

ISR201
Ruggedized Embedded System
with NXP ARM® Cortex®
A9 i.MX6 Dual SoC

User's Manual

Version 1.0
January 2026



Copyright

© 2026 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 that 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, registered trademarks, and brands mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners.

Compliance

CE

This product complies with applicable European Union directives and bears the 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 complies with the limits for a Class B digital device under Part 15 of the FCC Rules. It generates, uses, and can radiate radio frequency energy and may cause interference if not installed and used in accordance with the manufacturer's instructions. This device complies with Part 15 of the FCC Rules; operation is subject to the following two conditions: (1) it may not cause harmful interference, and (2) it must accept any interference received, including interference that may cause undesired operation.

WEEE



This product must not be disposed of as normal household waste, in accordance with the EU directive 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 2 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).

- Hexavalent chromium: 1,000 ppm
- Polybrominated biphenyls (PBBs): 1,000 ppm
- Polybrominated diphenyl ethers (PBDEs): 1,000 ppm
- Cadmium: 100 ppm
- Mercury: 1,000 ppm
- Lead: 1,000 ppm
- Bis(2-ethylhexyl) phthalate (DEHP): 1,000 ppm
- Butyl benzyl phthalate (BBP): 1,000 ppm
- Dibutyl phthalate (DBP): 1,000 ppm
- Diisobutyl phthalate (DIBP): 1,000 ppm

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 heat 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.

Care during use:

- Make sure to connect the correct voltage to the device. Failure to supply the correct voltage could damage the unit.
- If you use an extension cord, make sure it does not exceed the extension cord's ampere rating.
- Always unplug the power cord from the wall outlet before cleaning the device.
- Only use neutral cleaning agents to clean the device.
- Remove dust using compressed air or a vacuum cleaner designed for electronic equipment.

Product Disassembly

Do not disassemble beyond the steps described in this manual. Doing so will void the warranty and may result in damage to the product or personal injury.



WARNING

Attention during use:

- Do not use this product near water.
- Do not spill water or any other liquids on your device.
- Do not place heavy objects on the top of the device.
- Operate this device from the type of power indicated on the marking label. If you are not sure of the type of power available, consult your distributor or local power company.
- Do not walk on the power cord or allow anything to rest on it.
- If you use an extension cord, make sure that the total ampere rating of the product plugged into the extension cord does not exceed its limits.

Avoid Disassembly

You are not suggested to disassemble, repair or make any modification to the device. Disassembly, modification, or any attempt at repair could generate hazards and cause damage to the device, even bodily injury or property damage, and will void any warranty.



CAUTION

Danger of explosion if the internal lithium-ion battery is replaced with an incorrect type. Replace only with the same or an equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

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 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)

Table of Contents

Chapter 1	General Information	1
1.1	Introduction	2
1.2	Features.....	2
1.3	Packing List	2
1.4	Optional Accessories	2
1.5	Specifications.....	3
1.6	Product View.....	錯誤! 尚未定義書籤。
1.7	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 Antenna Installation	10
2.1.3	Mounting Installation.....	10
2.1.4	COM RS-232/422/485 Port.....	11
2.1.5	COM1 & COM2 Port.....	12
2.1.6	GPIO Port.....	12
2.2	Setting the Jumpers	13
2.3	Jumper & Connector Locations on Motherboard	14
2.4	Jumper & Connectors Quick Reference	15
2.4.1	RTC Lithium Cell Connector (P2).....	17
2.4.2	RS-232/422/485 Selection (SW3)	17
2.4.3	COM RS-232/422/485 Port (P10).....	18
2.4.4	USB Hub Connector (P13).....	19
2.4.5	Digital (GPIO) Connector (P18).....	19
2.4.6	SATA Power Connector (P21)	21
2.4.7	CAN Bus Connector (P25).....	21
2.4.8	I ² C Connector (P26).....	22
Chapter 3	Software Setup.....	23
3.1	Make a Recovery SD Card.....	24
3.1.1	Preparing the Recovery SD card to Install Linux / Android image into eMMC	24
3.1.2	Upgrade Firmware through the Recovery SD Card.....	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 U-Boot.....	28
4.1.4	Building Kernel	28
4.1.5	Building RAMdisk Image (Optional)	29
4.1.6	Install Linux to SD Card.....	29
4.1.7	Booting with Your SD Card.....	29
Appendix	30
A.	How to Use GPIO in Linux	31
B.	How to Use Watchdog in Linux	31
C.	eMMC Test	32
D.	USB (flash disk) Test	33
E.	SD Card Test	34
F.	RS-232 Test.....	35
G.	RS-485 Test.....	36
H.	Ethernet Test	37
I.	HDMI Test.....	38
J.	SATA (hard disk) Test.....	40
K.	CAN Test	42
L.	3G Test.....	43

Chapter 1

General Information

The information provided in this chapter includes:

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

1.1 Introduction

ISR201 is an ARM®-based embedded system with NXP Cortex® i.MX6 A9 processor. The device offers 2D, 3D graphics and multimedia acceleration while it also features numerous peripherals that are well suited for industrial applications, including RS-232/422/485, GPIO, USB, USB OTG, LAN, HDMI for a Full HD display, M.2 E2230 for wireless connectivity and mini-PCIe for expansion.



1.2 Features

- NXP ARM® Cortex® A9 i.MX6 Dual 1 GHz processor
- Video output through HDMI
- 1 GB DDR3, 4 GB eMMC and SD socket
- Rich I/O interface with RS-232/422/485, GPIO, USB, USB-OTG, 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, BitBlit for 2D and OpenVG1.1 hardware accelerators

1.3 Packing List

Your product package should include the items listed below. Drivers and this user manual are downloadable from our website.

- ISR201

1.4 Optional Accessories

IBASE provides optional accessories as listed below. Please contact us or your dealer if you would like to order any item(s).

- Power Adapter
- WiFi & BT Combo (M.2 card)
- USB OTG Cable (USB-81)
- VESA Mounting Kit (75 x 75 / 100 x 100)
- DIN Rail Mounting Kit

1.5 Specifications

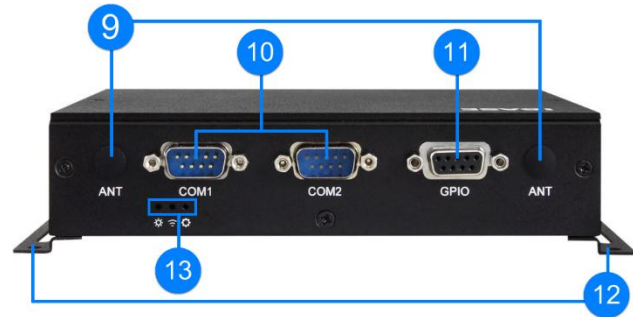
Product Name	ISR201
System Motherboard	IBR117 3.5" SBC
System	
Operating System	<ul style="list-style-type: none"> • Yocto 2.5 • Android 6.0
CPU Type	NXP Cortex™ A9 i.MX6 Dual Core 1 GHz SoC
CPU Speed	1 GHz
Memory	<ul style="list-style-type: none"> • System memory: 1 GB DDR3 • Data Memory: 4 GB eMMC
Video Codec	<ul style="list-style-type: none"> • 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 CTP
RTC	Yes
Wireless	WiFi / BT / 3G / LTE module (Optional)
Power Supply	12V DC-In
Watchdog Timer	Yes (256 segments, 0, 1, 2...128 secs)
Dimensions (W x H x D)	172 x 36 x 162 mm (6.78" x 1.42" x 6.38")
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 1080P at 60 Hz)
LAN	1 x RJ45 GbE LAN
USB	<ul style="list-style-type: none"> • 2 x USB 2.0 Type A • 1 x USB OTG via mini-USB Type B
Serial	<ul style="list-style-type: none"> • 1 x COM RS-232/422/485 port (DB9 male connector) • 2 x COM RS-232 ports (DB9 male connector)
Digital IO	8-In/Out
Expansion Slots	<ul style="list-style-type: none"> • 1 x M.2 E-key (2230) 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)
Relative Humidity	10 ~ 90%, non-condensing

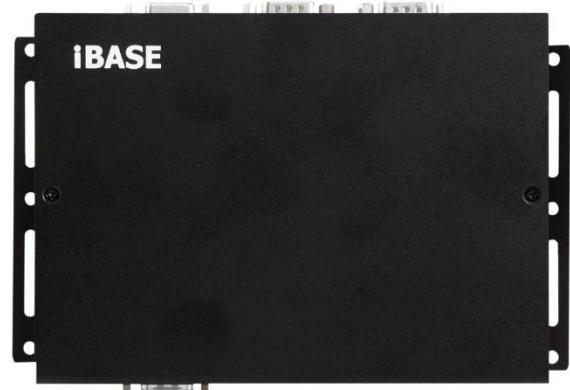
All specifications are subject to change without prior notice.

1.6 Product View

I/O View



Top View



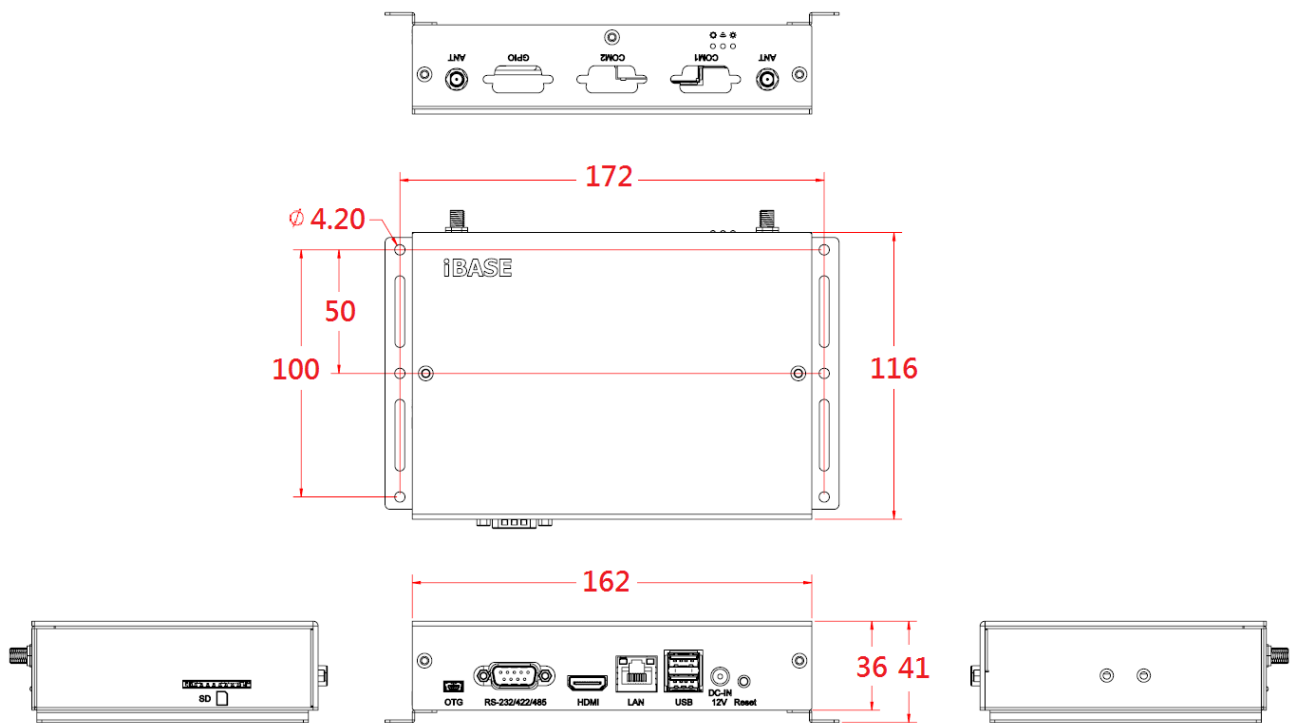
No.	Name	No.	Name
1	USB OTG Port	8	DIN Rail Mounting Holes
2	COM RS-232/422/485 Port	9	Antenna Holes
3	HDMI Port	10	COM1 & COM2 RS-232 Ports
4	GbE LAN Port	11	GPIO Port
5	USB 2.0 Ports	12	Wall Mount Kit
6	DC Jack	13	LED Indicators (from left to right: Power, Wireless, Programmable Setting)
7	Reset Button	14	SD Card Slot

DIN Rail Mounting View (Optional)



1.7 Dimensions

Unit: mm



Chapter 2

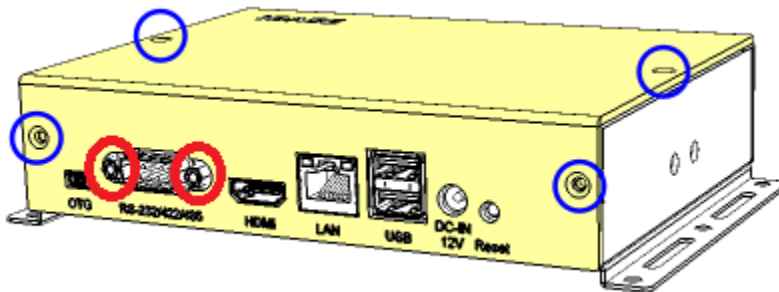
Hardware Configuration

This section contains general information about:

- Installations
- Jumper and connectors

2.1 Installations

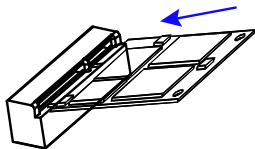
Before installing any card or module into the device, firstly remove the 2x RS-232/422/485 jack screws [in red circle] and then the 4x screws [in blue circle] as shown in the picture below to take away the device cover.



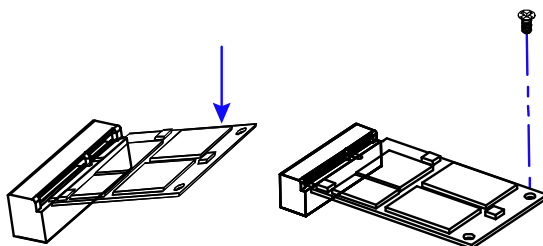
2.1.1 Mini-PCIe & M.2 Cards Installation

To install the mini-PCIe & NGFF M.2 card, remove the device cover first as mentioned above, locate the slot inside the device, and perform the following steps.

1. Align the keys of the mini-PCIe card with that of the mini-PCIe interface, and insert the card slantwise.
(Insert the M.2 card in the same way.)



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



2.1.2 WiFi Antenna Installation

Insert the WiFi 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 if 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).

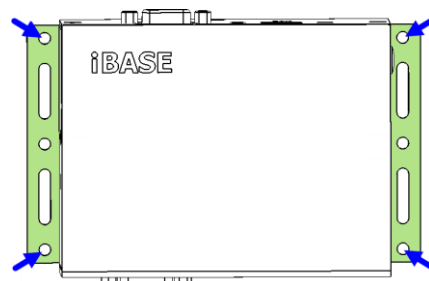
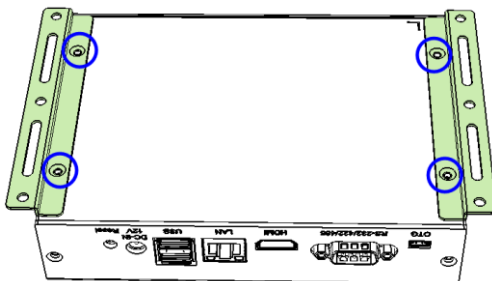
2.1.3 Mounting Installation

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 the weight of the device plus the weight of the suspending cables attached to the system. Use the following methods for mounting your system:

Wall Mounting Installation

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



2.1.4 COM RS-232/422/485 Port



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- (B)	DATA- (B)
2	RX	TX+ (A)	DATA+ (A)
3	TX	RX+ (A)	NC
4	DTR	RX- (B)	NC
5	Ground	Ground	Ground
6	DSR	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

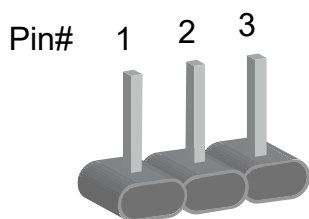


Pin	Signal Name	Pin	Signal Name
1	DIO1	6	DIO5
2	DIO2	7	DIO6
3	DIO3	8	DIO7
4	DIO4	9	DIO8
5	Ground		

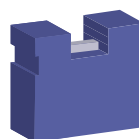
2.2 Setting the Jumpers

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.

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.

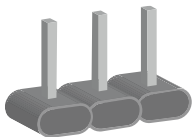
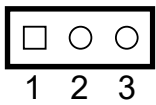
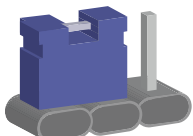
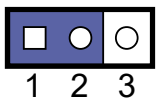
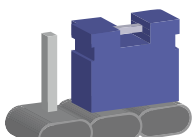
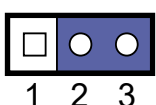


A 3-pin jumper



A jumper cap

Refer to the illustration below to set jumpers.

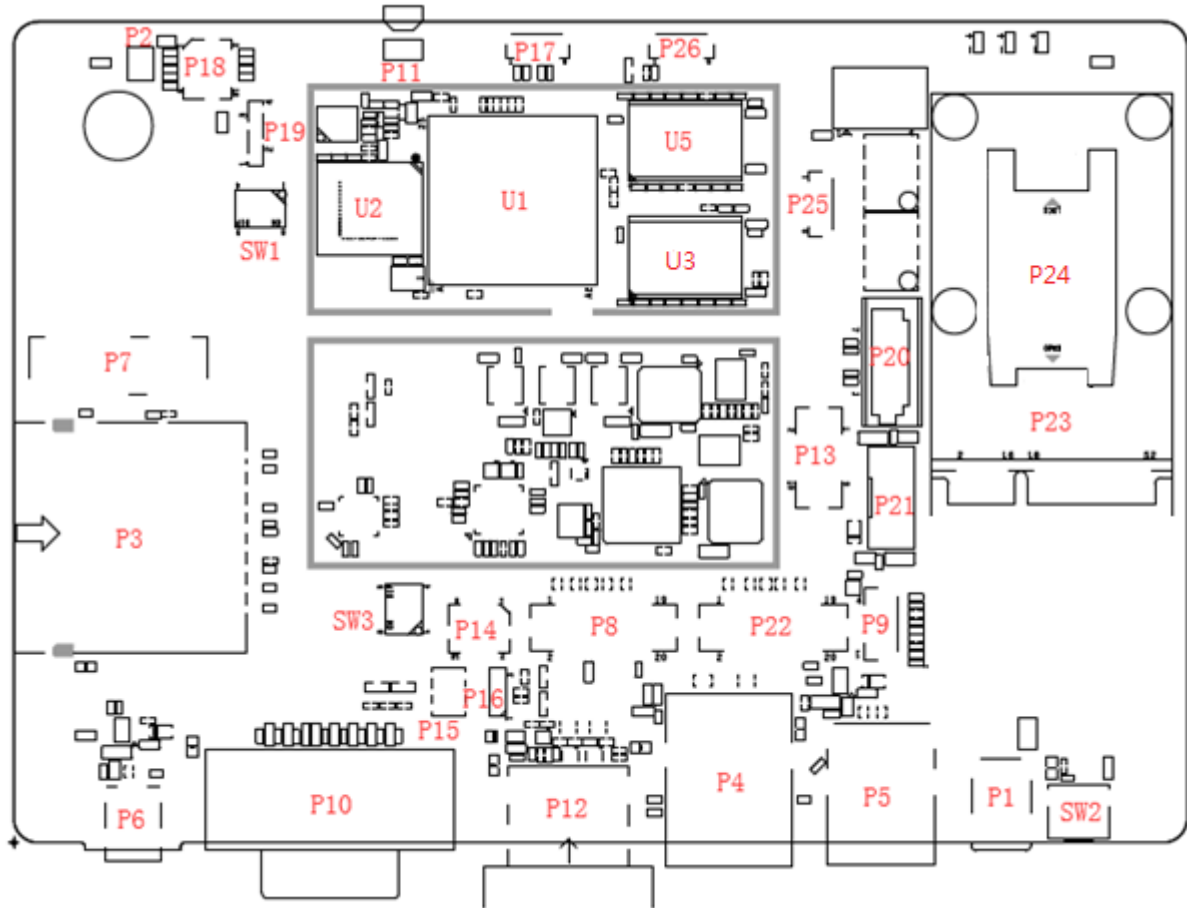
Jumper Setting	Physical View	Pin Assignment
Open		 1 2 3
1-2		 1 2 3
2-3		 1 2 3

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: IBR117



2.4 Jumper & Connectors Quick Reference

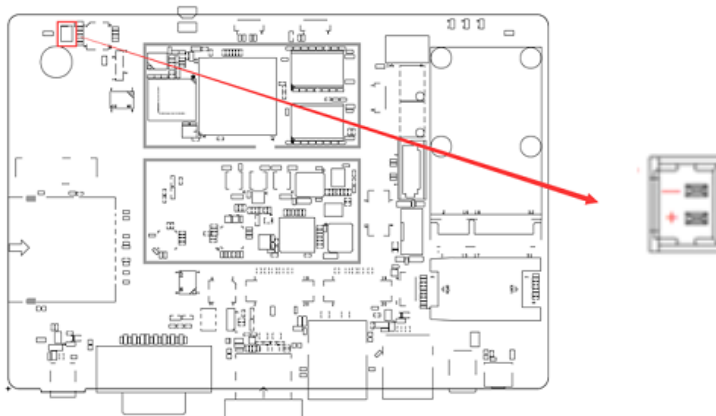
Connectors:

Function	Connector Name	Page
RTC Lithium Cell Connector	P2	17
COM RS-232/422/485 Selection	SW3	17
COM RS-232/422/485 Port	P10	18
USB Hub Connector	P13	19
Digital I/O (GPIO) Connector	P18	19
SATA Power Connector	P21	21
CAN Bus Connector	P25	21
I ² C Connector	P26	22
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	--
LVDS Display Connector	P8, P22	--
UART Connector	P17	--
LVDS Backlight Control Connector	P9	--
Audio Line-In & Line-Out Connector	P14	--
Factory Use Only	SW1, P11, P19	--

Jumpers:

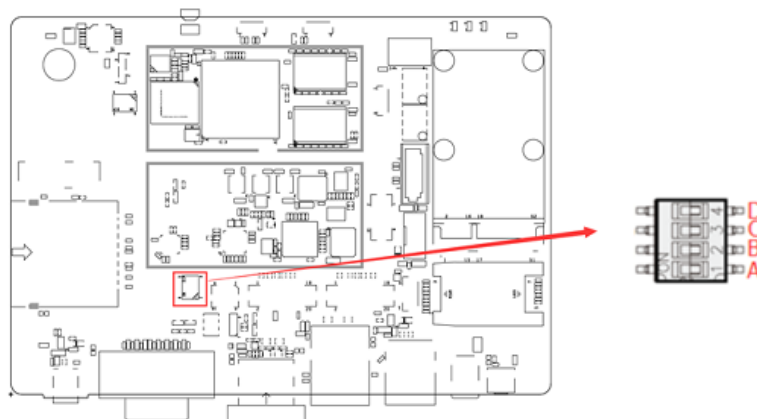
Function	Connector Name	Page
LVDS Power Setting	P16	--
LVDS Backlight Power Setting	P15	--

2.4.1 RTC Lithium Cell Connector (P2)



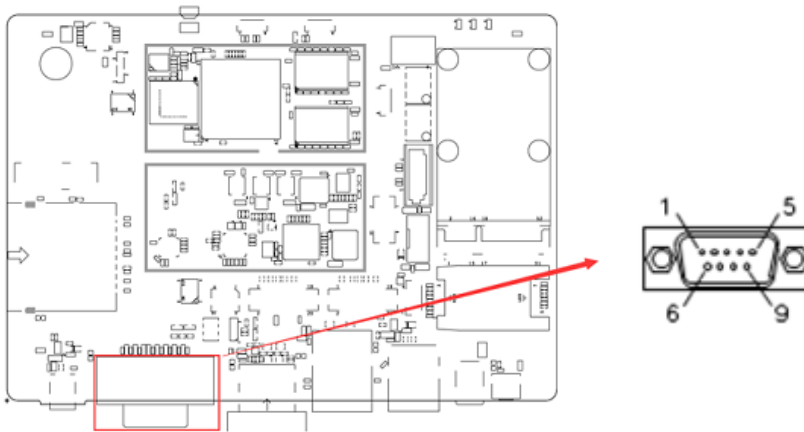
Pin	Signal Name	Pin	Signal Name
1	RTC_VCC	2	Ground

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



Panel Type	A	B	C	D
RS-422 Full Duplex	NC	ON	ON	ON
Pure RS232 (3T/5R) (Default)	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	NC	OFF	OFF	OFF

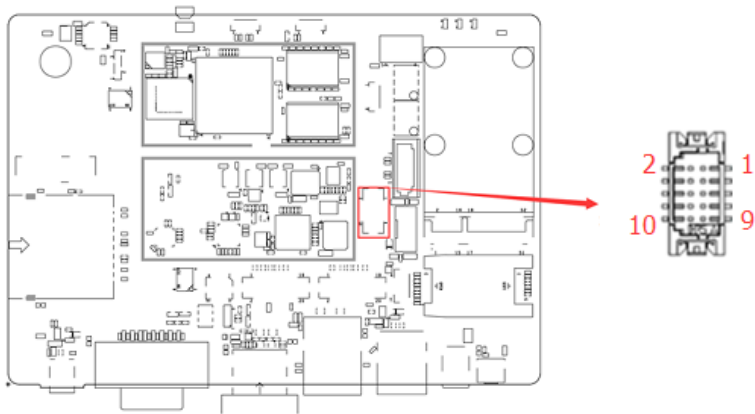
2.4.3 COM RS-232/422/485 Port (P10)



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		

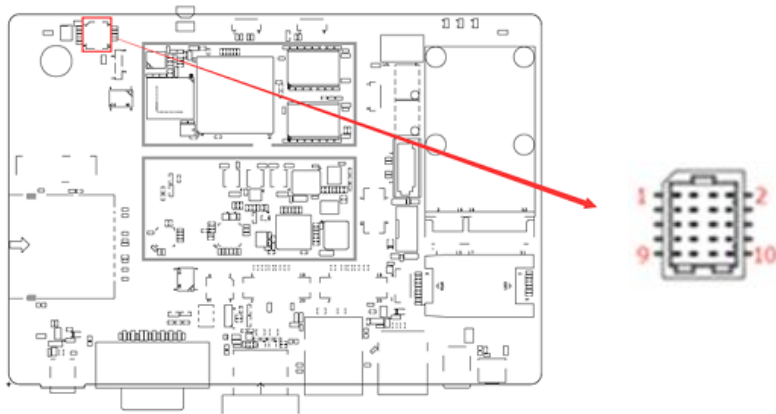
Pin	Assignment		
	RS-232	RS-422	RS-485
1	DCD	TX- (B)	DATA- (B)
2	RX	TX+ (A)	DATA+ (A)
3	TX	RX+ (A)	NC
4	DTR	RX- (B)	NC
5	Ground	Ground	Ground
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	NC	NC	NC

2.4.4 USB Hub Connector (P13)



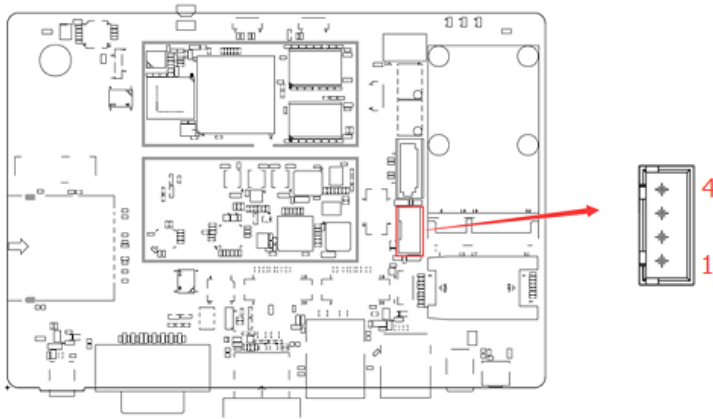
Pin	Signal Name	Pin	Signal Name
1	Ground	2	NC
3	USB1_DP	4	USB2_POWER
5	USB1_DM	6	USB2_DM
7	USB1_POWER	8	USB2_DP
9	NC	10	Ground

2.4.5 Digital (GPIO) Connector (P18)



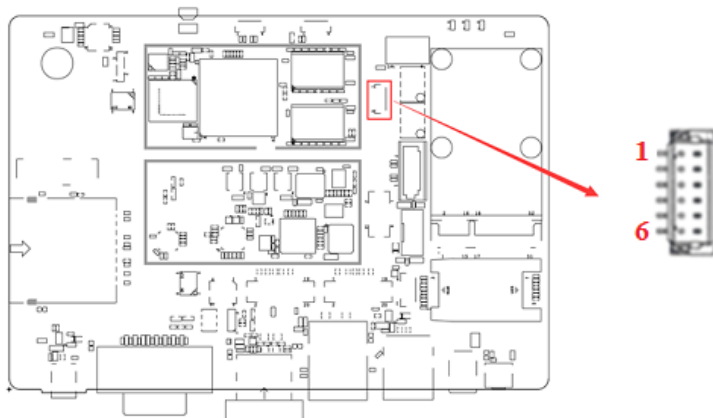
Pin	Signal Name	Pin	Signal Name
1	3.3V	2	DIO5
3	DIO1	4	DIO6
5	DIO2	6	DIO7
7	DIO3	8	DIO8
9	DIO4	10	Ground

2.4.6 SATA Power Connector (P21)

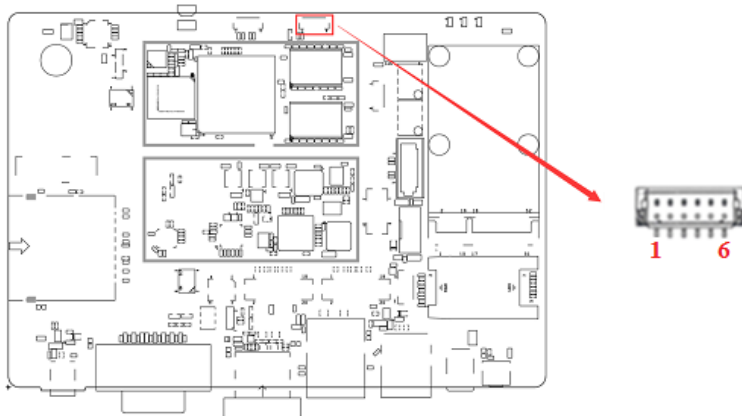


Pin	Signal Name	Pin	Signal Name
1	5V	3	Ground
2	Ground	4	12V

2.4.7 CAN Bus Connector (P25)



Pin	Signal Name	Pin	Signal Name
1	CAN1_H	4	CAN2_H
2	CAN1_L	5	CAN2_L
3	Ground	6	Ground

2.4.8 I²C Connector (P26)

Pin	Signal Name	Pin	Signal Name
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 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 who have an IBASE standard image file only.

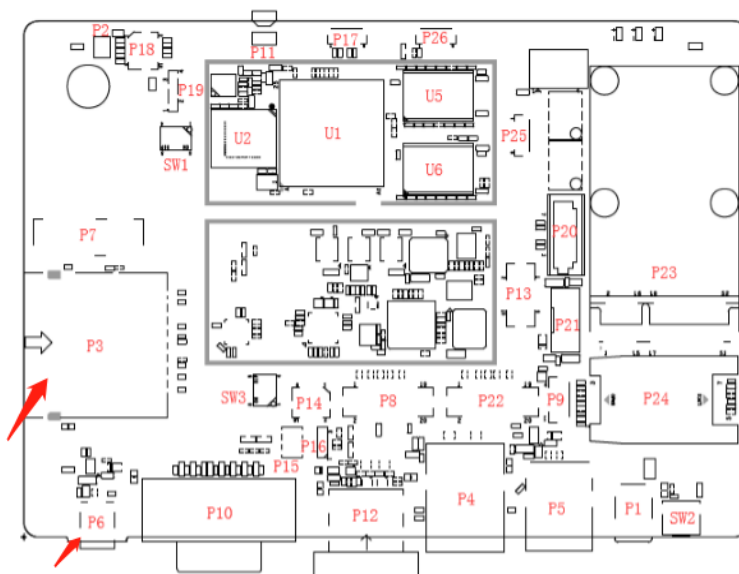
Basically, ISR201 is preloaded with an OS (Android / Linux) into eMMC by default. Connect the TFT-LCD with ISR201 (or HDMI), and 12V power directly.

This chapter guides you to make a recovery boot-up microSD card. IBASE provides HDMI display for you to prepare the software application pre-development easily under Linux.

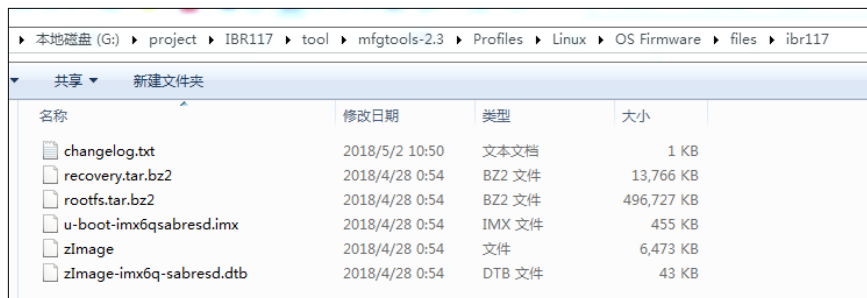
3.1.1 Preparing the Recovery SD card to Install Linux / Android image into eMMC

Note: All data in the eMMC will be erased.

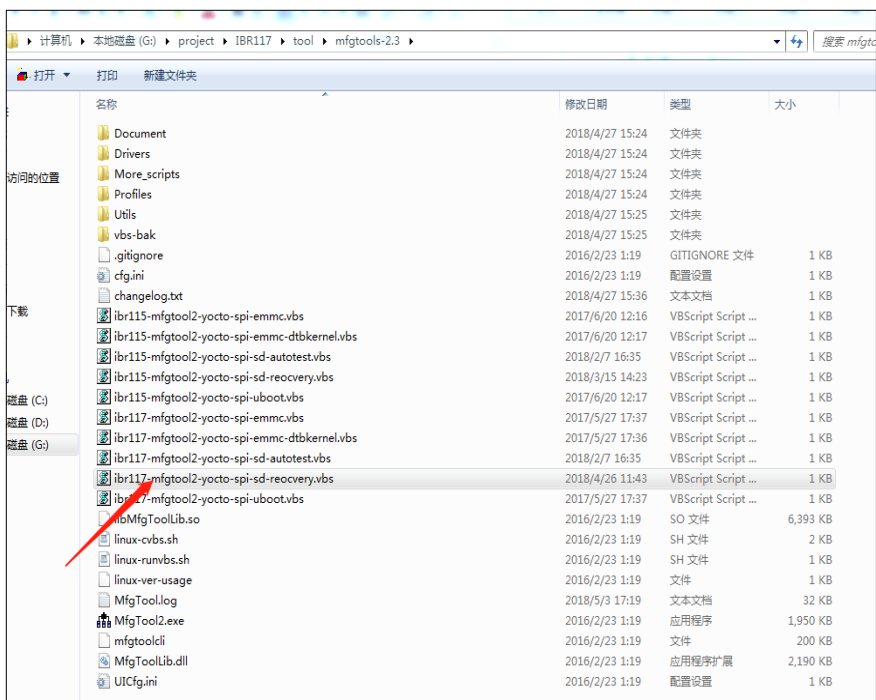
1. System requirements:
 - Operating System: Windows 7 or later
 - Tool: mfgtools-2.3
 - SD card: 4GB or greater in size
2. Insert your SD card into the board (i.e. the P3 connector), connect the board to a PC through the mini-USB port (i.e. the P6 connector), and change the boot mode to burning state.



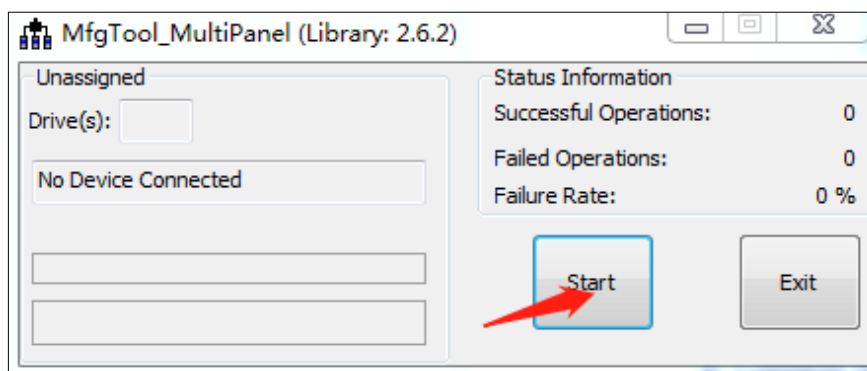
- Copy the image to the directory **mfgtools-2.3\Profiles\Linux\OS Firmware\files\ibr201**.



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

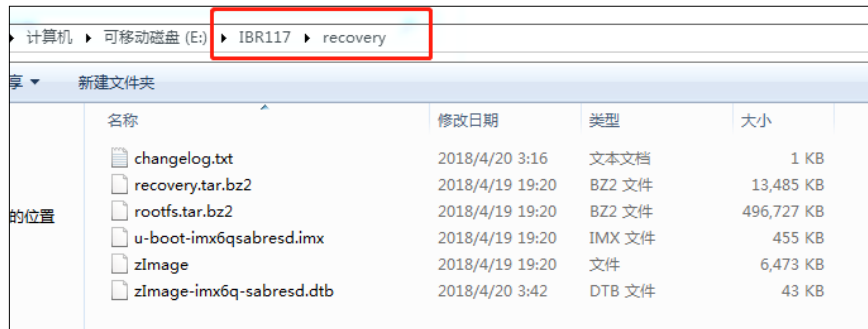


- Write the image to SD card. When the image burning finishes, the recovery SD card is ready.



3.1.2 Upgrade Firmware through the Recovery SD Card

1. Copy the image files (what you want to upgrade) to USB flash disk **/ISR201/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.

Chapter 4

BSP Source Guide

This chapter is intended 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 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 ISR201 source file **ISR201.tar.xz** into `"/home/"` folder.

4.1.2 Installing Toolchain

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

4.1.3 Building U-Boot

```
cd /home/ISR201
cd ISR201-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
```

4.1.4 Building Kernel

```
cd /home/ISR201
cd ISR201-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
```

4.1.5 Building RAMdisk Image (Optional)

1. Go to the ISR201 directory.

```
cd /home/ISR201
```

2. Running the script below.

```
./build.sh ISR201
```

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

```
ls release/ISR201-20170519-030011/  
rootfs.tar.bz2 u-boot-imx6dlsabresd.imx zImage zImage-imx6dl-  
sabresd.dtb
```

4.1.6 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

4.1.7 Booting with Your SD Card

Insert the SD card into 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/ Tera Term. Connect a display panel. Power on and you will see U-Boot prompt.

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/gpio32/direction
echo 0 > /sys/class/gpio/gpio32/value
echo 1 > /sys/class/gpio/gpio32/value

# GPIO example 2: Input
echo 32 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio32/direction
cat /sys/class/gpio/gpio32/value
```

B. How to Use Watchdog in Linux

```
#include <fcntl.h>
#include <linux/watchdog.h>
#include <stdio.h>
#include <sys/ioctl.h>
#include <unistd.h>

int main(void)
{
    int fd = open("/dev/watchdog", O_WRONLY);
    if (fd < 0) { perror("open"); return 1; }

    struct watchdog_info ident;
    int status = 0;
    int timeout_val = 10;
    int dummy = 0;

    ioctl(fd, WDIOC_GETSUPPORT, &ident);
    ioctl(fd, WDIOC_GETSTATUS, &status);
    ioctl(fd, WDIOC_GETTIMEOUT, &timeout_val);
    ioctl(fd, WDIOC_SETTIMEOUT, &timeout_val);
    ioctl(fd, WDIOC_KEEPAIVE, &dummy);

    close(fd);
    return 0;
}
```

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 ISR201 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 on the USB storage device being used.

- **Read, write, and check**

```
USB_DEV=/dev/sda1
USB_DIR=/mnt/usb
mkdir -p $USB_DIR
mount $USB_DEV $USB_DIR
dd if=/dev/urandom of=/var/data1 bs=1024k count=100
dd if=/var/data1 of=$USB_DIR/data2 bs=1024k count=100
cmp $USB_DIR/data2 /var/data1
umount $USB_DIR
```

- **USB speed test**

```
USB_DEV=/dev/sda1
USB_DIR=/mnt/usb
mkdir -p $USB_DIR
mount $USB_DEV $USB_DIR
TEST_FILE=$USB_DIR/dd_test.bin
dd if=/dev/zero of=$TEST_FILE bs=1M count=256 conv=fsync
dd if=$TEST_FILE of=/dev/null bs=1M count=256
rm -f $TEST_FILE
sync
umount $USB_DIR
```

E. SD Card Test

When ISR201 is booted from eMMC, SD card is “/dev/mmcblk1” and able to see by “ls /dev/mmcblk1*” command:

```
ls /dev/mmcblk1*
```

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

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

- Read, write, and check

```
SD_DEV=/dev/mmcblk1p1
SD_DIR=/mnt/sd
mkdir -p $SD_DIR
mount $SD_DEV $SD_DIR
dd if=/dev/urandom of=/var/data1 bs=1024k count=100
dd if=/var/data1 of=$SD_DIR/data2 bs=1024k count=100
cmp $SD_DIR/data2 /var/data1
umount $SD_DIR
```

- SD card speed test

```
SD_DEV=/dev/mmcblk1p1
SD_DIR=/mnt/sd
mkdir -p $SD_DIR
mount $SD_DEV $SD_DIR
dd if=/dev/zero of=$SD_DIR/test bs=1M count=1000 conv=fsync
echo 3 > /proc/sys/vm/drop_caches
dd if=$SD_DIR/test of=/dev/null bs=1M count=1000
rm -f $SD_DIR/test
sync
umount $SD_DIR
```

F. RS-232 Test

```
struct termios tio;
if (tcgetattr(fd, &tio) != 0) { perror("tcgetattr"); close(fd); return 1; }

cfmakeraw(&tio);
cfsetispeed(&tio, B115200);
cfsetospeed(&tio, B115200);

tio.c_cflag |= (CLOCAL | CREAD);
tio.c_cflag &= ~PARENB;
tio.c_cflag &= ~CSTOPB;
tio.c_cflag &= ~CSIZE;
tio.c_cflag |= CS8;

if (tcsetattr(fd, TCSANOW, &tio) != 0) { perror("tcsetattr"); close(fd); return 1; }

const char *msg = "RS232 test\r\n";
write(fd, msg, strlen(msg));

char buf[128];
int n = read(fd, buf, sizeof(buf) - 1);
if (n > 0) { buf[n] = 0; printf("RX: %s\n", buf); }

close(fd);
return 0;
```

G. RS-485 Test

```
struct termios tio;
if (tcgetattr(fd, &tio) != 0) { perror("tcgetattr"); close(fd); return 1; }

cfmakeraw(&tio);
cfsetispeed(&tio, B115200);
cfsetospeed(&tio, B115200);

tio.c_cflag |= (CLOCAL | CREAD);
tio.c_cflag &= ~PARENB;
tio.c_cflag &= ~CSTOPB;
tio.c_cflag &= ~CSIZE;
tio.c_cflag |= CS8;

if (tcsetattr(fd, TCSANOW, &tio) != 0) { perror("tcsetattr"); close(fd); return 1; }

struct serial_rs485 rs485;
memset(&rs485, 0, sizeof(rs485));
rs485.flags |= SER_RS485_ENABLED;
ioctl(fd, TIOCSRS485, &rs485);

const char *msg = "RS485 test\r\n";
write(fd, msg, strlen(msg));

char buf[128];
int n = read(fd, buf, sizeof(buf) - 1);
if (n > 0) { buf[n] = 0; printf("RX: %s\n", buf); }

close(fd);
return 0;
```

H. Ethernet Test

- **Ethernet FTP test**

```
SERVER_IP=192.168.1.123
#install lftp
chmod 755 /home/root/testscript/ethernet/lftp_library/lftp
cp /home/root/testscript/ethernet/lftp_library/lftp /usr/bin/
mkdir -p /usr/lib/lftp/4.6.3a
cp /home/root/testscript/ethernet/lftp_library/*.so /usr/lib/lftp/4.6.3a/
cp /home/root/testscript/ethernet/lftp_library/* /lib/
#lftp server
lftp $SERVER_IP <<EOF
put data1
get data1
bye
EOF
```

- **Ethernet Ping test**

```
ping -c 20 $SERVER_IP > /tmp/ethernet_ping.txt
```

- **Ethernet TCP test**

```
#server run: iperf3 -s
iperf3 -c $SERVER_IP -i 1 -t 20 -w 32M -P 4
```

- **Ethernet UDP test**

```
#server run: iperf3 -s
iperf3 -c $SERVER_IP -u -i 1 -t 20 -b 200M
```

I. HDMI Test

- **HDMI display test**

```
#include <fcntl.h>
#include <linux/fb.h>
#include <stdio.h>
#include <sys/ioctl.h>
#include <sys/mman.h>
#include <string.h>
#include <unistd.h>

int main(void)
{
    int x = 10;
    int y = 10;
    unsigned char color_r = 0xFF;
    unsigned char color_g = 0x00;
    unsigned char color_b = 0x00;

    int framebuffer_fd;
    struct fb_fix_screeninfo finfo;
    struct fb_var_screeninfo vinfo;
    long screensize;
    unsigned char *fbp;

    framebuffer_fd = open("/dev/fb2", O_RDWR);
    if (framebuffer_fd < 0) { perror("open"); return 1; }

    if (ioctl(framebuffer_fd, FBIOGET_FSCREENINFO, &finfo) < 0) {
        perror("FBIOGET_FSCREENINFO");
        close(framebuffer_fd);
        return 1;
    }

    if (ioctl(framebuffer_fd, FBIOGET_VSCREENINFO, &vinfo) < 0) {
        perror("FBIOGET_VSCREENINFO");
        close(framebuffer_fd);
        return 1;
    }

    screensize = vinfo.yres_virtual * finfo.line_length;

    fbp = (unsigned char *)mmap(0, screensize,
        PROT_READ | PROT_WRITE, MAP_SHARED, framebuffer_fd, 0);
    if (fbp == MAP_FAILED) {
        perror("mmap");
        close(framebuffer_fd);
        return 1;
    }

    memset(fbp, 0x00, screensize);
}
```

```

if (x >= 0 && x < (int)vinfo.xres && y >= 0 && y < (int)vinfo.yres) {
    long location =
        (x + vinfo.xoffset) * (vinfo.bits_per_pixel / 8) +
        (y + vinfo.yoffset) * vinfo.line_length;

    if (vinfo.bits_per_pixel == 32) {
        fbp[location + 0] = color_b;
        fbp[location + 1] = color_g;
        fbp[location + 2] = color_r;
        fbp[location + 3] = 0xFF;
    } else if (vinfo.bits_per_pixel == 24) {
        fbp[location + 0] = color_b;
        fbp[location + 1] = color_g;
        fbp[location + 2] = color_r;
    }
}

munmap(fbp, screensize);
close(framebuffer_fd);
return 0;
}

```

- **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

```

J. SATA (hard disk) Test

Insert the SATA hard disk (or SSD) and make sure it is detected by the system.

Note:

This operation may damage the data stored on the hard disk/SSD. Before starting the test, make sure there is no critical data on the SATA storage device being used.

- **Check device detection**

Run the following commands to confirm the SATA device name (commonly **/dev/sda**):

```
lsblk
```

```
dmesg | tail -n 50
```

Example: if the SATA partition is **/dev/sda1**, use it in the commands below.

- **Read, write, and check (data integrity test)**

```
SATA_DEV=/dev/sda1
```

```
SATA_DIR=/mnt/sata
```

```
mkdir -p $SATA_DIR
```

```
mount $SATA_DEV $SATA_DIR
```

```
# Create a random test file (100 MB)
```

```
dd if=/dev/urandom of=/tmp/sata_src.bin bs=1M count=100
```

```
# Write to SATA disk
```

```
dd if=/tmp/sata_src.bin of=$SATA_DIR/sata_dst.bin bs=1M conv=fsync
```

```
# Compare files
```

```
cmp /tmp/sata_src.bin $SATA_DIR/sata_dst.bin
```

```
# Cleanup
```

```
rm -f /tmp/sata_src.bin $SATA_DIR/sata_dst.bin
```

```
sync
```

```
umount $SATA_DIR
```

If `cmp` shows no output, the read/write check is successful.

- **SATA speed test (write and read)**

```
SATA_DEV=/dev/sda1
```

```
SATA_DIR=/mnt/sata
```

```
TEST_FILE=$SATA_DIR/sata_speed_test.bin
```

```
mkdir -p $SATA_DIR
```

```
mount $SATA_DEV $SATA_DIR
```

```
# Write speed
```

```
echo 3 > /proc/sys/vm/drop_caches
```

```
time dd if=/dev/zero of=$TEST_FILE bs=1M count=256 conv=fsync
```

```
# Read speed
```

```
echo 3 > /proc/sys/vm/drop_caches
```

```
time dd if=$TEST_FILE of=/dev/null bs=1M count=256
```

```
# Cleanup
```

```
rm -f $TEST_FILE
```

```
sync
```

```
umount $SATA_DIR
```

K. CAN Test

This section describes how to verify CAN GPIO control and CAN bus communication on the ISR201 system.

Note:

Before starting the CAN test, make sure:

- The CAN transceiver hardware is properly installed.
 - CAN_H and CAN_L are correctly wired.
 - A CAN loopback connection or another CAN device is available for communication.
-

1) Checking CAN GPIO

Some CAN interfaces require GPIO control to enable the CAN transceiver. Use the following steps to verify GPIO operation.

```
# Configure CAN pin to GPIO mode
```

```
/home/root/testscript/can/memtool -32 0x20e0208 = 5
```

```
# Export GPIO 106
```

```
echo 106 > /sys/class/gpio/export
```

```
# Set GPIO to output mode
```

```
echo out > /sys/class/gpio/gpio106/direction
```

```
# Set GPIO output high
```

```
echo 1 > /sys/class/gpio/gpio106/value
```

```
# Set GPIO output low
```

```
echo 0 > /sys/class/gpio/gpio106/value
```

```
# Set GPIO to input mode
```

```
echo in > /sys/class/gpio/gpio106/direction
```

```
# Read GPIO value
```

```
cat /sys/class/gpio/gpio106/value
```

If the GPIO value changes correctly, the CAN-related GPIO control is functioning properly.

2) CAN Bus Configuration

Configure the CAN interface parameters before enabling the CAN bus.

```
# Set CAN bitrate
```

```
ip link set can0 type can bitrate 125000 triple-sampling on
```

3) Enable and Disable CAN Interface

```
# Enable CAN interface
```

```
ifconfig can0 up
```

```
# Disable CAN interface
```

```
ifconfig can0 down
```

4) CAN Data Transmission Test

Send CAN data

```
cangen can0 &
```

Receive CAN data

```
candump -n 10 can1 -T 5000
```

If CAN frames are successfully received, the CAN bus communication is working correctly.

L. 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
```