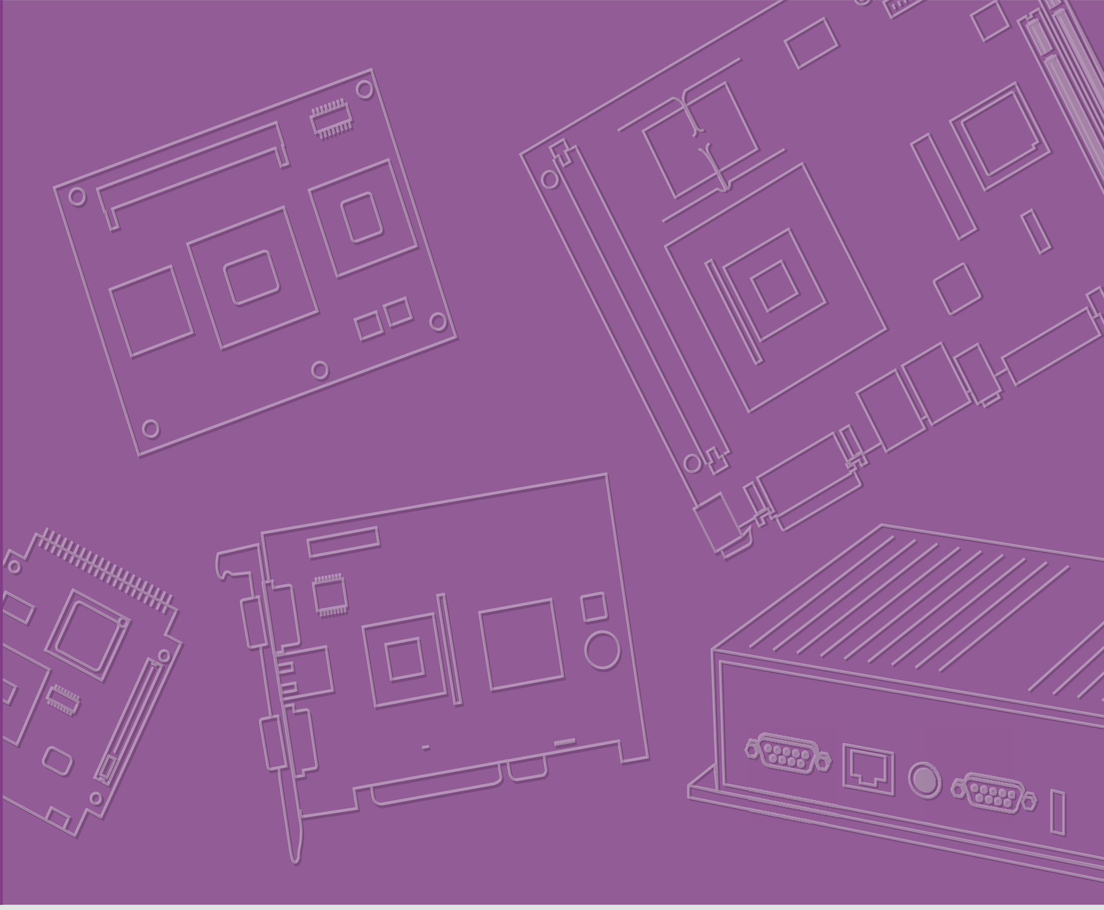


User Manual

UBC-220

RISC IoT Box Computer Powered by Freescale i.MX6 Dual Lite Processor ARM® Cortex™ A9 Architecture

Enabling an Intelligent Planet

Copyright

Acknowledgements

ARM is trademarks of ARM Corporation.

Freescale is trademarks of Freescale 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 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. 2006022001 Edition 2

Printed in Taiwan August 2016

Declaration of Conformity

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.

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

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 Part Number

- 1 x UBC-220 RISC Box Computer
- 1 x 6-pin terminal block connector

Ordering Information

Model Number	Description
UBC-220DL-MDA1E	Freescale i.MX6 Cortex-A9 Dual Lite Compact Box Computer

Optional Accessories

Model Number	Description
1960069533S001	Heatsink
96PSA-A36W12R1	Adaptor 100-240V 36W 12V 3A W/O PFC 9NA0361603
170203183C	Power cord 3P Europe (WS-010+WS-083) 183cm
170203180A	Power cord 3P UK 2.5A/3A 250V 1.83M
1700001524	Power cord 3P UL 10A 125V 180cm
96LEDK-V121SV45NE1	Panel TM121SDS01
1700021565-01	Debug Cable
1700022130-01	LVDS Cable
1700022131-01	Backlight cable

Note! *This product is intended to be supplied by a Listed Power Adapter or DC power source marked "L.P.S.", rated 12Vdc, 3A minimum, Tma = 40 degree C. If need further assistance, please contact Advantech for further information.*

Certification and Safety Instructions

This device complies with the requirements in part 15 of the FCC rules: Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation

This equipment has been tested and found to comply with the limits for a Class B 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 device in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his/her own expense. The user is advised that any equipment changes or modifications not expressly approved by the party responsible for compliance would void the compliance to FCC regulations and therefore, the user's authority to operate the equipment.

Caution! *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.

Contents

Chapter 1

General Introduction

This chapter gives background information on the UBC-220.

Sections include:

- **Introduction**
- **Product Features**
- **Specifications**

1.1 Introduction

UBC-220 is a compact box computer powered by Freescale i.MX6 Dual Lite high performance CPU, it is an ideal solution for IoT, automation and HMI applications. The User manual covers two parts: Hardware I/O specifications and software introduction. It is recommended that users completely read this document before starting UBC-220 evaluation since it contains critical information such as I/O pin definitions, handling processes and connector specifications.

Please comply with all warning notices and regulations described in this document to prevent damage to UBC-220 and yourself. Inappropriate handling may cause physical damage or abnormal behavior or unstable performance.

For more information of Advantech RISC products, please visit RISC Mini Site: http://risc.advantech.com.

1.2 Specifications

1.2.1 Functional Specifications

Processor: Freescale i.mx6 Series

- ARM Cortex™-A9 high performance processor, Dual Lite 1 GHz
- Supports 1 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 800 MHz
- **Capacity:** on board DDR3 1 GB

Gigabit Ethernet

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

Peripheral Interface

- 1 x Single channel 18/24 bit LVDS
- 1 x HDMI
- 1 x USB2.0 Type A and 1 USB 2.0 OTG
- 1 x SD Slot
- 1 x 4-wire UART terminal block connector
- 2 x miniPCIe slot
- 1 x SIM Slot

OS Support

UBC-220 supports Linux BSP 3.0.35

1.2.2 Mechanical Specifications

- **Dimension:** 100 x 72 mm (5.7”x4”)
- **Height:** 20.6 mm
- **Reference Weight:** 540 g (including whole package)

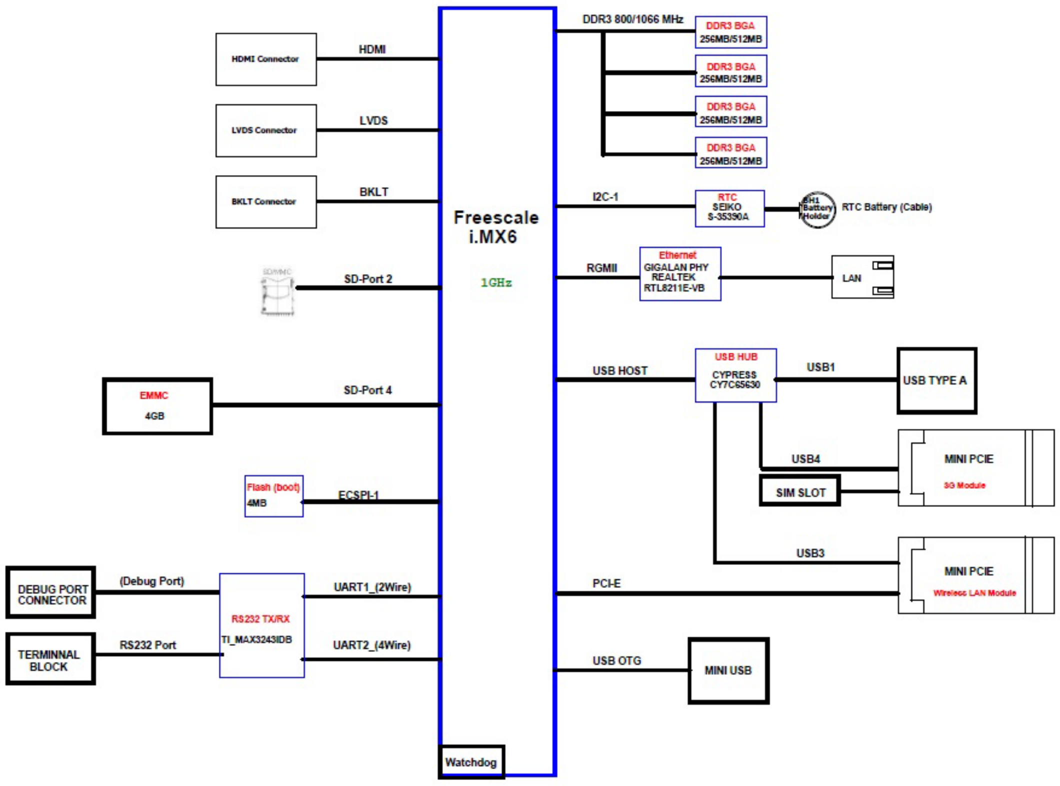
1.2.3 Electrical Specifications

- **Power supply type:** DC-in 12 V
- **Power consumption:**
  - Kernel Idle mode: 2.3 W
  - Max mode: 4.08 W
- **RTC Battery:**
  - Typical voltage: 3.0 V
  - Normal discharge capacity: 3 uA

1.3 Environmental Specifications

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

1.4 Block Diagram



Chapter 2

H/W Installation

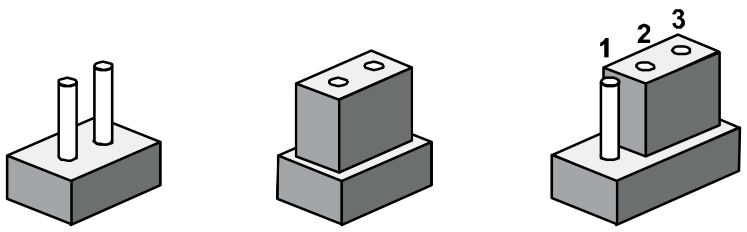
This chapter introduces the startup procedures of the UBC-220 hardware, including jumper setting and device integration. It also introduces the setting of switches, indicators and also shows the mechanical drawings.

Be sure to read all safety precautions before you begin installation procedure.

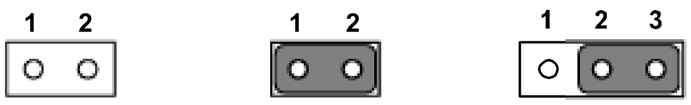
2.1 Jumpers

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.

Warning! *To avoid damaging the computer, always turn off the power supply before setting jumpers.*

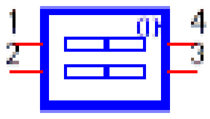
2.1.2 Jumper List

Table 2.1: Jumper List	
SW2	Boot Device
LVDS_VDD_SLT	LVDS Power
LVDS_BKLT_SLT	Backlight Power

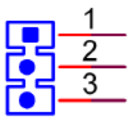
2.1.3 Jumper Settings

SW2	Boot device
Part number	1600000202
Footprint	SW_2x2P_50_161X315
Description	DIP SW CHS-02TB(29) SMD 4P SPST P=1.27mm W=5.4mm
Setting	Function
1 ON	Boot from SD
1 OFF	Boot from SPI

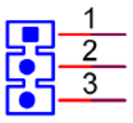
This switch is designed for selecting boot up method.



LVDS_VDD_SLT	LVDS Power
Part Number	1653003100
Footprint	HD_3x1P_100_D
Description	PIN HEADER 3x1P 2.54mm 180D(M) DIP 205-1x3GS
Setting	Function
(1-2)	+V3.3
(2-3)	+V5



LVDS_BKLT_SLT	LVDS Backlight Power
Part Number	1653003100
Footprint	HD_3x1P_100_D
Description	PIN HEADER 3x1P 2.54mm 180D(M) DIP 205-1x3GS
Setting	Function
(1-2)	+V5
(2-3)	+VIN (+12V)



2.2 Connectors

2.2.1 Connector List

CN1	RTC battery
MINI_PCIE_HALF	Half-size MiniPCIe
MINI_PCIE_FULL	Full-size Mini PCIe
SIM_SLOT	SIM socket
DEBUG_CONSOLE	UART1 debug port
USB_HOST	USB Type A Connector
USB_OTG	USB OTG Connector
LAN	Ethernet Connector
DCIN	DC power jack
HDMI	HDMI
SD	SD Card
COM	COM
LVDS	LVDS
LVDS_BKLT_PWR	Backlight

2.2.2 Connector Settings

2.2.2.1 RTC Battery Connector (CN1)

UBC-220 supports a lithium 3V/210mAH CR2032 battery with wire via battery connector.

2.2.2.2 Half-size MiniPCIe (MINI_PCIE_HALF)

UBC-220 supports half size miniPCIe slot both USB and PCIe interface.

Pin	Signal Name	Pin	Signal Name
1		2	3.3Vaux
3	Reserved	4	GND
5	Reserved	6	
7		8	
9	GND	10	
11	REFCLK-	12	
13	REFCLK+	14	
15	GND	16	
	Mechanical Key		
17		18	GND
19		20	W_DISABLE#
21	GND	22	PERST#
23	PERn0	24	3.3Vaux
25	PERp0	26	GND
27	GND	28	
29	GND	30	

31	PETn0	32	
33	PETp0	34	GND
35	GND	36	USB_D-
37	GND	38	USB_D+
39	3.3VAUX	40	GND
41	3.3VAUX	42	LED_WWAN#
43	GND	44	LED_WLAN#
45	Reserved	46	
47	Reserved	48	
49	Reserved	50	GND
51	Reserved	52	3.3VAUX

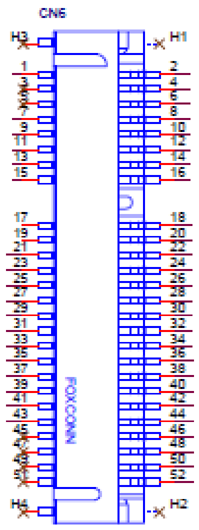


Figure 2.1 Half-size miniPCIE

2.2.2.3 Full-size MiniPCIe (MINI_PCIE_FULL))

RSB-3410 supports full size miniPCIe slot which supports USB interface only.

Table 2.2: Full-size miniPCIE

Pin	Description	Pin	Description
1	NA	27	GND
2	+3.3V	28	NA
3	Reserved	29	GND
4	GND	30	NA
5	Reserved	31	NA
6	NA	32	NA
7	NA	33	NA
8	UIM_PWR	34	GND
9	GND	35	GND
10	UIM_DATA	36	USB_DATA-
11	NA	37	Reserved
12	UIM_CLK	38	USB_DATA+
13	NA	39	Reserved
14	UIM_RESET	40	GND

Table 2.2: Full-size miniPCIE

15	GND	41	Reserved
16	UIM_VPP	42	LED_WWAN
17	Reserved	43	Reserved
18	GND	44	LED_WLAN
19	Reserved	45	Reserved
20	Reserved	46	LED_WPAN
21	GND	47	Reserved
22	PERST	48	NA
23	NA	49	Reserved
24	+3.3V	50	GND
25	NA	51	Reserved
26	GND	52	+3.3V

2.2.2.4 SIM Socket (SIM_SLOT)

UBC-220 supports on board SIM socket is for 3G integration. Please insert valid SIM card to dial to 3G network.

Pin	Signal Name	Pin	Signal Name
C1	UIM_PWR	C2	UIM_RESET
C3	UIM_CLK	C5	GND
C6		C7	UIM_DATA

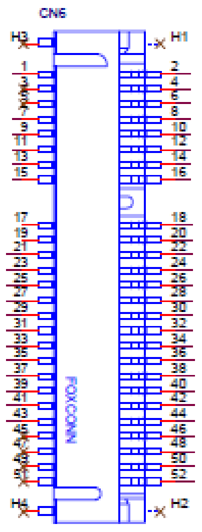


Figure 2.2 Full-size miniPCIE

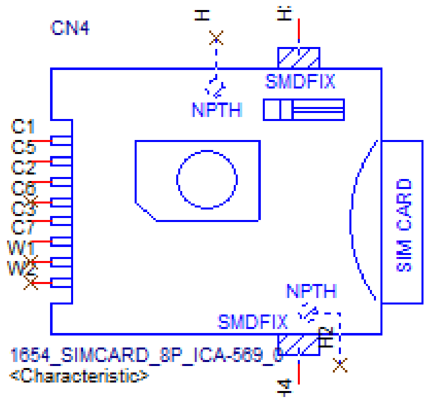


Figure 2.3 SIM Socket

2.2.2.5 UART1 Debug Port (DEBUG_CONSOLE)

UBC-220 can communicate with a host server (Windows or Linux) by using serial cables.

Pin	Description
1	+V3.3
2	DEBUG_TXD
3	DEBUG_RXD
4	GND

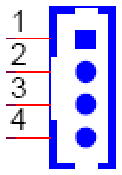


Figure 2.4 Debug Port

2.2.2.6 USB Type A Connector (USB_HOST)

UBC-220 supports one standard USB2.0 Type A connector in the coastline.

Pin	Description
1	+5V
2	USB Data-
3	USB Data+
4	GND

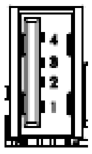


Figure 2.5 USB Type A Connector

2.2.2.7 Ethernet Connector (LAN)

UBC-220 provides one RJ45 LAN interface connector, it is fully compliant with IEEE 802.3u 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	MDI0+
2	MDI0-
3	MDI1+
4	MDI1-
5	GND
6	GND
7	MDI2+
8	MDI2-
9	MDI3+
10	MDI3-
11	VCC
12	ACT
13	Link100#
14	Link1000#

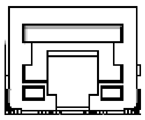


Figure 2.6 Ethernet Connector

2.2.2.8 DC power Jack (DCIN)

UBC-220 comes with a DC-Jack header that carries 12V DC external power input.

Pin	Description
1	DC_IN
2	GND

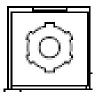


Figure 2.7 DC Power Jack

2.2.2.9 HDMI (HDMI)

UBC-220 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 resolutions 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_HPD
19	HDMI_HP

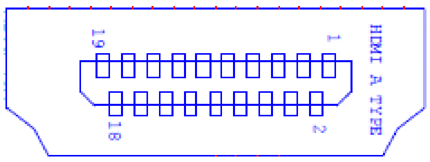


Figure 2.8 HDMI

2.2.2.10 USB OTG (USB_OTG)

UBC-220 supports USB-OTG mode that can be taken to replace UART debug console.

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

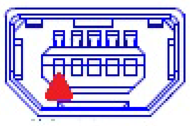


Figure 2.9 USB OTG

2.2.2.11 SD Slot

UBC-220 supports SD/MMC card in Class2, 4, 6, 8, 10. Supported capacity is up to 32G (SDHC)

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

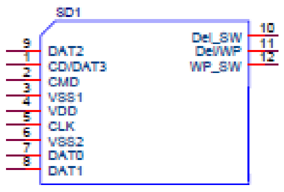


Figure 2.10 SD Slot

2.2.2.12 COM Port

UBC-220 provides one 6-pin terminal block connector as serial communication interface port. The port can support RS-232 mode communication.

Pin	Description
1	RX
2	RTS
3	TX
4	CTS
5	GND
6	N/C

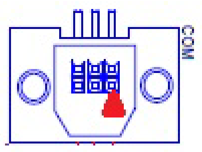


Figure 2.11 COM Port

2.2.2.13 LVDS Connector

UBC-220 provides a LVDS 10x2-pin board-to-board connector for single channel 18/24 bit LVDS panel up to 1366x768. Please also refer to jumper setting in page 7 before connecting LVDS panel.

Pin	Description
1	GND
2	GND
3	LVDS0_TX0_P
4	I2C1_SCL_LVDS0
5	LVDS0_TX0_N
6	I2C1_SDA_LVDS0
7	LVDS0_TX1_P
8	
9	LVDS0_TX1_N
10	
11	LVDS0_TX2_P
12	
13	LVDS0_TX2_N
14	
15	LVDS0_CLK_P
16	LVDS0_TX3_P
17	LVDS0_CLK_N
18	LVDS0_TX3_N
19	+VDD_LVDS
20	+VDD_LVDS

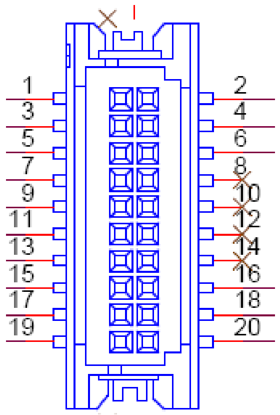


Figure 2.12 LVDS Connector

2.2.2.14 LVDS Backlight Connector

Please also refer to jumper setting in page 7 before connecting LVDS panel.

Pin	Description
1	+VDD_BKLT_LVDS
2	GND
3	LCD_BKLT_A
4	LCD_BKLT_PWM_A
5	+V5

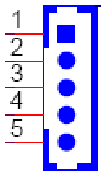


Figure 2.13 LVDS Backlight Connector

2.3 Mechanical

2.3.1 Jumper and Connector Locations

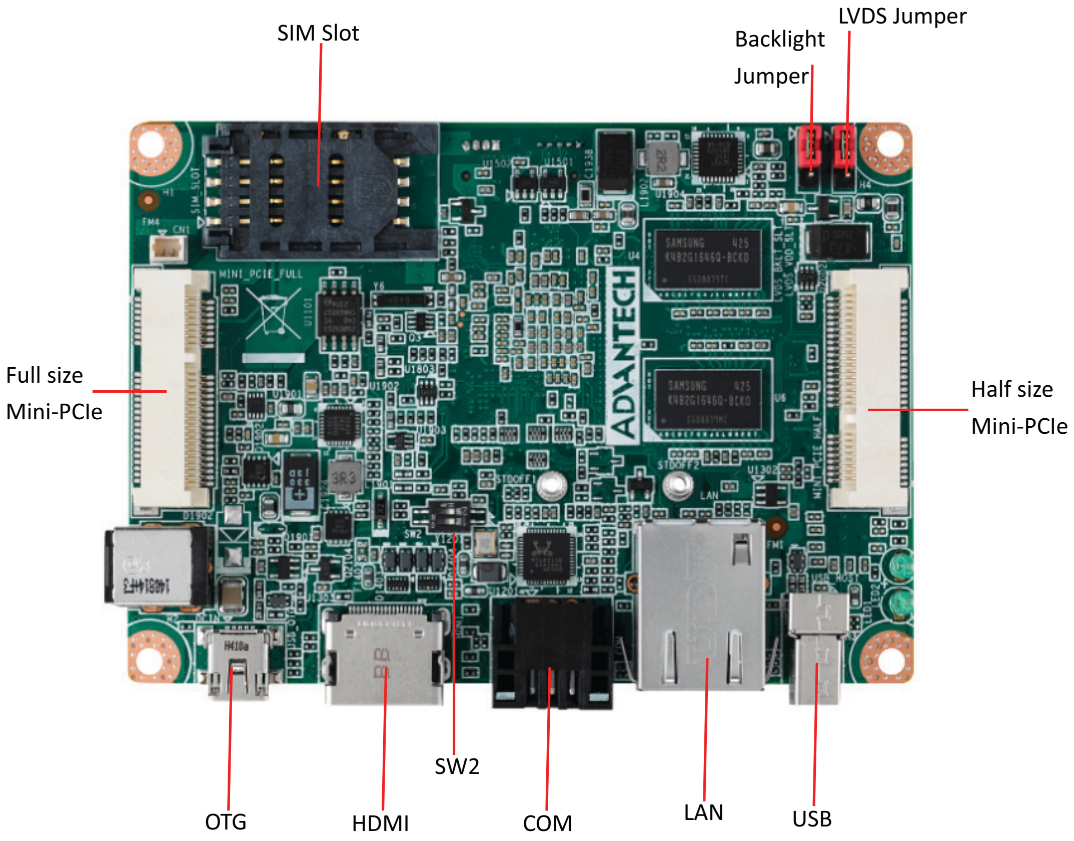


Figure 2.14 Jumper and Connector Layout (Top side)

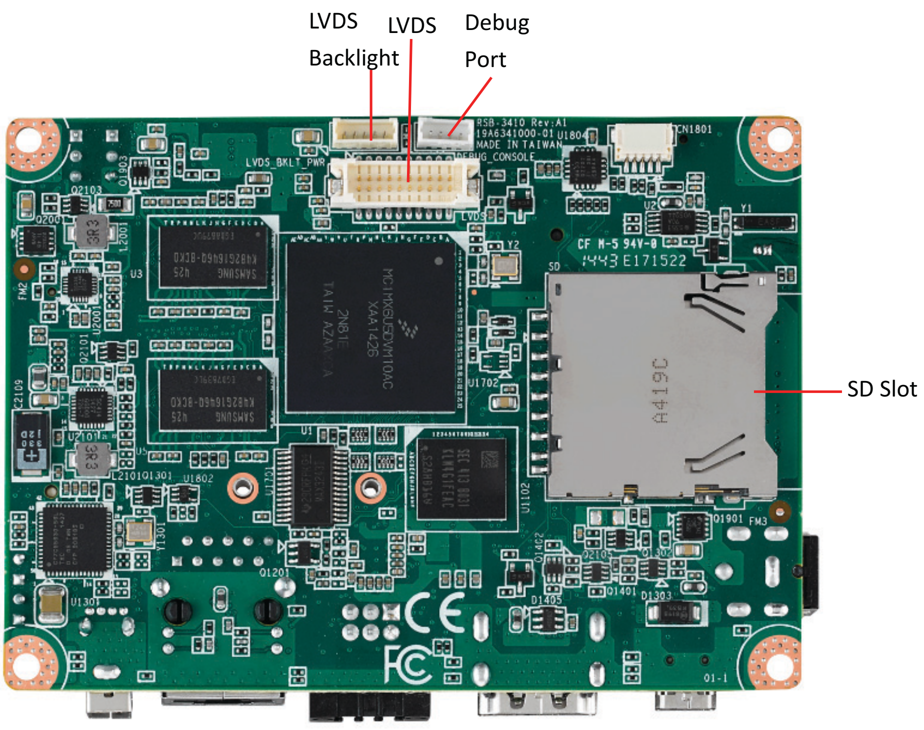


Figure 2.15 Jumper and Connector Layout (Bottom Side)

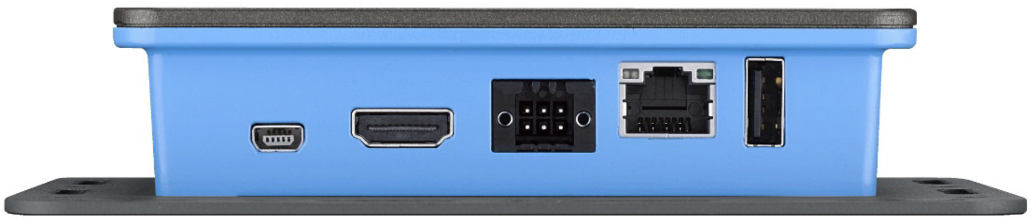
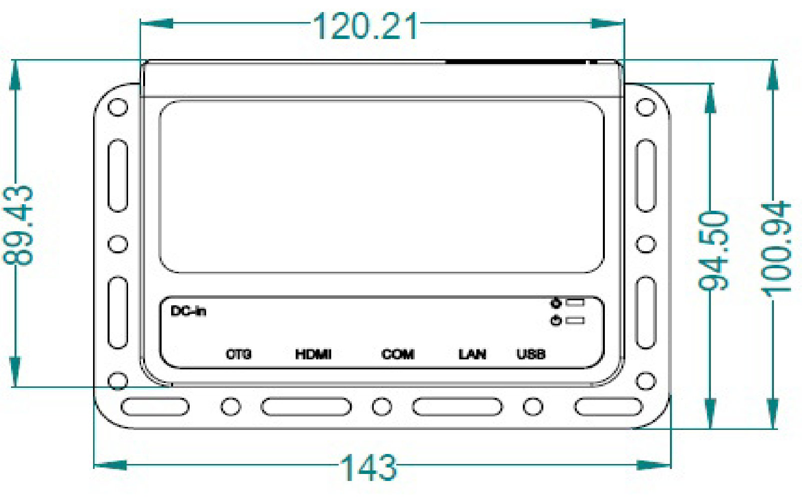


Figure 2.16 Coastline Layout

2.3.2 System Dimensions

2.3.2.1 System Drawing



unit: mm

Figure 2.17 System Dimension Layout (Top Side)

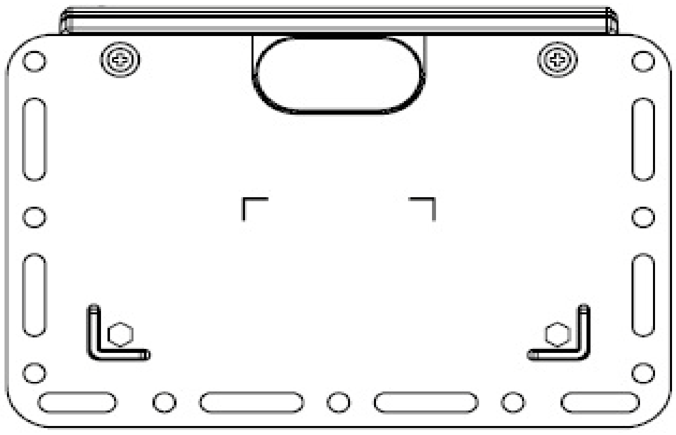


Figure 2.18 System Dimension Layout (Bottom Side)

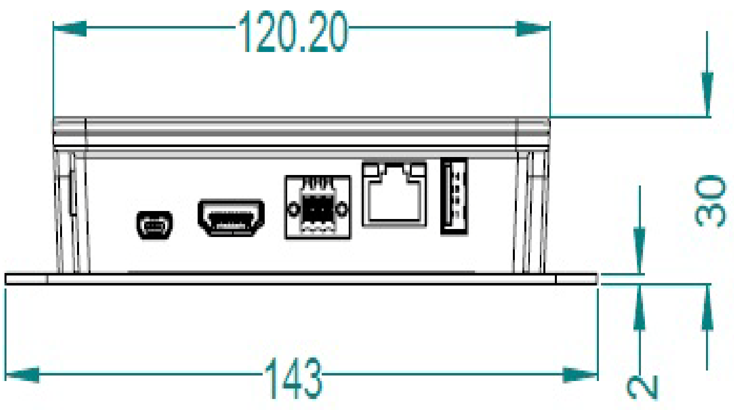


Figure 2.19 System Dimension Layout (Coastline)

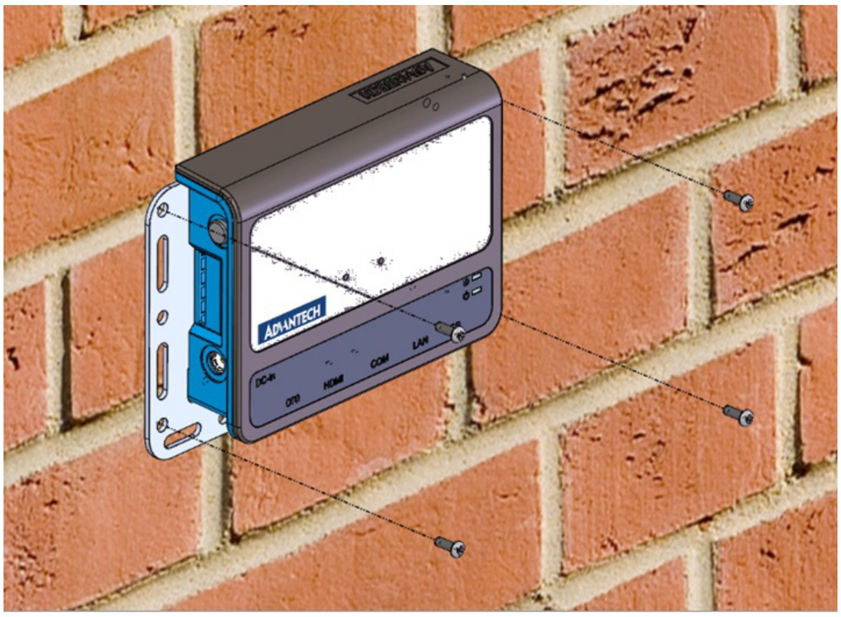


Figure 2.20 Mounting Method of UBC-220

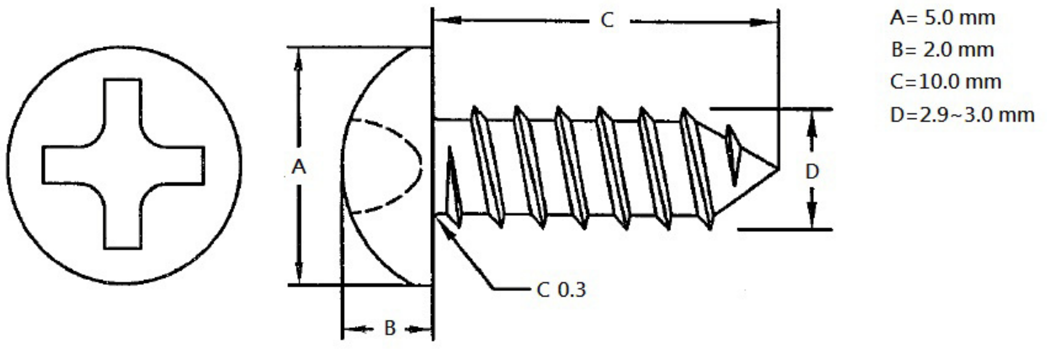


Figure 2.21 Drawing of mating screw

Part Number	Description	Quantity
1930000721	1/4R/S D=5.5 H=2.0 + T3*10 ST Zn	4

Figure 2.22 Mating screws for wall mounting

2.4 Quick Start of UBC-220

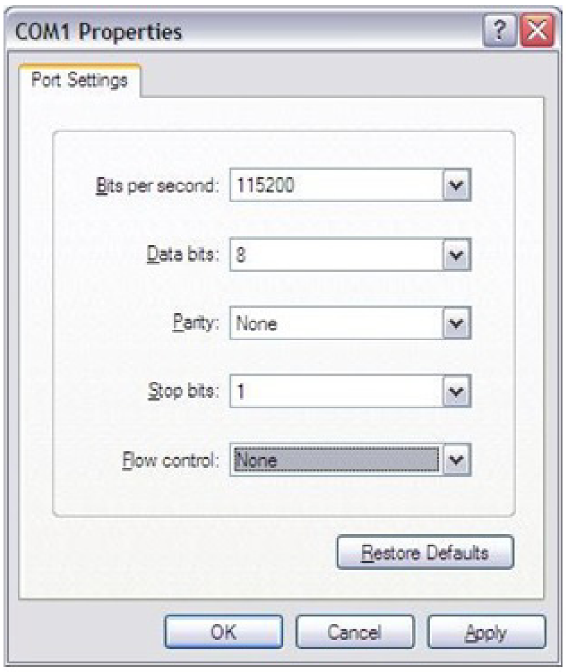
2.4.1 Debug Port Connection

1. Connect debug port cable to the UBC-220 debug port.
2. Connect the RS-232 extension cable to the debug cable.
3. Connector the other side of the extension cable to the USB-to-RS-232 cable then connect to your PC.

2.4.2 Debug Port setting

UBC-220 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 UBC-220 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 3.6.
3. After the bootloader is programmed on SD card, insert power adapter connector to DC jack on UBC-220 to power up the board. The bootloader prompt is displayed on the terminal screen.


COM1 Properties

Port Settings

Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Figure 2.23 Hyper Terminal Settings for Terminal Setup

2.5 Test Tools

All test tools must be verified on UBC-220, please prepare required test fixtures before verifying each specified I/O. If you have any problems to get the test fixture, please contact Advantech for help.

2.5.1 eMMC Test

1. Create a file and copy to eMMC.

```
#echo 123456789ABCDEF > test.txt
#dd if=./test.txt of=/dev/mmcblk0 bs=1024 count=1
seek=25118
```

```
 0+1 records in
 0+1 records out
 16 bytes (16 B) copied, 0.000109331 s, 146 kB/s
```

2. Check the data copied to eMMC

```
#hexdump -C /dev/mmcblk0 -s 25720832 -s 32
```

```
 01887800  31 32 33 34 35 36 37 38  39 41 42 43 44 45 46 0a
 |123456789ABCDEF.|
 01887810  1d 4f e2 19 d3 05 8b df  ab 4a 40 5a c5 23 3c f2
 |.O.......J@Z.#<.|
```

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.*

2.5.2 USB Test

1. Insert USB flash disk then assure it is in UBC-220 device list.

2. Create a file and copy to USB flash disk

```
#echo 123456789ABCDEF > test.txt
#dd if=./test.txt of=/dev/sda bs=1024 count=1 seek=25118
```

```
0+1 records in
0+1 records out
16 bytes (16 B) copied, 0.000109331 s, 146 kB/s
```

3. Check the data copied to USB flash disk

```
#hexdump -C /dev/sda -s 25720832 -s 32
```

```
01887800  31 32 33 34 35 36 37 38  39 41 42 43 44 45 46 0a
|123456789ABCDEF.|
 01887810  1d 4f e2 19 d3 05 8b df  ab 4a 40 5a c5 23 3c f2
|................|
```

Note! *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.*

2.5.3 SD Test

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

```
#ls /dev/mmcblk*
```

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

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.

```
#ls /dev/mmcblk*
```

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

3. Create a file and copy to SD

```
#echo 123456789ABCDEF > test.txt
#dd if=./test.txt of=/dev/mmcblk1 bs=1024 count=1
seek=25118
```

```
0+1 records in
0+1 records out
16 bytes (16 B) copied, 0.000109331 s, 146 kB/s
```

4. Check if the file is created successfully.

```
#hexdump -C /dev/mmcblk1 -s 25720832 -s 32
```

```
01887800  31 32 33 34 35 36 37 38  39 41 42 43 44 45 46 0a
|123456789ABCDEF.|
 01887810  1d 4f e2 19 d3 05 8b df  ab 4a 40 5a c5 23 3c f2
```

```
|.................|
```

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.*

2.5.4 LVDS/HDMI Test

2.5.4.1 Testing through gplay (for default single display)

1. #gplay /tools/Advantech.avi.
2. Then you can see the video demo on the default display screen.

2.5.4.2 Testing through gst-launch (for multi-display)

If you'd like to have multiple displays such as dual LVDS and HDMI output , you should set the parameters in uboot first. Please refer to section 3.7.5.3 for more details. Once the display method is set up, please follow the instruction below to run gst-launch to play video.

1. Turn ON the HDMI display, please type:

```
#gst-launch playbin2 uri=file:///tools/Advantech.avi
video-sink="mfw_v4lsink device=/dev/video16"&
```

You can see display independent both show Advantech.avi at the same time.

If you'd like to set the output audio as HDMI out, please add the parameter of plughw:

A. Plughw:1-->Output the audio through HDMI.

```
#gst-launch playbin2 uri=file:///tools/Advantech.avi
video-sink="mfw_v4lsink device=/dev/video17" audio-
sink="alsasink device=plughw:1"
```

If you'd like to change the display monitor, please refer to the below table:

video16	HDMI
video17	HDMI overlay
video18	LVDS0
video19	LVDS1

2.5.5 Mini PCIe (3G and Wifi) Test

The command used to test 3G module is as following, the supported module P/N is EWM-C106HT01E

```
#3glink

Send AT commands...
#send (AT^M)
send (ATDT*99#^M)
expect (CONNECT)
AT^M^M
OK^M
ATDT*99#^M^M
CONNECT
 -- got it
.........
```

The command used to test WIFI module is as follows, the supported module P/N is EWM-W142F01E.

```
#ifconfig wlan0 up
#iwlist wlan0 scanning
#wpa_passphrase "Wifi name" password > /tmp/wpa.conf
#wpa_supplicant -BDwext -iwlan0 -c/tmp/wpa.conf
#dhclient wlan0
```

2.5.6 LED Test

This test is to light up/turn off the configurable LED(LED2). These codes can be taken as the reference code for a SW AP.

```
cd /sys/class/gpio/
echo 1 > export
cd /sys/class/gpio/gpio1
```

LED on

```
echo 0 > value
```

LED off

```
echo 1 > value
```

2.5.7 OpenGL Test

Please follow below instructions to test OpenGL on UBC-220 platform:

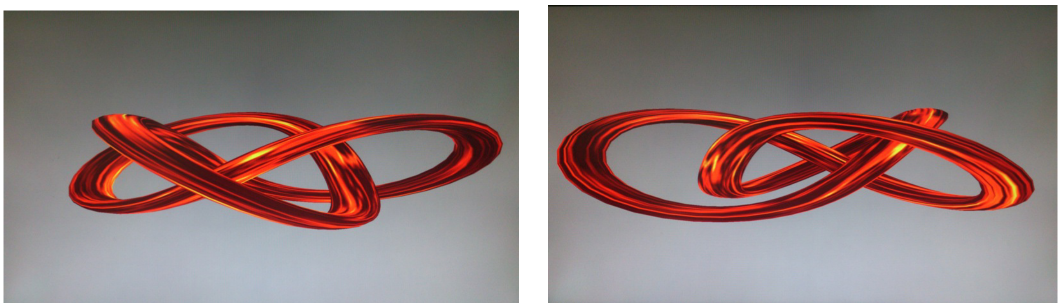
1. Change path to /opt/viv_samples/vdk

```
#cd /opt/viv_samples/vdk
#ls tutorial*
  tutorial1       tutorial2_es20  tutorial4
  tutorial5_es20
  tutorial1_es20  tutorial3       tutorial4_es20  tutorial6
  tutorial2       tutorial3_es20  tutorial5       tutorial7
```

2. Run tutorial7 for OpenGL ES 1.1

Using Vertex Buffer Objects (VBO) can substantially increase performance by reducing the bandwidth required to transmit geometry data. Information such vertex, normal vector, color, and so on is sent once to locate device video memory and then bound and used as needed, rather than being read from system memory every time. This example illustrates how to create and use vertex buffer objects.

```
#./tutorial7
```



3. Run tutorial3_es20 for OpenGL ES 2.0

A ball made of a mirroring material and centered at the origin spins about its Y-axis and reflects the scene surrounding it.

```
#./tutorial3_es20
```

2.5.8 LAN Test

UBC-220 sets DHCP as default network protocol.

```
#ifconfig
  eth0      Link encap:Ethernet  HWaddr 00:04:9F:01:30:E0
            inet addr:172.17.21.96  Bcast:172.17.21.255
            Mask:255.255.254.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500
            Metric:1
            RX packets:129 errors:0 dropped:18 overruns:0
            frame:0
            TX packets:2 errors:0 dropped:0 overruns:0
            carrier:0
            collisions:0 txqueuelen:1000
           RX bytes:15016 (14.6 KiB)  TX bytes:656 (656.0 B)

  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:0 errors:0 dropped:0 overruns:0
            frame:0
            TX packets:0 errors:0 dropped:0 overruns:0
            carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

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 (UBC-220) IP is 172.17.21.97; the target(A desktop computer) IP is 172.17.20.192

```
#ifconfig eth0 172.17.21.97 up
#ifconfig eth0

  eth0      Link encap:Ethernet  HWaddr 00:04:9F:01:30:E0
            inet addr:172.17.21.97  Bcast:172.17.255.255
            Mask:255.255.0.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500
            Metric:1
            RX packets:2851 errors:0 dropped:271 overruns:0
            frame:0
            TX packets:30 errors:0 dropped:0 overruns:0
            carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:291407 (284.5 KiB)  TX bytes:2000 (1.9
            KiB)
```

The target computer (Client) IP address is 172.17.20.192, so we can use the below command to see if we can get any response from the client

```
#ping 172.17.20.192

  PING 172.17.20.192 (172.17.20.192): 56 data bytes
  64 bytes from 172.17.20.192: seq=0 ttl=128 time=7.417 ms
  64 bytes from 172.17.20.192: seq=1 ttl=128 time=0.203 ms
  64 bytes from 172.17.20.192: seq=2 ttl=128 time=0.300 ms

  --- 172.17.20.192 ping statistics ---
  3 packets transmitted, 3 packets received, 0% packet loss
  round-trip min/avg/max = 0.203/2.640/7.417 ms
```

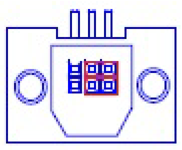
2.5.9 RS232 Test

As you can see below, there are 2 RS-232 supported by UBC-220. /dev/ttymxc0 is reserved for UBC-220 debug port (UBC-220 DEBUG_CONSOLE), the other one could be applied by the user.

```
#setserial -g /dev/ttymxc*

  /dev/ttymxc0, UART: undefined, Port: 0x0000, IRQ: 58
  /dev/ttymxc1, UART: undefined, Port: 0x0000, IRQ: 59
```

Below test was done with four 2.54mm pitch mini jumpers. Advantech P/N is 1653003100. This mini jumper is a bridge connecting Tx and Rx.



2.5.9.1 /dev/ttymxc1 testing (COM)

```
#stty -F /dev/ttymxc1 -echo
#cat /dev/ttymxc1
[CTRL+Z]
#echo hello > /dev/ttymxc1
#fg
  Hello
[CTRL+C]
```

2.5.10 Watchdog Timer Test

1. Executing ‘ wdt_driver_test.out ‘

```
./wdt_driver_test.out 5 10 0
  Usage: wdt_driver_test <timeout> <sleep> <test>
  timeout: value in seconds to cause wdt timeout/reset
  sleep: value in seconds to service the wdt
  test: 0 - Service wdt with ioctl(), 1 - with write()
```

2. Please try below command to set timeout as 10 seconds, system will reboot after then.

```
#/unit_tests/wdt_driver_test.out 5 10 0
  Starting wdt_driver (timeout: 10, sleep: 5, test: ioctl)
  Trying to set timeout value=10 seconds
  The actual timeout was set to 10 seconds
  Now reading back -- The timeout is 10 seconds
```

Press [CTRL+C] then you should be able to see below result:

```
  imx2-wdt imx2-wdt.0: Unexpected close: Expect reboot!
```

Then system will reboot in 10 seconds.

2.5.11 Photo Demo Test

Execute the following commands to run the Photo demo application on UBC-220.

```
#cd /tools
#./fbv Advantech.jpg
```

Then you can see the photo demo on the default display screen.

2.5.12 LED Test

Below is the test script for system LED2.

```
#cd /sys/class/gpio/
#echo 1 > export
#cd /sys/class/gpio/gpio1
```

LED on

```
#echo 0 > value
```

LED off

```
#echo 1 > value
```

Chapter 3

Software Functionality

This chapter details the Linux operating system on the UBC-220 platform.

3.1 Introduction

UBC-220 platform is an embedded system with Linux kernel 3.0.35 inside. It contains all system-required shell commands and drivers ready for UBC-220 platform. We do not offer IDE development environment in UBC-220 BSP, users can evaluate and develop under the Ubuntu 10.04LTS environment.

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

It will not be able to boot into the Linux environment successfully if one of above three files is missing from the boot media (SD card, SATA HDD or onboard flash)

The purpose of this chapter is to introduce software development of UBC-220 to you, so that you can develop your own application(s) efficiently.

UBC-220 is designed for Linux only. For now the official supported host version is Ubuntu 10.04 LTS, hosting any other version may have compatibility issues. In this case, we strongly recommend to have Ubuntu 10.04 LTS installed to your host PC before start UBC-220 evaluation/development.

3.2 Package Content

We can offer you two different kinds of Linux package for UBC-220. One is a prebuilt system image for system recovery another is source code package (BSP).

3.2.1 Source Code Package

UBC-220 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 the open source community. UBC-220 source code package is composed of six main folders: “cross_compiler”, “document”, “image”, “package”, “scripts”, and “source”.

Note! *UBC-220 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 U220LBV2660 package contents:

■ **“cross_compiler”**	--> This folder contains source code for cross compiler.
■ **“document”-**	--> This folder contains user guide.
■ **“image”**	-->This folder contains the uImage, u-boot_crc.bin, u- boot_crc.bin.crc.
■ **“image/rootfs”**	-->This folder contains Linux root file system
■ **“package”**	-->This folder contains source code provided by Freescale 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 Freescale
```

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 Freescale 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 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 "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-3.0.35" 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 portable 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 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-3.0.35/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 Set up Build Environment

All instructions in this guide are based on Ubuntu 10.04 LTS developing environment. Please install the Ubuntu 10.04 LTS at your PC/NB in advance.

When you obtain the UBC-220 Linux source code package, please refer to following instructions to extract to your developing environment:

1. Copy "U220LBV2660" package to your desktop.
2. Start your "Terminal" on Ubuntu 10.04 LTS.
3. **`$sudo su`** (Change to “root” authority)
4. Input user password
5. **`#cd Desktop/`**
6. **`#tar xvf U220LBV2660.tgz`** (Unzip file)

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 10.04 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. Then you can start to code the source code, build images, or compile applications.

3.3.1 setenv.sh

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.

The major part of setenv.sh is shown as following:

```
export SRCROOT=${PWD}/..
export CC_PATH=${SRCROOT}/cross_compiler/fsl-linaro-toolchain
export CROSS_COMPILE=${CC_PATH}/bin/arm-none-linux-gnueabi-
export CC=${CROSS_COMPILE}gcc
export STRIP=${CROSS_COMPILE}strip
export ARCH=$rm
export KROOT=${SRCROOT}/source/linux-3.0.35
export UBOOT_SOURCE=${SRCROOT}/source/u-boot-2009.08
export ROOTFS=${SRCROOT}/image/rootfs
export LOG=${SRCROOT}/Build.log
export PATH=${CC_PATH}/bin:${UBOOT_SOURCE}/tools:$PATH
```

Note! *You have to wrap "setenv.sh" once you open a new "Terminal" utility every time.*
(i.e. #source setenv.sh)

Note! *It is suggested to change to "root" authority to use the source code.*

3.4 Build Instructions

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

3.4.1 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 10.04 LTS..
2. **`$sudo su`** (Change to "root" authority)
3. Input user password.
4. **`#. setenv.sh`** (To configure the developing environment automatically)
5. **`#./cfg_uboot.sh imx6dl_ubc220_config`** (To set the u-boot configuration automatically)
6. **`#./mk_uboot.sh`** (Start to build the u-boot)
7. Then you can see u-boot_crc.bin and u-boot_crc.bin.crc are being built and located in ../image.

3.4.2 Build Linux Kernel Image

Advantech offer you a script to build the "uImage" quickly. You can build uImage by follow below steps:

1. Open "Terminal" on Ubuntu 10.04 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 imx6dl_ubc220_defconfig`** (To set the uImage configuration automatically).
7. **`#./mk_kernel.sh`** (Start to build the uImage)
8. Then you can see uImage is being built and located in ../image.

3.4.3 Build Log

You can find the build log from folder "U220LBV2660". 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.

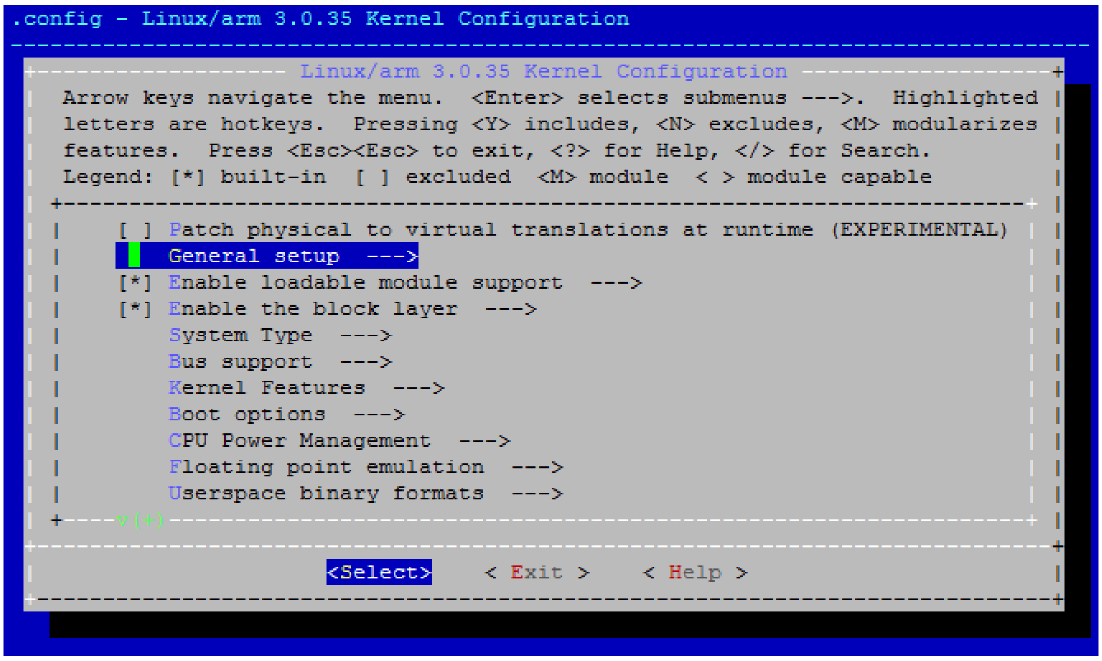
3.5 Source Code Modification

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.

3.5.1 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 use the following steps:

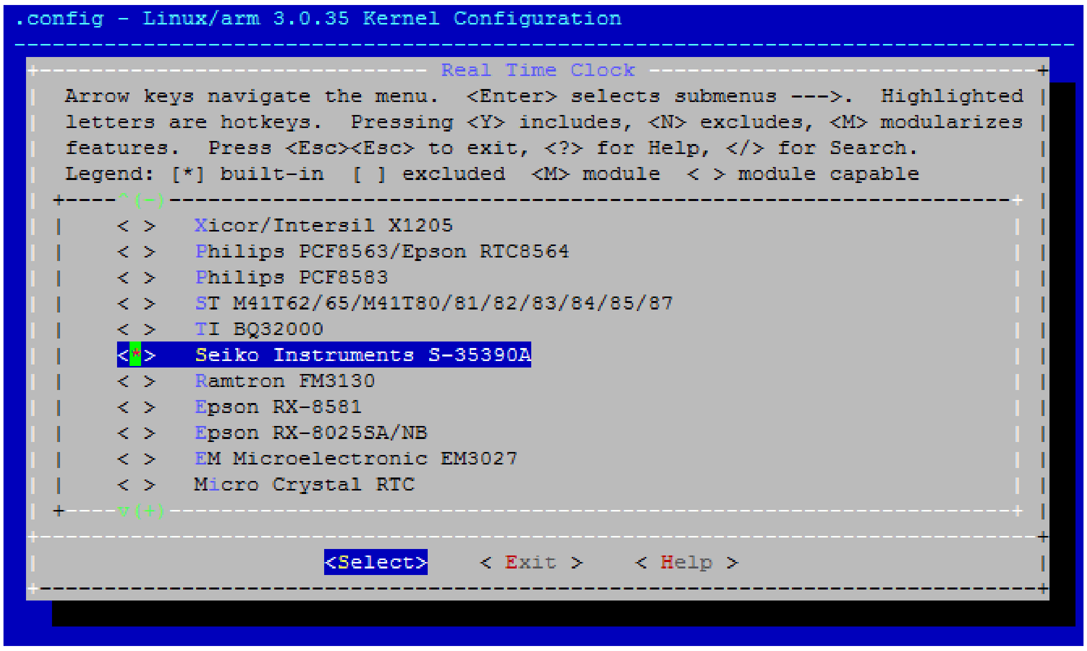
1. Open "Terminal" on Ubuntu 10.04 LTS.
2. **$sudo su** (Change to “root” authority)
3. Input user password.
4. Change directory to BSP's scripts folder.
5. **#. setenv.sh** (To configure the developing environment automatically)
6. **#./cfg_kernel.sh menuconfig**
7. Then you will see a GUI screen (Linux 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

8. 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.



```
.config - Linux/arm 3.0.35 Kernel Configuration
---------------------------------------------------------------------------------
+-------------------------------- Real Time Clock --------------------------------+
|  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              |
| +----^(-)--------------------------------------------------------------------+ |
| |        < >   Xicor/Intersil X1205                                           | |
| |        < >   Philips PCF8563/Epson RTC8564                                  | |
| |        < >   Philips PCF8583                                                | |
| |        < >   ST M41T62/65/M41T80/81/82/83/84/85/87                          | |
| |        < >   TI BQ32000                                                     | |
| |        <*>   Seiko Instruments S-35390A                                     | |
| |        < >   Ramtron FM3130                                                 | |
| |        < >   Epson RX-8581                                                  | |
| |        < >   Epson RX-8025SA/NB                                             | |
| |        < >   EM Microelectronic EM3027                                      | |
| |        < >   Micro Crystal RTC                                              | |
| +----v(+)--------------------------------------------------------------------+ |
+---------------------------------------------------------------------------------+
|                   <Select>    < Exit >    < Help >                              |
+---------------------------------------------------------------------------------+
```

Figure 3.2 Selecting Seiko Instruments S-35390A

9. Change directory to "source/linux-3.0.35/arch/arm/mach-mx6", edit the "board-mx6dl_ubc220.h" and "board-mx6q_advantech.c".
Please add below codes to source/linux-3.0.35/arch/arm/mach-mx6/board-mx6dl_ubc220.h:

```
static struct i2c_board_info mxc_i2c0_board_info[] __initdata
= {
        {
                I2C_BOARD_INFO("adv-wdt-i2c", 0x29),
        },
        {
                I2C_BOARD_INFO("s35390a", 0x30),
        },
};
```

Please add below codes to
source/linux-3.0.35/arch/arm/mach-mx6/board-mx6q_advantech.c

```
i2c_register_board_info(0, mxc_i2c0_board_info,
                                ARRAY_SIZE(mxc_i2c0_board_info));
```

10. Please refer to former Chapter 3.3.2 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.*

3.5.2 Chang UBC-220 Boot Logo

By default, UBC-220 shows a boot logo when booting up. You can replace the logo to whatever your want by following below steps:

1. You have to download “netpbm” corresponding to your OS version from internet first,
2. Install “netpbm” by typing **`$sudo apt-get install netpbm`**.
3. Prepare your boot logo. For example: bootlogo.png (Under folder Desktop/boot-logo)

Note! *This picture should be in PNG format and less than 224 colors. It is suggested to have the image resolution equal to your LCD panel size.*

4. Open "Terminal" on Ubuntu 10.04 LTS.
5. **`$sudo su`** (Change to “root” authority)
6. Input user password.
7. **`#cd Desktop/bootlogo`** (Go into the folder that bootlogo.png located)
8. **`#pngtopnm bootlogo.png | ppmquant 224 | pnmtoplainpnm > logo_linux_clut224.ppm.`**
9. **`Copy logo_linux_clut224.ppm to the directory source/linux-3.0.35/drivers/video/logo/.`**
10. Then you can refer Chapter 3.3.1 to rebuild the kernel with your own boot logo.

3.6 Create a Linux System Boot Media

UBC-220 supports boot from SD card and onboard flash. This section will guide you how to build a image for UBC-220 Linux system boot media.

3.6.1 Create a Linux System SD Card

3.6.1.1 From Source Code Package

When you receive the UBC-220 Linux source code package, you can refer following steps to create a Linux system SD card for booting up from it.

1. Open "Terminal" on Ubuntu 10.04 LTS.
2. **`$sudo su`** (Change to “root” authority)
3. Input your password.
4. Insert one SD card to your developing computer
5. Check the SD card location, like: /dev/sdf
6. Change directory to BSP's scripts folder.
7. **`#./mksd-linux.sh /dev/sdf`**
8. Type “y” (Start to copy files, wait until it shows [Done])

Then insert the Linux system SD card to UBC-220 SD card slot (SD1), it will boot up with Linux environment.

3.6.2 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.

1. Refer to Chapter 3.5.1 to make a Linux system SD card

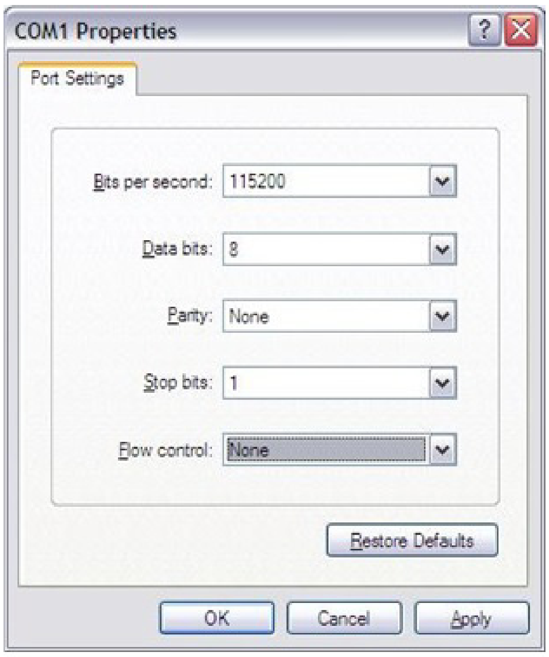
2. Insert this Linux system SD card to ROM-DB7500 and connect serial console.
3. On UBC-220 platform, type #root (Login)
4. On UBC-220 platform, type #cd /mk_inand
5. On UBC-220 platform, type #./mkinand-linux.sh /dev/mmcblk0
6. On UBC-220 platform, type "y "(Start to copy files, wait until it shows [Done])
7. Power off and remove this SD card.

Then you can boot from onboard flash without SD card.

3.7 Debug Message

UBC-220 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, Tera Term or PuTTY is must required. Below is the detail instruction of how to set up serial console, a "HyperTerminal" on a Windows host:

1. Connect UBC-220 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.



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

Figure 3.3 HyperTerminal Settings for Serial Console Setup

3.8 Linux Software AP and Testing on UBC-220

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 UBC-220 will be introduced in this section.

3.8.1 “Hello World!” Application and Execution

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

1. Open "Terminal" on Ubuntu 10.04 LTS.
2. `$sudo su` (Change to “root” authority)
3. Type user password.
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 UBC-220, then open serial console.
16. On UBC-220 platform, type `#root` (Login)
17. On UBC-220 platform, type `#cd /tool`
18. On UBC-220 platform, type `#./helloworld`
19. Now you should be able to see “Hello World!” shown on UBC-220.

3.8.2 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;
   }
   timeout = 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).`

3.8.3 RS232 Initial Code

The RS232 initial code as below. It shows you how to initialize COM2 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);
}
```

3.8.4 Display Output Setting

3.8.4.1 Single Display Settings

HDMI out, please set in u-boot as below:

```
setenv bootargs_base 'setenv bootargs console=ttymxc0,115200
enable_wait_mode=off
video=mxcfb1:off    video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24
video_mode=display3 pcie_testmode=off'
```

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

```
setenv bootargs_base 'setenv bootargs console=ttymxc0,115200
enable_wait_mode=off
video=mxcfb1:off       video=mxcfb0:dev=ldb,800x480M@60,if=RGB24
video_mode=display3 pcie_testmode=off'
```

3.8.4.2 Dual Display Settings

When you want to display LVDS and HDMI output , please set parameter in U-boot as following. This is the default settings in U-boot.

```
setenv bootargs_base 'setenv bootargs console=ttymxc0,115200
enable_wait_mode=off video_mode=display3 pcie_testmode=off'
```

For display interface clock, there are several options (Independently for each port) listed below:

1. Derived from the IPU internal clock (Master Mode)
2. Provided by an external source (Slave Mode)
3. The transfer rate supported

When a single port is active, the pixel clock rate is up to 264 MHz

When both LVDS ports are active, you have to follow below condition:

1) Each pixel clock rate may be up to 220 MHz**

2) The sum of pixel clock rates is up to 240 MHz

Note! *Specified pixel clocks frequencies are applicable for internal clocks, but may be limited by IO buffers speed capability. Final numbers are subjected to AC characterization.*

3.8.5 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  followed  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  interface 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 interface.

      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.
```

3.8.6 Storage (eMMC/SD Card)

The storages devices name as following:

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

3.8.7 3 G Sample Code

The code of 3glink, we have tried this command in 3 G test (Section 1.8).

```
#!/bin/bash

echo "Send AT commands..."

pppd connect 'chat -v -s -t 10 "" "AT" "" "ATDT*99#" "CON-
NECT" ""' user username password password /dev/ttyUSB2
460800 nodetach crtscts debug usepeerdns defaultroute &
```

Chapter 4

System Recovery

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

4.1 Introduction

This section provides detail procedures for restoring the eMMC image. You can do a system recovery following the steps below if you destroy the onboard flash image by accident.

1. Open "Terminal" on Ubuntu 10.04 LTS.
2. **`$sudo su`** (Change to “root” authority).
3. Input your password.
4. Insert one SD card to your developing computer.
5. Check the SD card location, like: /dev/sdf.
6. Change directory to BSP's scripts folder.
7. **`#./mksd-linux.sh /dev/sdf.`**
8. Type “y” (Start to copy files, wait until it shows [Done]).
9. Connect console cable to debug port (CN1) and open serial console program on Ubuntu 10.04 LTS, set baudrate to 115200. For detail console setting, please refer to section 3.6.
10. On UBC-220 platform, type **`#root`** (Login).
11. On UBC-220 platform, type **`#cd /mk_inand.`**
12. On UBC-220 platform, type **`#./mkinand-linux.sh /dev/mmcblk0.`**
13. On UBC-220 platform, type “`y` “
    (Start to copy files, wait until it shows [`Done`]).
14. Power off and remove this SD card.

Chapter 5

Advantech Services

This chapter introduces Advantech design in serviceability, technical support and warranty policy for UBC-220.

5.1 RISC Design-in Services

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 Global Service Policy

5.3.1 Warranty Policy

Below is the warranty policy of Advantech products:

5.3.1.1 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.

5.3.1.2 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.

5.3.1.3 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 lighting strike, flood, earthquake, etc.
- Product updates/upgrades and tests upon the request of customers who are without warranty.

5.3.2 Repair Process

5.3.2.1 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.

5.3.2.2 Returning the Product for Repair

It's possible customers can save time and meet end-user requirements by returning defective products to an y 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.

5.3.2.3 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 lighting 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.

5.3.2.4 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.

5.3.2.5 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.

5.3.2.6 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.

Enabling an Intelligent Planet