IBR115

NXP Cortex™ A9 i.MX6
Dual-Lite SoC

User's Manual

Version 1.2
(Dec. 2018)

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

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.

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 60˚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.

Care for your IBASE products:

- Before cleaning the PCB, unplug all cables and remove the battery.
- Clean the PCB with a circuit board cleaner or degreaser, or use cotton swabs and alcohol.

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

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 assistance from your distributor or sales representative concerning problems that you may have encountered, please prepare the following information:
   - 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

Compliance iii

Important Safety Information 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 3
1.4 Optional Accessories 3
1.5 Specifications 4
1.6 Overview 6
1.7 Dimensions 7

Chapter 2 Hardware Configuration 9

2.1 M.2 Card Installation 10
2.2 Setting the Jumpers 11
2.2.1 How to Set Jumpers 11
2.3 Jumper & Connector Locations on IBR115 12
2.4 Jumpers Quick Reference 13
2.4.1 LVDS Power Setting (P16) 13
2.4.2 LVDS Backlight Power Setting (P15) 14
2.5 Connectors Quick Reference 15
2.5.1 RS-232/422/485 Selection (SW3) 16
2.5.2 RTC Lithium Cell Connector (P2) 16
2.5.3 LVDS Display Connector (P8) 17
2.5.4 LVDS Backlight Control Connector (P9) 18
2.5.5 USB Hub Connector (P13) 18
2.5.6 COM RS-232/422/485 Port (P10) 19
2.5.7 Audio Connector (P14) 20
2.5.8 2-Wire UART Connector (P17) 20
2.5.9 Digital (GPIO) Connector (P18) 21
2.5.10 COM RS-232 Debug Connector (P19) 21
2.5.11 I^{2}C Connector (P20) 22

2.5.12 BOot Selection (SW1) 22

Chapter 3 Software Setup 23

3.1 Make a Recovery MicroSD Card..24

3.1.1 Preparing the Recovery MicroSD card to Install Linux/ Android image into eMMC 24

3.1.2 Upgrade Firmware through the Recovery SD Card...............26

3.2 Display Parameter Setting in Kernel 26

Chapter 4 BSP Source Guide...27

4.1 Building BSP Source 28

4.1.1 Preparation..28

4.1.2 Installing Toolchain 28

4.1.3 Building BSP Release..28

4.1.4 Building U-Boot Single 28

4.1.5 Building Kernel Single...28

Appendix 29

A. How to Use GPIO in Linux 30

B. How to Use Watchdog in Linux 30

C. eMMC Test 31

D. USB (flash disk) Test 32

E. MicroSD Card Test 33

F. RS-232 Test 34

G. RS-485 Test 35

H. Audio Test 35

I. I^{2}C Test 36

J. Ethernet Test 36

K. LVDS Test 37

L. HDMI Test 38

M. Onboard Connector Types...39

This page is intentionally left blank.

Chapter 1
General Information

The information provided in this chapter includes:

- Features
- Packing List
- Specifications
- Block Diagram
- Board Overview
- Board Dimensions

1.1 Introduction

IBR115 is a 2.5" Disk-Size SBC with ARM Based NXP i.MX6 Cortex-A9 1GHz CPU. The device offers 2D, 3D graphics and multimedia accelerations while also supporting numerous peripherals, including RS232/422/485, COM, GPIO, USB, USB OTG, LAN and audio interfaces. For display, it supports one HDMI for a full HD display and one LVDS. For wireless connectivity, it supports M.2 Key-E, Type 2230 for the M.2 module expansion that are well suited for industrial applications.

Photo of IBR115

1.2 Features

- NXP ARM Cortex™ A9 i.MX6 Dual-Lite 1 GHz processor
- Video output through HDMI and single-channel LVDS
- 1 GB DDR3, 4 GB eMMC and MicroSD socket for expansion
- Rich I/O interface with COM, GPIO, USB, USB-OTG, audio and Ethernet
- M.2 Key-E (2230) for wireless connectivity
- 2 x UART headers
- OpenGL ES 2.0 for 3D BitBlt for 2D and OPENVG1.1 hardware accelerators

1.3 Packing List

Your IBR115 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.

• IBR115 2.5” SBC	x 1
• This User Manual	x 1

1.4 Optional Accessories

IBASE provides optional accessories as follows. Please contact us or your dealer if you need any.

RF:

- WiFi & BT Combo (M.2 card)

Display:

- 7” TFT LCD – LVDS
- 15” 1024 x 768 TFT LCD – LVDS

Cable:

- Debug cable (PK1-100A)
- LVDS Cable for 7” TFT LCD – LVDS (LCD403)
- LVDS Cable for 15” 1024 x 768 TFT LCD – LVDS (LCD404)
- Backlight Cable (IVT-76)
- USB Cable (USB-134)
- GPIO Cable (PK1-124)
- UART Cable (PK1-125)
- Audio Cable (Audio-68)

1.5 Specifications

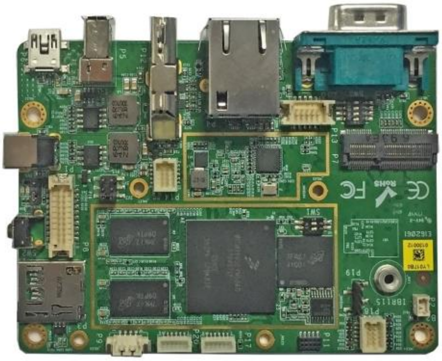
Product Name	**IBR115**
Form Factor	2.5” SBC
System	
Operating System	• Yocto (kernel 4.1) • Android 6.0
CPU Type	NXP Cortex™ A9 i.MX6 Dual-Lite
CPU Speed	1 GHz
Memory	• **System memory:** 1 GB DDR3 • **Data Memory:** 4 GB eMMC
Video Codec	• **Encoder:** 1080p, 30 fps (MPEG-4 SP, H.264 BP, H.263, MJPEG BP) • **Decoder:** 1080p, 30 fps (MPEG-4 ASP, H.264 HP, MPEG-2 MP, MJPEG BP)
Touch	USB headers for CPT
Audio	1 x 10-pin box header for Line-In & Line-Out
RTC	AnalogTEK AT8565S
Wireless	WiFi / BT module (Optional)
Power Supply	12V DC-In
Watchdog Timer	Yes (256 segments, 0, 1, 2…128 secs)
Dimensions	100 x 72 mm (4” x 2.8”)
RoHS	Yes
Certification	CE, FCC Class B
I/O Ports	
Display	• 1 x HDMI 1.4 (1080p at 60 Hz) • 1 x LVDS (1366 x 768 at 60 Hz, 18-bit / 24-bit, single channel)
LAN	1 x RJ45 GbE LAN

USB	• 1 x USB 2.0 Type A • 1 x USB OTG via mini-USB Type B
Serial	• 1 x COM RS-232/422/485 port • 2 x 2-wire UART port via an onboard pin header
MicroSD	1 x MicroSD socket (max.104 MB/s)
Audio Jack	• 1 x Line-In • 1 x Line-Out
Digital IO	8 In / Out
Expansion Slots	1 x M.2 E-key (2230)
Environment	
Operating Temperature	• 0 ~ 60 °C (32 ~ 140 °F) • -40 ~ 85 °C (-40 ~ 185 °F) with the optional heatsink
Relative Humidity	10 ~ 90 %, non-condensing

All specifications are subject to change without prior notice.

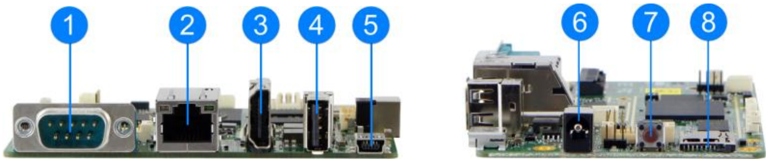
1.6 Overview

Top View



* The photos above are for reference only. Some minor components may differ.

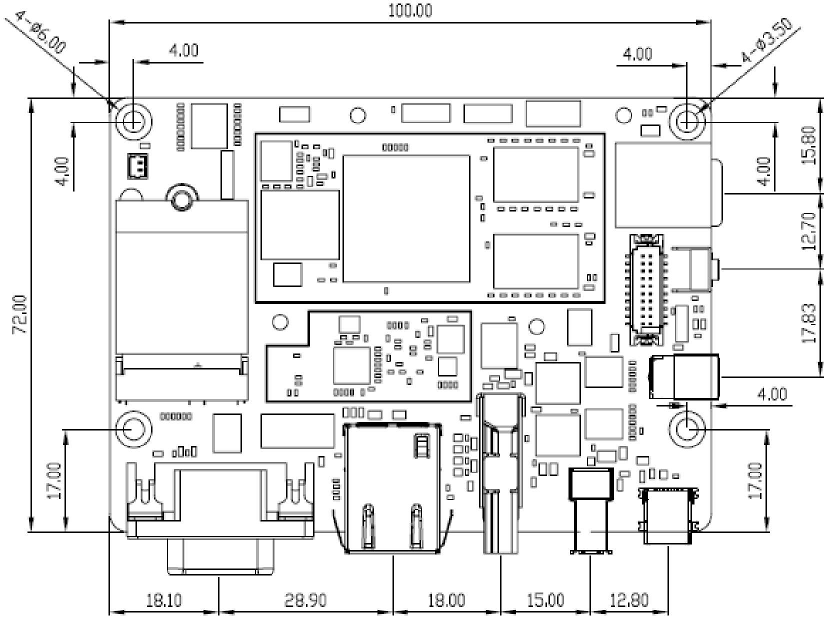
I/O View

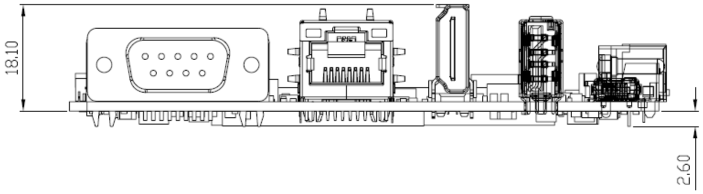


No.	Name	No.	Name
1	COM RS-232/422/485 Port	5	USB OTG Port
2	GbE LAN Port	6	DC Jack
3	HDMI Port	7	Reset Button
4	USB 2.0 Port	8	MicroSD slot

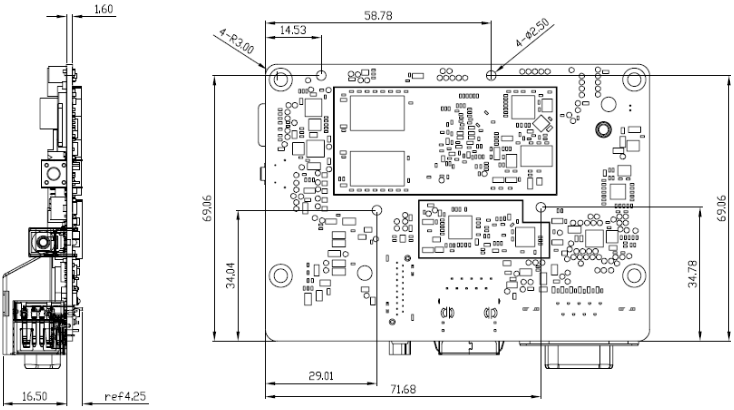
1.7 Dimensions

Unit: mm

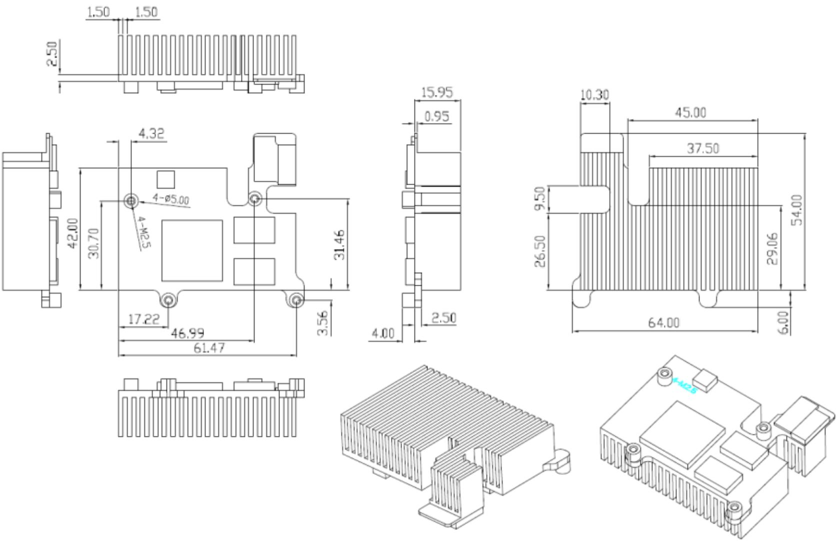




Unit: mm



Dimensions of the optional heatsink:



Chapter 2
Hardware Configuration

This section provides information on jumper settings and connectors on the IBR115 in order to set up a workable system. The topics covered are:

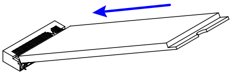
- M.2 card Installation
- Jumper and connector locations
- Jumper settings and information of connectors

2.1 M.2 Card Installation

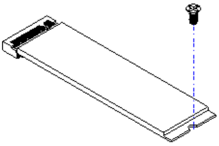
To install the NGFF M.2 card, locate the slot on the board and perform the following steps.

(This is illustrated by the example of M.2 Type 2280)

1. Align the key of the M.2 card to the interface, and insert the card slantwise.



2. Push the M.2 card down, fix it onto the brass standoff with a screw.

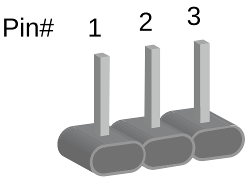


2.2 Setting the Jumpers

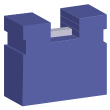
Set up and configure your IBR115 by using jumpers for various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your use.

2.2.1 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.

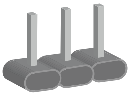
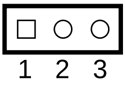
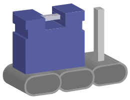
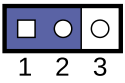
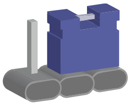
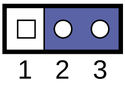


A 3-pin jumper



A jumper cap

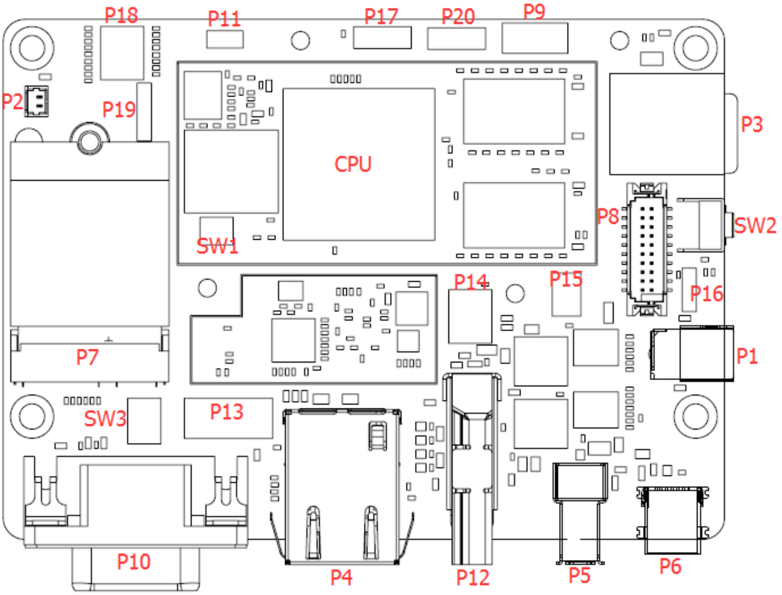
Refer to the illustration below to set jumpers.

Pin closed	Oblique view	Illustration
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 IBR115

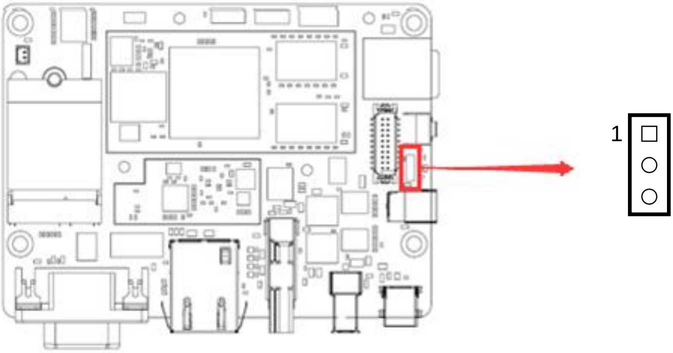


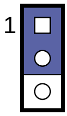
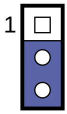
Board diagram of IBR115

2.4 Jumpers Quick Reference

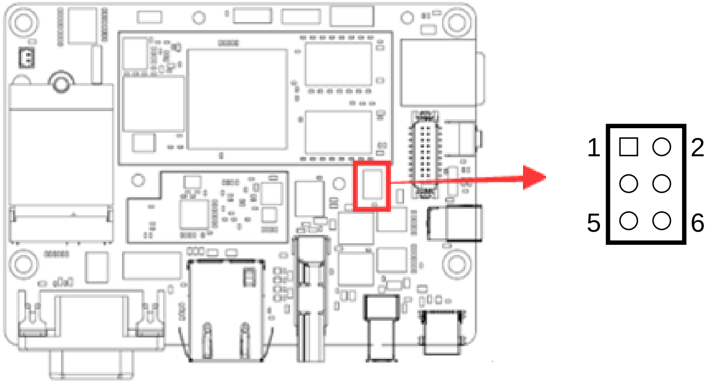
Function	Jumper Name	Page
LVDS Power Setting	P16	13
LVDS Backlight Power Setting	P15	14

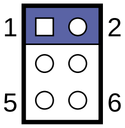
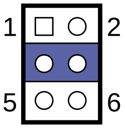
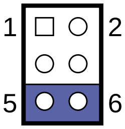
2.4.1 LVDS Power Setting (P16)



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

2.4.2 LVDS Backlight Power Setting (P15)

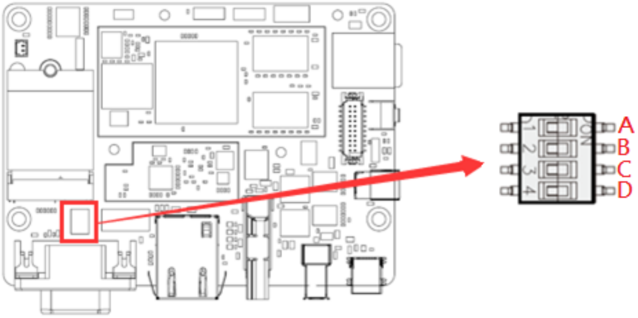


Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	3-4	
12V	5-6	

2.5 Connectors Quick Reference

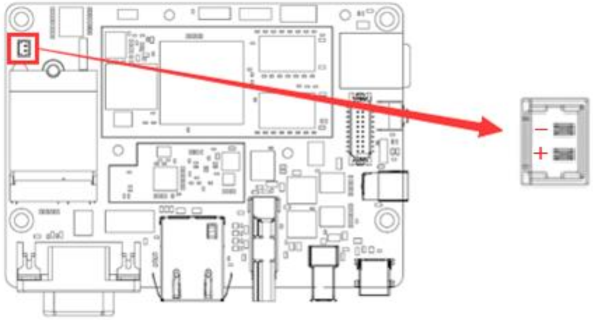
Function	Connector Name	Page
RS-232/422/485 Selection	SW3	16
RTC Lithium Cell Connector	P2	16
LVDS Display Connector	P8	17
LVDS Backlight Control Connector	P9	18
USB Hub Connector	P13	18
COM RS-232/422/485 Port	P10	19
Audio Connector	P14	20
2-Wire UART Connector	P17	20
Digital (GPIO) Connector	P18	21
COM RS-232 Debug Connector	P19	21
I^{2}C Connector	P20	22
Boot Mode Selection	SW1	22
System Reset Button	SW2	- -
DC Jack	P1	- -
MicroSD Card Connector	P3	- -
GbE LAN Port	P4	- -
USB 2.0 Type A Port	P5	- -
Mini-USB OTG Port	P6	- -
NGFF M.2 Slot	P7	- -
HDMI Port	P12	- -
Factory Use Only	P11, P19	- -

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



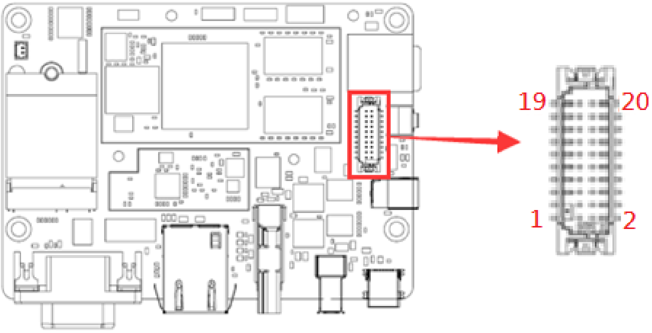
Panel Type	A	B	C	D
RS-422 Full Duplex	NC	ON	ON	ON
Pure RS232 (3T/5R)	NC	OFF	ON	ON
RS-485 Half Duplex (TX Low-Active)	NC	ON	OFF	ON
RS-485 Half Duplex (TX High-Active)	NC	OFF	OFF	ON
RS-422 Full Duplex	NC	ON	ON	OFF
RS-485 Half Duplex	NC	ON	OFF	OFF
Shutdown (Default)	NC	OFF	OFF	OFF

2.5.2 RTC Lithium Cell Connector (P2)



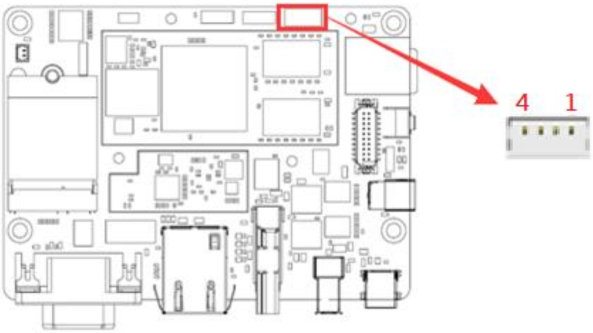
Pin	Signal Name	Pin	Signal Name
1	RTC_VCC	2	Ground

2.5.3 LVDS Display Connector (P8)



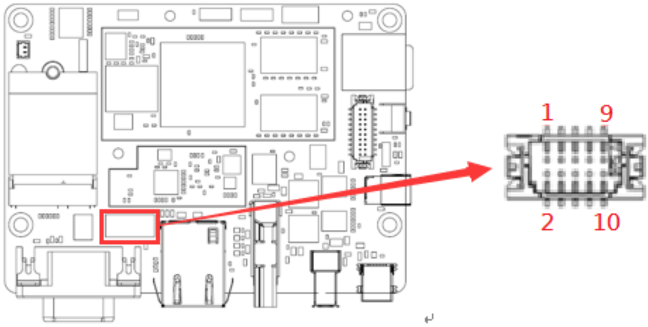
Pin	Signal Name	Pin	Signal Name
1	LCD1_TX0_P	2	LCD1_TX0_N
3	Ground	4	Ground
5	LCD1_TX1_P	6	LCD1_TX1_N
7	Ground	8	LCD_VDD
9	LCD1_TX3_P	10	LCD1_TX3_N
11	LCD1_TX2_P	12	LCD1_TX2_N
13	Ground	14	Ground
15	LCD1_CLK_P	16	LCD1_CLK_N
17	BTL_PWM	18	LCD_VDD
19	BKLT_VCC	20	BKLT_VCC

2.5.4 LVDS Backlight Control Connector (P9)



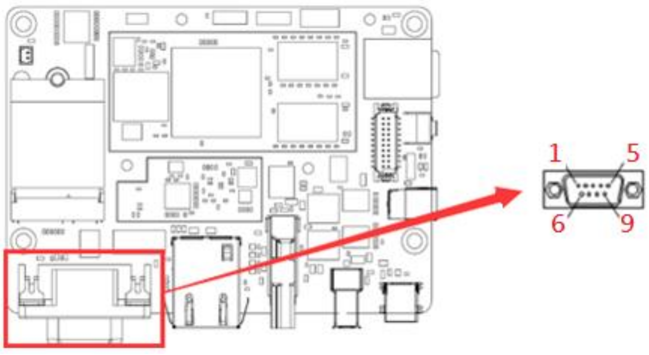
Pin	Signal Name	Pin	Signal Name
1	BKLT_VCC	3	LCD_BKLT_PWM
2	LCD_BKLT_EN	4	Ground

2.5.5 USB Hub Connector (P13)



Pin	Signal Name	Pin	Signal Name
1	Ground	2	NC
3	USB1_DP	4	USB2_POWER
5	USB1_DM	6	USB2_DM
7	USB1_POWER	8	USB2_DP
9	NC	10	Ground

2.5.6 COM RS-232/422/485 Port (P10)

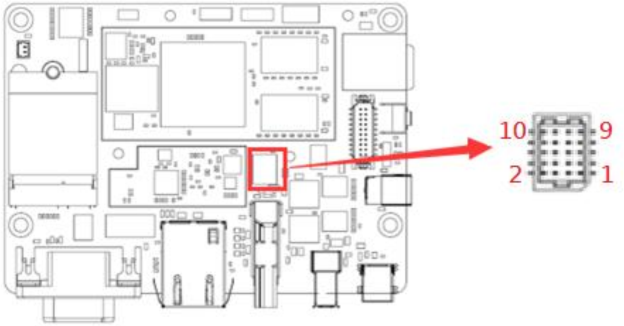


Refer to SW3 setting for RS-232/422/485 mode selection.

Pin	Signal Name	Pin	Signal Name
1	DCD, Data carrier detect	6	DSR, Data set ready
2	RXD, Receive data	7	RTS, Request to send
3	TXD, Transmit data	8	CTS, Clear to send
4	DTR, Data terminal ready	9	NC
5	Ground		

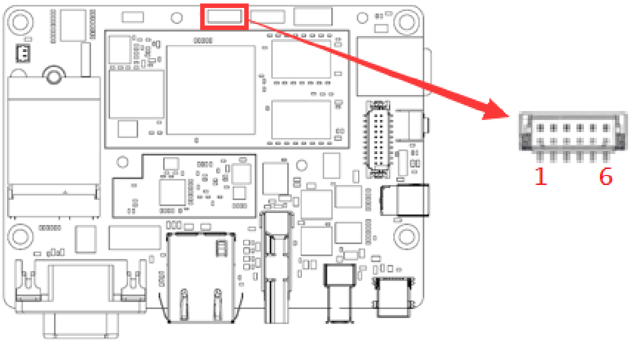
Pin	Assignment		
	RS-232	RS-422	RS-485
1	DCD	TX-	DATA-
2	RX	TX+	DATA+
3	TX	RX+	NC
4	DTR	RX-	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	NC	NC	NC

2.5.7 Audio Connector (P14)



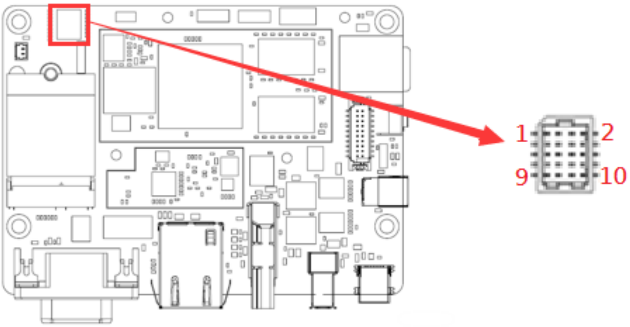
Pin	Signal Name	Pin	Signal Name
1	NC	2	Ground
3	LINE_IN_R	4	Ground
5	LINE_IN_L	6	Ground
7	Ground	8	LINE_OUT_L
9	Ground	10	LINE_OUT_R

2.5.8 2-Wire UART Connector (P17)



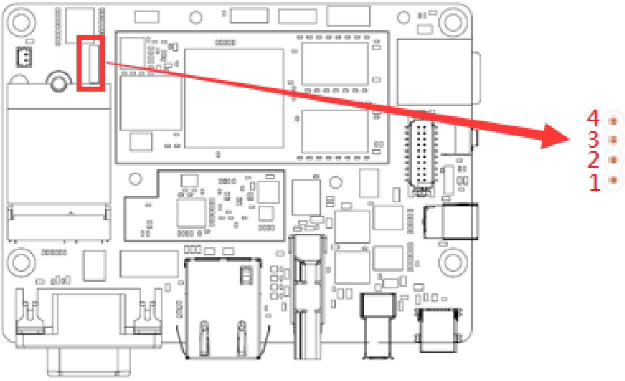
Pin	Signal Name	Pin	Signal Name
1	UART2_TXD	4	UART3_TXD
2	UART2_RXD	5	UART3_RXD
3	Ground	6	Ground

2.5.9 Digital (GPIO) Connector (P18)



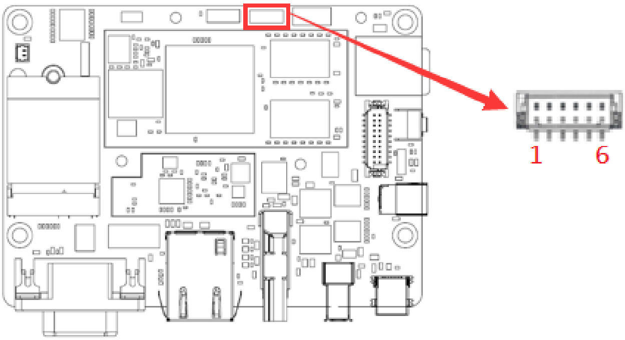
Pin	Signal Name	Pin	Signal Name
1	3.3V	2	DIO5
3	DIO1	4	DIO6
5	DIO2	6	DIO7
7	DIO3	8	DIO8
9	DIO4	10	Ground

2.5.10 COM RS-232 Debug Connector (P19)



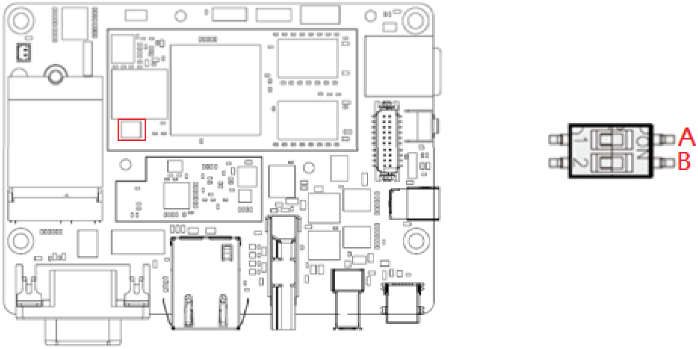
Pin	Signal Name	Pin	Signal Name
1	RX, Receive data	3	Ground
2	TX, Transmit data	4	5V

2.5.11 I²C Connector (P20)



Pin	Signal Name	Pin	Signal Name
1	VCC	4	I2C_SCL
2	TP_INT_B	5	I2C_SDA
3	TP_RST_B	6	GND

2.5.12 Boot Selection (SW1)



Boot Mode	SW1A	Boot Media	SW1B
Normal Boot (Default)	OFF	SDHC3(eMMC) (Default)	OFF
Download	ON	SDHC2(SD)	ON

Chapter 3
Software Setup

This chapter introduces installation of the following drivers:

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

3.1 Make a Recovery MicroSD Card

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

Basically, IBR115 is preloaded with O.S (Android / Linux) into eMMC by default. Connect the TFT-LCD with IBR115 (or HDMI), and 12V power directly.

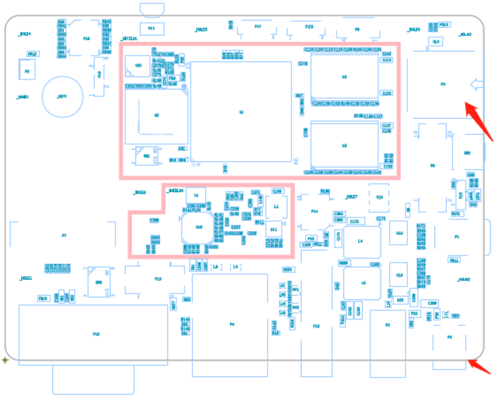
This chapter guides you to make a recovery boot-up microSD card. IBASE optionally provides 7" /15" LVDS panel and HDMI display for you to prepare the software application pre-development easily under Windows platform.

3.1.1 Preparing the Recovery MicroSD card to Install Linux/ Android image into eMMC

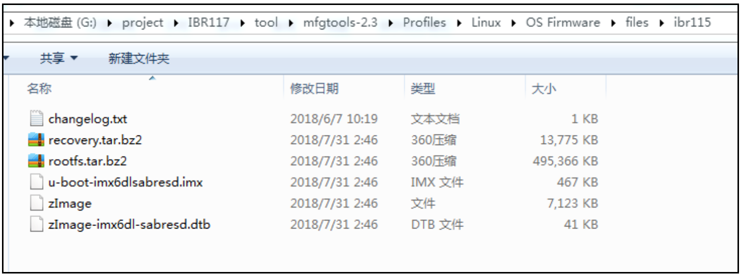
Note: All data in the eMMC will be erased.

1. System requirements:
   Operating System: Windows 7 or later
   Tool: mfgtools-2.3
   SD card: 4GB or greater in size

2. Insert your SD card to this board (i.e. the P3 connector), connect the board to PC through the mini-USB port (i.e. the P6 connector), and change the SW1 boot mode to "Download" state.



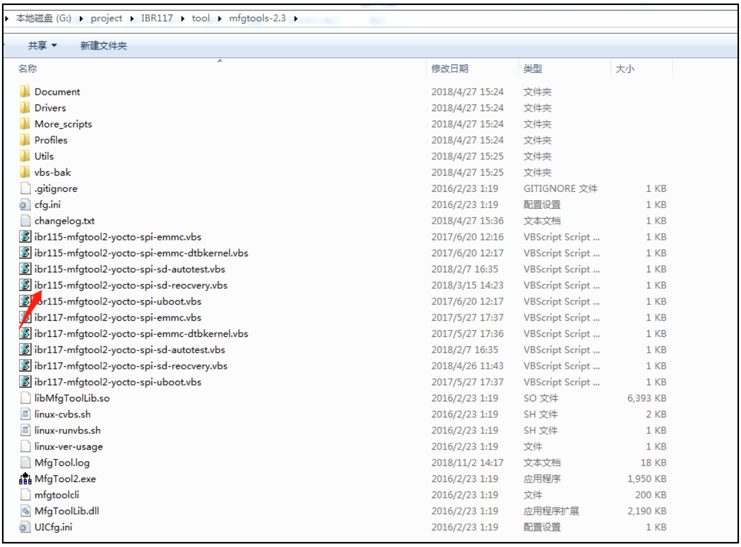
3. Copy the image to the directory **mfgtools-2.3\Profiles\Linux\OS Firmware\files\ibr115\**.



本地磁盘 (G:) ▸ project ▸ IBR117 ▸ tool ▸ mfgtools-2.3 ▸ Profiles ▸ Linux ▸ OS Firmware ▸ files ▸ ibr115

名称	修改日期	类型	大小
changelog.txt	2018/6/7 10:19	文本文档	1 KB
recovery.tar.bz2	2018/7/31 2:46	360压缩	13,775 KB
rootfs.tar.bz2	2018/7/31 2:46	360压缩	495,366 KB
u-boot-imx6dlsabresd.imx	2018/7/31 2:46	IMX 文件	467 KB
zImage	2018/7/31 2:46	文件	7,123 KB
zImage-imx6dl-sabresd.dtb	2018/7/31 2:46	DTB 文件	41 KB

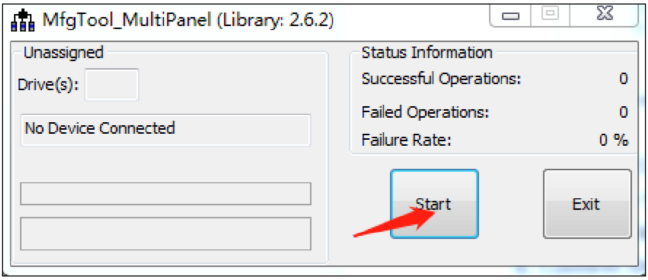
4. Select the script of **mfgtools sdcard recovery**.



本地磁盘 (G:) ▸ project ▸ IBR117 ▸ tool ▸ mfgtools-2.3 ▸

名称	修改日期	类型	大小
Document	2018/4/27 15:24	文件夹	
Drivers	2018/4/27 15:24	文件夹	
More_scripts	2018/4/27 15:24	文件夹	
Profiles	2018/4/27 15:24	文件夹	
Utils	2018/4/27 15:25	文件夹	
vbs-bak	2018/4/27 15:25	文件夹	
.gitignore	2016/2/23 1:19	GITIGNORE 文件	1 KB
cfg.ini	2016/2/23 1:19	配置设置	1 KB
changelog.txt	2018/4/27 15:36	文本文档	1 KB
ibr115-mfgtool2-yocto-spi-emmc.vbs	2017/6/20 12:16	VBScript Script ...	1 KB
ibr115-mfgtool2-yocto-spi-emmc-dtbkernel.vbs	2017/6/20 12:17	VBScript Script ...	1 KB
ibr115-mfgtool2-yocto-spi-sd-autotest.vbs	2018/2/7 16:35	VBScript Script ...	1 KB
ibr115-mfgtool2-yocto-spi-sd-reocvery.vbs	2018/3/15 14:23	VBScript Script ...	1 KB
ibr115-mfgtool2-yocto-spi-uboot.vbs	2017/6/20 12:17	VBScript Script ...	1 KB
ibr117-mfgtool2-yocto-spi-emmc.vbs	2017/5/27 17:37	VBScript Script ...	1 KB
ibr117-mfgtool2-yocto-spi-emmc-dtbkernel.vbs	2017/5/27 17:36	VBScript Script ...	1 KB
ibr117-mfgtool2-yocto-spi-sd-autotest.vbs	2018/2/7 16:35	VBScript Script ...	1 KB
ibr117-mfgtool2-yocto-spi-sd-reocvery.vbs	2018/4/26 11:43	VBScript Script ...	1 KB
ibr117-mfgtool2-yocto-spi-uboot.vbs	2017/5/27 17:37	VBScript Script ...	1 KB
libMfgToolLib.so	2016/2/23 1:19	SO 文件	6,393 KB
linux-cvbs.sh	2016/2/23 1:19	SH 文件	2 KB
linux-runvbs.sh	2016/2/23 1:19	SH 文件	1 KB
linux-ver-usage	2016/2/23 1:19	文件	1 KB
MfgTool.log	2018/11/2 14:17	文本文档	18 KB
MfgTool2.exe	2016/2/23 1:19	应用程序	1,950 KB
mfgtoolcli	2016/2/23 1:19	文件	200 KB
MfgToolLib.dll	2016/2/23 1:19	应用程序扩展	2,190 KB
UICfg.ini	2016/2/23 1:19	配置设置	1 KB

5. Start burning image to SD card. When the image buring finishes, the recovery SD card is ready.



MfgTool_MultiPanel (Library: 2.6.2)

Unassigned

Drive(s):

No Device Connected

Status Information

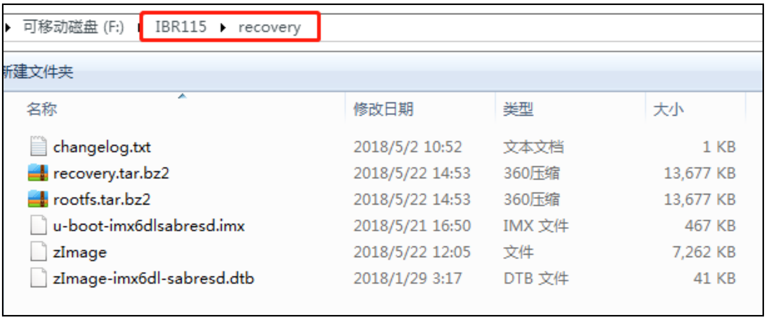
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

3.1.2 Upgrade Firmware through the Recovery SD Card

1. Copy the image files which you'd like to upgrade to U-disk **/IBR115/recovery/**.



可移动磁盘 (F:) ▸ IBR115 ▸ recovery

新建文件夹

名称	修改日期	类型	大小
changelog.txt	2018/5/2 10:52	文本文档	1 KB
recovery.tar.bz2	2018/5/22 14:53	360压缩	13,677 KB
rootfs.tar.bz2	2018/5/22 14:53	360压缩	13,677 KB
u-boot-imx6dlsabresd.imx	2018/5/21 16:50	IMX 文件	467 KB
zImage	2018/5/22 12:05	文件	7,262 KB
zImage-imx6dl-sabresd.dtb	2018/1/29 3:17	DTB 文件	41 KB

2. Insert the recovery SD card and U-disk to the board for upgrade.
3. Power up the board and the recovery program starts.
4. When the recovery finishes, power down the board, pull out the recovery SD card and U-disk.

3.2 Display Parameter Setting in Kernel

IBR115 supports LVDS 15" output by default.

1. If you use HDMI to display, run the command:

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

2. If you use LVDS 7" to display, run the command:

```
/home/root/display_config/config_displag_mode.sh 2
```

3. If you use LVDS 15" to display, run the command:

```
/home/root/display_config/config_displag_mode.sh 3
```

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 U-Boot
- Building Kernel
- Building RAMdisk Image (Optional)
- Installing Linux to MicroSD Card
- Booting with your MicroSD Card

4.1 Building BSP Source

4.1.1 Preparation

The suggested Host platform is Ubuntu 12.04 and 14.04 in x86 and x64 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 IBR115 source file ibr115-bsp.tar.bz2 into "/home/" folder.

4.1.2 Installing Toolchain

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

4.1.3 Building BSP Release

```
cd /home/ibr115-bsp
./build-bsp.sh
```

4.1.4 Building U-Boot Single

```
cd /home/ibr115-bsp/uboot/
./build.sh
```

4.1.5 Building Kernel Single

```
cd /home/ibr115-bsp/kernel/
./build.sh
```

Appendix

This section provides the information of reference code.

A. How to Use GPIO in Linux

```
# Take gpio1_0  as example, export value:1*32+0=32

# GPIO example 1: Output
echo 32 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio40/direction
echo 0 > /sys/class/gpio/gpio40/value
echo 1 > /sys/class/gpio/gpio40/value

# GPIO example 2: Input
echo 32 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio40/direction
cat /sys/class/gpio/gpio40/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 then assure it is in IBR115 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. MicroSD Card Test

When IBR115 is booted from eMMC, microSD card is “/dev/mmcblk1” and able to see by “ls /dev/mmcblk1*” command:

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

When booting from microSD card, replace test pattern “/dev/mmcblk1” to “/dev/mmcblk0”.

Note: This operation may damage the data stored the microSD 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
```

- **MicroSD 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 -D plughw:1,0
// record mp3 by audio (ALC5640)
arecord -f cd $basepath/b.mp3 -D plughw:1,0
```

I. I^{2}C Test

```
// Connect RTC-IC (at88sc25616c) to I2C bus
// run “i2cdetect –y 2” to read chip address
Sudo i2cdetect –y 2
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- 58 59 5a 5b 5c 5d 5e 5f
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

J. Ethernet Test

- ### Ethernet FTP test

```
#intall ftp
sudo chmod 777 /home/root/testscript/ethernet/lftp_library/lftp
sudo cp /home/root/testscript/ethernet/lftp_library/lftp /usr/bin/
sudo mkdir /usr/lib/lftp
sudo mkdir /usr/lib/lftp/4.6.3a
sudo cp /home/root/testscript/ethernet/lftp_library/*.so /usr/lib/lftp/4.6.3a
sudo cp /home/root/testscript/ethernet/lftp_library/* /lib/

#lftp server 192.168.1.123
lftp 192.168.1.123 <<EOF
#upload data to server  192.168.1.123
put data1
#download data from server  192.168.1.123
get data1
#exit lftp
bye
```

- ### 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
```

K. 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);
```

L. 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
```

M. Onboard Connector Types

Function	Connector Name	Onboard Type	Compatible Mating Type for Reference
LVDS Display Connector	P8	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
LVDS Backlight Control Connector	P9	TechBest 1024041008	Molex 51021-0400
USB Hub Connector	P13	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
Audio Connector	P14	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H
2-Wire UART Connector	P17	TechBest WT02M-30002-06132	JST SHR-03V-S-B
Digital (GPIO) Connector	P18	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H