of VAB-800. The onboard COM connector labeled as "COM2" is used to attach additional COM port for debug purpose only. The pinout of COM connector is as shown below.

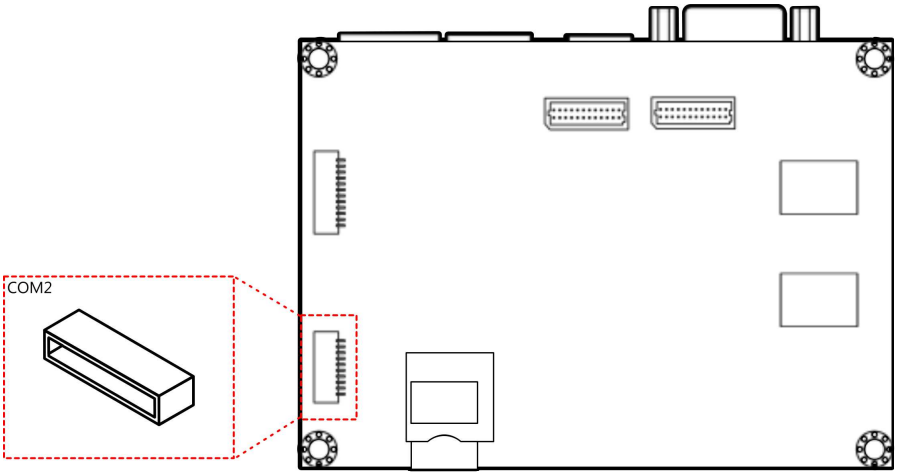


Figure 17: COM2 connector

Pin	Signal
1	--
2	COM_RXD2
3	COM_TXD2
4	--
5	--
6	GND
7	--
8	--
9	--
10	--

Table 17: COM2 connector pinout

2.2.10. RTC Battery Connector

The mainboard is equipped with onboard RTC battery connector used for connecting the external cable battery that provides power to the 32.768KHz crystal oscillator for Real Time Clock (RTC). The RTC battery connector is labeled as "J8". The connector pinout is shown below.

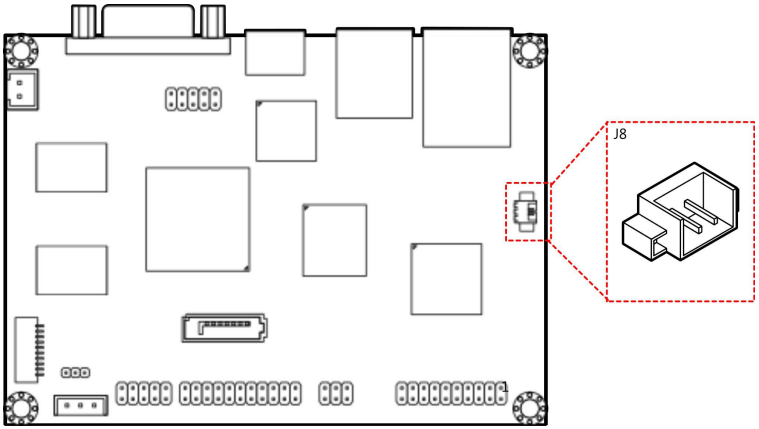


Figure 18: RTC Battery connector

Pin	Signal
1	+VBAT
2	GND

Table 18: RTC battery connector pinout

2.2.11. DC-In Power Connector

The mainboard supports +5V DC-In power connector to provide addition power to the rest of the system. The 2-pin power connector is used to connect the DC-In power jack. The connector is labeled as "PWR1". The pinout of the connector is shown below.

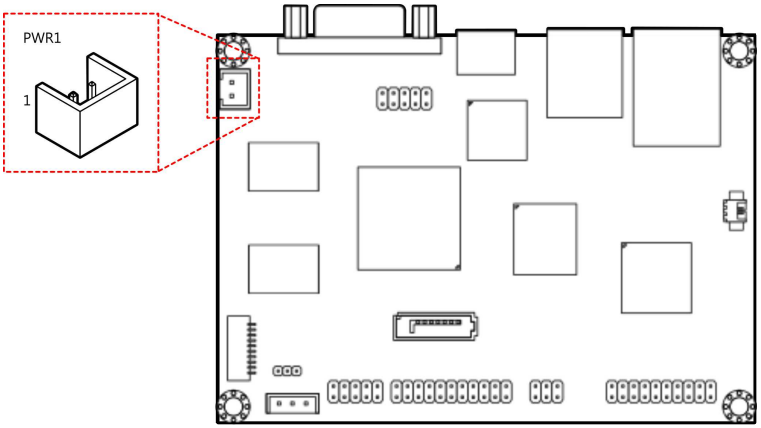


Figure 19: DC-In Power connector

Pin	Signal
1	+5V@3A Max.
2	GND

Table 19: DC-In Power connector

2.2.12. J_TAG Connector

The J_TAG connector provides a set of JTAG signals that allow JTAG debugging equipment to be used. The connector is labeled as "JTAG". The pinout of the connector is shown below.

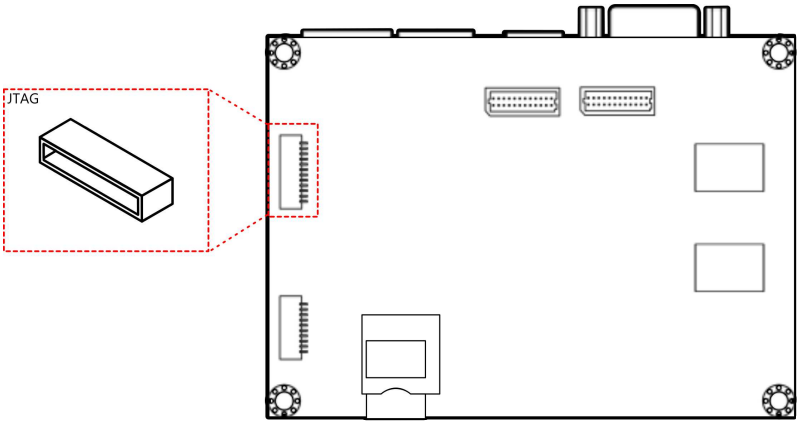


Figure 20: J_Tag connector

Pin	Signal
1	DCDC_3V2
2	VTREF_JTAG
3	JTAG_nTRST
4	JTAG_TD1
5	JTAG_TMS
6	JTAG_RTCK
7	JTAG_TCK
8	JTAG_TD0
9	JTAG_nSRST
10	JTAG_DE
11	JTAG_DACK
12	GND

Table 20: J_Tag connector

2.2.13. Boot Select

The Boots Select jumper labeled as J4 is to specify the boot device.

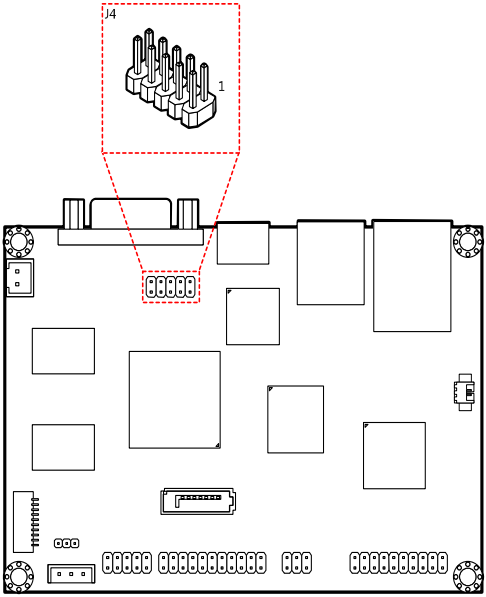


Figure 21: Boot Select jumper

Pin	Signal	Pin	Signal
1	DCDC_3V2	2	EIM_A20
3	DCDC_3V2	4	EIM_A21
5	DCDC_3V2	6	EIM_DA6
7	DCDC_3V2	8	EIM_DA7
9	1V8_SW5	10	MODE

	J4: Boot Select				
	1-2	3-4	5-6	7-8	9-10
*eMMC	short	short	short	open	open
Micro-SD	open	short	open	open	open
USB device	open	open	open	open	short

Table 21: Boot Select jumper settings

2.2.14. CAN BUS

The CAN BUS jumper labeled as J2 is used to enable/disable CAN BUS termination device.

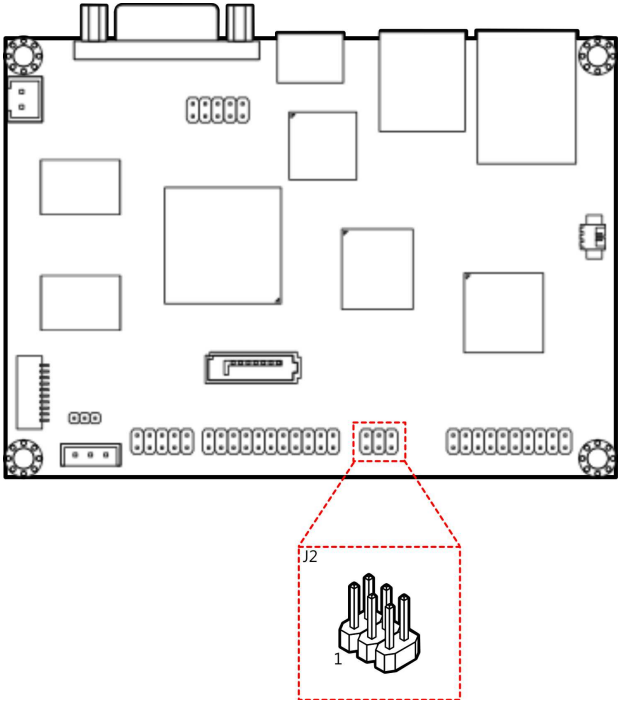


Figure 22: CAN BUS jumper

CAN Bus Jumper Setting	Pin 1	Pin 3	Pin 5
Enabled CAN Termination (default)	On	On	Off
	Pin 2	**Pin 4**	**Pin 6**
	On	On	Off

Table 22: CAN BUS jumper settings

3. Hardware Installation

3.1. Installing into a Chassis

The VAB-800 can be fitted into any chassis that has the mounting holes compatible with the standard Pico-ITX mounting hole locations. Additionally, the chassis must meet the minimum height requirements for specified areas of the mainboard.

3.1.1. Suggested minimum chassis dimensions

The figure below shows the suggested minimum space requirements that a chassis should have in order to work well with the VAB-800.

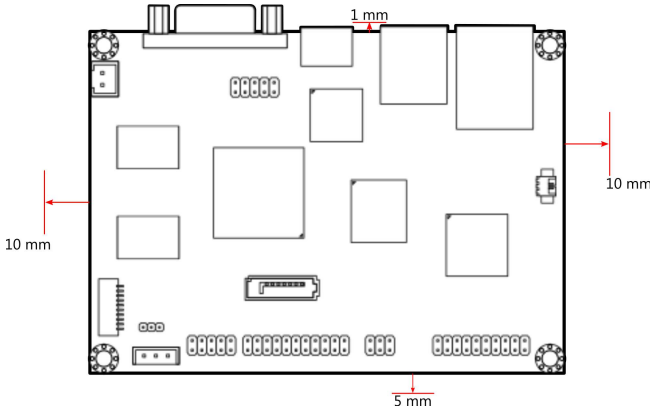


Figure 23: Suggested minimum chassis dimensions

Each side of the mainboard should have a buffer zone from the internal wall of the chassis. The side of the mainboard that accommodates the I/O coastline should have a buffer of 1.00 mm. The side on the opposite end of the I/O coastline should have a buffer of at least 5.00 mm. The two sides adjacent to the I/O coastline should have at least a 10.00 mm buffer.

3.1.2. Suggested minimum chassis height

The figure below shows the suggested minimum height requirements for the internal space of the chassis. It is not necessary for the internal ceiling to be evenly flat. What is required is that the internal ceiling height must be strictly observed for each section that is highlighted.

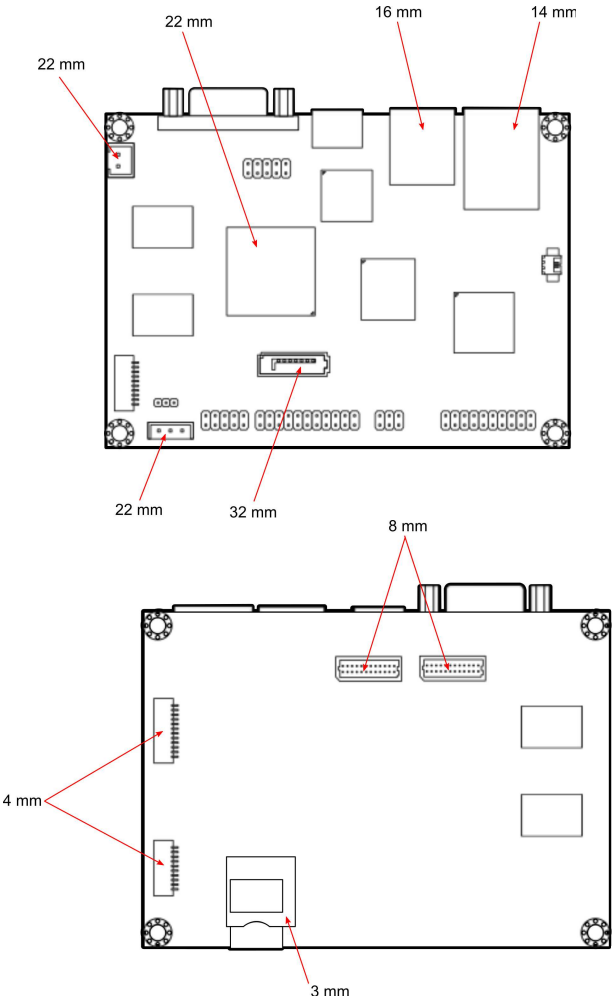


Figure 24: Suggested minimum internal chassis ceiling height

3.1.3. Suggested keepout areas

The figure below shows the areas of the mainboard that is highly suggested to leave unobstructed.

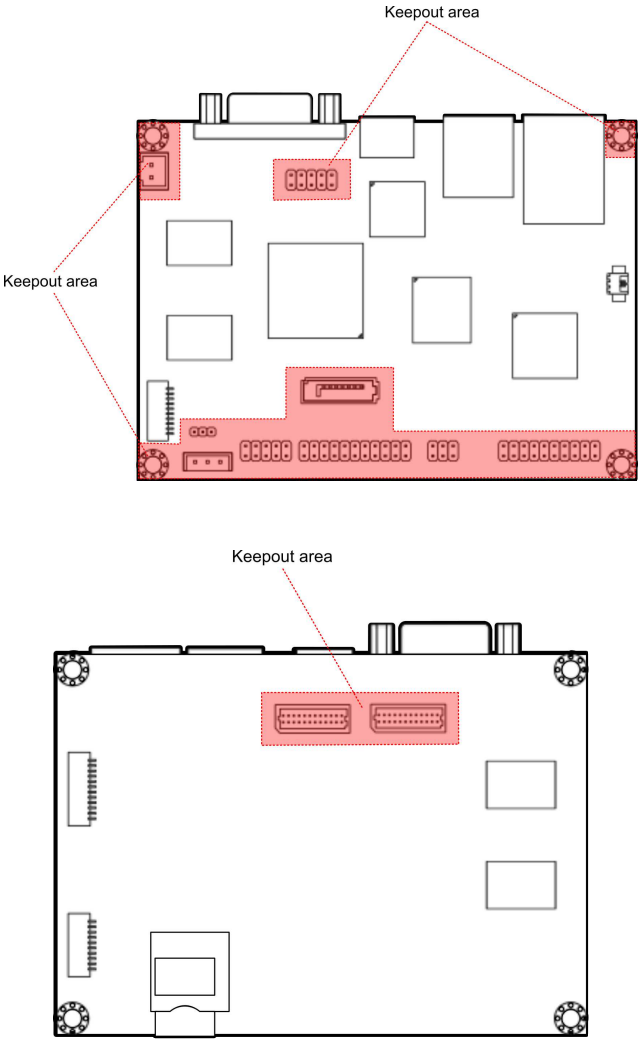


Figure 25: Suggested keepout areas

3.2. Connection of Cables

VAB-800 is equipped with 4 cables, including DC-in X 1, COM X 1, Audio X 1, USB/CAN X 1. The figure below shows the connection of these cables.

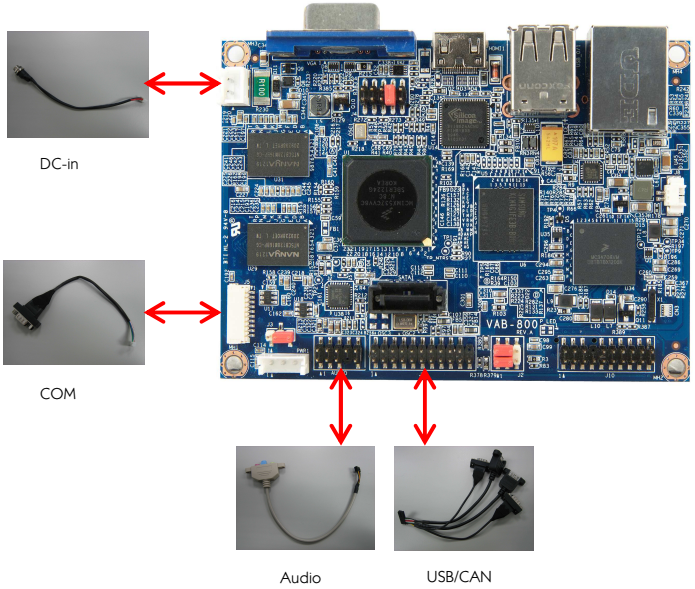


Figure 26: Connection of cables (top view)

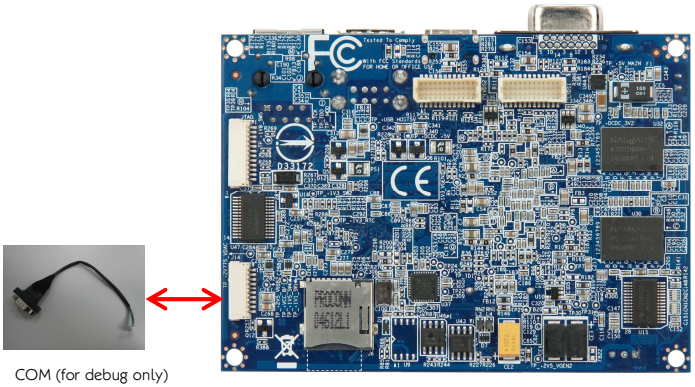


Figure 27: Connection of cables (bottom view)

4. Making Ubuntu Demo Image

This section describes how to evaluate the VAB-800 with Ubuntu image downloaded from Freescale official web site.

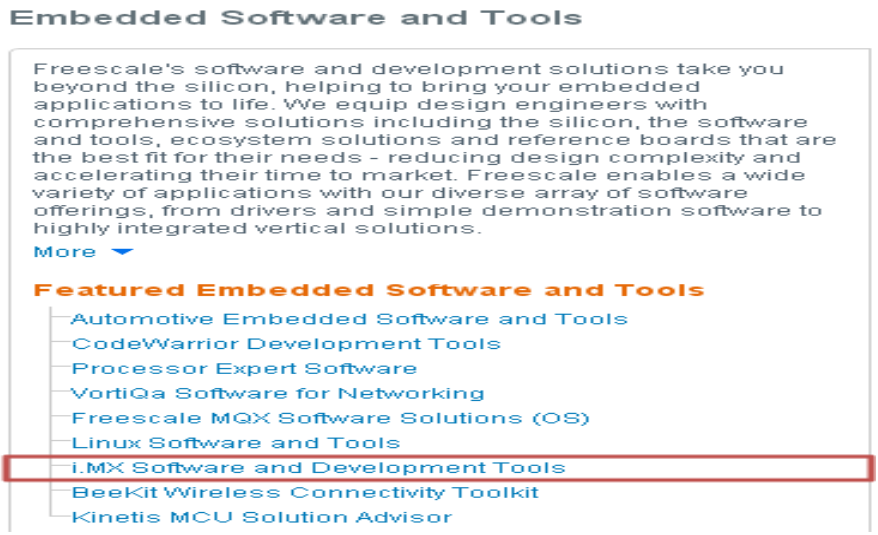
Note:

1. User should follow the Freescale's policy to download the Ubuntu demo image from Freescale official web site.
2. User should contact and follow Canonical's policy for Ubuntu commercial usage or redistribution.

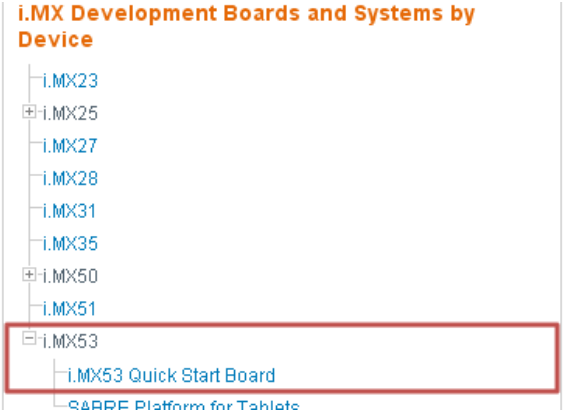
4.1. Getting Ubuntu demo image

1. Register your account at http://www.freescale.com/ to be able to download the files.
2. Go to "Design Resources" item --> Software and Tools--> then click "All Software and Tools".

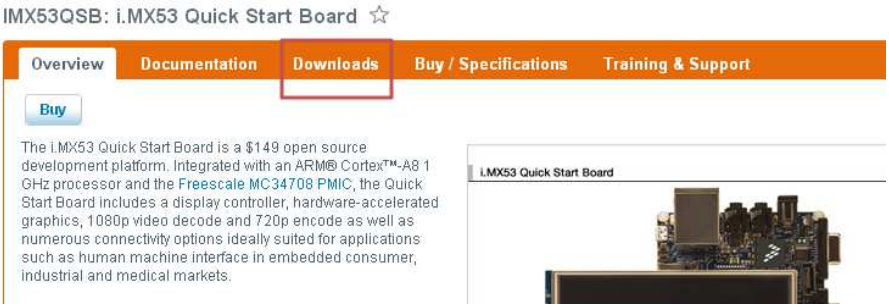
3. Go to "Featured Embedded Software and Tools". Click the "i.MX Softward and Development Tools".



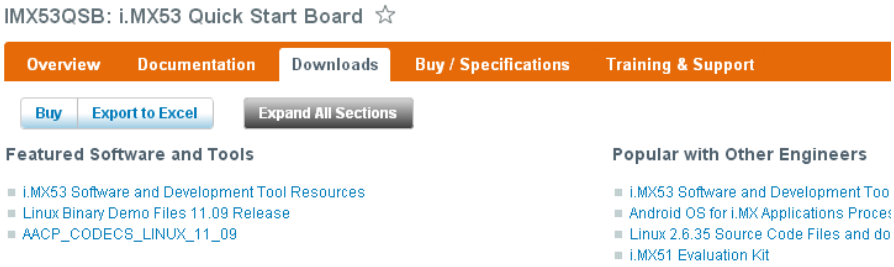
4. Go to "i.MX Development Boards and Systems by Device". Click the "i.MX53" plus sign to expand. Then click "i.MX53 Quick Start Board".



5. Click "Downloads".



There are development tools and prebuilt images shown here. You can select Run-time Software to expand all items.



6. Download MX53_QSB_UBUNTU_SD_DEMO_IMAGE

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB&fpsp=1&tab=Design_Tools_Tab#

User can get a zip file "MCIMX53-START-UBUNTU-11-09-Demo-Image.zip".

4.2. Making demo image into Micro SD

1. Prepare a Micro SD storage card (at least 4 GB size and Class 4).

2. Refer to the document "MCIMX53-START-R Flashing Doc.pdf" after unzipped. Follow the steps to make bootable Micro SD storage card.

4.3. Replace U-boot/Kernel/Modules of VAB-800

User has to replace u-boot, kernel and modules binary files that were built from VAB-800 BSP on the Micro SD storage card. These files can be found from EVK folder.

1. Getting u-boot, kernel and modules.

User can get u-boot.bin, uImage and modules.tar.bz2 from EVK folder.

Table 2

Binary	Description
u-boot.bin	U-Boot bootloader
uImage	Kernel
modules.tar.bz2	Driver modules made through LTIB
vab-800_rootfs_patch.tar.bz2	Modifications configure files if user would like to evaluate VAB-800 with Ubuntu

2. Copying the u-boot/uImage/driver modules.

2.1 Insert the Micro SD storage card which the OS has made by "MCIMX53-START-R-BSP-11-09_C.exe" into your Linux developing PC.

2.2 Umount the Micro SD storage card if Linux OS auto mount your Micro SD card.

The Micro SD card code name for this example is sdb1.

```
user@user:~ $ sudo umount /dev/sdb1
```

The path of EVK for this example is /home/user/EVK

2.3 Copy u-boot "u-boot.bin" to Micro SD storage card

```
user@user:~/EVK$ sudo dd if=./u-boot.bin of=/dev/sdb bs=512 seek=2
skip=2
```

2.4 Copy kernel "uImage" to Micro SD storage card

```
user@user:~/EVK$ sudo dd if=uImage of=/dev/sdb bs=512 seek=2048
```

2.5 Add driver modules of VAB-800

```
user@user:~/EVK$ sudo mount /dev/sdb1 /mnt
user@user:~/EVK$ sudo mkdir /mnt/mountpoint
user@user:~/EVK$ sudo cp ./modules.tar.bz2 /mnt/mountpoint/lib
user@user:~/EVK$ cd /mnt/mountpoint/lib
user@user:/mnt/mountpoint/lib$ sudo tar jxvf modules.tar.bz2
```

2.6 Put modification files in root file system

The configuration files should be modified for VAB-800. User can use files "vab-800_rootfs_patch.tar.bz2" in EVK folder.

■ Unzip vab-800_rootfs_patch.tar.bz2

```
user@user:~/EVK$ sudo tar jxvf vab-800_rootfs_patch.tar.bz2
```

User will get the following files as shown below:

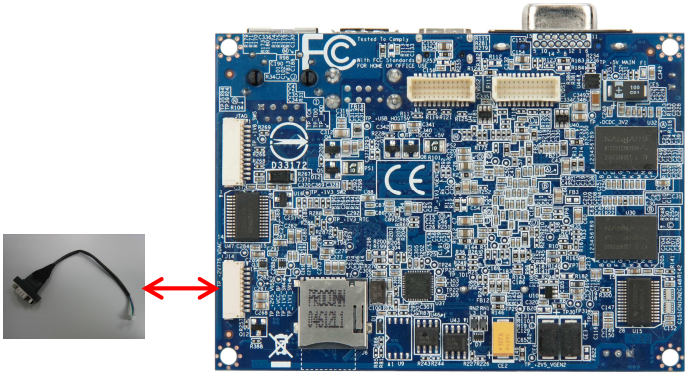
■ User runs the script "vab-800.sh" to copy modification files automatically.

```
user@user:~/EVK$ sudo ./vab-800.sh /mnt/mountpoint
user@user:~/EVK$ sudo sync && sync
user@user:~/EVK$ sudo umount /mnt
```

Now, unplug Micro SD storage card from your computer.

2.7 Boot from Micro SD storage card (on VAB-800)

■ Attach COM cable to VAB-800 COM2, and connect to your computer



■ To modify parameters in u-boot

To setup or modify u-boot parameters in u-boot:

[Display parts]

```
setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
setenv hdmi ‘setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary’
setenv lvds1 ‘video=mxcdi0fb:RGB24,480C60 di0_primary ldb=di0
vga=off’
setenv lvds2 ‘video=mxcdi1fb:RGB24,480C60 di1_primary ldb=di1
vga=off’
```

[Storage device parts]

```
setenv bootargs_mmc 'set bootargs ${bootargs} root=/dev/mmcblk1p1 rw
rootwait'
setenv bootargs_emmc ‘setenv bootargs ${bootargs}
root=/dev/mmcblk0p2 rootwait rw’
setenv bootcmd_emmc ‘run bootargs_base bootargs_emmc;fatload mmc 1
0x70800000 uImage;bootm’
```

[Boot storage device parts]

Here, user can set the boot device as either Micro SD or eMMC.

Set boot device as Micro SD:

```
setenv bootcmd 'run bootcmd_mmc; bootm ${loadaddr}'
```

Set boot device as eMMC:

```
setenv bootcmd 'run bootcmd_emmc; bootm ${loadaddr}'
```

[Others]

```
setenv bootargs 'console=ttymxc1,115200 gpu_nommu setenv bootargs
console=ttymxc1,115200 video=mxcdi0fb:RGB24,1920x1080M@60 hdmi
di0_primary root=/dev/nfs ip=dhcp
nfsroot=10.29.240.144:/tftpboot/rootfs,v3,tcp'
```

User has to save the parameters when completing the settings.

```
VAB-800 U-Boot > saveenv
VAB-800 U-Boot > boot
```

Finally, user should get the parameter in u-boot after modifying, type command *pri* in u-boot. In the example: the display is VGA and boot from Micro SD card.

```
baudrate=115200
loadaddr=0x70800000
netdev=eth0
ethprime=FEC0
uboot=u-boot.bin
kernel=uImage
ethact=FEC0
bootcd=run bootcmd_mmc
bootargs_nfs=setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp
nfsroot=${serverip}:${nfsroot},v3,tcp
bootcmd_net=run bootargs_base bootargs_nfs; tftpboot ${loadaddr}
${kernel}; bootm
bootcmd_mmc=run bootargs_base bootargs_mmc; mmc read ${loadaddr}
0x800 0x1800; bootm
bootargs_mmc=set bootargs ${bootargs} root=/dev/mmcblk1p1 rw
rootwait
vga=setenv bootargs console=ttymxc1,115200 video=mxcdi1fb:GBR24,VGA-
```

```
XGA di1_primary vga
bootcmd_obds=ext2load mmc 0:1 0x70800000 /unit_tests/obds.bin; go
70800000
nfsroot=/tftpboot/rootfs
lcd=video=mxcdi0fb:RGB24,SEIKO-WVGA di0_primary
lvds=video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0
lvds1=video=mxcdi0fb:RGB24,480C60 di0_primary ldb=di0 vga=off
lvds2=video=mxcdi1fb:RGB24,480C60 di1_primary ldb=di1 vga=off
bootcmd=run bootcmd_mmc; bootm ${loadaddr}
bootfile=uImage
bootargs=console=ttymxc1,115200 gpu_nommu setenv bootargs
console=ttymxc1,115200 video=mxcdi0fb:RGB24,1920x1080M@60 hdmi
di0_primary root=/dev/nfs ip=dhcp
nfsroot=10.29.240.144:/tftpboot/rootfs,v3,tcp
fileaddr=70800000
netmask=255.255.254.0
hdmi=setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary
serverip=192.168.0.1
ipaddr=192.168.0.2
gatewayip=192.168.0.1
bootargs_base=setenv bootargs console=ttymxc1,115200 gpu_nommu
${vga}
dnsip=192.168.0.1
bootdelay=3
stdin=serial
stdout=serial
stderr=serial
```

4.4. Setting U-boot

1. Setting the display devices

[VGA]

To set VGA as display output

```
setenv bootargs_base ‘setenv bootargs console=ttymxc1,115200
gpu_nommu ${vga}’
```

To set VGA resolution

```
setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
```

To replace the resolution, change the red color part with the desire resolution mode.

For example: To set 1680 x 1050 resolution, change VGA-XGA with VGA-WSXGA+:

```
setenv vga 'setenv bootargs console=ttymxc1,115200
video=mxcdi1fb:GBR24,VGA-WSXGA+ di1_primary vga'
```

Notes:

There are four modes to be set in VAB-800:

VGA-WSXGA+ : 1680x1050p-60

VGA-SXGA : 1280x1024p-60

VGA-XGA : 1024x768p-60

VGA-SVGA : 800x600p-60

[HDMI]

To set HDMI as display output

```
setenv bootargs_base ‘setenv bootargs console=ttymxc1,115200
gpu_nommu ${hdmi}’
```

To set HDMI resolution

```
setenv hdmi 'setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1024x768M@60 hdmi di0_primary'
```

To replace the resolution, change the red color part with the desire resolution mode.

For example: To set 1920x1080, change 1024x768 with 1920x1080:

```
setenv hdmi 'setenv bootargs console=ttymxc1,115200
video=mxcdi0fb:RGB24,1920x1080M@60 hdmi di0_primary'
```

[LVDS]

To set LVDS as display output

```
setenv bootargs_base 'setenv bootargs console=ttymxc1,115200
gpu_nommu ${lvds1}'
```

or

```
setenv bootargs_base 'setenv bootargs console=ttymxc1,115200
gpu_nommu ${lvds2}'
```

To set LVDS port

```
setenv lvds1 'video=mxcdi0fb:RGB24,480C60 di0_primary ldb=di0
vga=off'
setenv lvds2 'video=mxcdi1fb:RGB24,480C60 di1_primary ldb=di1
vga=off'
```

Notes: VAB-800 support AUO 7" 800x480 LVDS panel in default.

2. Setting storage devices

[Micro SD storage card]

```
setenv bootargs_mmc 'set bootargs ${bootargs} root=/dev/mmcblk1p1 rw
rootwait'
setenv bootcmd_mmc 'run bootargs_base bootargs_mmc; mmc read
${loadaddr} 0x800 0x1800; bootm'
```

[eMMC]

```
setenv bootargs_emmc 'setenv bootargs ${bootargs}
root=/dev/mmcblk0p2 rootwait rw'
setenv bootcmd_emmc 'run bootargs_base bootargs_emmc;fatload mmc 1
0x70800000 uImage;bootm'
```

3. Setting booting storage devices

Here, user can set the boot device as either Micro SD or eMMC.

Set boot device as Micro SD:

```
setenv bootcmd 'run bootcmd_mmc; bootm ${loadaddr}'
```

Set boot device as eMMC:

```
setenv bootcmd 'run bootcmd_emmc; bootm ${loadaddr}'
```

4. Setting MAC address

Here, user can set the MAC address in u-boot. The address is on the Ethernet physical port.

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

It is a must to reset after you set MAC address and save already:

```
saveenv
reset
```

4.5. Making demo image to eMMC (optional)

This section will guide user to copy images to eMMC for evaluation. However, it is a must to finish first the section 4.2~4.3 before applying this section.

To make a demo image compression file, follow the steps below:

1. Getting u-boot, and kernel image.

User can directly get the u-boot.bin, and uImage from EVK folder.

Table 3

Binary	Description
u-boot.bin	U-Boot bootloader Image
uImage	Linux Kernel Image

2. Making Ubuntu root file system

2.1 Insert the Micro SD storage card which the OS has made in section 4.3 on your Linux developing PC. (Ubuntu 10.04.x x86 at least)

2.2 Compress Micro SD storage card as a compression file.

The Micro SD card code name for this example is sdb1.

```
user@user:~ $ sudo mount /dev/sdb1 /mnt/mountpoint
user@user:~ $ cd /mnt/mountpoint
user@user: /mnt/mountpoint$ sudo tar jcvf ~/EVK/rootfs.tar.bz2 ./*
...
user@user: /mnt/mountpoint$ sudo sync && sync
user@user: /mnt/mountpoint$ cd ~
user@user:~ $ sudo umount /mnt/mountpoint
```

It would take 30~40 min to generate a "rootfs.tar.bz2". User can uncompress it into eMMC, and get an Ubuntu demo image.

3. Make demo image boot from eMMC.

5. Evaluation of Android 2.3 for the VAB800

5.1. Introduction

The VIA VAB-800 platform is an embedded system powered by ARM processor with Android 2.3 operating system as default. VAB-800's Android BSP offers a development package. Users can develop it under an Ubuntu environment.

There are six major boot components for Android, the **"u-boot.bin", "uImage"**, **"uramdisk.img", "system.img", "ramdisk.img",** and **"recovery.img"**. The system will not boot successfully into an Android environment if one of these files does not exist in the boot media.

5.1.1. Package Content

There are two folders in VAB-800 evaluation package.

Figure 28. VAB-800 evaluation package content

5.1.2. EVK Folder Contents

- **Images_autoinstall_sd:**
    - scripts: mk_install_sd.sh. User can run this script to make a VAB-800 linux bootable Micro SD storage card, in order to

install Android evaluation image to VAB-800 on board eMMC storage automatically.

- out:
  - Image files "u-boot.bin", "uImage", "rootfs.tar.bz2" to make a Linux bootable Micro SD card.
- img_android: Android image files, **"u-boot.bin", "uImage"**, **"uramdisk.img", "system.img", "ramdisk.img",** and **"recovery.img".**

- **app:** apk files "ApiDemos", "AppleVG", "MX_Video_Player_1.6e", "StarVG" and "TigerVG" which was provided by Freescale for evaluation.
- **Tools:** android_usb_fsl.zip, the driver when user would like to use android-sdk through USB-OTG.

Note:

If user needs the supporting files for all software mentioned in VAB-800 Android BSP document, please contact our regional sales representative for assistance.

5.2. Making Android System Booting Media

VAB-800 Android BSP only supports booting from onboard eMMC. This section will describe how to make Android system a boot media for VAB-800 from EVK folder.

5.2.1. Requirements

- Linux development computer
- Micro SD storage card. Recommended size is at least 2GB, class 4.
- EVK folder

5.2.2. Procedure

The following steps will show you how to make a bootable Micro SD card and to make Android system eMMC through the Micro SD.

Step 1

Insert the SD card to your Linux development computer.

Step 2

Open **"Terminal"** utility.

Step 3

The Micro SD storage card can be identified and auto mount once inserted to the computer. You can check Micro SD card code name by entering **$ df –h**

```
user@ubuntu:~$ df -h
...
Filesystem        Size   Used   Avail   Use%   Mounted on

/dev/sdb1         7.2G   531M   6.3G    8%      /media/usb
```

However, there are some instances that the Micro SD storage card could not identify or auto mount after inserting to the computer. In that case, you can try the other way to identify the Micro SD card.

```
user@ubuntu:~$ dmesg | grep -i removable
[105.502517] sd 9:0:0:0 [sdb] Attached SCSI removable disk
```

The Micro SD storage card code name for this example is identified as **sdb**.

Step 4

The EVK path for this example is /home/user/EVK, user can enter the folder "Images_autoinstall_sd/scripts" which is under EVK.

```
user@ubuntu:~$ sudo cd /home/user/EVK/Images_autoinstall_sd/scripts
user@ubuntu:~/EVK/Images_autoinstall_sd/scripts$ ls
mk_install_sd.sh
```

Step 5

User can run the script "mk_install_sd.sh" to make a bootable Micro SD card. Thus, user can install the Android image into VAB-800 eMMC through the Micro SD card. The target Micro SD storage card code name for this example is identified as **sdb**.

```
user@ubuntu:~/EVK/Images_autoinstall_sd/scripts$ sudo ./mk_install_sd.sh
/dev/sdb
```

User can unplug Micro SD card from Linux development PC after the message "Done!" is shown.

Step 6

Insert Micro SD card into VAB-800 and be sure the boot selection is Micro SD at J4.

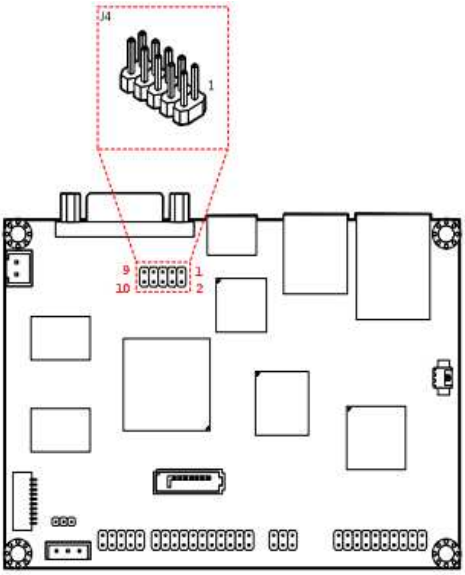


Figure 29. Select Micro SD card boot at J4

	J4 : Boot Sect				
	1-2	3-4	5-6	7-8	9-10
eMMC	Short	Short	Short	Open	Open
Micro-SD	Open	Short	Open	Open	Open
USB-OTG	Open	Open	Open	Open	Short

Table 23. J4 jumper settings

Step 7

The Android installation procedure will run automatically after power on the VAB-800. User can press "y" to install Android image into eMMC or other keys to exit without installing Android images. The message "power off" will be shown when image is installed. User can press "y" to power off directly.

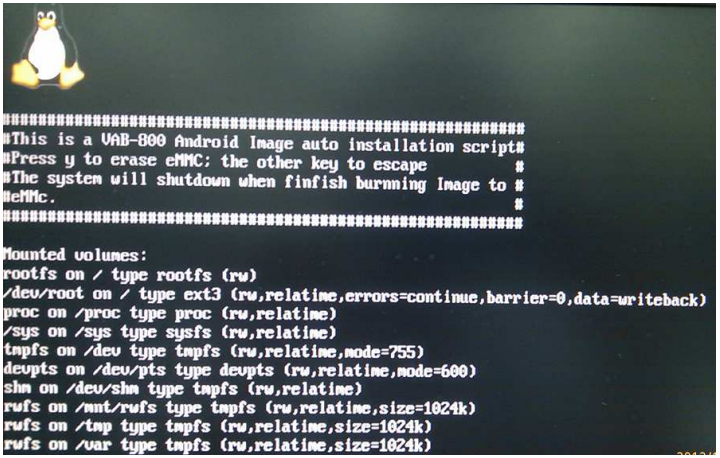


Figure 30. Android image auto installation

Step 8

Be sure the boot selection is eMMC at J4. The Android will display from VGA in default when user power on the VAB-800.

5.2.3. Setting u-boot Parameter for eMMC

Connect the VAB-800 and host PC through J14 (COM 2) of VAB-800. Run "putty" on host PC to receive the booting message. Power on the VAB-800 and press any key to stop the booting process as shown in Figure 31.

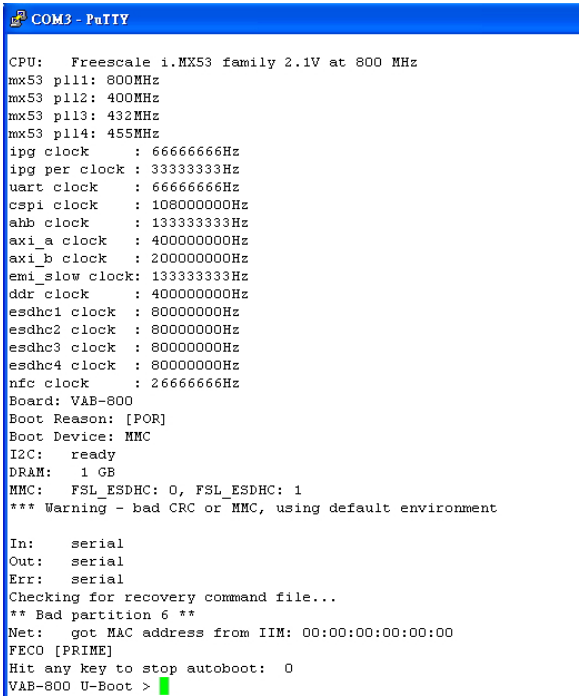


Figure 31. u-boot parameter

Make sure the parameters are correct in u-boot:

```
VAB-800 U-Boot > pri
bootcmd=run bootcmd_SD
bootcmd_SD=run bootcmd_SD1 bootcmd_SD2
bootcmd_SD1=run bootargs_base set_display bootargs_android
bootcmd_SD2=mmc read 1 ${loadaddr} 0x800 0x2000;mmc read 1
   ${rd_loadaddr} 0x3000 0x300;bootm ${loadaddr} ${rd_loadaddr}
```

User can refer to the settings below for more evaluation.

5.2.3.1. Setting the display devices

[VGA]

To set VGA as display output

```
setenv set_display 'run vga'
```

To set VGA resolution

```
Setenv vga 'setenv bootargs ${bootargs} di1_primary
video=mxcdi1fb:GBR24,VGA-XGA vga pmem=32M,64M fbmem=5M gpu_memory=64M'
```

To change the resolution, replace the red color part by the desire resolution mode.

For example: To set 1680 x 1050 resolution, change VGA-XGA to VGA-WSXGA+.

```
Setenv vga 'setenv bootargs ${bootargs} di1_primary
video=mxcdi1fb:GBR24, VGA-WSXGA+ vga pmem=32M,64M fbmem=5M
gpu_memory=64M'
```

Note:

There are four modes to be set in VAB-800:
VGA-WSXGA+ : 1680x1050p-60
VGA-SXGA : 1280x1024p-60
VGA-XGA : 1024x768p-60
VGA-SVGA : 800x600p-60

[HDMI]

To set HDMI as display output

```
setenv set_display 'run hdmi_1080p'
```

HDMI supports the following modes:

1920x1080, 1280 x 720, 640 x 480,

To set HDMI resolution

```
setenv hdmi_1080p 'setenv bootargs ${bootargs} di0_primary
video=mxcdi0fb:RGB24,1920x1080M@60 hdmi pmem=128M,64M fbmem=14M
gpu_memory=128M dmfc=3 vmalloc=576M'
```

To change the resolution, replace the red color part by the desire resolution mode.

For example: To set 1920 x 1080, change 1280 x 720 to1920 x 1080.

```
setenv hdmi ‘setenv hdmi 'setenv bootargs ${bootargs} di0_primary
video=mxcdi0fb:RGB24,1280x720@60 hdmi gpu_nommu  gpu_memory=64M’
```

[LVDS]

To set LVDS as display output

```
setenv set_display ‘run lvds1’
```

or

```
setenv set_display ‘run lvds2’
```

To set LVDS port

```
setenv lvds1 ‘setenv bootargs ${bootargs} di0_primary
video=mxcdi0fb:RGB24,WVGA ldb=di0 pmem=32M,64M fbmem=5M gpu_memory=64M’

setenv lvds2 ‘setenv bootargs ${bootargs} di1_primary
video=mxcdi1fb:RGB24,WVGA ldb=di1 pmem=32M,64M fbmem=5M gpu_memory=64M’
```

Note:

VAB-800 supports AUO 7" WVGA (800x480) LVDS panel in default.

5.2.3.2. Setting the miscellaneous

[Touch Panel]

To set LVDS1 with touch port

```
setenv touch ‘setenv bootargs ${bootargs} di0_primary
video=mxcdi0fb:RGB24,WVGA ldb=di0 pmem=32M,64M fbmem=5M gpu_memory=64M
calibration’
```

To set LVDS2 with touch port

```
setenv touch ‘setenv bootargs ${bootargs} di1_primary
video=mxcdi1fb:RGB24,WVGA ldb=di1 pmem=32M,64M fbmem=5M gpu_memory=64M
calibration’
```

To set which LVDS with touch as display output

```
setenv set_display ‘run touch’
```

[MAC Address]

Setting MAC address

Here, user can set the MAC address in u-boot. The address is on the Ethernet physical port.

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

It is a must to reset after you set MAC address and save it:

```
saveenv
reset
```

5.3. Testing

This chapter will introduce you how to evaluate VAB-800 Android evaluation image.

5.3.1. Installing APK through file manager

An **"open Manager"** file manager is prebuilt into image. User can install necessary testing APK.

Note:

There are known copy/paste issues in open Manager, it is recommended to use it only when installing APK.

Install APK from SD card (Boot from eMMC)

1. Open "Open Manager"
2. Choose "extsd" folder
3. Select the app you wish to install, the program will ask you "Do you want to install this application?", press "Install".

Install APK from USB disk

1. Open "Open Manager"
2. Choose "udisk" folder
3. Select the app you wish to install, the program will ask you "Do you want to install this application?", press "Install".

5.3.2. Installing APK through Android SDK (optional)

[Host PC: Windows XP]

Step 1

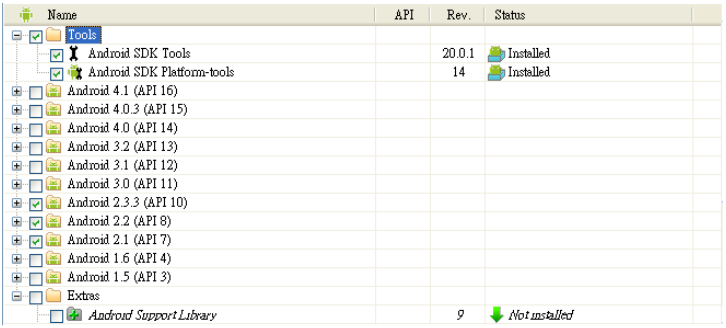
Download JDK (**jdk-7u6-windows-i586.exe**) from **http://www.oracle.com/technetwork/java/javase/downloads/index.html** and install it.

Step 2

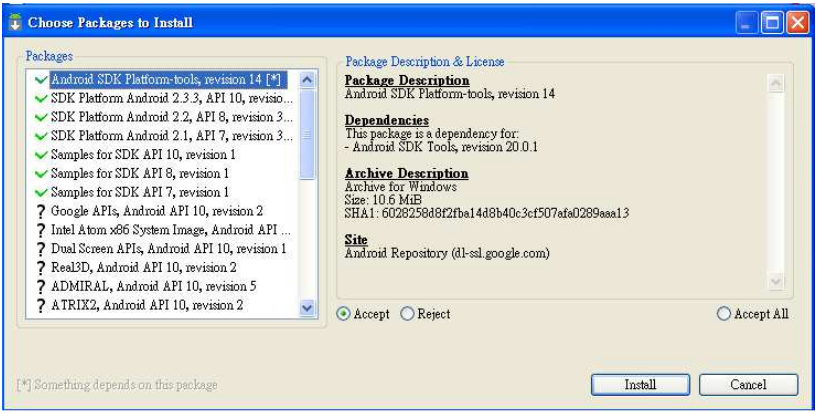
Download Android SDK (installer_r20.0.1-windows.exe) and install it.

Step 3

Execute SDK manager and select packages:



After download process, you will see:



Install it.

Step 4

Open a text console utility in XP.

Run -> cmd

```
cd <<Your SDK Installation Path>>\Android\android-sdk\tools\
#run command in the console
android.bat update adb
```

Step 5

Go to C:\Documents and Settings\ {User Account} \.android

Modify the file adb_usb.ini:

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

Step 6

Unpack android_usb_fsl.zip from Tools under EVK folder.

Step 7

Connect your PC with VAB-800. XP will detect a new android device. Install driver by choosing android_winusb.ini which is in the android_usb_fsl folder

Run -> cmd

```
cd <<Your SDK Installation Path>>\Android\Android\android-sdk\platform-tools
#run below command in the consle
adb.exe kill-server
adb.exe start-server
adb.exe devices
```

You will see as below:

List of devices attached: 0123456789ABCDEF device

[Host PC: Linux]

Step 1

Download JDK6 package and install it.

Step 2

Download the Android SDK.

Step 3

Update the adb configuration to scan for freescale's pid:

- Run the SDK's tools to generate a configuration 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. 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 by USB OTG

To ensure that Android device is connected, first, find adb in android-sdk/platform-tools

```
./adb devices
```

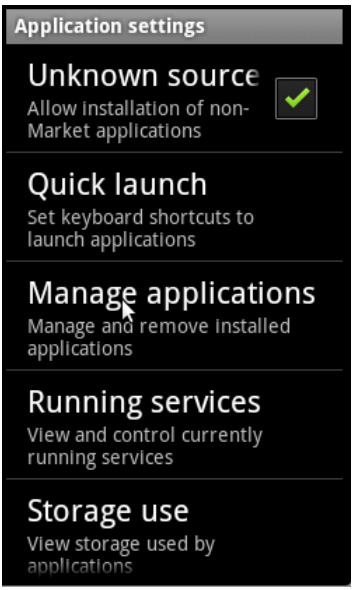
List of devices attached: 0123456789ABCDEF device

Step 4

Install APK through adb

After setup, you can use adb to install app.

Go to Settings->Application Settings and enable Unknown Sources



adb install {your apk file}

5.3.3. Testing Items

5.3.3.1. openGL ES 2.0

- Install ApiDemo.apk gotten from EVK/app folder
- Run ApiDemo
- Select Graphics-> OpenGL ES->OpenGL ES 2.0

5.3.3.2. OpenVG

OpenVG is not originally supported by Android; however, it is possible to use OpenVG in Android by downloading AndroidVG. Get more detailed information on:

http://code.google.com/p/androidvg/wiki/Installation_Guide_Linux

To install "AppleVG.apk", "StartVG.apk" and "TigerVG.apk" files into Android for testing.

5.3.3.3. Wired LAN Network

- Plug in the cable before booting
- To setup MAC address in U-boot (skip it, if user had already set it)

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

- After booting into Android, go to Settings->Wireless & networks and turn on the Ethernet.
- Run "Browser" and verify the network.
- Unplug the cable and plug it back immediately, check whether the network is still working or not.
- Unplug the cable and reboot, plug in again after booting and check the network.

5.3.3.4. SATA HDD

SATA HDD can only mount and operate file management through COM 2 debug port manually, most file manager APK cannot operate file management correctly, such as copy/paste…etc.

- Find SATA HDD under /dev/block as "sdx", x can be a,b,c…through COM 2 debug port.

```
ls /dev/block/
…
#Check whether there are sdx sdx1
sda
sda1
```

- Mount SATA HDD and operate file management.

```
mkdir /sdcard/sata
mount  -t {ext2 or other else}  /dev/block/sdX1  /sdcard/sata
busybox  cp  {any file}  /sdcard/sata
```

5.3.3.5. COM port

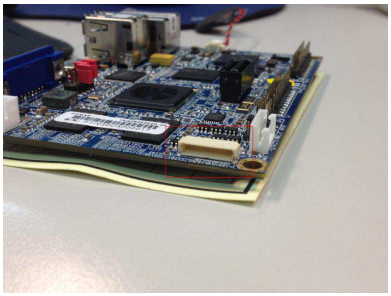
5.3.3.5.1. COM 1

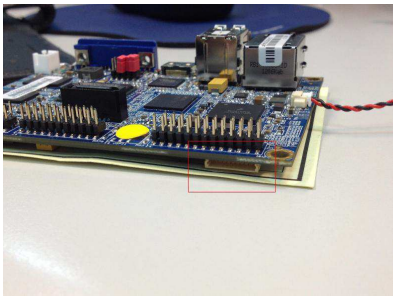
- Connect from VAB-800 to host PC via J5.
- Execute "putty" on host PC.
- Power on VAB-800.
- Setup baud rate of COM 1 and send characters to COM 1, the characters would be on the screen of the putty.

```
busybox stty –F /dev/ttymxc0 115200
echo VIA embedded > /dev/ttymxc0
```

Note:

You must type commands under COM 1. If you type in the terminal of Android, ">" can't be typed normally.

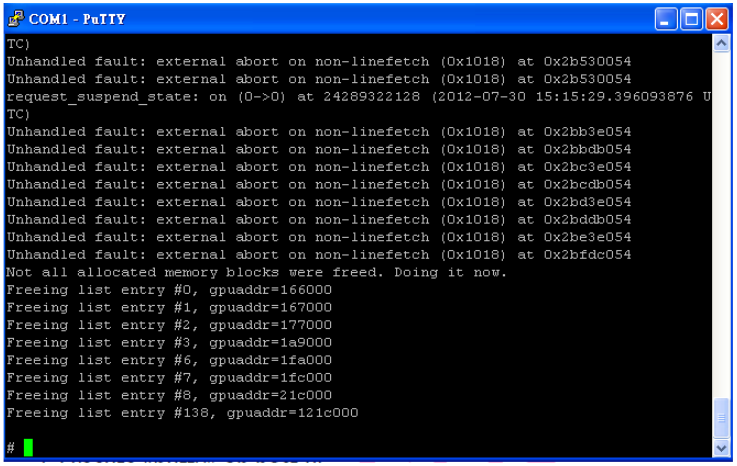




5.3.3.5.2. COM 2

- Connect from VAB-800 to host PC via J14.
- Execute "putty" on host PC.
- Power on VAB-800.
- Booting message would be on the screen of putty.

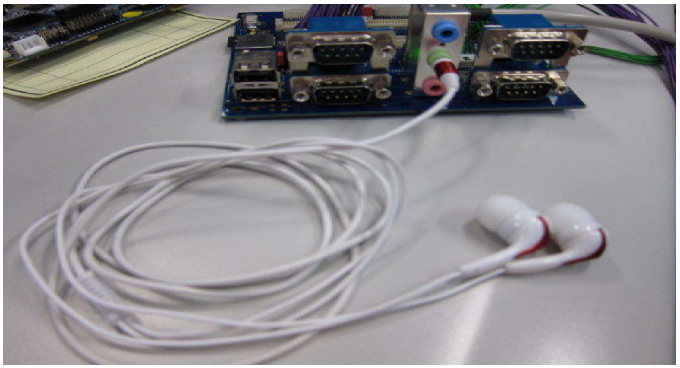
- PS. Putty setting: "Speed": 115200 "Connection type": serial
- Type command through COM 2.



5.3.3.6. Audio

5.3.3.6.1. Line-out

- Connect the headphone plug to the green jack socket.



- Open "MX player" app and play music.

5.3.3.6.2. MIC-in

- It is recommended to download Hi-Q MP3 Rec (Llite) 1.6.12
- Plug the microphone to the red jack socket.
- Run "Hi-Q" app in the Android to test sound record.

5.3.3.6.3. Line-in

- Plug audio cable to the blue jack socket. Use audio cable to connect.
- It is recommended to download Hi-Q MP3 Rec (Llite) 1.6.12 to test.
- Audio-out on player and Line-in on I/O board, and play the music.
- Line-in cannot be used directly. User needs alsa_amixer to set.

```
alsa_amixer contents
```

```
# alsa_amixer contents
numid=5,iface=MIXER,name='Headphone Volume'
  ; type=INTEGER,access=rw------,values=2,min=0,max=127,step=0
  : values=127,127
numid=11,iface=MIXER,name='Line In Function'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'off'
  ; Item #1 'on'
  : values=0
numid=7,iface=MIXER,name='ADC Mux'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'MIC_IN'
  ; Item #1 'LINE_IN'
  : values=0
numid=3,iface=MIXER,name='Capture Vol Reduction'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'No Change'
  ; Item #1 'Reduced by 6dB'
  : values=0
numid=2,iface=MIXER,name='Capture Volume'
```

```
alsa_amixer cset numid=11  1
alsa_amixer cset numid=7  1
```

- Run "Hi-Q" app

5.3.3.6.4. Echo while recording

- Plug headphone to green jack socket and microphone to red jack socket

```
alsa_amixer cset numid=6 1
alsa_amixer cset numid=11 1
```

```
numid=10,iface=MIXER,name='Speaker Function'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'off'
  ; Item #1 'on'
  : values=1
# alsa_amixer cset numid=6 1
numid=6,iface=MIXER,name='DAC Mux'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'DAC'
  ; Item #1 'LINE_IN'
  : values=1
# alsa_amixer cset numid=11 1
numid=11,iface=MIXER,name='Line In Function'
  ; type=ENUMERATED,access=rw------,values=1,items=2
  ; Item #0 'off'
  ; Item #1 'on'
  : values=1
# Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bc07054
Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bc07054
Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bca4054
Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bc07054
Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bc07054
Unhandled fault: external abort on non-linefetch (0x1018) at 0x2bca4054
```

- Open "Hi-Q" app. You can hear what you speak while recording.

5.3.3.7. I²C

5.3.3.7.1. Detect light sensor device

- Light sensor device will be found at address 0x10 and 0x11 on I2C2.

```
# i2cdetect 2
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-2.
I will probe address range 0x03-0x77.
Continue? [Y/n] y
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: 10 11 -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
#
```

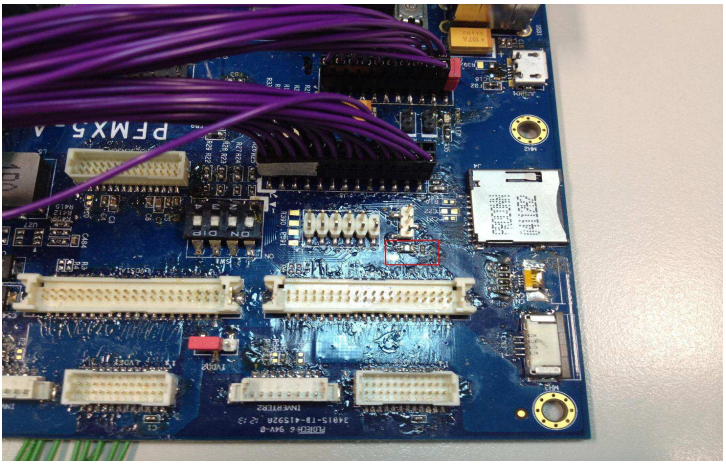
5.3.3.7.2. Read the register value from light sensor

- get the register value without light

- get the register value with light

```
# i2cget 2 0x10
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will read from device file /dev/i2c-2, chip address 0x10, current data
address, using read byte.
Continue? [Y/n] y
0x00
# i2cget 2 0x10
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will read from device file /dev/i2c-2, chip address 0x10, current data
address, using read byte.
Continue? [Y/n]
0x0c
#
```

The light sensor is located on the I/O board; please see the following picture for detail.



5.3.3.8. GPIO

- Change directory to /sys/class/gpio
- Generate the numbers corresponding to the GPIO pins

pin	VAB-800 name	FSL name
C7	GPI_2	GPIO1_2
C6	GPI_16	GPIO7_11
D7	GPI_18	GPIO7_13
B4	GPI_19	GPIO4_5
E14	GPI_10	GPIO1_10
C15	GPI_11	GPIO1_11
E13	GPI_12	GPIO1_12
D14	GPI_13	GPIO1_13

The number is (X-1)*32+Y for each GPIOX_Y.

Ex: The number for GPIO7_11 is (7-1)*32+11=203

The number for GPIO1_11 is (1-1)*32+11=11

```
echo 203 > export
echo 11 > export
```

The directories "gpio203" and "gpio11" will be created under

/sys/class/gpio/

- Set GPIO IN/OUT

```
echo in > gpio203/direction
echo out > gpio11/direction
```

Notes: 10, 11, 12, and 13 are out. 2, 203, 205 and 101 are in.

- Test

The following GPIO is connected by jump, we can test them together.

gpio101 --- gpio13

gpio205 --- gpio12

gpio203 --- gpio11

gpio2 --- gpio10

Write value 0 or 1 to "out" and read value from "in"

```
echo 0 > /sys/class/gpio/gpio203/value
cat /sys/class/gpio/gpio11/value
```

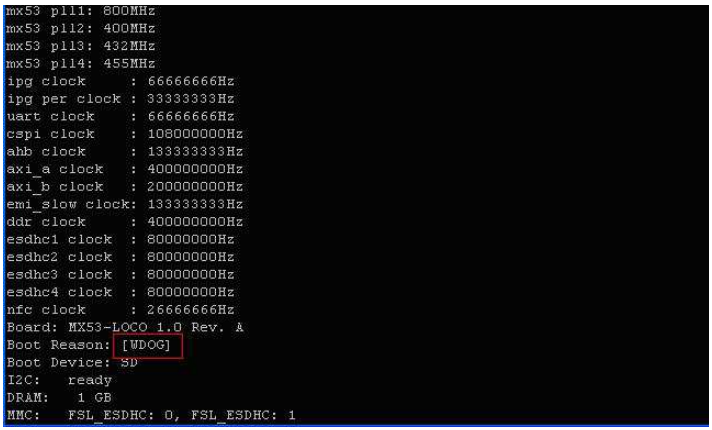
Check if the value is the same as you wrote it.

5.3.4. Watchdog

```
ioctl –r /dev/watchdog 2
```

The system will reboot in 60 seconds

[WDOG] will be shown at the line of "Boot Reason"



5.3.4.1. Video

- To install MX player
- Select video file to play. MX player will use hardware acceleration to decode the video by default and show "HW" at the up-right corner. If the file cannot be decoded by hw, player will decode by sw and show "SW" at the up-right corner.

5.3.4.2. RTC

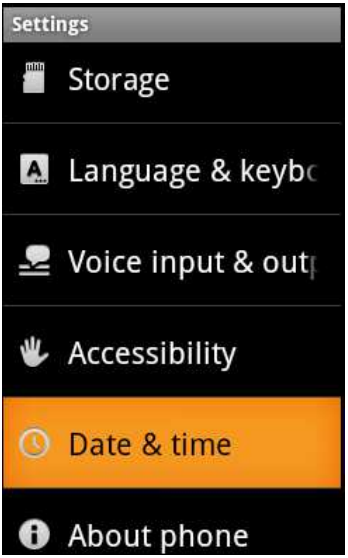
- Test way 1: Setup hardware time by command line

```
busybox date 073017002012
```

PS. 07 as month, 30 as date, 17 as hour(24H), 00 as minute and 2012 as year

```
busybox hwclock  -w  -f  /dev/rtc0
```

- Test way 2: Setup hardware time in Settings



- Reboot the system (turn off the power and then turn it on) and check the time

5.3.4.3. USB

5.3.4.3.1. USB storage

Android will detect USB disk automatically. Once the USB disk is detected, system will inform you. You can also mount the USB disk manually.

- Go to Settings->Storage
- Select "Mount U disk card"

5.3.4.3.2. USB OTG

Connect Android device to Linux or XP PC through USB OTG. Go to android-sdk/platform-tools and type:

```
./adb devices
```

If the device is detected successfully, terminal will show a message:

```
List of devices attached: 0123456789ABCDEF device
```

5.3.4.4. SD storage

SD card as storage: Detection and Access

- Find SD card under /dev/block as "mmcblk1"
- Android will detect SD card automatically although it may take several seconds to perform it. You can still mount it manually.
- Go to "Settings->Storage" and select "Mount EXTSD card". Then you can go to "Gallery" to find photos or video files which are stored in SD card.

5.3.5. Touch panel

- Connect touch panel to VAB-800-A and boot it, go to u-boot.
- Be sure the touch panel had been set in u-boot.

For example: to set LVDS1 with touch port.

```
setenv touch 'setenv bootargs ${bootargs} di0_primary
video=mxcdi0fb:RGB24,WVGA ldb=di0 pmem=32M,64M fbmem=5M gpu_memory=64M
calibration'
```

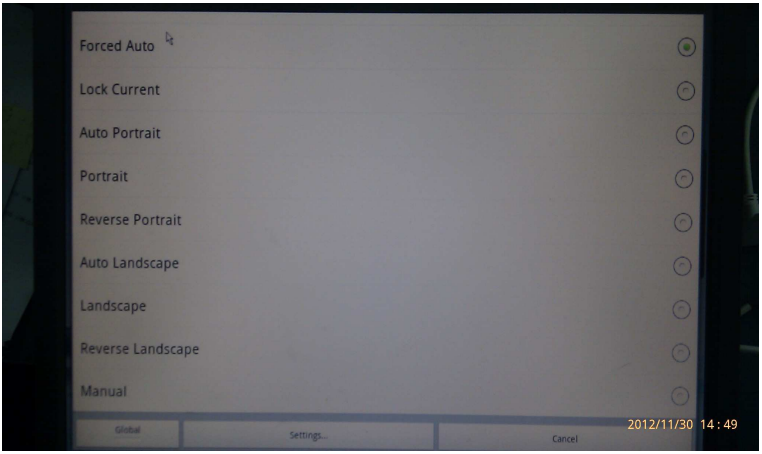
To set which LVDS with touch as display output

```
setenv set_display 'run touch'
saveenv
boot
```

- When calibrating, '+' will be displayed on the screen, use your fingertip to touch it. This will repeat 3 times.
- Now you can surf the Android by your finger.

5.3.5.1. Rotate testing manually

There is no G-Sensor to do auto-rotate on VAB-800. Rotate_4.6.2.apk has to be installed before testing it manually.


Forced Auto
Lock Current
Auto Portrait
Portrait
Reverse Portrait
Auto Landscape
Landscape
Reverse Landscape
Manual
Global
Settings...
Cancel
2012/11/30 14:49

Appendix A. Starter Kit

A.1. Starter Kit Assembly

The starter kit includes the following items:

- ☐ 1 x VAB-800
- ☐ 1 x VAB-800-A
- ☐ 1 x 7" LCD display with 4-wire resistive touch screen
- ☐ 1 x DC-in cable
- ☐ 1 x LVDS cable
- ☐ 1 x COM cable

To install the VAB-800-A to the VAB-800 mainboard, align and attach the board-to-board connectors (J3, J1 & J2) on the bottom of VAB-800-A with the pin headers (F_audio, J9 & J10 respectively) on the top of VAB-800 mainboard.

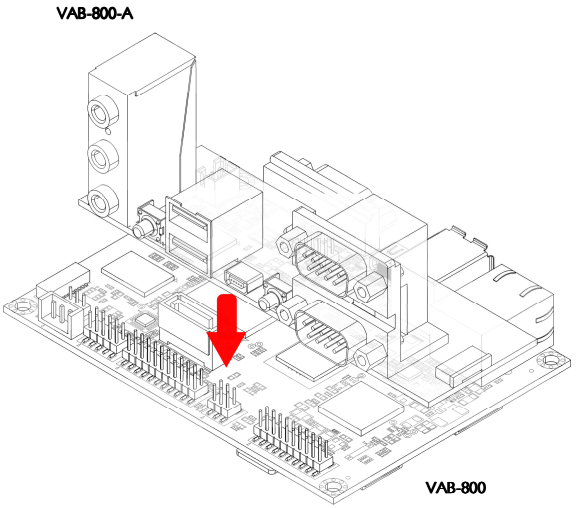


Figure 32: Connecting VAB-800-A to VAB-800 mainboard

The figure below shows the connection of panel.

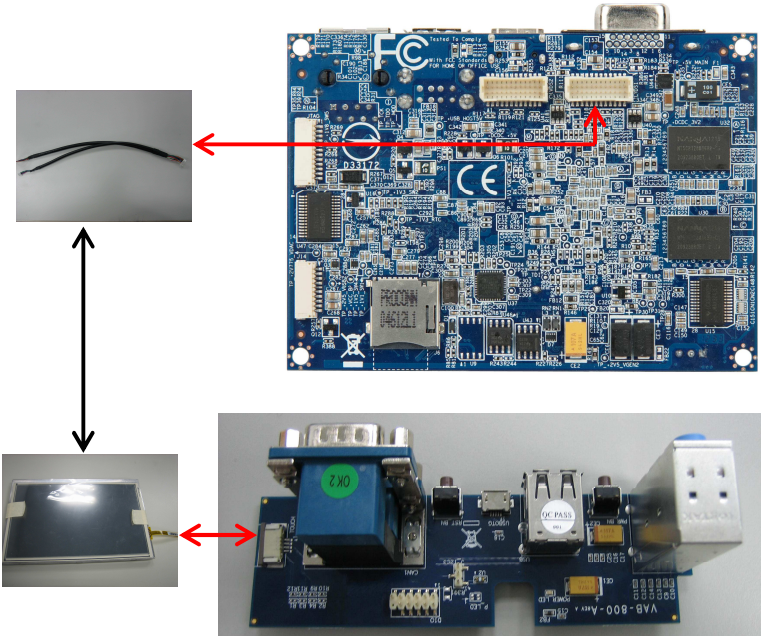


Figure 33: Connection of panel

A.2. VAB-800-A Specifications

- **Onboard I/O**
  - 1 x I²C pin header
  - 1 x DPIO pin header (4 Ins and 4 OUTS)
  - 1 x 4-wire connector for the resistive touch screen

- **Back Panel I/O**
  - 1 x Stack type 3 ports audio I/O connector (support Line-out, Line-in and MIC-in)
  - 1 x Power button
  - 1 x Stack type 2 ports USB2.0 connector
  - 1 x USBOTG connector for USB device
  - 1 x Reset button
  - 1 x Stack type 2 ports CAN D-sub connector

- **Form Factor**
  - 4-layers
  - 11.3 cm x 3.5 cm

A.3. VAB-800-A Layout

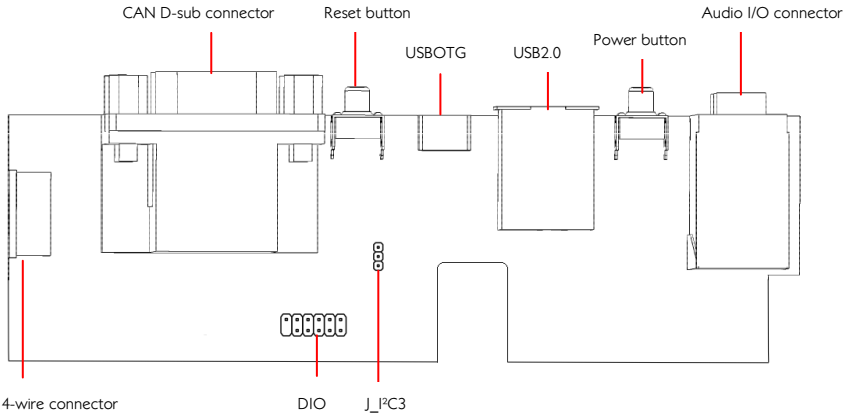


Figure 34: VAB-800-A Layout (top view)

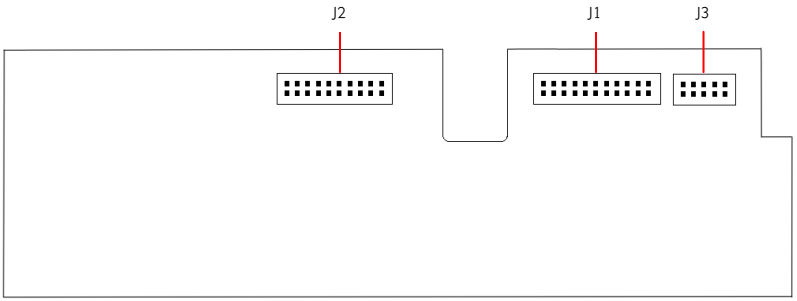


Figure 35: VAB-800-A Layout (bottom view)

A.4. VAB-800-A Pinouts and Jumpers

DIO

Pin	Signal	Pin	Signal
1	NC	2	--
3	GPO_10	4	GPI_2
5	GPO_11	6	GPI_16
7	GPO_12	8	GPI_18
9	GPO_13	10	GPI_19
11	GND		

J_I²C3

Pin	Signal	Pin	Signal
1	I2C3_SCL	2	I2C3_SDA
3	GND		

4-wire resistive touch

Pin	Signal
1	TOPUCH_X+
2	TOPUCH_Y+
3	TOPUCH_X-
4	TOPUCH_Y-

Appendix B. Mating Connector Vendor Lists

The following table listed the mating connector vendor lists of VAB-800 mainboard.

Connectors	Vendor & P/N		Mating Vendor & P/N	
COM1	Neltron	1600R-XX-SM-TR	JST	SHR-10V-S-B
J9	Neltron	2208SM-XXG-EXX-XX-CR	Samtec	MMS-1XX-01-XX-DV series
J10	Neltron	2208SM-XXG-EXX-XX-CR	Samtec	MMS-1XX-01-XX-DV series
DC5V	Neltron	2317SJ-XX-F4	JST	XHP-2
JTAG	Neltron	1600R-XX-SM-TR	JST	SHR-12V-S-B
LVDS	ACES	87216-2416-06	ACES	87219-2400

Table 24: VAB-800 mating connector vendor lists