IBR210

3.5” ARM-based SBC
With NXP Cortex™ A53 Quad i.MX8M

User’s Manual

Version 1.3
(Nov 2021)

iBASE

Copyright

Disclaimer

IBASE reserves the right to make changes and improvements to the products described in this document without prior notice. Every effort has been made to ensure the information in the document is correct; however, IBASE does not guarantee this document is error-free.
IBASE assumes no liability for incidental or consequential damages arising from misapplication or inability to use the product or the information contained herein, nor for any infringements of rights of third parties, which may result from its use.

Trademarks

All the trademarks, registrations and brands mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners.

Compliance

CE

This product conforms to health, safety, and environmental protection standards for items sold within the European Economic Area (EEA).

FC

This product has been tested and found to comply with the limits for a Class B 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 manufacturer's instructions, may cause harmful interference to radio communications.

WEEE

This product must not be disposed of as normal household waste, in accordance with the EU directive of for waste electrical and electronic equipment (WEEE - 2012/19/EU). Instead, it should be disposed of by returning it to a municipal recycling collection point. Check local regulations for disposal of electronic products.

Green IBASE

This product is compliant with the current RoHS restrictions and prohibits use of the following substances in concentrations exceeding 0.1% by weight (1000 ppm) except for cadmium, limited to 0.01% by weight (100 ppm).

- Lead (Pb)
- Mercury (Hg)
- Cadmium (Cd)
- Hexavalent chromium (Cr6+)
- Polybrominated biphenyls (PBB)
- Polybrominated diphenyl ether (PBDE)

Important Safety Information

Carefully read the precautions before using the board.

Environmental conditions:

- Use this product in environments with ambient temperatures between 0˚C and 70˚C. (Industrial grade: -20° C and 85° C)
- Do not leave this product in an environment where the storage temperature may be below -40° C or above 85° C. To prevent from damages, the product must be used in a controlled environment.

WARNING

Attention during use:

- Do not use this product near water.
- Do not spill water or any other liquids on this product.
- Do not place heavy objects on the top of this product.

Anti-static precautions

- Wear an anti-static wrist strap to avoid electrostatic discharge.
- Place the PCB on an anti-static kit or mat.
- Hold the edges of PCB when handling.
- Touch the edges of non-metallic components of the product instead of the surface of the PCB.
- Ground yourself by touching a grounded conductor or a grounded bit of metal frequently to discharge any static.

CAUTION

Danger of explosion if the internal lithium-ion battery is replaced by an incorrect type. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions or recycle them at a local recycling facility or battery collection point.

Warranty Policy

- **IBASE standard products:**

  24-month (2-year) warranty from the date of shipment. If the date of shipment cannot be ascertained, the product serial numbers can be used to determine the approximate shipping date.

- **3rd-party parts:**

  12-month (1-year) warranty from delivery for the 3rd-party parts that are not manufactured by IBASE, such as CPU, CPU cooler, memory, storage devices, power adapter, panel and touchscreen.

* PRODUCTS, HOWEVER, THAT FAIL DUE TO MISUSE, ACCIDENT, IMPROPER INSTALLATION OR UNAUTHORIZED REPAIR SHALL BE TREATED AS OUT OF WARRANTY AND CUSTOMERS SHALL BE BILLED FOR REPAIR AND SHIPPING CHARGES.

Technical Support & Services

1. Visit the IBASE website at www.ibase.com.tw to find the latest information about the product.
2. If you need any further assistance from your distributor or sales representative, prepare the following information of your product and elaborate upon the problem.
   - Product model name
   - Product serial number
   - Detailed description of the problem
   - The error messages in text or in screenshots if there is any
   - The arrangement of the peripherals
   - Software in use (such as OS and application software, including the version numbers)
3. If repair service is required, you can download the RMA form at http://www.ibase.com.tw/english/Supports/RMAService/. Fill out the form and contact your distributor or sales representative.

Table of Contents

This page is intentionally left blank.

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Specifications
- Block Diagram
- Product View
- Board Dimensions

1.1 Introduction

IBR210 is a 3.5" Disk-Size SBC powered by the NXP i.MX8M Cortex-A53 1.3GHz processor. It offers 2D, 3D graphics and multimedia accelerations, and supports numerous peripheral interfaces, including RS-232/422/485, COM port, GPIO, USB3.0/2.0, LAN and audio. For the display, it also supports 1 HDMI for a 4K display or FHD Dual-channel LVDS. Other features are an M.2 Key-E, type 2230 and mini-PCIe expansion slots that are well suited for IoT applications.

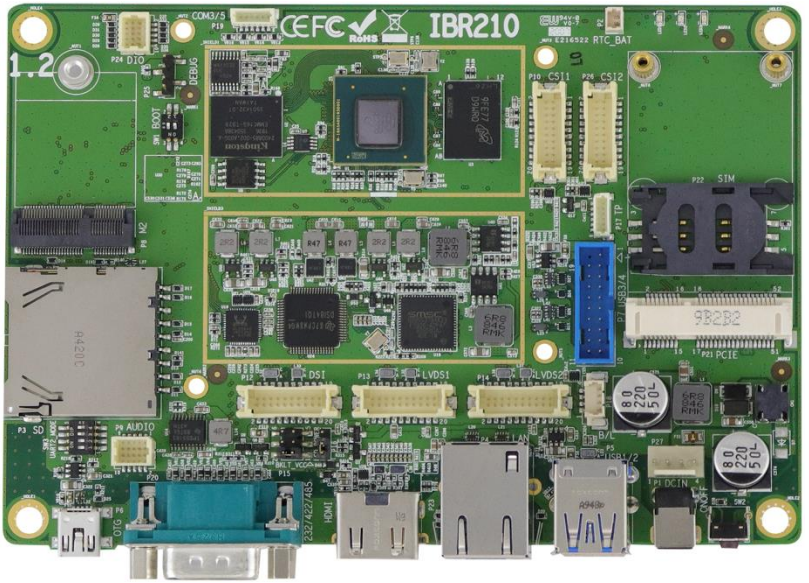


Photo of IBR210

1.2 Features

- With NXP Cortex-A53/Cortex-M4, i.MX 8M Quad 1.3GHz Industrial-Grade Processor
- Supports HDMI2.0a, or dual-channel FHD LVDS
- Supports 3GB LPDDR4, 16GB eMMC and SD socket
- Supports embedded I/O for COM, GPIO, USB3.0, Audio and Ethernet
- Supports M.2 Key-E (2230) and mini-PCI-E with SIM socket for wireless and LTE connectivity

1.3 Packing List

Your IBR210 package should include the items listed below. If any of the items below is missing, contact the distributor or dealer from whom you purchased the product.

- IBR210 3.5” SBC x 1

1.4 Specifications

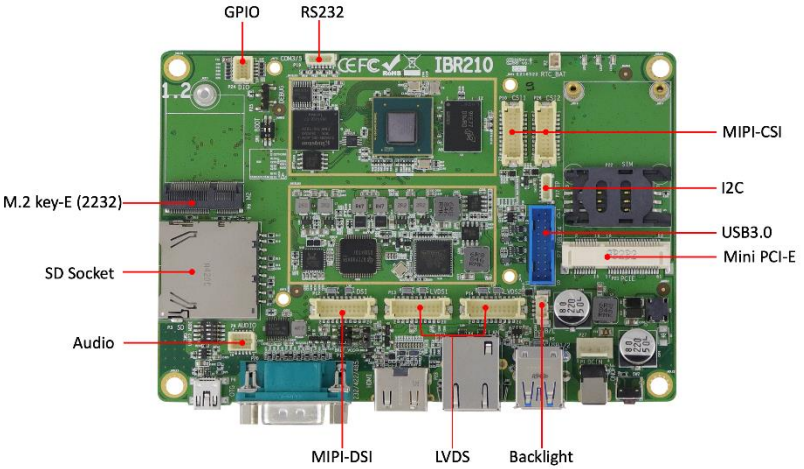
Product Name	IBR210
Form Factor	3.5” SBC
System	
Operating System	• Yocto v2.5 (Kernel 4.14.62) • Android 9 (Kernel 4.14.98)
CPU Type	NXP Cortex™ A53 i.MX8M Quad Core Industrial-Grade SoC
CPU Speed	Up to 1.3 GHz
Memory	• **System memory:** 3 GB LPDDR4 • **Data Memory:** 16 GB eMMC
Video Codec	• 4Kp60 HEVC/H.265 main, and main 10 decoder • 4Kp60 VP9 decoder • 4Kp30 AVC/H.264 decoder • 1080p60 MPEG-2, MPEG-4p2, VC-1, VP8, RV9, AVS, MJPEG, H.263 decoder
RTC	IDT 1337AGDVGI8
Wireless	Wi-Fi/BT, LTE module (Optional)
Power Supply	12-24VDC-In Jack and Internal header
Watchdog Timer	Yes (256 segments, 0, 1, 2…128 secs)
Dimensions	146 x 102 mm (5.74” x 4.02”)
RoHS	Yes
Certification	CE, FCC Class B
I/O Ports	
DC Jack	• 1 x 12-24V DC jack
Display	• Dual-Channel LVDS (FHD) • HDMI V2.0a
Camera	• 2x MIPI-CSI (2*10 pin header)
LAN	• 1 x RJ45 GbE LAN
USB	• 2 x USB 3.0 Type A • 2 x USB 3.0 internal port

Serial	• 1x I2C header • 1x 2-wire RS232 header (for Debug Console Port) • 2x 2-wire RS232 header • 1x RS232/422/485 D-Sub connector
Audio	• 1 x Audio header (Line-in and Line-out)
Digital IO	• 8x GPIO (2*5 pin header 1.0mm)
Expansion Slots	• 1x M.2 Key-E (2230) w/ USB, SDIO, UART, PCI-E • 1x Mini PCI-E w/ SIM socket
Environment	
Operating Temperature	• 0 ~ 70 °C (32 ~ 158 °F) • -20 ~ 80 °C (-40 ~ 185 °F) / Industrial Grade
Relative Humidity	10 ~ 90 %, non-condensing

All specifications are subject to change without prior notice.

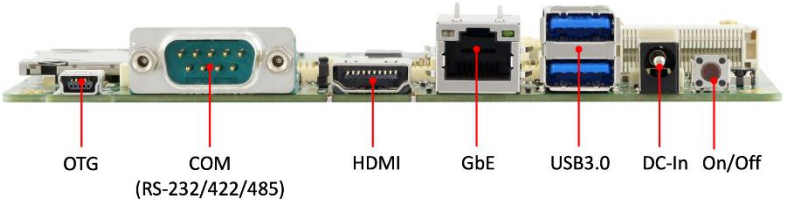
1.5 Product View

Top View



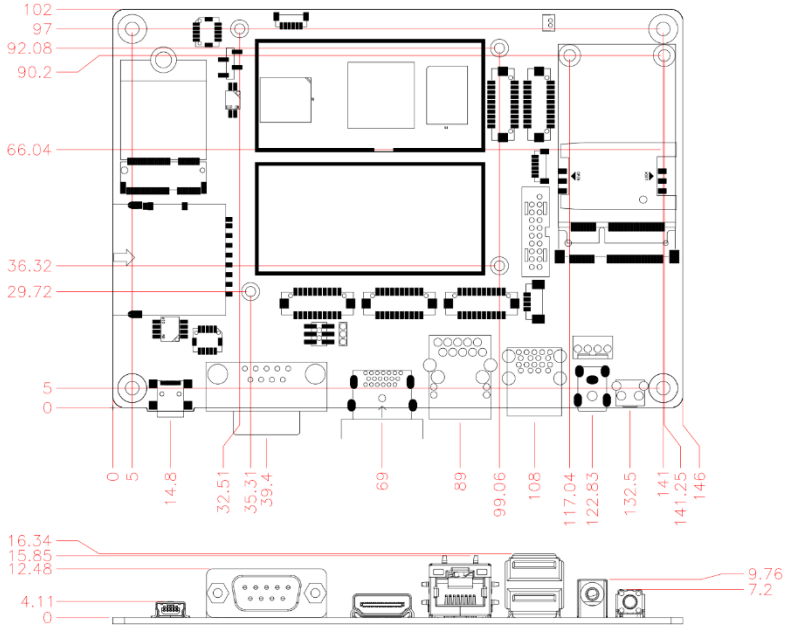
* The above photo is for reference only. Some minor components may differ.

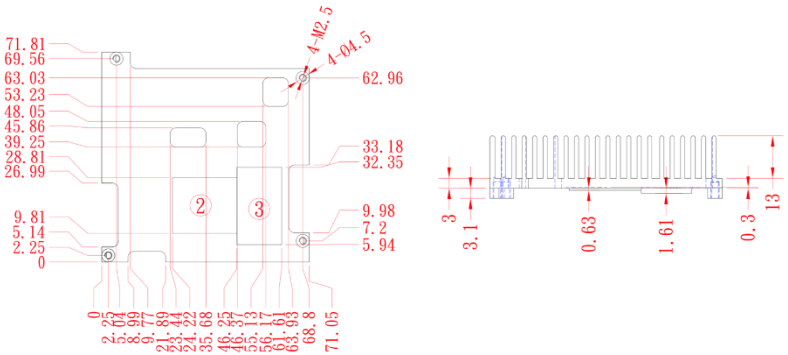
I/O View



1.6 Dimensions

Unit: mm




71.81
69.56
63.03
53.23
48.05
45.86
39.25
28.81
26.99
9.81
5.14
2.25
0
4-M2.5
4-Φ4.5
62.96
33.18
32.35
9.98
7.2
5.94
2
3
0
2.25
5.04
8.99
9.77
21.89
23.44
24.22
35.68
46.25
46.37
55.13
56.17
61.61
63.93
68.8
71.05
3
3.1
0.63
1.61
0.3
13

Chapter 2 Hardware Configuration

This section provides information on jumper settings and connectors on the IBR210 in order to set up a workable system. The topics covered are:

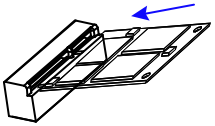
- Mini-PCIe & M.2 card Installation
- Jumper and connector locations
- Jumper settings and connector information

2.1 Mini-PCIe & M.2 Card Installation

To install the mini-PCIe and M.2 cards, perform the following steps.

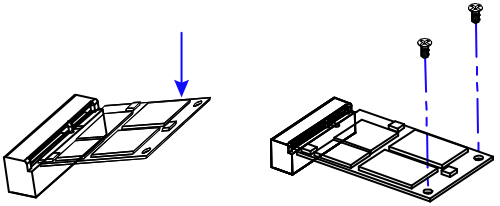
1. Locate the mini-PCIe slot, align the key of the mini-PCIe card to the interface, and insert the card slantwise.

   (Insert the M.2 card in the same way.)



2. Push the mini-PCIe card down and fix it with 2 flat head screws.

   (Fix the M.2 card with one screw.)

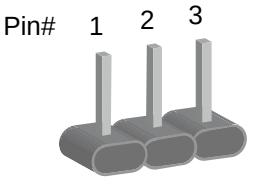


2.2 Setting the Jumpers

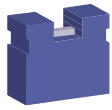
Set up and configure your IBR210 by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

How to Set Jumpers

Jumpers are short-length conductors consisting of several metal pins with a non-conductive base mounted on the circuit board. Jumper caps are used to have the functions and features enabled or disabled. If a jumper has 3 pins, you can connect either PIN1 to PIN2 or PIN2 to PIN3 by shorting.

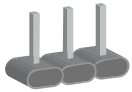
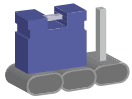
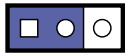
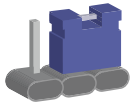


A 3-pin jumper



A jumper cap

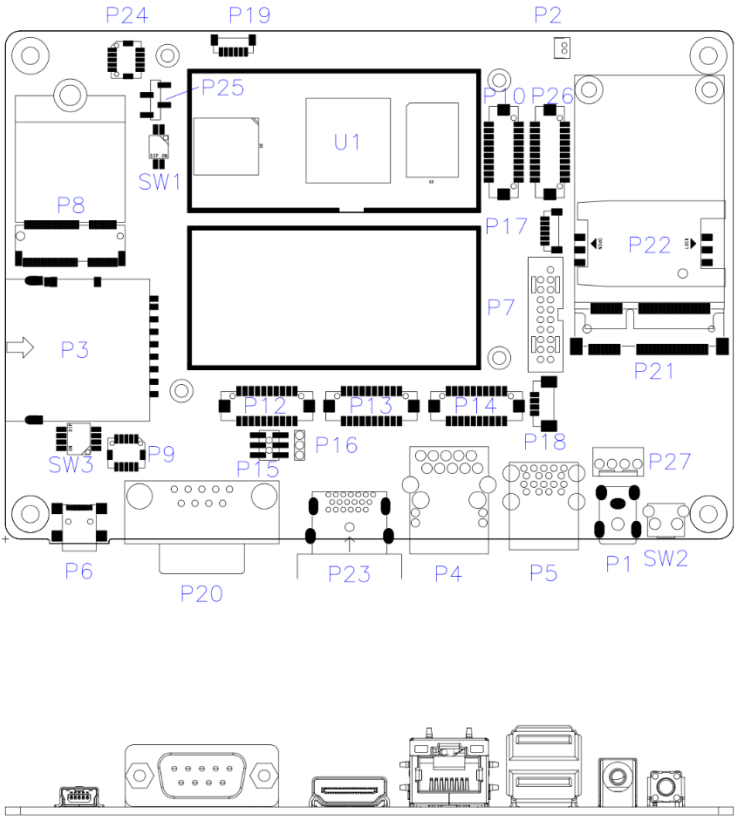
Refer to the illustration below to set jumpers.

Pin closed	Oblique view	Illustration in the manual
Open		
1-2		
2-3		

When two pins of a jumper are encased in a jumper cap, this jumper is **closed**, i.e. turned **On**.

When a jumper cap is removed from two jumper pins, this jumper is **open**, i.e. turned **Off**.

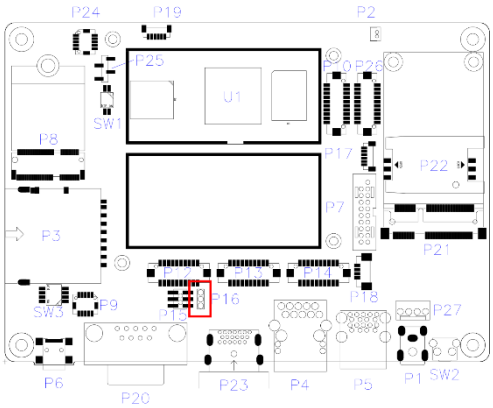
2.3 Jumper & Connector Locations on IBR210

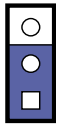
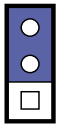


2.4 Jumpers Quick Reference

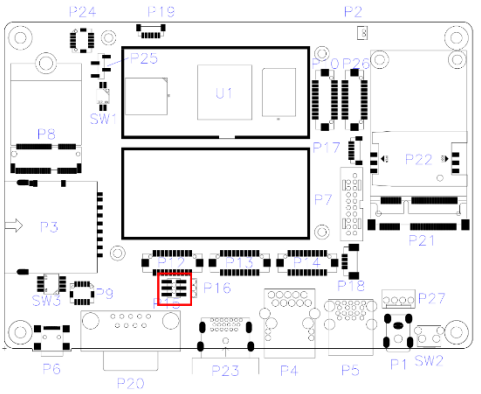
Function	Jumper	Page
LVDS Power Setting	P16	13
LVDS Backlight Power Setting	P15	14

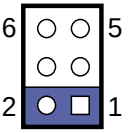
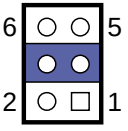
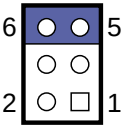
2.4.1 LVDS Power Setting (P16)



Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	2-3	

2.4.2 LVDS Backlight Power Setting (P15)

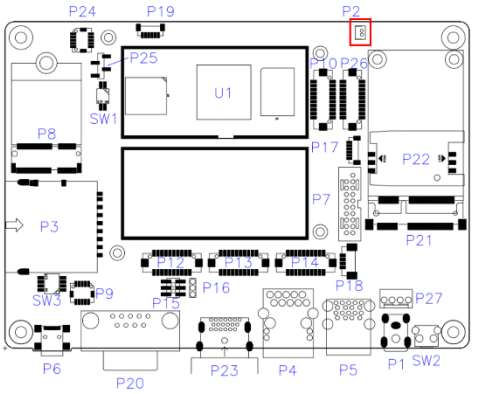


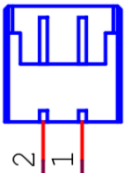
Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	3-4	
12V	5-6	

2.5 Connectors Quick Reference

Function	Connector Name	Page
RTC Lithium Cell Connector	P2	24
RS-232/422/485 COM Port Selection	SW3	24
RS-232/422/485 COM Port	P20	25
LVDS Display Connector	P13, P14	26
RS-232 COM Port Connector	P19	27
LVDS Backlight Control Connector	P18	28
Audio Line-In & Line-Out Connector	P9	28
USB Hub Connector	P7	29
Digital I/O (GPIO) Connector	P24	30
MIPI-CSI Connector	P10, P26	31
MIPI-DSI Connector	P12	32
I^2C Connector	P17	33
DC Power Input	P1, P27	34
SD Card Slot	P3	- -
HDMI Port	P23	- -
GbE LAN Port	P4	- -
Dual USB 3.0 Type-A Port	P5	- -
Mini-USB OTG Port	P6	- -
NGFF M.2 E2230 Slot	P8	- -
Mini-PCIe Slot	P21	- -
SIM Card Socket	P22	- -
System ON/OFF Button	SW2	- -
Factory Use Only	SW1, P25	35,36

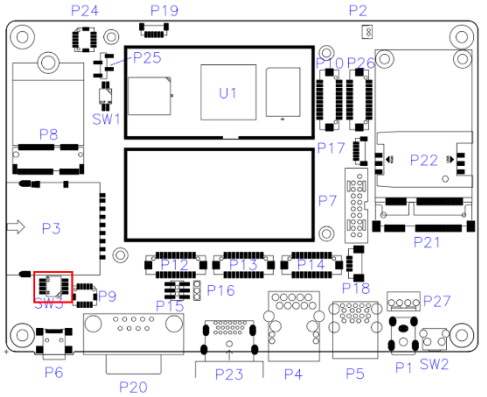
2.5.1 RTC Lithium Cell Connector (P2)

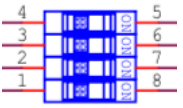




Pin	Assigment	Pin	Assigment
1	RTC_VCC	2	Ground

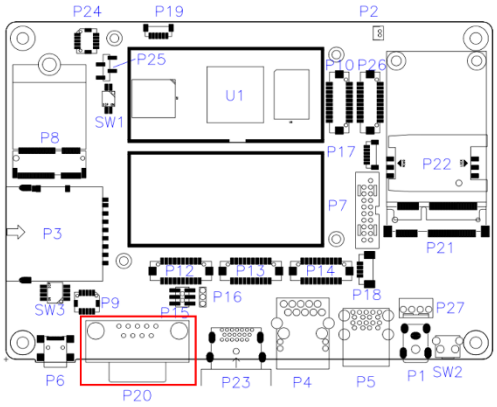
2.5.2 RS-232/422/485 COM Port Selection (SW3)

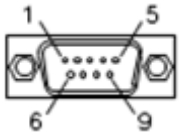




Panel Type	1-8	2-7	3-6	4-5
RS-422 Full Duplex	Off	On	On	On
RS-232 (Default)	Off	Off	On	On
RS-485 Half Duplex (TX Low-Active)	Off	On	Off	On
RS-485 Half Duplex (TX High-Active)	Off	Off	Off	On
Function Off	Off	Off	Off	Off

2.5.3 RS-232/422/485 COM Port (P20)

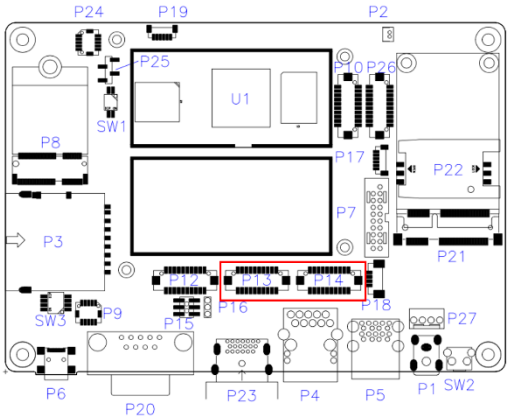


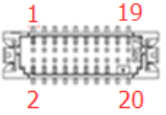


Refer to the SW3 setting for RS-232/422/485 mode selection.

Pin	Assignment		
	RS-232	RS-422	RS-485
1	NC	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	NC	RX-	NC
5	Ground	Ground	Ground
6	NC	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	NC	NC	NC

2.5.4 LVDS Display Connector (P13, P14)





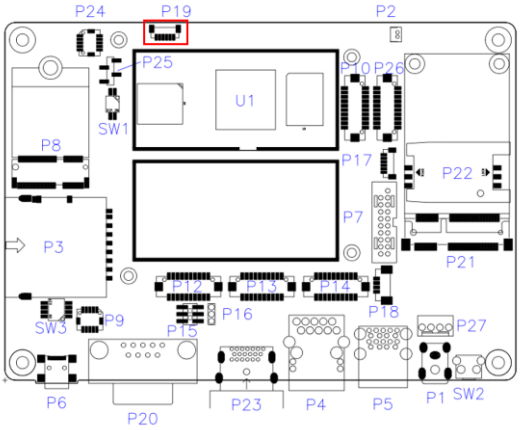
P13：

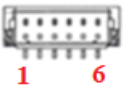
Pin	Assigment	Pin	Assigment
1	LCD0_TX0_P	2	LCD0_TX0_N
3	Ground	4	Ground
5	LCD0_TX1_P	6	LCD0_TX1_N
7	Ground	8	LCD_VDD
9	LCD0_TX3_P	10	LCD0_TX3_N
11	LCD0_TX2_P	12	LCD0_TX2_N
13	Ground	14	Ground
15	LCD0_CLK_P	16	LCD0_CLK_N
17	BTL_PWM	18	LCD_VDD
19	BKLT_VCC	20	BKLT_VCC

P14：

Pin	Assigment	Pin	Assigment
1	LCD1_TX0_P	2	LCD1_TX0_N
3	Ground	4	Ground
5	LCD1_TX1_P	6	LCD1_TX1_N
7	Ground	8	LCD_VDD
9	LCD1_TX3_P	10	LCD1_TX3_N
11	LCD1_TX2_P	12	LCD1_TX2_N
13	Ground	14	Ground
15	LCD1_CLK_P	16	LCD1_CLK_N
17	BTL_PWM	18	LCD_VDD
19	BKLT_VCC	20	BKLT_VCC

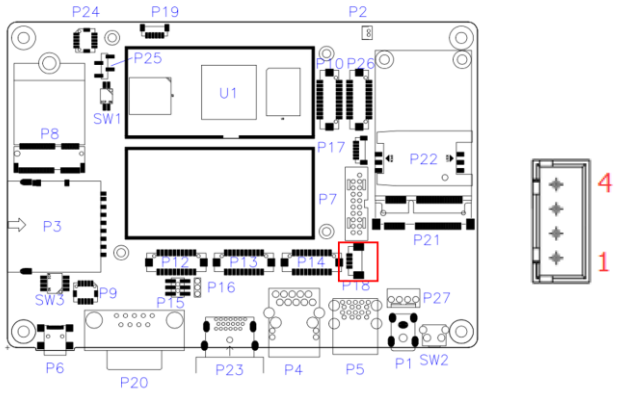
2.5.5 RS232 COM Port Connector (P19)





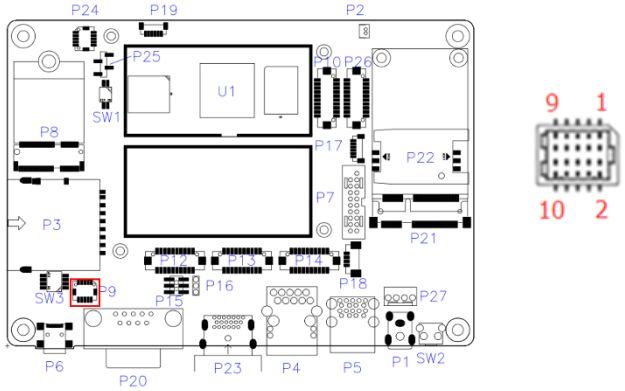
Pin	Assigment	Pin	Assigment
1	COM5_TXD	4	COM3_TXD
2	COM5_RXD	5	COM3_RXD
3	Ground	6	Ground

2.5.6 LVDS Backlight Control Connector (P18)



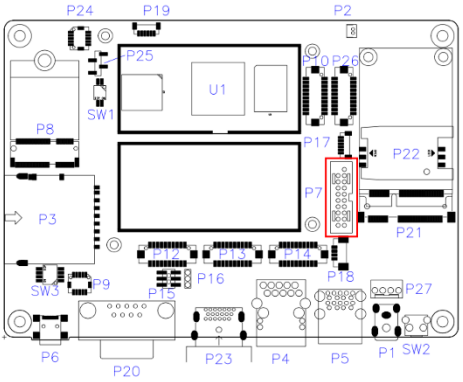
Pin	Assigment	Pin	Assigment
1	BKLT_VCC	3	LCD_BKLT_PWM
2	LCD_BKLT_EN	4	Ground

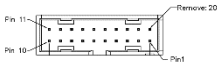
2.5.7 Audio Line-In & Line-Out Connector (P9)



Pin	Assigment	Pin	Assigment
1	NC	2	Ground
3	LINE_IN_R	4	Ground
5	LINE_IN_L	6	Ground
7	Ground	8	LINE_OUT_L
9	Ground	10	LINE_OUT_R

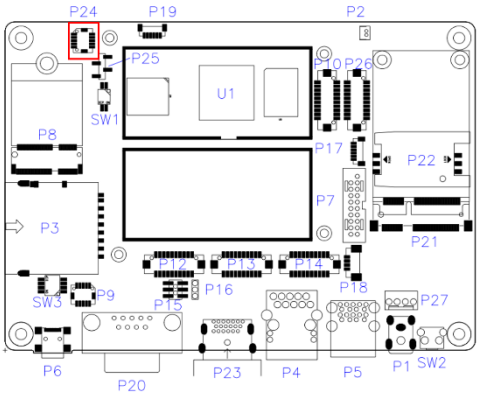
2.5.8 Internal USB3.0 Connector (P7)

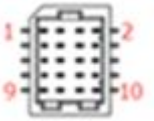




Pin	Assigment	Pin	Assigment
1	VCC(900mA)	2	P1_SSRX-
3	P1_SSRX+	4	GND
5	P1_SSTX-	6	P1_SSTX+
7	GND	8	P1_U2_D-
9	P1_U2_D+	10	NC
11	P2_U2_D+	12	P2_U2_D-
13	GND	14	P2_SSTX+
15	P2_SSTX-	16	GND
17	P2_SSRX+	18	P2_SSRX-
19	VCC(900mA)	X	

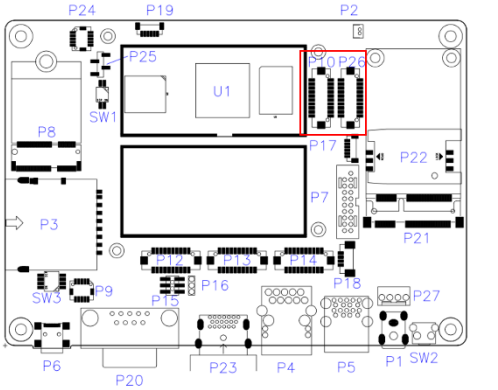
2.5.9 Digital I/O (GPIO) Connector (P24)

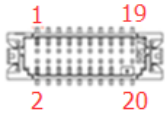




Pin	Assigment	Pin	Assigment
1	3.3V	2	DIO5 (gpio74)
3	DIO1 (gpio146)	4	DIO6 (gpio75)
5	DIO2 (gpio147)	6	DIO7 (gpio76)
7	DIO3 (gpio148)	8	DIO8 (gpio77)
9	DIO4 (gpio149)	10	Ground

2.5.10 MIPI-CSI Connector (P10, P26)





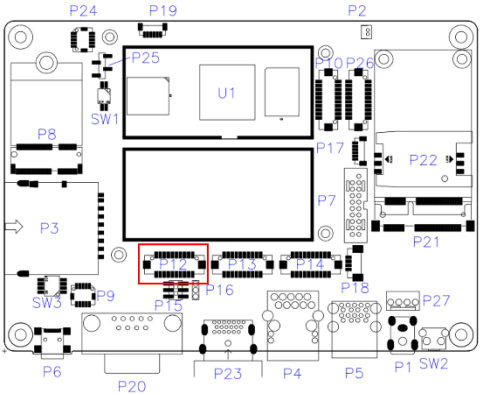
P10:

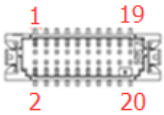
Pin	Assigment	Pin	Assigment
1	MIPI_CSI1_CKP	2	MIPI_CSI1_CKN
3	MIPI_CSI1_DP0	4	MIPI_CSI1_DN0
5	MIPI_CSI1_DP1	6	MIPI_CSI1_DN1
7	MIPI_CSI1_DP2	8	MIPI_CSI1_DN2
9	MIPI_CSI1_DP3	10	MIPI_CSI1_DN3
11	GND	12	GND
13	CSI1_SCL	14	CSI1_SDA
15	CSI1_RST_B	16	VDD_2V8
17	CSI1_PWEN_B	18	VDD_1V8
19	CSI1_MCLK	20	GND

P26:

Pin	Assigment	Pin	Assigment
1	MIPI_CSI2_CKP	2	MIPI_CSI2_CKN
3	MIPI_CSI2_DP0	4	MIPI_CSI2_DN0
5	MIPI_CSI2_DP1	6	MIPI_CSI2_DN1
7	MIPI_CSI2_DP2	8	MIPI_CSI2_DN2
9	MIPI_CSI2_DP3	10	MIPI_CSI2_DN3
11	GND	12	GND
13	CSI2_SCL	14	CSI2_SDA
15	CSI2_RST_B	16	VDD_2V8
17	CSI2_PWEN_B	18	VDD_1V8
19	CSI2_MCLK	20	GND

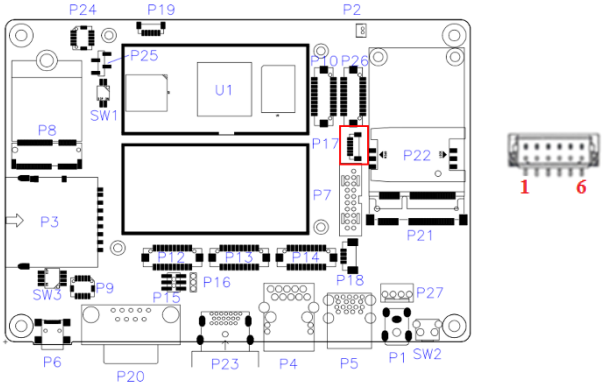
2.5.11 MIPI-DSI Connector (P12)





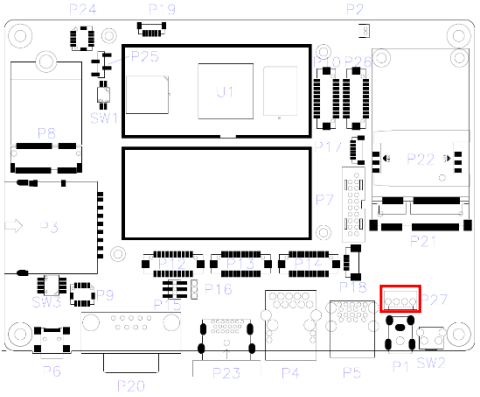
Pin	Assigment	Pin	Assigment
1	MIPI_DSI_CKP	2	MIPI_DSI_CKN
3	GND	4	GND
5	MIPI_DSI_DP0	6	MIPI_DSI_DN0
7	GND	8	VCC_LCD
9	MIPI_DSI_DP1	10	MIPI_DSI_DN1
11	MIPI_DSI_DP2	12	MIPI_DSI_DN2
13	HDMI_INT	14	GND
15	MIPI_DSI_DP3	16	MIPI_DSI_DN3
17	I2C2_SCL	18	VCC_LCD
19	I2C2_SDA	20	BKLT

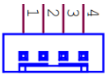
2.5.12 I2C Connector (P17)



Pin	Assigment	Pin	Assigment
1	3V3	4	I2C2_SCL
2	TP_INT_B	5	I2C2_SDA
3	TP_RST_B	6	GND

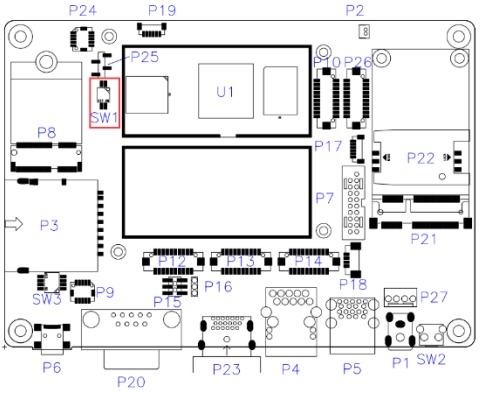
2.5.13 DC Power Input (P27)





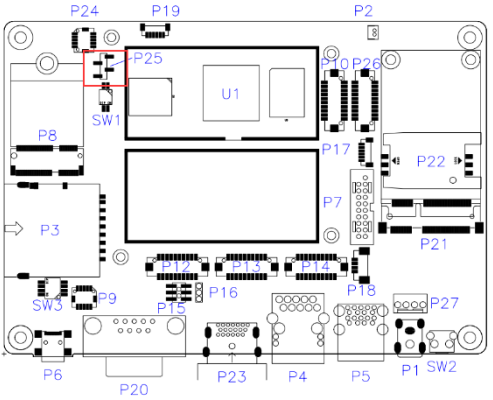
Pin	Assignment	Pin	Assignment
1	GND	2	GND
3	VIN	4	VIN

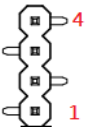
2.5.14 Boot mode select (SW1 Factory use only)



Panel Type	1-4	2-3
OTG Update Mode	On	N/A
(Default) 1st : Boot from SD 2nd : Boot from eMMC	Off	Off
Boot form SD only	Off	On

2.5.15 RS232 Debug Port (P25)





Pin	Assigment
1	Debug_RX
2	Debug_TX
3	GND
4	NC

Chapter 3
Software Setup

This chapter introduces installation of the following drivers:

- Make a recovery SD card (for advanced users only)
- Display parameter setting in kernel

3.1 Make a Recovery SD Card

Note: This is for advanced users who has IBASE standard image file only.

In general, ISR301 is preloaded with O.S (Android or Yocto) into eMMC by default. Connect the HDMI cable to ISR301, and the12V-24V power directly.

This chapter guides you to make a recovery boot-up microSD card.

3.1.1 Preparing the Recovery SD Card to Install Linux / Android Image into eMMC

Note: All data in the eMMC will be erased.

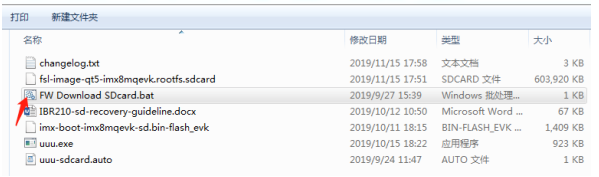
1. System requirements:

   Operating System: Windows 7 or later
   Tool: uuu
   SD card: 4GB or greater in size

2. Insert the SD card to the board (via the P3 connector). Connect the board to the PC through the mini-USB port (via the P6 connector). Change the boot mode to download mode.

3. Boot IBR210 and flash SD via CMD command “uuu.exe uuu-sdcard.auto”

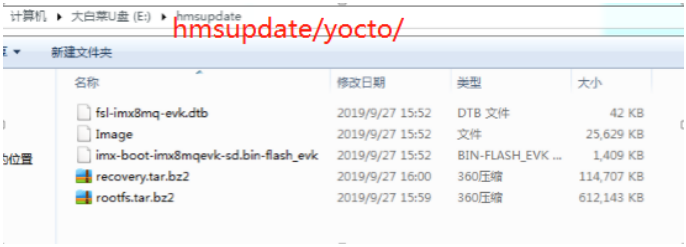
   or double click “FW Download SDcard.bat” (Same way as PCBA update)



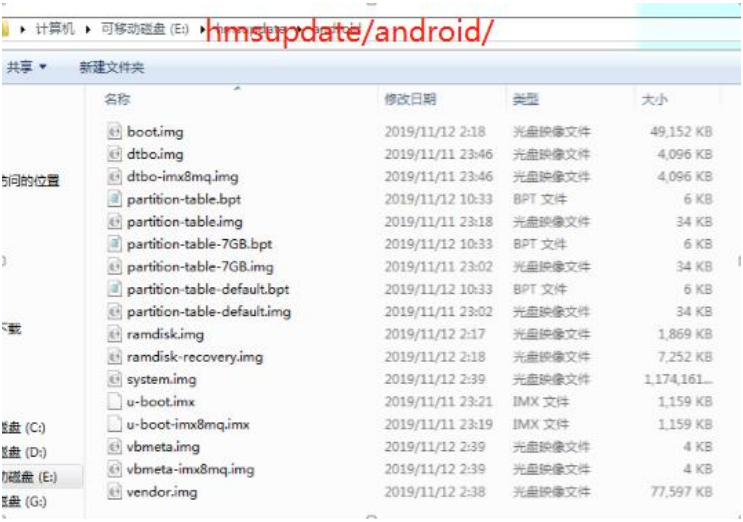
3.1.2 Upgrade Firmware through the Recovery SD Card

1. Put the **recovery files into the USB flash disk (FAT32)**

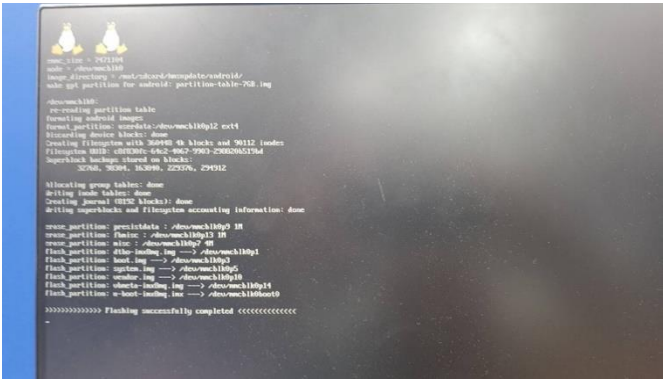
   A) Yocto/Ubuntu: Copy all the recovery files into the PATH: /USB_flash_disk/hmsupdate/yocto/



   B) Android: Copy all the recovery files into the PATH: /USB_flash_disk/hmsupdate/android/



2. Insert (Chapter 3.1.1) SD and (Chapter 3.1.2)USB flash disk into IBR210.
3. Normal boot IBR210 (SW1 Pin1 OFF), start recovery eMMC automatically.
4. The update information will show on HDMI.



5. Show “Flashing successfully completed”, then power off and remove recovery SD and USB flash disk.

3.2 Display Parameter Setting in Kernel(Feature not ready yet)

*IBR210 supports HDMI output by default.

1. If you use HDMI for display, run the command below.

```
/home/root/display_config/config_displag_mode.sh 1
```

2. If you use LVDS 21.5” for display, run the command below.

```
/home/root/display_config/config_displag_mode.sh 4
```

Note: Script content may be changed by different LVDS models.

This page was intentionally left blank.

Chapter 4
BSP Source Guide

This chapter is dedicated for advanced software engineers to build BSP source. The topics covered in this chapter are as follows:

- Preparation
- Installing Toolchain
- Building release
- Installing release to board

4.1 Building BSP Source

4.1.1 Preparation

The suggested Host platform is Ubuntu 12.04 and 14.04 in 32-bit and 64-bit versions.

1. Install necessary packages before building:

```
sudo apt-get install gawk wget Git-core diffstat unzip texinfo
sudo apt-get install gcc-multilib build-essential chrpath socat
sudo add-apt-repository ppa:git-core/ppa
sudo apt-get update
sudo apt-get install git
sudo apt-get install texinfo
```

2. Decompress the IBR210 source file (e.g. ibr210-bsp.tar.bz2) into "/home/" folder.

4.1.2 Installing Toolchain

Decompress Toolchain **poky.tar** into directory "/opt".

```
fsl-imx-wayland-glibc-x86_64-meta-toolchain-cortexa9hf-neon-toolchain-4.14-sumo.sh
```

4.1.3 Building release

For Yocto/uBuntu/Debian

```
cd /home/bsp-folder

./build-bsp-4.14.sh
```

For Android

```
cd /home/bsp-folder
source build/envsetup.sh
lunch evk_8mq-userdebug
make  ANDROID_COMPILE_WITH_JACK=false
```

4.1.4 Installing release to board

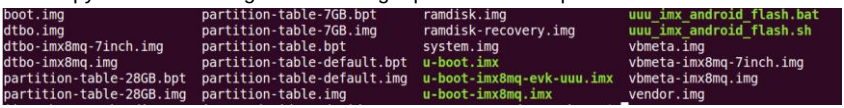
cd /home/bsp-folder

for yocto/Ubuntu/debian

1. cp file in release/ to windows
2. set board to download mode, and connect otg to usb
3. run uuu.exe uuu.auto

for android9

1. copy out the following file in out/target/product/imx8mq/



2. set board to download mode, and connect otg to usb
3. run :

For 7GByte emmc

uuu_imx_android_flash.bat -f imx8mq -tos -c 7 –e

For 16GByte emmc

uuu_imx_android_flash.bat -f imx8mq -tos –e

This page was intentionally left blank.

Appendix

This section provides reference code information.

A. How to Use GPIO in Linux

```
# GPIO Value Rule : gpioX_N >> 32*(X-1)+N
# Take gpio5_18  as example, export value should be 32*(5-1)+18=146

# GPIO example 1: Output
echo 32 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio146/direction
echo 0 > /sys/class/gpio/gpio146/value
echo 1 > /sys/class/gpio/gpio146/value

# GPIO example 2: Input
echo 32 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio146/direction
cat /sys/class/gpio/gpio146/value
```

B. How to Use Watchdog in Linux

```
// create fd
int fd;
//open watchdog device
fd = open("/dev/watchdog", O_WRONLY);
//get watchdog support
ioctl(fd, WDIOC_GETSUPPORT, &ident);
//get watchdog status
ioctl(fd, WDIOC_GETSTATUS, &status);
//get watchdog timeout
ioctl(fd, WDIOC_GETTIMEOUT, &timeout_val);
//set watchdog timeout
ioctl(fd, WDIOC_SETTIMEOUT, &timeout_val);
//feed dog
ioctl(fd, WDIOC_KEEPALIVE, &dummy);
```

C. eMMC Test

Note: This operation may damage the data stored in eMMC flash. Before starting the test, make sure there is no critical data in the eMMC flash being used.

- **Read, write, and check**

```
MOUNT_POINT_STR="/var"


#create data file
dd if=/dev/urandom of=/tmp/data1 bs=1024k count=10
#write data to emmc
dd if=/tmp/data1 of=$MOUNT_POINT_STR/data2 bs=1024k count=10
#read data2, and compare  with data1
cmp $MOUNT_POINT_STR/data2 /tmp/data1
```

- **eMMC speed test**

```
MOUNT_POINT_STR="/var"


#get  emmc write speed"
time dd if=/dev/urandom of=$MOUNT_POINT_STR/test bs=1024k count=10
# clean caches
echo 3 > /proc/sys/vm/drop_caches
#get  emmc read speed"
time dd if=$MOUNT_POINT_STR/test of=/dev/null bs=1024k count=10
```

D. USB (flash disk) Test

Insert the USB flash disk. Make sure it is in IBR210 device list.

Note: This operation may damage the data stored in the USB flash disk. Before starting the test, make sure there is no critical data in the eMMC flash being used.

- **Read, write, and check**

```
USB_DIR="/run/media/mmcblk1p1"
#create data file
dd if=/dev/urandom of=/var/data1 bs=1024k count=100
#write data to usb flash disk
dd if=/var/data1 of=$USB_DIR/data2 bs=1024k count=100
#read data2, and compare  with data1
cmp $USB_DIR/data2 /var/data1
```

- **USB speed test**

```
USB_DIR="/run/media/mmcblk1p1"
# usb write speed
dd if=/dev/zero of=$BASIC_DIR/$i/test bs=1M count=1000 oflag=nocache

# usb read speed
dd if=$BASIC_DIR/$i/test of=/dev/null bs=1M oflag=nocache
```

E. SD Card Test

When IBR210 is booted from eMMC, SD card is “/dev/mmcblk1” and able to see by “ls /dev/mmcblk1*”

Command:

```
/dev/mmcblk1  /dev/mmcblk1p2 /dev/mmcblk1p4 /dev/mmcblk1p5 /dev/mmcblk1p6
```

When IBR210 is booted from SD card, replace test pattern “/dev/mmcblk1” to “/dev/mmcblk0”.

Note: This operation may damage the data stored the SD card. Before starting the test, make sure there is no critical data in the eMMC flash being used.

- **Read, write, and check**

```
SD_DIR="/run/media/mmcblk1"
#create data file
dd if=/dev/urandom of=/var/data1 bs=1024k count=100
#write data to SD card
dd if=/var/data1 of=$ SD_DIR/data2 bs=1024k count=100
#read data2, and compare  with data1
cmp $SD_DIR/data2 /var/data1
```

- **SD card speed test**

```
SD_DIR="/run/media/mmcblk1"

# SD write speed
dd if=/dev/zero of=$SD_DIR/test bs=1M count=1000 oflag=nocache

# SD read speed
dd if=$SD_DIR/test of=/dev/null bs=1M oflag=nocache
```

F. RS-232 Test

```
//open ttymxc1
fd  = open(/dev/ttymxc1,O_RDWR );

//set speed
tcgetattr(fd, &opt);
cfsetispeed(&opt, speed);
cfsetospeed(&opt, speed);
tcsetattr(fd, TCSANOW, &opt)

//get_speed
tcgetattr(fd, &opt);
speed = cfgetispeed(&opt);

//set_parity
// options.c_cflag
options.c_cflag &= ~CSIZE;
options.c_cflag &= ~CSIZE;
options.c_lflag  &= ~(ICANON | ECHO | ECHOE | ISIG);  /*Input*/
options.c_oflag  &= ~OPOST;   /*Output*/
//options.c_cc
options.c_cc[VTIME] = 150;
options.c_cc[VMIN] = 0;
#set parity
tcsetattr(fd, TCSANOW, &options)

//write ttymxc1
write(fd, write_buf, sizeof(write_buf));

//read ttymxc1
read(fd, read_buf, sizeof(read_buf)))
```

G. RS-485 Test

```
//open ttymxc1
fd  = open(/dev/ttymxc1,O_RDWR );

//set speed
tcgetattr(fd, &opt);
cfsetispeed(&opt, speed);
cfsetospeed(&opt, speed);
tcsetattr(fd, TCSANOW, &opt

//get_speed
tcgetattr(fd, &opt);
speed = cfgetispeed(&opt);

//set_parity
// options.c_cflag
options.c_cflag &= ~CSIZE;
options.c_cflag &= ~CSIZE;
options.c_lflag  &= ~(ICANON | ECHO | ECHOE | ISIG);  /*Input*/
options.c_oflag  &= ~OPOST;   /*Output*/
//options.c_cc
options.c_cc[VTIME] = 150;
options.c_cc[VMIN] = 0;
#set parity
tcsetattr(fd, TCSANOW, &options)

//write ttymxc1
write(fd, write_buf, sizeof(write_buf));

//read ttymxc1
read(fd, read_buf, sizeof(read_buf)))
```

H. Audio Test

```
// play mp3 by audio (ALC5640)
gplay-1.0 /home/root/ testscript/audio/a.mp3  --audio-sink=”alsasink –device=hw:1”
// record mp3 by audio (ALC5640)
arecord -f cd $basepath/b.mp3 -D plughw:1,0
```

Note: for Android, please use apk to test.

I. Ethernet Test

- **Ethernet Ping test**

```
#ping server 192.168.1.123
ping -c 20 192.168.1.123 >/tmp/ethernet_ping.txt
```

- **Ethernet TCP test**

```
#server 192.168.1.123 run command  “iperf3 -s”
#communicate with server 192.168.1.123 in tcp mode by iperf3
iperf3 -c 192.168.1.123 -i 1 -t 20 -w 32M -P 4
```

- **Ethernet UDP test**

```
#server 192.168.1.123 run command  “iperf3 -s”
#communicate with server 192.168.1.123 in udp mode by iperf3
iperf3 -c $SERVER_IP -u -i 1 -b 200M
```

J. LVDS Test

```
//Open the file for reading and writing
framebuffer_fd = open("/dev/fb0", O_RDWR);

// Get fixed screen information
ioctl(framebuffer_fd, FBIOGET_FSCREENINFO, &finfo)

// Get variable screen information
ioctl(framebuffer_fd, FBIOGET_VSCREENINFO, &vinfo)

// Figure out the size of the screen in bytes
screensize = vinfo.xres * vinfo.yres * vinfo.bits_per_pixel / 8;

// Map the device to memory
fbp = (char *)mmap(0, screensize, PROT_READ | PROT_WRITE, MAP_SHARED, framebuffer_fd,
0);

// Figure out where in memory to put the pixel
memset(fbp, 0x00,screensize);

//draw point by fbp
long int location = 0;
location = (x+g_xoffset) * (g_bits_per_pixel/8) +
 (y+g_yoffset) * g_line_length;
*(fbp + location + 0) = color_b;
*(fbp + location + 1) = color_g;
*(fbp + location + 2) = color_r;

//close framebuffer fd
close(framebuffer_fd);
```

Note: Android is not supported.

K. HDMI Test

- **HDMI display test**

```
//Open the file for reading and writing
framebuffer_fd = open("/dev/fb2", O_RDWR);

// Get fixed screen information
ioctl(framebuffer_fd, FBIOGET_FSCREENINFO, &finfo)

// Get variable screen information
ioctl(framebuffer_fd, FBIOGET_VSCREENINFO, &vinfo)

// Figure out the size of the screen in bytes
screensize = vinfo.xres * vinfo.yres * vinfo.bits_per_pixel / 8;

// Map the device to memory
fbp = (char *)mmap(0, screensize, PROT_READ | PROT_WRITE, MAP_SHARED,
framebuffer_fd, 0);

// Figure out where in memory to put the pixel
memset(fbp, 0x00,screensize);

//draw point by fbp
long int location = 0;
location = (x+g_xoffset) * (g_bits_per_pixel/8) +
 (y+g_yoffset) * g_line_length;
*(fbp + location + 0) = color_b;
*(fbp + location + 1) = color_g;
*(fbp + location + 2) = color_r;

//close framebuffer fd
close(framebuffer_fd);
```

- **HDMI audio test**

```
#enable hdmi audio
echo 0 > /sys/class/graphics/fb2/blank
#play wav file by  hdmi audio
aplay /home/root/testscript/hdmi/1K.wav -D plughw:0,0
```

L. 3G Test

- **Checking 3G state**

```
#Check UC20 module state and sim state
cat /dev/ttyUSB4  &
```

- **Testing 3G**

```
# the command will connect 3g to network
# make sure that the simcard is inserted right, and ANT connected
pppd call quectel-ppp

echo "ping www.baidu.com to make sure the network ok"
ping www.baidu.com
```

Note: Since Android includes 3G config in setting, this portion is not suited for the Android version.

M. Onboard Connector Types

Function	Connector Name	Onboard Type	Compatible Mating Type for Reference
LVDS Display Connector	P13, P14	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
UART Connector	P19	TechBest WT02M-30002-06132	JST SHR-03V-S-B
LVDS Backlight Control Connector	P18	TechBest 1024041008	Molex 51021-0400
Audio Line-In & Line-Out Connector	P9	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H
Internal USB3.0 Connector	P7	Pinrex 52X-40-20GU52	TBD
Digital I/O (GPIO) Connector	P24	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H
MIPI-CSI Connector	P10, P26	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
MIPI-DSI	P12	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
I^2C Connector	P26	TechBest WT02M-30002-06132	JST SHR-03V-S-B
Internal DC Power Input	P27	TechBest 2542-WS-04-LF	

Connector types may be subject to change without prior notice.