IBR117

3.5" ARM-based SBC
With NXP Cortex™ A9 Dual Core i.MX6

User's Manual

Version 1.1
(June 2018)

iBASE

Copyright

© 2018 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 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

In a domestic environment, this product may cause radio interference in which case users may be required to take adequate measures.

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)

iBASE

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.
- Vacuum the dust with a computer vacuum cleaner to prevent the fan from being clogged.

⚠️ **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.

iBASE

Table of Contents

iBASE

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

iBASE

1.1 Introduction

IBR117 is a 3.5" Disk-Size SBC w/ ARM Base NXP i.MX6 Cortex-A9
1 GHz CPU. The device offers 2D, 3D graphics and multimedia
accelerations, while also supporting numerous peripherals, including
RS-232/422/485, CAN bus, COM, GPIO, USB, USB OTG, LAN and audio
interfaces. For the display, it also supports 1 HDMI for a full HD display and
Dual LVDS, for the wireless connectivity, it supports M.2 Key-E, type 2230
for the M.2 module and mini-PCIe module expansion that are well suited for
industrial applications.
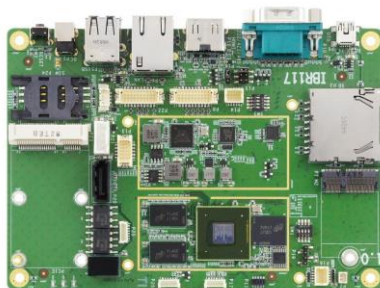
Photo of IBR117

1.2 Features

- NXP ARM Cortex™ A9 i.MX6 Dual Core 1 GHz processor

- Video output through HDMI and single/dual channel LVDS

- 1 GB DDR3, 4 GB eMMC and SD socket for expansion

- Rich I/O interface with CAN bus, COM, GPIO, USB, USB-OTG, audio
  and Ethernet

- M.2 E2230 & mini-PCIe (USB only) with the SIM socket for wireless
  connectivity

- 2 x 2 wire UART headers

- OpenGL ES 2.0 for 3D BitBlt for 2D and OPENVG1.1 hardware
  accelerators

1.3   Packing List

Your IBR117 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.

- IBR117 3.5" 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)
- 3G / LTE Module (Mini-PCIe card)

Display:
- 7" TFT LCD – LVDS
- 15" 1024 x 768 TFT LCD – LVDS
- 21" 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)
- 21.5" LVDS Panel (LCD419)
- Backlight Cable for the LED Driving Board (IVT-82)
- LED Driving Board for 21.5" panel (IVT-83)

iBASE

## 1.5   Specifications

Product Name	IBR117
Form Factor	3.5" SBC
System	
Operating System	• Yocto (kernel 4.1) • Android 6.0
CPU Type	NXP Cortex™ A9 i.MX6 Dual Core 1 GHz SoC
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
RTC	AnalogTEK AT8565S
Wireless	WiFi / BT / 3G / LTE module (Optional)
Power Supply	12V DC-In
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 12V DC jack
Display	• 1 x HDMI 1.4 (up to 1080P30 at 60 Hz) • 2 x 18/24-bit single LVDS / 1 x dual LVDS (up to 1366 x 768 at 60 Hz for 1 channel / 1920 x 1080 for 2 channels)
LAN	1 x RJ45 GbE LAN
USB	• 2 x USB 2.0 Type A • 1 x USB OTG via mini-USB Type B

|	
Serial	• 1 x COM RS-232/422/485 port (DB9 male connector) • 2 x 2 wire UART (6-pin header)
Audio Jack	1 x 10-pin header for Line-In & Line-Out
Digital IO	8-In/Out
Expansion Slots	• 1 x M.2 E2230 with USB, SDIO and UART • 1 x Mini-PCIe with USB only (interrelated with the SIM socket) • 2 x CAN bus (6-pin header)
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.

iBASE

1.6　Overview

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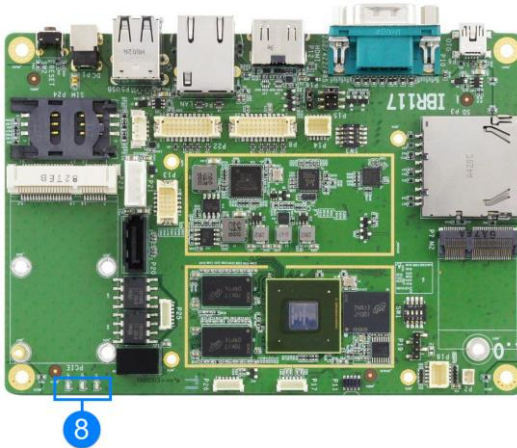
* The photos above are for reference only. Some minor components may differ.

I/O View
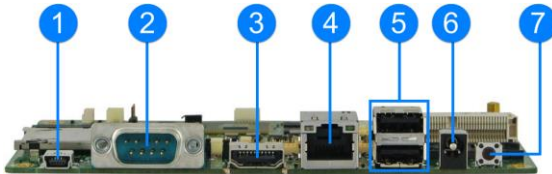
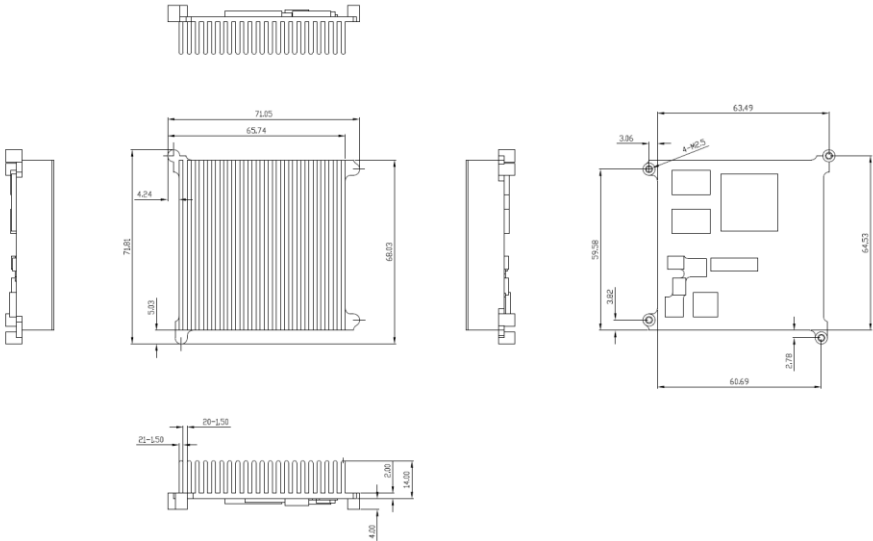| No.	Name	No.	Name
1	USB OTG Port	5	USB 2.0 Ports
2	COM RS-232/422/485 Port	6	DC Jack
3	HDMI Port	7	Reset Button
4	GbE LAN Port	8	LED Indicators (from left to right: Power, Wireless, Programmable Setting)

1.7   Dimensions

Unit: mm
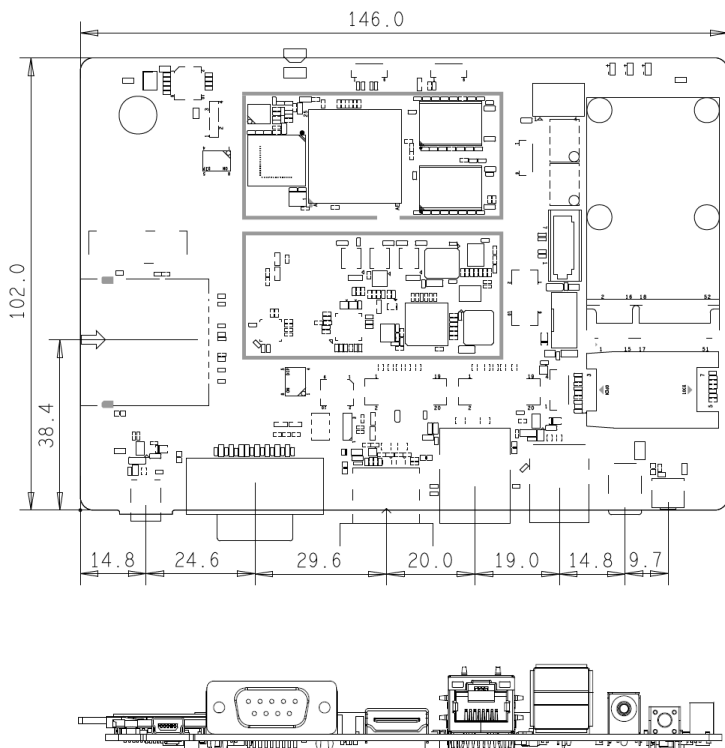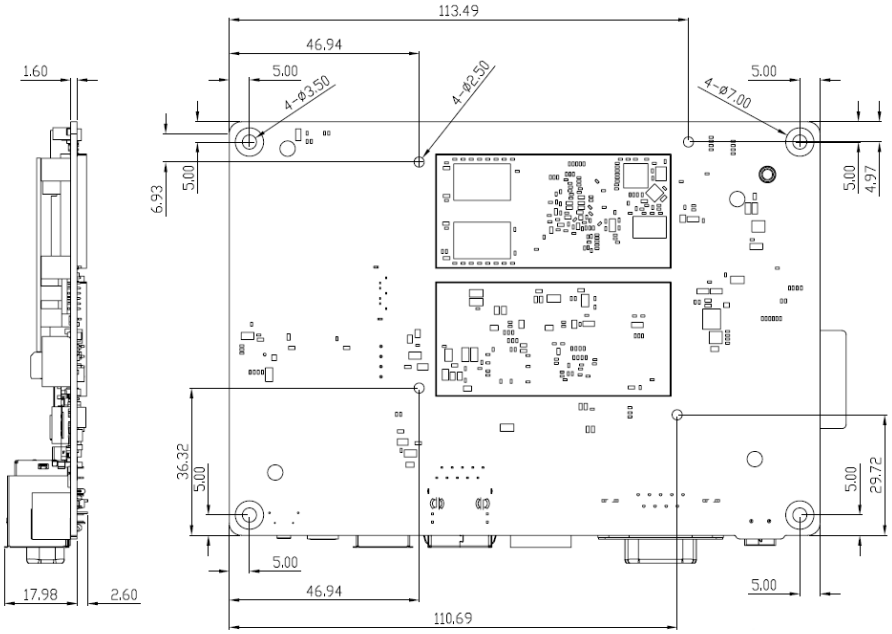
IBR117 Reference Heat Sink

Chapter 2
Hardware Configuration

This section provides information on jumper settings and connectors on the IBR117 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　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.)
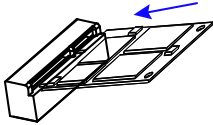
2.　Push the mini-PCIe card down and fix it with 2 flat head screws.

　　(Fix the M.2 card with one screw.)

2.2   Setting the Jumpers

Set up and configure your IBR117 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.
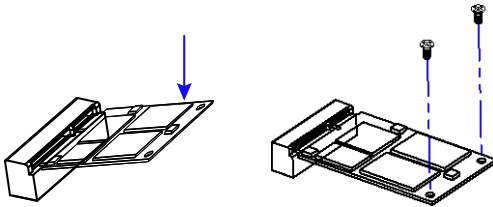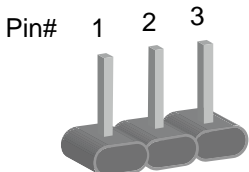
2.2.1   How to Set Jumpers

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.

Pin#   1   2   3
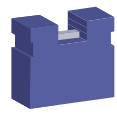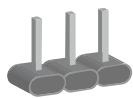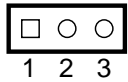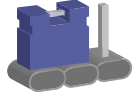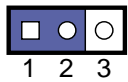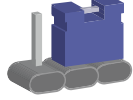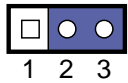
A 3-pin jumper   **A jumper cap**

Refer to the illustration below to set jumpers.

Pin closed	Oblique view	Schematic illustration in the manual
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**.

iBASE

2.3 Jumper & Connector Locations on IBR117
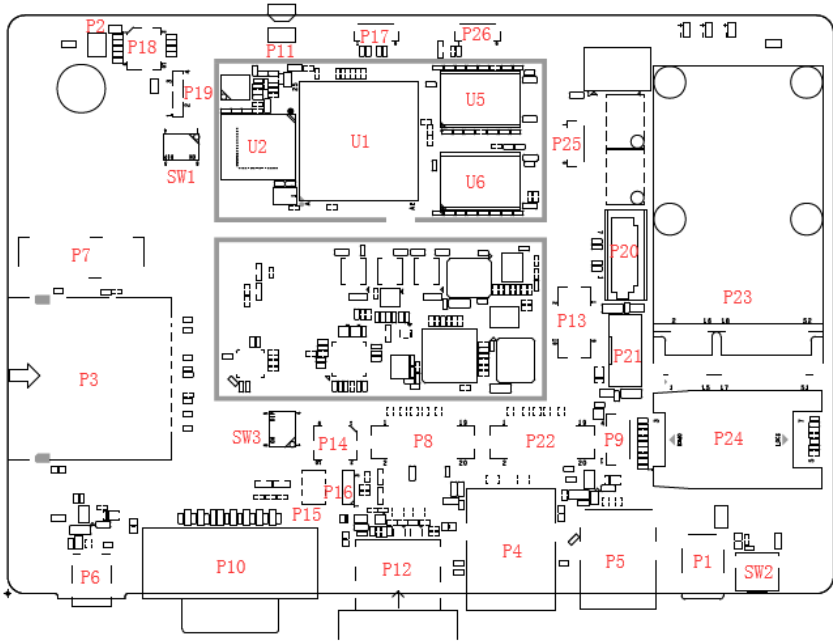
Board diagram of IBR117

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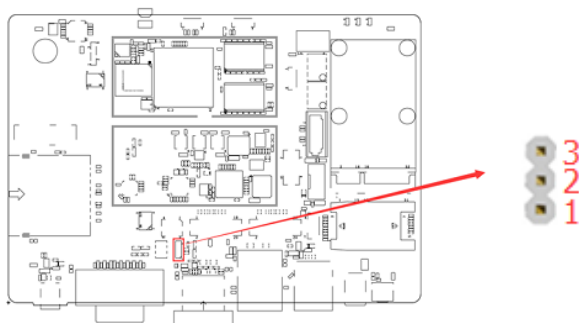
Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	2-3	

2.4.2   LVDS Backlight Power Setting (P15)
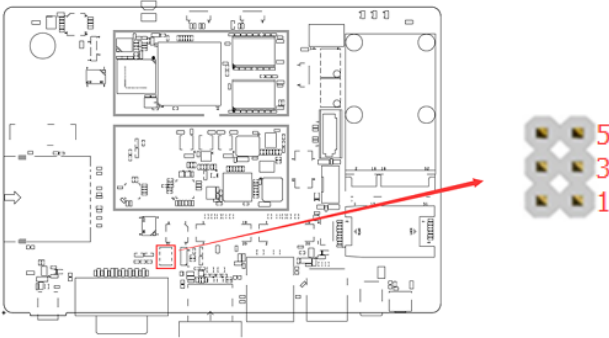
Function	Pin closed	Illustration
3.3V (default)	1-2	
5V	3-4	
12V	5-6	

2.5   Connectors Quick Reference

Function	Connector Name	Page
RTC Lithium Cell Connector	P2	16
COM RS-232/422/485 Selection	SW3	16
COM RS-232/422/485 Port	P10	17
LVDS Display Connector	P8, P22	18
UART Connector	P17	19
LVDS Backlight Control Connector	P9	19
Audio Line-In & Line-Out Connector	P14	20
USB Hub Connector	P13	20
Digital I/O (GPIO) Connector	P18	21
SATA Power Connector	P21	21
CAN Bus Connector	P25	22
I²C Connector	P26	
DC Power Input	P1	- -
SD Card Slot	P3	- -
HDMI Port	P12	- -
GbE LAN Port	P4	- -
Dual USB 2.0 Type-A Port	P5	- -
Mini-USB OTG Port	P6	- -
NGFF M.2 E2230 Slot	P7	- -
Mini-PCIe Slot	P23	- -
SIM Card Socket	P24	- -
SATA II Connector	P20	- -
System Reset Button	SW2	- -
Factory Use Only	SW1, P11, P19	- -

2.5.1   RTC Lithium Cell Connector (P2)
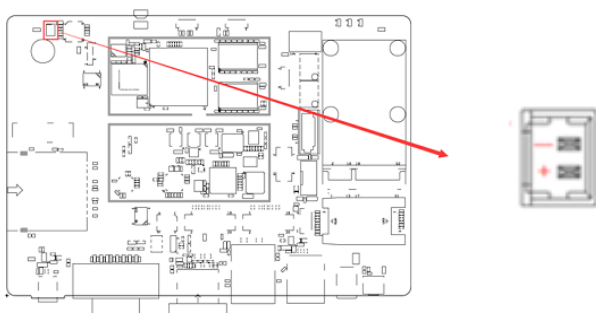
Pin	Assigment	Pin	Assigment
1	RTC_VCC	2	Ground

2.5.2   COM RS-232/422/485 Selection (SW3)
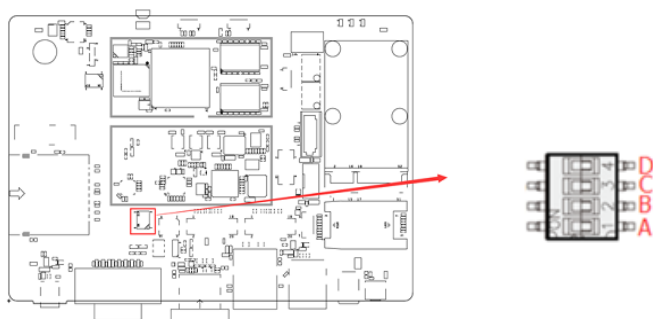
Panel Type	A	B	C	D
RS-422 Full Duplex	NC	ON	ON	ON
Pure RS232 (3T/5R)	NC	OFF	ON	ON
RS-485 Half Duplex (TX Low-Active)	NC	ON	OFF	ON
RS-485 Half Duplex (TX High-Active)	NC	OFF	OFF	ON
RS-422 Full Duplex	NC	ON	ON	OFF
Pure RS232 (1T/1R)	NC	OFF	ON	OFF
RS-485 Half Duplex	NC	ON	OFF	OFF
Shutdown (Default)	NC	OFF	OFF	OFF

2.5.3   COM RS-232/422/485 Port (P10)
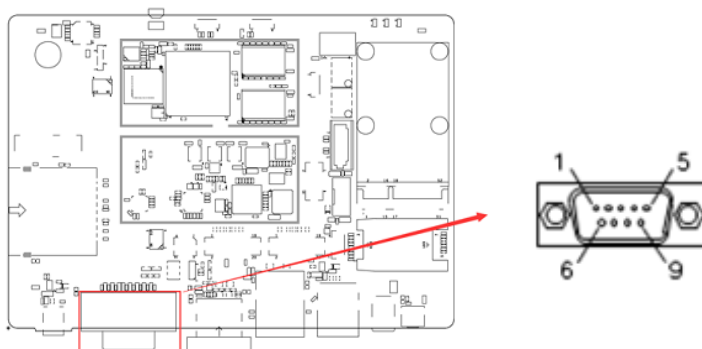
Refer to the SW3 setting for RS-232/422/485 mode selection.

Pin	Assigment	Pin	Assigment
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

iBASE

2.5.4   LVDS Display Connector (P8, P22)
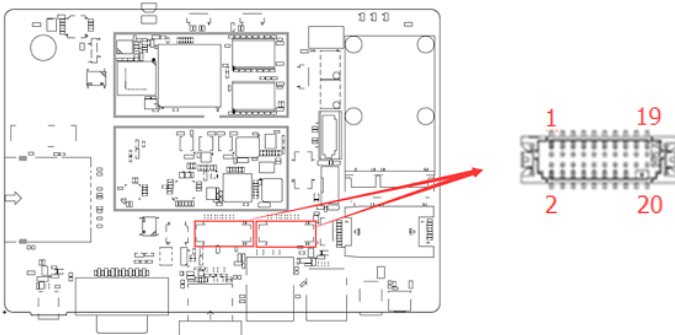
Pin	Assigment	Pin	Assigment
1	LCD1_TX0_P	2	LCD1_TX0_N
3	Ground	4	Ground
5	LCD1_TX1_P	6	LCD1_TX1_N
7	Ground	8	LCD_VDD
9	LCD1_TX3_P	10	LCD1_TX3_N
11	LCD1_TX2_P	12	LCD1_TX2_N
13	Ground	14	Ground
15	LCD1_CLK_P	16	LCD1_CLK_N
17	BTL_PWM	18	LCD_VDD
19	BKLT_VCC	20	BKLT_VCC

2.5.5 UART Connector (P17)
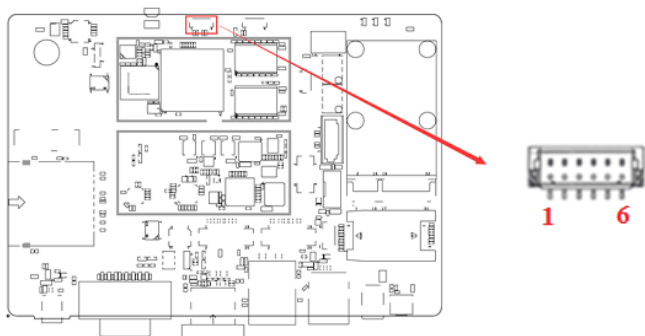
Pin	Assigment	Pin	Assigment
1	UART2_TXD	4	UART3_TXD
2	UART2_RXD	5	UART3_RXD
3	Ground	6	Ground

2.5.6 LVDS Backlight Control Connector (P9)
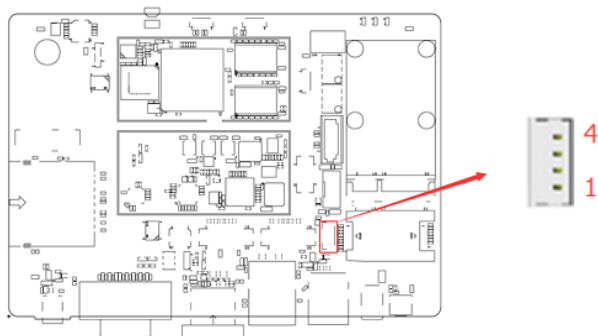
Pin	Assigment	Pin	Assigment
1	BKLT_VCC	3	LCD_BKLT_PWM
2	LCD_BKLT_EN	4	Ground

2.5.7 Audio Line-In & Line-Out Connector (P14)
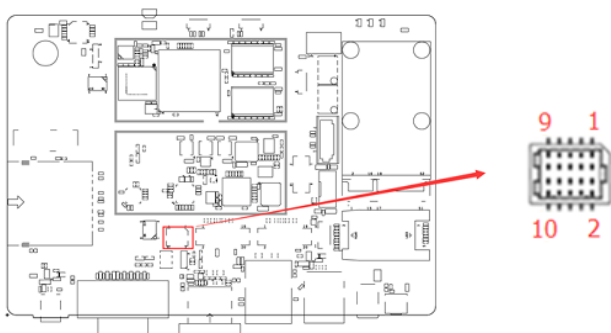
Pin	Assigment	Pin	Assigment
1	NC	2	Ground
3	LINE_IN_R	4	Ground
5	LINE_IN_L	6	Ground
7	Ground	8	LINE_OUT_L
9	Ground	10	LINE_OUT_R

2.5.8 USB Hub Connector (P13)
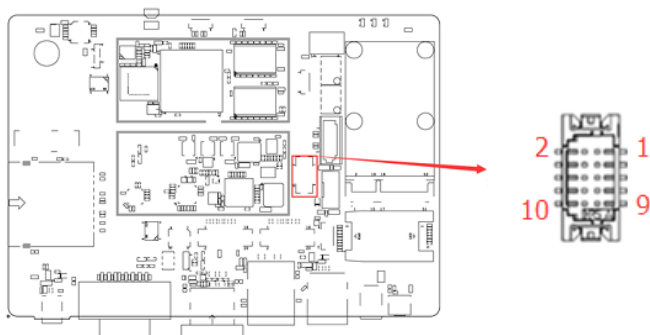
Pin	Assigment	Pin	Assigment
1	Ground	2	NC
3	USB1_DP	4	USB2_POWER
5	USB1_DM	6	USB2_DM
7	USB1_POWER	8	USB2_DP
9	NC	10	Ground

2.5.9   Digital I/O (GPIO) Connector (P18)
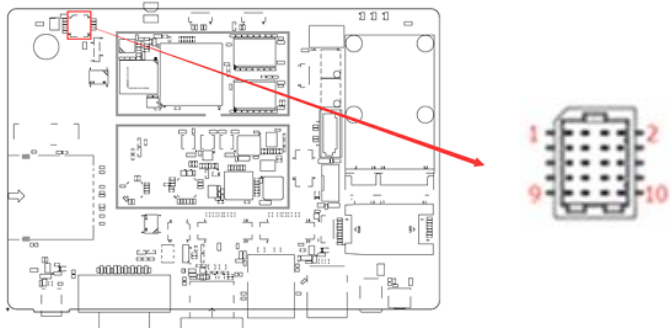
Pin	Assigment	Pin	Assigment
1	3.3V	2	DIO5
3	DIO1	4	DIO6
5	DIO2	6	DIO7
7	DIO3	8	DIO8
9	DIO4	10	Ground

2.5.10   SATA Power Connector (P21)
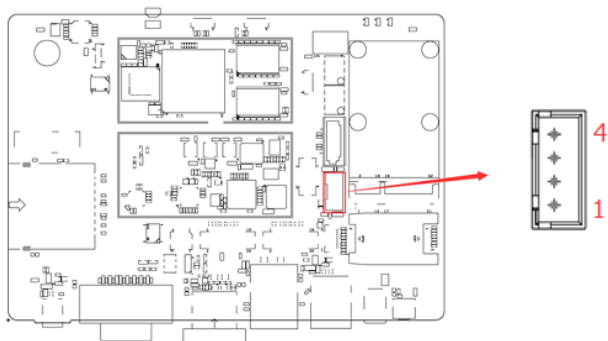
Pin	Assigment	Pin	Assigment
1	5V	3	Ground
2	Ground	4	12V

2.5.11 CAN Bus Connector (P25)
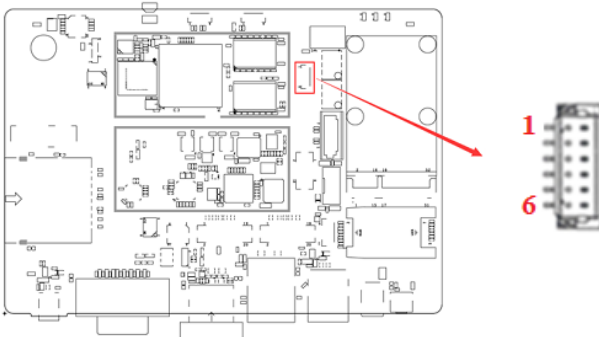
Pin	Assigment	Pin	Assigment
1	CAN1_TXD	4	CAN2_TXD
2	CAN1_RXD	5	CAN2_RXD
3	Ground	6	Ground

2.5.12 I²C Connector (P26)
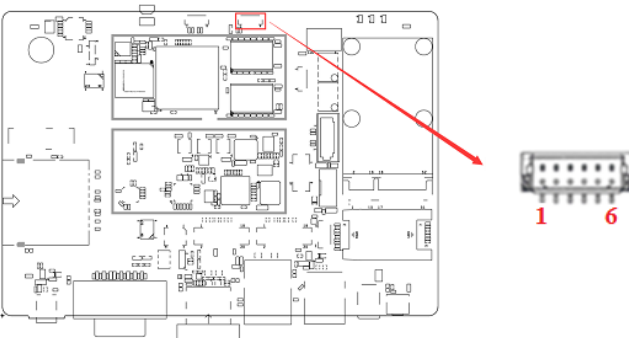
Pin	Assigment	Pin	Assigment
1	VCC	4	I2C3_SCL
2	TP_INT_B	5	I2C3_SDA
3	TP_RST_B	6	GND

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

iBASE

3.1   Make a Recovery SD Card

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

Basically, IBR117 is preloaded with O.S (Android / Linux) into eMMC by default. Connect the TFT-LCD with IBR117 (or HDMI), and 12V power directly.

This chapter guides you to make a recovery boot-up SD card. IBASE optionally provides 7" /15" / 21.5" LVDS panel and HDMI display for you to prepare the software application pre-development easily under Linux platform.

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: 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 boot mode to burning state.
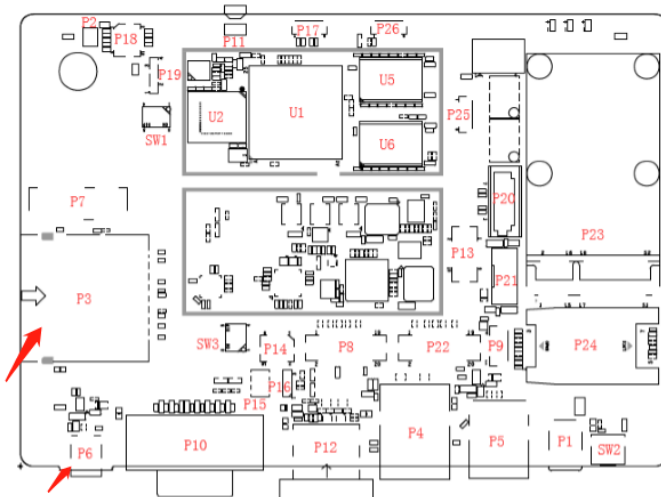
3. Copy the image to the directory **mfgtools-2.3\Profiles\Linux\OS Firmware\files\ibr117**.
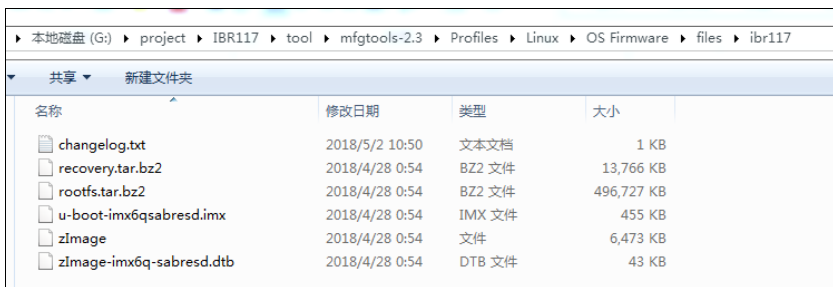
4. Select the script of **mfgtools sdcard recovery**.

iBASE
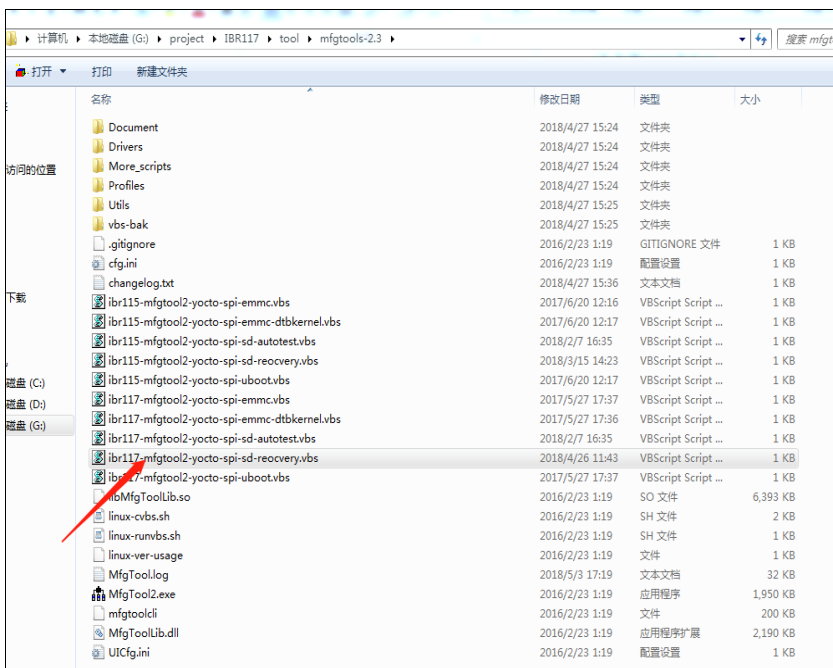
5. Start burning image to SD card. When the image buring finishes, the recovery SD card is ready.
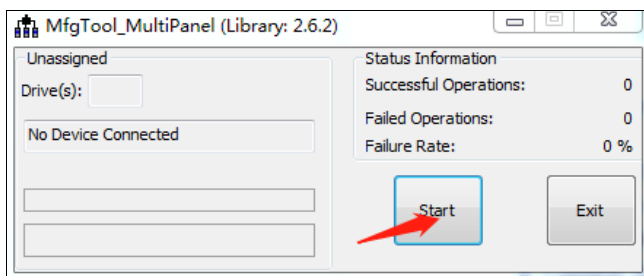
3.1.2   Upgrade Firmware through the Recovery SD Card

1. Copy the image files (what you want to upgrade) to U disk **/IBR117/recovery/**.
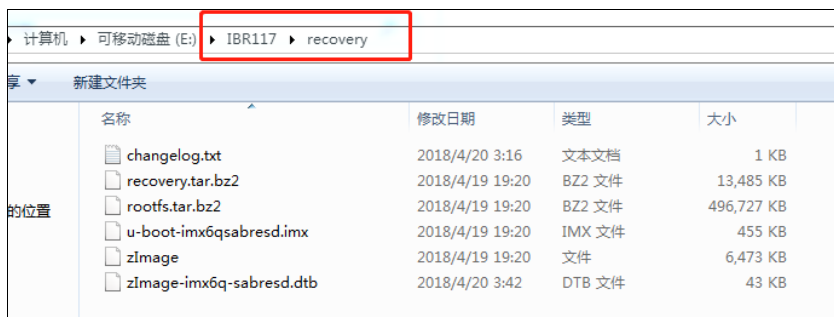
2. Insert the recovery SD card and U disk to the board for upgrade.

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

IBR117 supports HDMI output by default.

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

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

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

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

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

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

4.   If you use LVDS 21.5" to display, run the command below.

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

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 SD Card
- Booting with your SD Card

4.1   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 IBR117 source file **ibr117.tar.xz** into "/home/" folder.

4.2   Installing Toolchain

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

4.3   Building U-Boot

```
cd /home/ibr117
cd  ibr117-x11/tmp/work/imx6dlsabresd-poky-linux-gnueabi/u-boot-imx/2016.03-r0/git
source /opt/poky/1.8/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
make mx6dlsabresd_defconfig
make
```

4.4   Building Kernel

```
cd /home/ibr117
cd ibr117-x11/tmp/work/imx6dlsabresd-poky-linux-gnueabi/linux-imx/4.1.15-r0/git
source /opt/poky/1.8/environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
make imx_v7_defconfig
make zImage LOADADDR=0x10008000
make imx6dl-sabresd.dtb
```

iBASE

4.5   Building RAMdisk Image (Optional)

1.   Enter IBR117.

```
cd /home/ibr117
```

2.   Running the script below.

```
./build.sh ibr117
```

3.   See the image in the "release" directory.

```
ls release/ibr117-20170519-030011/
rootfs.tar.bz2  u-boot-imx6dlsabresd.imx  zImage  zImage-imx6dl-sabresd.dtb
```

4.6   Install Linux to SD Card

1.   Refer to the section *3.1 Make a Recovery SD Card* in chapter 3.

2.   Run the make card script below.

```
./make_sd.sh /dev/sdd 20150814-111827
```

3.   The appearance of the sentence "SD card for booting create done" indicates the installation is successfully done.

4.   Pull out the SD card.

4.7   Booting with Your SD Card

Insert SD card to your board and insert special COM port dongle to boot from SD. Connect a debug cable to debug port with serial port 115200/N/8/1 setting on your PC's serial port program, such as hyperterminal/teraterm. Connect LVDS 15" display panel. Power on and you will see U-Boot prompt.

If you want to change to another monitor (or display panel) type, refer to the section *3.2 Display Parameter Setting in Kernel* to set video output command.

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
```

iBASE

D. USB (flash disk) Test

Insert the USB flash disk then assure it is in IBR117 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 IBR117 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 IBR117 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 -D plughw:1,0
// record mp3 by audio (ALC5640)
arecord -f cd $basepath/b.mp3 -D plughw:1,0
```

iBASE

I.   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
```

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);
```

iBASE

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.   SATA (hard disk) Test

Insert the hard disk then assure it is in IBR117 device list.

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

- **Reading & writing test**

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

- **Hard disk speed test**

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

# hard disk read speed
dd if=$HD_DIR/$i/test of=/dev/null bs=1M oflag=nocache
```

IBASE

M.   CAN Test

- **Checking CAN GPIO**

```
#config can pin to gpio
/home/root/testscript/can/memtool -32  0x20e0208 = 5
echo "106"  > /sys/class/gpio/export
#config gpio to out mode
echo "out" > /sys/class/gpio/gpio106/direction
#set gpio to 1
echo  1    > /sys/class/gpio/gpio106/value
#set gpio to 1
echo  0    > /sys/class/gpio/gpio106/value
#config gpio to in mode
echo "in" > /sys/class/gpio/gpio106/direction
#get gpio value
ret0_0=`cat /sys/class/gpio/gpio106/value`
```

- **Testing CAN**

```
#config can parrameter
ip link set can0 type can bitrate 125000 triple-sampling on
#enable can
ifconfig can0 up
#disable
ifconfig can0 down
#send data
cangen can0 &
#receive data
candump -n 10 can1 -T 5000
```

N.   3G Test

- **Checking 3G state**

```
#Check UC20 module state and sim state
cat /dev/ttyUSB2  &
```

- **Testing 3G**

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

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

O.   Onboard Connector Types

Function	Connector Name	Onboard Type	Compatible Mating Type for Reference
LVDS Display Connector	P8, P22	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
UART Connector	P17	TechBest WT02M-30002-06132	JST SHR-03V-S-B
LVDS Backlight Control Connector	P9	TechBest 1024041008	Molex 51021-0400
Audio Line-In & Line-Out Connector	P14	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H
USB Hub Connector	P13	Hirose DF13E-10DP-1.25V	Hirose DF13E-10DP-1.25C
Digital I/O (GPIO) Connector	P18	E-Call 0110-01-53101100	JCTC 11002H00-1P Well-Lin, 1010H
CAN Bus Connector	P25	TechBest WT02M-30002-06132	JST SHR-03V-S-B
I^{2}C Connector	P26	TechBest WT02M-30002-06132	JST SHR-03V-S-B

Connector types may be subject to change without prior notice.