

User Manual

ROM-3310

RISC base RTX 2.0 module-TI
Cortex A8 AM3352 1Ghz

ADVANTECH

Enabling an Intelligent Planet

Copyright

The documentation and the software included with this product are copyrighted 2015 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

ARM is trademarks of ARM Corporation.

TI is trademarks of TI Corporation

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 merchandise 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.

Declaration of Conformity

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.

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.

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.

Packing List

Before installation, please ensure the following items have been shipped.

- 1 ROM-3310 RTX Module

Optional Accessories

Part No.	Description
1757002943	ADAPTER 100-240V 65W 19V 3.42A W/O P 9NA0651217
9696MEG700E	ROM-EG70 A101 ASS'Y SPI/I2C Board For ROM-DB5900
9696MEG520E	ROM-EG52 A101 ASS'Y CODEC Board for ROM-3900
1700019077	M Cable USB-A(M)/M-USB(M) 150cm OTG Device
1700019076	M Cable USB-A(F)/M-USB(M) 10cm OTG Host
SQF-ISDS1-4G-82C	SQF SD C6 SLC 4G, 1CH
1700001524	Power Cord 3P UL 10A 125V 180cm
9696ED2000E	ASS'Y ROM-ED20 DB A101-1
170203180A	Power Cord 3P UK 2.5A/3A 250V 1.83M
1700001524	Power Cord 3P UL 10A 125V 180cm

Development board

Part No.	Description
ROM-DB3900-SW00E	Development carrier board for RTX Rev.A1

For more information please refer to "Advantech Baseboard Check List" and "Evaluation Board Reference Schematic".

You can download "Advantech Baseboard Check List" and "Evaluation Board Reference Schematic" from <http://com.advantech.com/>

Ordering Information

Model Number Description

	Commercial Grade	Industrial grade
Part No.	ROM-3310CS-MCA1E	ROM-3310WS-MCA1E
CPU	TI Cortex A8 AM3352 1GHz	TI Cortex A8 AM3352 1GHz
Memory	512MB	512MB
NAND Flash	4GB	4GB
Nor Flash	4MB	4MB
24bit TTL	Yes	Yes
USB	2	2
Giga LAN	1	1
I2S	1	1
I2C	1	1
GPIO	10	10
UART	5	5
CAN	2	2
SPI	1	1
Heat sink	N/A	Optional
Operation Temp.	0 ~60°C	-40 ~85°C

Contents

Chapter 1	General Introduction	1
1.1	Introduction	2
1.2	Product Features.....	2
1.3	Mechanical Specifications.....	3
1.3.1	Dimensions (mm).....	3
1.3.2	Heat Spreader Dimensions (mm)	3
1.4	Electrical Specifications	3
1.4.1	Power supply Voltage	3
1.5	Environmental Specifications	3
1.5.1	Operating temperature.....	3
1.5.2	Operating Humidity	3
1.5.3	Storage temperature	3
1.5.4	Storage Humidity	3
Chapter 2	H/W Installation.....	5
2.1	Board Connector	6
2.1.1	Connector List.....	6
	Table 2.1: External IO Connector	6
	Table 2.2: SW10 (Boot selection).....	6
	Table 2.3: CN1803 (MCU programming port)	6
	Table 2.4: CN1201 (JTAG connector)	7
	Table 2.5: CN1202 (Debug connector).....	7
2.2	ROM-3310 Board Block Diagram.....	8
	Figure 2.1 ROM-3310 Block Diagram.....	8
2.3	ROM-3310 Board Looks	8
Chapter 3	Software Functionality	9
3.1	Test Tools	10
3.2	eMMC Test.....	10
3.3	USB Test.....	11
3.4	SD Test	11
3.5	SPI Nor Flash Test.....	13
3.6	GPIO Test	14
	Table 3.1: ROM-3310 GPIO Default Setting	14
3.7	VGA Test.....	15
3.8	I2C Test.....	15
3.9	CAN Test.....	17
3.10	I2S Test.....	17
3.11	LAN Test	18
3.12	RS232 /RS422/RS485 Test	19
3.13	Watchdog Timer Test.....	20
3.14	RGB Test	21
3.15	Package Content.....	22
3.15.1	Pre-built System Image.....	22
3.15.2	Source Code Package	22
3.15.3	Set up Build Environment	25
3.15.4	Build Instructions.....	26
3.15.5	Source Code Modification	27
	Figure 3.1 Linux Kernel Configuration	27
	Figure 3.2 Selecting Seiko Instruments S-35390A.....	28
3.15.6	Create a Linux System Boot Media	29

	3.15.7	Debug Message.....	30
		Figure 3.3 HyperTerminal Settings for Serial Console Setup	30
	3.15.8	Linux System Configuration and Use	30
		Figure 3.4 IP Configuration.....	34
		Figure 3.5 Date/Time Settings.....	35
		Figure 3.6 Serial Control.....	36
		Figure 3.7 Matrix.....	36
	3.15.9	Development Guide and Reference	39
Chapter	4	System Recovery	43
	4.1	System Recovery.....	44
Chapter	5	Advantech Services	45
	5.1	RISC Design-in Services	46
	5.2	Contact Information.....	49
	5.3	Technical Support and Assistance.....	50
	5.4	Global Service Policy	50
		5.4.1 Warranty Policy.....	50
		5.4.2 Warranty Period.....	50
		5.4.3 Repairs under Warranty.....	50
		5.4.4 Exclusions from Warranty.....	51
	5.5	Repair Process	51
		5.5.1 Obtaining an RMA Number.....	51
		5.5.2 Returning the Product for Repair	51
		5.5.3 Service Charges	52
		5.5.4 Repair Report	53
		5.5.5 Custody of Products Submitted for Repair	53
		5.5.6 Shipping Back to Customer	53

Chapter 1

General Introduction

This chapter gives background information on the ROM-3310.

Sections include:

- Introduction
- Specification

1.1 Introduction

ROM-3310 adopts TI Cortex A8 AM3352 1Ghz architecture as its SOC solution. The main features of this platform are followed by RTX 2.0 standard, with a heat-sink-less, compact, reliable & great power management. Therefore, ROM-3310 platform is suitable for following applications:

- Industrial data collector

And the main features of TI Cortex A8 processors are shown as following:

- ARM Cortex™-A8 high performance processor, single core 1GHz
- Support Parallel RGB 24bit TTL
- TI AM3352 Cortex A8 Technology support low power consumption
- Capabilities of I/O expansion: UART(5), 24bit TTL, USB Host, USB OTG, Gigabit Ethernet, SD, GPIO(10), I2C(1), SPI(1), I2S, CAN with 3.3V level (2)
- Supports Linux Kernel 3.2.0
- Support working temperature 0 ~ 60°C & -40 ~ 85°C

1.2 Product Features

Form factor		RTX 2.0
Processor System	CPU	TI Cortex A8 AM3352 1Ghz
Memory	Technology	DDR3 800 MHz
	Capacity	On-board DDR3 512MB
	Flash	4GB eMMC for O.S. and 4MB NOR Flash for ADV Loader
Graphic	24bit TTL	Parallel RGB VGA
Ethernet	Transceiver	RTL8211
	Speed	1 x 10/100/1000 Mbps
Watch Dog Timer	HW WDT	TI MSP430G2202 (time out 1~6553s, default 60s)
I/O	SDIO	4-bit STD SD
	Serial Port	5 x UART (1 x RS485, 4 x RS232)
	SPI	1 x SPI
	CAN	2 x CAN 2.0B
	GPIO	10 x GPIO
	I2C	1 x I2C
	I2S	1 x I2S
	USB	1 x USB 2.0 host & 1 x USB 2.0 OTG
O.S	Linux Kernel	V 3.2.0
Dimensions	W x L x T	68 x 68 mm x 2mm
Power	DC-In	5~24V
Environment	Operational Temperature	0 ~ 60°C/ -40 ~ 85° C
	Operating Humidity	0% ~ 90% relative humidity, non-condensing
Certifications	Level	CE / FCC / Class B

1.3 Mechanical Specifications

1.3.1 Dimensions (mm)

RTX, 68mm(W) * 68mm(D) * 2mm(T)

1.3.2 Heat Spreader Dimensions (mm)

68mm*68mm (D x W, Heat sink for option)

1.4 Electrical Specifications

1.4.1 Power supply Voltage

Voltage requirements: 5~24V DC-In

1.5 Environmental Specifications

1.5.1 Operating temperature

Operating temperature: 0 ~ 60° C (32~140°F) or -40 ~ 85° C (-40~185°F)

The operating temperature refers to the environmental temperature for the model.

1.5.2 Operating Humidity

Operating humidity: 0% ~ 90% relative humidity, non-condensing

1.5.3 Storage temperature

Storage temperature: -40~85° C

1.5.4 Storage Humidity

Relative humidity: 95% @ 60°C

Chapter 2

H/W Installation

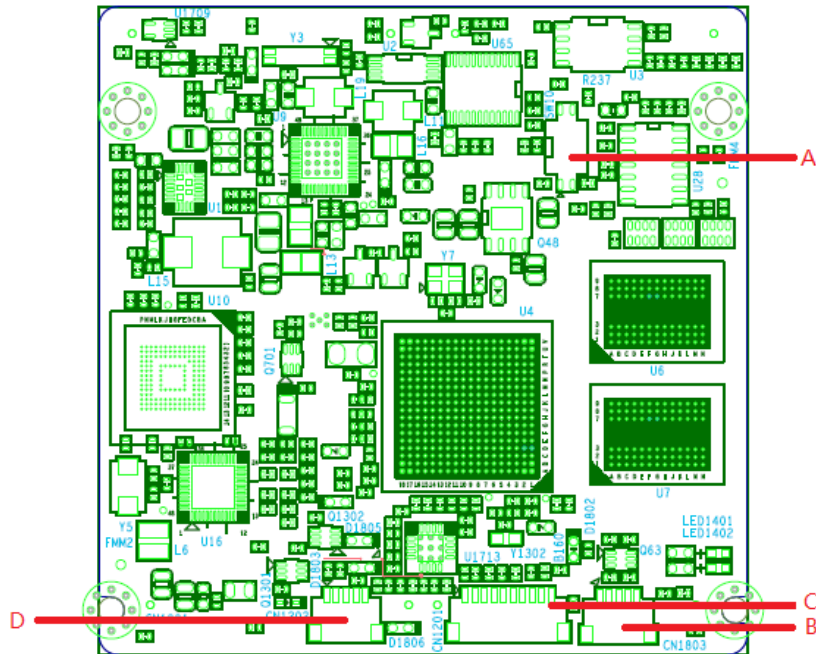
This chapter gives mechanical and connector information on the ROM-3310 CPU Computer on Module.

Sections include:

- Connector Information
- Mechanical Drawing

2.1 Board Connector

The board has four connectors that allow you to configure your system to your application.



2.1.1 Connector List

Table 2.1: External IO Connector

Position	Description	
SW10	Boot selection	A
CN1803	MCU programming port	B
CN1201	JTAG connector	C
CN1202	Debug port	D

Table 2.2: SW10 (Boot selection)

Jumper	Mode	Jumper	Mode
1-ON	SPI-ROM	1-OFF	SD
2-OFF		2-ON	

Table 2.3: CN1803 (MCU programming port)

Pin	Signal	Pin	Signal
1	JTAG VCC	2	SBWTDIO
3	SBWTCK	4	SBWTCK
5	GND		

Table 2.4: CN1201 (JTAG connector)

Pin	Signal	Pin	Signal
1	+3.3V	2	+3.3V
3	JTAG_TMS	4	JTAG_TDO
5	JTAG_TDI	6	JTAG_TCK
7	JTAG_EMU0	8	JTAG_RESETn
9	JTAG_EMU1	10	GND

Table 2.5: CN1202 (Debug connector)

Pin	Signal	Pin	Signal
1	+3.3V	2	COM0_TX
3	COM0_RX	4	GND

SW2 (Boot select)

Location on carrier board ROM-DB3900

ROM-3310 default boot from module eMMC flash so please set carrier board SW2 110, The detail please refer to rom-db3900 manual.

BOOT_SEL2#	BOOT_SEL1#	BOOT_SELOW	Boot Source
GND	GND	GND	Carrier SATA
GND	GND	Float	Carrier SD card
GND	Float	GND	Carrier eMMC Flash
GND	Float	Float	Carrier SPI
Float	GND	GND	Module device (NAND, NOR) - vendor specific
Float	GND	Float	Remote boot (GBE, serial) - vendor specific
Float	Float	GND	Module eMMC
Float	Float	Float	Module SPI

2.2 ROM-3310 Board Block Diagram

Below is the block diagram of ROM-3310.

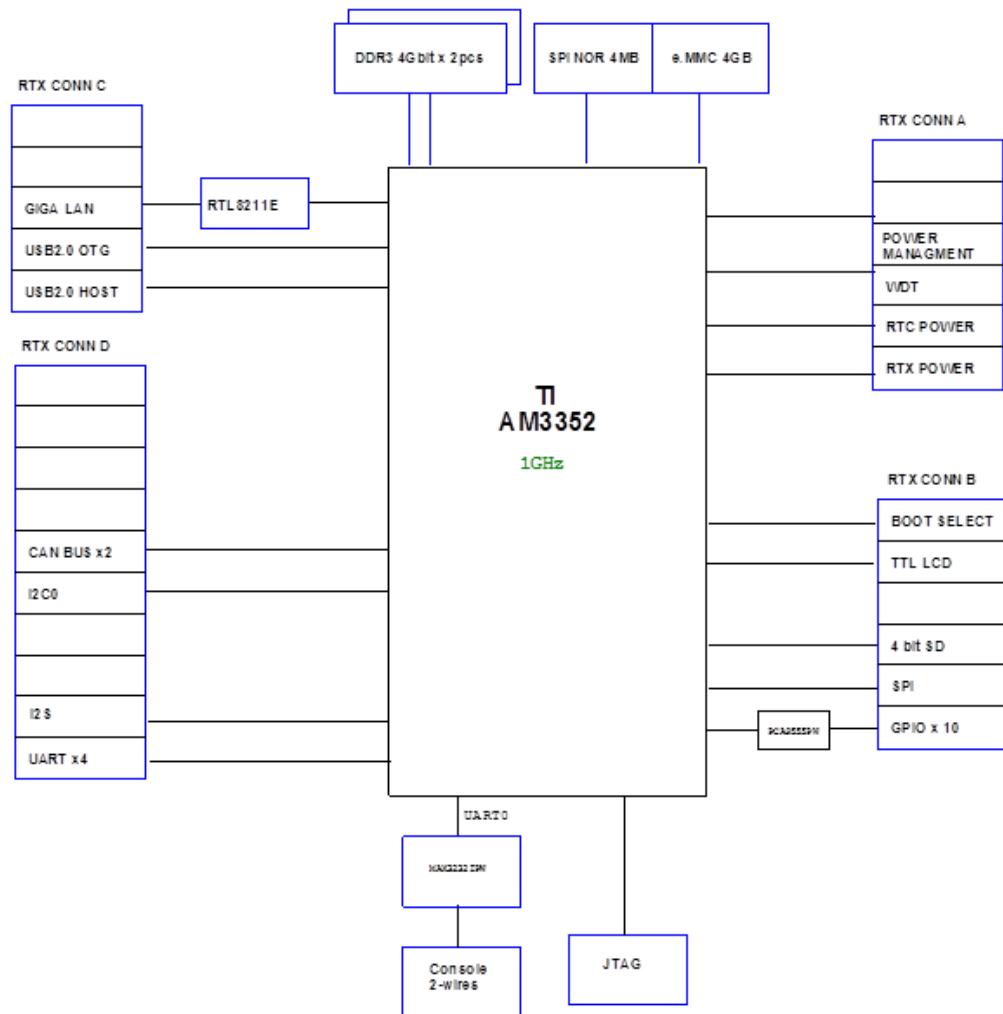
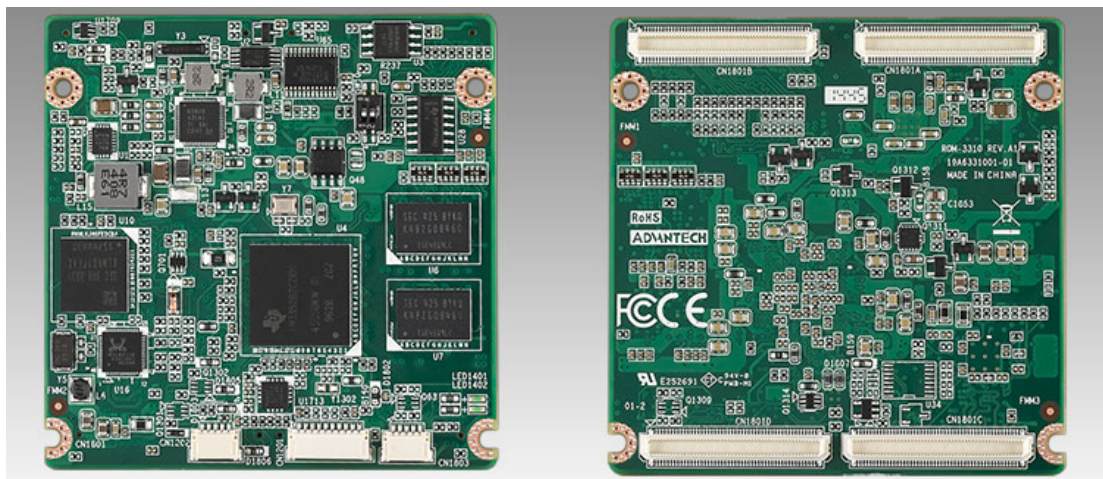


Figure 2.1 ROM-3310 Block Diagram

2.3 ROM-3310 Board Looks



Chapter 3

Software Functionality

This chapter details the software programs on the ROM-3310 platform.

3.1 Test Tools

All test tools must be verified on ROM-3310 Evaluation kit, please prepare required test fixtures before verifying each specified I/O. If you have any problem to get the test fixture, please contact your Advantech contact window for help.

Note! Boot Linux System, input "root" & no password.



3.2 eMMC Test

1. Erase and check.

```
#dd if=/dev/zero of=/dev/mmcblk0 bs=1024 count=1 seek=25118
1+0 records in
1+0 records out

#hexdump -C /dev/mmcblk0 -s 25720832 -n 16
01887800  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|.....|
```

2. Write and check.

```
#echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk0 bs=1024 count=1
seek=25118
0+1 records in
0+1 records out

#hexdump -C /dev/mmcblk0 -s 25720832 -n 16
01887800  30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
|0123456789ABCDEF|
```

Note! Insert SD Card, if insert SD Card, the eMMC NAND Flash is mmcblk1.



3.3 USB Test

1. Insert USB flash disk then assure it is in ROM-3310 device list.
2. Erase and check.


```
#dd if=/dev/zero of=/dev/sda bs=1024 count=1 seek=25118
1+0 records in
1+0 records out

#hexdump -C /dev/sdb -s 25720832 -n 16
01887800  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|.....|
```

3. Write and check.

```
#echo -n "0123456789ABCDEF" | dd of=/dev/sda bs=1024 count=1
seek=25118
0+1 records in
0+1 records out

#hexdump -C /dev/sda -s 25720832 -n 16
01887800  30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
|0123456789ABCDEF|
```

- Note!**  1. TI AM3352 has the limitation on USB device collection, we recommend the follow brands: Logitech K120 / Lenovo K5819 LXH-EKB-10YA / RAPOO 1800.Pro / Dell MS111-P / Microsoft Wired Keyboard 200 (Model:1406) and so on.
2. This operation may damage the data stored in USB flash disk. Please make sure there is no critical data in the USB flash disk being used for this test. If your U Disk size is small, the seek value need to be small.

3.4 SD Test

1. When booting from eMMC, you would see only below directories:

```
#ls /dev/mmcblk*

/dev/mmcblk0 /dev/mmcblk0boot0 /dev/mmcblk0boot1 /dev/
mmcblk0p1 /dev/mmcblk0p2
```

2. Insert SD card to SD card slot (SD1) and check your device again. You should be able to see more directories. /dev/mmcblk1 is the SD card storage (Ex. SD Card have 2 partitions).

```
#ls /dev/mmcblk*

/dev/mmcblk0 /dev/mmcblk0boot1 /dev/mmcblk1 /dev/
mmcblk1p2
/dev/mmcblk0boot0 /dev/mmcblk0p1 /dev/mmcblk1p1 /dev/
mmcblk0p2
```

3. Erase and check.

```
#dd if=/dev/zero of=/dev/mmcblk1 bs=1024 count=1 seek=25118
1+0 records in
1+0 records out

#hexdump -C /dev/mmcblk1 -s 25720832 -n 16
01887800  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
|.....|
```

4. Write and check.

```
#echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk1 bs=1024 count=1
seek=25118
0+1 records in
0+1 records out

#hexdump -C /dev/mmcblk1 -s 25720832 -n 16
01887800  30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
|0123456789ABCDEF|
```

Note! *Please make sure parameter “seek” is equal to 25118 as indicated in red in above codes. If you create the file to a wrong sector, that may damage the system.*



3.5 SPI Nor Flash Test

```

#cd /unit_tests/
#./AutoRun_spi.sh
#####---SPI Test for AM335X---#####

the spi falsh info:

mtd.type = MTD_NORFLASH
mtd.flags = MTD_CAP_NORFLASH
mtd.size = 4194304 (4M)
mtd.erasesize = 4096 (4K)
mtd.writesize = 1
mtd.oobsize = 0
regions = 0

Erased 4096 bytes from address 0x00030000 in flash
Copied 4096 bytes from address 0x00030000 in flash to temp.img
0000000 ffff ffff ffff ffff ffff ffff ffff ffff
*
0001000

=====
#####                               SPI mtdblock0  Read 1 PASS !!!
=====

1+0 records in
1+0 records out
0000000 0000 0000 0000 0000 0000 0000 0000 0000 0000
*
0001000
Copied 4096 bytes from zero.img to address 0x00030000 in flash
Copied 4096 bytes from address 0x00030000 in flash to temp.img
0000000 0000 0000 0000 0000 0000 0000 0000 0000 0000
*
0001000

=====
##### -----SPI mtdblock0 Write 0 PASS !!!
=====

=====
#####-----> SPI Test all mtdblock0 PASS !!!
=====

=====
##### Finish SPI all blocks  Test PASS !!!
=====

```

Note! ROM3310CS-MCA1E default support 5 x UART & 1 x SPI.



3.6 GPIO Test

Table 3.1: ROM-3310 GPIO Default Setting

ROM-DB3900	Linux OS /sys/class/gpio	ROM-DB3900 GPIO CHIP
GPIO0	gpio200	GPIO(3)
GPIO1	gpio201	GPIO(4)
GPIO2	gpio202	GPIO(5)
GPIO3	gpio203	GPIO(6)
GPIO4	gpio204	GPIO(7)
GPIO5	gpio205	GPIO(8)
GPIO6	gpio206	GPIO(9)
GPIO7	gpio207	GPIO(10)
GPIO8	gpio208	GPIO(11)
GPIO9	gpio209	GPIO(12)

```
#cd /sys/class/gpio
```

You can use “ls” to list all GPIO devices, and you should also see GPIO ports in above table.

Example of testing GPIO

Set gpio200 GPI(in)

```
#echo in > ./gpio200/direction
#cat ./gpio200/direction
in
```

Set gpio201 GPO (out)

```
#echo out > ./gpio201/direction
#cat ./gpio201/direction
out
```

Set gpio201 GPO value “1”

```
#echo 1 > ./gpio201/value
```

Get gpio200 value

```
#cat ./gpio200/value
1
```

As you can see in above procedure A and B, we set gpio 200 as GPI and gpio 201 as GPO, once we send data out from gpio 201, it should be able to receive the same data from gpio 200.

3.7 VGA Test

Testing through gst-launch (for default single display)

1. `#cd /unit_tests/.`

```
#./AutoRun_lvds.sh
```

2. Then you can see the video demo on the default display screen.



3.8 I2C Test

There is one I2C buses in ROM-3310.

```
#ls /sys/class/i2c-dev
i2c-1
#i2cdetect -l
i2c-1  i2c                OMAP I2C adapter                I2C
adapter
```

Please try below command to know if there is any device connected to i2c bus 1.

```
#i2cdetect -y -r 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  UU  --  --  --  --  --
20:  --  --  UU  --  --  --  --  --  --  UU  --  --  --  UU  --  --
30:  UU  UU  UU  UU  UU  UU  UU  UU  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  49  --  --  --  --  --  --
50:  50  --  --  --  --  --  --  UU  --  59  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  UU  --
```

The 0x2d0x2d is the PMIC address. So you can try below command to know I2C bus is work or not.

```
# i2cdump -f -y 1 0x2d

No size specified (using byte-data access)
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
10: 00 00 00 00 00 00 00 00 00 00 00 00 9f 0d 09 00
.....???.
20: 41 0d 3d 00 05 2e 00 05 00 00 00 00 00 00 00 00
A?=.?..?.....
30: 09 0d 01 0d 0d 0d 09 01 0d 00 00 00 00 00 3b 70
?????????.....;p
40: 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5.....
50: 16 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00
??.....
60: 0a 0a 0a 0a 0a 0a 00 00 00 00 00 00 00 00 00 00
??????.....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
80: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
?.....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....
```


3.9 CAN Test

1. Check CAN network device.

```
# ls /sys/class/net/
can0 can1 eth0 lo
```

2. Test CAN device (Connect CAN0 & CAN1).

```
#canconfig can0 stop >/dev/null;canconfig can0 bitrate 1000000
>/dev/null;canconfig can0 start >/dev/null
#canconfig can1 stop >/dev/null;canconfig can1 bitrate 1000000
>/dev/null;canconfig can1 start >/dev/null
#candump can0 >can0.log &
#candump can1 >can1.log &
# cansend can0 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
# cansend can1 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
#cat can0.log
interface = can0, family = 29, type = 3, proto = 1
<0x001> [8] 11 22 33 44 55 66 77 88
<0x001> [8] 11 22 33 44 55 66 77 88
#cat can1.log
interface = can1, family = 29, type = 3, proto = 1
<0x001> [8] 11 22 33 44 55 66 77 88
<0x001> [8] 11 22 33 44 55 66 77 88
```

If can0.log and can1.log store receive data the same, it CAN test PASS

3.10 I2S Test

MIC IN command as following: (44100 is the sample rate)

```
#amixer sset 'Right PGA Mixer Mic3R' on
#arecord -t wav -c 1 -r 44100 -d 5 2.wav
```

Audio out command as following:

```
#aplay /unit_tests/Advantech.wav
```

3.11 LAN Test

ROM-3310 sets DHCP as default network protocol.

```
#ifconfig
eth0      Link encap:Ethernet  HWaddr 78:A5:04:DD:E4:9C
          inet addr:172.21.170.76  Bcast:172.21.171.255
Mask:255.255.254.0
          UP BROADCAST RUNNING ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:16740 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1454 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1935525 (1.8 MiB)  TX bytes:198083 (193.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:130 errors:0 dropped:0 overruns:0 frame:0
          TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:234899 (229.3 KiB)  TX bytes:234899 (229.3 KiB)
```

If you would like to config IP manually, please use below command:

```
#ifconfig eth0 xxx.xxx.xxx.xxx up
```

Here is a real case for your reference. The hosts(ROM-3310) IP is 192.168.0.10; the target (A desktop computer) IP is 192.168.0.12

```
#/etc/init.d/netplugd stop
#ifconfig eth0 down
#ifconfig eth0 192.168.0.10
#ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 78:A5:04:DD:E4:9C
          inet addr:192.168.0.10  Bcast:172.168.0.255
Mask:255.255.255.0
          UP BROADCAST RUNNING ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:16740 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1454 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1935525 (1.8 MiB)  TX bytes:198083 (193.4 KiB)
```

The target computer (Client) IP address is 192.168.0.12, so we can use below command to see if we can get any response from the client.

```
#ping 192.168.0.12

PING 192.168.0.12 (192.168.0.12): 56 data bytes
64 bytes from 192.168.0.12: seq=0 ttl=128 time=7.417 ms
64 bytes from 192.168.0.12: seq=1 ttl=128 time=0.203 ms
64 bytes from 192.168.0.12: seq=2 ttl=128 time=0.300 ms

--- 172.17.20.192 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
```

Note! The target computer (Client) firewall need close.



3.12 RS232 /RS422/RS485 Test

As you can see below, there are 5 UART supported by ROM-3310. /dev/tty00 is reserved for ROM-3310 debug port (ROM-3310 CN1202), the rest UART ports could be applied by user.

Test RSB232

```
#cd /unit_tests/
#./AutoRun_uart232

====test rs232!====
rs232 number: 4
/dev/tty01 PASS!
/dev/tty02 PASS!
/dev/tty03 PASS!
/dev/tty04 PASS !

+-----+
| [RS232] Test Pass!|
+-----+
```

Note! *UART1 support RSB232/RSB422/RSB485, test rsb3232 mode, please jump rsb232 mode, you can refer the detail jump information from ROM-DB3900 what a develop board of Advantech.*



UART1 Test RSB422

```
#cd /unit_tests/
#./AutoRun_uart422 -p /dev/tty01 -t 1

Open uart /dev/tty01 PASS ....

->Writing : hello world!
->Reading : hello world!
->TX/RX Signal pass

+-----+
| UART RS422 Testing PASS |
+-----+
```

Note! *UART1 Jump RSB422 mode, you can refer the detail jump information from ROM-DB3900 what a develop board of Advantech.*



UART1 Test RSB485

```
#cd /unit_tests/  
#./AutoRun_uart485 -p /dev/tty01 /dev/tty04 -t 6  
  
Open uart /dev/tty01 OK ....  
Open uart /dev/tty04 OK ....  
  
Writing : helloworld!  
Reading : helloworld!  
->TX/RX Signal pass  
+-----+  
| UART RS485 Testing PASS |  
+-----+  
  
Close uart /dev/tty01 OK ....  
Close uart /dev/tty04 OK ....
```

Note! *UART1 Jump RSB485 mode, you can refer the detail jump information from ROM-DB3900 what a develop board of Advantech.*



3.13 Watchdog Timer Test

Executing 'AutoRun_WTD '

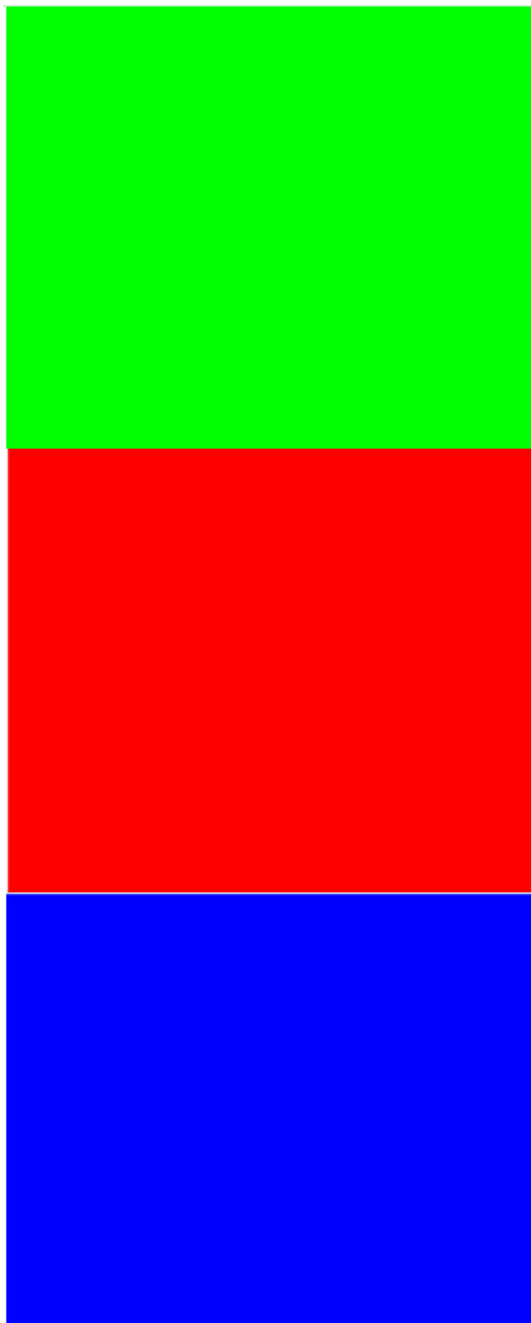
```
#cd /unit_tests  
#./AutoRun_WTD 15  
  
Get the timeout value from driver: timeout = 60 seconds  
  
Now, we set the timeout value to 10 seconds  
Get the timeout value from driver again: timeout = 10 seconds  
  
Setting succeeded and watchdog is enabled.  
Feed the watchdog every 15 seconds.  
Feed watchdog!  
  
After reboot, user will see the follow the boot messages:  
  
U-Boot SPL 2013.01.01-svn132 (Sep 28 2014 - 11:39:20)  
musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, bulk combine, bulk  
split, HB-ISO Rx, HB-ISO Tx, SoftConn)  
musb-hdrc: MHDRC RTL version 2.0  
musb-hdrc: setup fifo_mode 4  
musb-hdrc: 28/31 max ep, 16384/16384 memory  
USB Peripheral mode controller at 47401000 using PIO, IRQ 0  
musb-hdrc: ConfigData=0xde (UTMI-8, dyn FIFOs, bulk combine, bulk  
split, HB-ISO Rx, HB-ISO Tx, SoftConn)
```

3.14 RGB Test

Execute the following commands to run the Photo demo application on ROM-3310.

```
#cd /unit_tests/  
#./AutoRun_RGB  
  
test framebuffer!  
Please enter the test result (y:Pass , n:Fail)  
Y  
  
+-----+  
| [Frame Buffer] Test Pass! |  
+-----+
```

Then you can see the White, Red, Blue, Green, Black, color on the default display screen.



ROM-3310 platform is an embedded system with Linux kernel 3.2.0 inside. It contains all system-required shell commands and drivers ready for ROM-3310 platform. We do not offer IDE developing environment in ROM-3310 BSP, users can evaluate and develop under Ubuntu 12.04LTS environment.

There are three major boot components for Linux, “u-boot.img”, “ulmage” and “File System”. The “u-boot.bin” is for initializing peripheral hardware parameters; the “ulmage” is the Linux kernel image and the “File System” is for Linux O.S. used.

It will not be able to boot into Linux environment successfully if one of above two files is missing from booting media (SD card, onboard flash)

The purpose of this chapter is to introduce software development of ROM-3310 to you, so that you can develop your own application(s) efficiently.

ROM-3310 is designed for supporting Linux host only so you may fail developing your AP on Windows/Android host PC. For now the official supported host version is Ubuntu 12.04 LTS, host PC in any other version may have compatibility issue. In this case, we strongly recommend to have Ubuntu 12.04 LTS installed to your host PC before start ROM-3310 evaluation/development.

3.15 Package Content

We would offer you two different kinds of Linux package for ROM-3310. One is pre-built system image for system recovery another is source code package (BSP).

3.15.1 Pre-built System Image

You are able to find the pre-built image 3310LIVxxxx_yyyy-mm-dd.tar.gz from ROM-3310 evaluation kit DVD image downloaded from Advantech website. ROM-3310 supports booting from SD card so you can extract the image to SD card then dump the image file to onboard eMMC to complete system recovery.

3.15.2 Source Code Package

ROM-3310 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. ROM-3310 source code package is composed of six main folders: “cross_compiler”, “document”, “image”, “package”, “scripts”, and “source”.

Note! *ROM-3310 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 ROM3310 package contents:

Folder name	Description
cross_compiler	This folder contains source code for cross compiler.
document	This folder contains user guide.
image	This folder contains the ulmage, u-boot.img
image/rootfs	This folder contains Linux root file system
package	This folder contains source code provided by TI without any modification
scripts	This folder contains scripts for configure system and compile images automatically.
source	This folder contains source code owned by Advantech.

3.15.2.1 cross_compiler

You can use the cross compiler toolchain to compile the ulmage and related applications. (gcc version is 4.7.3 20130226)

3.15.2.2 document

User guide of how to setup up the environment of development.

3.15.2.3 image

This folder includes ulmage & u-boot.img.

3.15.2.4 image/rootfs

Linux adopts Hierarchical File System (HFS), image/rootfs is the Linux file system in highest level of the tree structure. image/rootfs is just like the trunk of the tree. Its sub-directories are the branches and the files in these directories are the leaves of the tree. image/rootfs contains all subdirectories and files used in the file system, that's why it is called the root of the whole file system.

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!
- 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.
- unit_tests → just for sample test tools

3.15.2.5 scripts

Some scripts provided by Advantech will help you configure system or build the images more quickly. Please check them as follows:

- 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 “ulmage” and copy the “ulmage” to “image” folder after building.
- mksd-linux.sh → A script to setup up a bootable SD card if users build their images.

3.15.2.6 source

This folder contains sub-directories “linux-3.2.0-psp04.06.00.11” and “u-boot-2013.01.01-psp06.00.00.00”. They are the source codes of the Linux kernel and U-boot.

Linux is OS that is including true multitasking, virtual memory, shared libraries, demand loading, shared copy-on-write executables, proper memory management, and multitask networking.

Linux is easily portable to most general-purpose 32-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-3.2.0” are listed as following:

- 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-3.2.0-ppc04.06.00.11/Documentation`, you can find the kernel-specified installations and notes for drivers. You can refer to `./source/ linux-3.2.0-ppc04.06.00.11/Documentation/00-INDEX` for a list of the purpose of each README/note.

3.15.3 Set up Build Environment

All instructions in this guide are based on Ubuntu 12.04 LTS developing environment. Please install the Ubuntu 12.04 LTS at your PC/NB in advance.

When you obtain the ROM-3310 Linux source code package, please refer to following instructions to extract to your developing environment:

1. Copy "335XLBVxxxx_yyyy-mm-dd.bin" package to `/root/`.
2. Start your "Terminal" on Ubuntu 12.04 LTS.
3. `$sudo su` (Change to "root" authority)
4. Input user password
5. `#cd /root/`
6. `#chmod a+x 335XLBVxxxx_yyyy-mm-dd.bin`
7. `#!/335XLBVxxxx_yyyy-mm-dd.bin`
8. Input "yes"
9. Then you can see folder "335XLBVxxxx_yyyy-mm-dd" on `/root/`.

Note! *xxxx is the version number, yyyy is the year, mm is month, dd is the day.
For example: 335XLBV1010_2015-03-01.*



Advantech offer you a script to setup the developing environment quickly. You can refer following steps to setup your developing environment:

1. Open "Terminal" on Ubuntu 12.04 LTS
2. `$sudo su` (Change to "root" authority)
3. Input user password
4. `#cd /root/335XLBVxxxx_yyyy-mm-dd/scripts/`
5. `#. setenv.sh` (To configure the developing environment automatically)
6. Then you can start to code the source code, build images, or compile applications.

This script is used to configure the developing environment quickly. It will configure the folder paths for system, and you can also add/modify the `setenv.sh` by yourself if you have added/changed the folders and paths.

Note! You have to run “#source setenv.sh” every time once you open a new “Terminal” utility.



Note! It is suggested to change to “root” authority to use the source code.



3.15.4 Build Instructions

This section will guide you how to build the u-boot & Linux kernel.

3.15.4.1 Build u-boot Image

Advantech has written a script to build the u-boot quickly. You can build u-boot image by follow below steps:

1. Open "Terminal" on Ubuntu 12.04 LTS..
2. \$sudo su (Change to “root” authority)
3. Input user password.
4. Change directory to BSP's scripts folder
5. #cd /root/335XLBVxxxx_yyyy-mm-dd/scripts/
6. #. setenv.sh (To configure the developing environment automatically)
7. #./cfg_uboot.sh am335x_rom3310
(To set the u-boot configuration automatically)
8. #./mk_uboot.sh (Start to build the u-boot)

Then you can see u-boot.img is being built and located in ./image

3.15.4.2 Build Linux Kernel Image

Advantech offer you a script to build the “ulmage” quickly. You can build ulmage by follow below steps:

1. Open "Terminal" on Ubuntu 12.04 LTS.
2. \$sudo su (Change to “root” authority)
3. Input user password.
4. Change directory to BSP's scripts folder
5. #cd /root/335XLBVxxxx_yyyy-mm-dd/scripts/
6. #. setenv.sh (To configure the developing environment automatically)
7. #./cfg_kernel.sh am335x_rom3310_defconfig (To set the ulmage configuration automatically)
8. #./mk_kernel.sh (Start to build the ulmage)

Then you can see ulmage is being built and located in ./image.

3.15.4.3 Build Log

You can find the build log from folder “./335XLBVxxxx_yyyy-mm-dd/”. If you got any error message when building Linux kernel, it is suggested to look into the log file to learn more detail about it

3.15.5 Source Code Modification

This section will guide you how to use the Linux source code. You will see some examples of using BSP source code in this section.

3.15.5.1 Add a Driver to Kernel by menuconfig

You can add a driver to kernel by menuconfig. Here is an example to guide you how to add a RTC driver (Seiko Instruments S-35390A) to Linux kernel. Please refer to the following steps:

1. Open "Terminal" on Ubuntu 12.04 LTS.
2. \$sudo su (Change to "root" authority)
3. Input user password.
4. Change directory to BSP's scripts folder
5. #. setenv.sh (To configure the developing environment automatically)
6. #./cfg_kernel.sh menuconfig

Then you will see a GUI screen (Linux Kernel Configuration) as below:

```

.config - Linux/arm 3.0.35 Kernel Configuration
-----
Linux/arm 3.0.35 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
-----
[*] Patch physical to virtual translations at runtime (EXPERIMENTAL)
[*] General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
-----
v[+]
-----
<Select> < Exit > < Help >

```

Figure 3.1 Linux Kernel Configuration

Select “Device Drivers” → ”Real Time Clock”, you will see an option “Seiko Instruments S-35390A” on the list. Choose this option then exit and save your configuration.

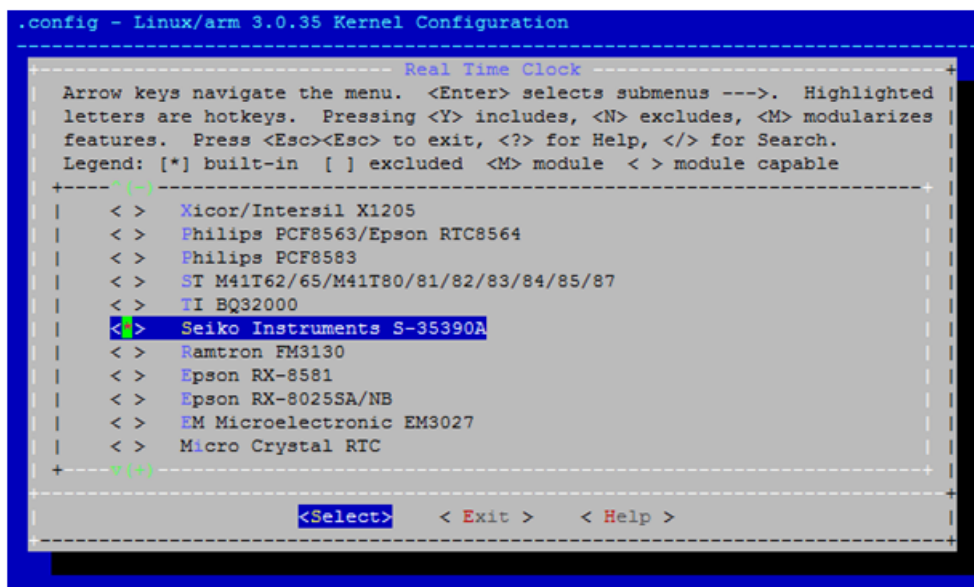


Figure 3.2 Selecting Seiko Instruments S-35390A

Change directory to “source/linux-3.2.0-psp04.06.00.11/arch/arm/mach-omap2”, edit the “board-rom3310.h” and “board-advantech.c”.

Please add below codes

to source/linux-3.2.0-psp04.06.00.11/arch/arm/mach-omap2/board-rom331.h:

```
static struct i2c_board_info mxc_i2c0_board_info[] __initdata = {
    {
        I2C_BOARD_INFO("s35390a", 0x30),
    },
};
```

Please add below codes to

source/ linux-3.2.0-psp04.06.00.11/arch/arm/mach- omap2/board-advantech.c

```
omap_register_i2c_bus(1, am335x_i2c0_boardinfo ,
    ARRAY_SIZE( am335x_i2c0_boardinfo ));
```

Please refer to former Chapter 3.15.4.2 to rebuild the kernel with RTC driver (Seiko Instruments S-35390A) after completing above steps.

Note! *If you cannot find the driver for your device from the list, please contact your hardware vender.*



3.15.6 Create a Linux System Boot Media

ROM-3310 supports boot from SD card or onboard flash. This section will guide you how to build an image for ROM-3310 Linux system boot media.

The storages devices name as following:

Device	Name
SD caed	/dev/mmcblk0
eMMC	/dev/mmcblk1

3.15.6.1 Create a Linux System SD Card

From Source Code Package

You are able to find the pre-built image from Advantech website.

Please follow below steps to create a SD card for boot up.

1. Copy "3310LIVxxxx_yyyy-mm-dd.tar.gz" package to your /root/.
2. Open "Terminal" on Ubuntu 12.04 LTS.
3. \$sudo su (Change to "root" authority)
4. Input your password.
5. #cd /root/
6. #tar xzvf 3310LIVxxxx_yyyy-mm-dd.tar.gz (Unzip files)
7. Insert one SD card to your developing computer
8. Check the SD card location, like /dev/sdb
9. #cd ./3310LIVxxxx_yyyy-mm-dd/scripts
10. #./mkzd-linux.sh /dev/sdb
11. Type "y" (Start to copy files, wait until it shows [Done])

Then insert the Linux system SD card to ROM-3310, it will boot up with Linux environment.

Boot from Onboard Flash

If you've already had a Linux system SD card, you can refer following steps to copy the content to onboard flash and then boot from onboard flash. Advantech also provide you a script "mkinand-linux.sh" to speed up the process of installing system image to onboard flash.

Refer to Chapter 3.25.1 to make a Linux system SD card.

Insert this Linux system SD card to ROM-DB3900 and connect serial console.

1. On ROM-3310 platform, type #root (Login)
2. On ROM-3310 platform, type #cd /mk_inand
3. On ROM-3310 platform, type #./mkinand-linux.sh /dev/mmcblk1
4. On ROM-3310 platform, type "y" (Start to copy files, wait until it shows [Done]).
5. Power off and remove this SD card.

Then you can boot from onboard flash without SD card.

3.15.7 Debug Message

ROM-3310 can connect to a host PC (Linux or Windows) by using console cable and debug port adapter. In order to communicate with host PC, serial communication program such as HyperTerminal, Putty is must require. Below is the detail instruction of how to set up serial console, a “HyperTerminal” on a Windows host:

1. Connect ROM-3310 to your Windows PC by using 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.

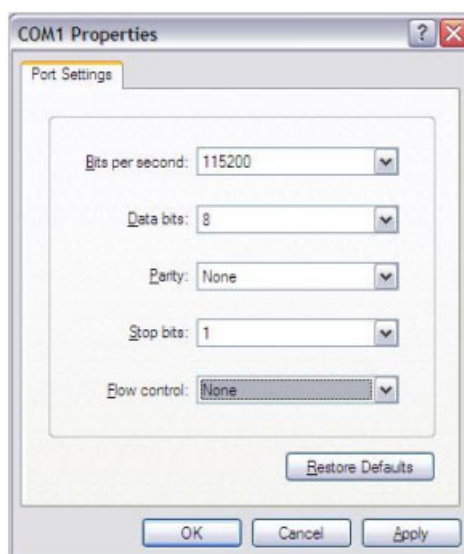


Figure 3.3 HyperTerminal Settings for Serial Console Setup

3.15.8 Linux System Configuration and Use

3.15.8.1 Display Output Setting

VGA Setting

Please set environment in u-boot as below:

```
setenv mmcargs "run bootargs_defaults;setenv console=${console}
bootargs ${bootargs} root=${mmccroot} rootfstype=${mmccrootfstype}
ip=${ip_method} video_mode= VGA_1024x768@60"
```

LDB-XGA is an example for the resolution of your LCD control. You can input the actual resolution of your VGA screen here, such as 800x600, 1024x768, etc. The system will accomplish the corresponding parameters automatically.

If the panel has problem to be activated, you may need to check the panel datasheet to configure the panel related parameters. The VGA video mode database is stored in linux-3.2.0/drivers/video/da8xx-fb.c. You can add a new one for your support resolution.

```
static struct da8xx_panel known_lcd_panels[] = {
    {
```

```

.name = " INNOLUX_VGA_1024x768@60",
..width = 1024,
..height = 768,
..hfp = 32,
..hbp = 228,
..hsw = 60,
..vfp = 7,
..vbp = 21,
..vsw = 10,
..pxl_clk = 65000000,
..invert_pxl_clk = 1,
.},
.}

```

The definition of da8xx_panel in linux-3.2.0/ drivers/video/da8xx-fb.c:

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 $1012 / (\text{Total horizontal line} * \text{Total vertical line} * \text{DCLK})$. 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 $1012 / (1344 * 806 * 60) = 15385$.

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 da8xx_panel {
const charname[25];
/* Full name <vendor>_<model> */
unsigned shortwidth;
                unsigned shortheight;
                int      hfp;      /* Horizontal front porch
*/
                int      hbp;      /* Horizontal back porch
*/
                int      hsw;      /* Horizontal Sync Pulse
Width */
                int      vfp;      /* Vertical front porch */
                int      vbp;      /* Vertical back porch */
                int      vsw;      /* Vertical Sync Pulse
Width */
                unsigned intpxl_clk; /* Pixel clock */
                unsigned charinvert_pxl_clk; /* Invert Pixel
clock */
};

```

Display Setting

VGA (Single) out, please set in u-boot as below:

1024x768@60 resolution:

```
setenv mmcargs "run bootargs_defaults;setenv console=${console} bootargs ${bootargs} root=${mmccroot} rootfstype=${mmc-rootfstype} ip=${ip_method} video_mode= VGA_1024x768@60"
```

800x600@40 resolution:

```
setenv mmcargs "run bootargs_defaults;setenv console=${console} bootargs ${bootargs} root=${mmccroot} rootfstype=${mmc-rootfstype} ip=${ip_method} video_mode= VGA_800x600@60"
```

1360x768@60 resolution:

```
setenv mmcargs "run bootargs_defaults;setenv console=${console} bootargs ${bootargs} root=${mmccroot} rootfstype=${mmc-rootfstype} ip=${ip_method} video_mode= VGA_1360x768@60"
```


3.15.8.2 Service Configuration

ROM-3310 has built five common network services: tftp service, ftp service, ssh service, telnet service and http service.

ftp Server

When boot up the ROM-3310, the tftp service is already started by default and the tftp server's working directory is /tftpboot. You need execute "chmod 777 /tftpboot" on ROM-3310 to let the tftp server work. Then, user can tftp to ROM-3310 by tftp client in host PC. Use command to get and put file like this:

```
hostPC$ tftp TARGET_SYSTEM_IP
tftp>get file1
tftp>put file2
```

Note!  Command "get file1" is to download file1 from tftp server. File "file1" must exist under the directory /tftpboot on ROM-3310; Command "put file2" is to upload file2 to tftp server. If put file2 success, file2 will be put to directory /tftpboot on ROM-3310.

The service start command is:

```
root@am335x-adv:/ # /etc/init.d/tftpd start
```

And the stop is:

```
root@am335x-adv:/ # /etc/init.d/tftpd stop
```


ftp Server

The ftp server on ROM-3310 is vsftpd and you should manually start it using following command.

```
root@am335x-adv:/ # /etc/init.d/vsftpd start
```

While, the stoping command is:

```
root@am335x-adv:/ # /etc/init.d/vsftpd stop
```

Note! After start the ftp server. You had to manually add user ftp:



```
root@am335x-adv:/ # adduser ftp
root@am335x-adv:/ # chown root:root /home/ftp/
```

Then you can ftp the ROM-3310 using user ftp.

ssh Server

When boot up the ROM-3310, the ssh service is already started by default. You can run the following command on your host PC to login the ROM-3310:

```
hostPC$ sudo ssh -l root TARGET_SYSTEM_IP
```

The service start command is:

```
root@am335x-adv:/ # /etc/init.d/dropbear start
```

And the stop is:

```
root@am335x-adv:/ # /etc/init.d/dropbear stop
```

telnet Server

When boot up the ROM-3310, the telnet service is already started by default. You can run the following command on your host PC to login the ROM-3310:

```
hostPC$ sudo telnet TARGET_SYSTEM_IP
```

The service start command is:

```
root@am335x-adv:/ # /etc/init.d/telnetd start
```

And the stop is:

```
root@am335x-adv:/ # /etc/init.d/telnetd stop
```

http Server

We support an embedded web server name lighttpd and the matrix GUI is based on it.

The service start command is:

```
root@am335x-adv:/ # /etc/init.d/lighttpd start
```

And the stop is:

```
root@am335x-adv:/ # /etc/init.d/lighttpd stop
```

3.15.8.3 Network configuration

Configuration via UI

You can get an IP address via DHCP, also you can configure a static IP address for ROM-3310. Click on the "Network Cfg" icon on the screen. Then Advantech NIC Configuration utility will be started. You can do some configuration of NIC.

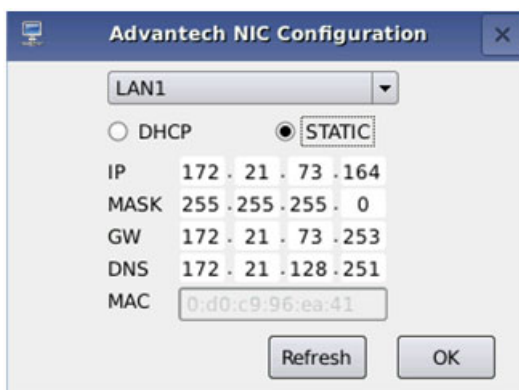


Figure 3.4 IP Configuration

Configuration via UI

As a choice, you can also do the network configuration by console using telnet. Run the following command on ROM-3310:

Get IP by DHCP:

```
root@am335x-adv:/ # /etc/init.d/dhcpc eth0 start
```

If you want to reserve the setting after rebooting the device, set as below:

```
root@am335x-adv:/ # echo "/etc/init.d/dhcpc eth0 start" > /etc/adv.d/netcfg.eth0
```

Set static IP:

Stop the DHCP process

```
root@am335x-adv:/ # /etc/init.d/dhcpc eth0 stop
```

Set the static IP as below.

```
root@am335x-adv:# /sbin/ifconfig eth0 172.21.73.191 netmask
255.255.255.0
root@am335x-adv:# /sbin/route add default gw 172.21.73.253 eth0
root@am335x-adv:# echo 'nameserver 172.21.128.251' >> /etc/
resolv.conf
root@am335x-adv:# /sbin/ifconfig eth1 192.168.3.102 netmask
255.255.255.0
root@am335x-adv:# /sbin/route add default gw ... eth1
root@am335x-adv:# echo 'nameserver 172.21.128.251' >> /etc/
resolv.conf
```

If you want to reserve the setting after rebooting the device, set as below:

```
root@am335x-adv:/ # echo "/sbin/ifconfig eth0 172.21.73.191 netmask
255.255.255.0; /sbin/route add default gw 172.21.73.253 eth0;echo
'nameserver 172.21.128.251' > /etc/resolv.conf;" > /etc/adv.d/
netcfg.eth0
```

Note! *The IP address in above should be replaced according to user's the requirement.*



For examples: IP 172.21.128.251 in above is the DNS server's IP, user should replace it with the correct DNS IP address.

3.15.8.4 Date/Time Configuration

You can use the tool we provide to modify the system time.

Click on the "Time Settings" icon on the screen. Then Advantech Date/Time Settings utility will be started.

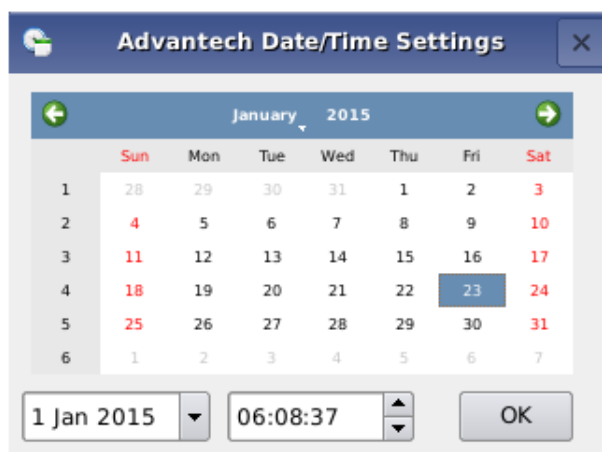


Figure 3.5 Date/Time Settings

After the time is adjusted, please click "OK" button, and the date will be saved. Meanwhile, the RTC time will be synchronized to the time you just set.

3.15.8.5 Serial Tools

We have five serial ports, named ttyO1~ttyO4. And we provide a serial test tool to let it easily to validate the serial ports.



Figure 3.6 Serial Control

3.15.8.6 Matrix GUI User's Guide

Overview

When you boot up the target system, Matrix GUI should be automatically started. Matrix is an HTML 5 based application launcher created to highlight available applications and demos provided. There are two forms of Matrix, local and remote Matrix. All of the example applications and demos are available using either the local or remote version. Matrix comes as a 6x4 matrix of icons or as a 4x3 matrix depending on the display resolution.

The launcher for Matrix is just a simple QT application that displays a Webkit base browser that points to the URL <http://localhost:80>.



Figure 3.7 Matrix

Launching and Stopping Matrix

If the Matrix GUI does not start with the system, you can manually start the program by the following command:

```
root@am335x-adv:# /etc/init.d/matrix-gui-2.0 start
```

If you want the Matrix to start with the system by default, please run the following command on ROM-3310:

```
root@am335x-adv:# cp /etc/init.d/matrix-gui-2.0 /etc/rc5.d/S97matrix-gui-2.0
```

When you want to cancel the default startup, just remove the S97matrix-gui-2.0 file. For more information on the use of Matrix, please refer to the following website: http://processors.wiki.ti.com/index.php/Matrix_Users_Guide

Adding a Matrix Application

Below are step-by-step instructions for Adding a New Application/Directory to Matrix.

1. Create a new folder on your target file system at `/usr/share/matrix-gui-2.0/apps/`. The name should be a somewhat descriptive representation of the application or directory. The folder name must be different from any existing folders at that location.
2. Create a desktop file based on the parameters discussed below. It is recommended the name of the desktop file match the name of the newly created folder. No white spaces can be used for the desktop filename. The desktop file parameters should be set depending on if you want to add a new application or a new directory to Matrix. The Type field must be set according to your decision. The desktop file must have the desktop suffix.
3. Update the Icon field in the desktop to refer to any existing Icon in the `/usr/share/matrix-gui-2.0` directory or subdirectories. You can also add a new 96x96 png image and place it into your newly created folder.
4. Optionally for applications you can add a HTML file that contains the application description in your newly created directory. If you add a description page then update the X-Matrix-Description field in the .desktop file.
5. Refresh Matrix using the application "Refresh Matrix" located in the Settings submenu.

Blank template icons for Matrix can be found here:

http://gforge.ti.com/gf/download/frsrelease/712/5167/blank_icons_1.1.tar.gz

The .desktop file is based on standard specified at the URL:

<http://standards.freedesktop.org/desktop-entry-spec/latest/>

Additional fields were added that are unique for Matrix.

Format for each parameter:

<Field>=<Value>

The fields and values are case sensitive.

3.15.8.7 Screen rotation for Qt application

Please export the Environments:

export QWS_DISPLAY=Transformed:Rot90 or

run directly run :

```
./example -qws -display "Transformed:Rot270"
```

3.15.8.8 Add a Startup items when boot

Remove a Startup items:

1. update-rc.d [-n] [-f] [-r <root>] <basename> remove
basename is your service script name

eg. update-rc.d -f matrix-gui-2.0 remove

2. Add a Startup items:

Firstly, You must ensure that the service script is exists, then run the flowing command:

```
update-rc.d [-n] [-r <root>] [-s] <basename> start|stop NN runlvl [runlvl] [...]
```

.start|stop : when system start /shutdown the basename will run automatically

NN: 0~99 runlvl: ROM3310 run level is 5(default);

eg. update-rc.d networking start 40 5 .

Then you can find the S40networking in rc5.d directory;

3.15.8.9 Package online install

OPKG Package Manager

Opkg is a lightweight package management system. It is written in C and resembles apt/dpkg/yum in operation. It is intended for use on embedded Linux devices and is used in this capacity in the OpenEmbedded and OpenWrt projects.

Advantech Embedded Linux for ROM-3310 has built-in OPKG package manager, with this tool you can install most of the required software online, and manage them, such as uninstall, upgrades and so on.

Installation New Software package

If you want to install a software which is not exist in the current OS, you should follow the steps below:

1. Update the online software source:

```
root@am335x-adv:# opkg update
```

2. Search whether the software source server has the software you need.

```
root@am335x-adv:# opkg list | grep package
```

Note! *Package is the keywords of the software name, for example, you want to search an ftp server, and the package should be 'ftp'.*



3. Find the full name of the software you need in the search result list. And install it by following command:

```
root@am335x-adv:# opkg install packagename
```

More about OPKG

More about use and development of OPKG, Please refer to the project website of OPKG:

<https://code.google.com/p/opkg/>

3.15.9 Development Guide and Reference

3.15.9.1 Development of C/C++ Programs

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 and Change to “root”:

```
hostPC$ sudo
```

Input user password.

2. Create the develop environment using flowing command:

```
#source /usr/local/cross_compiler/linux-devkit/environment-setup
# cd /root//3310LBVxxxx_yyyy_mm_dd/source
# mkdir helloworld
# cd helloworld
# gedit hellowrold.c
```

3. Edit the helloworld.c with the following source code:

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

Save the file and exit.

4. Compile helloworld.c using flowing command:

```
#$CC $CFLAGS -o helloworld helloworld.c
```

Then you can see “helloworld” in current directory.

5. Run the executable file helloworld on the ROM-3310.
Insert the Linux system SD card to your developing computer.

```
# cp helloworld /media/rootfs/tool
```

Note! */media/rootfs is the mounted point of your Linux system SD card
Remove this SD card and insert it to ROM-3310, then open serial console.*



On ROM-3310 platform, type #root (Login)

On ROM-3310 platform, type #cd /tool

On ROM-3310 platform, type #./helloworld

Now you should be able to see “Hello World!” shown on ROM-3310.

3.15.9.2 Development of GUI Programs with QT Library

With the development kit, you can develop a qt-based GUI program. Follow these steps, you can quickly convert your QT Project to a GUI application for ROM-3310:

On your host PC, set up QT Build Environment.

```
#source /usr/local/cross_compiler/linux-devkit/environment-setup
```

Build QT Instructions:

```
# cd projectdir
# qmake projectName.pro
# make
```

Run QT demo:

```
# ./qtappName -qws
```

Note! The *-qws* Parameter tell the QT Application to run as a server.



3.15.9.3 Demo program source code

Serial Port Programming

Please refer to <BSP_PATH>/source/demo/uart

It is an example of sending and receiving data via the serial port.

Receiving data:

```
# ./uart_ctrl read /dev/tty01
```

Sending data:

```
# ./uart_ctrl send /dev/tty01
```

Before using your program of serial port, please ensure that your serial port is in 232/422/485 mode.

User can reference the uart demo source code to develop the uart application.

Watchdog Programming

ROM-3310 support hardware watchdog, the watchdog API is follow posix standards. The valid timeout value is from 1 to 6553 seconds, if the timeout value to set is not in this scope, driver will set timeout value to default value (60 seconds).

Sample C code:

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <signal.h>
#include <linux/watchdog.h>
```



```

int fd;
int main(int argc, const char *argv[]) {
int timeout = 10;

/*open watchdog device, the watchdog device node is /dev/watchdog */
fd=open("/dev/watchdog",O_WRONLY);

/*set timeout to 10 seconds*/
/*when set timeout value, the watchdog driver will enable the
watchdog automatically.*/
ioctl (fd, WDIOC_SETTIMEOUT, &timeout);
while (1){
/*feed the watchdog every 5 seconds*/
/*when call this funtion to feed watchdog, the watchdog will
reset its internal timer so it doesn't trigger the board reset.
If do not feed the watchdog more than 10 seconds, the watchdog
will timeout and the board will reset.*/
ioctl( fd, WDIOC_KEEPAALIVE, NULL );
sleep (5);
}
close (fd);
}

```

Here are some other APIs for watchdog.

Disable the watchdog timer sample code:

```

/*if user want to disable the watchdog before timeout,
call the following ioctl function*/
int i_dis = WDIOS_DISABLECARD;
ioctl( fd, WDIOC_SETOPTIONS, &i_dis );

```

Enable the watchdog timer sample code:

```

/*if user want to enable the watchdog again before timeout when
it is disabled, call the following ioctl function. */
int i_en = WDIOS_ENABLECARD;
ioctl( fd, WDIOC_SETOPTIONS, & i_en);

```

Get the current timeout value:

```

/*get the current timeout value the driver used*/
int timeout = 0;
ioctl (fd, WDIOC_GETTIMEOUT, &timeout);

```

Please refer to <BSP_PATH>/source/demo/watchdog folder to get more information.

GPIO Programming

ROM-3310 has 8 gpios. Please refer to <BSP_PATH>/source/demo/gpio Usage:

```
root@am335x-adv:# ./gpio 200 out 1
```

Note! *“200” means gpio0, and so 200-209 corresponds to gpio0-gpio9 “out” means output.*



“1” is the value set to the corresponding gpio port.

Can Programming

Please refer to <BSP_PATH>/source/demo/can_test.

Note! *Can sending data sample c code, please refer to can_write.c.*



Can receiving data sample c code, please refer to can_read.c.

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. You can do system recovery through below steps if you destroy onboard flash image by accident.

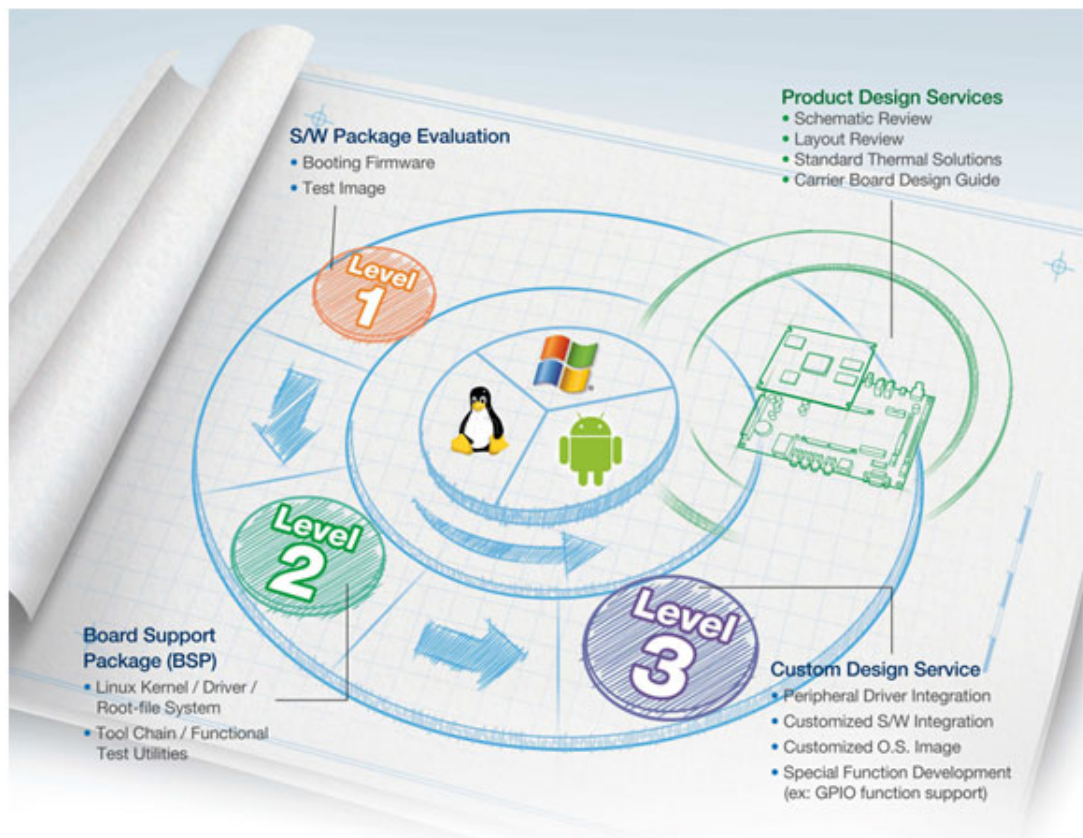
1. Copy "3310LIVxxxx_yyyy-mm-dd.tar.gz" package to your /root/.
2. Open "Terminal" on Ubuntu 12.04 LTS.
3. \$sudo su (Change to "root" authority).
4. Input your password.
5. #cd root/.
6. #tar xvf 3310LIVxxxx_yyyy-mm-dd.tar.gz (Unzip files).
7. Insert one SD card to your developing computer.
8. Check the SD card location, like /dev/sdb.
9. Please wait until dump disk is done.
10. On ROM-3310 platform, type #cd /mk_inand.
11. On ROM-3310 platform, type #./mkinand-linux.sh /dev/mmcblk1.
12. On ROM-3310 platform, type "y".
(Start to copy files, wait until it shows [Done])
13. Power off and remove this SD card.

Chapter 5

Advantech Services

This chapter introduces Advantech design in serviceability, technical support and warranty policy for ROM-5420 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 check list for engineer for easy 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. It helps to save a lot of time and costs with regard to developing 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 thermal solution services 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, SIs 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 SIs 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 professional 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 Freescale, 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 of 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

On the other hand, you can reach our service team through below website, 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 website 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

5.4 Global Service Policy

5.4.1 Warranty Policy

Below is the warranty policy of Advantech products:

5.4.2 Warranty Period

Advantech branded off-the-shelf products and 3rd party off-the-shelf products used to assemble Advantech Configure to Order products are entitled to a 2 years complete and prompt global warranty service. Product defect in design, materials, and workmanship, are covered from the date of shipment.

All customized products will by default carry a 15 months regional warranty service. The actual product warranty terms and conditions may vary based on sales contract.

All 3rd party products purchased separately will be covered by the original manufacturer's warranty and time period, and shall not exceed one year of coverage through Advantech.

5.4.3 Repairs under Warranty

It is possible to obtain a replacement (Cross-Shipment) during the first 30 days of the purchase, thru your original ADVANTECH supplier to arrange DOA replacement if the products were purchased directly from ADVANTECH and the product is DOA (Dead-on-Arrival). The DOA Cross-Shipment excludes any shipping damage, customized and/or build-to-order products.

For those products which are not DOA, the return fee to an authorized ADVANTECH repair facility will be at the customers' expense. The shipping fee for reconstructive products from ADVANTECH back to customers' sites will be at ADVANTECH's expense.

5.4.4 Exclusions from Warranty

The product is excluded from warranty if

- The product has been found to be defective after expiry of the warranty period.
- Warranty has been voided by removal or alternation of product or part identification labels.
- The product has been misused, abused, or subjected to unauthorized disassembly/modification; placed in an unsuitable physical or operating environment; improperly maintained by the customer; or failure caused which ADVANTECH is not responsible whether by accident or other cause. Such conditions will be determined by ADVANTECH at its sole unfettered discretion.
- The product is damaged beyond repair due to a natural disaster such as a lightning strike, flood, earthquake, etc.
- Product updates/upgrades and tests upon the request of customers who are without warranty.

5.5 Repair Process

5.5.1 Obtaining an RMA Number

All returns from customers must be authorized with an ADVANTECH RMA (Return Merchandise Authorization) number. Any returns of defective units or parts without valid RMA numbers will not be accepted; they will be returned to the customer at the customer's cost without prior notice.

An RMA number is only an authorization for returning a product; it is not an approval for repair or replacement. When requesting an RMA number, please access ADVANTECH's RMA web site: <http://erma.ADVANTECH.com.tw> with an authorized user ID and password.

You must fill out basic product and customer information and describe the problems encountered in detail in "Problem Description". Vague entries such as "does not work" and "failure" are not acceptable.

If you are uncertain about the cause of the problem, please contact ADVANTECH's Application Engineers (AE). They may be able to find a solution that does not require sending the product for repair.

The serial number of the whole set is required if only a key defective part is returned for repair. Otherwise, the case will be regarded as out-of-warranty.

5.5.2 Returning the Product for Repair

It's possible customers can save time and meet end-user requirements by returning defective products to any authorized ADVANTECH repair facility without an extra cross-region charge. It is required to contact the local repair center before offering global repair service.

It is recommended to send cards without accessories (manuals, cables, etc.). Remove any unnecessary components from the card, such as CPU, DRAM, and CF Card. If you send all these parts back (because you believe they may be part of the problem), please note clearly that they are included. Otherwise, ADVANTECH is not responsible for any items not listed. Make sure the "Problem Description" is enclosed.

European Customers that are located outside European Community are requested to use UPS as the forwarding company. We strongly recommend adding a packing list to all shipments. Please prepare a shipment invoice according to the following guidelines to decrease goods clearance time:

1. Give a low value to the product on the invoice, or additional charges will be levied by customs that will be borne by the sender.
2. Add information "Invoice for customs purposes only with no commercial value" on the shipment invoice.
3. Show RMA numbers, product serial numbers and warranty status on the shipment invoice.
4. Add information about Country of origin of goods

In addition, please attach an invoice with RMA number to the carton, then write the RMA number on the outside of the carton and attach the packing slip to save handling time. Please also address the parts directly to the Service Department and mark the package "Attn. RMA Service Department".

All products must be returned in properly packed ESD material or anti-static bags. ADVANTECH reserves the right to return unrepaired items at the customer's cost if inappropriately packed.

Besides that, "Door-to-Door" transportation such as speed post is recommended for delivery, otherwise, the sender should bear additional charges such as clearance fees if Air-Cargo is adopted.

Should DOA cases fail, ADVANTECH will take full responsibility for the product and transportation charges. If the items are not DOA, but fail within warranty, the sender will bear the freight charges. For out-of-warranty cases, customers must cover the cost and take care of both outward and inward transportation.

5.5.3 Service Charges

The product is excluded from warranty if:

- The product is repaired after expiry of the warranty period.
- The product is tested or calibrated after expiry of the warranty period, and a No Problem Found (NPF) result is obtained.
- The product, though repaired within the warranty period, has been misused, abused, or subjected to unauthorized disassembly/modification; placed in an unsuitable physical or operating environment; improperly maintained by the customer; or failure caused which ADVANTECH is not responsible whether by accident or other cause. Such conditions will be determined by ADVANTECH at its sole unfettered discretion.
- The product is damaged beyond repair due to a natural disaster such as a lightning strike, flood, earthquake, etc.
- Product updates and tests upon the request of customers who are without warranty.

If a product has been repaired by ADVANTECH, and within three months after such a repair the product requires another repair for the same problem, ADVANTECH will do this repair free of charge. However, such free repairs do not apply to products which have been misused, abused, or subjected to unauthorized disassembly/modification; placed in an unsuitable physical or operating environment; improperly maintained by the customer; or failure caused which ADVANTECH is not responsible whether by accident or other cause.

Please contact your nearest regional service center for detail service quotation.

Before we start out-of-warranty repairs, we will send you a pro forma invoice (P/I) with the repair charges. When you remit the funds, please reference the P/I number listed under "Our Ref". ADVANTECH reserves the right to deny repair services to customers that do not return the DOA unit or sign the P/I. Meanwhile, ADVANTECH will scrap defective products without prior notice if customers do not return the signed P/I within 3 months.

5.5.4 Repair Report

ADVANTECH returns each product with a “Repair Report” which shows the result of the repair. A “Repair Analysis Report” is also provided to customers upon request. If the defect is not caused by ADVANTECH design or manufacturing, customers will be charged US\$60 or US\$120 for in-warranty or out-of-warranty repair analysis reports respectively.

5.5.5 Custody of Products Submitted for Repair

ADVANTECH will retain custody of a product submitted for repair for one month while it is waiting for return of a signed P/I or payment (A/R). If the customer fails to respond within such period, ADVANTECH will close the case automatically. ADVANTECH will take reasonable measures to stay in proper contact with the customer during this one month period.

5.5.6 Shipping Back to Customer

The forwarding company for RMA returns from ADVANTECH to customers is selected by ADVANTECH. Per customer requirement, other express services can be adopted, such as UPS, FedEx and etc. The customer must bear the extra costs of such alternative shipment. If you require any special arrangements, please indicate this when shipping the product to us.

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. 2015