Our Products Make Your Product Better®

To learn more about EMAC's products and services and how they can help your project

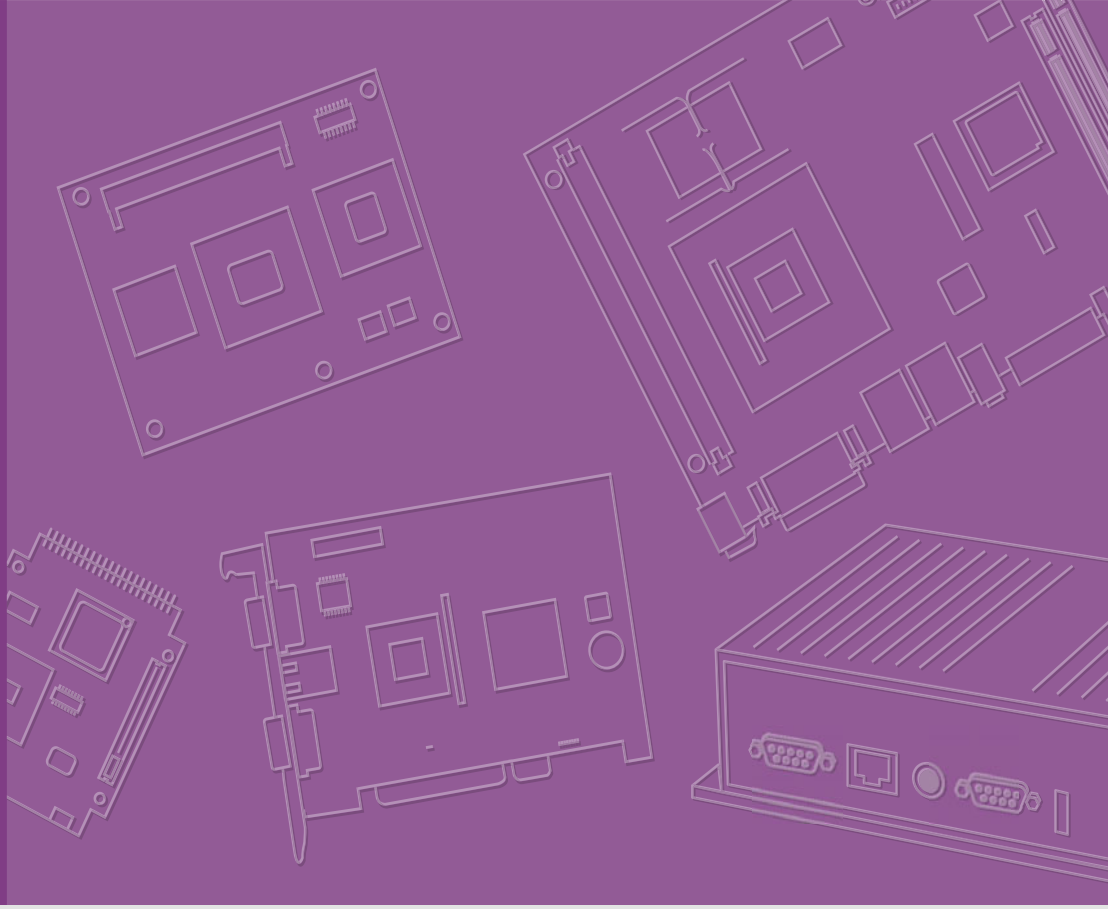
http://ftp.emacinc.com/Tech_Info/About_EMAC_Products_and_Services.pdf

Authorized Distributor, Integrator, and Value-Added Reseller

Manual downloaded from ftp.emacinc.com

For purchase information please contact info@emacinc.com

For technical support please submit a ticket at www.emacinc.com/support



User Manual

RSB-4411

3.5" SBC with NXP i.MX6 Processor ARM® Cortex™ A9 Architecture

Copyright

The documentation and the software included with this product are copyrighted 2017 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.

Acknowledgements

Intel and Pentium are trademarks of Intel Corporation.

Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corp.

All other product names or trademarks are properties of their respective owners.

Product Warranty (2 years)

Advantech 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 Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

Because of Advantech's high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product 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 speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, obtain an RMA (return merchandize authorization) number from your dealer. This allows us to process your return more quickly.
4. Carefully pack the defective product, a fully-completed 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.

Part No. XXXXXXXXXX Edition 1

Printed in Taiwan June 2017

Declaration of Conformity

CE

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

CE

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class A

Note: 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.

FCC Class B

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FM

This equipment has passed the FM certification. According to the National Fire Protection Association, work sites are classified into different classes, divisions and groups, based on hazard considerations. This equipment is compliant with the specifications of Class I, Division 2, Groups A, B, C and D indoor hazards.

Technical Support and Assistance

1. Visit the Advantech website at http://support.advantech.com where you can find the latest information about the product.
2. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance. Please have the following information ready before you call:
   - Product name and serial number
   - Description of your peripheral attachments
   - Description of your software (operating system, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

Warnings, Cautions and Notes

Warning! *Warnings indicate conditions, which if not observed, can cause personal injury!*

Caution! *Cautions are included to help you avoid damaging hardware or losing data. e.g.*

There is a danger of a new battery exploding if it is incorrectly installed. Do not attempt to recharge, force open, or heat the battery. Replace the battery only with the same or equivalent type recommended by the manufacturer. Discard used batteries according to the manufacturer's instructions.

Note! *Notes provide optional additional information.*

Document Feedback

To assist us in making improvements to this manual, we would welcome comments and constructive criticism. Please send all such - in writing to: support@advantech.com

Packing List

Before setting up the system, check that the items listed below are included and in good condition. If any item does not accord with the table, please contact your dealer immediately.

- Item XXXXXXXX
- Box XXXXXXXXX

Safety Instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Disconnect this equipment from any AC outlet before cleaning. Use a damp cloth. Do not use liquid or spray detergents for cleaning.
4. For plug-in equipment, the power outlet socket must be located near the equipment and must be easily accessible.
5. Keep this equipment away from humidity.
6. Put this equipment on a reliable surface during installation. Dropping it or letting it fall may cause damage.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. DO NOT COVER THE OPENINGS.
8. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
9. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
10. All cautions and warnings on the equipment should be noted.
11. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient overvoltage.
12. Never pour any liquid into an opening. This may cause fire or electrical shock.
13. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
14. If one of the following situations arises, get the equipment checked by service personnel:
    - The power cord or plug is damaged.
    - Liquid has penetrated into the equipment.
    - The equipment has been exposed to moisture.
    - The equipment does not work well, or you cannot get it to work according to the user's manual.
    - The equipment has been dropped and damaged.
    - The equipment has obvious signs of breakage.
15. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -20° C (-4° F) OR ABOVE 60° C (140° F). THIS COULD DAMAGE THE EQUIPMENT. THE EQUIPMENT SHOULD BE IN A CONTROLLED ENVIRONMENT.
16. CAUTION: DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH THE SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER, DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS.

The sound pressure level at the operator's position according to IEC 704-1:1982 is no more than 70 dB (A).

DISCLAIMER: This set of instructions is given according to IEC 704-1. Advantech disclaims all responsibility for the accuracy of any statements contained herein.

Safety Precaution - Static Electricity

Follow these simple precautions to protect yourself from harm and the products from damage.

- To avoid electrical shock, always disconnect the power from your PC chassis before you work on it. Don't touch any components on the CPU card or other cards while the PC is on.
- Disconnect power before making any configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

Contents

Chapter 1

General Introduction

1.1 Introduction

RSB-4411 is a RISC 3.5" single board computer (SBC) powered by a high-performance NXP ARM® Cortex®-A9 i.MX6 processor that supports full HD video encoding/decoding and Gigabit Ethernet networking. RSB-4411 also features mini PCIe, M.2, and SIM card slot for integrating Wi-Fi, Bluetooth, and 3G modules. Equipped with complete Linux and Android BSPs, this system enables customers to easily develop unique application software for specific OS.

1.2 Specifications

1.2.1 Functional Specifications

Processor: Freescale i.mx6 Series

- ARM Cortex™-A9 high performance processor, Dual core 1 GHz
- Supports 2 IPU, OpenGL ES 2.0 for 3D BitBLT for 2D and OpenVG™ 1.1
- Video decoder: MPEG-4 ASP, H.264 HP, H.263, MPEG-2 MP, MJPEG BP
- Video Encoder: MPEG-4 SP, H.264 BP, H.263, MJPEG BP

System Memory Support

- DDR3 1066 MHz
- Capacity: on board DDR3 1 GB

Gigabit Ethernet

- Chipset: Freescale i.MX6 integrated RGMII
- 1 x10/100/1000 Mbps

Peripheral Interface

- 1 x dual channel 18/24 bit LVDS
- 1 x HDMI
- 1 x VGA
- 1 x USB OTG, 2 x USB Type A, and 3 x USB pin headers
- 1 x Line Out
- 1 x Mic In
- 1 x SD slot
- 2 x 2 wires RS-232 pin header, 1x 4-wire RS-232/422/485 DB9 Connector
- 2 x CAN
- 1 x miniPCIe slot
- 1 x M.2 slot
- 1 x SIM slot
- 1 x SPI
- 2 x I2C
- 1 x SATA
- 1 x MIPI CSI
- 20 x GPIO

OS Support

- RSB-4411 supports Yocto Linux

1.2.2 Mechanical Specifications

- Dimension: 146 x 102 mm (5.7”x4”)
- Height: 15.92 mm
- Reference Weight: 640 g (including whole package)

1.2.3 Electrical Specifications

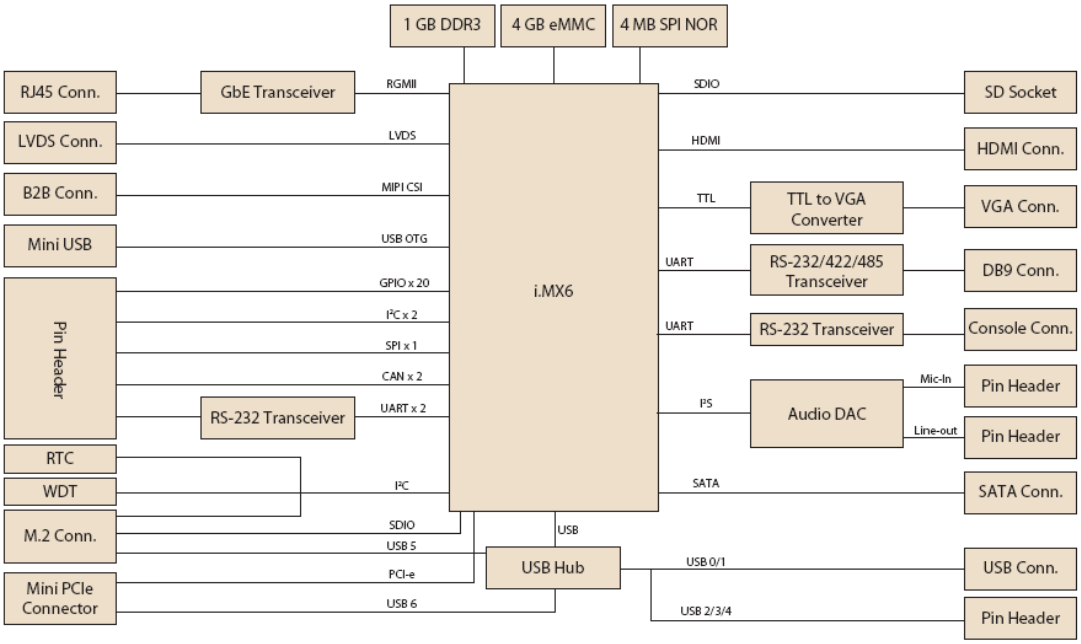
- Power supply type: DC-in 12 V/ 19V /24V
- Power consumption:
  - Kernel Idle mode: TBD
  - Max mode: TBD
- RTC Battery:
  - Typical voltage: 3.0 V
  - Normal discharge capacity: TBD

1.3 Environmental Specifications

- Operating temperature: 0 ~ 60° C (32 ~ 140° F)
- Operating humidity: 40° C @ 95% RH Non-condensing
- Storage temperature: -40 ~ 85° C (-40 ~ 185° F)
- Storage humidity: 60° C @ 95% RH Non-condensing

1.4 Block Diagram

Block Diagram



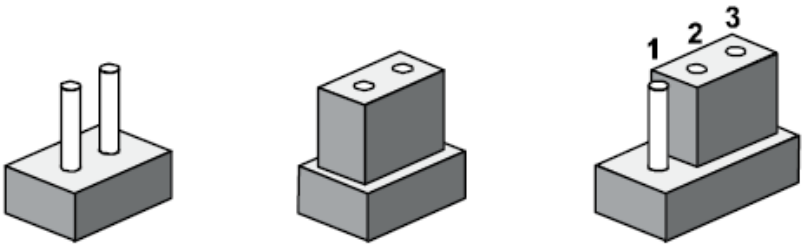
Chapter 2

H/W Installation

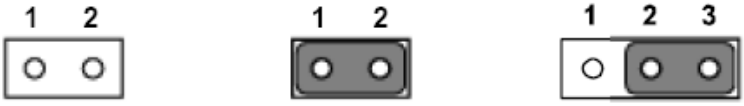
2.1 Jumpers

2.1.1 Jumper Description

Cards can configured by setting jumpers. A jumper is a metal bridge used to close an electric circuit. 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 pins 1 and 2 or 2 and 3.



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



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

Generally, you simply need a standard cable to make most connections.

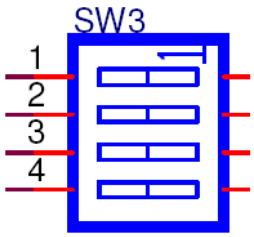
2.1.2 Jumper List

Table 2.1: Jumper List	
LVDS_VDD_SLT	LVDS Power
LVDS_BKLT_SLT	Backlight Power
SW2	Boot device
SW3	RS-232/422/485 selection

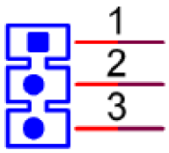
2.1.3 Jumper Settings

SW2	Boot device
Part number	1600000202
Description	DIP SW CHS-02TB(29) SMD 4P SPST P=1.27mm W=5.4mm
Setting	Function
1 ON 2 OFF	Boot from SD
1 OFF 2 ON	Boot from SPI

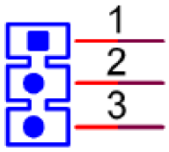
SW3	RS-232/422/485 selection
Part number	1600000084
Description	DIP SW CHS-02TB(29) SMD 4P SPST P=1.27mm W=5.4mm
Setting	Function
1 OFF 2 OFF	Loopback mode
1 OFF 2 ON	RS-232
1 ON 2 OFF	RS-485 Half Duplex
1 ON 2 ON	RS-422 Full Duplex



LVDS_VDD_SLT	LVDS Power
Part number	1653003101
Description	PIN HEADER 3x1P 2.0mm 180D(M) DIP 2000-13 WS
Setting	Function
(1-2)	+ V3.3
(2-3)	+ V5



LVDS_BKLT_SLT	LVDS Backlight Power
Part number	1653003101
Description	PIN HEADER 3x1P 2.0mm 180D(M) DIP 2000-13 WS
Setting	Function
(1-2)	+ V5
(2-3)	+ V12



2.2 Connectors

2.2.1 Connector List

Table 2.2: Connector List	
CN1	RTC Battery
CN10	DC Power Jack
CRT1	VGA
COM1	RS-232/422/485
CN8	Ethernet Connector
CN11	HDMI
CN28	USB port 0/1
CN30	USB OTG
LVDS0	LVDS
LVDS_BKLT_PWR	Backlight
CN27	SATA
CN25	SATA Power
CN5	Debug Port
CN12	USB port 3/4
CN29	USB port 5
CN23	MIC in
CN14	Line Out
CN21	RS-232
CN31	GPIO
CN32	CAN
CN33	I2C_1
CN34	I2C_3
CN35	SPI
CN3	MiniPCIe
CN4	SIM socket
SD1	SD socket
CN22	M.2

2.2.2 Connector Settings

2.2.2.1 RTC Battery connectors (CN1)

RSB-4411 supports a lithium 3V/210mAH CR2032 battery with wire via battery connector.

2.2.2.2 DC power Jack (CN10)

RSB-4411 comes with a DC-Jack header that carries 12V/19V/24V DC external power input.

Pin	Description
1(Inner)	DC_In
2(Outer)	GND

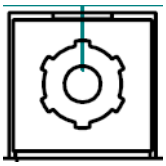


Figure 2.1 DC Power Jack

2.2.2.3 VGA (CRT1)

RSB-4411 provides standard VGA connector. VGA resolution supports up to 1920x1080.

Pin	Description
1	RED
2	GREEN
3	BLUE
4	N/C
5	GND
6	GND
7	GND
8	GND
9	+5V
10	GND
11	N/C
12	DDC DATA
13	HSYNC
14	VSYNC
15	DDC CLK

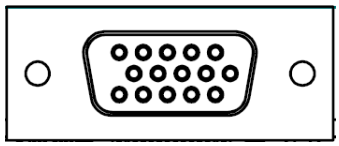


Figure 2.2 VGA Connector

2.2.2.4 RS-232/422/485 (COM1)

RSB-4411 provides one D-Sub 9-pin connector serial communication interface port. The port can support RS-232/422/485 mode communication.

Pin	Description		
1	N/C	RS-422 TX-	RS-485-
2	COM 2_RXD	RS-422 TX+	RS-485+
3	COM2_TXD	RS-422 RX+	
4	N/C	RS-422 RX-	
5	GND		
6	N/C		
7	COM2_RTS		
8	COM2_CTS		
9	N/C		

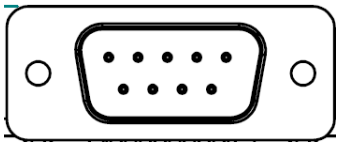


Figure 2.3 COM Port

2.2.2.5 Ethernet Connector (CN8)

RSB-4411 provides one RJ45 LAN interface connector; it is fully compliant with IEEE802.3u 10/100/1000 Base-T CSMA/CD standards. The Ethernet port provides standard RJ-45 jack connector with LED indicators on the front side to show Active/ Link status and Speed status.

Pin	Description
1	MIDI0+
2	MIDI0-
3	MIDI1+
4	MIDI1-
5	GND
6	GND
7	MIDI2+
8	MIDI2-
9	MIDI3+
10	MIDI3-
11	VCC
12	ACT
13	Link100#
14	Link1000#

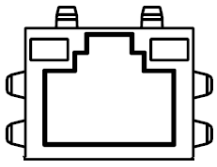


Figure 2.4 Ethernet Connector

2.2.2.6 HDMI (CN11)

RSB-4411 provides one HDMI interface connector which provides all digital audio/video interfaces to transmit the uncompressed audio/video signals and is HDCP and CEC compliant Connect the HDMI audio/video device to this port. HDMI technology can support a maximum resolution of 1920 x 1080p but the actual resolution supported depends on the monitor being used.

Pin	Description
1	HDMI_TD2+
2	GND
3	HDMI_TD2-
4	HDMI_TD1+
5	GND
6	HDMI_TD1-
7	HDMI_TD0+
8	GND
9	HDMI_TD0-
10	HDMI_CLK+
11	GND
12	HDMI_CLK-
13	HDMI_CEC_A
14	GND
15	DDC_CLK_HDMI_A
16	DDC_DATA_HDMI_A
17	GND
18	+5V
19	HDMI_HPD

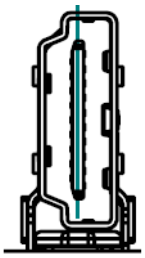


Figure 2.5 HDMI

2.2.2.7 USB Connector (CN11)

RSB-4411 supports one standard USB2.0 Type A connector in the coastline.

Pin	Description
1	+5V
2	USB1_D-
3	USB1_D+
4	GND
5	+5V
6	USB2_D-
7	USB2_D+
8	GND

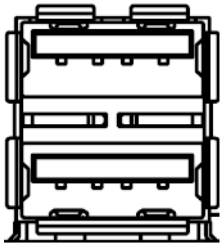


Figure 2.6 USB Type A Connector

2.2.2.8 USB OTG Connector (CN30)

RSB-4411 supports one USB OTG port in the coastline.

Pin	Description
1	+5V
2	USB1_D-
3	USB1_D+
4	ID
5	GND

Figure 2.7 USB OTG Connector

2.2.2.9 LVDS Connector (CN30)

RSB-4411 provides a LVDS 20x2-pin board-to-board connector for dual channel 18/24 bit LVDS panel up to 1920 x 1080. Please also refer to jumper setting before connecting LVDS panel.

Pin	Description
1	+VDD_LVDS
2	+VDD_LVDS
3	GND
4	GND
5	+VDD_LVDS
6	+VDD_LVDS
7	LVDS0_TX0_N
8	LVDS1_TX0_N

9	LVDS0_TX0_P
10	LVDS1_TX0_P
11	GND
12	GND
13	LVDS0_TX1_N
14	LVDS0_TX1_N
15	LVDS0_TX1_P
16	LVDS1_TX1_P
17	GND
18	GND
19	LVDS0_TX2_N
20	LVDS1_TX2_N
21	LVDS0_TX2_P
22	LVDS1_TX2_P
23	GND
24	GND
25	LVDS0_CLK_N
26	LVDS1_CLK_N
27	LVDS0_CLK_P
28	LVDS1_CLK_P
29	GND
30	GND
31	I2C1_SCL_LVDS0
32	I2C1_SDA_LVDS0
33	GND
34	GND
35	LVDS0_TX3_N
36	LVDS1_TX3_N
37	LVDS0_TX3_P
38	LVDS1_TX3_P
39	GND
40	+VDD_LVDS

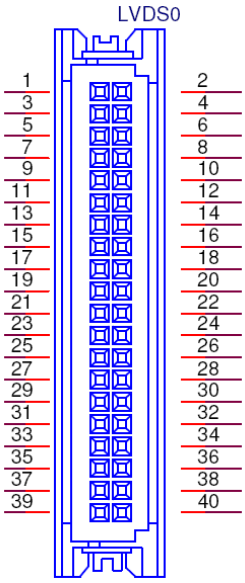


Figure 2.8 LVDS Connector

2.2.2.10 LVDS Backlight Connector (LVDS_BKLT_PWR)

Please also refer to jumper setting in page before connecting LVDS panel.

Pin	Description
1	+VDD_BKLT_LVDS
2	GND
3	LCD_BKLT_EN
4	LCD_BKLT_PWM
5	+V5

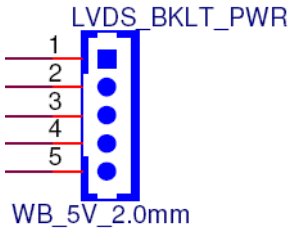


Figure 2.9 LVDS Backlight Power Connector

2.2.2.11 SATA (CN27)

RSB-4411 supports one SATA connector.

Pin	Description
1	GND
2	SATA_TX+
3	SATA_TX-
4	GND
5	SATA_RX-
6	SATA_RX+
7	GND

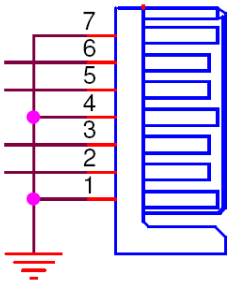


Figure 2.10 SATA Connector

2.2.2.12 SATA Power (CN25)

RSB-4411 supports one SATA Power connector.

Pin	Description
1	+5V
2	GND
3	GND
4	+12V

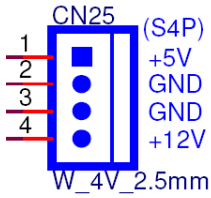


Figure 2.11 SATA Power Connector

2.2.2.13 Debug Port (CN5)

RSB-4411 can communicate with a host server (Windows or Linux) by using serial cables.

Pin	Description
1	+3.3V
2	Debug_TXD
3	Debug_RXD
4	GND

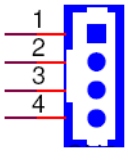


Figure 2.12 Debug Port

2.2.2.14 USB (Internal Pin Header) (CN12)

RSB-4411 provides extra internal 2x USB2.0 pin headers.

Pin	Description
1	+5V
2	+5V
3	USB4_D-
4	USB3_D-
5	USB4_D+
6	USB3_D+
7	GND
8	GND
9	GND

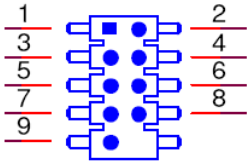


Figure 2.13 USB Internal Pin Header

2.2.2.15 USB (Internal Pin Header) (CN29)

RSB-4411 provides extra internal 1x USB2.0 pin headers.

Pin	Description
1	+5V
2	+5V
3	USB5_D-
4	NC
5	USB5_D+
6	NC
7	GND
8	GND
9	GND

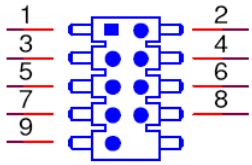


Figure 2.14 USB Internal Pin Header

2.2.2.16 MIC in (CN23)

RSB-4411 offers MIC in, microphone can be connected to the MIC in pin header.

Pin	Description
1	MIC IN
2	GND

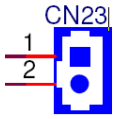


Figure 2.15 MIC in Internal Pin Header

2.2.2.17 Line out (CN14)

RSB-4411 offers Line-out stereo speakers; earphone can be connected to the lineout pin header.

Pin	Description
1	LINEOUT_L
2	LINEOUT_R
3	GND

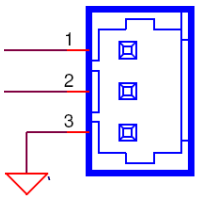


Figure 2.16 Line our Internal Pin Header

2.2.2.18 RS-232 (CN21)

RSB-4411 provides 2x internal 2 wires RS-232 ports.

Pin	Description
1	COM 3_RXD
2	GND
3	COM 3_TXD
4	COM 4_TXD
5	GND
6	COM 4_RXD

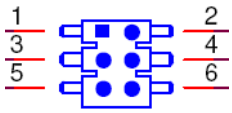


Figure 2.17 UART Pin Header

2.2.2.19 GPIO (CN31)

RSB-4411 provides internal GPIO interface by 2x11 pin headers.

Pin	Description
1	+3.3V
2	GND
3	GPIO_01
4	GPIO_02
5	GPIO_03
6	GPIO_04
7	GPIO_05
8	GPIO_06
9	GPIO_07
10	GPIO_08
11	GPIO_09
12	GPIO_10
13	GPIO_11
14	GPIO_12
15	GPIO_13
16	GPIO_14
17	GPIO_15
18	GPIO_16
19	GPIO_17
20	GPIO_18
21	GPIO_19
22	GPIO_20

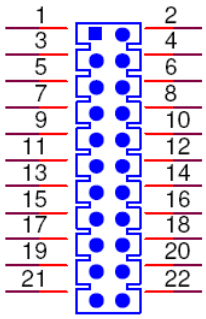


Figure 2.18 GPIO Internal Pin Header

2.2.2.20 GPIO (CN31)

RSB-4411 provides 2 x CAN bus by 2x3 pin headers.

Pin	Description
1	CAN1_D+
2	GND
3	CAN1_D-
4	CAN2_D-
5	GND
6	CAN2_D+

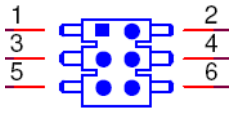


Figure 2.19 CAN Pin Header

2.2.2.21 I2C (CN33/CN34)

RSB-4411 provides 2 I2C pin headers.

Pin (CN33)	Description
1	GND
2	I2C1_SDA
3	I2C1_SCL
4	+V3.3

Pin (CN34)	Description
1	GND
2	I2C3_SDA
3	I2C3_SCL
4	+V3.3

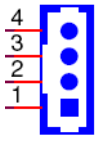


Figure 2.20 I2C Pin Headers

2.2.2.22 SPI (CN35)

RSB-4411 provides 1 SPI pin headers.

Pin	Description
1	+V3.3
2	GND
3	SPI_CS
4	SPI_CLK
5	SPI_MISO
6	SPI_MOSI

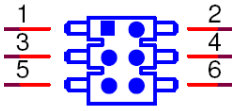


Figure 2.21 SPI Pin Headers

2.2.2.23 MiniPCIe (CN3)

RSB-4411 supports full size miniPCIe slot both USB and PCIe interface. If the WiFi card is only half-sized, please purchase extending bracket (P/N: 1960047454N000) for WiFi card fixing.

Pin	Description	Pin	Description
1	NC	2	3.3V
3	NC	4	GND
5	NC	6	NC
7	NC	8	UIM_PWR
9	GND	10	UIM_DATA
11	REFCLK-	12	UIM_CLK
13	REFCLK+	14	UIM_RESET
15	GND	16	NC
	Mechanical Key		
17	NC	18	GND
19	NC	20	3G_RF_OFF#
21	GND	22	WIFI_RESET#
23	PCIe_RXM	24	3.3V
25	PCIe_RXP	26	GND
27	GND	28	NC
29	GND	30	NC
31	PCIe_TXM	32	NC
33	PCIe_TXP	34	GND
35	GND	36	USD_D-
37	GND	38	UDB_D+
39	3.3V	40	GND
41	3.3V	42	WIMAX-3G_LED#
43	GND	44	WLAN_LED#
45	NC	46	NC
47	NC	48	NC
49	NC	50	GND
51	NC	52	3.3V

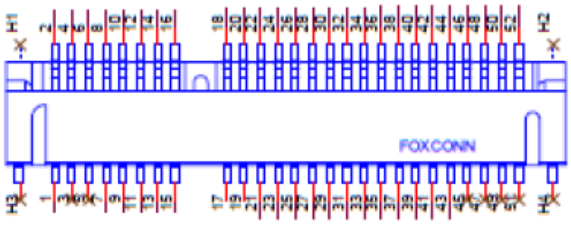


Figure 2.22 SPI miniPCIe

2.2.2.24 SIM Socket (CN4)

RSB-4411 supports on board SIM socket is for 3G integration. Please insert valid SIM card to dial to 3G network.

Pin	Description	Pin	Description
C1	UIM_PWR	C2	UIM_RESET
C3	UIM_CLK		
C5	GND	C6	NC
C7	UIM_DATA	SW1	NC
SW2	NC		

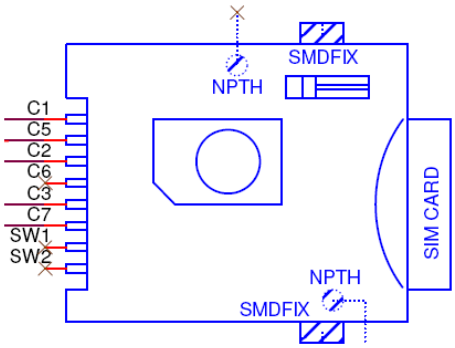


Figure 2.23 SIM Socket

2.2.2.25 SD Socket (SD1)

RSB-4411 supports SD/MMC card in Class2, 4, 6, 8, 10. Supported capacity is up to 32G(SDHC).

Pin	Description	Pin	Description
1	DAT3	2	CMD
3	GND	4	+3.3V
5	CLK	6	GND
7	DAT0	8	DAT1
9	DAT2		

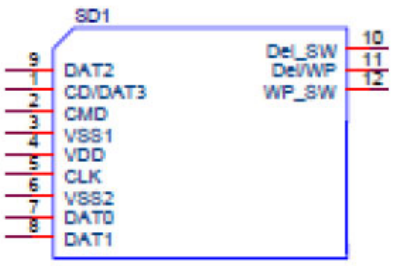


Figure 2.24 SD Slot

2.2.2.26 M.2 (CN3)

RSB-4411 supports M.2 2230 Key.E slot with SDIO,I2C and USB interface.

Pin	Description	Pin	Description
1	GND	2	+V3.3V
3	USB5_D+	4	+V3.3V
5	USB5_D-	6	M.2_WLAN_LED#
7	GND	8	NC
9	SD1_CLK	10	NC
11	SD1_CMD	12	NC
13	SD1_DATA0	14	NC
15	SD1_DATA1	16	M.2_BT_LED#
17	SD1_DATA2	18	GND
19	SD1_DATA3	20	NC
21	M2_SDIO_WAKE#	22	NC
23	SDIO_ RESET#		
	Mechanical Key		
		32	NC
33	GND	34	NC
35	NC	36	NC
37	NC	38	NC
39	GND	40	NC
41	NC	42	NC
43	NC	44	NC
45	GND	46	NC
47	NC	48	NC
49	NC	50	NC
51	GND	52	NC
53	NC	54	M.2_BT_X_OFF#
55	NC	56	M.2_WLAN_X_OFF#
57	GND	58	I2C1_SDA
59	NC	60	I2C1_SCL
61	NC	62	I2C1_ALERT#
63	GND	64	NC
65	NC	66	NC
67	NC	68	NC
69	GND	70	NC
71	NC	72	+3.3V
73	NC	74	+3.3V
75	GND		

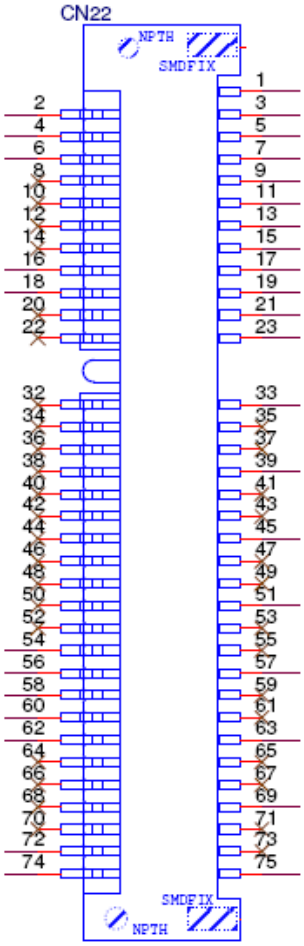
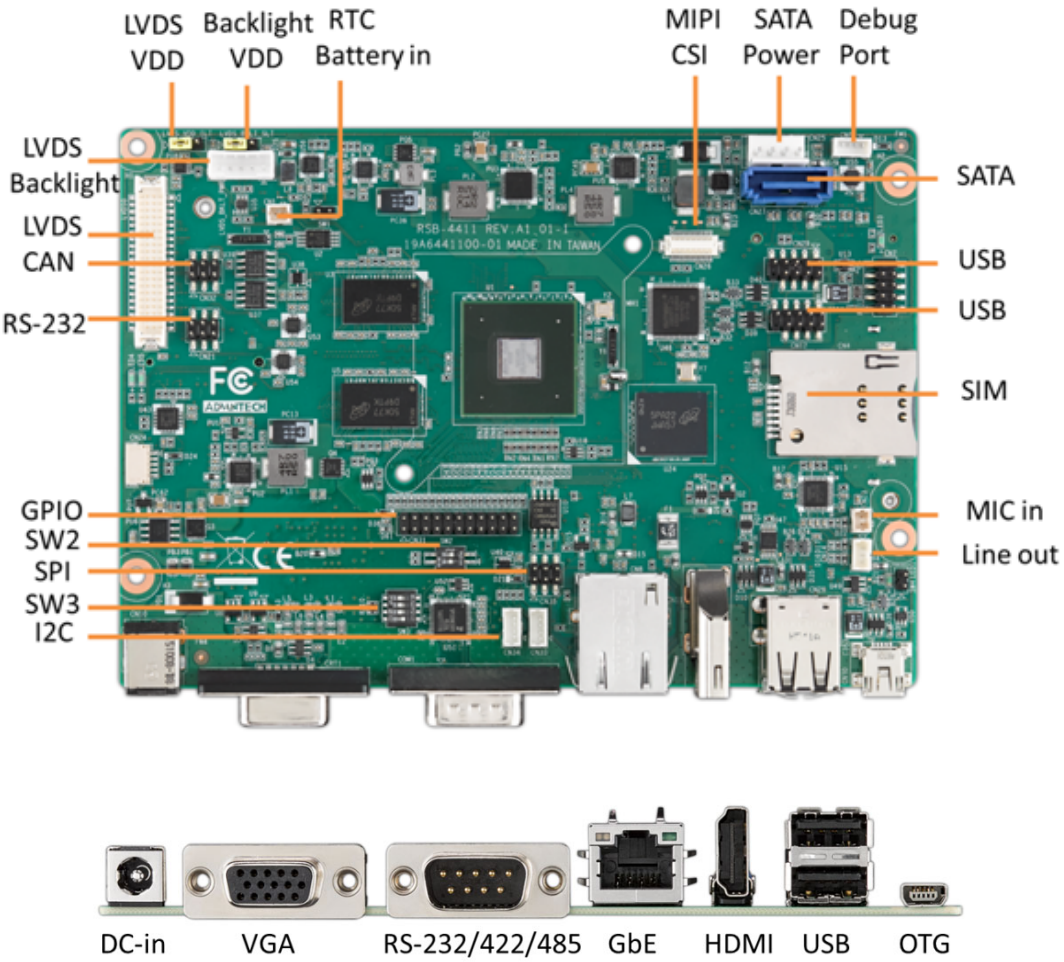


Figure 2.25 M.2 Connector

2.3 Mechanical

2.3.1 Jumper and Connector Location



2.3.2 Board Dimensions

2.3.2.1 Board Drawing

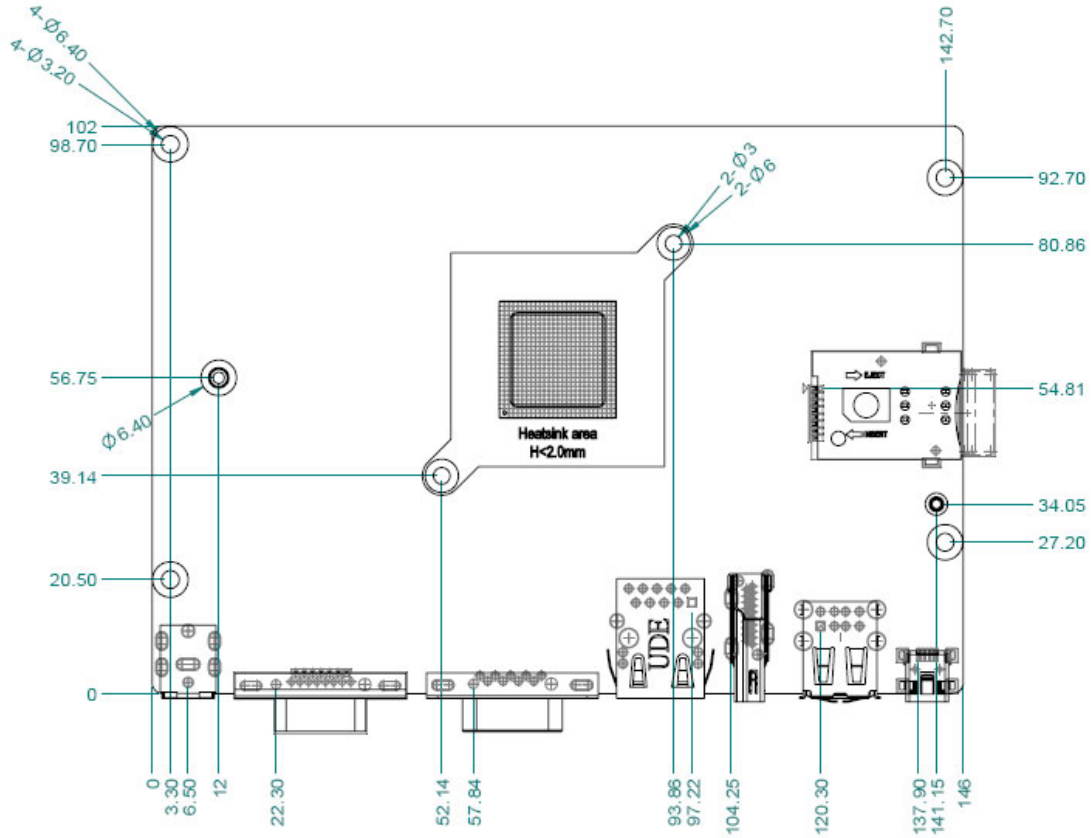


Figure 2.26 Board Dimension Layout (Top Side)

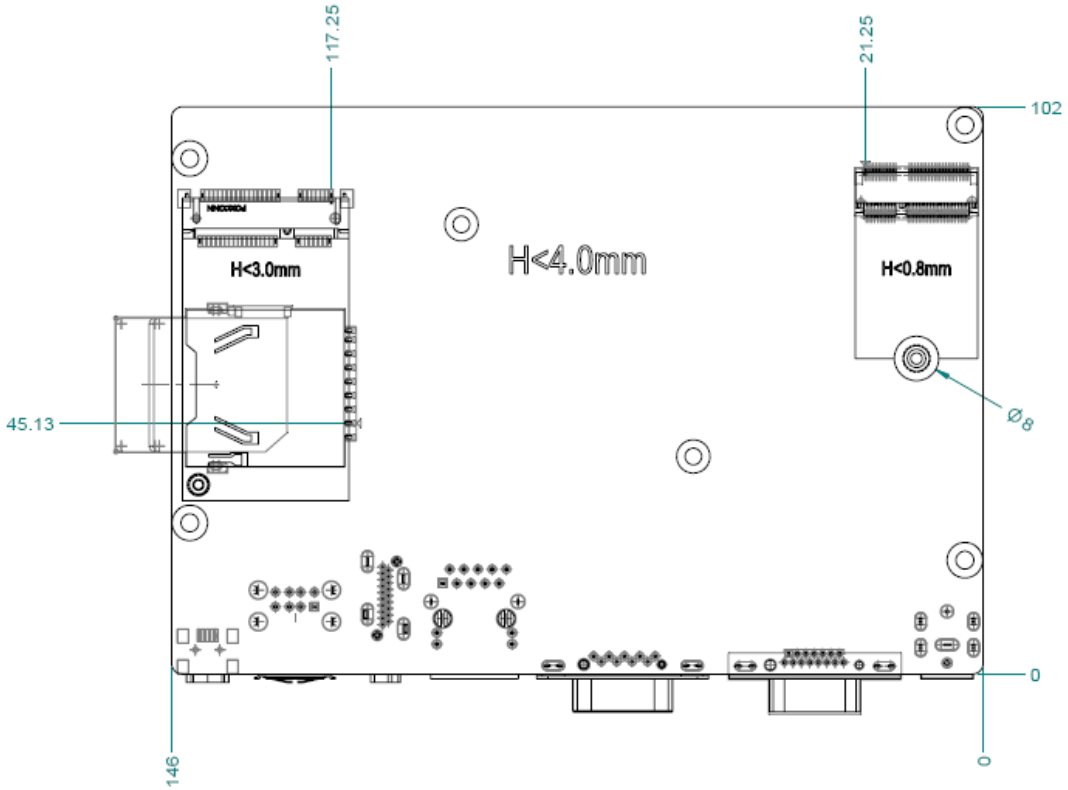


Figure 2.27 Board Dimension Layout (Bottom Side)

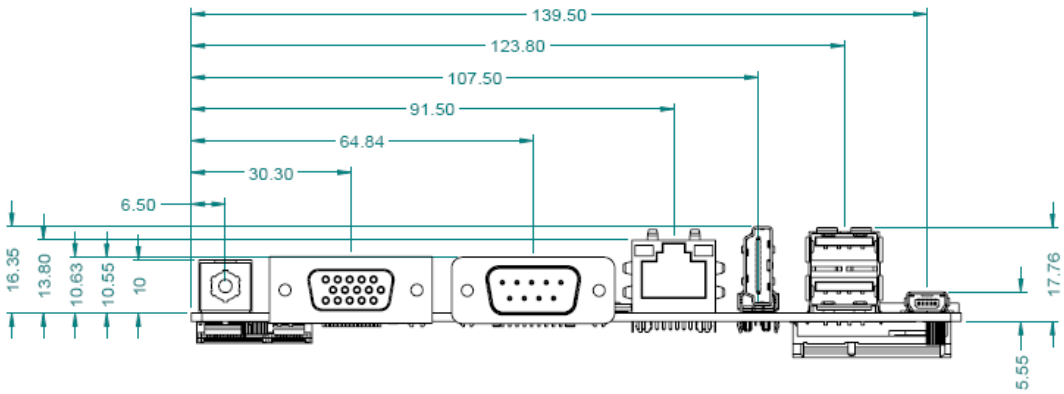


Figure 2.28 Board Dimension Layout (Coastline)

2.4 Quick Start of RSB-4411

2.4.1 Debug Port Connection

1. Connect debug port cable to the RSB-4411 debug port.
2. Connect the RS-232 extension cable to the debug cable.
3. Connector the other sides of the extension cable to the USB-to-RS-232 cable then connect to your PC.

2.4.2 Debug Port Setting

RSB-4411 can communicate with a host server (Windows or Linux) by using serial cables. Common serial communication programs such as Hyper Terminal, Tera Term or PuTTY can be used in this case. The example below describes the serial terminal setup using Hyper Terminal on a Windows host:

1. Connect RSB-4411 with your Windows PC by using a serial cable.
2. Open Hyper Terminal on your Windows PC, and select the settings as shown in Figure 2.25.
3. After the bootloader is programmed on SD card, insert power adapter connector to DC jack on RSB-4411 to power up the board. The bootloader prompt is displayed on the terminal screen.

Figure 2.29 Hyper Terminal Settings for Terminal Setup

Chapter 3

Software Functionality

This chapter details the software programs on the RSB-4411 platform.

3.1 Test Tools

All test tools must be verified on the RSB-4411 Evaluation kit. Please prepare the required test fixtures before verifying each specified I/O. If you have any problems with testing, please contact your Advantech contact window for help.

3.1.1 Display Test

Login to device:

```
# cat /sys/class/graphics/fb0/fsl_disp_dev_property
hdmi
# cat /sys/class/graphics/fb1/fsl_disp_dev_property
overlay
# cat /sys/class/graphics/fb2/fsl_disp_dev_property
ldb
# cat /sys/class/graphics/fb3/fsl_disp_dev_property
overlay
# cat /sys/class/graphics/fb4/fsl_disp_dev_property
ldb
```

2. Modify file:

```
vi /usr/share/imx_6q_display_config

original)
[master]
device = /dev/video17

++++++++++++++++++++++++++++++++

Modify:

[hdmi]
device = /dev/video17
fmt = RGBP
width = 1920
height = 1080
[ldb]
device = /dev/video19
fmt = RGBP
width = 800
height = 480
alpha = 0
[ldb1]
device = /dev/video20
fmt = RGBP
width = 800
```

```
height = 480
alpha = 0

3. Multi display video:
gst-launch-1.0 playbin uri=file:////tools/Advantech.avi video-
sink="overlaysink display-ldb=true display-ldb1=true"
```

3.1.1.1 Multi Display - HDMI + LVDS1 + LVDS2

```
vi /usr/share/imx_6q_display_config

original)
[master]
device = /dev/video17

++++++++++++++++++++++++++++++++

Modify:

[hdmi]
device = /dev/video17
fmt = RGBP
width = 1920
height = 1080
[ldb]
device = /dev/video19
fmt = RGBP
width = 800
height = 480
alpha = 0
[ldb1]
device = /dev/video20
fmt = RGBP
width = 800
height = 480
alpha = 0

3. Multi display video:
gst-launch-1.0 playbin uri=file:////unit_tests/akiyo.mp4 video-
sink="overlaysink display-ldb=true display-ldb1=true"
```

3.1.2 Audio Test

audio codec module connected to I2S_CN1

```
# dmesg | grep sgtl
```

```
root@imx6qrom5420:~# dmesg | grep sgtl
[    2.990620] sgtl5000 4-000a: sgtl5000 revision 0x11
[    3.020788] sgtl5000 4-000a: Failed to get supply 'VDDD': -19
[    3.027787] sgtl5000 4-000a: Using internal LDO instead of VDDD
[    3.050667] imx-sgtl5000 sound.22:  sgtl5000 <-> 202c000.ssi mapping ok
[    3.198085]   #0: imx-audio-sgtl5000
```

```
# cat /proc/asound/cards
```

```
root@imx6qrom5420:~# cat /proc/asound/cards
 0 [imxaudiosgtl500]: imx-audio-sgtl5 - imx-audio-sgtl5000
                      imx-audio-sgtl5000
 1 [imxspdif       ]: imx-spdif - imx-spdif
                      imx-spdif
 2 [imxhdmisoc     ]: imx-hdmi-soc - imx-hdmi-soc
                      imx-hdmi-soc
```

```
# amixer get Mic
```

```
root@imx6qrom5420:~# amixer get Mic
Simple mixer control 'Mic',0
  Capabilities: volume volume-joined
  Playback channels: Mono
  Capture channels: Mono
  Limits: 0 - 3
  Mono: 0 [0%] [0.00dB]
```

```
# amixer set Mic 100
```

```
root@imx6qrom5420:~# amixer set Mic 100
Simple mixer control 'Mic',0
  Capabilities: volume volume-joined
  Playback channels: Mono
  Capture channels: Mono
  Limits: 0 - 3
  Mono: 3 [100%] [40.00dB]
```

```
# arecord -t wav -c 1 -r 44100 -d 5 2.wav
# aplay 2.wav
```

```
root@imx6qrom5420:~# arecord -t wav -c 1 -r 44100 -d 5 2.wav
Recording WAVE '2.wav' : Unsigned 8 bit, Rate 44100 Hz, Mono
root@imx6qrom5420:~# aplay 2.wav
Playing WAVE '2.wav' : Unsigned 8 bit, Rate 44100 Hz, Mono
```

3.1.3 Camera Test

MIPI Camera ov5640 connected to MIPI1

```
# lsmod | grep "capture\|ov5640"
```

```
root@imx6qrom5420:~# lsmod | grep "capture\|ov5640"
mxc_v4l2_capture       22274  1
ipu_bg_overlay_sdc      4001  1 mxc_v4l2_capture
ipu_still               1663  1 mxc_v4l2_capture
ipu_prp_enc             4645  1 mxc_v4l2_capture
ipu_csi_enc             2841  1 mxc_v4l2_capture
ipu_fg_overlay_sdc      4877  1 mxc_v4l2_capture
ov5640_camera_mipi     20774  0
```

Preview

```
# gst-launch-1.0 imxv4l2src device=/dev/video0 ! 'video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1' ! imxv4l2sink
```

Capture

```
# gst-launch-1.0 imxv4l2src num-buffers=1 ! jpegenc ! filesink location=test.jpeg
```

Display

```
# VSALPHA=1 gst-launch-1.0 filesrc location=test.jpeg ! jpegdec ! imagefreeze ! imxv4l2sink
```

Record

```
# gst-launch-1.0 imxv4l2src device=/dev/video0 num-buffers=300 ! 'video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1' ! queue ! vpuenc_h264 ! matroskamux ! filesink location=output.avi
```

Play Video

```
# gst-launch-1.0 playbin uri=file:////home/root/output.avi
```

3.1.4 PCI-e Test

Test it by a WiFi - PCI-e board (Advantech EWM-142)

```
root@imx6qrom5420:~# dmesg | grep -i ath
[    4.298164] usbcore: registered new interface driver ath3k
[    5.031208] ath: EEPROM regdomain: 0x65
[    5.031215] ath: EEPROM indicates we should expect a direct regpair map
[    5.031228] ath: Country alpha2 being used: 00
[    5.031232] ath: Regpair used: 0x65
[    5.095655] ieee80211 phy0: Atheros AR9287 Rev:2 mem=0xc0d80000, irq=155
root@imx6qrom5420:~# lsmod | grep -i ath
ath3k                   5282  0
ath9k                  87250  0
ath9k_common            1627  1 ath9k
ath9k_hw              360976  2 ath9k_common,ath9k
ath                    13650  3 ath9k_common,ath9k,ath9k_hw
mac80211              226378  1 ath9k
cfg80211              176047  3 ath,ath9k,mac80211
```

Check wlan0 existing or not

```
# ifconfig wlan0 up

# ifconfig
# wpa_passphrase "ESSD Testing" adv27927818 > /tmp/wpa.conf
#wpa_supplicant -BDwext -iwlan0 -c/tmp/wpa.conf
# cat /tmp/wpa.conf
# udhcpc -b -i wlan0
```

```
root@imx6qrom5420:~# ifconfig eth0 down
root@imx6qrom5420:~# IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready

root@imx6qrom5420:~# killall wpa_supplicant
root@imx6qrom5420:~# rm /etc/resolv.conf
root@imx6qrom5420:~# wpa_passphrase "$SSID" $PSWD > /tmp/wpa.conf
root@imx6qrom5420:~# wpa_supplicant -BDwext -iwlan0 -c/tmp/wpa.conf
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
root@imx6qrom5420:~# wlan0: authenticate with c8:60:00:ac:91:7c
wlan0: send auth to c8:60:00:ac:91:7c (try 1/3)
wlan0: authenticated
wlan0: associate with c8:60:00:ac:91:7c (try 1/3)
wlan0: RX AssocResp from c8:60:00:ac:91:7c (capab=0x411 status=0 aid=7)
IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: associated

root@imx6qrom5420:~# udhcpc -b -i wlan0
udhcpc (v1.21.1) started
Sending discover...
Sending select for 192.168.1.236...
Lease of 192.168.1.236 obtained, lease time 28800
/etc/udhcpc.d/50default: Adding DNS 192.168.1.1
root@imx6qrom5420:~# rm /tmp/wpa.conf
```

Check wlan0 working or not

```
# ifconfig wlan0
# ping 8.8.8.8
```

```
root@imx6qrom5420:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr A8:54:B2:73:CF:34
          inet addr:192.168.1.236  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:112 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15316 (14.9 KiB)  TX bytes:14702 (14.3 KiB)

root@imx6qrom5420:~# ping -c5 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=49 time=40.270 ms
64 bytes from 8.8.8.8: seq=1 ttl=49 time=25.565 ms
64 bytes from 8.8.8.8: seq=2 ttl=49 time=13.378 ms
64 bytes from 8.8.8.8: seq=3 ttl=49 time=12.352 ms
64 bytes from 8.8.8.8: seq=4 ttl=49 time=23.441 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 12.352/23.001/40.270 ms
```

3.1.5 Battery Test

3.1.5.1 AC power on, no battery connected

```
# cat /sys/class/power_supply/battery/uevent
```

```
# cat /sys/class/power_supply/battery/uevent
POWER_SUPPLY_NAME=battery
POWER_SUPPLY_STATUS=Unknown
POWER_SUPPLY_HEALTH=Dead
POWER_SUPPLY_PRESENT=0
POWER_SUPPLY_TECHNOLOGY=Li-ion
POWER_SUPPLY_CYCLE_COUNT=-1472700156
POWER_SUPPLY_VOLTAGE_NOW=0
POWER_SUPPLY_CURRENT_NOW=0
POWER_SUPPLY_CAPACITY=0
POWER_SUPPLY_TEMP=0
POWER_SUPPLY_TIME_TO_EMPTY_AVG=0
POWER_SUPPLY_TIME_TO_FULL_AVG=0
POWER_SUPPLY_SERIAL_NUMBER=0000
POWER_SUPPLY_VOLTAGE_MAX_DESIGN=0
POWER_SUPPLY_ENERGY_NOW=441297984
POWER_SUPPLY_ENERGY_FULL=441297984
POWER_SUPPLY_ENERGY_FULL_DESIGN=441297984
POWER_SUPPLY_CHARGE_NOW=-1472700156
POWER_SUPPLY_CHARGE_FULL=-1472700156
POWER_SUPPLY_CHARGE_FULL_DESIGN=-1472700156
```

3.1.5.2 AC power on, battery connected

```
# cat /sys/class/power_supply/battery/uevent
```

3.1.5.3 AC power off, battery connected

```
# cat /sys/class/power_supply/battery/uevent
```

3.1.6 RS-232 Test

```
#stty -F /dev/ttymxc1 115200
#stty -F /dev/ttymxc1 -echo
#cat /dev/ttymxc1 &
#echo test > /dev/ttymxc1
test
```

```
root@imx6qrom5420:~# ./test_rs232.sh
ttymxc1 : PASS
ttymxc2 : PASS
ttymxc3 : PASS
ttymxc4 : PASS
```

3.1.7 I2C Test

```
# i2cdetect -l
```

```
root@imx6qrom5420:~# i2cdetect -l
i2c-0   i2c             21a0000.i2c                             I2C adapter
i2c-1   i2c             21a4000.i2c                             I2C adapter
i2c-2   i2c             21a8000.i2c                             I2C adapter
i2c-3   i2c             i2c-1-mux (chan_id 0)                   I2C adapter
i2c-4   i2c             i2c-1-mux (chan_id 1)                   I2C adapter
i2c-5   i2c             i2c-1-mux (chan_id 2)                   I2C adapter
i2c-6   i2c             i2c-1-mux (chan_id 3)                   I2C adapter
```

Test it by an audio codec module (ROM-ED51)

```
# dmesg | grep sgtl
```

```
root@imx6qrom5420:~# dmesg | grep sgtl
[    2.990620] sgtl5000 4-000a: sgtl5000 revision 0x11
[    3.020788] sgtl5000 4-000a: Failed to get supply 'VDDD': -19
[    3.027787] sgtl5000 4-000a: Using internal LDO instead of VDDD
[    3.050667] imx-sgtl5000 sound.22:  sgtl5000 <-> 202c000.ssi mapping ok
[    3.198085]   #0: imx-audio-sgtl5000
```

```
# i2cdetect -y 4
```

```
root@imx6qrom5420:~# i2cdetect -y 4
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- UU -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: UU -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- 66 -- -- -- -- -- -- -- -- --
70: UU -- -- -- -- -- -- --
```

```
# i2cset -f -y 4 0x0a 0 0xff00 w
# i2cget -f -y 4 0x0a 0 w
```

```
# i2cset -f -y 4 0x0a 0 0xff00 w
# i2cget -f -y 4 0x0a 0 w
0x11a0
```

```
no module connected
# dmesg | grep sgtl
```

```
root@imx6qrom5420:~# dmesg | grep sgtl
[    2.989110] sgtl5000: probe of 4-000a failed with error -5
[    3.004054] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[    3.009140] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[    3.014406] platform sound.22: Driver imx-sgtl5000 requests probe deferral
[    3.130699] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[    3.135833] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[    3.141076] platform sound.22: Driver imx-sgtl5000 requests probe deferral
[    3.159441] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[    3.164557] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[    3.169807] platform sound.22: Driver imx-sgtl5000 requests probe deferral
[    4.392564] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[    4.449532] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[    4.485975] platform sound.22: Driver imx-sgtl5000 requests probe deferral
[    5.939398] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[    5.944724] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[    5.944738] platform sound.22: Driver imx-sgtl5000 requests probe deferral
[   10.696587] imx-sgtl5000 sound.22: ASoC: CODEC (null) not registered
[   10.703019] imx-sgtl5000 sound.22: snd_soc_register_card failed (-517)
[   10.709328] platform sound.22: Driver imx-sgtl5000 requests probe deferral
```

```
# i2cdetect -y 4
```

```
root@imx6qrom5420:~# i2cdetect -y 4
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: UU -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- 66 -- -- -- -- -- -- -- -- --
70: UU -- -- -- -- -- -- --
```

3.1.8 USB Test

Insert a USB disk

```
usb 1-1.2: new high-speed USB device number 4 using ci_hdrc
usb-storage 1-1.2:1.0: USB Mass Storage device detected
scsi2 : usb-storage 1-1.2:1.0
scsi 2:0:0:0: Direct-Access     Generic- SD/MMC           1.00 PQ: 0
ANSI: 0 CCS
sd 2:0:0:0: [sda] 3862528 512-byte logical blocks: (1.97 GB/1.84
GiB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] No Caching mode page found
sd 2:0:0:0: [sda] Assuming drive cache: write through
sd 2:0:0:0: [sda] No Caching mode page found
sd 2:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 2:0:0:0: [sda] No Caching mode page found
sd 2:0:0:0: [sda] Assuming drive cache: write through
```

```
sd 2:0:0:0: [sda] Attached SCSI removable disk
```

```
root@imx6qprom7421a1:/# usb 1-1.2: USB disconnect, device number 3
FAT-fs (sda1): FAT read failed (blocknr 42)
usb 1-1.2: new high-speed USB device number 4 using ci_hdrc
usb-storage 1-1.2:1.0: USB Mass Storage device detected
scsi2 : usb-storage 1-1.2:1.0
imx-sgtl5000 sound.21: ASoC: CODEC (null) not registered
imx-sgtl5000 sound.21: snd_soc_register_card failed (-517)
platform sound.21: Driver imx-sgtl5000 requests probe deferral
scsi 2:0:0:0: Direct-Access     SanDisk  Extreme          0001 PQ: 0 ANSI: 6
imx-sgtl5000 sound.21: ASoC: CODEC (null) not registered
sd 2:0:0:0: [sda] 61282631 512-byte logical blocks: (31.3 GB/29.2 GiB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
 sda: sda1
imx-sgtl5000 sound.21: snd_soc_register_card failed (-517)
sd 2:0:0:0: [sda] Attached SCSI removable disk
platform sound.21: Driver imx-sgtl5000 requests probe deferral
FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run f
sck.
```

```
root@imx6qprom7421a1:/#  dd if=/dev/urandom of=data bs=1 count=1024
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.028809 s, 35.5 kB/s
```

```
root@imx6qprom7421a1:/# dd if=/dev/sda of=backup bs=1 count=1024 skip=4096
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.0328767 s, 31.1 kB/s
```

```
root@imx6qprom7421a1:/# dd if=data of=/dev/sda bs=1 seek=4096
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.0137947 s, 74.2 kB/s
```

```
root@imx6qprom7421a1:/# dd if=/dev/sda of=data1 bs=1 count=1024 skip=4096
1024+0 records in
1024+0 records out
1024 bytes (1.0 kB) copied, 0.0326443 s, 31.4 kB/s
```

According to the above content, we can know sda is our usb disk

Generate random file

```
# dd if=/dev/urandom of=data bs=1 count=1024
```

Back up

```
# dd if=/dev/sda of=backup bs=1 count=1024 skip=4096
```

Write to usb disk

```
# dd if=data of=/dev/sda bs=1 seek=4096
```

Read and Verify

```
# dd if=/dev/sda of=data1 bs=1 count=1024 skip=4096
# diff data data1
```

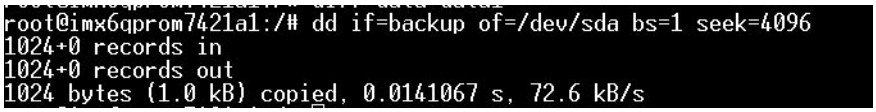
```
root@imx6qprom7421a1:/# diff data data1
root@imx6qprom7421a1:/#
```

If fail, it shows as below:

Binary files data1 and data differ

Restore

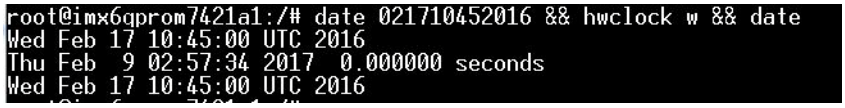
```
# dd if=backup of=/dev/sda bs=1 seek=4096
```



3.1.9 RTC Test

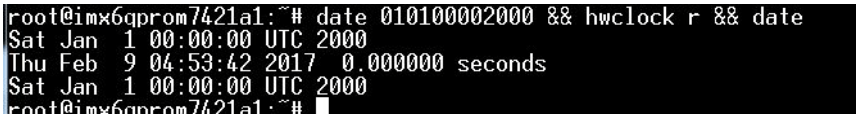
set system time to current, then write to RTC

```
# date 021710452016 && hwclock -w && date
```



set one incorrect time, then read time from RTC to verify

```
# date 010100002000 && hwclock -r && date
```



restore the RTC time to system time

```
#   hwclock -s && date
```

3.1.10 MMC (eMMC/SD) Test

booting from MB EMMC

Message from U-Boot

```
CPU:   Freescale i.MX6Q rev1.5 at 792 MHz
CPU:   Temperature 27 C, calibration data: 0x5774c97d
Reset cause: POR
Board: MX6Q/SDL-SabreSD
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
JEDEC ID: 0x20:0xbb:0x16
In:    serial
Out:   serial
Err:   serial
boardcfg_get_mac
JEDEC ID: 0x20:0xbb:0x16
MAC address is invailed !!
Use default MAC adderss:0x00:0x04:0x9F:0x01:0x30:0xE0
Net:   FEC [PRIME]
booting from iNAND
Normal Boot
Hit any key to stop autoboot:  0
```

```
# cat /proc/cmdline
```

```
root@imx6qrom5420:~# cat /proc/cmdline
console=ttymxc0,115200 root=/dev/mmcblk0p2 rootwait rw advboot_version=2009.08-advantech_rom5420_1G_V2.760
_svn1416 uboot_version=2013.04-advantech_rom5420_V3.500_svn1502
```

SD

Boot from eMMC and insert a bootable SD card. The mmcblk1p1 & mmcblk1p2 should be mounting automatically.

```
root@imx6qprom7421a1:/# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root        3498000 1460612   1851044  45% /
devtmpfs          870088      76    870012   1% /dev
tmpfs                 40       0        40   0% /mnt/.psplash
tmpfs            1034100     352   1033748   1% /run
tmpfs            1034100     144   1033956   1% /var/volatile
/dev/mmcblk0p1      8168    5460      2708  67% /run/media/mmcblk0p1
/dev/sdb1       15615976 3555208  12060768  23% /run/media/sdb1
/dev/sde1       15625632   10944  15614688   1% /run/media/sde1
/dev/sdd1       30626320 5123776  25502544  17% /run/media/sdd1
/dev/sdc1        7622592  104428   7518164   2% /run/media/sdc1
/dev/sda4      306288504      20 290729848   1% /run/media/sda4
/dev/sda7          11342      13     10744   1% /run/media/sda7
/dev/sda6         472834      13    448407   1% /run/media/sda6
/dev/sda5         354224  314828     32084  91% /run/media/sda5
/dev/mmcblk2p2   1771824 1455272    220296  87% /run/media/mmcblk2p2
/dev/mmcblk2p1      8168    5460      2708  67% /run/media/mmcblk2p1
root@imx6qprom7421a1:/#
```

booting from CB SD Card
Message from U-Boot

```
CPU:   Freescale i.MX6Q rev1.5 at 792 MHz
CPU:   Temperature 32 C, calibration data: 0x5774c97d
Reset cause: POR
Board: MX6Q/SDL-SabreSD
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
JEDEC ID: 0x20:0xbb:0x16
In:    serial
Out:   serial
Err:   serial
boardcfg_get_mac
JEDEC ID: 0x20:0xbb:0x16
MAC address is invailed !!
Use default MAC adderss:0x00:0x04:0x9F:0x01:0x30:0xE0
Net:   FEC [PRIME]
booting from SD
Normal Boot
Hit any key to stop autoboot:  0
```

```
# cat /proc/cmdline
```

```
root@imx6qrom5420:~# cat /proc/cmdline
console=ttymxc0,115200 root=/dev/mmcblk1p2 rootwait rw advboot_version=2009.08-advantech_rom5420_1G_V2.760
_svn1416 uboot_version=2013.04-advantech_rom5420_V3.500_svn1502
```

eMMC
Boot from SD card and mmcblk0p1 & mmcblk0p2 should be mounting automatically.
SPI

Generate random file
```
# dd if=/dev/urandom of=data bs=1 count=1024
```
Back up
```
# dd if=/dev/mtdblock1 of=backup bs=1 count=1024 skip=$((0xD3000))
```
Write to mtdblock1
```
# dd if=data of=/dev/mtdblock1 bs=1 seek=$((0xD3000))
```
Read and Verify
```
# dd if=/dev/mtdblock1 of=data1 bs=1 count=1024 skip=$((0xD3000))
# diff data data1
```
If fail, it shows as below:
Binary files data1 and data differ
Restore
```
# dd if=backup of=/dev/mtdblock1 bs=1 seek=$((0xD3000))
```

3.1.11 SATA Test

Insert SATA disk before boot

```
#find /sys/ . -name block | grep ata | xargs ls $1
sda
```

According to the above content, we can know sda is our SATA disk

Generate random file

```
# dd if=/dev/urandom of=data bs=1 count=1024
```

Back up

```
# dd if=/dev/sda of=backup bs=1 count=1024 skip=4096
```

Write to SATA disk

```
# dd if=data of=/dev/sda bs=1 seek=4096
```

Read and Verify

```
# dd if=/dev/sda of=data1 bs=1 count=1024 skip=4096
# diff data data1
```

If fail, it shows as below:

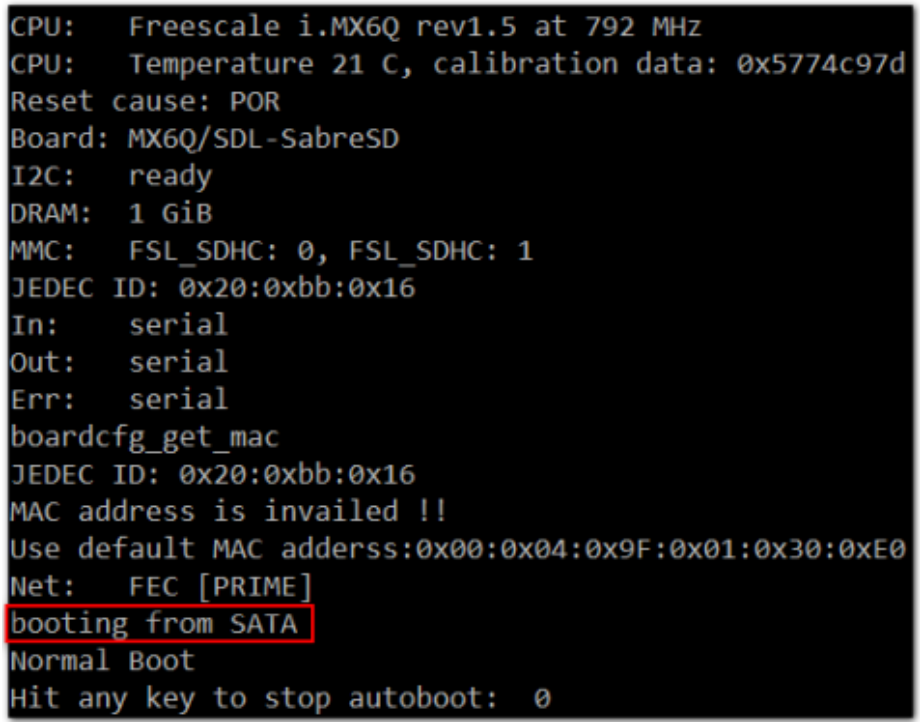
Binary files data1 and data differ

Restore

```
# dd if=backup of=/dev/sda bs=1 seek=4096
```

booting from CB SATA

Message from U-Boot




```
# cat /proc/cmdline
```

3.1.12 Ethernet Test

#ifconfig

```
root@imx6qprom7421a1:/tools# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:19:0F:2C:37:5D
          inet addr:172.22.16.84  Bcast:172.22.19.255  Mask:255.255.252.0
          inet6 addr: fe80::219:fff:fe2c:375d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4062 errors:0 dropped:555 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:516043 (503.9 KiB)  TX bytes:12611 (12.3 KiB)

eth0:0    Link encap:Ethernet  HWaddr 00:19:0F:2C:37:5D
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 B)  TX bytes:700 (700.0 B)
```

#ping 8.8.8.8

```
          collisions:0 txqueuelen:1000
          RX bytes:516043 (503.9 KiB)  TX bytes:12611 (12.3 KiB)

eth0:0    Link encap:Ethernet  HWaddr 00:19:0F:2C:37:5D
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 B)  TX bytes:700 (700.0 B)

root@imx6qprom7421a1:/tools# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=44 time=14.708 ms
64 bytes from 8.8.8.8: seq=1 ttl=44 time=9.733 ms
64 bytes from 8.8.8.8: seq=2 ttl=44 time=9.843 ms
64 bytes from 8.8.8.8: seq=3 ttl=44 time=9.761 ms
64 bytes from 8.8.8.8: seq=4 ttl=44 time=9.781 ms
```

3.1.13 CAN Test

Connect device with CAN0 (CN8A)

```
# ip link set can0 up type can bitrate 125000
```

```
root@imx6qrom5420:~# ip link set can0 up type can bitrate 125000
flexcan 2090000.can can0: writing ctrl=0x0e31a055
```

3.1.14 GPIO Test

3.1.14.1 GPIO Pin define:

GPIO_Q7-1: NANDF_CS0/GPIO6_IO11
GPIO_Q7-2: NANDF_CS1/GPIO6_IO14
GPIO_Q7-3: NANDF_CS2/GPIO6_IO15
GPIO_Q7-4: NANDF_CS3/GPIO6_IO16
GPIO_Q7-5: EIM_OE/GPIO2_IO25
GPIO_Q7-6: EIM_CS0/GPIO2_IO23
GPIO_Q7-7: EIM_CS1/GPIO2_IO24
GPIO_Q7-8: EIM_A25/GPIO5_IO2

3.1.14.2 GPIO Number:

GPIO1 number = 171 [32*(6-1) + 11]
GPIO2 number = 174 [32*(6-1) + 14]
GPIO3 number = 175 [32*(6-1) + 15]
GPIO4 number = 176 [32*(6-1) + 16]
GPIO5 number = 57 [32*(2-1) + 25]
GPIO6 number = 55 [32*(2-1) + 23]
GPIO7 number = 56 [32*(2-1) + 24]
GPIO8 number = 130 [32*(5-1) + 2]

3.1.14.3 GPIO Test method:

1) cd /sys/class/gpio

2) Before setting:

```
root@imx6qprom7421:/sys/class/gpio# ls
export     gpiochip128  gpiochip192  gpiochip32  gpiochip96
gpiochip0  gpiochip160  gpiochip247  gpiochip64  unexport
```

3) export 8 gpio pins

```
export GPIO1, command:
echo 171 > ./export

export GPIO2, command:
echo 174 > ./export

export GPIO3, command:
echo 175 > ./export

export GPIO4, command:
echo 176 > ./export

export GPIO5, command:
echo 57 > ./export

export GPIO6, command:
```

```
echo 55 > ./export

export GPIO7, command:
echo 56 > ./export

export GPIO8, command:
echo 130 > ./export

        --> display gpio1 ~ gpio8, command:
ls
```

Console log:

```
export  gpio3  gpio6  gpiochip0    gpiochip192  gpiochip64
gpio1   gpio4  gpio7  gpiochip128  gpiochip247  gpiochip96
gpio2   gpio5  gpio8  gpiochip160  gpiochip32   unexport
```

4) See gpio1 ~ gpio8 direction:

For example, See gpio1 direction [input(in) or output(out)], command:

```
cat gpio1/direction
```

```
Console log:
in
```

5) See gpio1 ~ gpio8 value:

For example, See gpio1 value [low active(0) or high (1)] , command:

```
cat gpio1/value
```

Console log:

```
1
```

6) Set gpio1 ~ gpio8 direction:

a1) For example, Set gpio1 direction input(in) , command:

```
echo in > gpio1/direction
```

a2) get direction, command:

```
cat gpio1/direction
```

Console log:

```
in
```

b1) For example, Set gpio1 direction output(out), command:

```
echo out > gpio1/direction
```

b2) get direction, command:

```
cat gpio1/direction
```

Console log:

out

7) Set gpio1 ~ gpio8 value, when gpio direction is output:

a1) For example, See gpio1 value 0 (low active) , command:

```
echo 0 > gpio1/value
```

a2) get value, command:

```
cat gpio1/value
```

Console log:

```
0
```

b1) For example, See gpio1 value 1(high active), command:

```
echo 1 > gpio1/value
```

b2) get value, command:

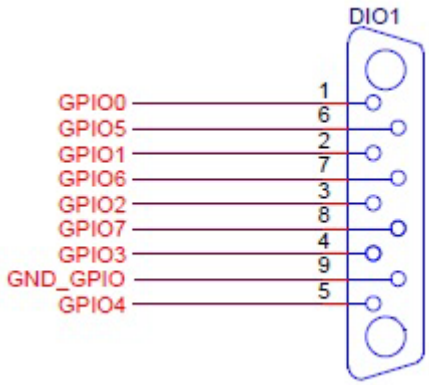
```
cat gpio1/value
```

Console log:

```
1
```

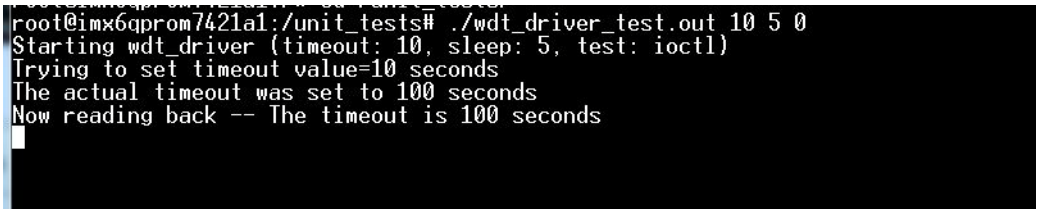
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

Connect jumpers to DIO1 connector by follows jumper setting



3.1.15 Watchdog Test

```
cd unit_tests
./wdt_driver_test.out 10 5 0
```



press Ctrl+C

```
root@imx6qprom7421a1:/unit_tests# ./wdt_driver_test.out 10 5 0
Starting wdt_driver (timeout: 10, sleep: 5, test: ioctl)
Trying to set timeout value=10 seconds
The actual timeout was set to 100 seconds
Now reading back -- The timeout is 100 seconds
^Cadv-wdt-i2c 0-0029: Unexpected close: Expect reboot!
```

System will reboot after 10 seconds.

3.2 Package Content

We would offer you two different kinds of Linux package for RSB-4411. One is pre-built system image for system recovery another is source code package (BSP).

3.2.1 Source Code Package

RSB-4411 source code package (BSP) contains cross compiler, Linux source code, Uboot source code, root file system and some scripts used in OS development. Some of above components are developed by Advantech and the others are developed by open source community. RSB-4411 source code package is composed of six main folders: “cross_compiler”, “document”, “image”, “package”, “scripts”, and “source”.

Note! *RSB-4411 source code package (BSP) is Advantech’s Intellectual Property. If you need to access this package, please contact your Advantech support window.*

The description of 7421LBVxxxx package contents:

- **“cross_compiler”** → This folder contains source code for cross compiler.
- **“document”** → This folder contains user guide.
- **“image”** → This folder contains the uImage, and the script for making Linux system media automatically.
- **“image/rootfs”** → This folder contains Linux root file system
- **“package”** → This folder contains source code provided by NXP without any modification
- **“scripts”** → This folder contains scripts for configure system and compile images automatically.
- **“source”** → This folder contains source code owned by Advantech

3.2.1.1 cross_compiler

You can use the cross compiler toolchain to compile the uImage and related applications. (gcc version is 4.6.2 20110630)

Toolchain directory structure is as follow:

|-- bin // toolchain with prefix, such as arm-none-linux-gnueabi-gcc etc.

|-- lib // library files used for toolchain itself, not for application

|-- arm-fsl-linux-gnueabi

|-- bin // toolchain without prefix, such as gcc.

|-- debug-root // all debug tools

|-- multi-libs // all libraries and headers.

|-- armv5 // library for armv5 (i.mx 2xx). only support soft float point

|-- armv6 // library for armv6 (i.mx 3xx), soft fpu version

|-- armv7-a // library for armv7-a (i.mx5xx and i.mx6xx), hardware fpu version

|-- lib //default library. It can be used for armv4t and above.

|-- usr

|-- include //header files for the application development

|-- lib //three-part library and static built library NXP

3.2.1.2 document

User guide of how to setup up the environment of development

3.2.1.3 image

This folder includes uImage & u-boot.

3.2.1.4 image/rootfs

Linux adopts Hierarchical File System (HFS), image/rootfs is the Linux file system in highest level of the tree structure.

The main folders in “rootfs” are listed as follows:

- bin → Common programs, shared by the system, the system administrator and the users.
- dev → Contains references to all the CPU peripheral hardware, which are represented as files with special properties.
- etc → Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows
- home → Home directories of the common users.
- lib → Library files, includes files for all kinds of programs needed by the system and the users.
- mnt → Standard mount point for external file systems.
- opt → Typically contains extra and third party software.
- proc → A virtual file system containing information about system resources. More information about the meaning of the files in proc is obtained by entering the command man proc in a terminal window. The file proc.txt discusses the virtual file system in detail.
- root → The administrative user's home directory. Mind the difference between / , the root directory and /root, the home directory of the root user.
- sbin → Programs for use by the system and the system administrator.
- sys → Linux sys file system
- tmp → Temporary space for use by the system, cleaned upon reboot, so doesn’t use this for saving any work!
- unit_tests → unit test tools are provided by NXP i.MX6 product
- usr → Programs, libraries, documentation etc. for all user-related programs.
- var → Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet.
- tools → just for sample test.

3.2.1.5 scripts

Some scripts provided by Advantech will help you configure your system or build images more quickly:

- setenv.sh → A script to setup the developing environment quickly.
- cfg_uboot.sh → A script to configure the u-boot building setup quickly.
- mk_uboot.sh → A script to build the u-boot and copy the “u-boot” to “image” folder after building.
- cfg_kernel.sh → A script to configure the kernel building setup quickly.
- mk_kernel.sh → A script to build the “uImage” and copy the “uImage” to “image” folder after building.
- mksd-linux.sh → A script to setup up a bootable SD card if users build their images

3.2.1.6 source

This folder contains sub-directories “linux-10.17” and “u-boot-2009.08”. They are the source codes of the Linux kernel and U-boot.

Linux is a clone of the operating system UNIX. It has all the features you would expect in a modern fully-fledged UNIX, including true multitasking, virtual memory, shared libraries, demand loading, shared copy-on-write executables, proper memory management, and multitask networking including IPv4 and IPv6.

Linux is easily ported to most general-purpose 32- or 64-bit architectures as long as they have a paged memory management unit (PMMU) and a port of the GNU C compiler (gcc) (part of The GNU Compiler Collection, GCC). Linux has also been ported to a number of architectures without a PMMU, although functionality is then obviously somewhat limited. Linux has also been ported to itself.

The main sub-directories under “linux-3.0.35” are listed as follows:

- arch → The items related to hardware platform, most of them are for CPU.
- block → The setting information for block.
- crypto → The encryption technology that kernel supports.
- Documentation → The documentation for kernel.
- drivers → The drivers for hardware.
- firmware → Some of firmware data for old hardware.
- fs → The file system the kernel supports.
- include → The header definition for the other programs used.
- init → The initial functions for kernel.
- ipc → Define the communication for each program of Linux O.S.
- kernel → Define the Kernel process, status, schedule, signal.
- lib → Some of libraries.
- mm → The data related the memory.
- net → The data related the network.
- security → The security setting.
- sound → The module related audio.
- virt → The data related the virtual machine.

There are plenty of documentations or materials available on Internet and also could be obtained from books and magazines, you can easily find the answers for both Linux-specific and general UNIX questions.

There are also various README files in ./source/linux-3.0.35/Documentation, you can find the kernel-specified installations and notes for drivers. You can refer to ./source/linux-3.0.35/Documentation/00-INDEX for a list of the purpose of each README/note.

3.3 Setting up a Build Environment

All operations in this guide are based on Ubuntu 12.04 LTS 64bit only. First please install Ubuntu 12.04 LTS 64bit* with minimum 2GB memory. * ubuntu-12.04.1-desktop-amd64.iso

Please login and perform the following commands to install required packages:

```
$ sudo apt-get install ssh
$ sudo apt-get install ia32-libs libx11-dev:i386 libreadline6-dev:i386 \
libgl1-mesa-glx:i386 zlib1g-dev:i386 uuid-dev:i386 liblzo2-dev:i386 \
libncurses5-dev:i386
$ sudo apt-get install \
bison build-essential ccache dpkg flex gcc g++ gettext intltool \
libarchive-zip-perl libfreetype6-dev libdbus-glib-1-dev liborbit2-dev \
libxml2-dev libx11-dev libgtk2.0-dev liblzo2-2 libtool m4 \
patch rpm tcl uboot-mkimage uuid zlib1g zlib1g-dev \
git gnupg flex bison gperf build-essential  zip \
curl libc6-dev libncurses5-dev x11proto-core-dev libx11-dev:i386 \
libreadline6-dev:i386 libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib \
mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386 \
gcc-4.6 g++-4.6 cpp-4.6 gcc-4.6-multilib uuid-dev liblzo2-dev \
uboot-mkimage libarchive-zip-perl \
wget git-core unzip texinfo gawk diffstat build-essential chrpath \
sed cvs subversion coreutils texi2html \
docbook-utils python-pysqlite2 help2man make gcc g++ \
desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev mercurial \
autoconf automake groff curl lzop asciidoc xterm
$ sudo apt-get install libncurses5-dev:i386 liblzo2-dev:i386 uuid-dev:i386
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
$ tar zcvf ~/usr_lib_i386-linux-gnu_for_Building_Android_KK.tar.gz \
/usr/lib/i386-linux-gnu/{libuuid.a,libuuid.so,liblzo2.so,liblzo2.a}
$ sudo apt-get install uuid-dev liblzo2-dev
$ sudo tar zxvf ~/usr_lib_i386-linux-gnu_for_Building_Android_KK.tar.gz -C /
```

Please follow below instruction install JDK:

Download "jdk-6u45-linux-x64.bin" manually from Oracle Official Website and put it to directory ~/FILES/ and perform following commands:

```
$ cd /usr/lib
$ sudo ~/FILES/jdk-6u45-linux-x64.bin
```

```
$ sudo mkdir jvm
$ cd jvm
$ sudo mv ../jdk1.6.0_45 .
$ cd jdk1.6.0_45/
$ sudo update-alternatives --install /usr/bin/java    java    /usr/
lib/jvm/jdk1.6.0_45/jre/bin/java  2
$ sudo update-alternatives --install /usr/bin/javac   javac   /usr/
lib/jvm/jdk1.6.0_45/bin/javac   2
$ sudo update-alternatives --install /usr/bin/jar     jar     /usr/
lib/jvm/jdk1.6.0_45/bin/jar     2
$ sudo update-alternatives --install /usr/bin/javap   javap   /usr/
lib/jvm/jdk1.6.0_45/bin/javap   2
$ sudo update-alternatives --install /usr/bin/javadoc javadoc /usr/
lib/jvm/jdk1.6.0_45/bin/javadoc 2
$ sudo update-alternatives --config javap
$ sudo update-alternatives --config javadoc
$ sudo update-alternatives --config java
$ sudo update-alternatives --config javac
$ sudo update-alternatives --config jar
$ cd ~/
$ sudo sh -c "echo "JAVA_HOME=/usr/lib/jvm/jdk1.6.0_45" >> /etc/
environment"
```

If you have some troubles on setting up build environment, we recommend that you use Docker, and it is more safely & quickly. Learn more about Docker at: http://ess-wiki.advantech.com.tw/view/IoTGateway/Docker

3.3.1 Conventions

Below is the Naming Conventions of the SW package we provide:

```
${PREBUILT_IMAGE} : compressed prebuilt image (*.img.gz)
${BSP_TARBALL} : BSP tarball (*.tgz)
${BSP_HOME} : home directory of the BSP
${BDIR} : build directory (e.g. build_x11)
${MX6PROC} : i.MX6 Processor: mx6q for iMX6 Quad Core / Dual Core;
mx6dl for iMX6 Dual Lite / Solo
${IMX6PROC} : i.MX6 Processor: imx6q / imx6dl
${BOARD} : available target boards: rom5420 / ubc220 / rom7421/
rom3420 / rom7420 / rsb4410 / rsb4411 / rsb6410 / ubcds31 / wise3310
${BOARD_REV} : board revision a1 / a2 / b1
${MC} : machine code combined with ${IMX6PROC}${BOARD}${BOARD_REV}
for example,
imx6dlrom5420b1 for ROM-5420-Solo B1
imx6dlubc220a1 for UBC-220-Dual-Light A1
imx6qprom7421a1 for ROM-7421-Dual/Quad-Plus A1
imx6qrom3420a1 for ROM-3420-Dual/Quad A1
imx6qrom5420a1 for ROM-5420-Dual/Quad A1
```

```
imx6qrom5420b1 for ROM-5420-Dual/Quad B1
imx6qrom7420a1 for ROM-7420-Dual/Quad A1
imx6qrsb4410a2 for RSB-4410-Dual A2
imx6qrsb4410a1 for RSB-4410-Dual A1
imx6qrsb4411a1 for RSB-4411-Dual/Quad A1
imx6qrsb6410a1 for RSB-6410-Dual A1
imx6qubcds31a1 for UBC-DS31-Dual A1
imx6qwise3310a1 for WISE-3310-Dual A1
${MEM_SIZE} : memory size 1G / 2G / 512M
${SD_DEVICE} : device name of SD card in Ubuntu (e.g. /dev/sdf)
${SDCARD_IMAGE} : sdcard image built by bitbake (*.sdcard)
${UBOOT} :u-boot version(e.g. 2015.04)
${KERNEL} : linux kernel version(e.g. 3.14.52)
${TOOLCHAIN} : toolchain installed directory(e.g. /opt/fsl-imx-x11/
3.14.52-1.1.0)
debug console / serial console:  serial terminal program (e.g. mini-
com, putty, teraterm ...) that serial port is configured to 115200
8N1
terminal console: terminal program (e.g. gnome-terminal, xfce4-ter-
minal ...)
```

3.3.2 Introducing BSP

The BSP is based on Yocto Project with Freescale enhanced features for i.MX6, plus specific target board features from Advantech Inc..

3.3.2.1 Naming Rule

The tarball/prebuilt image name is consist of the model name followed by "LB" or "LI" plus version number and released date.

For example, 4410A1LBV6000_2016-03-07.tgz which "4410A1" stands for RSB-4410 A1, "LB" is acronym of Linux BSP, "V6000" stands for Version 6.000.

For example, 4410A1LIV6000_DualQuad_2016-03-07.img.gz which "LI" is acronym for prebuilt Linux Image, DualQuad means this image is fit for Dual Core/Quad Core.

3.3.2.2 BSP tarball

Unpack BSP tarball to home directory by performing the following command:

```
$ tar xvf ${BSP_TARBALL} -C ~/
```

(Every BSP with different version contains an unique folder, e.g. after unpacking 4410A1LBV6000_2016-03-07.tgz to home directory, the directory , ~/ imx6LBV6000_2016-03-07 is the BSP's home folder.)

The description of some important folders list below:

```
sources/
meta-advantech/ : meta layer by Advantech
meta-fsl-*/ : meta layer by Freescale
fsl-setup-release.sh : to create one new build environment
setup-environment : to continue an exist build environment
```

3.3.2.3 Pre-built image

Perform the following command to build one boot-up SD card

```
$ gunzip -c ${PREBUILT_IMAGE} | dd of=${SD_DEVICE} bs=1M
```

3.3.3 Build Instructions

To create one new build environment ,perform following commands in terminal console:

```
$ cd ${BSP_HOME}
$ MACHINE=${MC} source fsl-setup-release.sh -b ${BDIR} -e x11
```

You need to read and accept the EULA. Press "Y"

To continue an exist build environment, perform following commands in terminal console:

```
$ cd ${BSP_HOME}
$ source setup-environment ${BDIR}
```

3.3.3.1 To build sdcard image

To create/continue a build environment, perform following command in terminal console:

```
$ bitbake fsl-image-qt5
```

The file, fsl-image-qt5-${MC}.sdcard, will be located in directory, ./tmp/deploy/images/${MC}, while building process finished successfully.

3.3.3.2 To build toolchain installer

To create/continue a build environment, perform following command in terminal console:

```
$ bitbake meta-toolchain-qt5
```

The below installer will be located in the directory "./tmp/deploy/sdk".

```
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa9hf-vfp-neon-toolchain-${KERNEL}-1.1.0.sh
```

3.3.3.3 To build u-boot

To create/continue a build environment, perform following commands in terminal console:

```
$ bitbake u-boot-imx
```

The two files, u-boot_crc.bin & u-boot_crc.bin.crc, will be located in directory, ./tmp/deploy/images/${MC}.

3.3.3.4 To build Linux kernel

To create/continue a build environment, perform following commands in terminal console and do menuconfig:

```
$ bitbake linux-imx -c menuconfig
```

to start build

```
$ bitbake linux-imx -c compile -f
$ bitbake linux-imx -c compile_kernelmodules -f
$ bitbake linux-imx -c deploy
```

The two files, zImage & zImage-${IMX6PROC}-${BOARD}-${BOARD_REV}.dtb, will be located in directory, ./tmp/deploy/images/${MC}.

3.3.4 Creating boot-up on-board flash from pre-built image

3.3.4.1 To create one boot-up SD card, perform following command in terminal console:

```
$ gunzip -c ${PREBUILT_IMAGE | dd of=${SD_DEVICE} bs=1M
```

3.3.4.2 To transfer whole system to on-board flash

Boot up from SD card, perform following commands in debug console:

```
# cd /mk_inand
# ./mksd-linux.sh /dev/mmcblk0
```

press y followed by Enter, while "[Done]" shows up means the transferring is finished.

3.3.5 Creating boot-up on-board flash from built sdcard image

3.3.5.1 To create one boot-up SD card, perform following commands in terminal console:

```
$ pushd ${BSP_HOME}/${BDIR}/tmp/deploy/images/${MC}
$ dd if=${SDCARD_IMAGE} of=${SD_DEVICE} bs=1M
$ popd
```

3.3.5.2 To transfer whole system to on-board flash

First you need to boot up from SD card, then insert USB stick that contains ${SDCARD_IMAGE}, USB stick will be auto mounted to /run/media/sda1.

Perform following commands in debug console:

```
# umount /dev/mmcblk0p?
# cd /run/media/sda1
# dd if=${SDCARD_IMAGE} of=/dev/mmcblk0 bs=4M conv=fsync
# P2START=$(fdisk -lu | grep mmcblk0p2 | awk '{print $2}')
# echo -e "d\n2\nn\np\n2\n${P2START}\n\nw\n" | fdisk -u /dev/mmcblk0
# umount /dev/mmcblk0p2
# e2fsck -f -y /dev/mmcblk0p2
# resize2fs /dev/mmcblk0p2
# poweroff
```

3.4 Customization

3.4.1 Package addition

3.4.1.1 To add tcf-agent & openssh-sftp-server

Navigate to the directory where fsl-image-adv.inc located:

```
$ cd ${BSP_HOME}/sources/meta-advantech/recipes-fsl/images
```

Add following line to fsl-image-adv.inc:

```
IMAGE_INSTALL += " tcf-agent openssh-sftp-server "
```

Continue an exist build environment and build sdcard image

3.4.1.2 To add chromium browser,

Navigate to the directory where local.conf located

```
$ cd ${BSP_HOME}/${BDIR}/conf
```

Add following two lines to local.conf

```
CORE_IMAGE_EXTRA_INSTALL += "chromium"
LICENSE_FLAGS_WHITELIST="commercial"
```

Continue an exist build environment and build sdcard image

3.4.2 Setting up SDK

Please follow the section 3.3.3.2 to build toolchain installer, then perform following commands in terminal console:

```
$ cd ${BSP_HOME}/${BDIR}/tmp/deploy/sdk
$ sudo ./fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa9hf-vfp-neon-toolchain-${KERNEL}-1.1.0.sh
```

Enter new installed directory or just press Enter to use default directory.

While Proceed[y/n]? shows up, please enter the correct one "y"

Waiting for the SDK installed

3.4.3 Setting up cross compiling environment

SDK has been set up (ref. Setting up SDK, section 3.4.2), perform following command in terminal console:

```
$ source ${TOOLCHAIN}/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
```

3.4.4 Building & updating u-boot manually

To build u-boot, the cross compiling environment must have been set up. (ref. 3.4.3 Setting up cross compiling environment)

Make one copy from Yocto working directory:

```
$ mkdir -p ~/code
$ pushd ${BSP_HOME}/${BDIR}/tmp/work/${MC}-poky-linux-gnueabi/
$ rm -rf ~/code/u-boot-imx
$ cp -a ./u-boot-imx/${UBOOT}-r0/git ~/code/u-boot-imx
$ popd
```

3.4.5 Configure u-boot

```
$ cd ~/code/u-boot-imx
$ make distclean
$ make ${MX6PROC}${BOARD}${BOARD_REV}_${MEM_SIZE}_config
```

3.4.6 Start building u-boot

```
$ make -j4 LDFLAGS=
```

The two files, u-boot-crc.bin & u-boot-crc.bin.crc, are located in directory "~/code/u-boot-imx".

3.4.7 To update u-boot to target device

Perform the following command to transfer to exist boot-up SD card

```
$ dd if=u-boot_crc.bin.crc of=${SD_DEVICE} bs=512 seek=2 conv=fsync
$ dd if=u-boot-crc.bin of=${SD_DEVICE} bs=512 seek=3 conv=fsync
Make sure new u-boot does work then perform the following commands to transfer to on-board flash
# dd if=u-boot_crc.bin.crc of=/dev/mmcblk0 bs=512 seek=2 conv=fsync
# dd if=u-boot-crc.bin of=/dev/mmcblk0 bs=512 seek=3 conv=fsync
```

3.4.8 Building & updating kernel/modules/dtb manually

To build kernel/modules/dtb, the cross compiling environment must have been set up. (ref. 3.4.3 Setting up cross compiling environment)

Make one copy from Yocto working directory:

```
$ mkdir -p ~/code
$ pushd ${BSP_HOME}/${BDIR}/tmp/work-shared/${MC}/
$ rm -rf ~/code/linux-imx
$ cp -a ./kernel-source ~/code/linux-imx
$ popd
```

3.4.9 Configure linux kernel

```
$ cd ~/code/linux-imx
$ make distclean
$ make imx_v7_adv_defconfig
$ make menuconfig PKG_CONFIG_SYSROOT_DIR= PKG_CONFIG_PATH=
```

3.4.10 Start building linux kernel

```
$ make -j4 zImage LOADADDR=0x10008000 LDFLAGS=
```

The kernel image file, zImage, is located in the directory "./arch/arm/boot/".

3.4.11 Start building kernel modules

```
$ make -j4 modules LDFLAGS=
```

Copy all modules to a temporary rootfs directory, "~/temp/rootfs"

```
$ make modules_install INSTALL_MOD_PATH=~/temp/rootfs
```

3.4.12 Start building device tree blob

```
$ make -j4 ${IMX6PROC}-${BOARD}-${BOARD_REV}.dtb
```

The device tree blob, ${IMX6PROC}-${BOARD}-${BOARD_REV}.dtb, is located in the directory "./arch/arm/boot/dts/".

3.4.13 To update kernel/modules/dtb to target device

Copy zImage & ${IMX6PROC}-${BOARD}-${BOARD_REV}.dtb to the 1st partition of SD card

Copy modues to the 2nd partition of SD card.

Make sure all new linux kernel, device tree and kernel modules work well, then copy all of the

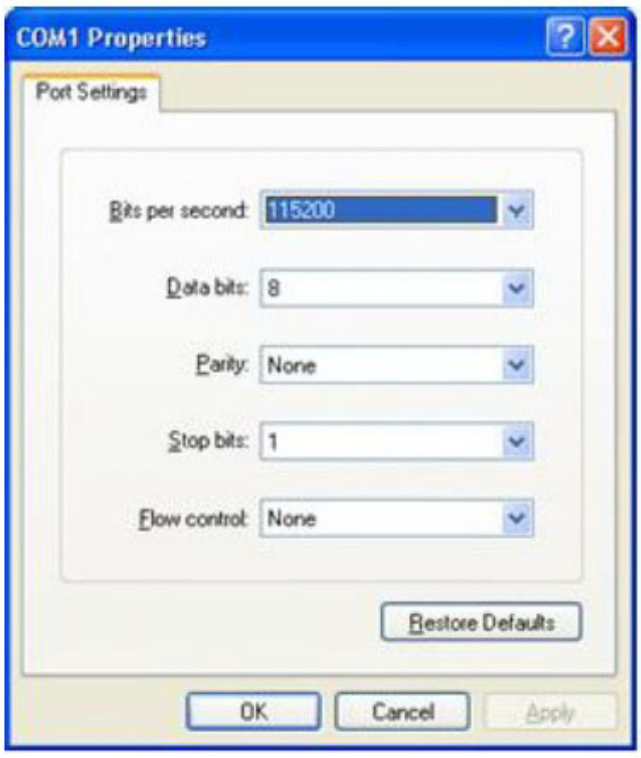
3.5 System Recovery

Please refer to section 3.3.4 Creating boot-up on-board flash from prebuilt image / sdcard image to create a boot-up SD card and transfer whole system to on-board flash.

3.6 Debug Message

RSB-4411 can connect to a host PC (Linux or Windows) by using a console cable and debug port adapter. In order to communicate with a host PC, a serial communication program such as HyperTerminal, Tera Term or PuTTY is required. Below are the detailed instructions for how to set up a serial console, a “HyperTerminal” on a Windows host:

1. Connect RSB-4411 to your Windows PC by using the serial cable, debug port adapter and console cable.
2. Open HyperTerminal on your Windows PC, and select the settings as shown in Figure 3.6.



3. Press “POWER” key to power up the board. The bootloader prompt is displayed on the terminal screen.

3.7 Linux Software AP and Testing on RSB-4411

This section will guide you how to develop your own application under Linux environment. First of all, an example “Hello World” will be shown. And then you will see some pre-installed test programs on RSB-4411 will be introduced in this section.

3.7.1 “Hello World!” Application and Execution

This section will guide you how to write a sample application “Hello World”. You can refer to following steps:

1. Open "Terminal" on Ubuntu 12.04 LTS.
2. **$sudo su** (Change to “root” authority)
3. Type user password.
4. 4)Change directory to BSP's scripts folder
5. **#. setenv.sh** (To configure the developing environment automatically)
6. **#cd ../source**
7. **#mkdir helloworld** (Create your own work directory on the Desktop)
8. **#cd helloworld** (Enter the work directory)
9. **#gedit helloworld.c** (Create a new C source file)

Edit the helloworld.c with the following source code:

```
#include <stdio.h>
void main()
{
   printf("Hello World!\n");
}
```

10. Save the file and exit.
11. **#$CC -o helloworld helloworld.c** (To compile helloworld.c)
12. Then you can see “helloworld” in current directory.
13. Insert the Linux system SD card to your developing computer.
14. **#cp helloworld /media/rootfs/tool** (/media/rootfs is the mounted point of your Linux system SD card)
15. Remove this SD card and insert it to RSB-4411, then open serial console.
16. On RSB-4411 platform, type **#root** (Login)
17. On RSB-4411 platform, type **#cd /tool**
18. On RSB-4411 platform, type **#./helloworld**
19. Now you should be able to see “Hello World!” shown on RSB-4411.

3.7.2 Watchdog Timer Sample Code

WatchDog Timer (WDT) sample code is as below:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <linux/watchdog.h>
#include <sys/ioctl.h>
#include <unistd.h>

void help_info(void);
int main(int argc, const char *argv[])
{
     int fd, timeout, sleep_sec, test;
     int count=1;
     if (argc < 2) {
          help_info();
          return 1;
     }
t    imeout = atoi(argv[1]);
     sleep_sec = atoi(argv[2]);
     if (sleep_sec <= 0) {
          sleep_sec = 1;
          printf("correct 0 or negative sleep time to %d seconds\n",
                 sleep_sec);
     }
     test = atoi(argv[3]);
     printf("Starting wdt_driver (timeout: %d, sleep: %d, test: %s)\n",
          timeout, sleep_sec, (test == 0) ? "ioctl" : "write");
     fd = open("/dev/watchdog", O_WRONLY);
     if (fd == -1) {
          perror("watchdog");
          exit(1);
     }
     printf("Trying to set timeout value=%d seconds\n", timeout);
     ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
     printf("The actual timeout was set to %d seconds\n", timeout);
     ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
     printf("Now reading back -- The timeout is %d seconds\n", timeout);
     while (1) {
          printf("WDT Time out counter:%d\n",count);
          if ((test !=0) && (test ==count)) {
               printf("Ping Watchdog (reset wdt)\n");
               ioctl(fd, WDIOC_KEEPALIVE, 0);
               test=0;
```

```
                count=0;
            }
            sleep(sleep_sec);
            count+=sleep_sec;
        }
        return 0;
}

void help_info(void)
{
        printf("Usage: wdt_driver_test <timeout> <sleep> <trigger>\n");
       printf("    timeout: value in seconds to cause wdt timeout/reset\n");
        printf("    sleep: value in seconds to display wdt timeout\n");
        printf("    trigger: value in seconds to ping the wdt\n");
}
```

If you would like to change the WDT time, please modify:

`ioctl(fd, WDIOC_SETTIMEOUT, &timeout).`

3.7.3 GPIO Setting

Please see GPIO initial code listed below. Below code is to assign the starting value to GPIO variable.

```
/* Enable GPIO */
gpio_request(SABRESD_GPIO0, "gpio-0");
gpio_request(SABRESD_GPIO1, "gpio-1");
gpio_request(SABRESD_GPIO2, "gpio-2");
gpio_request(SABRESD_GPIO3, "gpio-3");
gpio_request(SABRESD_GPIO4, "gpio-4");
gpio_request(SABRESD_GPIO5, "gpio-5");
gpio_request(SABRESD_GPIO6, "gpio-6");
gpio_request(SABRESD_GPIO7, "gpio-7");
```

3.7.4 RS232 Initial Code

The RS232 initial code as below. It shows you how to initial COM ports.

```
int open_port(void)
{
        int fd;
        fd=open("/dev/ttymxc1",O_RDWR|O_NOCTTY|O_NDELAY);
        if(fd == -1){
        perror("open error");
        }
        return(fd);
}
```

3.7.5 Display Output Setting

3.7.5.1 LVDS Settings

Please set environment in u-boot as below:

```
setenv bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/
mmcblk1p1 rootwait rw video=mxcfb0:dev=ldb,LDB-XGA,if=RGB24'
```

LDB-XGA is an example for the resolution of your LVDS panel. You can input the actual resolution of your LVDS panel here, such as 800x480, 1024x768, etc. The system will accomplish the corresponding parameters automatically.

If the panel has a problem to be activated, you may need to check the panel datasheet to configure the panel related parameters. The LVDS video mode database is stored in linux-3.0.35/drivers/video/mxc/ldb.c. You can add a new one for your LVDS panel.

```
static struct fb_videomode ldb_modedb[] = {
     {
     "LDB-XGA", 60, 1024, 768, 15385,
     220, 40,
     21, 7,
     60, 10,
     0,
     FB_VMODE_NONINTERLACED,
     FB_MODE_IS_DETAILED,},
}
```

The definition of fb_videomode in linux-3.0.35/include/linux/fb.h:

The name field is optional. If you input this value, it can be used in U-Boot environment settings.

The refresh field is the screen refresh frame rate, such as 60Hz, 70Hz. The resolution can be filled in the xres & yres fields.

The pixel clock (pixclock) is equaled to 10^{12}/(*Total horizontal line * Total vertical line * DCLK*). For example, the total horizontal line is 1344 DCLK, and total vertical number is 806 horizontal lines. The DCLK frequency is 60 MHz. Therefore, we can get 10^{12}/(1344*806*60) = 15385.

The *margin* values can be seen as front porch & back porch.

The *sync_len* means pulse width.

The *sync* value indicates the sync polarity (low or high).

```
struct fb_videomode {
     const char *name;      (optional)
     u32 refresh;            (optional)
     u32 xres;
     u32 yres;
     u32 pixclock;
```

```
        u32 left_margin;
        u32 right_margin;
        u32 upper_margin;
        u32 lower_margin;
        u32 hsync_len;
        u32 vsync_len;
        u32 sync;
        u32 vmode;
        u32 flag;
    };
```

3.7.5.2 Single Display Settings

```
# cat /sys/class//graphics/fb0/fsl_disp_dev_property
hdmi
# cat /sys/class//graphics/fb1/fsl_disp_dev_property
overlay
# cat /sys/class//graphics/fb2/fsl_disp_dev_property
ldb
# cat /sys/class//graphics/fb3/fsl_disp_dev_property
overlay
# cat /sys/class//graphics/fb4/fsl_disp_dev_property
ldb


2. Modify file:
vi /usr/share/imx_6q_display_config

original)
[master]
device = /dev/video17

++++++++++++++++++++++++++++++++

Modify:

[hdmi]
device = /dev/video17
fmt = RGBP
width = 1920
height = 1080
[ldb]
device = /dev/video19
fmt = RGBP
width = 800
height = 480
alpha = 0
```

```
[ldb1]
device = /dev/video20
fmt = RGBP
width = 800
height = 480
alpha = 0
```

3.7.5.3 Multi display video

gst-launch-1.0 playbin uri=file:////tools/Advantech.avi video-sink="overlaysink display-ldb=true display-ldb1=true"

3.7.6 Network Setup

Default: IP get form DHCP.

Manual: Set IP by below command:

#ifconfig eth0 192.168.0.1 up

ifconfig is to configure network interfaces, the manual page is as below.

```
SYNOPSIS
       ifconfig [-v] [-a] [-s] [interface]
       ifconfig [-v] interface [aftype] options | address ...
OPTIONS
       -a     display all interfaces which are currently available, even if
              down
       -s     display a short list (like netstat -i)
       -v     be more verbose for some error conditions
          interface
             The name of the interface. This is usually a driver name fol-
             lowed by a unit number, for example eth0 for the first Ethernet
             interface. If your kernel supports alias interfaces, you can
             specify them with eth0:0 for the first alias of eth0. You can
             use them to assign a second address. To delete an alias inter-
             face use ifconfig eth0:0 down. Note: for every scope (i.e. same
             net with address/netmask combination) all aliases are deleted,
             if you delete the first (primary).
          [aftype]
          up      This flag causes the interface to be activated. It is
                  implicitly specified if an address is assigned to the inter
                  face.

         down      This flag causes the driver for this interface to be shut down.
          address  The IP address to be assigned to this interface.
          netmask [addr]
             Set the IP network mask for this interface. This value defaults
             to the usual class A, B or C network mask (as derived from the
             interface IP address), but it can be set to any value.
          broadcast [addr]
```

```
        If the address argument is given, set the protocol broadcast
        address for this interface. Otherwise, set (or clear)
    the IFF_BROADCAST flag for the interface.
      del addr/prefixlen
        Remove an IPv6 address from an interface.
```

3.7.7 Storage (SATA /eMMC/SD Card)

The storages devices are named as follows:

Device	Name
SATA	/dev/sda
eMMC	/dev/mmcblk0
SD card	/dev/mmcblk1
Onboard SD card	/dev/mmcblk1
Carrier board SD card	/dev/mmcblk2

Chapter 4

System Recovery

This chapter introduces how to recover Linux operating system if it is damaged accidentally.

4.1 System Recovery

This section provides detail procedures of restoring the eMMC image. If you destroy the onboard flash image by accident, you can recover a system following these steps.

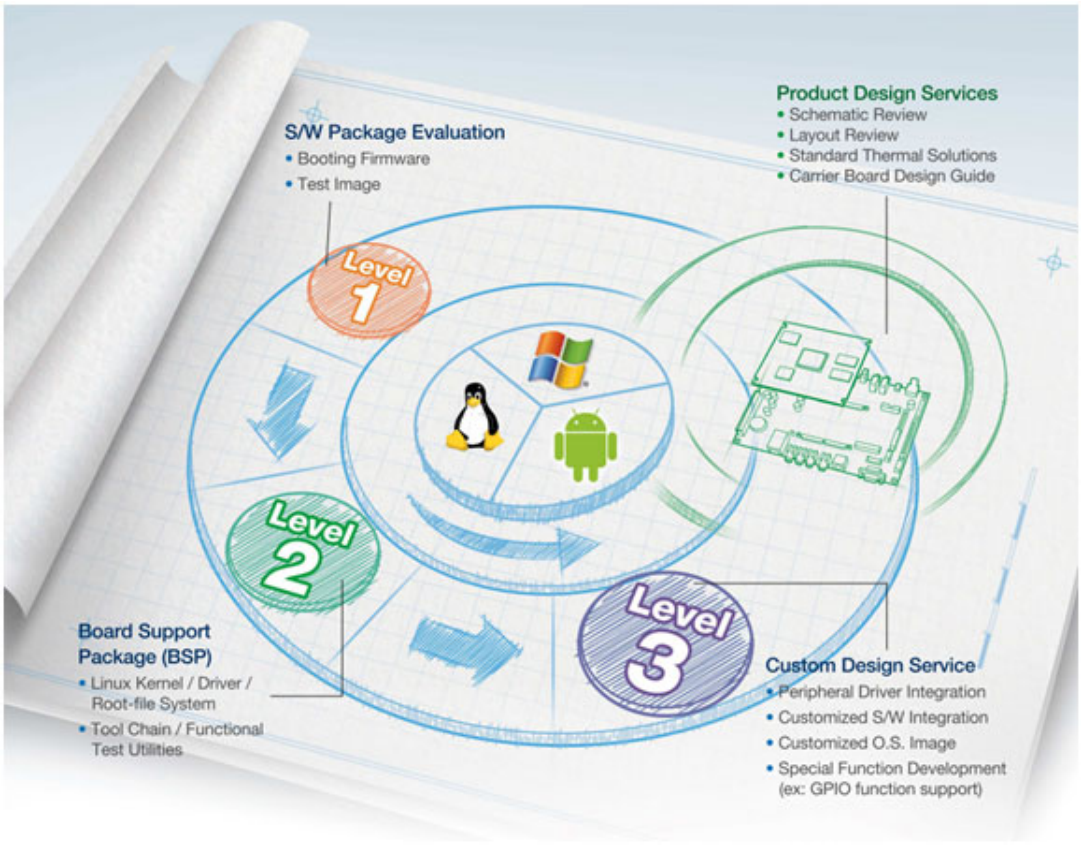
1. Copy “7421LIVxxxx.tar.bz2” package to your desktop.
2. Open "Terminal" on Ubuntu 12.04 LTS.
3. **`$sudo su`** (Change to “root” authority)
4. Input your password.
5. **`#cd Desktop/`**
6. **`#tar xvf 7421LIVxxxx.tar.bz2`** (Unzip files)
7. Insert one SD card to your developing computer
8. Check the SD card location, like /dev/sdf
9. **`#cd ./7421LBVxxxx_prebuilt_image`**
10. **`#dd if=7421LIVxxxx.img of=/dev/sdf`**
11. Please wait until dump disk is done
12. Connect console cable to debug port (CONSOLE) and open serial console program on Ubuntu 12.04 LTS, set baudrate to 115200. For detailed console setting, please refer to section 3.6.
13. On RSB-4411 platform, type **`#root`** (Login)
14. On RSB-4411 platform, type **`#cd /mk_inand`**
15. On RSB-4411 platform, type **`#./mkinand-linux.sh /dev/mmcblk0`**
16. On RSB-4411 platform, type “y “
    (Start to copy files, wait until it shows [Done])
17. Power off and remove this SD card.

Chapter 5

Advantech Services

This chapter introduces Advantech design in serviceability, technical support and warranty policy for RSB-4411 evaluation kit.

5.1 RISC Design-in Services



Advantech RISC Design-in Services help customers to reduce the time and work involved with designing new carrier boards. We handle the complexities of technical research and greatly minimize the development risk associated with carrier boards.

Easy Development

Advantech has support firmware, root file-system, BSP or other develop tools for customers. It helps customers to easy develop their carrier board and differentiate their embedded products and applications.

- Full Range of RISC Product Offerings
- Comprehensive Document Support

Design Assistance Service

Advantech provides a check list for engineers to check their schematics and also review service based on customer carrier board schematics. Those services are preventative, and help to catch design errors before they happen. This helps to save a lot of time and cost with regard to development of carrier boards.

- Schematic Review
- Placement and Layout Review
- Debugging Assistance Services
- General/Special Reference Design Database.

Thermal Solution Services

In order to provide quicker and more flexible solutions for customer's thermal designs. Advantech provides a thermal solution service including modularized thermal solutions and customized thermal solutions.

- Standard Thermal Solutions
- Customized Thermal Solutions

Embedded Software Services

Supports driver, software integration or customized firmware, root file-system and Linux image. Customer can save lot of time and focus on their core development.

- Embedded Linux/ Android OS
- Advantech boot loader Customization

With the spread of industrial computing, a whole range of new applications have been developed, resulting in a fundamental change in the IPC industry. In the past System Integrators (SI) were used to completing projects without outside assistance but now such working models have moved on. Due to diverse market demands and intense competition, cooperation for (both upstream and downstream) vertical integration has become a much more effective way to create competitive advantages. As a result, ARM-based CPU modules were born out of this trend. Concentrating all necessary components on the CPU module and placing other parts on the carrier board in response to market requirements for specialization, provides greater flexibility while retaining its low power consumption credentials.

Advantech has been involved in the industrial computer industry for many years and found that customers usually have the following questions when implementing modular designs.

General I/O design capability

Although customers possess the ability for vertical integration and have enough know-how and core competitiveness in the professional application field, the lack of expertise and experience in general power and I/O design causes many challenges for them, especially integrating CPU modules into their carrier board.

The acquisition of information

Even if the individual client is able to obtain sufficient information to make the right decision for the specialized vertical application, some customers encounter difficult problems dealing with platform design in general and communicating with CPU or chipset manufacturers, thereby increasing carrier board design difficulties and risk as well as seriously impacting on Time-to-market and lost market opportunities.

Software development and modification

Compared to x86 architectures, RISC architectures use simpler instruction sets, therefore the software support for x86 platforms cannot be used on RISC platforms. System integrators need to develop software for their system and do the hardware and software integration themselves. Unlike x86 platforms, RISC platforms have less support for Board Support Packages (BSP) and drivers as well. Even though driver support is provided, SI still have to make a lot of effort to integrate it into the system core. Moreover, the BSP provided by CPU manufacturers are usually for carrier board design, so it’s difficult for SI to have an environment for software development.

In view of this, Advantech proposed the concept of Streamlined Design-in Support Services for RISC-based Computer On Modules (COM). With a dedicated profes-

sional design-in services team, Advantech actively participates in carrier board design and problem solving. Our services not only enable customers to effectively distribute their resources but also reduce R&D manpower cost and hardware investment.

By virtue of a close interactive relationship with leading original manufacturers of CPUs and chipsets such as ARM, TI and NXP, Advantech helps solve communication and technical support difficulties, and that can reduce the uncertainties of product development too. Advantech's professional software team also focuses on providing a complete Board Support Package and assists customers to build up a software development environment for their RISC platforms.

Advantech RISC design-in services helps customers overcome their problems to achieve the most important goal of faster time to market through a streamlined RISC Design-in services.

Along with our multi-stage development process which includes: planning, design, integration, and validation, Advantech's RISC design-in service provides comprehensive support to the following different phases:

Planning stage

Before deciding to adopt Advantech RISC COM, customers must go through a complete survey process, including product features, specification, and compatibility testing with software. So, Advantech offers a RISC Customer Solution Board (CSB) as an evaluation tool for carrier boards which are simultaneously designed when developing RISC COMs. In the planning stage, customers can use this evaluation board to assess RISC modules and test peripheral hardware. What's more, Advantech provides standard software Board Support Package (BSP) for RISC COM, so that customers can define their product's specifications as well as verifying I/O and performance at the same time. We not only offer hardware planning and technology consulting, but also software evaluation and peripheral module recommendations (such as WiFi, 3G, BT). Resolving customer concerns is Advantech's main target at this stage. Since we all know that product evaluation is the key task in the planning period, especially for performance and specification, so we try to help our customers conduct all the necessary tests for their RISC COM.

Design stage

When a product moves into the design stage, Advantech will supply a design guide of the carrier board for reference. The carrier board design guide provides pin definitions of the COM connector with limitations and recommendations for carrier board design, so customers can have a clear guideline to follow during their carrier board development. Regarding different form factors, Advantech offers a complete pin-out check list for different form factors such as Q7, ULP and RTX2.0, so that customers can examine the carrier board signals and layout design accordingly. In addition, our team is able to assist customers to review the placement/layout and schematics to ensure the carrier board design meets their full requirements. For software development, Advantech RISC software team can assist customers to establish an environment for software development and evaluate the amount of time and resources needed. If customers outsource software development to a 3rd party, Advantech can also cooperate with the 3rd party and provide proficient consulting services. With Advantech's professional support, the design process becomes much easier and product quality will be improved to meet their targets.

Integration stage

This phase comprises HW/SW integration, application development, and peripheral module implementation. Due to the lack of knowledge and experience on platforms, customers need to spend a certain amount of time on analyzing integration problems. In addition, peripheral module implementation has a lot to do with driver designs on carrier boards, RISC platforms usually have less support for ready-made drivers on the carrier board, therefore the customer has to learn from trial and error and finally get the best solution with the least effort. Advantech's team has years of experience in customer support and HW/SW development knowledge. Consequently, we can support customers with professional advice and information as well as shortening development time and enabling more effective product integration.

Validation stage

After customer's ES sample is completed, the next step is a series of verification steps. In addition to verifying a product's functionality, the related test of the product's efficiency is also an important part at this stage especially for RISC platforms.

As a supportive role, Advantech primarily helps customers solve their problems in the testing process and will give suggestions and tips as well. Through an efficient verification process backed by our technical support, customers are able to optimize their applications with less fuss. Furthermore, Advantech's team can provide professional consulting services about further testing and equipment usage, so customers can find the right tools to efficiently identify and solve problems to further enhance their products quality and performance.

5.2 Contact Information

Below is the contact information for Advantech customer service

Region/Country	Contact Information
America	1-888-576-9688
Brazil	0800-770-5355
Mexico	01-800-467-2415
Europe (Toll Free)	00800-2426-8080
Singapore & SAP	65-64421000
Malaysia	1800-88-1809
Australia (Toll Free)	1300-308-531
China (Toll Free)	800-810-0345 800-810-8389 Sales@advantech.com.cn
India (Toll Free)	1-800-425-5071
Japan (Toll Free)	0800-500-1055
Korea (Toll Free)	080-363-9494 080-363-9495
Taiwan (Toll Free)	0800-777-111
Russia (Toll Free)	8-800-555-01-50

You can also reach our service team through the website below; our technical support engineer will provide quick response once the form is filled out:

http://www.advantech.com.tw/contact/default.aspx?page=contact_form2&subject=Technical+Support

5.3 Technical Support and Assistance

For more information about this and other Advantech products, please visit our web-site at:

http://www.advantech.com/

http://www.advantech.com/ePlatform/

For technical support and service, please visit our support website at:

<http://support.advantech.com.tw/support/>

1. Visit the Advantech web site at www.advantech.com/support where you can find the latest information about the product.
2. Contact your distributor, sales representative, or Advantech's customer Service center for technical support if you need additional assistance. Please have the following information ready before you call:
   - Product name and serial number
   - Description of your peripheral attachments
   - Description of your software (operating system, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

ADVANTECH

Enabling an Intelligent Planet

www.advantech.com

Please verify specifications before quoting. This guide is intended for reference purposes only.

All product specifications are subject to change without notice.

No part of this publication may be reproduced in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission of the publisher.

All brand and product names are trademarks or registered trademarks of their respective companies.

© Advantech Co., Ltd. 2017