

ISR301
Ruggedized Embedded System
with NXP ARM[®] Cortex[®]
A53 i.MX8M Quad SoC

User's Manual

Document Version 1.2
(Oct. 2020)



Copyright

© 2020 IBASE Technology, Inc. All rights reserved.

No part of this publication may be reproduced, copied, stored in a retrieval system, translated into any language or transmitted in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written consent of IBASE Technology, Inc. (hereinafter referred to as “IBASE”).

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 misuse or inability to use the product or the information contained herein, and 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

The product described in this manual complies with all applicable European Union (CE) directives if it has a CE marking. For systems to remain CE compliant, only CE-compliant parts may be used. Maintaining CE compliance also requires proper cable and cabling techniques.

FCC

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 complies with the current RoHS directives restricting the use of the following substances in concentrations not to exceed 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 following safety information before using this device.

Setting up your system:

- Put the device horizontally on a stable and solid surface.
- Do not use this product near water or any heated source.
- Leave plenty of space around the device and do not block the ventilation openings. Never drop or insert any objects of any kind into the openings.
- Use this product in environments with ambient temperatures between 0°C and 60°C.

Care during use:

- Do not place heavy objects on the top of the device.
- Make sure to connect the correct voltage to the device. Failure to supply the correct voltage could damage the unit.
- Do not walk on the power cord or allow anything to rest on it.
- If you use an extension cord, make sure the total ampere rating of all devices plugged into the extension cord does not exceed the cord's ampere rating.
- Do not spill water or any other liquids on your device.
- Always unplug the power cord from the wall outlet before cleaning the device.
- Only use neutral cleaning agents to clean the device.
- Vacuum dust and particles from the vents by using a computer vacuum cleaner.

Product Disassembly

Do not try to repair, disassemble, or make modifications to the device. Doing so will void the warranty and may result in damage to the product or personal injury.



CAUTION

Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries by observing local regulations.

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 3rd-party parts that are not manufactured by IBASE, such as CPU, CPU cooler, memory, storage devices, power adaptor, display panel and touch screen.
- * 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 encounter any technical problems and require assistance from your distributor or sales representative, please prepare and send the following information:
 - Product model name
 - Product serial number
 - Detailed description of the problem
 - Error messages in text or screenshots if any
 - The arrangement of the peripherals
 - Software used (such as OS and application software)
3. If repair service is required, please 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

Compliance.....	iii
Important Safety Information	iv
CAUTION	iv
Warranty Policy	v
Technical Support & Services	v
Chapter 1 General Information	1
1.1 Introduction.....	2
1.2 Features	2
1.3 Packing List	2
1.4 Specifications	3
1.5 Product View	5
1.6 Dimensions.....	6
Chapter 2 Hardware Configuration	7
2.1 Installations.....	8
2.1.1 Mini-PCIe & M.2 Cards Installation	8
2.1.2 WiFi / LTE Antenna Installation	9
2.1.3 Mounting Installation.....	9
2.1.4 RS-232/422/485 COM Port.....	10
2.1.5 COM1 & COM2 Port	11
2.1.6 GPIO Port	11
2.2 Setting the Jumpers.....	12
2.2.1 How to Set Jumpers	12
2.3 Jumper & Connector Locations on Motherboard.....	13
2.4 Jumper & Connectors Quick Reference	14
2.4.1 RTC Lithium Cell Connector (P2)	15
2.4.2 RS-232/422/485 Selection (SW3).....	16
2.4.3 RS-232/422/485 COM Port (P20).....	17
2.4.4 LVDS Display Connector (P13, P14).....	18
2.4.5 RS232 COM Port Connector (P19)	19
2.4.6 LVDS Backlight Control Connector (P18).....	20
2.4.7 Audio Line-In & Line-Out Connector (P9)	21
2.4.8 Internal USB3.0 Connector (P7)	22
2.4.9 Digital I/O (GPIO) Connector (P24)	23

2.4.10	MIPI-CSI Connector (P10, P26)	24
2.4.11	MIPI-DSI Connector (P12)	25
2.4.12	I ² C Connector (P17).....	26
2.4.13	Boot mode select (SW1 Factory use only)	27
2.4.14	RS232 Debug Port (P25).....	28
Chapter 3 Software Setup.....		29
3.1	Make a Recovery SD Card.....	30
3.1.1	Preparing the Recovery SD card to Install Linux / Android image into eMMC	30
3.1.2	Upgrade Firmware through the Recovery SD Card.....	31
3.2	Display Parameter Setting in Kernel (Feature not ready yet)	33
Chapter 4 BSP Source Guide		34
4.1	Building BSP Source	35
4.1.1	Preparation	35
4.1.2	Installing Toolchain	35
4.1.3	Building release	35
	For Yocto/uBuntu/Debian	35
	./build-bsp-4.14.sh	35
	For Android.....	35
4.1.4	Installing release to board.....	36
.....		37
Appendix		37

Chapter 1

General Information

The information provided in this chapter includes:

- Features
- Packing List
- Specifications
- Product View
- Dimensions

1.1 Introduction

ISR301 is an ARM[®]-based embedded system built with NXP Cortex[®] i.MX8M A53 processor. The device features 2D, 3D graphics and multimedia acceleration, aside from the numerous peripheral support it has that are well suited for industrial applications, including RS-232/422/485, GPIO, USB, USB OTG, LAN, HDMI interface, M.2 E2230 for wireless connectivity and a mini-PCIe expansion slot.



1.2 Features

- NXP ARM[®] Cortex[®] A53 i.MX8M Quad 1.3GHz industrial-grade processor
- Video output through the HDMI interface
- 3GB LPDDR4, 16GB eMMC and SD socket
- Rich I/O interface with RS-232/422/485 GPIO, USB, USB-OTG and Ethernet
- M.2 E2230 & mini-PCIe slots with a SIM socket for wireless connectivity

1.3 Packing List

Your product package should include the items listed below.

- ISR301-Q316I

The user's manual can be downloaded from the IBASE website.

1.4 Specifications

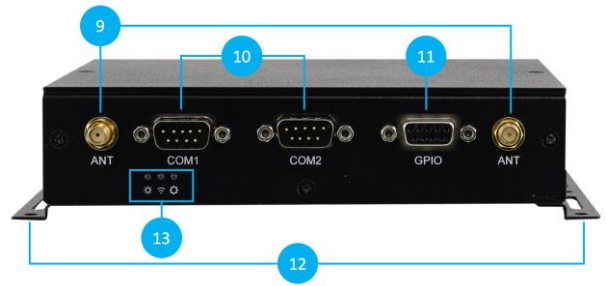
Product Name	ISR301-Q316I
System Motherboard	IBR210-Q316I-01 3.5" SBC
System	
Operating System	<ul style="list-style-type: none"> • Android 9.0 • Yocto v2.5
CPU Type	NXP Cortex™ A53 i.MX8M Quad Core 1.3GHz Industrial-Grade SoC
CPU Speed	1.3 GHz
Memory	<ul style="list-style-type: none"> • System memory: 3GB LPDDR4 • Data Memory: 16GB eMMC
Video Codec	<ul style="list-style-type: none"> • 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 / 3G / LTE module (Optional)
Power Supply	12V-24V DC-In
Watchdog Timer	Yes (256 segments, 0, 1, 2...128 secs)
Dimensions (W x H x D)	162 x 36 x 116 mm (6.38" x 1.42" x 4.57")
RoHS	Yes
Certification	CE, FCC Class B
I/O Ports	
DC Jack	1 x 12V-24V DC jack
Display	1 x HDMI2.0a (up to 4K resolution)
LAN	1 x RJ45 GbE LAN
USB	<ul style="list-style-type: none"> • 2 x USB 3.0 Type A • 1 x USB OTG via mini-USB Type B (Factory use only)
Serial	<ul style="list-style-type: none"> • 1 x RS-232/422/485 COM port (DB9 male connector) • 2 x RS-232 COM ports (DB9 male connector)
Digital IO	8-In/Out
Expansion Slots	<ul style="list-style-type: none"> • 1 x M.2 E-key (2230) • 1 x Mini-PCIe (interrelated with the SIM socket) • 1 x SD socket (UHS-I SDR-104, 104MB/s max.)

Environment	
Operating Temperature	-10 ~ 60 °C (32 ~ 140 °F)
Relative Humidity	10 ~ 90 %, non-condensing

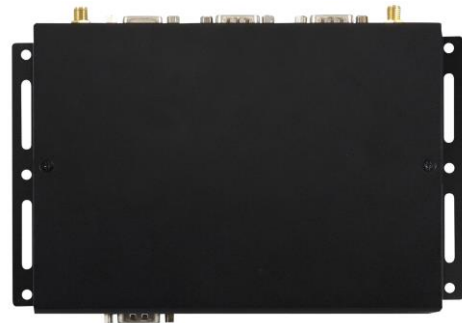
All specifications are subject to change without prior notice.

1.5 Product View

I/O View



Top View



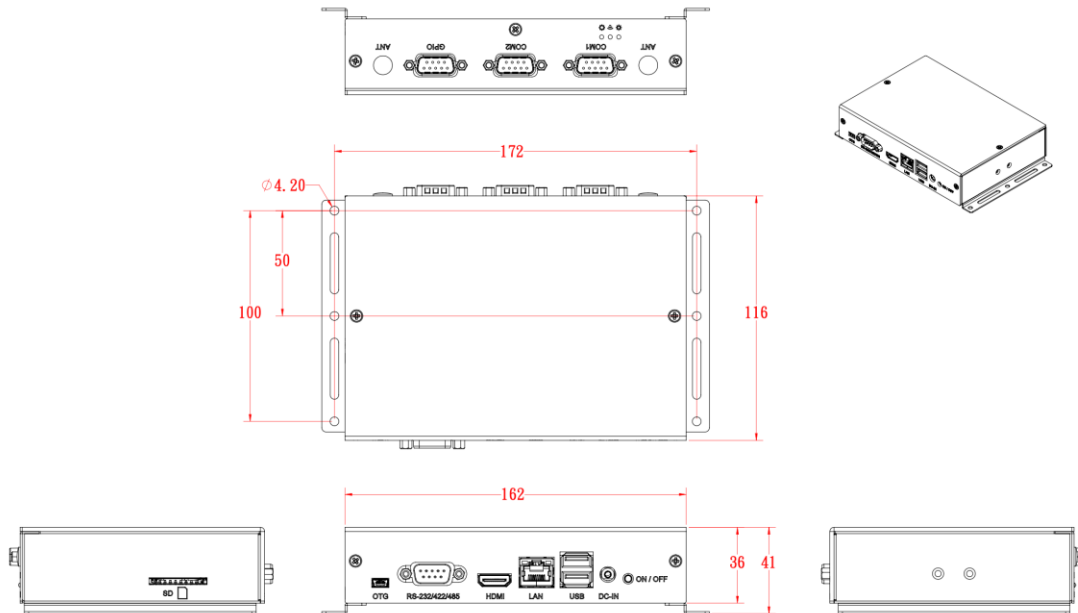
No.	Name	No.	Name
1	USB OTG Port	8	DIN Rail Mounting Holes
2	RS-232/422/485 COM Port	9	Antenna Holes
3	HDMI Port	10	RS-232 COM1 & COM2 Ports
4	GbE LAN Port	11	GPIO Port
5	USB 3.0 Ports	12	Wall Mount Kit
6	DC Jack	13	LED Indicators (from left to right: Power, Wi-Fi status, OS status)
7	On/Off Button	14	SD Card Slot

DIN Rail Mounting View (Optional)



1.6 Dimensions

Unit: mm



Chapter 2

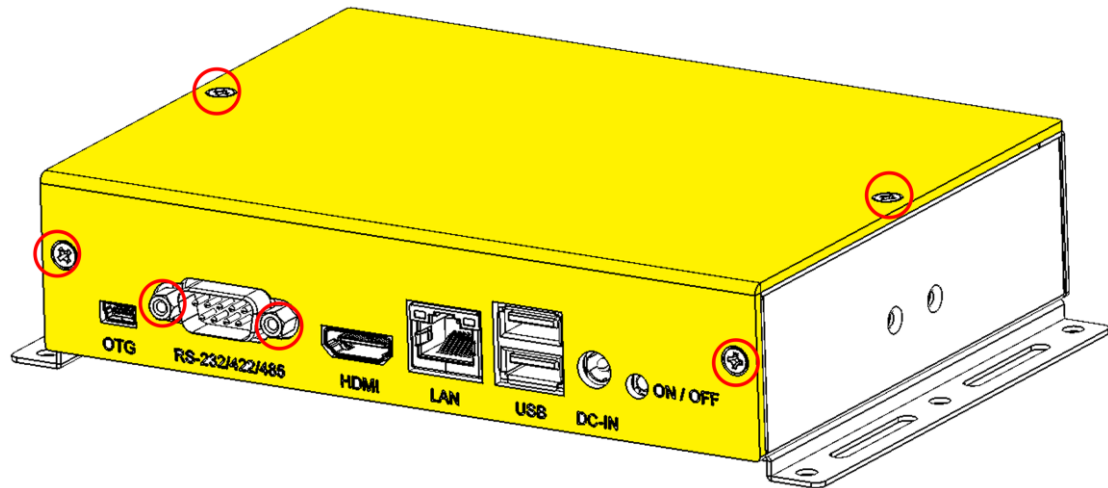
Hardware Configuration

This section contains general information about:

- Installations
- Jumper and connectors

2.1 Installations

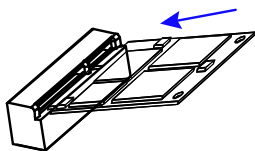
Before installing the M.2, wireless, SIM card or any other module into the device, loosen the 6 screws as shown below to remove the device cover.



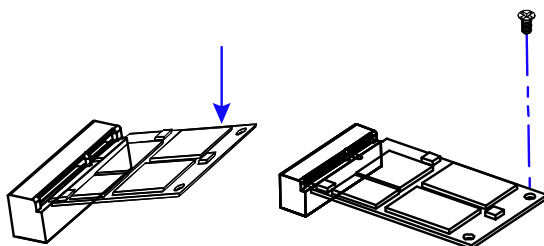
2.1.1 Mini-PCIe & M.2 Cards Installation

To install the mini-PCIe or NGFF M.2 card, remove the device cover, locate the slot in the system, and perform the following steps.

1. To insert the mini-PCIe card, align the keys of the card with those of the mini-PCIe interface. Insert the card slantwise as shown below.
(Insert the M.2 card in the same way as instructed above.)



2. Push the mini-PCIe card downwards as shown below, and fix the card onto the brass standoff with a screw.
(Fix the M.2 card also with one screw.)



2.1.2 WiFi / LTE Antenna Installation

Insert the WiFi / 3G / 4G antenna extension cable through the antenna hole of the front I/O cover and fasten the antenna as shown below. Then apply adhesive around the hex nut behind the front I/O cover to prevent the extension cable from falling off, should the cable becomes loose.

1. Fasten the hex nut and the washer. Then install the antenna.
2. Apply adhesive around here.



Info: The diameter of the nut is around 6.35 mm (0.25"-36UNC).

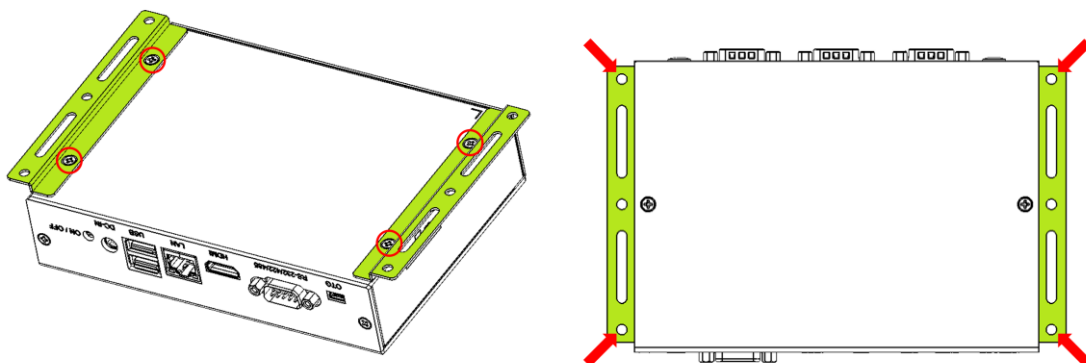
2.1.3 Mounting Installation

Requirements

Before mounting the system, ensure that you have enough room for the power adaptor and signal cable routing, and have good ventilation for the power adaptor. The method of mounting must be able to support weight of the device plus the weight of the suspending cables attached to the system. Use the following methods for mounting the device:

Wall Mounting Installation

1. Turn the device upside down. Attach the wall-mount kit to the device using the supplied 4 screws.
2. Prepare at least 4 screws (M3) to install the device on the wall.



2.1.4 RS-232/422/485 COM Port



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

2.1.5 COM1 & COM2 Port



Pin	Signal Name	Pin	Signal Name
1	NC	6	NC
2	RXD, Receive data	7	NC
3	TXD, Transmit data	8	NC
4	NC	9	NC
5	Ground		

2.1.6 GPIO Port



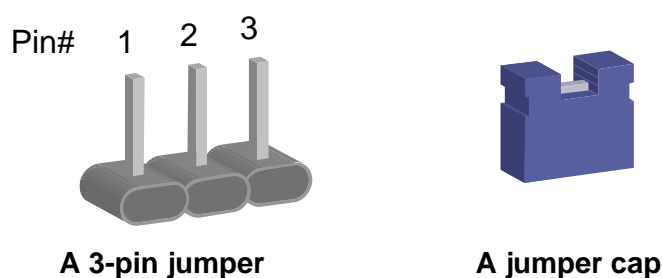
Pin	Signal Name	Pin	Signal Name
1	DIO1	6	DIO5
2	DIO2	7	DIO6
3	DIO3	8	DIO7
4	DIO4	9	DIO8
5	Ground		

2.2 Setting the Jumpers

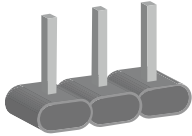
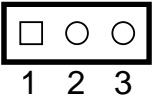
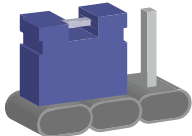
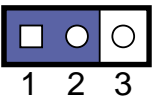
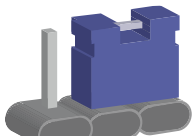
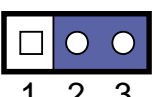
Configure your device by using jumpers to enable the features that you need based on your applications. Contact your supplier if you have doubts about the best configuration for your use.

2.2.1 How to Set Jumpers

Jumpers are short-length conductors consisting of several metal pins with a base mounted on the circuit board. Jumper caps are placed (or removed) on the pins to enable or disable functions or features. If a jumper has 3 pins, you can connect Pin 1 with Pin 2 or Pin 2 with Pin 3 by shorting the jumper.



Refer to the illustration below to set jumpers.

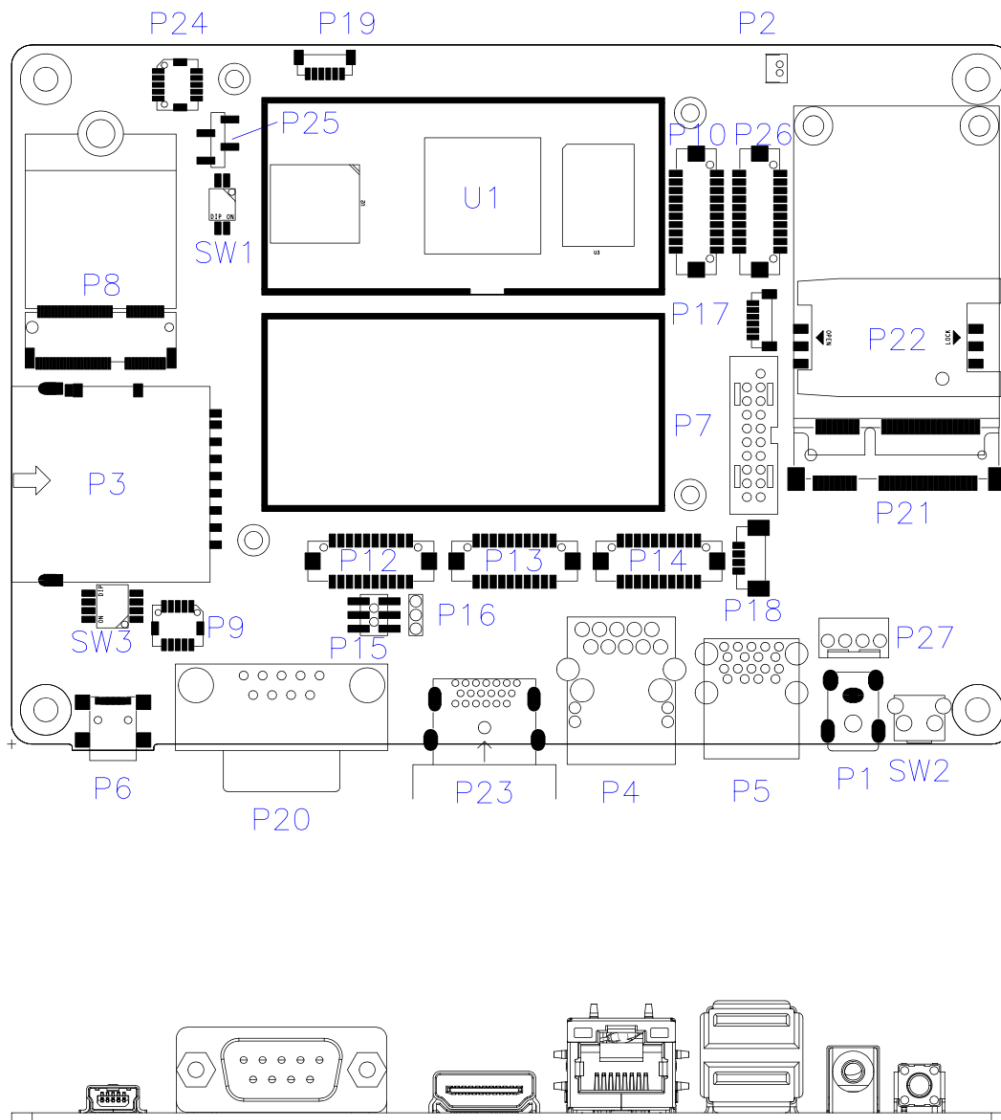
Pin closed	Oblique view	Setting
Open		 1 2 3
1-2		 1 2 3
2-3		 1 2 3

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 Motherboard

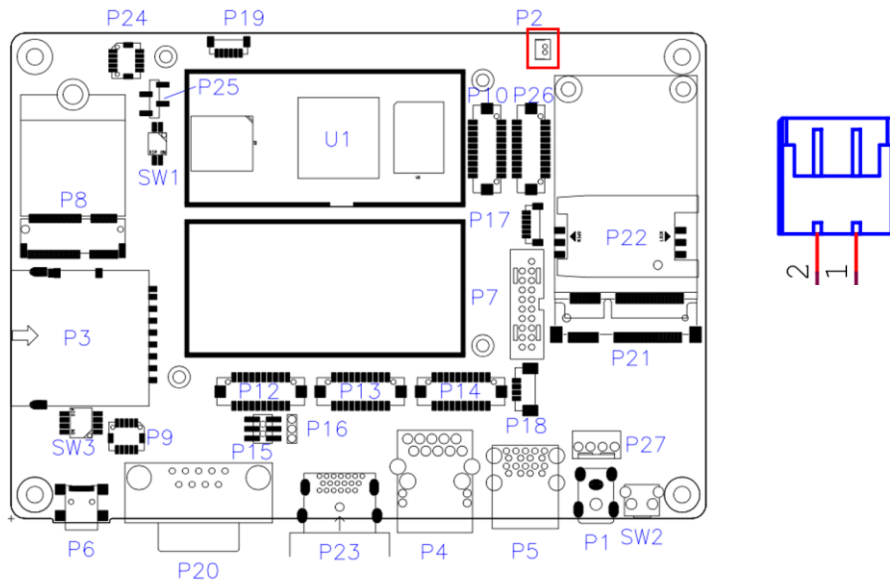
Motherboard: IBR210



2.4 Jumper & Connectors Quick Reference

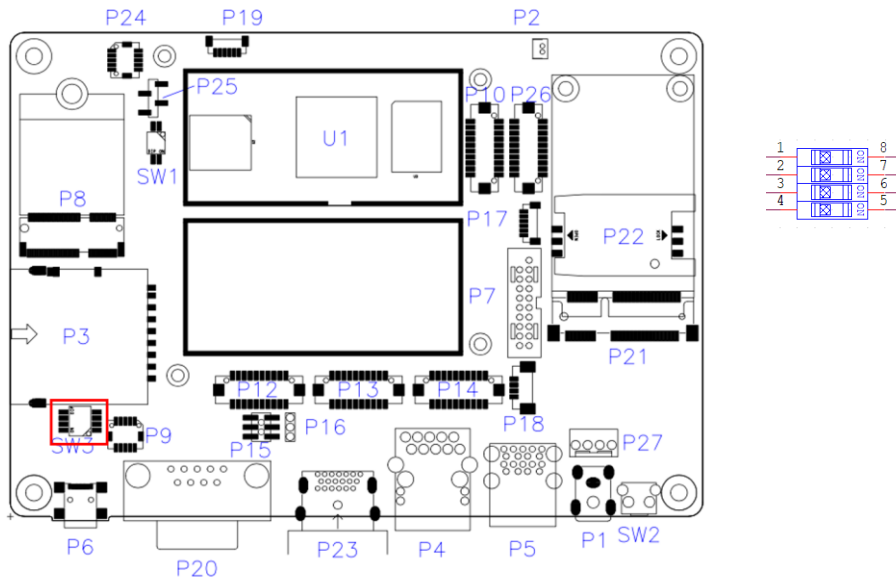
Function	Connector Name	Page
RTC Lithium Cell Connector	P2	15
RS-232/422/485 COM Selection	SW3	16
RS-232/422/485 COM Port	P20	17
LVDS Display Connector	P13, P14	18
RS232 COM Port Connector	P19	19
LVDS Backlight Control Connector	P18	20
Audio Line-In & Line-Out Connector	P9	21
Internal USB3.0 Connector	P7	22
Digital I/O (GPIO) Connector	P24	23
MIPI-CSI Connector	P10, P26	24
MIPI-DSI Connector	P12	25
I ² C Connector	P17	26
DC Power Input	P1, P27	--
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	27, 28

2.4.1 RTC Lithium Cell Connector (P2)



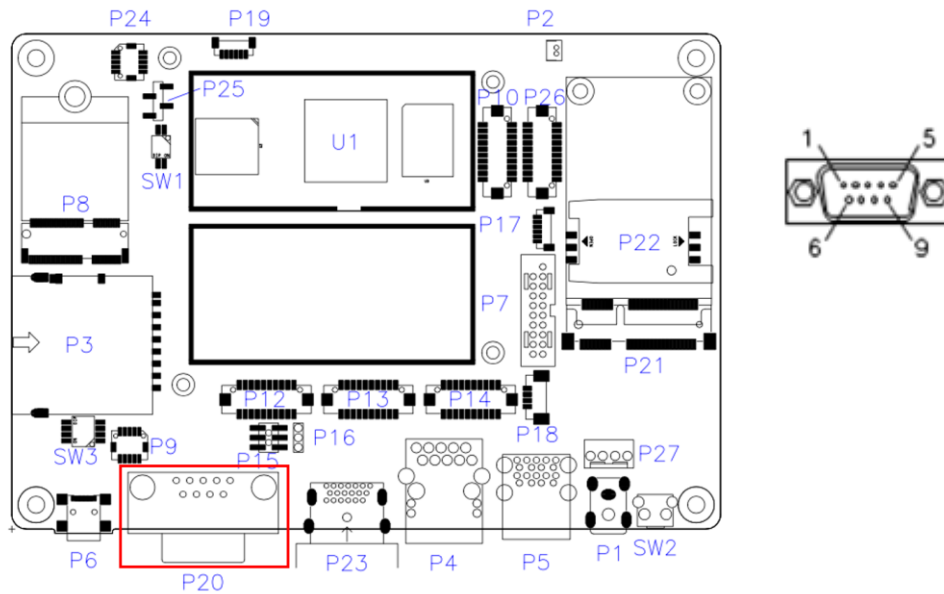
Pin	Signal Name	Pin	Signal Name
1	RTC_VCC	2	Ground

2.4.2 RS-232/422/485 Selection (SW3)



Panel Type	1-8	2-7	3-6	4-5
RS-422 Full Duplex	Off	On	On	On
RS-232 Rx/Tx (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

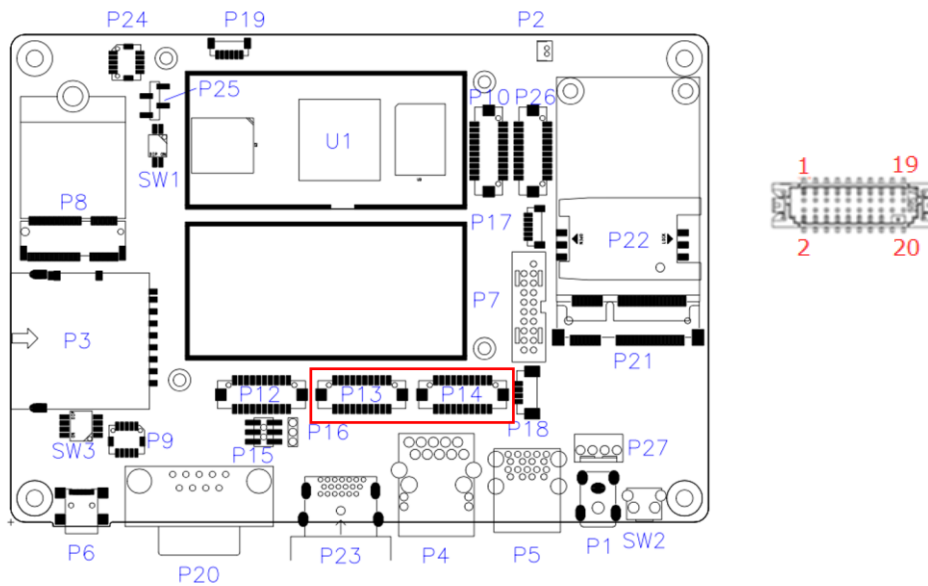
2.4.3 RS-232/422/485 COM Port (P20)



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

2.4.4 LVDS Display Connector (P13, P14)



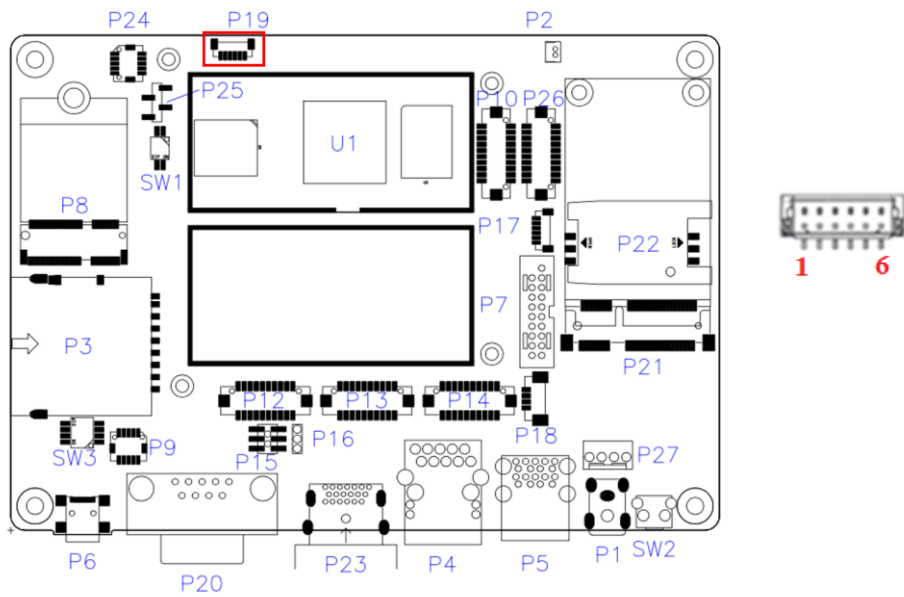
P13 :

Pin	Assignment	Pin	Assignment
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 :

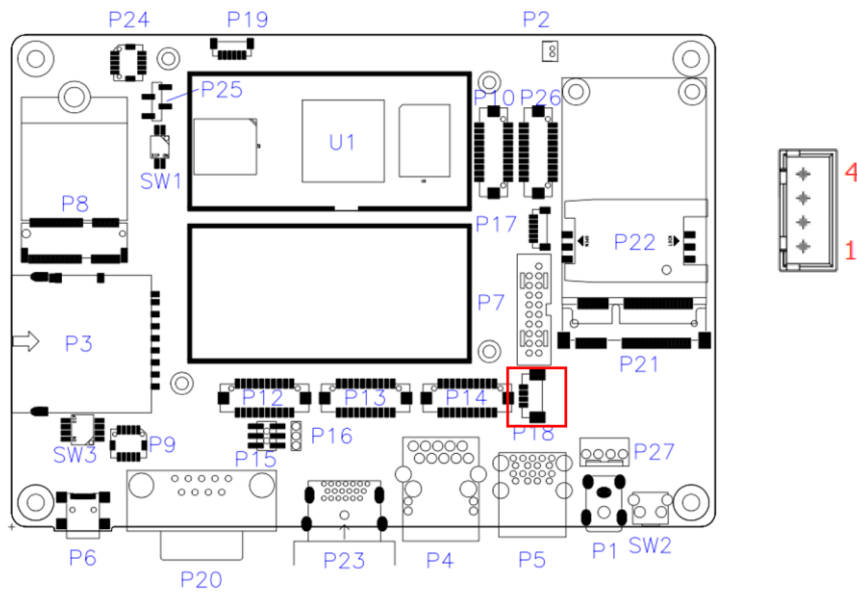
Pin	Assignment	Pin	Assignment
1	LCD1_TX0_P	2	LCD1_TX0_N
3	Ground	4	Ground
5	LCD1_TX1_P	6	LCD1_TX1_N
7	Ground	8	LCD_VDD
9	LCD1_TX3_P	10	LCD1_TX3_N
11	LCD1_TX2_P	12	LCD1_TX2_N
13	Ground	14	Ground
15	LCD1_CLK_P	16	LCD1_CLK_N
17	BTL_PWM	18	LCD_VDD
19	BKLT_VCC	20	BKLT_VCC

2.4.5 RS232 COM Port Connector (P19)



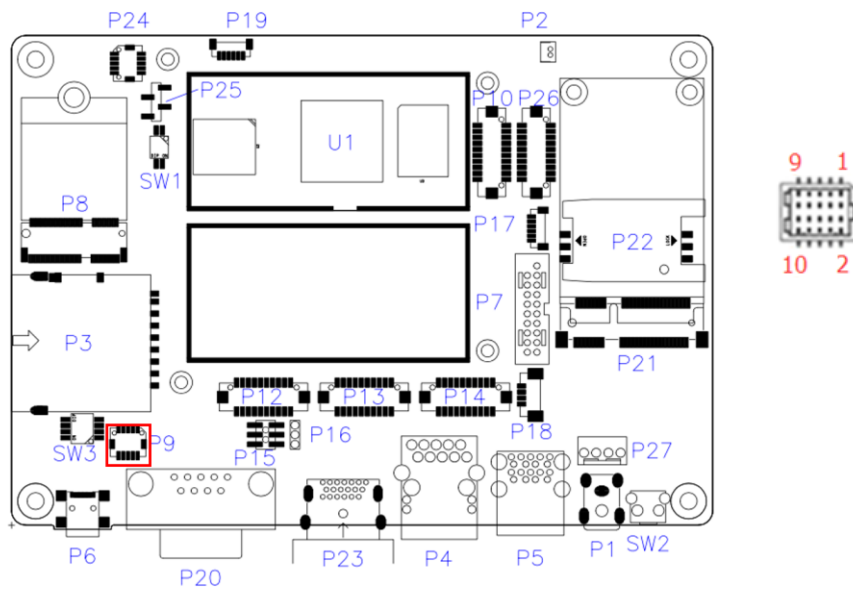
Pin	Assignment	Pin	Assignment
1	COM5_TXD	4	COM3_TXD
2	COM5_RXD	5	COM3_RXD
3	Ground	6	Ground

2.4.6 LVDS Backlight Control Connector (P18)



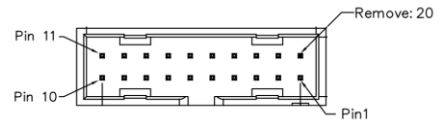
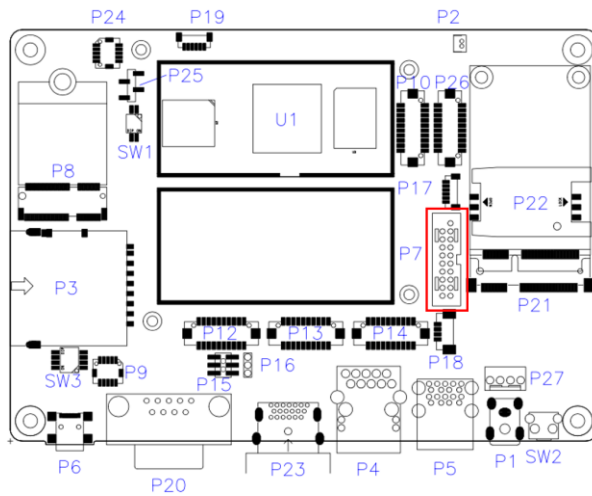
Pin	Assignment	Pin	Assignment
1	BKLT_VCC	3	LCD_BKLT_PWM
2	LCD_BKLT_EN	4	Ground

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



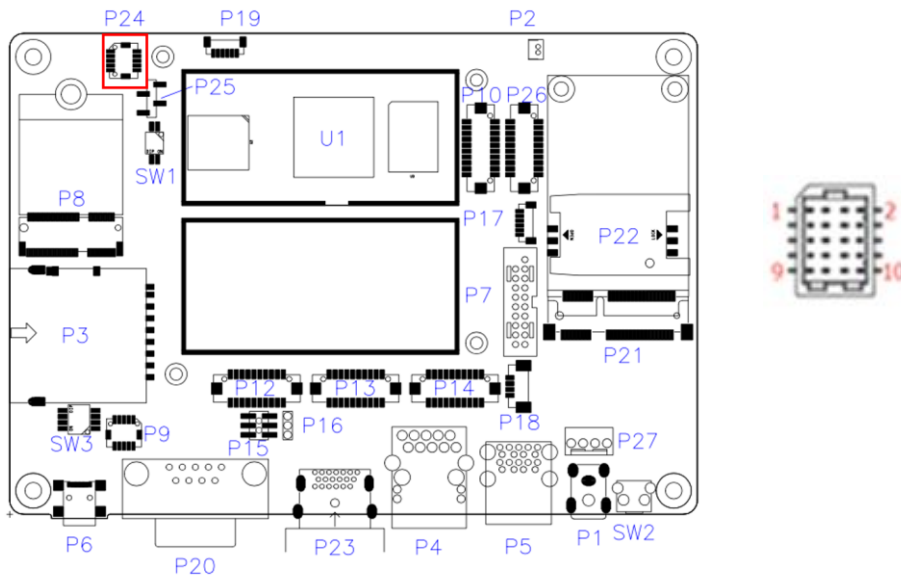
Pin	Assignment	Pin	Assignment
1	NC	2	Ground
3	LINE_IN_R	4	Ground
5	LINE_IN_L	6	Ground
7	Ground	8	LINE_OUT_L
9	Ground	10	LINE_OUT_R

2.4.8 Internal USB3.0 Connector (P7)



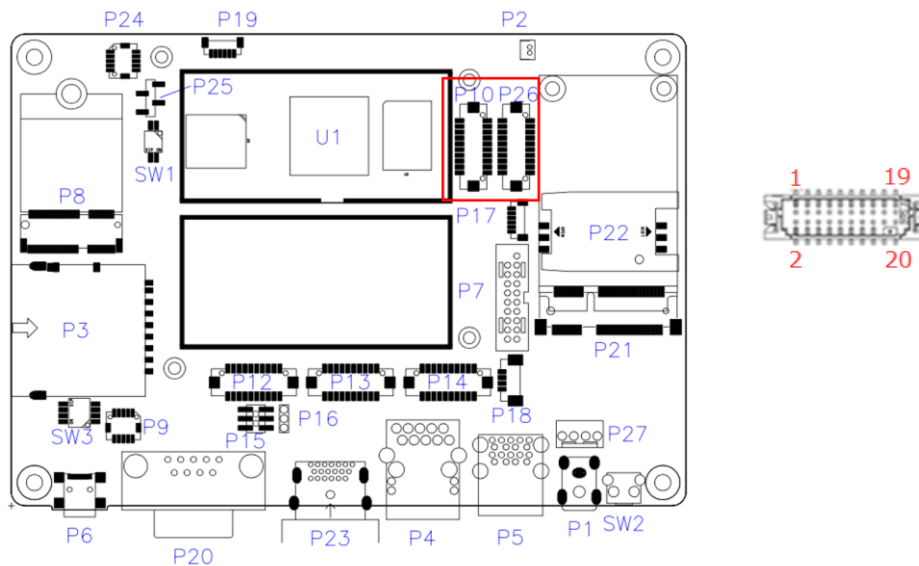
Pin	Assignment	Pin	Assignment
1	VCC(900mA)	2	P1_SSRX-
3	P1_SSRX+	4	GND
5	P1_SSTX-	6	P1_SSTX+
7	GND	8	P1_U2_D-
9	P1_U2_D+	10	NC
11	P2_U2_D+	12	P2_U2_D-
13	GND	14	P2_SSTX+
15	P2_SSTX-	16	GND
17	P2_SSRX+	18	P2_SSRX-
19	VCC(900mA)	X	

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



Pin	Assignment	Pin	Assignment
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

2.4.10 MIPI-CSI Connector (P10, P26)



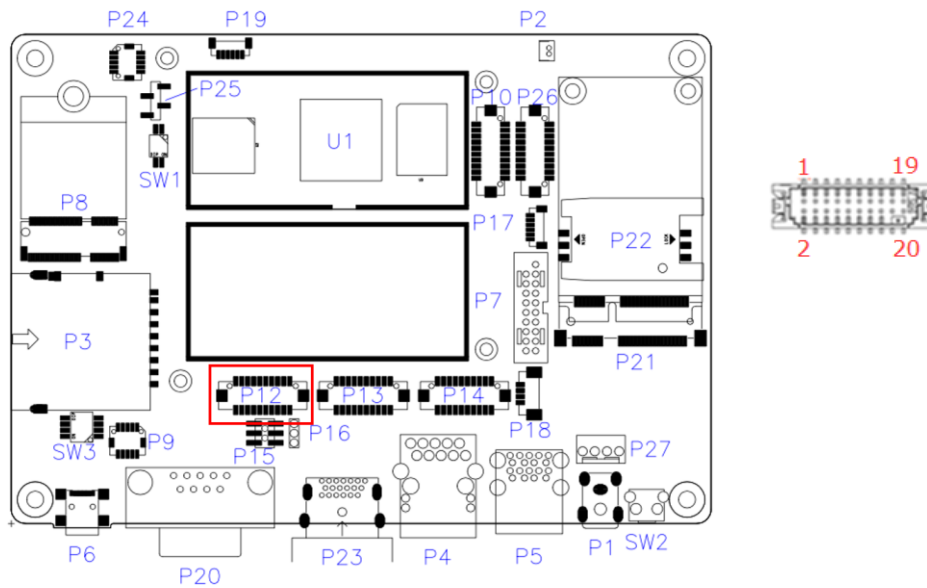
P10:

Pin	Assignment	Pin	Assignment
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:

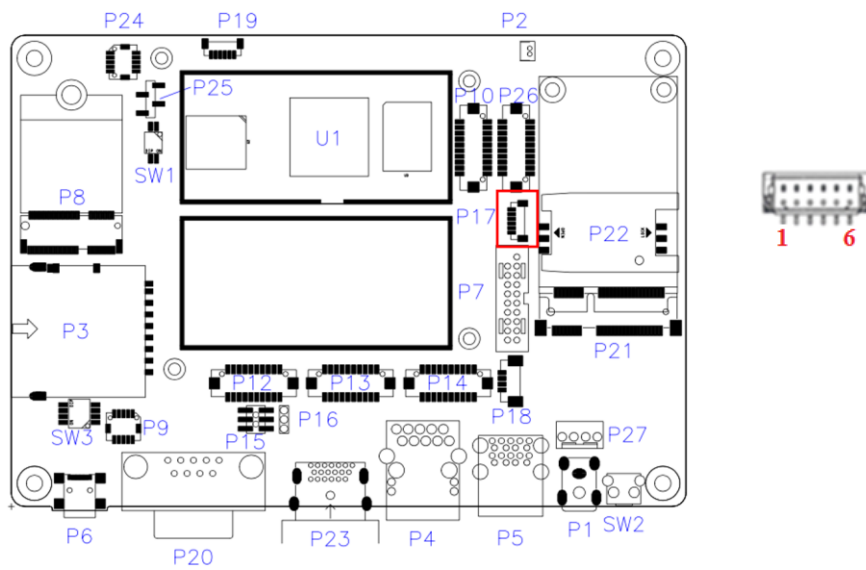
Pin	Assignment	Pin	Assignment
1	MIPI_CSI2_CKP	2	MIPI_CSI2_CKN
3	MIPI_CSI2_DP0	4	MIPI_CSI2_DN0
5	MIPI_CSI2_DP1	6	MIPI_CSI2_DN1
7	MIPI_CSI2_DP2	8	MIPI_CSI2_DN2
9	MIPI_CSI2_DP3	10	MIPI_CSI2_DN3
11	GND	12	GND
13	CSI2_SCL	14	CSI2_SDA
15	CSI2_RST_B	16	VDD_2V8
17	CSI2_PWEN_B	18	VDD_1V8
19	CSI2_MCLK	20	GND

2.4.11 MIPI-DSI Connector (P12)



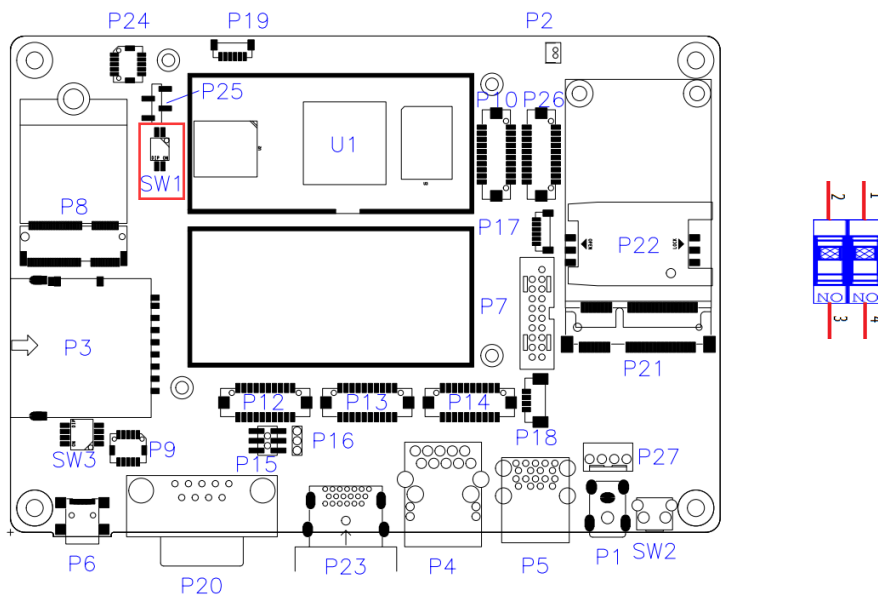
Pin	Assignment	Pin	Assignment
1	MIPI_DSI_CKP	2	MIPI_DSI_CKN
3	GND	4	GND
5	MIPI_DSI_DP0	6	MIPI_DSI_DN0
7	GND	8	VCC_LCD
9	MIPI_DSI_DP1	10	MIPI_DSI_DN1
11	MIPI_DSI_DP2	12	MIPI_DSI_DN2
13	HDMI_INT	14	GND
15	MIPI_DSI_DP3	16	MIPI_DSI_DN3
17	I2C2_SCL	18	VCC_LCD
19	I2C2_SDA	20	BKLT

2.4.12 I²C Connector (P17)



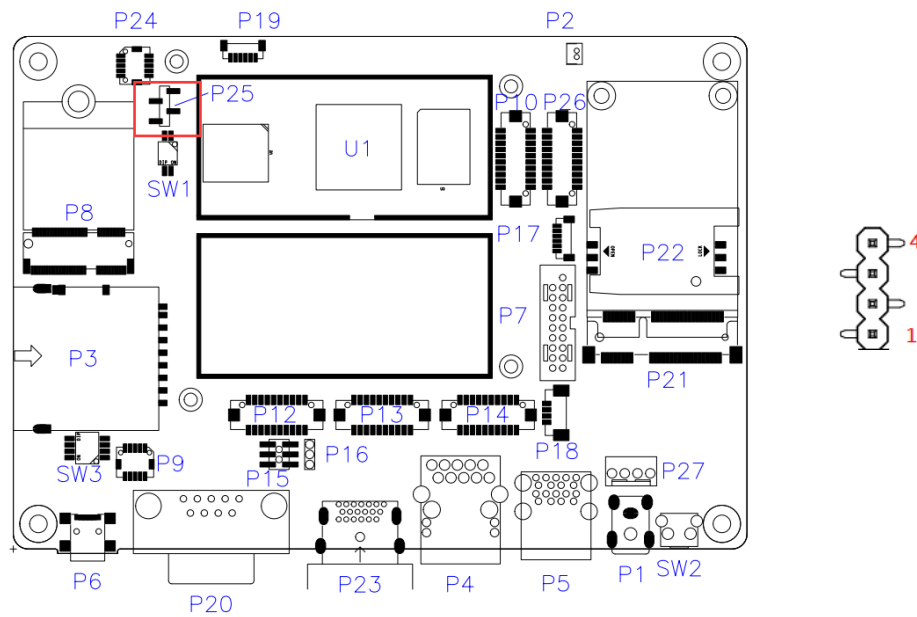
Pin	Assignment	Pin	Assignment
1	3V3	4	I2C2_SCL
2	TP_INT_B	5	I2C2_SDA
3	TP_RST_B	6	GND

2.4.13 Boot mode select (SW1 Factory use only)



Panel Type	1-4	2-3
USB download mode	On	Off
EMMC Normal Boot	Off	Off
SD Normal Boot	Off	On

2.4.14 RS232 Debug Port (P25)



Pin	Assignment
1	Debug_RX
2	Deubg_TX
3	GND
4	NC

Chapter 3

Software Setup

This chapter introduces the following setup on the device:

(for advanced users only)

- Make a recovery SD card
- Upgrade firmware through the recovery SD card

3.1 Make a Recovery SD Card

Note: This is for advanced users with the IBASE standard image file only.

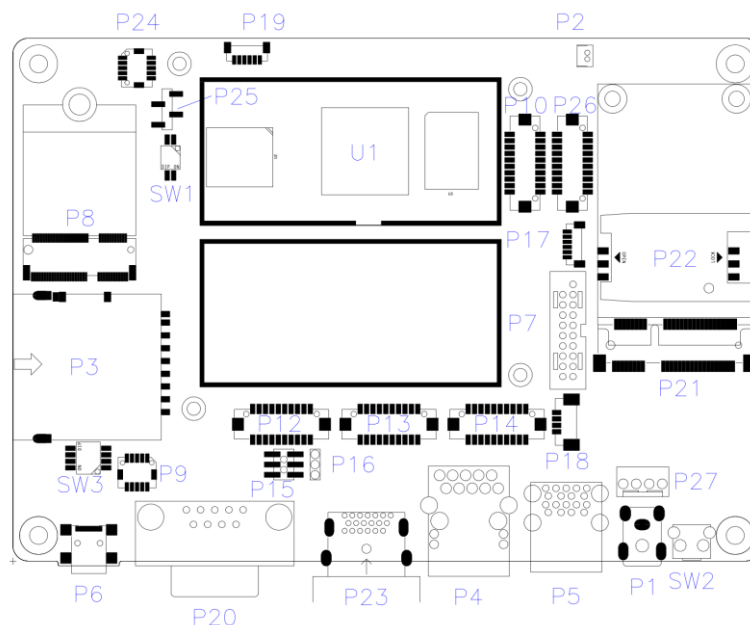
In general, ISR301 is preloaded with O.S (Android or Yocto) into eMMC by default. Connect the HDMI cable to ISR301, and the 12V-24V power directly.

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.



Set dip switch: SW1 pin1 ON to enter 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)

名称	修改日期	类型	大小
changelog.txt	2019/11/15 17:58	文本文档	3 KB
fsl-image-qt5-imx8mqevk.rootfs.sdcard	2019/11/15 17:51	SDCARD 文件	603,920 KB
FW Download SDcard.bat	2019/9/27 15:39	Windows 批处理...	1 KB
IBR210-sd-recovery-guideline.docx	2019/10/12 10:50	Microsoft Word ...	67 KB
imx-boot-imx8mqevk-sd.bin-flash_evk	2019/10/11 18:15	BIN-FLASH_EVK ...	1,409 KB
uuu.exe	2019/10/15 18:22	应用程序	923 KB
uuu-sdcard.auto	2019/9/24 11:47	AUTO 文件	1 KB

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/`

名称	修改日期	类型	大小
fsl-imx8mq-evk.dtb	2019/9/27 15:52	DTB 文件	42 KB
Image	2019/9/27 15:52	文件	25,629 KB
imx-boot-imx8mqevk-sd.bin-flash_evk	2019/9/27 15:52	BIN-FLASH_EVK ...	1,409 KB
recovery.tar.bz2	2019/9/27 16:00	360压缩	114,707 KB
rootfs.tar.bz2	2019/9/27 15:59	360压缩	612,143 KB

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

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

1. If you have used the HDMI display, run the command below.

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

2. If you use an LVDS 21.5" display, run the command below.

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

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

Chapter 4

BSP Source Guide

This chapter is dedicated for advanced software engineers only 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/

boot.img	partition-table-7GB.bpt	ramdisk.img	uuu_imx_android_flash.bat
dtbo.img	partition-table-7GB.img	ramdisk-recovery.img	uuu_imx_android_flash.sh
dtbo-imx8mq-7inch.img	partition-table.bpt	system.img	vbmeta.img
dtbo-imx8mq.img	partition-table-default.bpt	u-boot.img	vbmeta-imx8mq-7inch.img
partition-table-28GB.bpt	partition-table-default.img	u-boot-imx8mq-evk-uuu.img	vbmeta-imx8mq.img
partition-table-28GB.img	partition-table.img	u-boot-imx8mq.img	vendor.img

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
```

Appendix

This section provides information on the reference code.

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_KEEPAIVE, &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. Then 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 USB 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 SD card 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_iflag  &= ~(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_iflag &= ~(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: Due to Android includes 3G config in setting, this portion is not suit for Android version.

M. Onboard Connector Types

Function	Connector Name	Onboard Type	Compatible Mating Type for Reference
LVDS Display Connector	P13, P14	Hirose DF13E-20DP-1.25V	Hirose DF13-20DS-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-20DP-1.25V	Hirose DF13-20DS-1.25C
MIPI-DSI	P12	Hirose DF13E-20DP-1.25V	Hirose DF13-20DS-1.25C
I ² C 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.

N. 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
```

O. 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_KEEPAIVE, &dummy);
```