DEBIX SOM A I/O Board User Guide

Version: V1.6（2025-08）

Compiled by: Polyhex Technology Company Limited（http://www.polyhex.net/）

DEBIX SOM A IO Board is a full-featured carrier board specially designed for DEBIX SOM A i.MX 8M Plus core board. It connects DEBIX SOM A through 4 double-sided board-to-board Socket connectors on the front of the board, and supports all functions and interfaces on the core board, providing a comprehensive solution for applications in industrial control, IoT connection and multimedia fields.

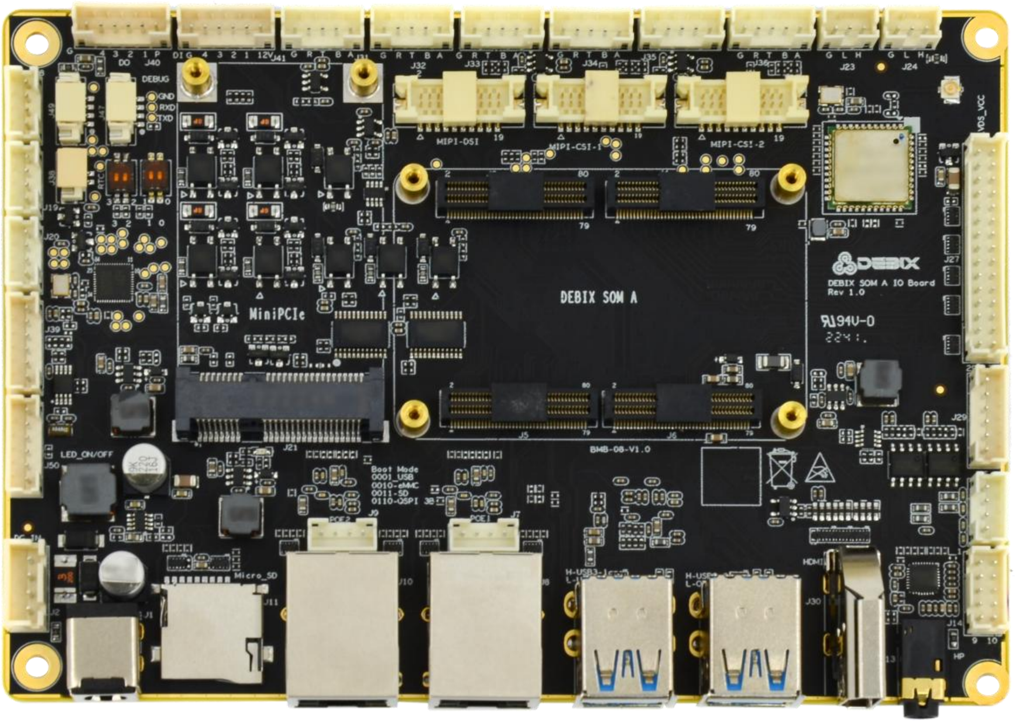


Figure 1 DEBIX SOM A I/O Board

REVISION HISTORY

Rev.	Date	Description
1.0	2023.08.29	First edition
1.1	2023.11.06	Added heat dissipation methods, usage of display screen.
1.2	2025.02.07	1. Added the content of the JTAG test points 2. Updated the part of Technical Support
1.3	2025.05.07	Revised the command and figure in the section 3.1.2.3 USB Flash
1.4	2025.05.14	Revised the section 4.12.Usage of Display Screen
1.5	2025.05.20	Revised the section 3.1.Software Installation
1.6	2025.08.19	Added support for the DEBIX 8-inch MIPI DSI display in Table 48

INDEX

Chapter 1 Security

1.1. Safety Precaution

This document inform how to make each cable connection. In most cases, you will simply need to connect a standard cable.

Table 1 Terms and conventions

Symbol	Meaning
Warning! ⚠	Always disconnect the power cord from the chassis whenever there is no workload required on it. Do not connect the power cable while the power is on. Sudden power surges can damage sensitive electronic components. Only experienced electricians should open the chassis.
Caution! ⚠	Always ground yourself to remove any static electric charge before touching *DEBIX* product. Modern electronic devices are very sensitive to electric charges. Use a grounding wrist strap at all times. Place all electronic components on a static-dissipative surface or in a static-shielded bag.

1.2. Safety Instruction

To avoid malfunction or damage to this product please observe the following:

1．Disconnect the device from the DC power supply before cleaning. Use a damp cloth. Do not use liquid detergents or spray-on detergents.

2．Keep the device away from moisture.

3．During installation, set the device down on a reliable surface. Drops and bumps will lead to damage.

4．Before connecting the power supply, ensure that the voltage is in the required range, and the way of wiring is correct.

5．Carefully put the power cable in place to avoid stepping on it.

6．If the device is not used for a long time, power it off to avoid damage caused by sudden overvoltage.

7．For safety reasons, the device can only be disassembled by professional personnel.

8．If one of the following situations occur, get the equipment checked by service personnel:

- The power cord or plug is damaged.
- Liquid has penetrated into the equipment.
- The equipment has been exposed to moisture.
- The equipment does not work well, or you cannot get it to work according to the user's manual.
- The equipment has been dropped and damaged.
- The equipment has obvious signs of breakage.

9．Do not place the device outside the specified ambient temperature range. This will damage the machine. It needs to be kept in an environment at controlled temperature.

10．Due to the sensitive nature of the equipment, it must be stored in a restricted access location, only accessible by qualified engineer.

DISCLAIMER: Polyhex assuems no liability for the accuracy of any statement of this instructional document.

1.3. Declaration of Compliance

This product has passed the following certifications:

Table 2 Compliance Certification

Symbol	Meaning
CE	This equipment has passed CE certified.
RoHS	This equipment is manufactured in compliance with RoHS regulations.

UK CA	This equipment has passed UKCA certified.
FC	This equipment has passed FCC certified.
PS E	This equipment has passed PSE certified.
✓	This equipment has passed C-Tick certified.
✓	This equipment is manufactured in compliance with RCM regulations.

1.4. Technical Support

1. Visit DEBIX website https://www.debix.io/ where you can find the latest information about the product.

■ Skip Links:

Debix Documentation: https://debix.io/Document/manual.html

Debix Blog: https://debix.io/Software/blog.html

Debix GitHub: https://github.com/debix-tech

2. Contact your distributor, sales representative or Polyhex's customer service center for technical support if you need additional assistance. Please have the following info ready before you call:

- Product name and memory size
- Description of your peripheral attachments
- Description of your software(operating system, version, application software, etc.)
- A complete description of the problem

- The exact wording of any error messages

■ TechSupport Platforms：

Discord Community (recommended): https://discord.com/invite/adaHHaDkH2

Email: teksupport@debix.io

Chapter 2 DEBIX SOM A I/O Board Introduction

DEBIX SOM A IO Board is a carrier board specially designed for DEBIX SOM A, which combining the functions required for industrial control, IoT connection and multimedia.

Main features:

- Feature rich interfaces to take advantage of the i.MX 8M Plus processor to the fullest extent
- Support three boot mode:
- ■ Boot from eMMC on DEBIX SOM A
- ■ Boot from Micro SD Card of DEBIX SOM A I/O Board
- ■ Boot from SPI Nor Flash on DEBIX SOM A I/O Board (reserved)
- The serial ports, CAN and GPIO of the I/O Board are designed with isolation, dedicated to industrial and IoT applications
- Support dual Gigabit Ethernet with POE function (need POE power device module), 2.4GHz & 5GHz Wi-Fi and Bluetooth 5.0

2.1. Overview

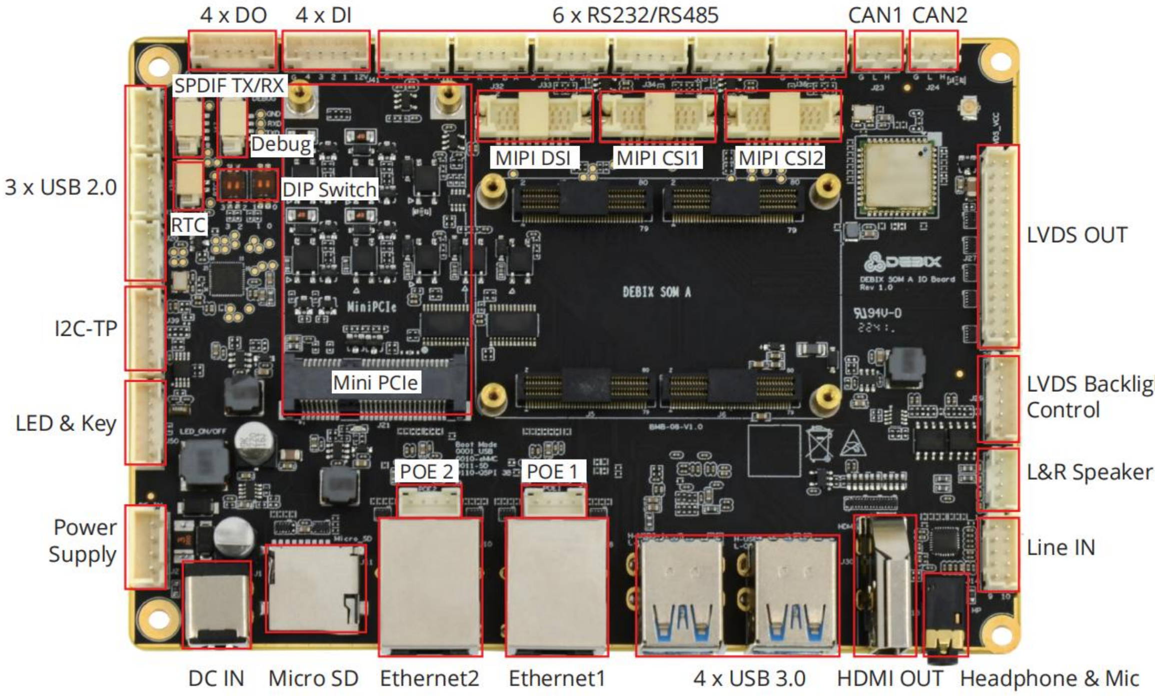


Figure 2 DEBIX SOM A I/O Board Front

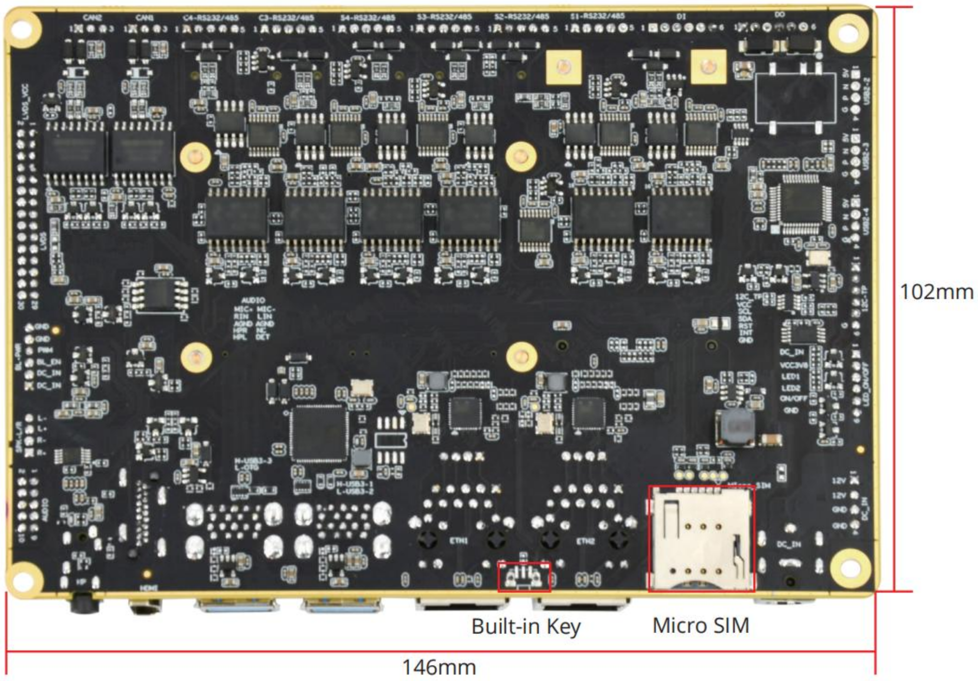


Figure 3 DEBIX SOM A I/O Board Back

DEBIX SOM A I/O Board supports all the functions and interfaces on the DEBIX SOM A core board, the data specifications are as follows:

Table 3 DEBIX SOM A I/O Board specification

System	
Core Board	DEBIX SOM A
Boot Mode	1) Support DEBIX SOM A onboard eMMC boot (default) 2) Support DEBIX SOM A I/O Board Micro SD card boot 3) Support DEBIX SOM A I/O Board SPI Nor Flash boot (reserved)
Communication	
Ethernet	2 x Independent MAC Gigabit Ethernet ports, both support POE power supply (need POE power device module)
Wi-Fi & BT	2.4GHz & 5GHz dual-band Wi-Fi, Bluetooth 5.0, external Wi-Fi&BT 2-in-1 IPEX first generation antenna connector
Video & Audio	
HDMI	1 x HDMI output, the connector is Type A HDMI female
LVDS	1) 1 x LVDS output, single & dual channel 8 bit, the connector is 2*15Pin/2.0mm header 2) 1 x LVDS Backlight control interface, the connector is 1*6Pin/2.0mm header 3) 1 x I2C Touch screen interface, the connector is 1*6Pin/2.0mm header
MIPI DSI	1 x 4Lane MIPI DSI output, the connector is 2*10Pin/1.25mm header
MIPI CSI	2 x 4Lane MIPI CSI input, the connector is 2*10Pin/1.25mm header
Audio	1) 1 x headphone output and microphone input combo interface, the connector is a 3.5mm socket, compatible with the built-in header design 2) 1 x Line in analog input, the connector is 2*5Pin/2.0mm

	header 3) 1 x L&R speaker output, maximum support 3W@4Ω speaker, the connector is 1*4Pin/2.0mm header 4) 1 x SPDIF TX/RX audio interface, the connector is 1*4Pin/1.25mm header
External I/O Interface	
USB	1) 4 x USB 3.0 Host, the connector is double layer Type-A interface 2) 3 x USB 2.0 Host, the connector is 1*4Pin/2.0mm header
Serial Ports	1) 6 x isolated RS232/RS485 (can only choose one of the two) 2) 1 x UART TTL 3.3V system debug serial port
CAN	2 x isolated CAN bus
GPIO	1) 4 x isolated DI, support wet and dry nodes 2) 4 x isolated DO, support wet nodes, compatible with external relay dry nodes
Mini PCIe	1 x Mini PCIe, 1) Support Mini PCIe 4G module, such as Quectel 4G module, built-in SIM card 2) Support Mini PCIe LoRa module 3) Support Mini PCIe expansion modules, such as network card, SATA card, serial port card
DIP Switch	4 x DIP Switch
Slot	1) 1 x Micro SIM pop-up card slot 2) 1 x Micro SD pop-up card slot
LED & Key	1) 1 x Power indicator port 2) 2 x Programmable Control Indicator ports 3) 1 x ON/OFF port

	4) Built-in small key for eMMC upgrade The above share a connector, the connector is 1*6Pin/2.0mm header
DC Block	1) 1 x DC socket for 5.5mm x 2.1mm plug 2) 1 x Built-in power interface, the connector is 1*4Pin/2.54mm header
Power Supply	
Power Input	Default DC 12V/2A power input, support DC 12V~36V wide voltage input
Mechanical & Environmental	
Size (L x W)	146.0mm x 102.0mm (±0.5mm)
Weight	206g (±0.5g)
Operating Temperature	1) Industrial grade: -20°C~70°C 2) Industrial grade: -40°C~85°C (optional)

2.2. Interface

DEBIX SOM A I/O Board has feature rich I/O interfaces, which can fully unlock the potential of the i.MX 8M Plus processor.

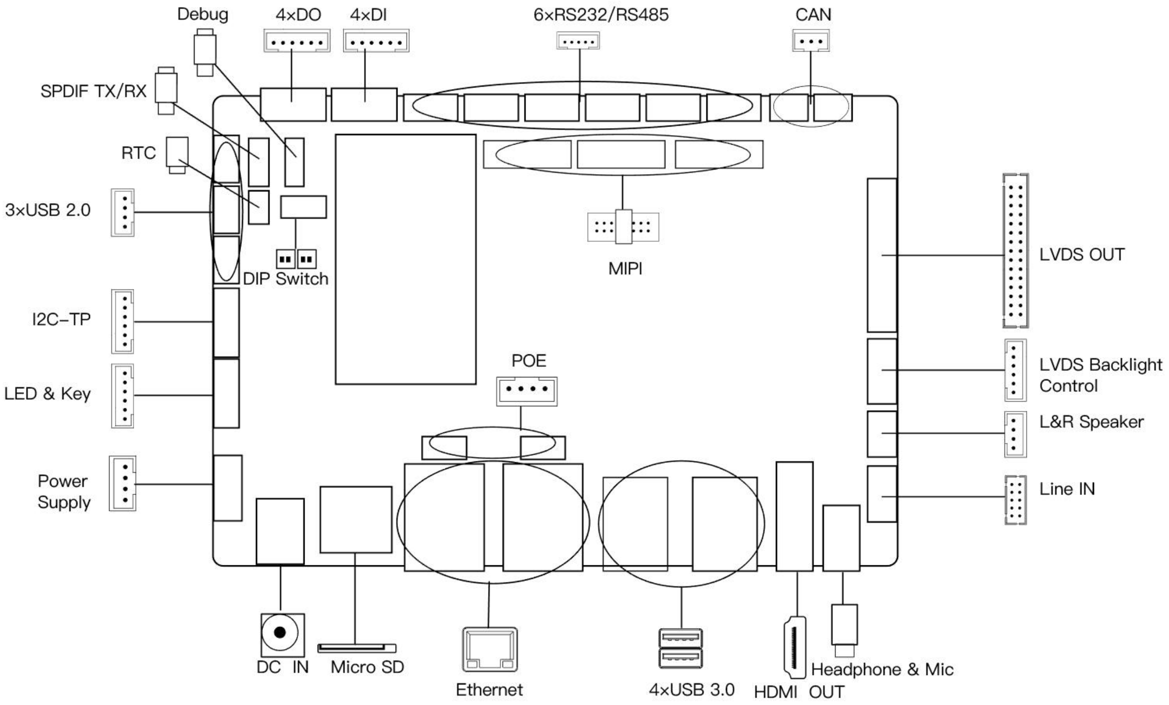


Figure 4 DEBIX SOM A I/O Board interfaces

2.2.1. Power Interface

2.2.1.1. Power Interface

There are 2 power interfaces (J2, J1) on DEBIX SOM A I/O Board, the voltage range is 12-36V.

- One is a DC socket (J1) for 5.5mm x 2.1mm plug
- One is a 1*4Pin/2.54mm header (J2)

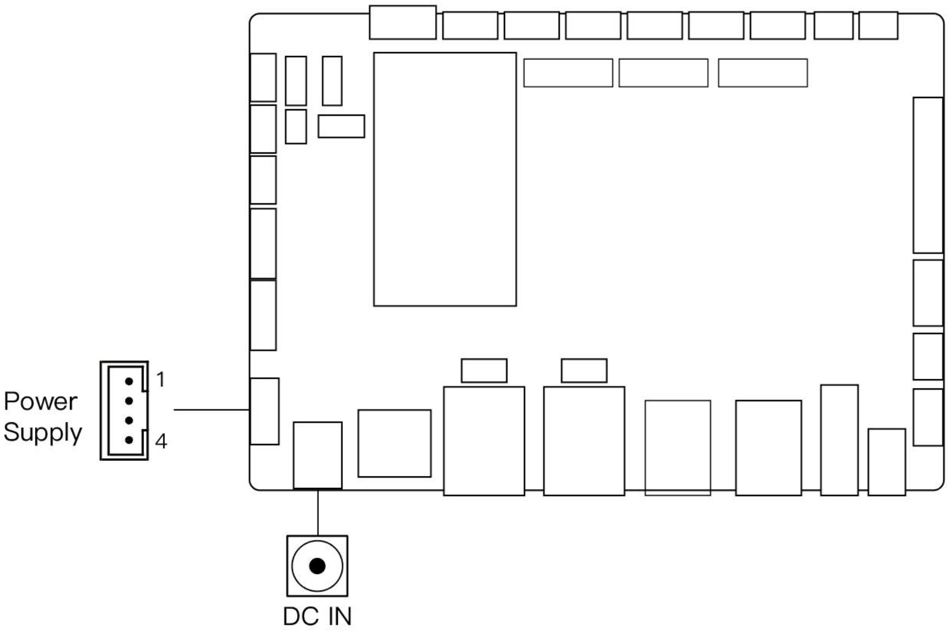


Figure 5 Power Interface

The pin sequence of J1 and J2 is as shown in the figure:

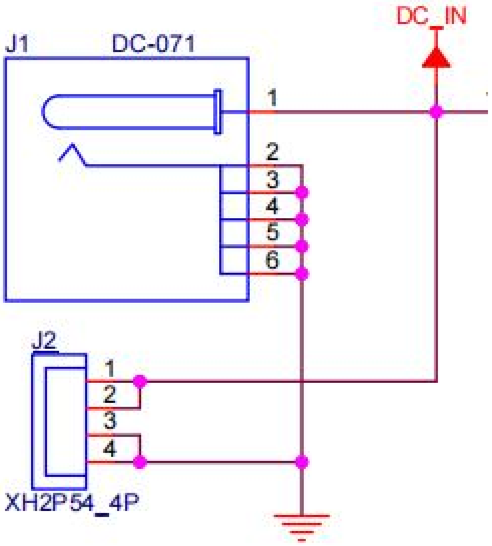


Figure 6 Pin sequence of J1&J2

The J2 interface is defined as follows:

Table 4 Pin definition of J2

Pin	Definition	Description
1	DC_IN	Voltage input range is 12-36V

2	DC_IN	
3	GND	To Ground
4	GND	

2.2.1.2. POE

Two POE interfaces (POE1: J7, POE2: J9) with 1*4Pin/2.0mm header, which need to be connected to POE power device module, and a switch or router with POE function will be connected to the Ethernet interface for DC power supply of DEBIX device.

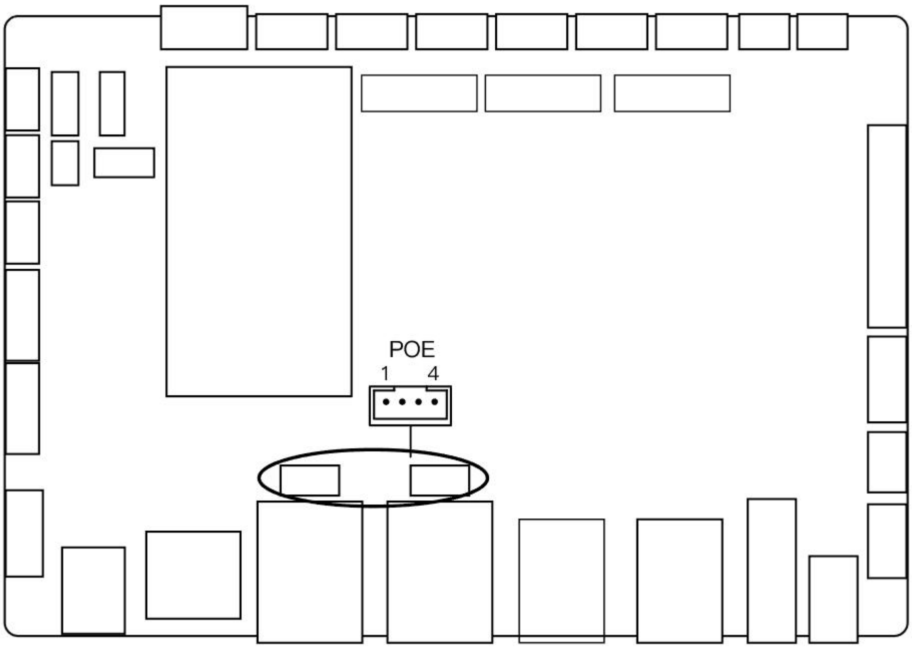


Figure 7 POE Interface

The pin sequence of J7 and J9 is as shown in the figure:

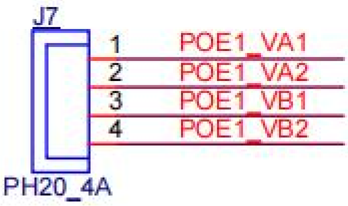


Figure 8 Pin sequence of J7

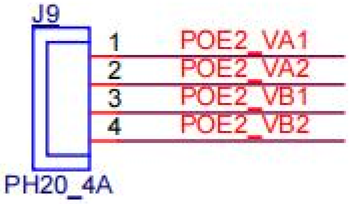


Figure 9 Pin sequence of J9

The J7 interface is defined as follows:

Table 5 Pin definition of J7

Pin	Definition	Description
1	POE1_VA1	POE1 power input pin A1
2	POE1_VA2	POE1 power input pin A2
3	POE1_VB1	POE1 power input pin B1
4	POE1_VB2	POE1 power input pin B2

The J9 interface is defined as follows:

Table 6 Pin definition of J9

Pin	Definition	Description
1	POE2_VA1	POE2 power input pin A1
2	POE2_VA2	POE2 power input pin A2
3	POE2_VB1	POE2 power input pin B1
4	POE2_VB2	POE2 power input pin B2

2.2.2. Ethernet Interface

DEBIX SOM A implements two Ethernet controllers, both of which can operate synchronously.

- ENET_QOS (Ethernet Quality of Service) (ETH1), based on Synopsys proprietary, supports time-sensitive networking (TSN), EEE, Ethernet AVB (IEEE802.1Qav), IEEE1588

- ENET1 (ETH2), Gigabit Ethernet controller, supports EEE, Ethernet AVB (IEEE802.1Qav), IEEE1588 time stamp module, the time stamp module is distributed control for industrial automation applications nodes provide accurate clock synchronization.

Two independent MAC Gigabit Ethernet ports on board (ETH1: J8, ETH2: J10), both support POE power supply (POE power supply module is required), connect DEBIX to the network through the network cable of the RJ45 connector, and there is also a set of status indicators below the interface to display the signal, one is Link and the other is Active.

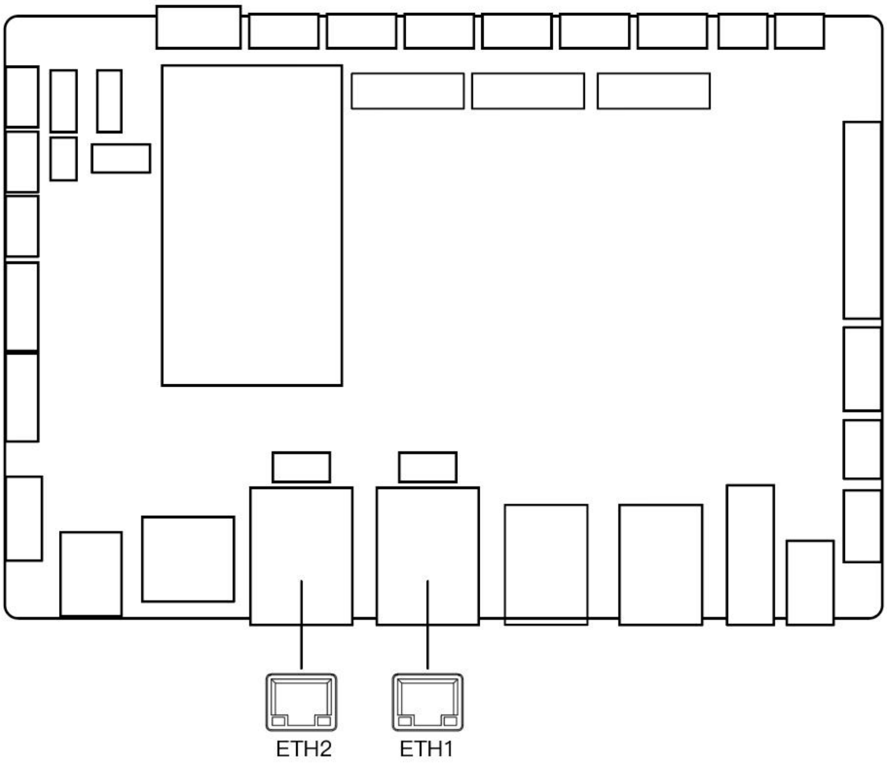


Figure 10 Ethernet Interface

Table 7 Description of Gigabit Ethernet Port Status Indicator

LED	Color	Description
Link	Green	Light, the network cable is plugged in, network connection status is good
Active	Yellow	Blinking, network data is being transmitted

2.2.3. USB Interface

2.2.3.1. USB3.0

There are four USB 3.0 Host A type interfaces (J16, J17) on DEBIX SOM A IO Board, and the connectors are double-layer Type-A interfaces.

- Among them, three USB 3.0 HOST (H-USB3-1, L-USB3-2, H-USB3-3) are CPU USB controller PHY extended by USB3.0 HUB, they share a CPU USB controller PHY; L-OTG is another USB controller PHY of CPU, the default is USB 3.0 Host when it works normally, and is used as USB OTG when USB boot is selected.
  - H-USB3-1 represents the USB 3.0 HOST port on the upper layer of J16
  - L-USB3-2 represents the USB 3.0 HOST port on the lower layer of J16
  - H-USB3-3 represents the USB 3.0 HOST port on the upper layer of J17
  - L-OTG means the USB 3.0 port on the lower layer of J17

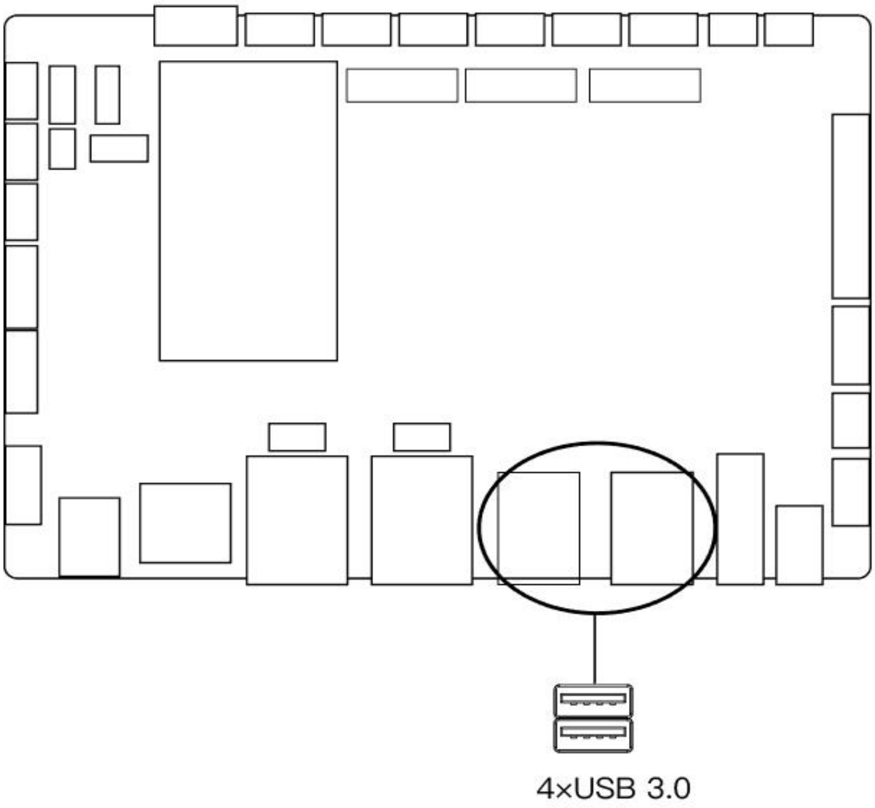


Figure 11 USB 3.0

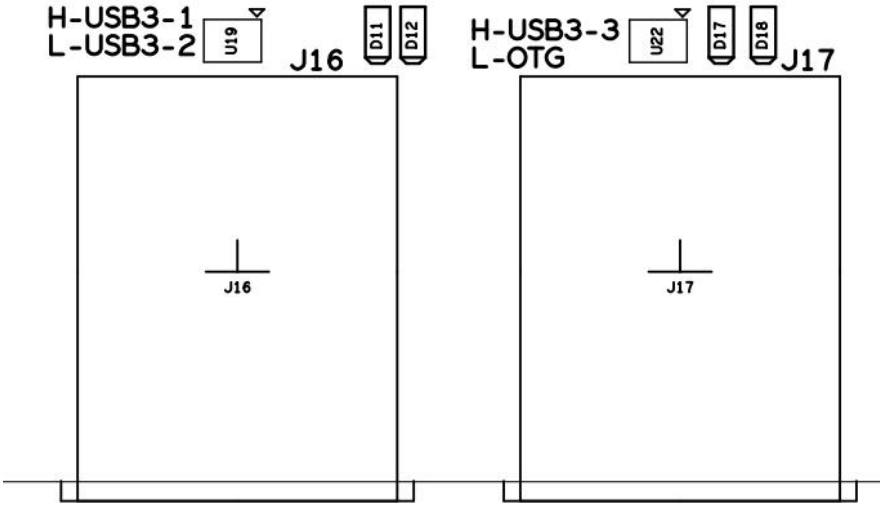


Figure 12 USB 3.0 silkscreen

The pin sequence of USB 3.0 is as shown in the figure:

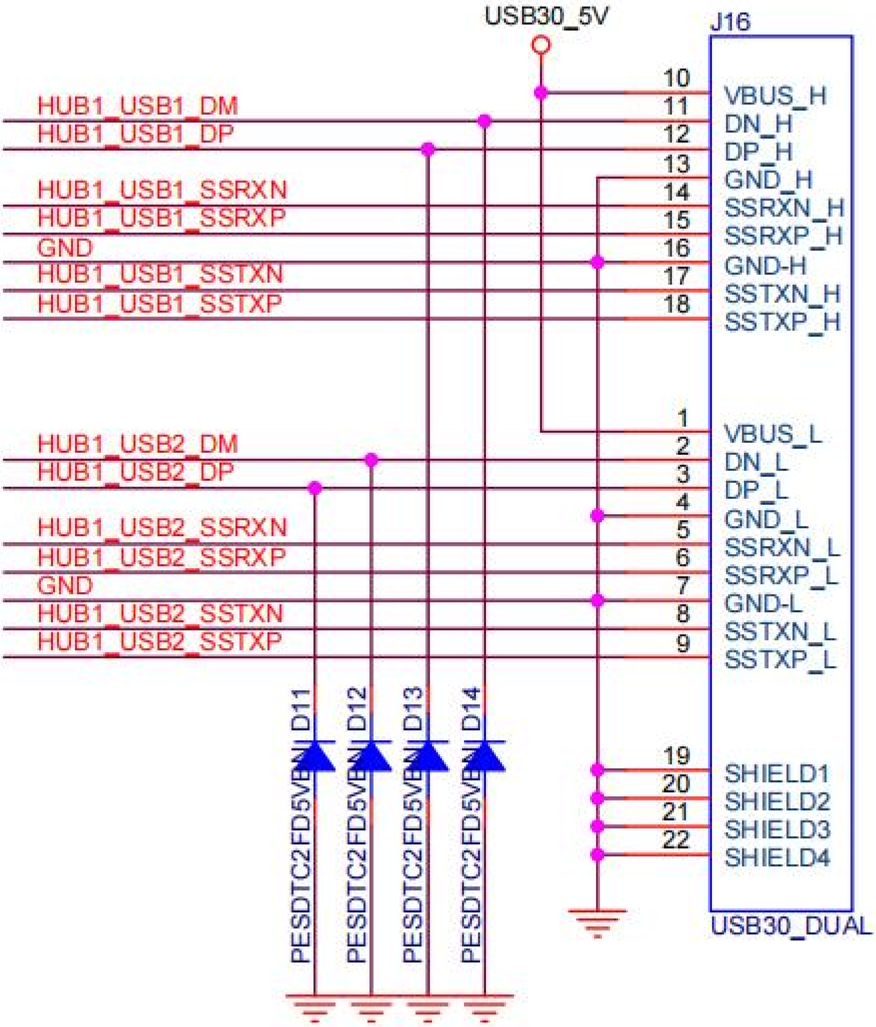


Figure 13 Pin sequence of J16

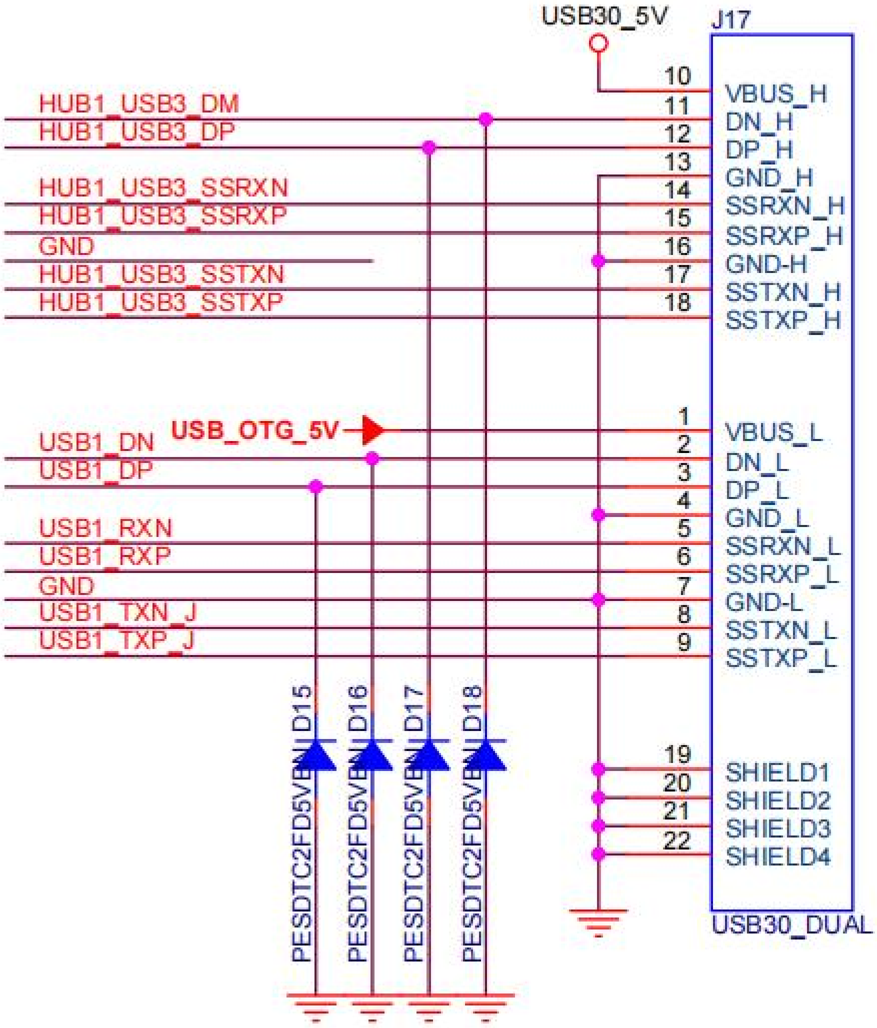


Figure 14 Pin sequence of J17

2.2.3.2. USB2.0

There are three USB 2.0 interfaces (J18, J19, J20) on DEBIX SOM A I/O Board, the connector is 1*4Pin/2.0mm header.

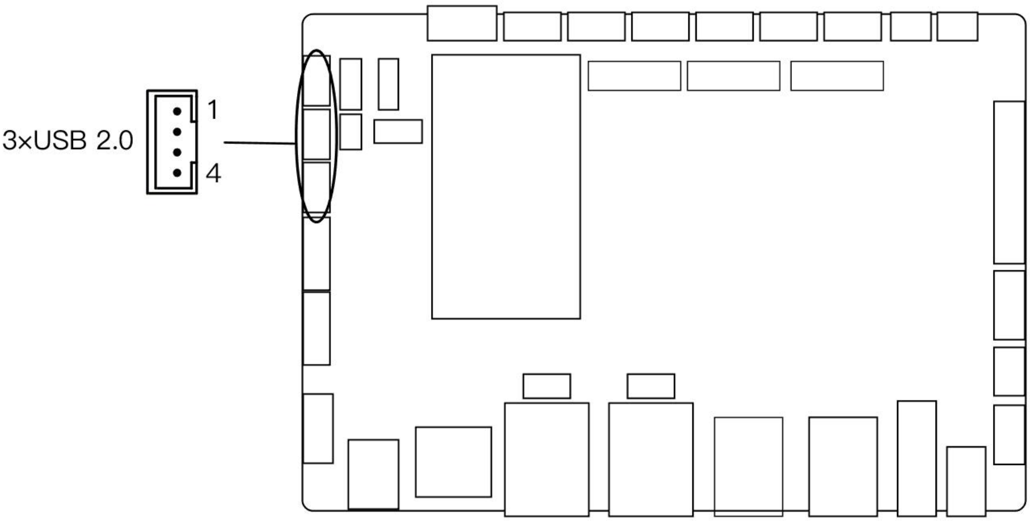


Figure 15 USB 2.0

The pin sequence of USB 2.0 is as shown in the figure:

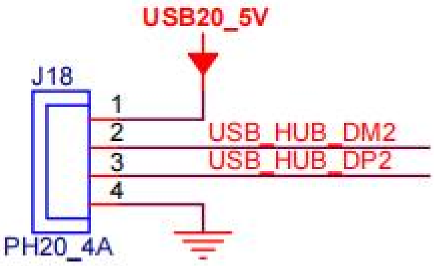


Figure 16 Pin sequence of J18

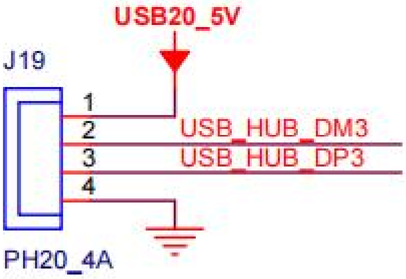


Figure 17 Pin sequence of J19

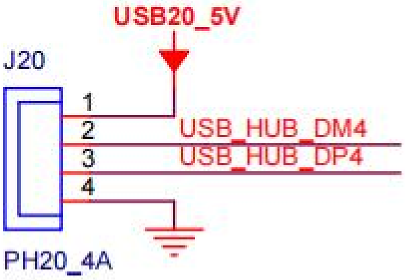


Figure 18 Pin sequence of J20

The J18 interface is defined as follows:

Table 8 Pin definition of J18

Pin	Definition	Description
1	USB 5V	USB power 5V

2	USB_HUB_DM2	USB data(-)
3	USB_HUB_DP2	USB data(+)
4	GND	To ground

The J19 interface is defined as follows:

Table 9 Pin definition of J19

Pin	Definition	Description
1	USB 5V	USB power 5V
2	USB_HUB_DM3	USB data(-)
3	USB_HUB_DP3	USB data(+)
4	GND	To ground

The J20 interface is defined as follows:

Table 10 Pin definition of J20

Pin	Definition	Description
1	USB 5V	USB power 5V
2	USB_HUB_DM4	USB data(-)
3	USB_HUB_DP4	USB data(+)
4	GND	To ground

2.2.4. Display Interface

DEBIX SOM A I/O Board supports the following displays:

- One LCDIF driver for MIPI DSI, up to UWHD and WUXGA
- One LCDIF driver for LVDS Tx, up to 1920x1080p60
- One LCDIF driver HDMI 2.0a Tx, up to 4kp30

When less than or equal to two LCD interfaces are used at the same time, each LCD interface supports 1920x1200p60 display. When three LCD interfaces are used at the same time, it supports two 1080p60 + HDMI 4kp30.

2.2.4.1. HDMI

There is an HDMI interface (J30) on the DEBIX SOM A I/O Board, and the connector is an A-type HDMI female socket, which is used to connect a monitor, TV or projector. HDMI resolution up to 3840x2160p30.

Audio supports 32 channel audio output and supports 1 S/PDIF audio eARC input.

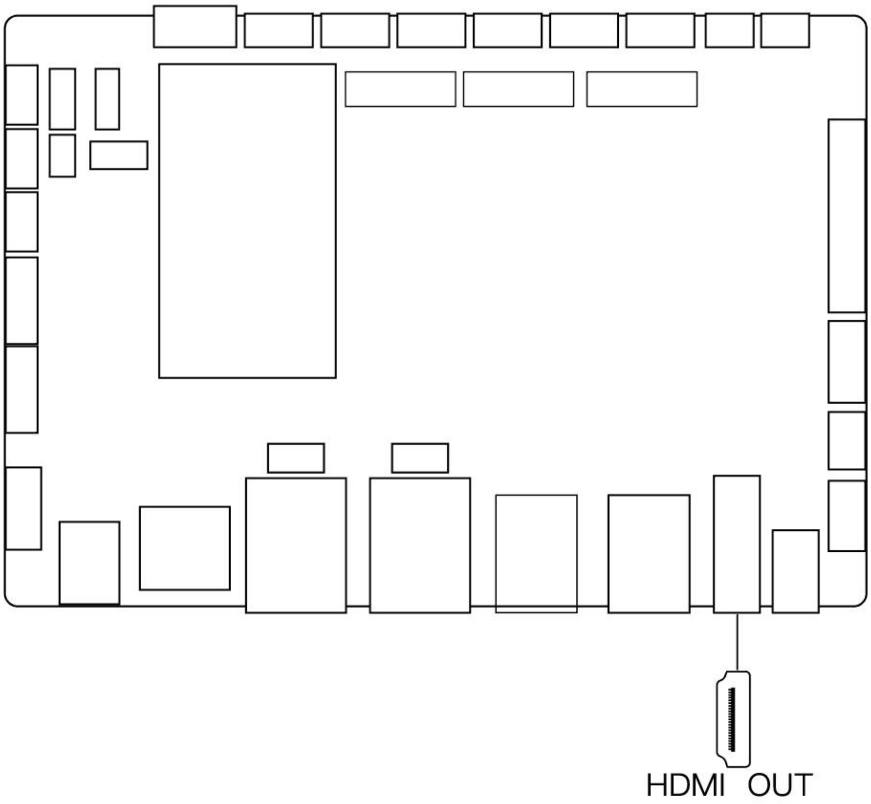


Figure 19 HDMI

The pin sequence is as shown in the figure:

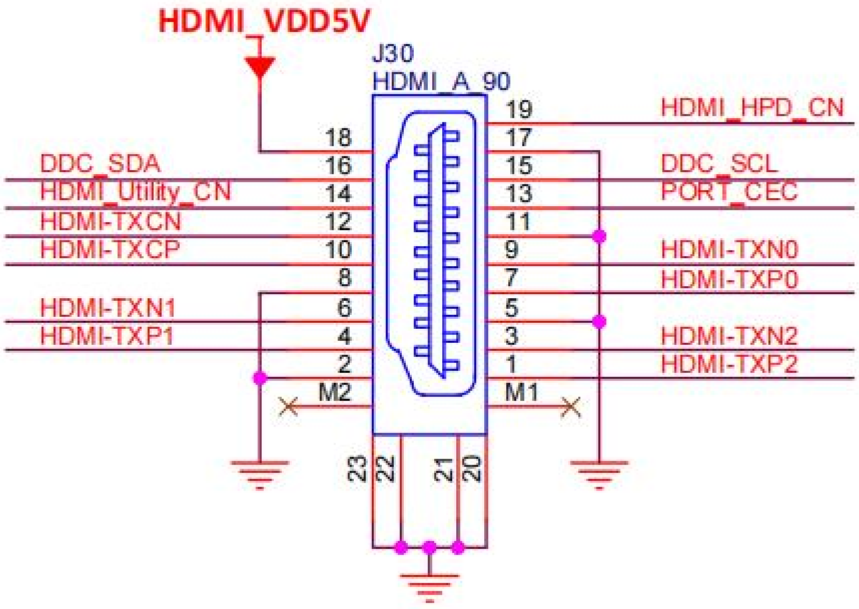


Figure 20 Pin sequence of HDMI

The interface is defined as follows:

Table 11 Pin definition of HDMI

Pin	Definition	Pin	Definition
1	HDMI-TXP2	2	GND
3	HDMI-TXN2	4	HDMI-TXP1
5	GND	6	HDMI-TXN1
7	HDMI-TXP0	8	GND
9	HDMI-TXN0	10	HDMI-TXCP
11	GND	12	HDMI-TXCN
13	PORT_CEC	14	HDMI_Utility_CN
15	DDC_SCL	16	DDC_SDA
17	GND	18	VDD5V
19	HDMI_HPD_CN	20	GND
21	GND	22	GND
23	GND		

2.2.4.2. LVDS

The LVDS display bridge (LDB) connects the LCDIF inside the CPU with the external LVDS display device. The purpose of the LVDS display bridge (LDB) is to transmit synchronous RGB data to an external display device through the LVDS interface.

DEBIX SOM A I/O Board provides a LVDS output interface (J27) with 2*15Pin/2.0mm header, and driven by LDB to support single or dual LVDS display.

- Single channel (4 lanes) 80MHz pixel clock and LVDS clock output. It supports resolutions up to 1366x768p60.
- Asynchronous dual channel (8 data, 2 clocks). This is for a screen with two interfaces, which are transmitted through two channels (odd pixel/even pixel). It supports pixels higher than 1366x768p60 and up to 1080p60.

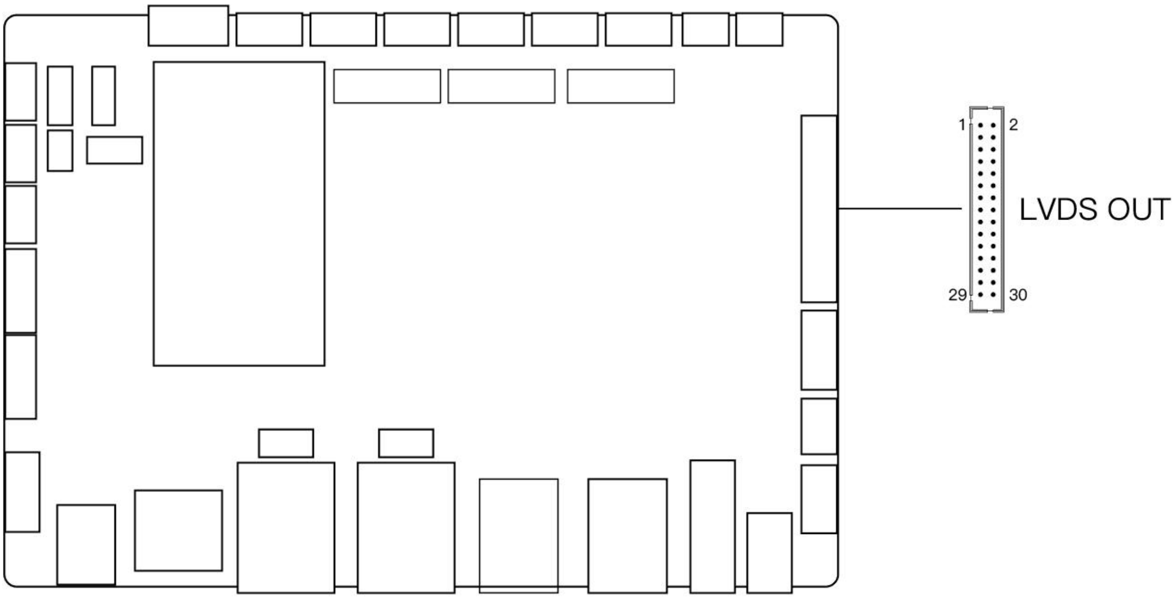


Figure 21 LVDS

The pin sequence is as shown in the figure:

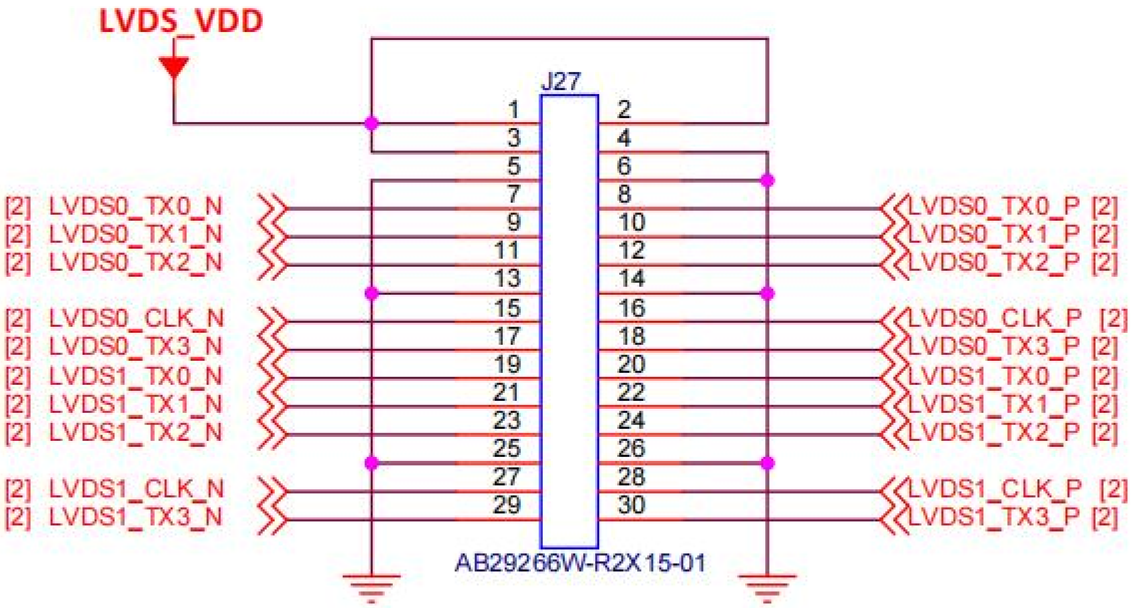


Figure 22 Pin sequence of LVDS

The interface is defined as follows:

Table 12 Pin definition of LVDS

Pin	Definition	Description
1	LVDS_VDD	Default 5V(3.3V, 5V, 12-36V optional)
2	LVDS_VDD	Default 5V(3.3V, 5V, 12-36V optional)
3	LVDS_VDD	Default 5V(3.3V, 5V, 12-36V optional)
4	GND	To Ground
5	GND	To Ground
6	GND	To Ground
7	LVDS0_TX0_N	LVDS0 Differential data channel 0(-)
8	LVDS0_TX0_P	LVDS0 Differential data channel 0(+)
9	LVDS0_TX1_N	LVDS0 Differential data channel 1(-)
10	LVDS0_TX1_P	LVDS0 Differential data channel 1(+)
11	LVDS0_TX2_N	LVDS0 Differential data channel 2(-)
12	LVDS0_TX2_P	LVDS0 Differential data channel 2(+)
13	GND	To Ground

14	GND	To Ground
15	LVDS0_CLK_N	LVDS0 Clock differential signal path(-)
16	LVDS0_CLK_P	LVDS0 Clock differential signal path(+)
17	LVDS0_TX3_N	LVDS0 Differential data channel 3(-)
18	LVDS0_TX3_P	LVDS0 Differential data channel 3(+)
19	LVDS1_TX0_N	LVDS1 Differential data channel 0(-)
20	LVDS1_TX0_P	LVDS1 Differential data channel 0(+)
21	LVDS1_TX1_N	LVDS1 Differential data channel 1(-)
22	LVDS1_TX1_P	LVDS1 Differential data channel 1(+)
23	LVDS1_TX2_N	LVDS1 Differential data channel 2(-)
24	LVDS1_TX2_P	LVDS1 Differential data channel 2(+)
25	GND	To Ground
26	GND	To Ground
27	LVDS1_CLK_N	LVDS1 Clock differential signal path(-)
28	LVDS1_CLK_P	LVDS1 Clock differential signal path(+)
29	LVDS1_TX3_N	LVDS1 Differential data channel 3(-)
30	LVDS1_TX3_P	LVDS1 Differential data channel 3(+)

2.2.4.3. LVDS Backlight Control Adjustment Interface

There is a LVDS backlight control adjustment interface (J29) with 1*6Pin/2.0mm header on the DEBIX SOM A I/O Board. Through this interface, the LVDS backlight can be turned on or off, and the brightness of the backlight can be adjusted.

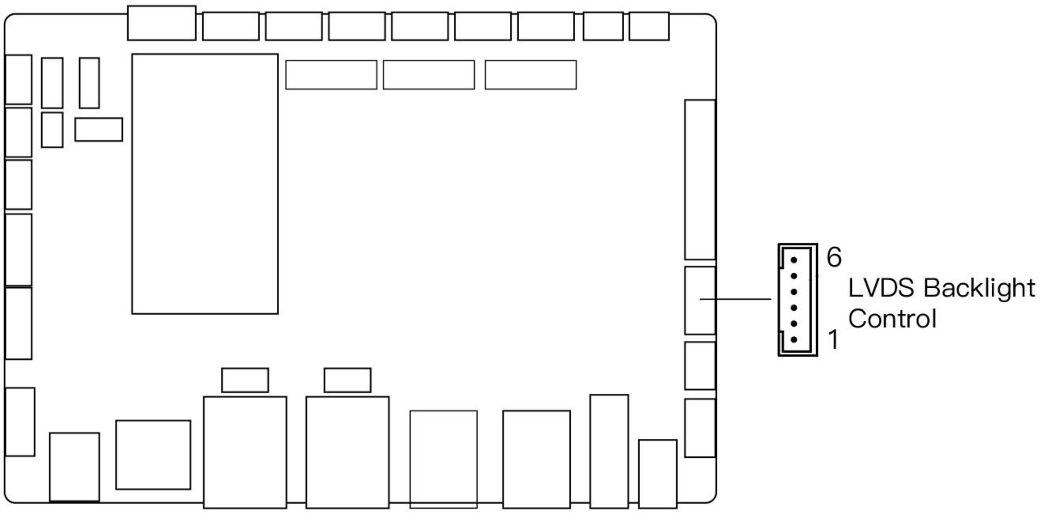


Figure 23 LVDS Backlight Control

The pin sequence is as shown in the figure:

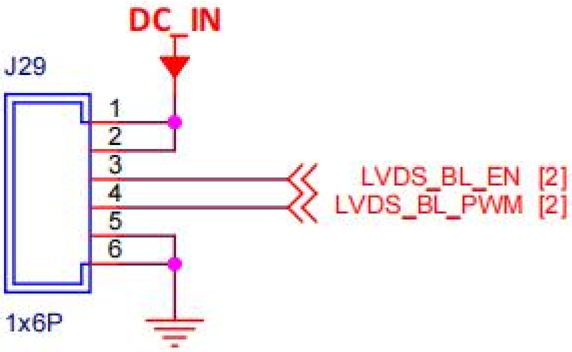


Figure 24 Pin sequence of J29

The interface is defined as follows:

Table 13 Pin definition of J29

Pin	Definition	Description
1	DC_IN	DC 12-36V
2	DC_IN	DC 12-36V
3	LVDS_BL_EN	Backlight switch enable port
4	LVDS_BL_PWM	Backlight brightness PWM control port
5	GND	To Ground
6	GND	To Ground

2.2.4.4. MIPI DSI

There is a MIPI-DSI interface (J28) with 2*10Pin/1.25mm header on the DEBIX SOM A I/O Board, which can be used to connect a MIPI display touch screen.

Key features of MIPI DSI include:

- MIPI DSI complies with MIPI-DSI standard V1.2, compatible with standard specification V1.01r11
- The commonly used MIPI DSI resolutions are supported as follows:
  - 1080 p60, WUXGA (1920x1200) at 60 Hz, 1920x1440 at 60 Hz, UWHD (2560x1080) at 60 Hz
  - Maximum resolution up to WQHD(2560x1440), it depends on bandwidth between input clock (video clock) and output clock (D-PHY HS clock)
  - Support 1, 2, 3 or 4 data lanes
  - Support pixel format: 16bpp, 18bpp packed, 18bpp loosely packed (3 bytes format), 24bpp.
- Interface
  - Compliant with Protocol-to-PHY Interface (PPI) at 1.0Gbps/1.5Gbps MIPI DPHY
  - Support RGB interface for video image input from general display controller.

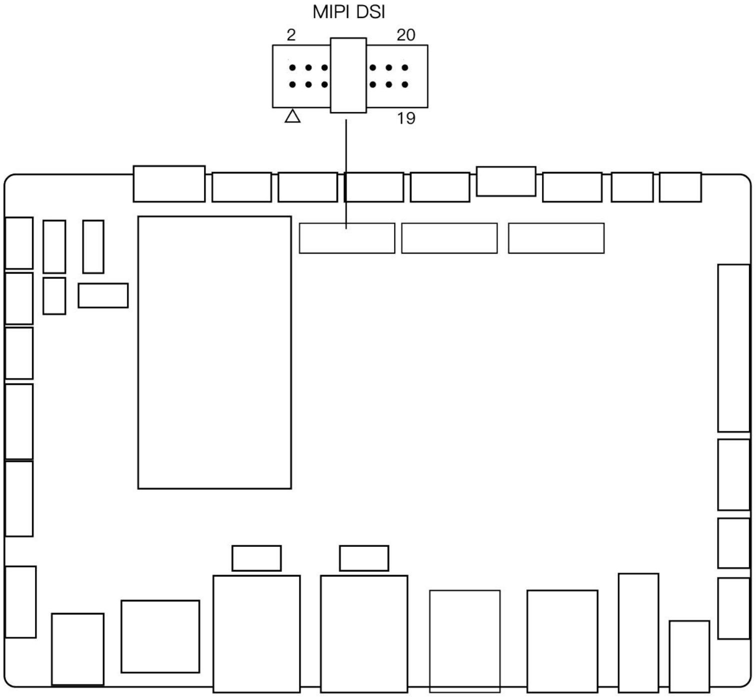


Figure 25 MIPI DSI

The pin sequence is as shown in the figure:

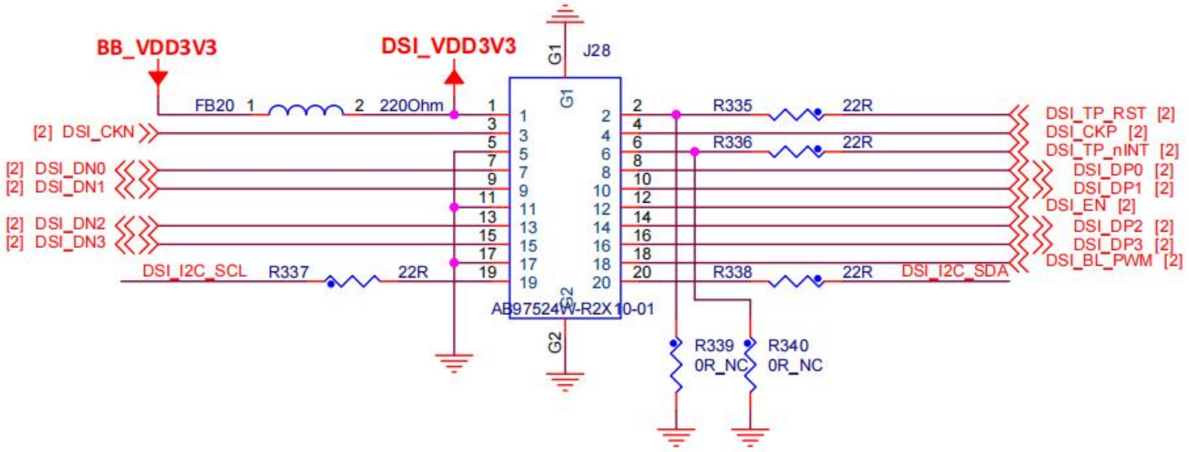


Figure 26 Pin sequence of MIPI DSI

The interface is defined as follows:

Table 14 Pin definition of MIPI DSI

Pin	Definition	Description
1	VDD3V3	3.3V output
2	DSI_TP_RST	DSI reset

3	DSI_CKN	DSI Differential Clock Channels(-)
4	DSI_CKP	DSI Differential Clock Channels(+)
5	GND	To Ground
6	DSI_TP_nINT	touch interrupt pin
7	DSI_DN0	DSI Differential data channel 0(-)
8	DSI_DP0	DSI Differential data channel 0(+)
9	DSI_DN1	DSI Differential data channel 1(-)
10	DSI_DP1	DSI Differential data channel 1(+)
11	GND	To Ground
12	DSI_EN	LCD enable signal
13	DSI_DN2	DSI Differential data channel 2(-)
14	DSI_DP2	DSI Differential data channel 2(+)
15	DSI_DN3	DSI Differential data channel 3(-)
16	DSI_DP3	DSI Differential data channel 3(+)
17	GND	To Ground
18	DSI_BL_PWM	Backlight control signal
19	DSI_I2C_SCL	Touch the clock terminal of I2C (controlled by I2C2)
20	DSI_I2C_SDA	Touch the clock terminal of I2C (controlled by I2C2)

2.2.5. MIPI CSI

The MIPI CSI-2 controller has the following features:

- Supports major and minor image formats
  - YUV420, YUV420(Legacy), YUV420(CSPS), 8-bits and 10-bits YUV422
  - RGB565, RGB666, RGB888
  - RAW6, RAW7, RAW8, RAW10, RAW12, RAW14
- Support D-PHY up to 4 lanes

- Interfaces
  - Image output data bus width: 32 bits
  - Image SRAM storage size is 4KB
  - Pixel clock can be controlled when no PPI data is coming

There are two MIPI-CSI interfaces (J25, J26) with 2*10Pin/1.25mm header on DEBIX SOM A I/O Board, which is used to connect DEBIX's camera module.

- When one MIPI-CSI interface is used, the MIPI-CSI interface supports up to 12MP @30fps or 4kp45.
- When two MIPI-CSI interfaces are used at the same time, each MIPI-CSI interface supports a maximum of 1080p80.

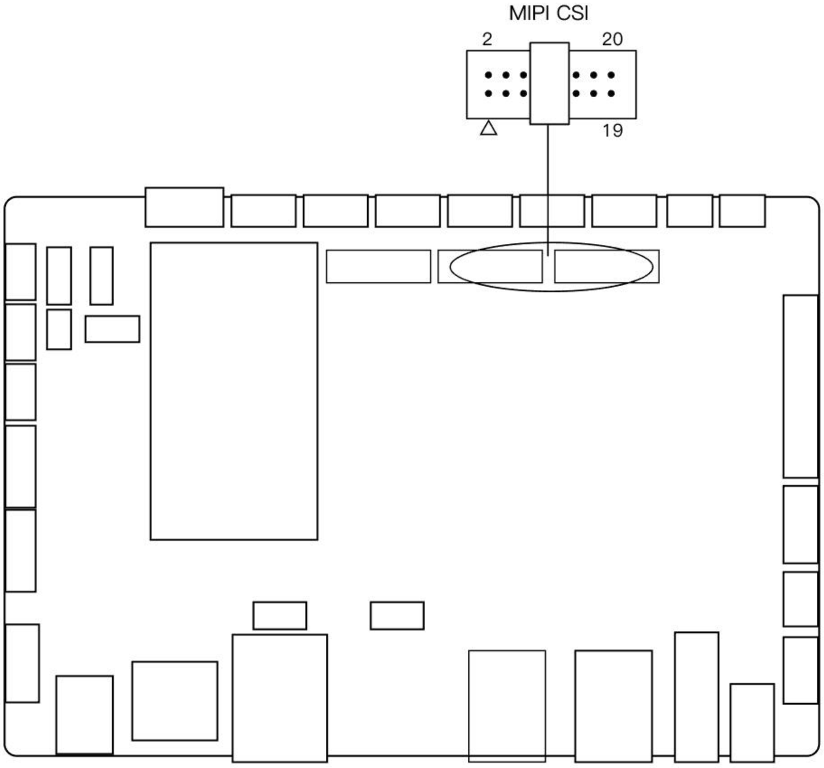


Figure 27 MIPI CSI

The pin sequence is as shown in the figure:

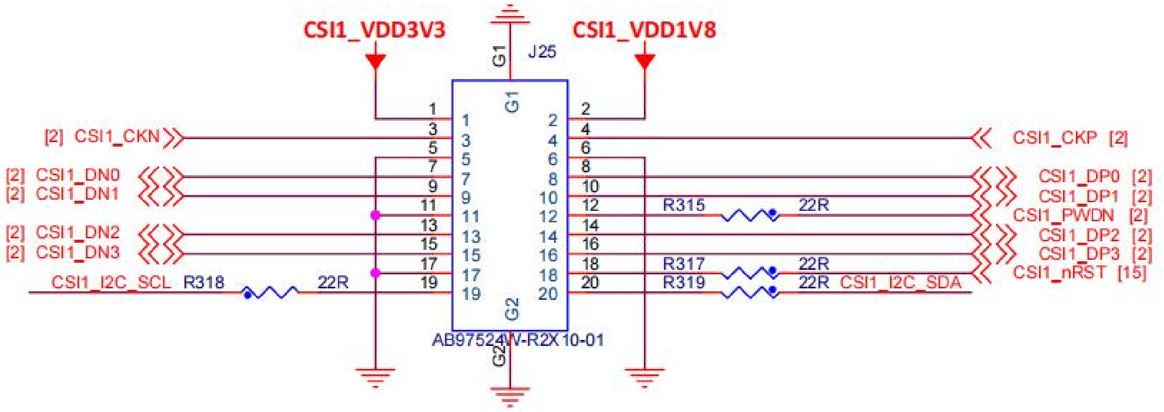


Figure 28 Pin sequence of J25

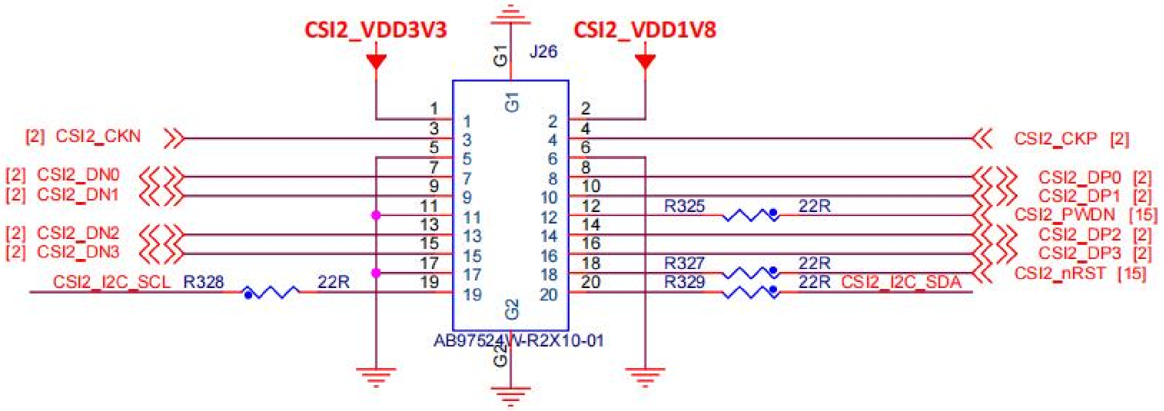


Figure 29 Pin sequence of J26

The interface is defined as follows:

Table 15 Pin definition of J25

Pin	Definition	Description
1	VDD3V3	3.3V output
2	VDD1V8	1.8V output
3	CSI1_CKN	CSI Differential Clock Signal Line(-)
4	CSI1_CKP	CSI Differential Clock Signal Line(+)
5	GND	To ground
6	GND	To ground
7	CSI1_DN0	CSI Differential data signal line0(-)
8	CSI1_DP0	CSI Differential data signal line 0(+)

9	CSI1_DN1	CSI Differential data signal line 1(-)
10	CSI1_DP1	CSI Differential data signal line 1(+)
11	GND	To ground
12	CSI1_PWDN	Camera enable
13	CSI1_DN2	CSI Differential data signal line 2(-)
14	CSI1_DP2	CSI Differential data signal line 2(+)
15	CSI1_DN3	CSI Differential data signal line 3(-)
16	CSI1_DP3	CSI Differential data signal line 3(+)
17	GND	To ground
18	CSI1_nRST	Reset
19	CSI1_I2C_SCL	Camera controls the clock line of I2C (controlled by I2C2)
20	CSI1_I2C_SDA	Camera controls the I2C data line (controlled by I2C2)

Table 16 Pin definition of J26

Pin	Definition	Description
1	VDD3V3	3.3V output
2	VDD1V8	1.8V output
3	CSI2_CKN	CSI differential clock signal line(-)
4	CSI2_CKP	CSI differential clock signal line (+)
5	GND	To ground
6	GND	To ground
7	CSI2_DN0	CSI Differential data signal line 0(-)
8	CSI2_DP0	CSI Differential data signal line 0(+)
9	CSI2_DN1	CSI Differential data signal line 1(-)
10	CSI2_DP1	CSI Differential data signal line 1(+)
11	GND	To ground

12	CSI2_PWDN	Camera enable
13	CSI2_DN2	CSI Differential data signal line 2(-)
14	CSI2_DP2	CSI Differential data signal line 2(+)
15	CSI2_DN3	CSI Differential data signal line 3(-)
16	CSI2_DP3	CSI Differential data signal line 3(+)
17	GND	To ground
18	CSI2_nRST	Reset
19	CSI2_I2C_SCL	Camera controls the clock line of I2C (controlled by I2C3)
20	CSI2_I2C_SDA	Camera controls the I2C data line (controlled by I2C3)

2.2.6. Audio Interface

2.2.6.1. Headphone and Microphone Interface

There is a combined headphone and microphone input interface (J13). The connector is a 3.5mm socket, compatible with the built-in header design, has audio input/output functions, and supports rated voltage 1.5V MIC audio input.

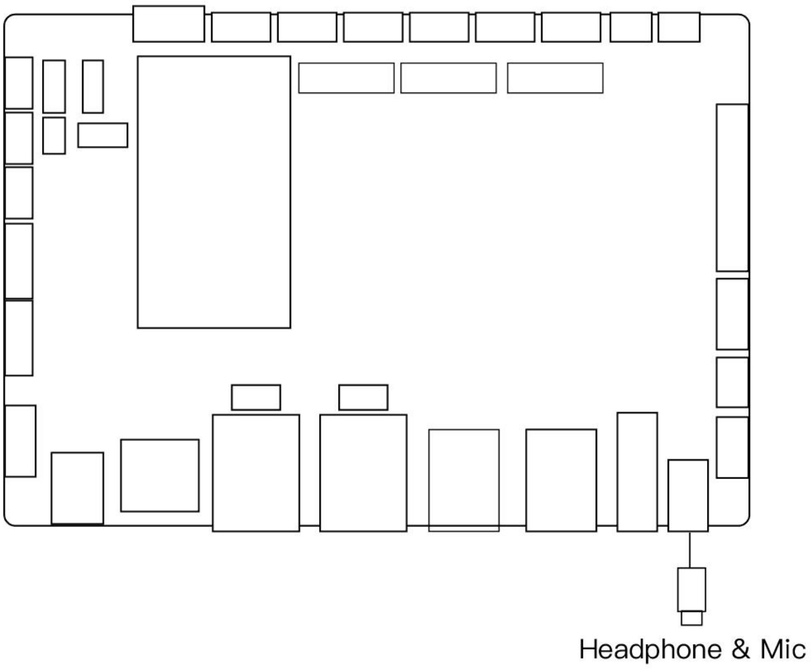


Figure 30 Headphone and Mic

The pin sequence is as shown in the figure:

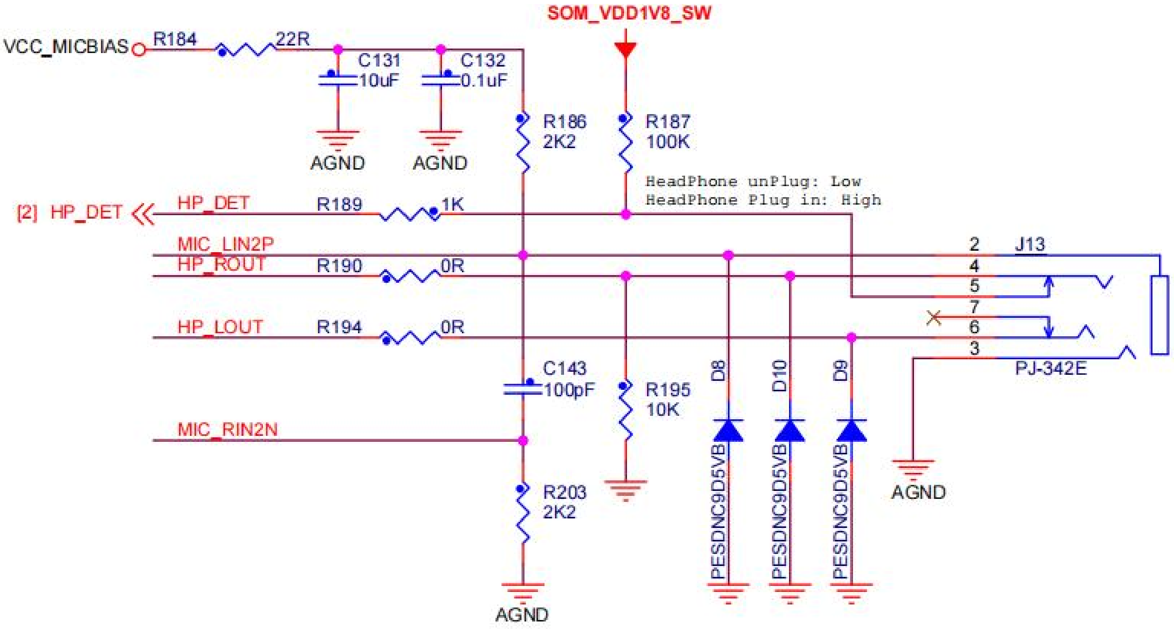


Figure 31 Pin sequence of Headphone and Mic

NOTE

DEBIX uses MIC and only supports four-segment headphones. The definition is shown in the following figure, which includes left channel, right channel, GND, and MIC recording. It is necessary to connect to the DEBIX audio interface according to the GND and MIC connection lines for normal use.

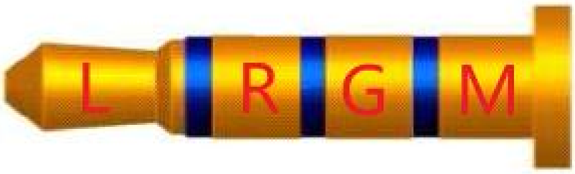


Figure 32 Definition of four-segment headphones

2.2.6.2. Audio Internal Interface

There is a 2*5Pin audio internal interface (J14), in addition to the internal pins of the headphone and microphone input combination interface (J13), and also leads to an audio input and output, with audio input/output functions.

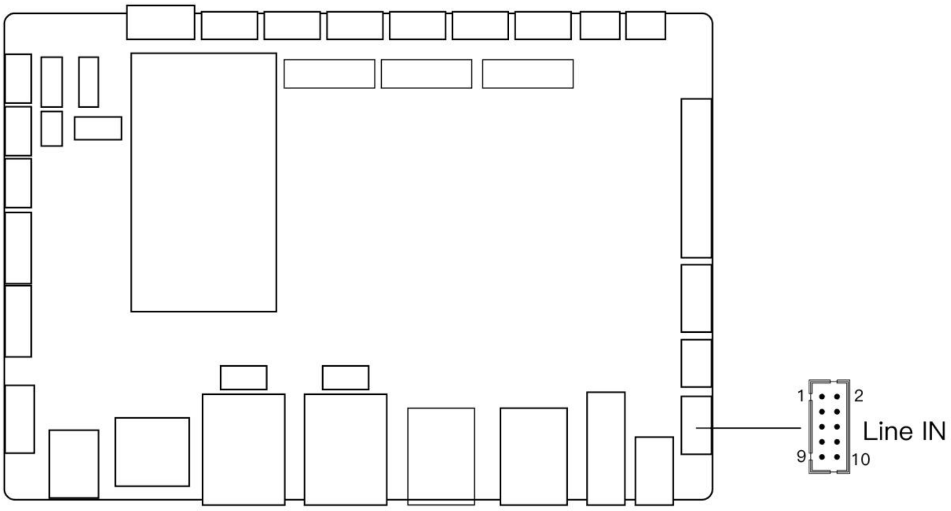


Figure 33 Audio Internal Interface

The pin sequence is as shown in the figure:

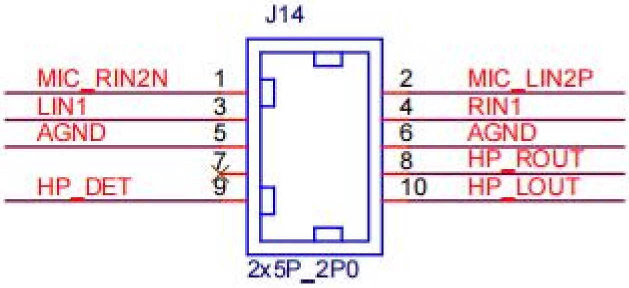


Figure 34 Pin sequence of Audio Internal Interface

The interface is defined as follows:

Table 17 Pin definition of Audio Internal Interface

Pin	Definition	Description
1	MIC_RIN2N	Headphone MIC input (support rated voltage 1.5V MIC)
2	MIC_LIN2P	Headphone MIC input (support rated voltage 1.5V MIC)
3	LIN1	Left channel input
4	RIN1	Right channel input
5	AGND	Audio ground
6	AGND	Audio ground
7	Not used	-
8	HP_ROUT	Headphone right channel output
9	HP_DET	Headphone insertion detection terminal
10	HP_LOUT	Headphone left channel output

2.2.6.3. Speaker Interface

There is a 1*4Pin left and right speaker output interface (J15).

- Support dual audio channel
- Support 4Ω 3w or 8Ω 1.7W

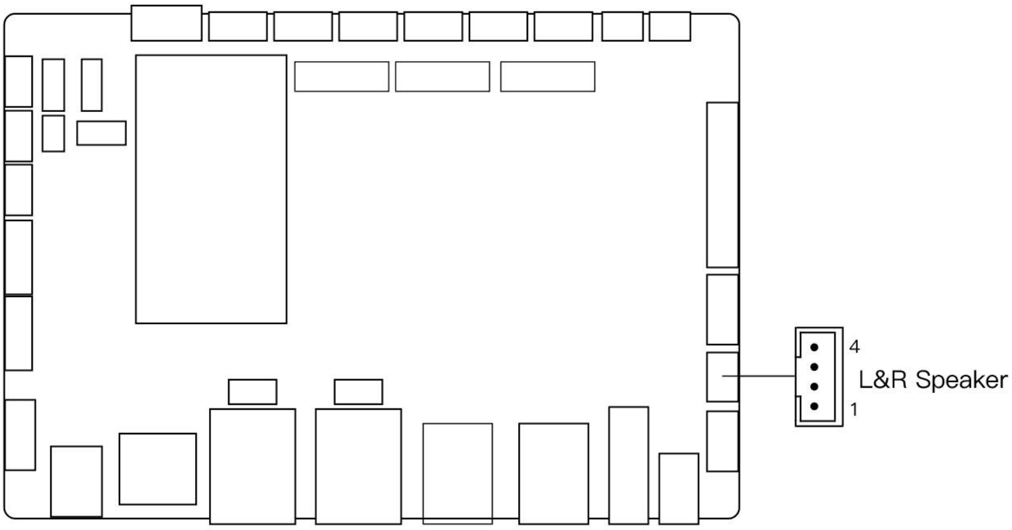


Figure 35 L&R Speaker

The pin sequence is as shown in the figure:

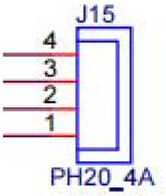


Figure 36 Pin sequence of L&R Speaker

The interface is defined as follows:

Table 18 Pin definition of L&R Speaker

Pin	Definition	Description
1	Right speaker output	Right channel speaker interface
2	Right speaker output	
3	Left speaker output	Left channel speaker interface
4	Left speaker output	

2.2.6.4. SPDIF TX/RX Interface

There is a 1*4Pin SPDIF TX/RX interface (J49), which is used to transfer audio signals between devices such as CD players and amplifiers or computers and sound cards.

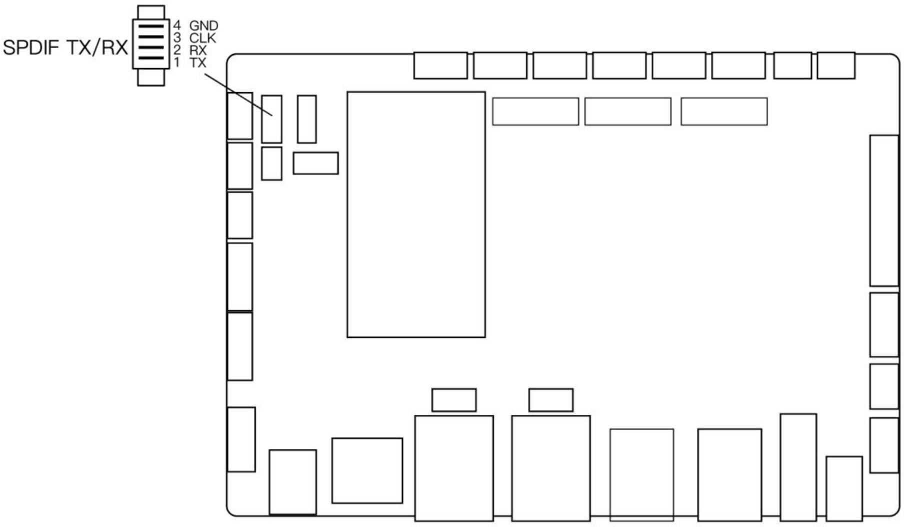


Figure 37 SPDIF TX/RX

The pin sequence is as shown in the figure:

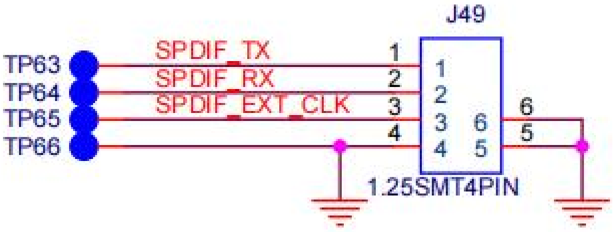


Figure 38 Pin sequence of SPDIF TX/RX

The interface is defined as follows:

Table 19 Pin definition of SPDIF TX/RX

Pin	Definition	Description
1	SPDIF_TX	Digital Audio sender
2	SPDIF_RX	Digital Audio receiver
3	SPDIF_EXT_CLK	Digital Audio Clock Terminal
4	GND	To ground

2.2.7. Mini PCIe

A Mini PCIe interface (J21) onboard supports the following modules, which makes expansion very convenient:

- Mini PCIe LoRa module
- Mini PCIe 4G module, such as Quectel 4G module, built-in SIM card
- Mini PCIe expansion module, such as network card, SATA card, serial port card, etc.

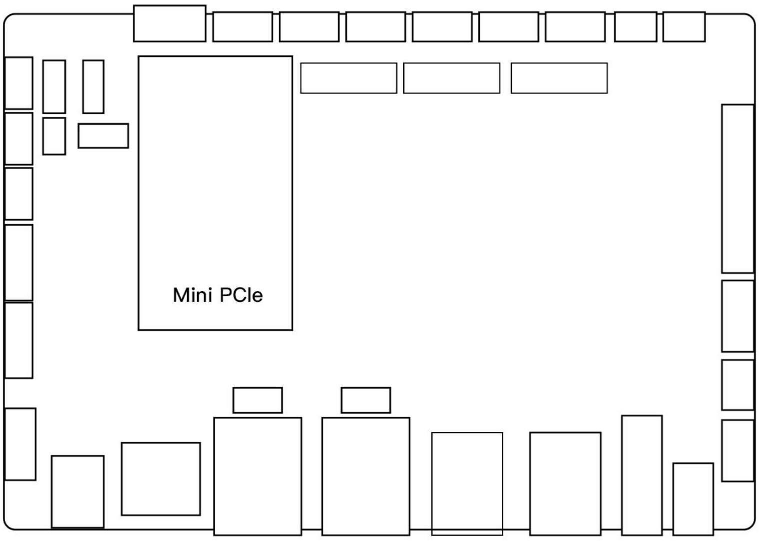


Figure 39 Mini PCIe

The pin sequence is as shown in the figure:

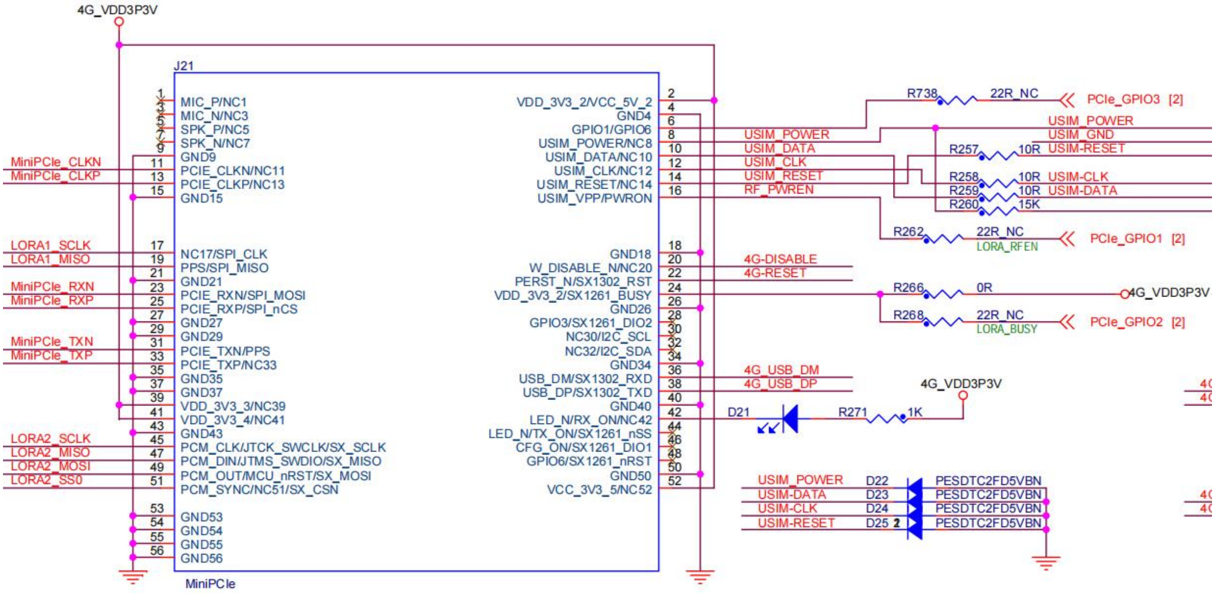


Figure 40 Pin sequence of Mini PCIe

The interface is defined as follows:

Table 20 Pin definition of Mini PCIe

Pin	Definition	Pin	Definition
1	Not used	2	VDD3P3V
3	Not used	4	GND
5	Not used	6	PCIe_GPIO3
7	Not used	8	USIM_POWER
9	GND	10	USIM_DATA
11	MiniPCIe_CLKN	12	USIM_CLK
13	MiniPCIe_CLKP	14	USIM_RESET
15	GND	16	RF_PWREN
17	LORA1_SCLK	18	GND
19	LORA1_MISO	20	4G-DISABLE
21	GND	22	4G-RESET
23	MiniPCIe_RXN	24	VDD3P3V
25	MiniPCIe_RXP	26	GND
27	GND	28	Not used
29	GND	30	Not used
31	MiniPCIe_TXN	32	Not used
33	MiniPCIe_TXP	34	GND
35	GND	36	4G_USB_DM
37	GND	38	4G_USB_DP
39	VDD3P3V	40	GND
41	VDD3P3V	42	LEDN
43	GND	44	Not used
45	LORA2_SCLK	46	Not used

47	LORA2_MISO	48	Not used
49	LORA2_MOSI	50	GND
51	LORA2_SS0	52	VDD3P3V
53	GND	54	GND
55	GND	56	GND

2.2.8. UART/RS232/RS485

2.2.8.1. UART

DEBIX SOM A I/O Board uses **serial port 2** as UART TTL 3.3V system debug serial interface (J47), the connector is a 1*4Pin/1.25mm pitch header.

NOTE

By default, it is used as the DEBUG port for system debugging.

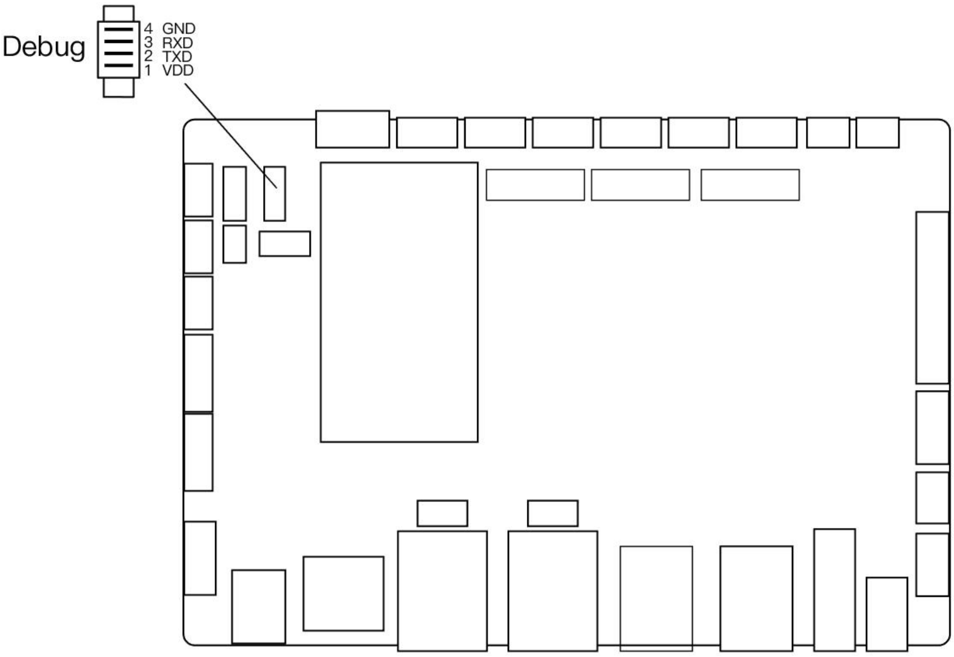


Figure 41 UART

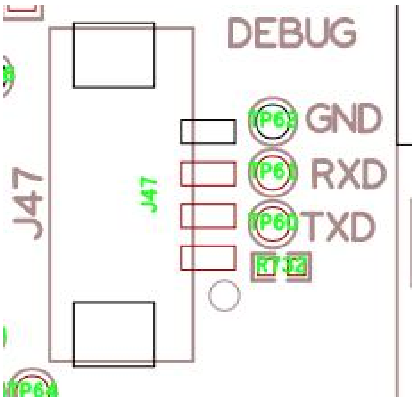


Figure 42 UART silkscreen

The pin sequence is as shown in the figure:

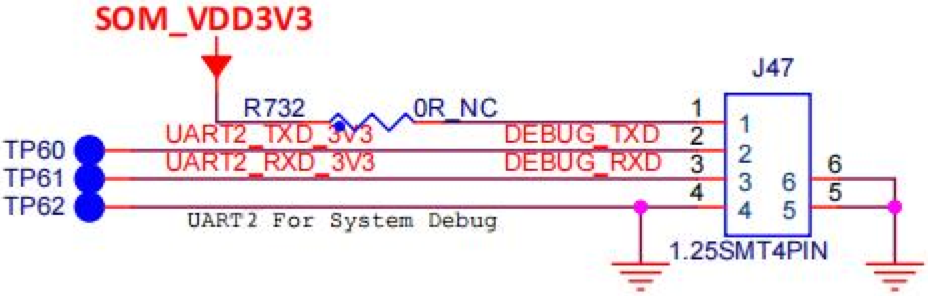


Figure 43 Pin sequence of UART

The interface is defined as follows:

Table 21 Pin definition of UART

Pin	Definition	Description	Device Node
1	VDD_3v3	3.3v	
2	DEBUG_TXD	Default system DEBUG serial port 2 sender	/dev/ttymxc1 (default baud rate 115200)
3	DEBUG_RXD	Default system DEBUG serial port 2 receiver	
4	GND	To ground	

2.2.8.2. RS232/RS485

There are six RS232/RS485 interfaces (J31-J36) with 1*5Pin/2.0mm headers, which are compatible with RS232/RS485 without physical isolation.

IMPORTANT

> RS232 and RS485 on the same socket, only one can be used at the same time.

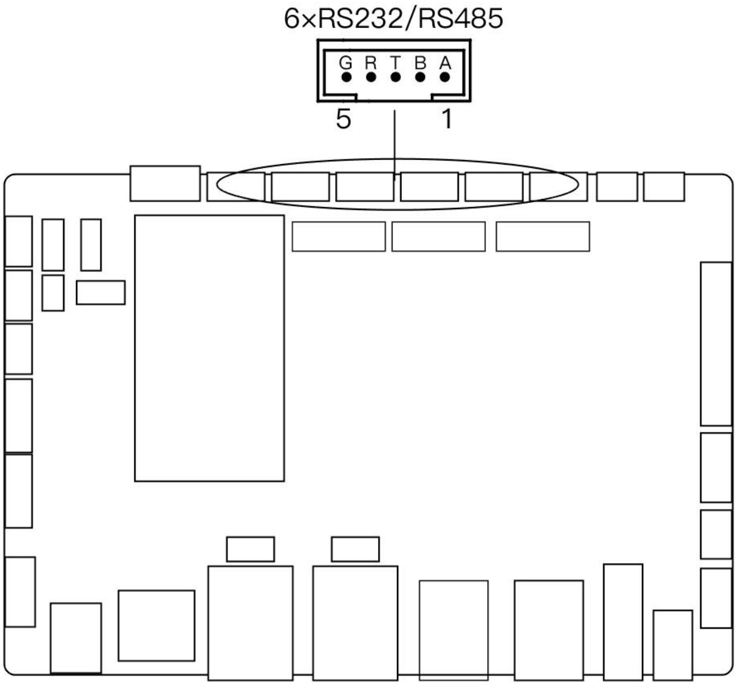


Figure 44 RS232/RS485

The pin sequence of the RS232/RS485 interface (J31-J36) is as shown in the figure:

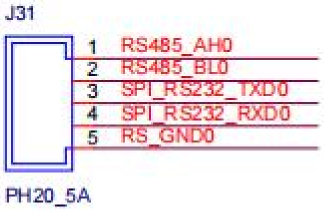


Figure 45 Pin sequence of J31

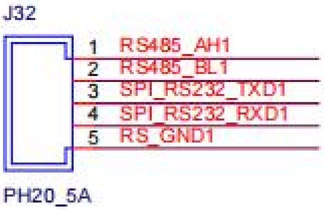


Figure 46 Pin sequence of J32

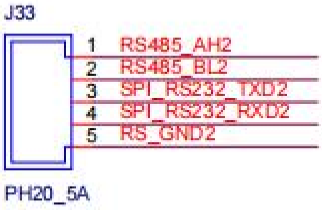


Figure 47 Pin sequence of J33

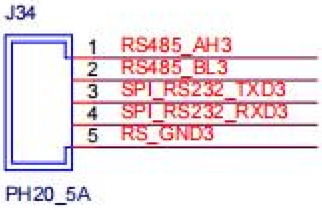


Figure 48 Pin sequence of J34

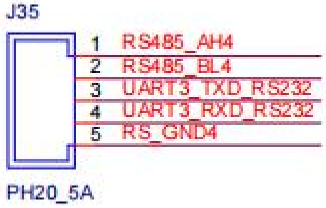


Figure 49 Pin sequence of J35

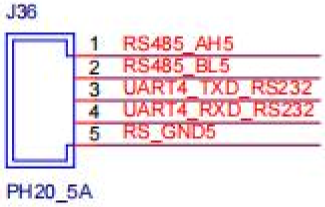


Figure 50 Pin sequence of J36

The RS232/RS485 interface (J31-J36) is defined as follows:

Table 22 Pin definition of J31

Pin	Definition	Description	Device Node
1	RS485_AH0	RS485 differential signal line A	/dev/ttyWCH0
2	RS485_BL0	RS485 differential signal line B	
3	SPI_RS232_TXD0	RS232 sender	
4	SPI_RS232_RXD0	RS232 receiver	
5	RS_GND0	To ground	

Table 23 Pin definition of J32

Pin	Definition	Description	Device Node
1	RS485_AH1	RS485 differential signal line A	/dev/ttyWCH1
2	RS485_BL1	RS485 differential signal line B	
3	SPI_RS232_TXD1	RS232 sender	
4	SPI_RS232_RXD1	RS232 receiver	
5	RS_GND1	To ground	

Table 24 Pin definition of J33

Pin	Definition	Description	Device Node
1	RS485_AH2	RS485 differential signal line A	/dev/ttyWCH2
2	RS485_BL2	RS485 differential signal line B	
3	SPI_RS232_TXD2	RS232 sender	

4	SPI_RS232_RXD2	RS232 receiver	
5	RS_GND2	To ground	

Table 25 Pin definition of J34

Pin	Definition	Description	Device Node
1	RS485_AH3	RS485 differential signal line A	/dev/ttyWCH3
2	RS485_BL3	RS485 differential signal line B	
3	SPI_RS232_TXD3	RS232 sender	
4	SPI_RS232_RXD3	RS232 receiver	
5	RS_GND3	To ground	

Table 26 Pin definition of J35

Pin	Definition	Description	Device Node
1	RS485_AH4	RS485 differential signal line A	/dev/ttymxc2
2	RS485_BL4	RS485 differential signal line B	
3	SPI_RS232_TXD4	RS232 sender	
4	SPI_RS232_RXD4	RS232 receiver	
5	RS_GND4	To ground	

Table 27 Pin definition of J36

Pin	Definition	Description	Device Node
1	RS485_AH5	RS485 differential signal line A	/dev/ttymxc3
2	RS485_BL5	RS485 differential signal line B	
3	SPI_RS232_TXD5	RS232 sender	
4	SPI_RS232_RXD5	RS232 receiver	
5	RS_GND5	To ground	

2.2.9. CAN

There are two physically isolated CAN bus interfaces (J23, J24) with 1*3Pin/2.0mm headers on the DEBIX SOM A I/O Board.

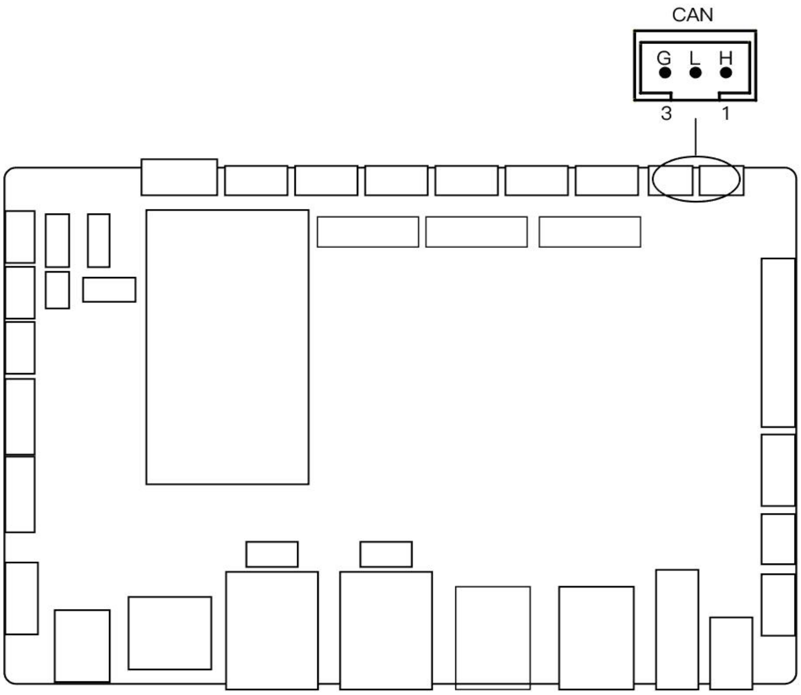


Figure 51 CAN

The pin sequence is as shown in the figure:

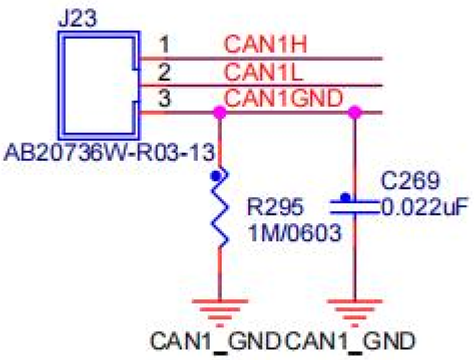


Figure 52 Pin sequence of J23

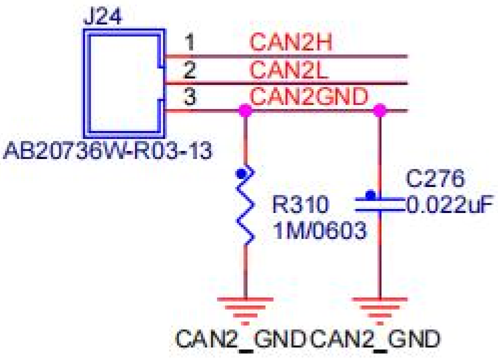


Figure 53 Pin sequence of J24

The CAN interface (J23-J24) is defined as follows:

Table 28 Pin definition of J23

Pin	Definition	Description	Device Node

1	CAN1H	CAN1 differential signal line H	can0
2	CAN1L	CAN1 differential signal line L	
3	CAN1GND	To ground	

Table 29 Pin definition of J24

Pin	Definition	Description	Device Node
1	CAN2H	CAN2 differential signal line H	can1
2	CAN2L	CAN2 differential signal line L	
3	CAN2GND	To ground	

2.2.10. DO

There is a DO interface (J40) with a 1*6Pin/2.0mm header, a physically isolated DO that supports wet nodes, and is compatible with external relay dry nodes.

DO electrical parameters:

- Node DO-PCOM supports a voltage range of 5~30V DC;
- When the SOM A signal is low, the corresponding DO signal output voltage follows the node DO-PCOM, the higher the node DO-PCOM voltage, the higher the DO signal output voltage (compared with the node DO-PCOM, there is a 1~3V voltage drop);
- When the SOM A signal is high level, the corresponding DO signal output voltage is 0.

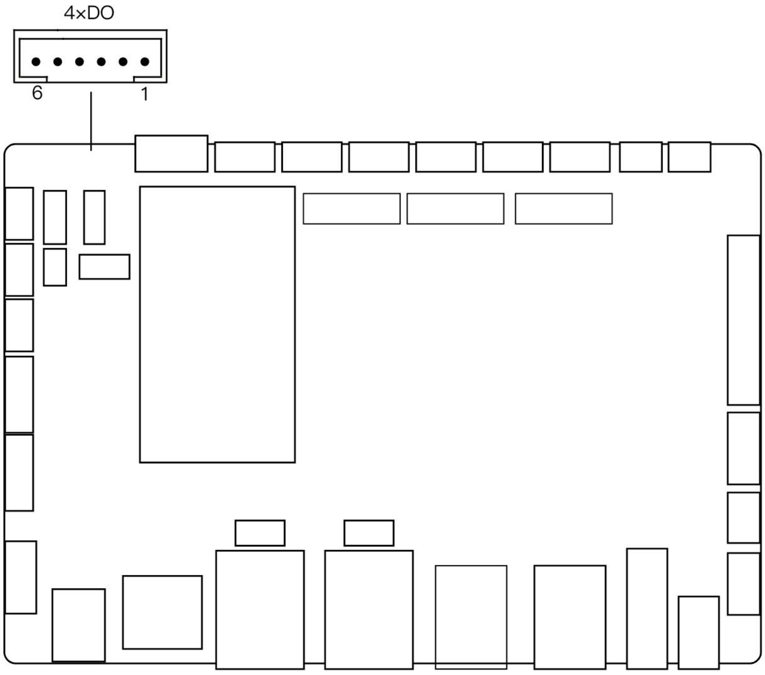


Figure 54 DO

The pin sequence is as shown in the figure:

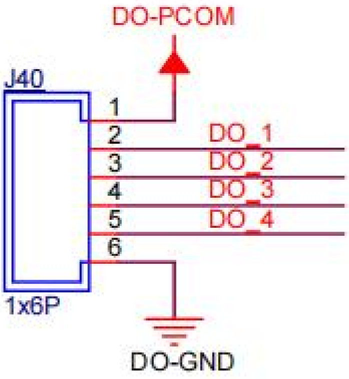


Figure 55 Pin sequence of DO

The interface is defined as follows:

Table 30 Pin definition of DO

Pin	Definition	Description
1	DO_PCOM	Isolated Digital Input Power Common
2	DO_1	Digital isolated output 1
3	DO_2	Digital isolated output 2
4	DO_3	Digital isolated output 3

5	DO_4	Digital isolated output 4
6	GND	To ground

Circuit schematic of DO interface:

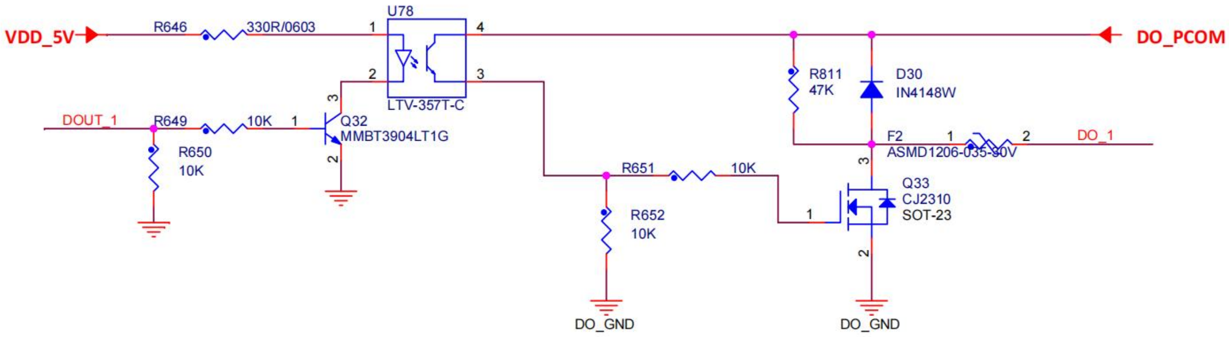


Figure 56 Circuit schematic of DO interface

2.2.11. DI

There is a DI interface (J41) with a 1*6Pin/2.0mm header, a digital isolated DI that supports dry node input and wet node input.

DI electrical parameters:

- Wet contact:
  - The signal input voltage range is 0~30V DC;
  - When the signal input voltage is 0~3V DC, the corresponding SOM A signal is low level;
  - When the signal input voltage is 5~30V DC, the corresponding SOM A signal is high level;
- Dry contact:
  - Open state: high level
  - Short state with GND: low level

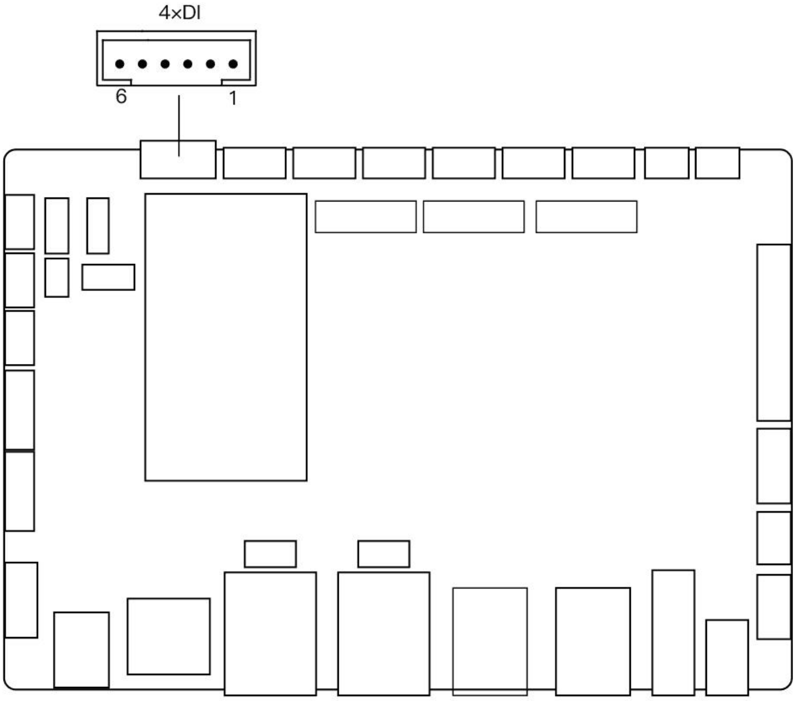


Figure 57 DI

The pin sequence is as shown in the figure:

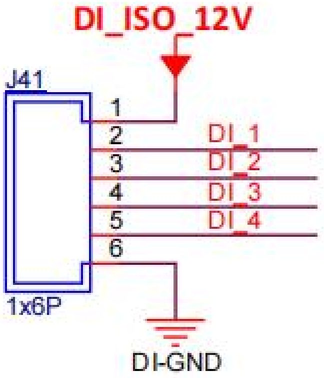


Figure 58 Pin sequence of DI

The interface is defined as follows:

Table 31 Pin definition of DI

Pin	Definition	Description
1	DI_ISO_12V	Digital isolated power supply 12V output
2	DI_1	Digital isolated input 1

3	DI_2	Digital isolated input 2
4	DI_3	Digital isolated input 3
5	DI_4	Digital isolated input 4
6	GND	To ground

Circuit schematic of DI interface:

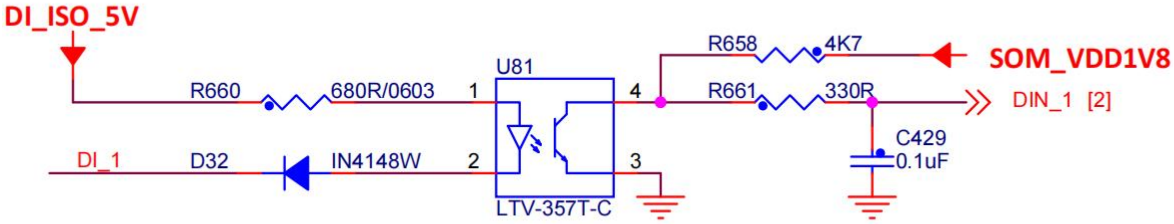


Figure 59 Circuit schematic of DI interface

2.2.12. RTC

There is a RTC interface (J38) with a 1*2Pin/1.25mm pitch header on board.

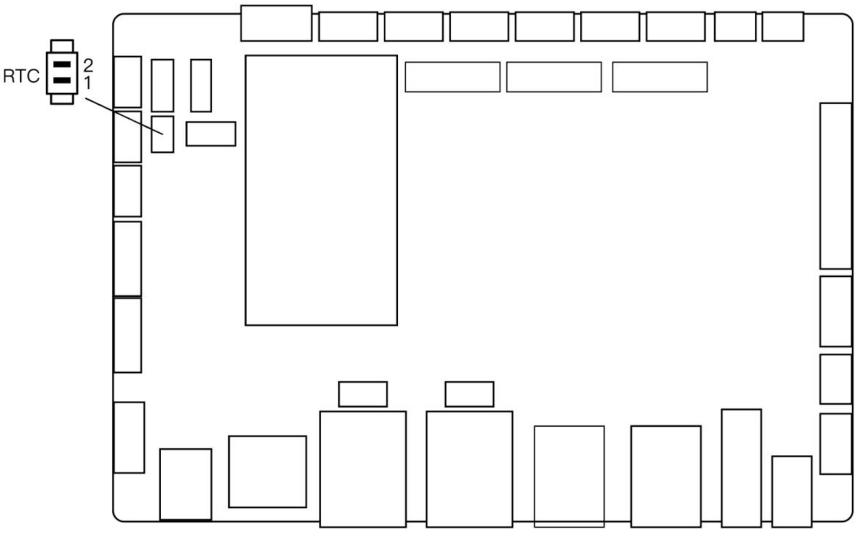


Figure 60 RTC

The pin sequence is as shown in the figure:

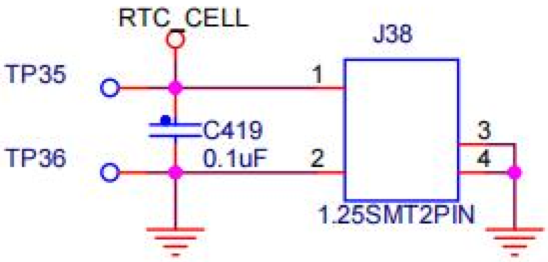


Figure 61 Pin sequence of RTC

The interface is defined as follows:

Table 32 Pin definition of RTC

Pin	Definition	Description
1	RTC_CELL	RTC battery positive, 3.0V
2	GND	To ground

2.2.13. I2C Touch Screen Interface

There is a I2C touch screen interface (J39) with a 1*6Pin/2.0mm pitch header, which is powered by 3.3V by default (1.8V optional).

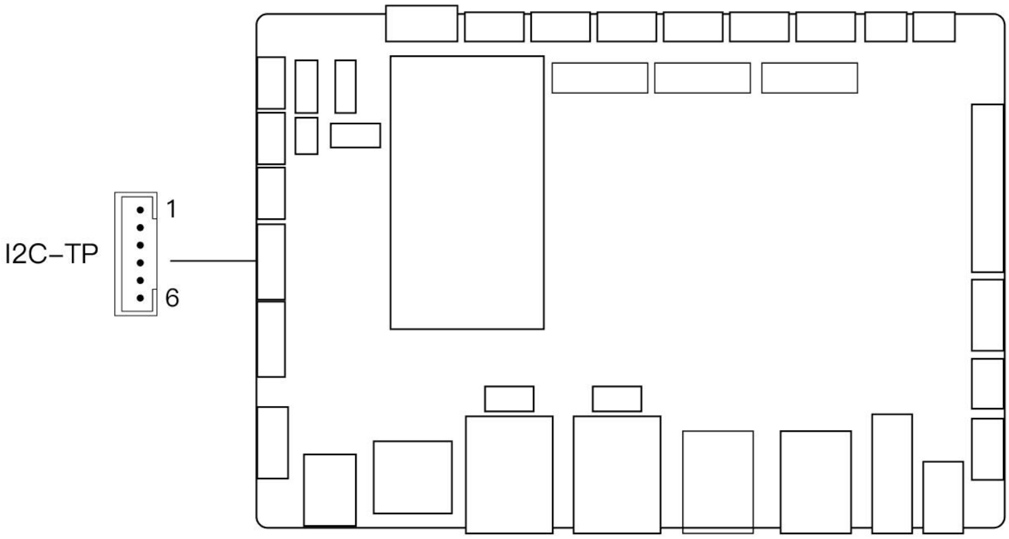


Figure 62 I2C Touch Screen Interface

The pin sequence is as shown in the figure:

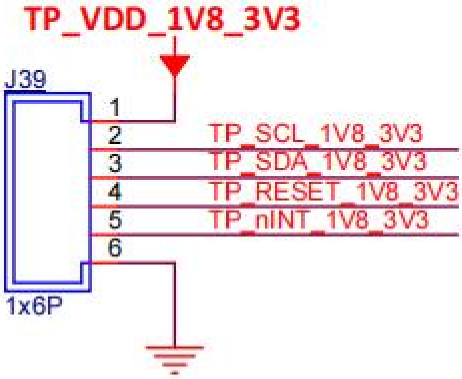


Figure 63 Pin sequence of I2C Touch Screen interface

The interface is defined as follows:

Table 33 Pin definition of I2C Touch Screen interface

Pin	Definition	Description
1	VDD	Touch screen power supply, the default voltage is 3.3V (1.8V optional)
2	TP_SCL	The I2C clock port of the touch screen (controlled by I2C4), the default voltage is 3.3V (1.8V optional)
3	TP_SDA	The I2C data port of the touch screen (controlled by I2C4), the default voltage is 3.3V (1.8V optional)
4	TP_RESET	Touch screen reset, the default voltage is 3.3V (1.8V optional)
5	TP_nINT	Touch screen interrupt port, the default voltage is 3.3V (1.8V optional)
6	GND	To ground

2.2.14. LED & Key

There is a extended LED and Key interface (J50) with a 1*6Pin/2.0mm pitch header onboard, Pin2 is the power indicator pin, Pin3 and Pin4 are the programmable control indicator pin.

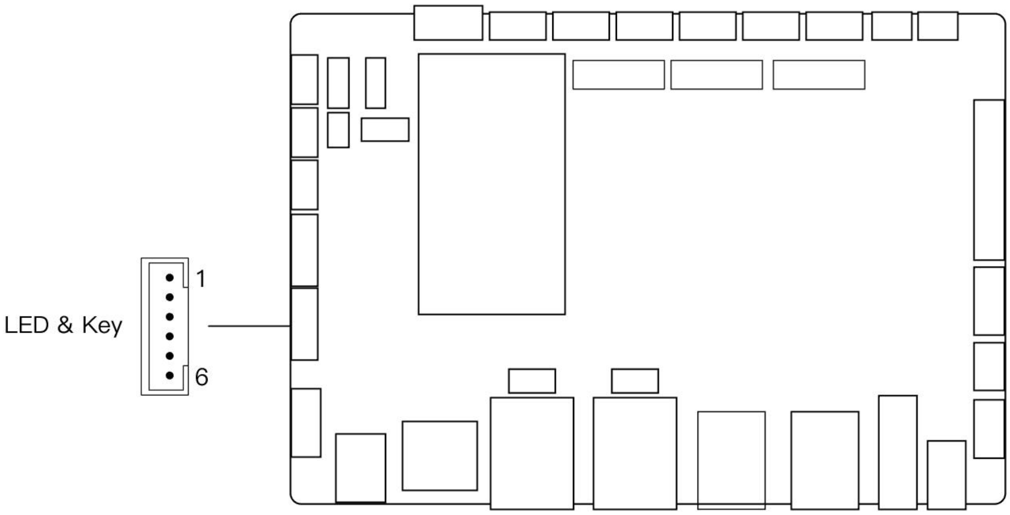


Figure 64 LED & Key

The pin sequence is as shown in the figure:

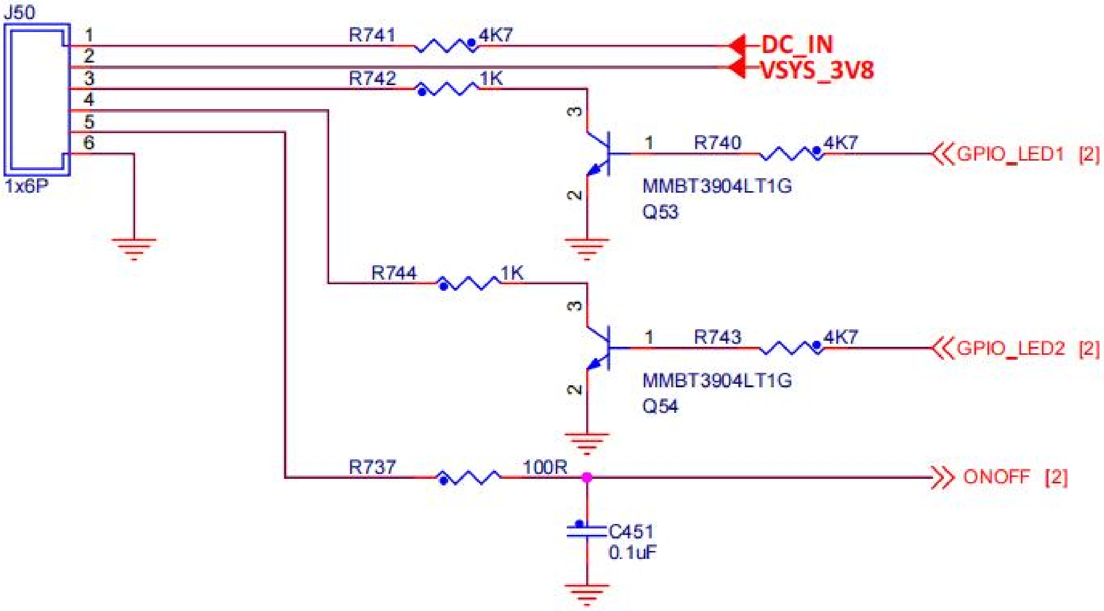


Figure 65 Pin sequence of LED &Key

The interface is defined as follows:

Table 34 Pin definition of LED & Key

Pin	Definition	Description
1	DC_IN	12-36V
2	VSYS_3V8	3.8V
3	GPIO_LED1	Pull-down output high level (controlled by GPIO4_IO01)

4	GPIO_LED2	Pull-down output high level (controlled by GPIO4_IO20)
5	ON/OFF	System On/Off key, active low
6	GND	To ground

2.2.15. DIP Switch

There is a dip-switch combination (SW1, SW2), which is used to determine the BOOT startup mode. There are four switches in total, and each switch has the two states of ON/OFF. By default, the switch is turned ON. BOOT startup modes as follows:

- 0001-USB burning mode
- 0010-DEBIX SOM A On-board eMMC Boot
- 0011-DEBIX SOM A I/O Board Micro SD Card Boot
- 0110-DEBIX SOM A I/O Board SPI Nor Flash boot (reserved)

Warning

In order to protect the DIP switch, the following guidelines need to be observed:

1. The yellow tape on the DIP switch needs to be torn off;
2. It is not allowed to use sharp objects to toggle the DIP switch.

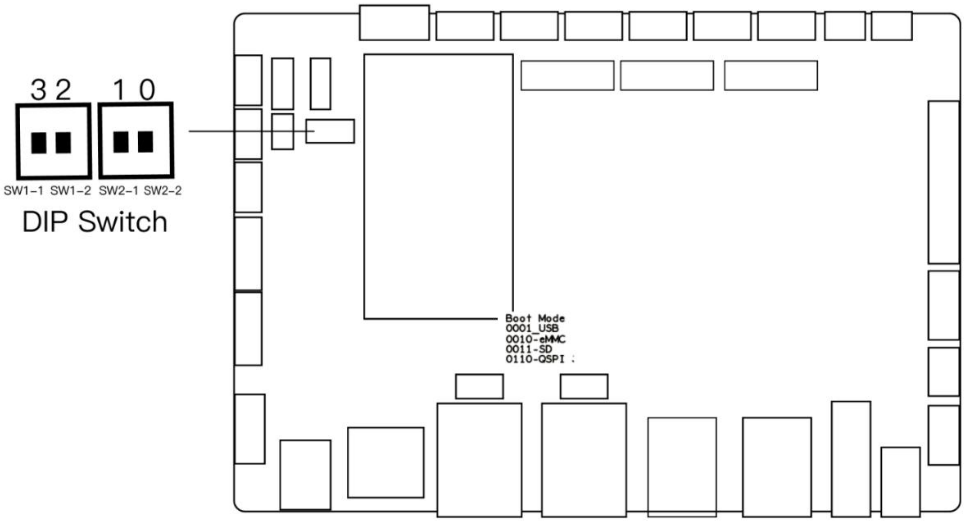
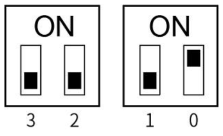
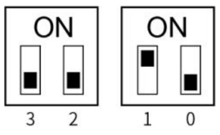
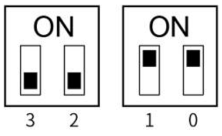
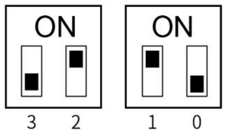


Figure 66 DIP Switch

SW1 is on the left, and SW2 is on the right. Switch the switches of SW1 and SW2 to the corresponding state. The selected boot mode is shown in the table below:

Table 35 DIP switch set boot mode

Mode **Switch**	**USB**	**eMMC**	**Micro SD**	**QSPI**
SW state setting				
Note: The switch is facing up, it is ON state, the switch is facing down, it is OFF state.				

2.2.16. Slot

There are two slots on DEBIX SOM A I/O Board:

- One Micro SD card slot (J11) on the front of the board
- One Micro SIM card slot (J22) on the back side of the board

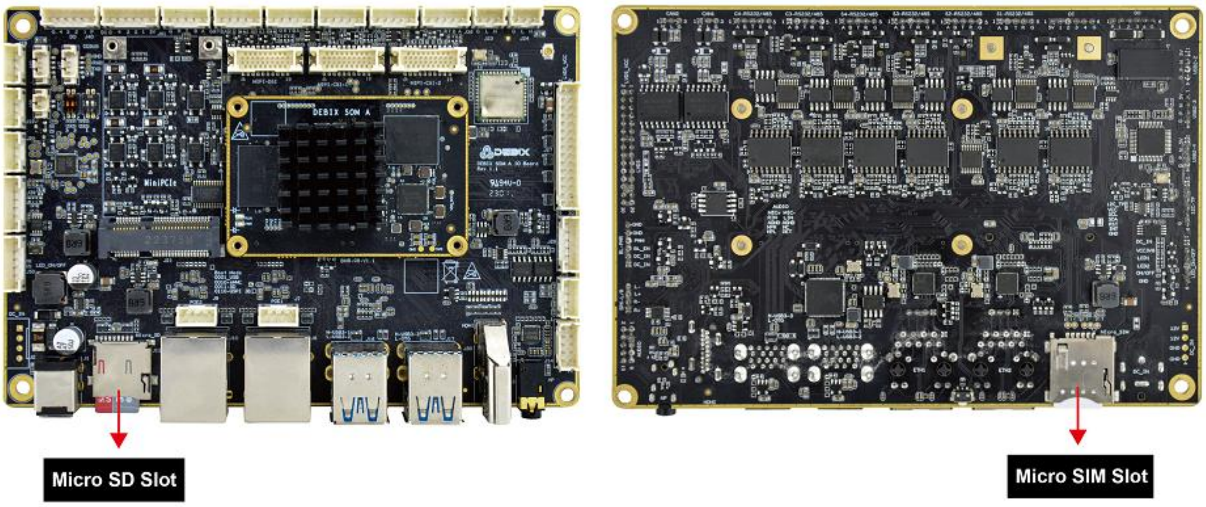


Figure 67 Direction of card slot insertion

2.2.16.1. Micro SD Slot

DEBIX SOM A I/O board provides a Micro SD slot (J11), set the DIP switch to "0011" (Micro SD card boot mode), Micro SD card can be used as a system boot card, insert the Micro SD card with the system installed here, and then power on DEBIX to start the system in the Micro SD card. Please refer to Setting BOOT Mode by DIP Switch for Micro SD card boot mode.

When the DIP switch is set to other modes and the device is power on, the Micro SD card can be used as a standard memory card to save user data.

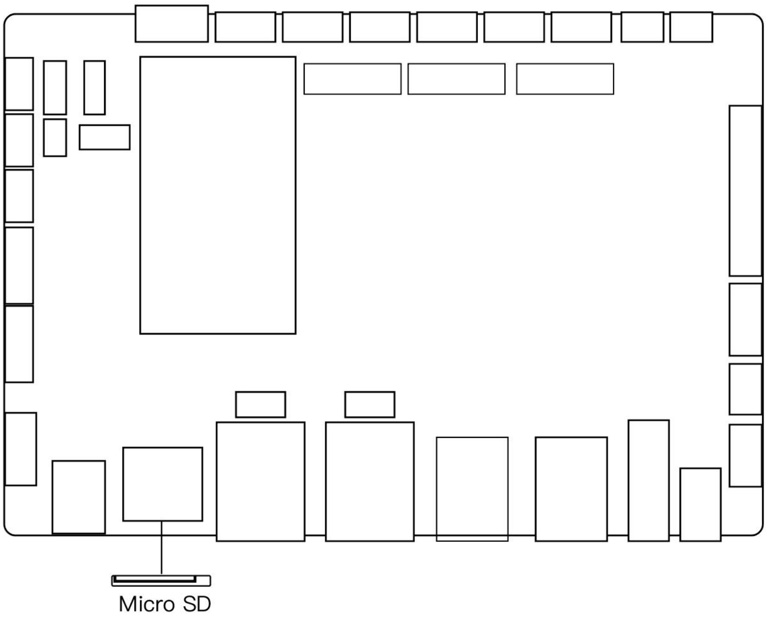


Figure 68 Micro SD Slot

The pin sequence is as shown in the figure:

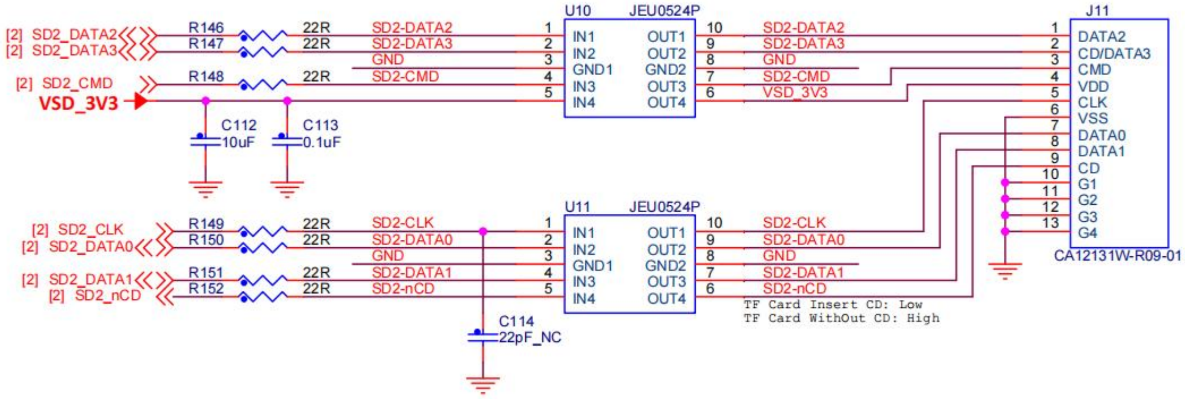


Figure 69 Pin sequence of Micro SD Slot

The interface is defined as follows:

Table 36 Pin definition of Micro SD Slot

Pin	Definition	Description
1	DATA2	Micro SD data 2
2	CD/DATA3	Micro SD data 3
3	CMD	Micro SD command line
4	VDD	3.3V power
5	CLK	Micro SD clock signal
6	VSS	To ground
7	DATA0	Micro SD data 0
8	DATA1	Micro SD data 1
9	CD	Micro SD card insertion detection, active low
10	G1	To ground
11	G2	To ground
12	G3	To ground
13	G4	To ground

2.2.16.2. Micro SIM Slot

DEBIX SOM A I/O board provides a Micro SIM slot (J22), which is used to insert a SIM card to provide network connection and data transmission for the 4G module when Mini PCIe is connected to the 4G module.

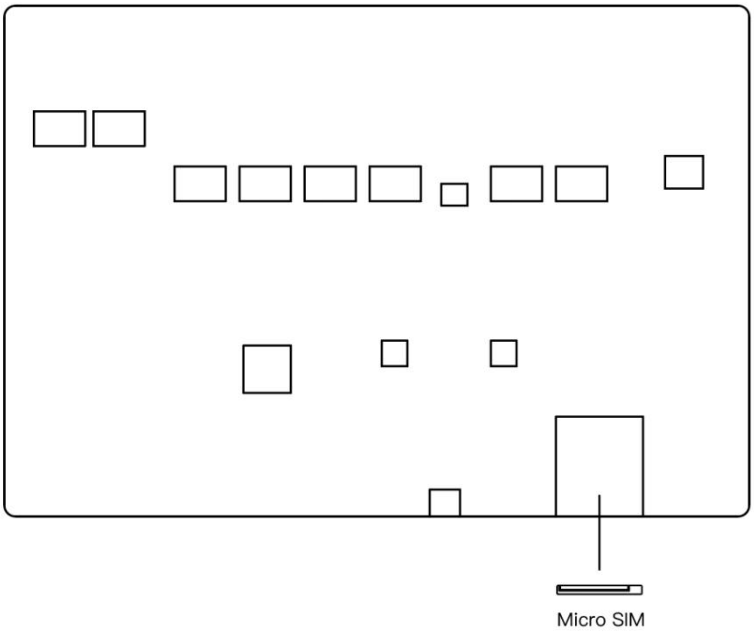


Figure 70 Micro SIM Slot

The pin sequence is as shown in the figure:

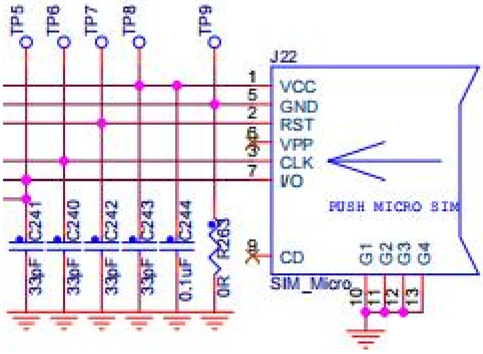


Figure 71 Pin sequence of Micro SIM Slot

The interface is defined as follows:

Table 37 Pin definition of Micro SIM Slot

Pin	Definition	Description
1	VCC	Micro SIM power
2	RST	Micro SIM reset
3	CLK	Micro SIM clock
4	Not used	-
5	GND	To ground

6	Not used	-
7	I/O	Micro SIM data
8	Not used	-
9	Not used	-
10	GND	To ground
11	GND	To ground
12	GND	To ground
13	GND	To ground

2.2.17. Built-in/upgrade Button

There is a built-in/upgrade button (K1) on the back of DEBIX SOM A I/O Board, which is used for eMMC upgrade.

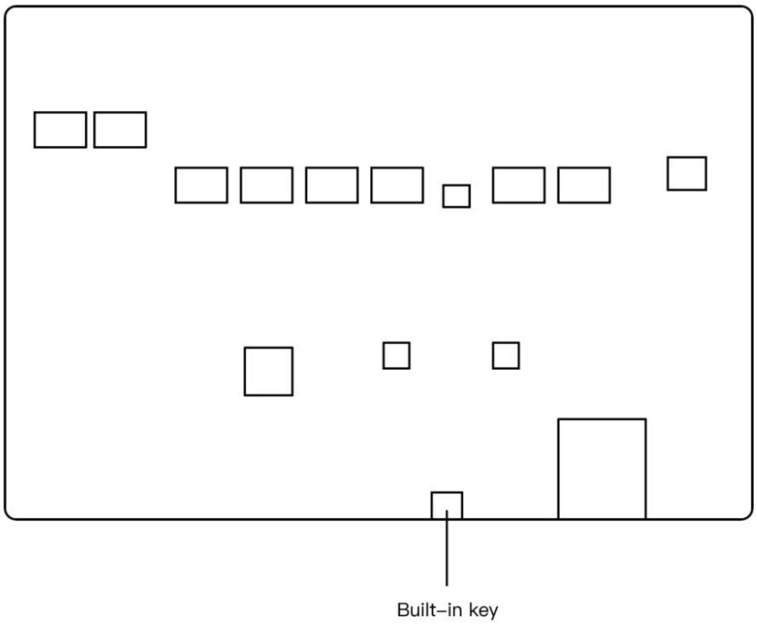


Figure 72 Built-in Key

Table 38 Description of K1 state

Key	Status	Description
Built-in Key	Press	BOOT_MODE1 level is pulled down to force **USB mode boot**

2.2.18. JTAG Test Points

There is a set of JTAG test points, which can be used for hardware debugging program burning, and system testing.

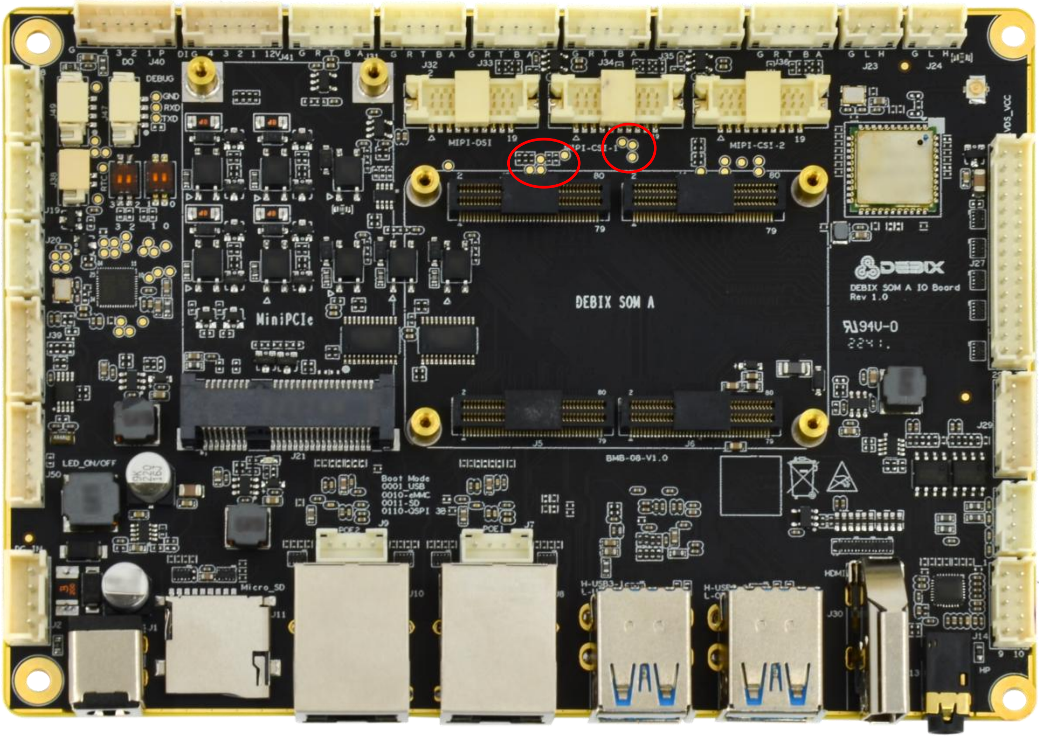


Figure 73 Debix Front View

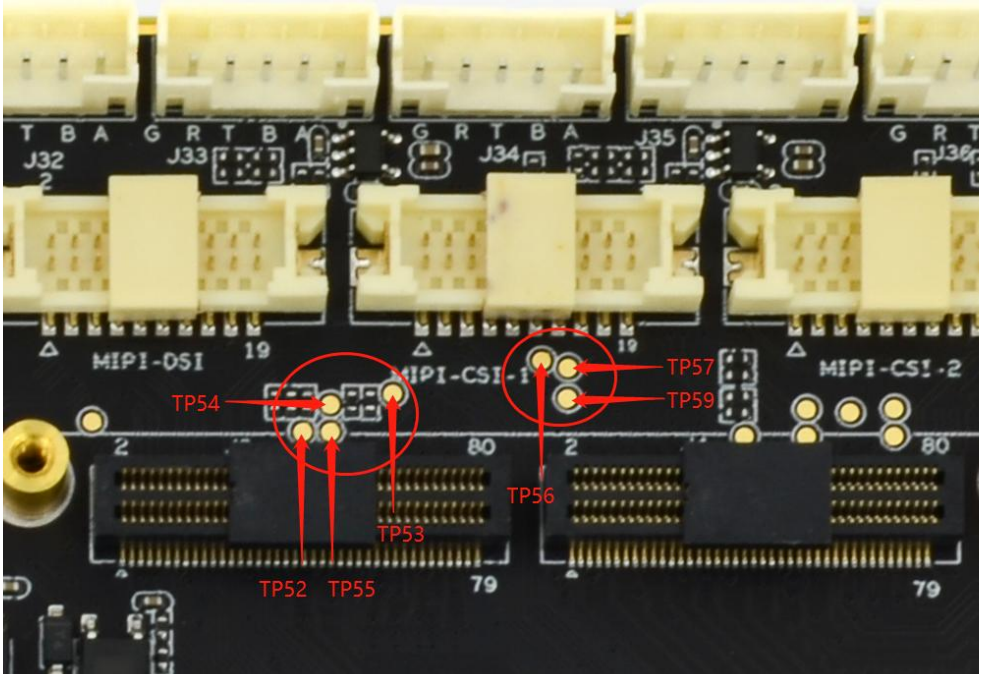


Figure 74 JTAG Test Points

The schematic of these JTAG test points is shown in the figure:

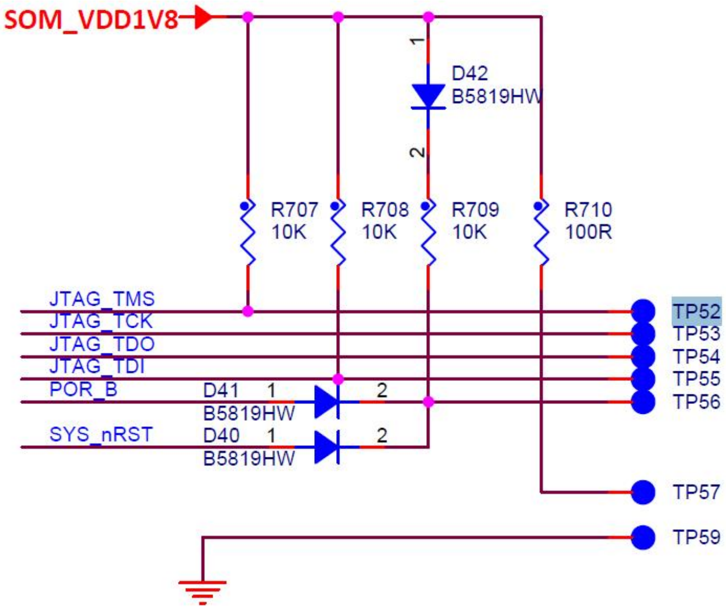


Figure 75 The Schematic of the JTAG Test Points

The JTAG test points are shown in the table below:

Table 39 The Definition of the JTAG Test Points

Pad	Definition
TP52	JTAG_TMS
TP53	JTAG_TCK
TP54	JTAG_TDO
TP55	JTAG_TDI
TP56	POR_B
TP57	SOM_VDD1V8
TP59	GND

2.3. Accessories Cable

DEBIX SOM A I/O Board is equipped with 11 types of cables, all of which are 20cm in length, as shown in the following table. For pin definitions, please refer to the detailed description of the interface.

Table 40 DEBIX SOM A I/O Board cables specification

No.	Name	Description	Location	Pin	Definition	Color	Number
1	RS232/RS485serial cable	AB20736W-R05-11, PH2.0-5A, 5pin/2.0mm pitch	J31~J36	1	RS485_A	yellow	6
				2	RS485_B	green	
				3	RS232_TXD	white	
				4	RS232_RXD	blue	
				5	GND	black	
2	CAN signal cable	AB20736W-R03-13, PH2.0-3A, 3pin/2.0mm pitch	J23, J24	1	CAN_H	white	2
				2	CAN_L	blue	
				3	GND	black	
3	DI signal cable	AB20736W-R06-13, PH2.0-6A, 6pin/2.0mm pitch	J41	1	DI_ISO	red	4
				2	DI_1	yellow	
				3	DI_2	green	
				4	DI_3	white	
				5	DI_4	blue	
				6	DI_GND	black	
	DO signal cable		J40	1	DO_COM	red	
				2	DO_1	yellow	
				3	DO_2	green	
				4	DO_3	white	
				5	DO_4	blue	
				6	DO_GND	black	
	I2C signal cable		J39	1	I2C-3.3V	red	
				2	I2C-SCL	yellow	

				3	I2C-SDA	green	
				4	I2C-RESET	white	
				5	I2C-nINT	blue	
				6	I2C-GND	black	
	LED signal cable		J50	1	DC_IN	red	
				2	VSYS_3V8	yellow	
				3	GPIO_LED1	green	
				4	GPIO_LED2	white	
				5	ONOFF	blue	
				6	GND	black	
4	LVDS backlight cable	AB20736W-R06-13, PH2.0-6A, 6pin/2.0mm pitch	J29	1	DC_IN	red	1
				2	DC_IN	red	
				3	EN	white	
				4	PWM	blue	
				5	GND	black	
				6	GND	black	
5	USB cable	AB20736W-R04-13, PH2.0-4A, 4pin/2.0mm pitch	J18, J19, J20	1	USB_5V	red	3
				2	DM	white	
				3	DP	green	
				4	GND	black	
6	SPK signal cable	AB20736W-R04-13, PH2.0-4A, 4pin/2.0mm pitch	J15	1	R+	yellow	3
				2	R-	green	
				3	L+	white	
				4	L-	blue	
	POE signal cable		J7, J9	1	POE_VA1	yellow	
				2	POE_VA2	green	
				3	POE_VB1	white	

				4	POE_VB2	blue	
7	SPDIF signal cable	1.25SMT4PIN, 4Pin/1.25mm pitch	J49	1	SPDIF_TX	yellow	2
				2	SPDIF_RX	green	
				3	SPDIF_CLK	white	
				4	GND	black	
	DEBUG signal cable		J47	1	VDD3V3	yellow	
				2	DEBUG_TXD	green	
				3	DEBUG_RXD	white	
				4	GND	black	
8	RTC battery cable	1.25SMT2PIN, 2Pin/1.25mm pitch	J38	1	RTC_CELL	red	1
				2	GND	black	
9	AUDIO audio cable	AB29266W-R2X5-01, 2*5Pin/2.0mm pitch	J14	1	MIC_R	yellow	1
				2	MIC_L	green	
				3	LIN	white	
				4	RIN	blue	
				5	AGND	black	
				6	AGND	black	
				7	NC	NC	
				8	HP_R	orange	
				9	HP_DET	gray	
				10	HP_L	purple	
10	LVDS cable	AB29266W-R2X15-01，2*15Pin/2.0mm pitch	J27	1	LVDS_VDD	red	1
				2	LVDS_VDD	red	
				3	LVDS_VDD	red	
				4	GND	black	
				5	GND	black	
				6	GND	black	

				7	A0_N	blue	
				8	A0_P	white	
				9	A1_N	blue	
				10	A1_P	white	
				11	A2_N	blue	
				12	A2_P	white	
				13	GND	black	
				14	GND	black	
				15	ACLK_N	blue	
				16	ACLK_P	white	
				17	A3_N	blue	
				18	A3_P	white	
				19	B0_N	blue	
				20	B0_P	white	
				21	B1_N	blue	
				22	B1_P	white	
				23	B2_N	blue	
				24	B2_P	white	
				25	GND	black	
				26	GND	black	
				27	BCLK_N	blue	
				28	BCLK_P	white	
				29	B3_N	blue	
				30	B3_P	white	
11	MIPI_CSI cable	AB97524W-R2X10-01, 2*10Pin/1.25mm pitch	J25, J26	1	VDD3V3	red	3
				2	VDD1V8	yellow	
				3	CKN	blue	

				4	CKP	white	
				5	GND	black	
				6	GND	green	
				7	DN0	blue	
				8	DP0	white	
				9	DN1	blue	
				10	DP1	white	
				11	GND	black	
				12	CSI_PWDN	orange	
				13	DN2	blue	
				14	DP2	white	
				15	DN3	blue	
				16	DP3	white	
				17	GND	black	
				18	CSI_nRST	gray	
				19	SCL	blue	
				20	SDA	white	
	MIPI_DSI cable		J28	1	VDD3V3	red	
				2	DSI_RST	yellow	
				3	CKN	blue	
				4	CKP	white	
				5	GND	black	
				6	DSI_nINT	green	
				7	DN0	blue	
				8	DP0	white	
				9	DN1	blue	
				10	DP1	white	

				11	GND	black	
				12	DSI_EN	orange	
				13	DN2	blue	
				14	DP2	white	
				15	DN3	blue	
				16	DP3	white	
				17	GND	black	
				18	DSI_PWM	gray	
				19	SCL	blue	
				20	SDA	white	

Chapter 3 Getting Started

3.1. Software Installation

3.1.1. Download Image

1. Download the latest system image from the software download page of DEBIX official website;

IMPORTANT

> The boot type of the image downloaded depends on which boot mode image you choose to install. For example, if you need to install an image with eMMC boot mode, you can choose the image name with (eMMC Flashing).

2. If the downloaded image file is a zip file, you need to decompress it into an .img file;
3. Write the .img file into the Micro SD card by balenaEtcher tool.

3.1.2. System Boot

DEBXI SOM A + I/O Board has three boot modes: eMMC (default), Micro SD card, SPI Nor Flash (reserved).

3.1.2.1. Boot from Micro SD

- **Component Preparation**

✓ DEBIX SOM A + I/O board

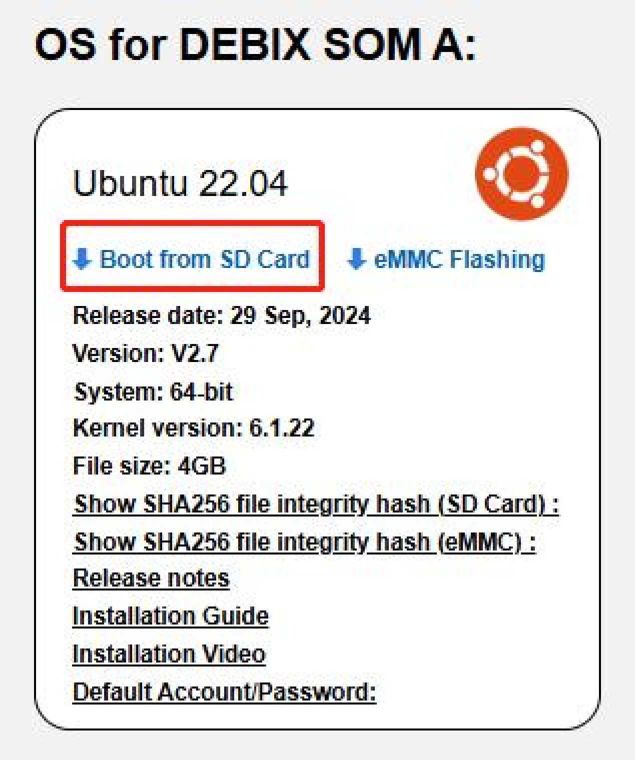
✓ Micro SD card, and card reader

✓ DC 12V/3A power adapter

✓ PC (windows 10/11)

- **Installing the Boot from Micro SD Card Image**

On the DEBIX official website, choose to download Ubuntu 22.04 Boot from SD Card image link: Debix-SOMA-SD-V2.4-202XXXXX.img, as shown below.



OS for DEBIX SOM A:

Ubuntu 22.04

Release date: 29 Sep, 2024
Version: V2.7
System: 64-bit
Kernel version: 6.1.22
File size: 4GB
Show SHA256 file integrity hash (SD Card) :
Show SHA256 file integrity hash (eMMC) :
Release notes
Installation Guide
Installation Video
Default Account/Password:

Figure 76

1. Install and open the Etcher tool on your PC, insert the Micro SD card, select the img file to be installed and the disk partition corresponding to the Micro SD card;

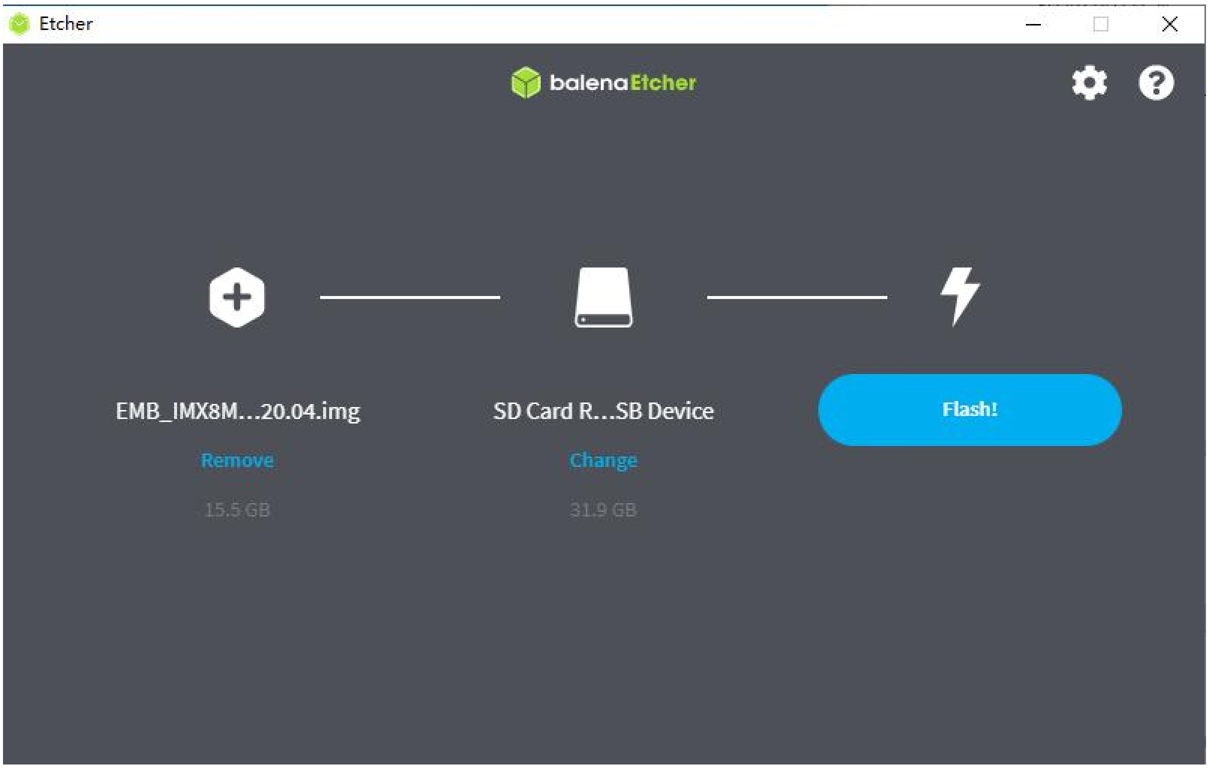


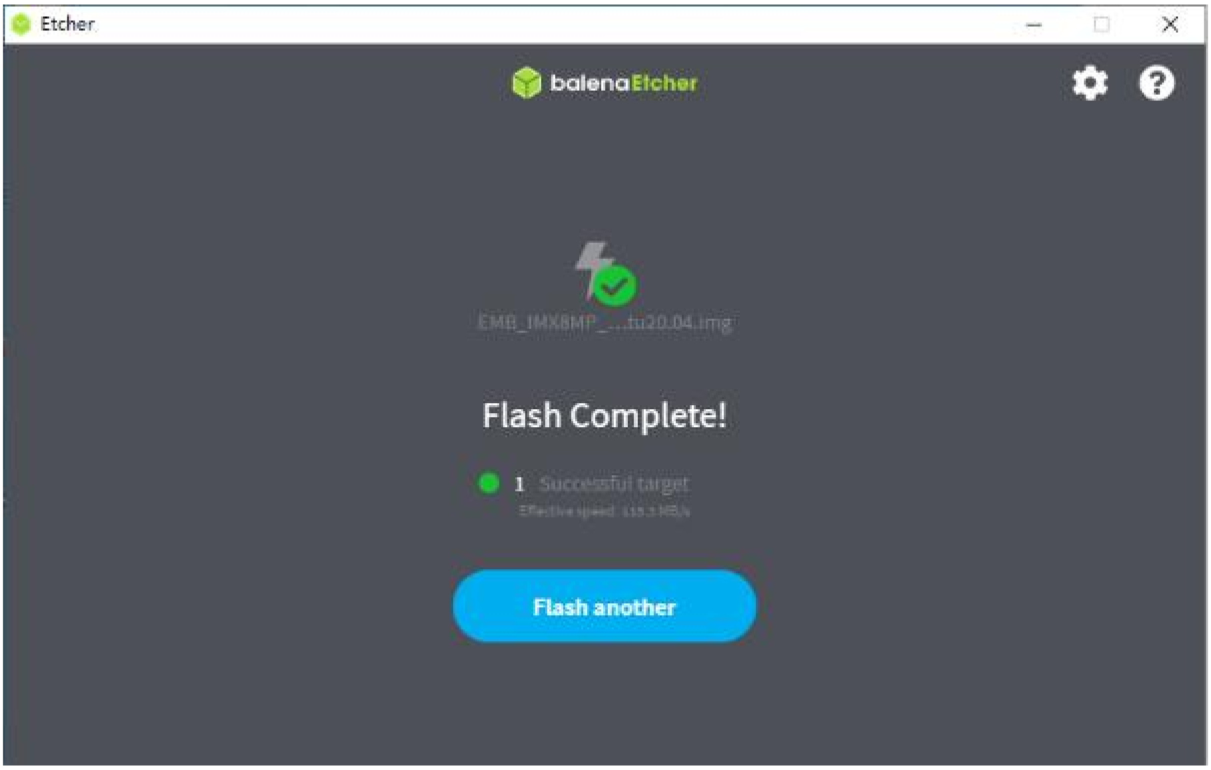
Figure 77

2. Click **Flash!** Wait patiently and the program will write the system to the Micro SD card;

NOTE

> The system may prompt you that the disk is unavailable and needs to be formatted, please ignore it, it is not an error!

3. When **Flash Complete!** appears, it means the system has been successfully programmed to the Micro SD card;



Flash Complete!

1 Successful target

Effective speed: [illegible]

Figure 78

4. Set the DIP switch to "0011" (boot from the Micro SD card), please refer to [Setting BOOT Mode by DIP Switch]() for Micro SD card boot mode.
5. Insert the Micro SD card into the slot of DEBIX SOM A I/O Board, connect the display and power on, then you can see the boot screen.

3.1.2.2. Boot from eMMC (default)

- **Component Preparation**

✓ DEBIX SOM A + I/O board

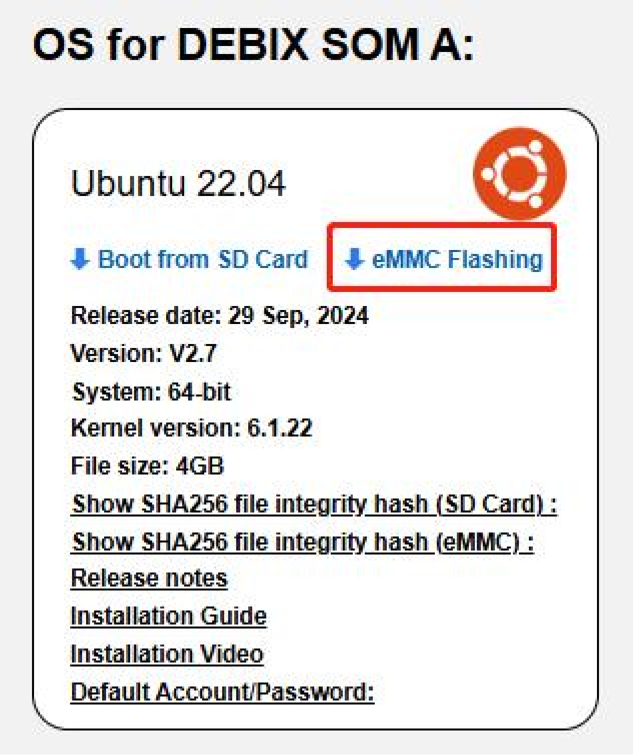
✓ Micro SD card above 16GB, and card reader

✓ DC 12V/3A power adapter

✓ PC (windows 10/11)

- **Installing the Boot from eMMC Image on the Micro SD Card**

On the DEBIX official website, choose to download Ubuntu 22.04 Boot from eMMC image link: Debix-SOMA-SD-UPGRADE-EMMC-V2.4-202XXXXX.img, as shown below.



OS for DEBIX SOM A:

Ubuntu 22.04

Release date: 29 Sep, 2024
Version: V2.7
System: 64-bit
Kernel version: 6.1.22
File size: 4GB
Show SHA256 file integrity hash (SD Card) :
Show SHA256 file integrity hash (eMMC) :
Release notes
Installation Guide
Installation Video
Default Account/Password:

Figure 79

Write the downloaded system image to the Micro SD card, and set the DIP switch to Micro SD card boot mode according to the steps 1-4 operation of "Boot from Micro SD Card". Then burn it to eMMC with the following steps:

1. Insert the Micro SD card into DEBIX SOM A I/O board, and power on. The system will automatically write to eMMC through the Micro SD card, this burn process will not be displayed on screen. When burning, the green LED on the motherboard will flash quickly, please wait. When the green LED changes from fast flash to slow flash, that is, the programming is complete.

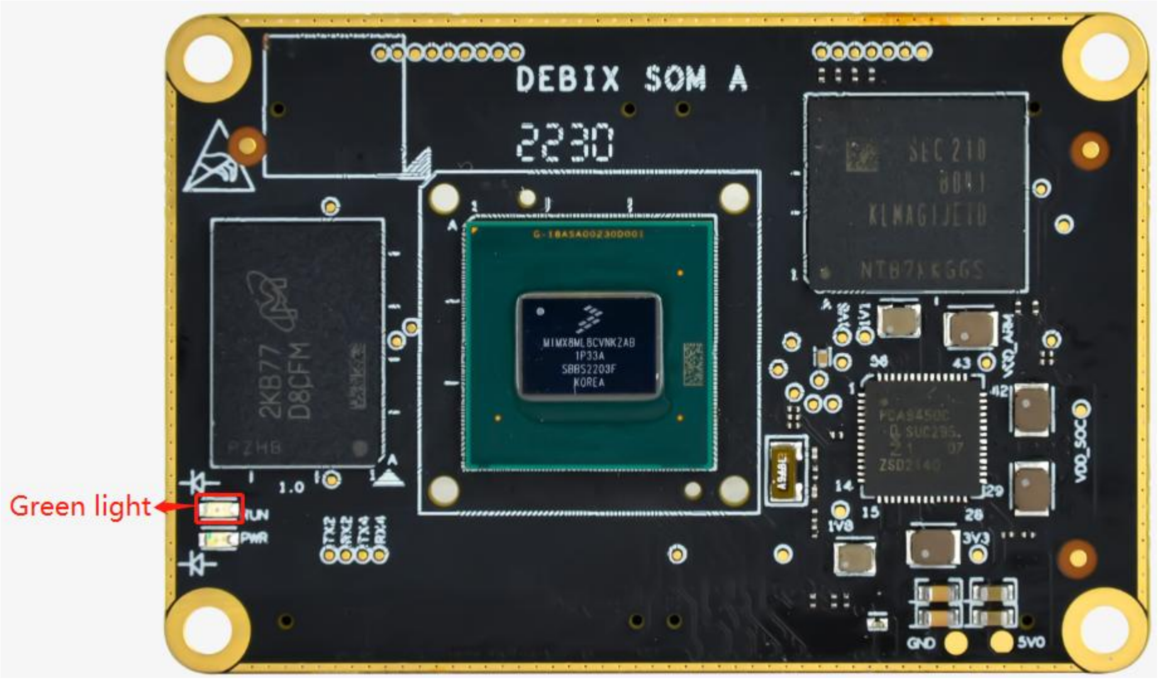


Figure 80

IMPORTANT

If the system with the same version as the Micro SD card has been burned to eMMC, the system will not be burned again, and the indicator light will not flash quickly.

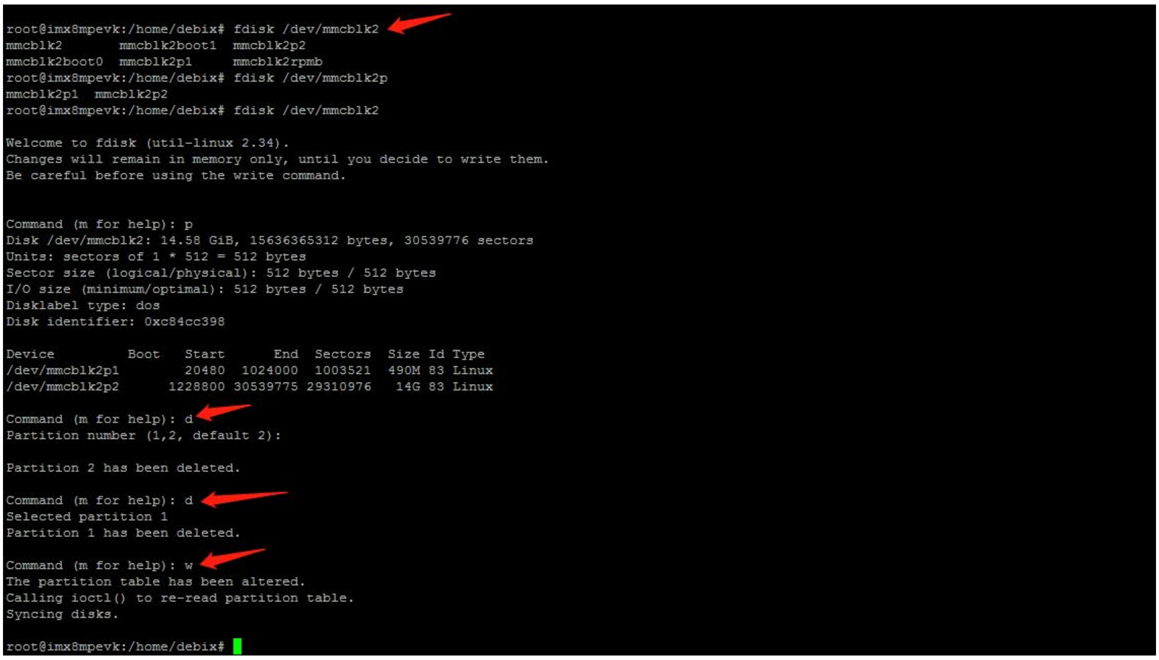
If you need to flash the eMMC system again, you need to format the eMMC first. Proceed as follows:

1) Connect the motherboard to the keyboard, mouse and HDMI display, set the DIP switch to "0011" to start the system from the Micro SD card, and power on.

2) In the Terminal, enter the default username “debix” and password "debix" to enter the command line, and run the following commands (as shown in the figure below):

```
#sudo su (password: debix)
#fdisk /dev/mmcblk2
d
d
w
```

3) Repeat step 2 to burn the system to eMMC again.



```
root@imx8mpevk:/home/debix# fdisk /dev/mmcblk2
mmcblk2       mmcblk2boot1  mmcblk2p2
mmcblk2boot0  mmcblk2p1     mmcblk2rpmb
root@imx8mpevk:/home/debix# fdisk /dev/mmcblk2p
mmcblk2p1  mmcblk2p2
root@imx8mpevk:/home/debix# fdisk /dev/mmcblk2

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/mmcblk2: 14.58 GiB, 15636365312 bytes, 30539776 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc84cc398

Device         Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk2p1        20480  1024000  1003521  490M 83 Linux
/dev/mmcblk2p2      1228800 30539775 29310976   14G 83 Linux

Command (m for help): d
Partition number (1,2, default 2):

Partition 2 has been deleted.

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@imx8mpevk:/home/debix# 
```

2. Disconnect the power supply, and set the DIP switch to "0010", the system will boot from eMMC, connect to HDMI and power on, then you can see the boot screen.

3.1.2.3. USB Flash

- **Component Preparation**

✓ DEBIX SOM A + I/O board

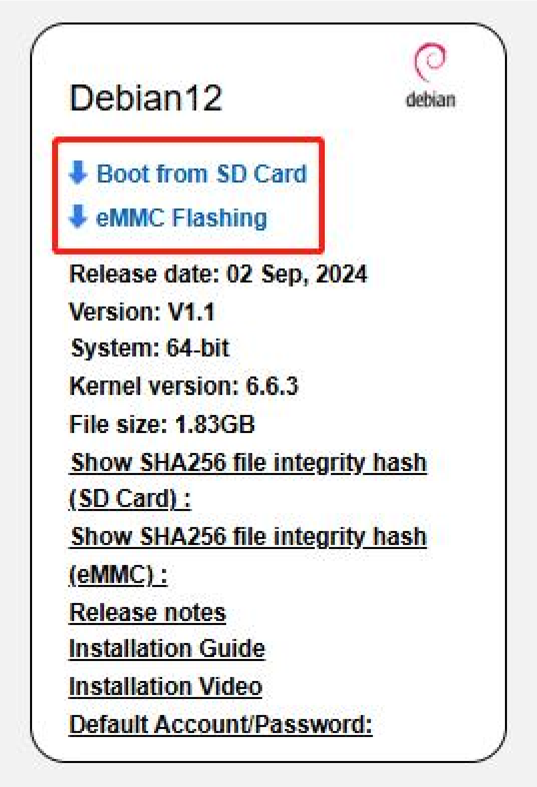
✓ Dual Male USB Type-A data cable

✓ DC 12V/3A power adapter

✓ PC (windows 10/11)

- **Burning to eMMC via USB**

1. Download the system installation package and UUU tool we provided to DEBIX, check the MD5 match after downloading, and then unzip it to PC;



Debian12

⬇ Boot from SD Card

⬇ eMMC Flashing

Release date: 02 Sep, 2024

Version: V1.1

System: 64-bit

Kernel version: 6.6.3

File size: 1.83GB

Show SHA256 file integrity hash (SD Card) :

Show SHA256 file integrity hash (eMMC) :

Release notes

Installation Guide

Installation Video

Default Account/Password:

Figure 81

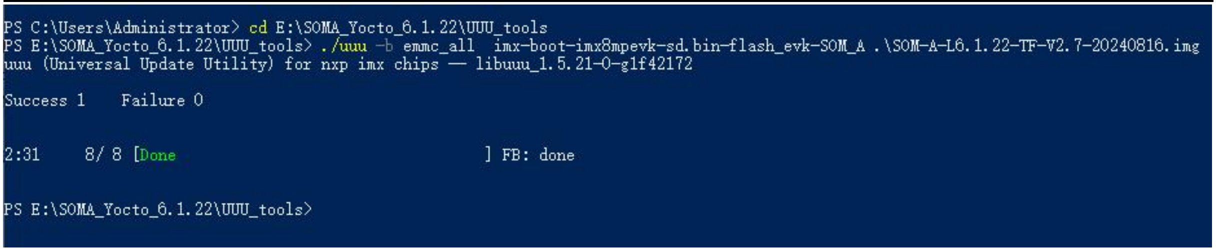
2. Use USB cable to connect the OTG port of DEBIX SOM A I/O Board to the USB port of PC, set the DIP switch to "0001" or press built-in key, connect the power supply, the system will enter the USB burning mode;
3. Run **Windows PowerShell** as administrator;
4. Type cd command to enter the root directory of the system installation package, for example:

```
cd E:\SOMA_Yocto_6.1.22\UUU_tools
```

5. Run the following command to download the file and start burning the system to eMMC;

```
./uuu -b emmc_all imx-boot-imx8mpevk-sd.bin-flash_evk-SOM_A SOM-A-L6.1.22-TF
-V2.7-20240816.img
```

6. Wait for the system burning to finish; when the terminal shows green "Done", it means the burning is finished;



```
PS C:\Users\Administrator> cd E:\SOMA_Yocto_6.1.22\UUU_tools
PS E:\SOMA_Yocto_6.1.22\UUU_tools> ./uuu -b emmc_all  imx-boot-imx8mpevk-sd.bin-flash_evk-SOM_A .\SOM-A-L6.1.22-TF-V2.7-20240816.img
uuu (Universal Update Utility) for nxp imx chips — libuuu_1.5.21-0-g1f42172

Success 1    Failure 0


2:31     8/ 8 [Done                                  ] FB: done


PS E:\SOMA_Yocto_6.1.22\UUU_tools>
```

7. After burning, disconnect the power supply and OTG USB cable, make sure the DEBIX is completely powered off, and then connect the power supply to start.

3.2. Hardware Installation

- **Hardware connection:** Refer to the board silkscreen, insert the 4 board-to-board Plug connectors of DEBIX SOM A correspondingly into the interfaces of DEBIX SOM A I/O Board (J3-J3, J4-J4, J5-J5, J6-J6), and gently plug them in to complete the installation, as shown in the figure below:

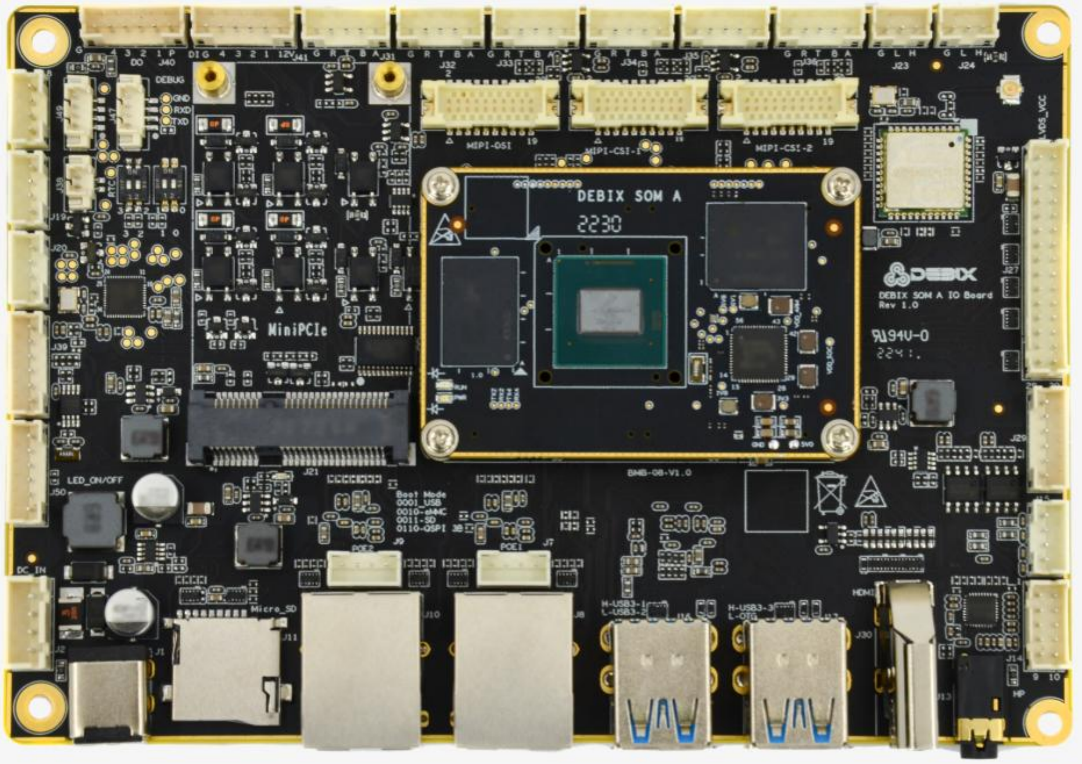


Figure 82 DEBIX SOM A + I/O Board

- **Peripheral connection:** as shown in the figure and the steps are as follows:

1. **Insert the Micro SD card with the system installed:** Insert it into the slot on DEBIX

SOM A I/O Board; if you need to remove it, just gently pull out the card after power off.

2. **Connect the HDMI monitor**
3. **Connect the keyboard**
4. **Connect the mouse**
5. **Connect the network cable**
6. **Connect the power adapter:** Plug in the power supply, DEBIX SOM A will power on, and the indicator light of motherboard will be on (if the boot fails, the indicator light will not be on).

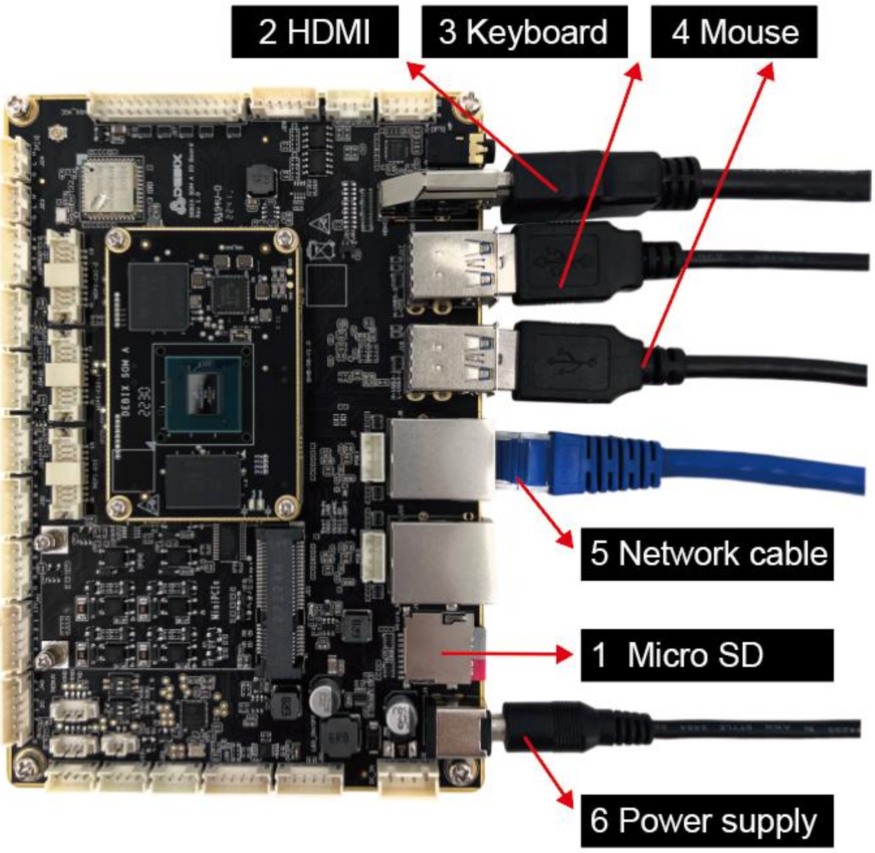


Figure 83

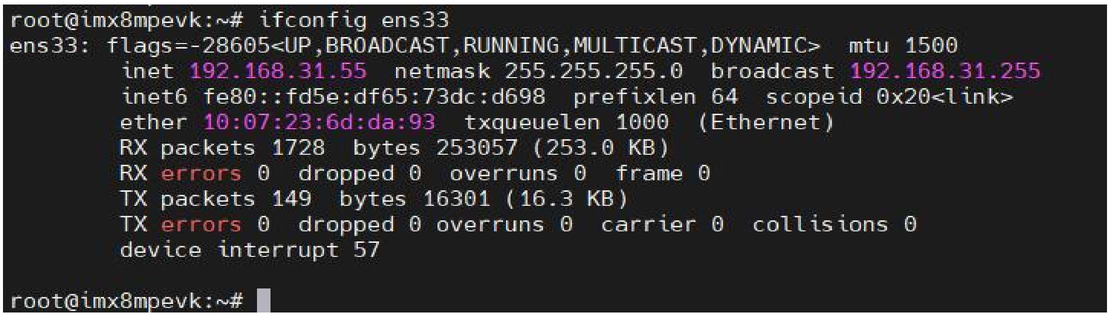
Chapter 4 Software Application Examples

4.1. Usage of Ethernet

- Network port 1 (ENET_QOS), bit number: J8, device node: ens33

1. Enter the system desktop, open a terminal and type the command to query network port 1;

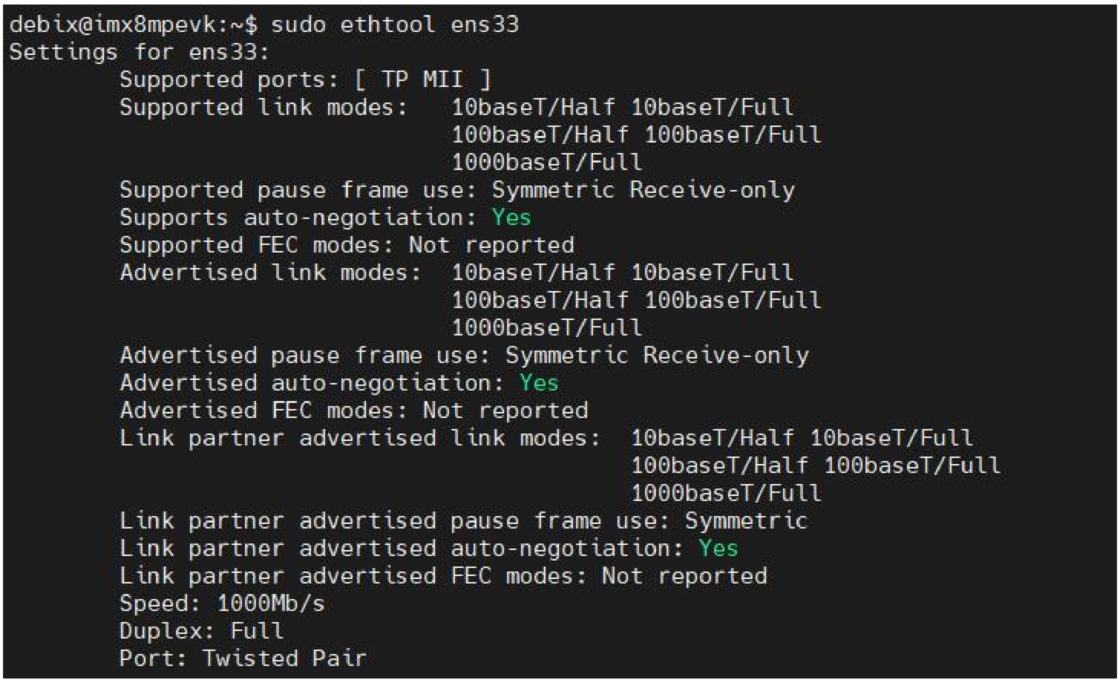
```
ifconfig ens33
```



```
root@imx8mpevk:~# ifconfig ens33
ens33: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.31.55  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::fd5e:df65:73dc:d698  prefixlen 64  scopeid 0x20<link>
        ether 10:07:23:6d:da:93  txqueuelen 1000  (Ethernet)
        RX packets 1728  bytes 253057 (253.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 149  bytes 16301 (16.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 57

root@imx8mpevk:~#
```

2. Query the speed of network port 1.

```
sudo ethtool ens33
```

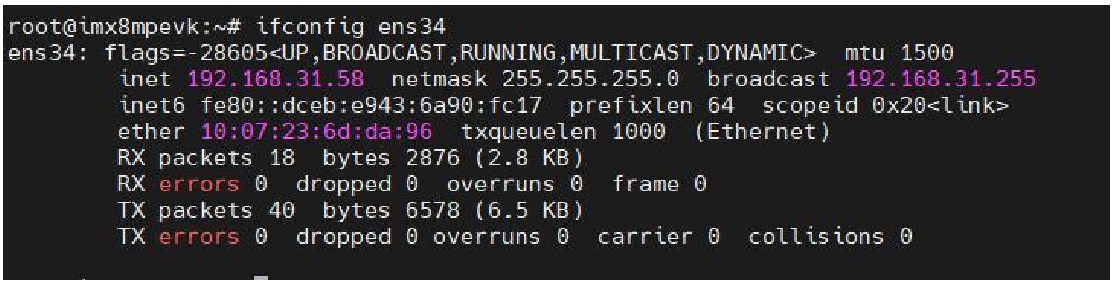


```
debix@imx8mpevk:~$ sudo ethtool ens33
Settings for ens33:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric Receive-only
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric Receive-only
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
                                             1000baseT/Full
        Link partner advertised pause frame use: Symmetric
        Link partner advertised auto-negotiation: Yes
        Link partner advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
```

- Network port 2 (ENET1) bit number: J10, device node: ens34

1. Type the command to query network port 2;

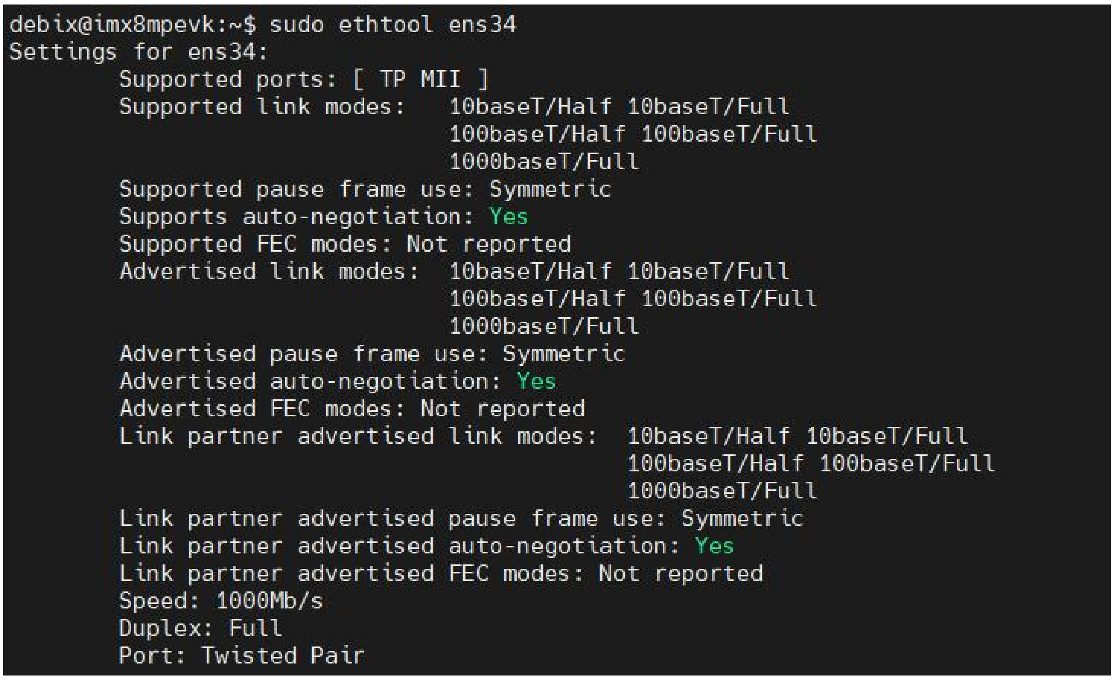
```
ifconfig ens34
```



```
root@imx8mpevk:~# ifconfig ens34
ens34: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.31.58  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::dceb:e943:6a90:fc17  prefixlen 64  scopeid 0x20<link>
        ether 10:07:23:6d:da:96  txqueuelen 1000  (Ethernet)
        RX packets 18  bytes 2876 (2.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 6578 (6.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

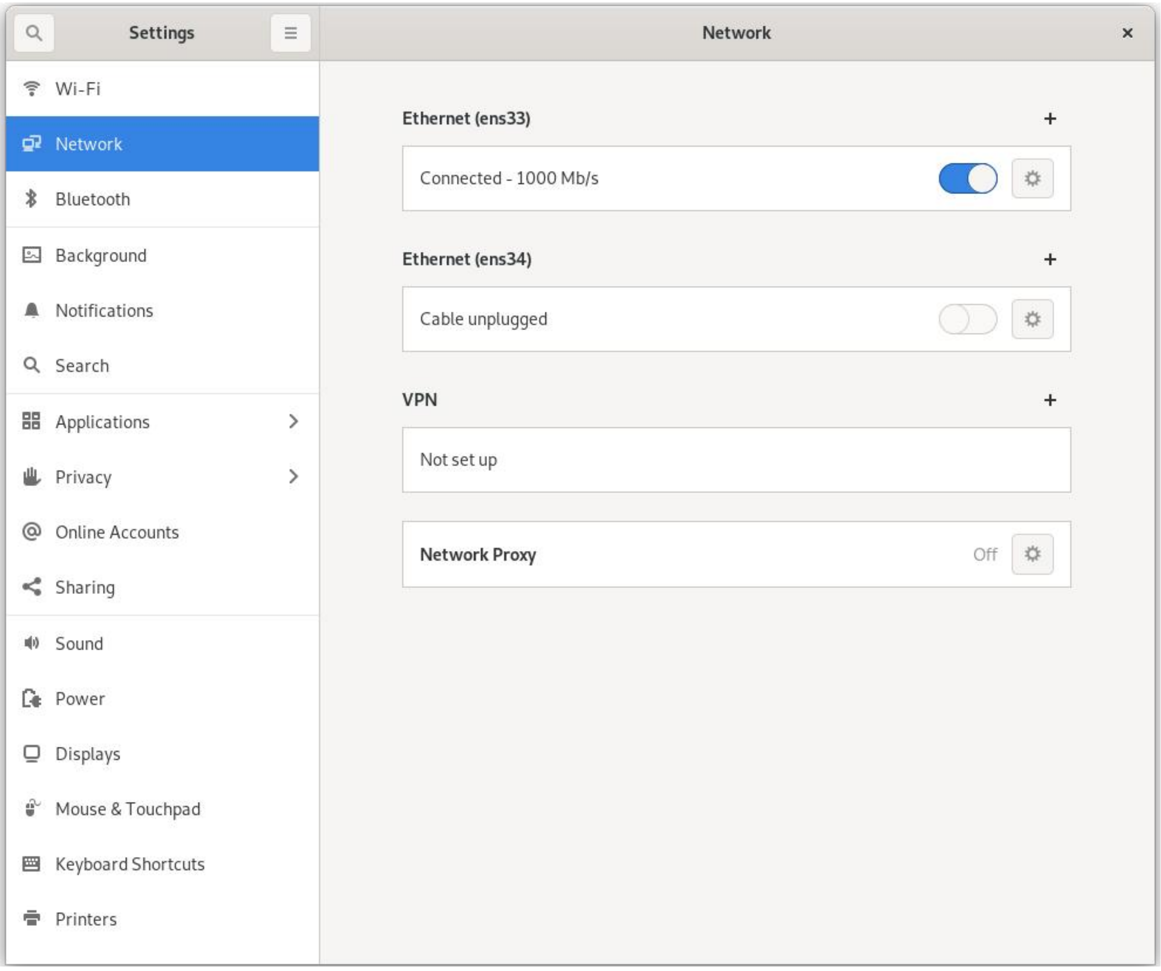
2. Query the speed of network port 2.

```
sudo ethtool ens34
```



```
debix@imx8mpevk:~$ sudo ethtool ens34
Settings for ens34:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
                                             1000baseT/Full
        Link partner advertised pause frame use: Symmetric
        Link partner advertised auto-negotiation: Yes
        Link partner advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
```

- **The desktop settings for Ethernet (Settings >> Network) are shown below:**



Connected - 1000 Mb/s

Cable unplugged

Not set up

Network Proxy Off

Figure 84 Ethernet desktop setting

4.2. Usage of USB

1. **Hardware connection:** Connect a USB 2.0 interface to the USB female connector, as shown in the figure below:

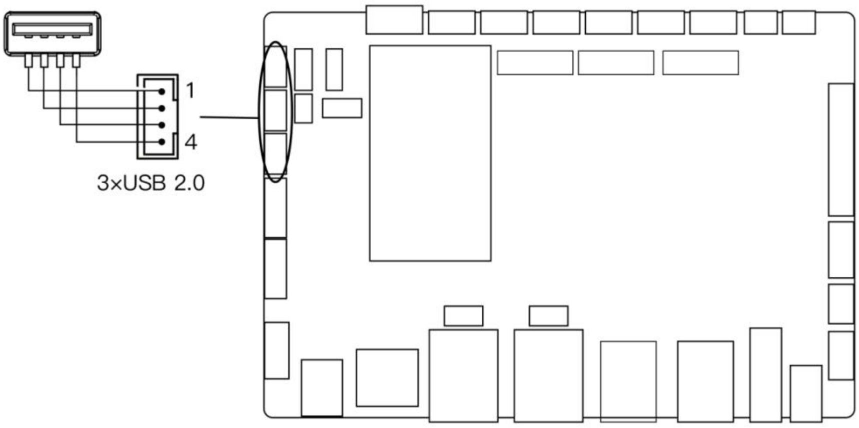


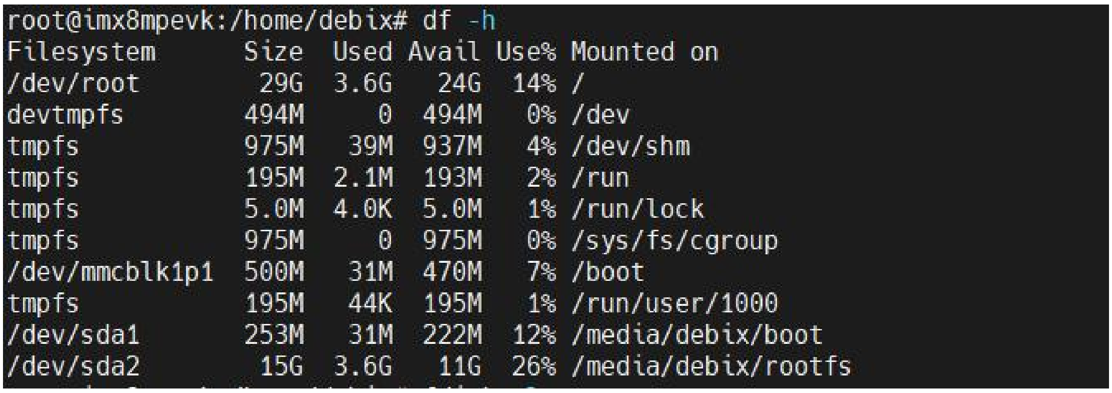
Figure 85 USB 2.0 connection

2. Enter the system desktop, open a terminal and switch to the root user;

```
sudo su
```

3. Access the U disk in FAT32 format, the system will automatically mount it to the /mnt path.

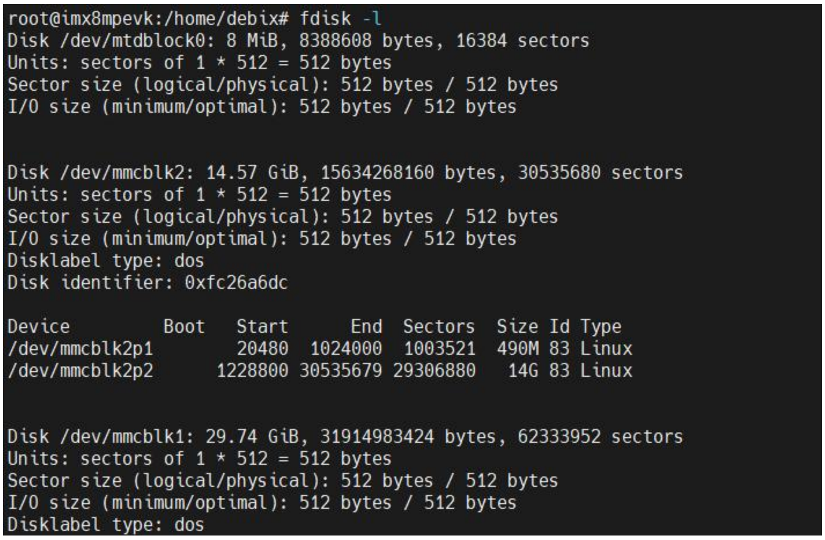
```
df -h
```



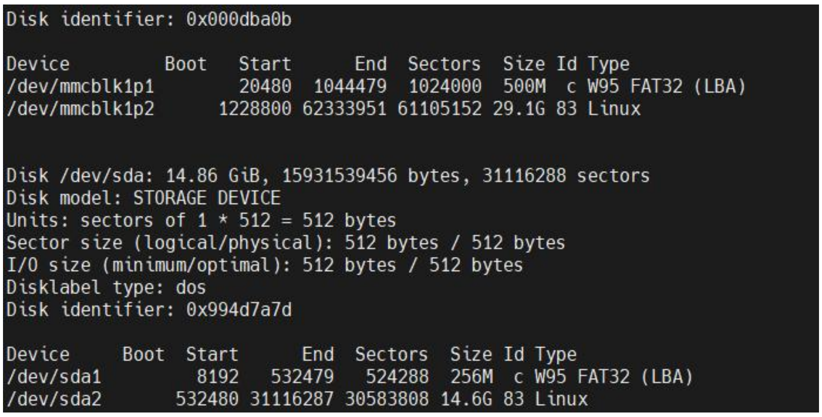
```
root@imx8mpevk:/home/debix# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        29G  3.6G   24G  14% /
devtmpfs        494M     0  494M   0% /dev
tmpfs           975M   39M  937M   4% /dev/shm
tmpfs           195M  2.1M  193M   2% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           975M     0  975M   0% /sys/fs/cgroup
/dev/mmcblk1p1  500M   31M  470M   7% /boot
tmpfs           195M   44K  195M   1% /run/user/1000
/dev/sda1       253M   31M  222M  12% /media/debix/boot
/dev/sda2        15G  3.6G   11G  26% /media/debix/rootfs
```

- If the U disk is not mounted, you can mount the U disk with the following command:
  - Query the U disk letter.

```
fdisk -l
```



```
root@imx8mpevk:/home/debix# fdisk -l
Disk /dev/mtdblock0: 8 MiB, 8388608 bytes, 16384 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mmcblk2: 14.57 GiB, 15634268160 bytes, 30535680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xfc26a6dc

Device         Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk2p1        20480  1024000  1003521  490M 83 Linux
/dev/mmcblk2p2      1228800 30535679 29306880   14G 83 Linux


Disk /dev/mmcblk1: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
```



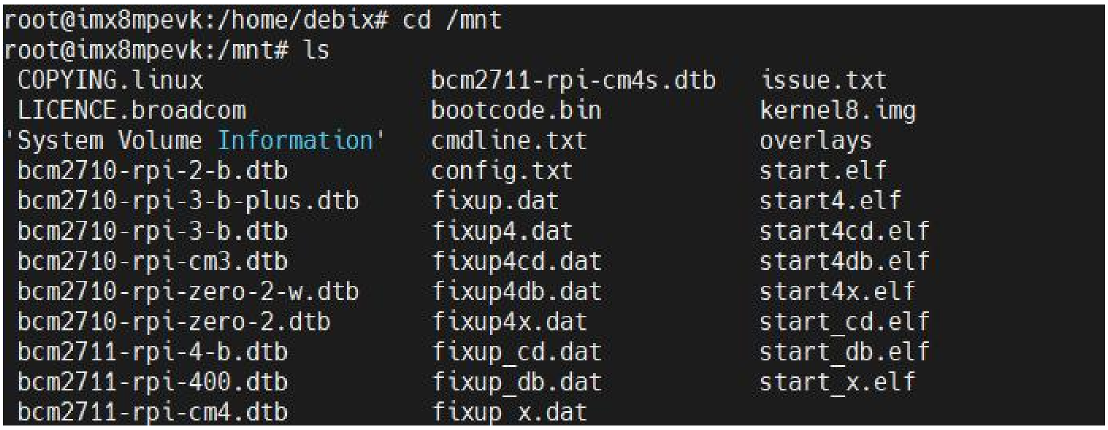
```
Disk identifier: 0x000dba0b

Device         Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk1p1        20480  1044479  1024000  500M  c W95 FAT32 (LBA)
/dev/mmcblk1p2      1228800 62333951 61105152 29.1G 83 Linux


Disk /dev/sda: 14.86 GiB, 15931539456 bytes, 31116288 sectors
Disk model: STORAGE DEVICE
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x994d7a7d

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sda1         8192   532479   524288  256M  c W95 FAT32 (LBA)
/dev/sda2       532480 31116287 30583808 14.6G 83 Linux
```

- Mount the U disk.

```
mount /dev/sda1 /mnt
```

4. Enter the U disk directory.

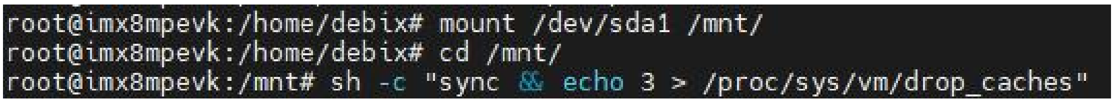
```
cd /mnt
```



```
root@imx8mpevk:/home/debix# cd /mnt
root@imx8mpevk:/mnt# ls
 COPYING.linux                bcm2711-rpi-cm4s.dtb   issue.txt
 LICENCE.broadcom             bootcode.bin           kernel8.img
'System Volume Information'   cmdline.txt            overlays
 bcm2710-rpi-2-b.dtb          config.txt             start.elf
 bcm2710-rpi-3-b-plus.dtb     fixup.dat              start4.elf
 bcm2710-rpi-3-b.dtb          fixup4.dat             start4cd.elf
 bcm2710-rpi-cm3.dtb          fixup4cd.dat           start4db.elf
 bcm2710-rpi-zero-2-w.dtb     fixup4db.dat           start4x.elf
 bcm2710-rpi-zero-2.dtb       fixup4x.dat            start_cd.elf
 bcm2711-rpi-4-b.dtb          fixup_cd.dat           start_db.elf
 bcm2711-rpi-400.dtb          fixup_db.dat           start_x.elf
 bcm2711-rpi-cm4.dtb          fixup_x.dat
```

5. Clear the cache: run before each read and write test command.

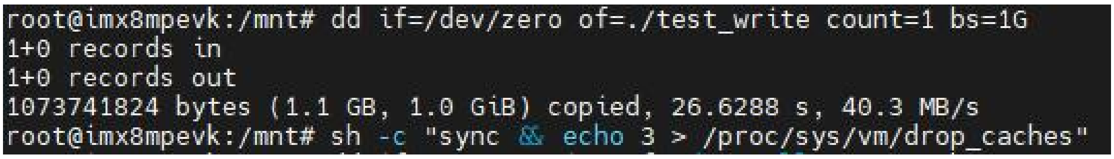
```
sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```



```
root@imx8mpevk:/home/debix# mount /dev/sda1 /mnt/
root@imx8mpevk:/home/debix# cd /mnt/
root@imx8mpevk:/mnt# sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```

6. Test write speed.

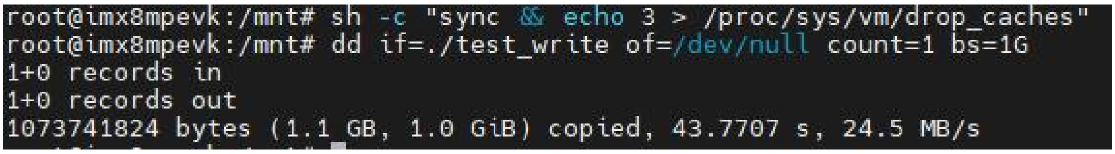
```
sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"     #clear cache

dd if=/dev/zero of=./test_write count=1 bs=1G
```



```
root@imx8mpevk:/mnt# dd if=/dev/zero of=./test_write count=1 bs=1G
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 26.6288 s, 40.3 MB/s
root@imx8mpevk:/mnt# sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```

7. Test read speed.

```
sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"      #clear cache

dd if=./test_write of=/dev/null count=1 bs=1G
```



```
root@imx8mpevk:/mnt# sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
root@imx8mpevk:/mnt# dd if=./test_write of=/dev/null count=1 bs=1G
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 43.7707 s, 24.5 MB/s
```

4.3. Nor-Flash Verification

There is a Nor-Flash on board, the functions are as follows:

Table 41 Nor Flash function node

Function Name	Capacity	Device Node
Nor-Flash	8MB	/dev/mtd0

1. Enter the system desktop, open a terminal and switch to the root user

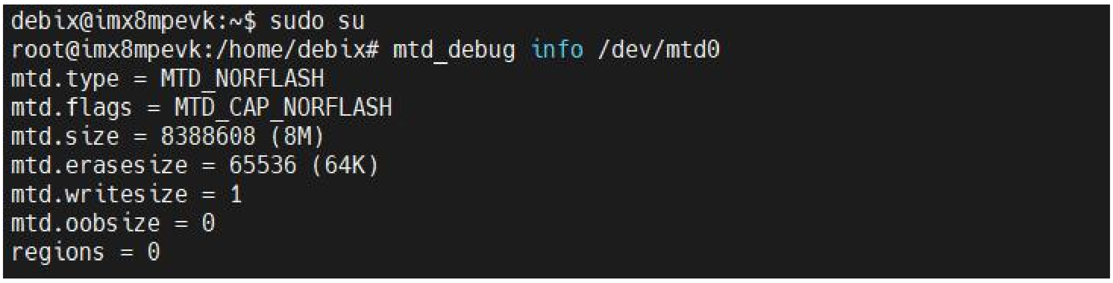
```
sudo su
```

2. Install mtd-utils via the command:

```
apt update

apt install mtd-utils
```

3. Query the Nor Flash information.

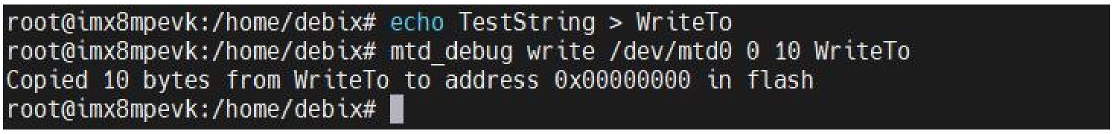
```
mtd_debug info /dev/mtd0
```



```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# mtd_debug info /dev/mtd0
mtd.type = MTD_NORFLASH
mtd.flags = MTD_CAP_NORFLASH
mtd.size = 8388608 (8M)
mtd.erasesize = 65536 (64K)
mtd.writesize = 1
mtd.oobsize = 0
regions = 0
```

4. Create a WriteTo document for write verification:

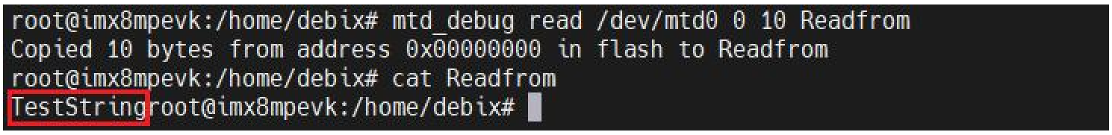
```
echo TestString > WriteTo

mtd_debug write /dev/mtd0 0 10 WriteTo
```



```
root@imx8mpevk:/home/debix# echo TestString > WriteTo
root@imx8mpevk:/home/debix# mtd_debug write /dev/mtd0 0 10 WriteTo
Copied 10 bytes from WriteTo to address 0x00000000 in flash
root@imx8mpevk:/home/debix#
```

5. Read verification:

```
mtd_debug read /dev/mtd0 0 10 Readfrom

cat Readfrom
```



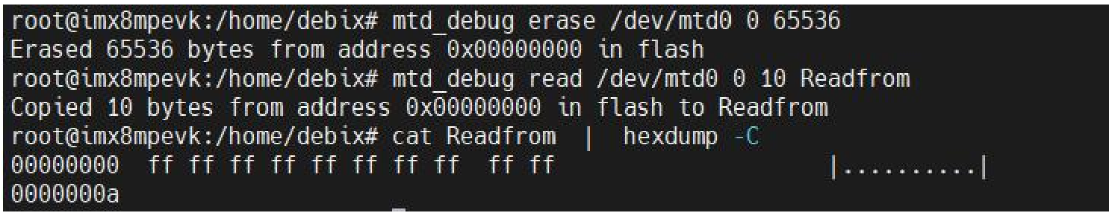
```
root@imx8mpevk:/home/debix# mtd_debug read /dev/mtd0 0 10 Readfrom
Copied 10 bytes from address 0x00000000 in flash to Readfrom
root@imx8mpevk:/home/debix# cat Readfrom
TestStringroot@imx8mpevk:/home/debix#
```

6. Erase Verification:

NOTE

The erase operation will set the Nor-Flash data to 1 and display it in ARCII format, which may be garbled.

```
mtd_debug erase /dev/mtd0 0 65536
mtd_debug read /dev/mtd0 0 10 Readfrom
cat Readfrom  |  hexdump -C                    #display in hexadecimal
```



```
root@imx8mpevk:/home/debix# mtd_debug erase /dev/mtd0 0 65536
Erased 65536 bytes from address 0x00000000 in flash
root@imx8mpevk:/home/debix# mtd_debug read /dev/mtd0 0 10 Readfrom
Copied 10 bytes from address 0x00000000 in flash to Readfrom
root@imx8mpevk:/home/debix# cat Readfrom  |  hexdump -C
00000000  ff ff ff ff ff ff ff ff  ff ff                    |..........|
0000000a
```

4.4. LED & Key

4.4.1. LED

The LED device nodes are described in the following table:

Table 42 Description of LED device node

Function Name	Device Node	File Path
LED	GPIO_LED1	/sys/devices/platform/gpio-leds/leds/GPIO_LED1
	GPIO_LED2	/sys/devices/platform/gpio-leds/leds/GPIO_LED2

The following is an example of GPIO_LED1:

1. **Hardware connection:** Connect two LEDs and one key to the J50 interface, as shown in the figure below:

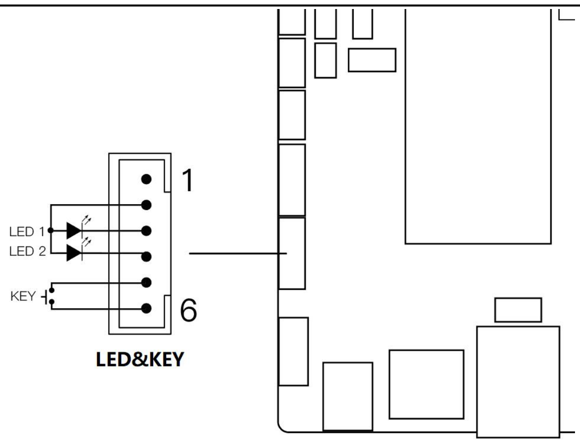


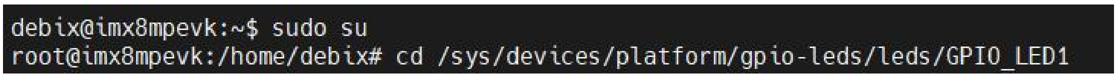
Figure 86

2. Enter the system desktop, open a terminal and switch to the root user;

```
sudo su
```

3. Switch to the GPIO_LED1 device file path

```
cd /sys/devices/platform/gpio-leds/leds/GPIO_LED1
```



```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/devices/platform/gpio-leds/leds/GPIO_LED1
```

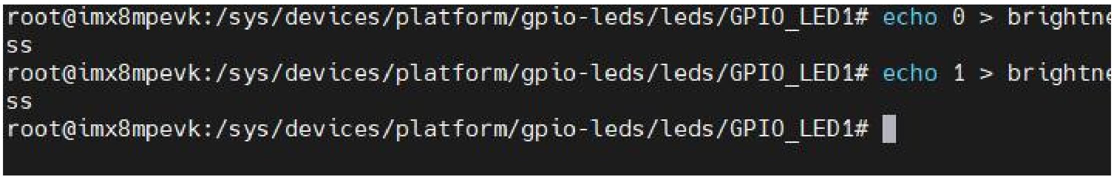
4. Output setting

- GPIO_LED1 outputs low level, LED1 is off.

```
echo 0 > brightness
```

- GPIO_LED1 outputs high level, LED1 is on.

```
echo 1 > brightness
```



```
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/GPIO_LED1# echo 0 > brightne
ss
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/GPIO_LED1# echo 1 > brightne
ss
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/GPIO_LED1# 
```

4.4.2. Key

DEBIX SOM A I/O Board automatically starts up when powered on.

Table 43 Description of KEY function

Function Name	Status	Description
KEY	Short Press	● When the green light is off, the system enters the sleep state ● Short press again, the green light blinks to wake up the system
	Long Press	● Press and hold until the green light turns off to shut down the device ● Press and hold again until the green light stays on to power on

4.5. UART/RS232/RS485/CAN

4.5.1. UART

NOTE

The IO level of the debugging serial port is 3.3V.

1. **Hardware connection:** Connect a UART interface to the USB-TTL module, as shown in the figure below:
   - Connect the RXD of the debugging serial port to the TXD port of the USB-TTL module
   - Connect the TXD of the debugging serial port to the RXD port of the USB-TTL module
   - Connect the GND of the debugging serial port to the GND port of the USB-TTL module

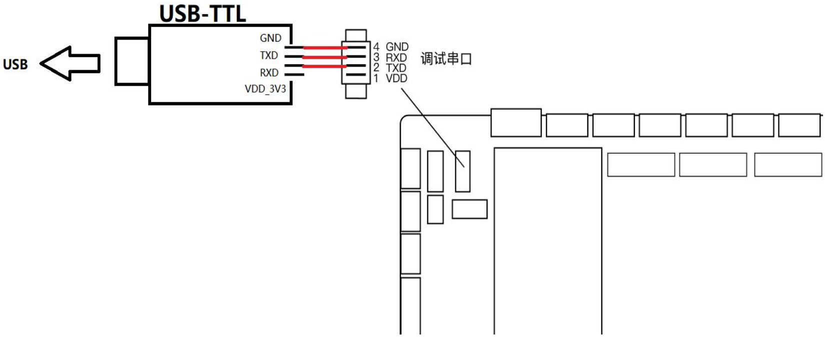


Figure 87

2. Open the **Windows Device Manager** and check the serial port number of the USB-TTL 3.3V device.

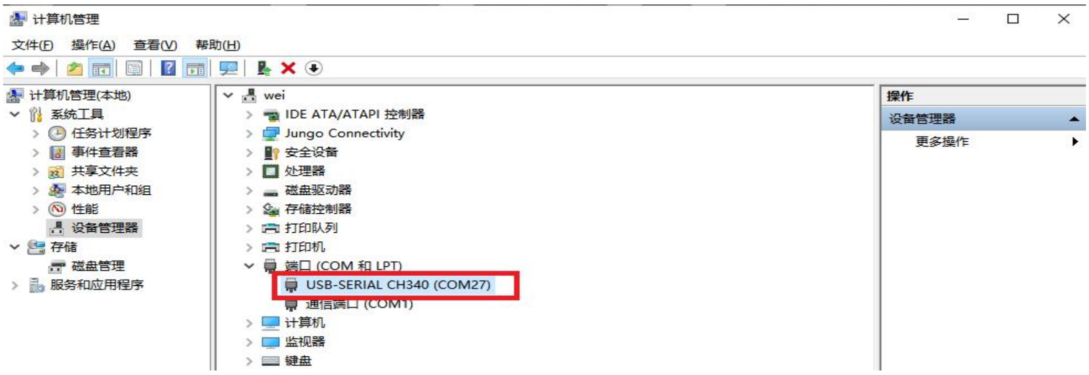


Figure 88

3. Open MobaXterm, click **Sessions** on the menu bar, select **New session**.

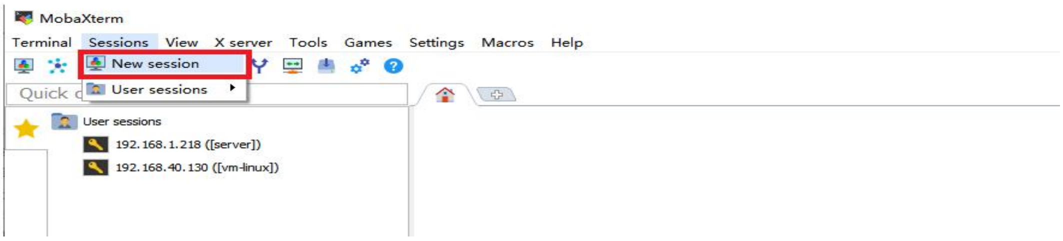


Figure 89

4. Select **Serial** in the pop-up "Session settings" dialog box.

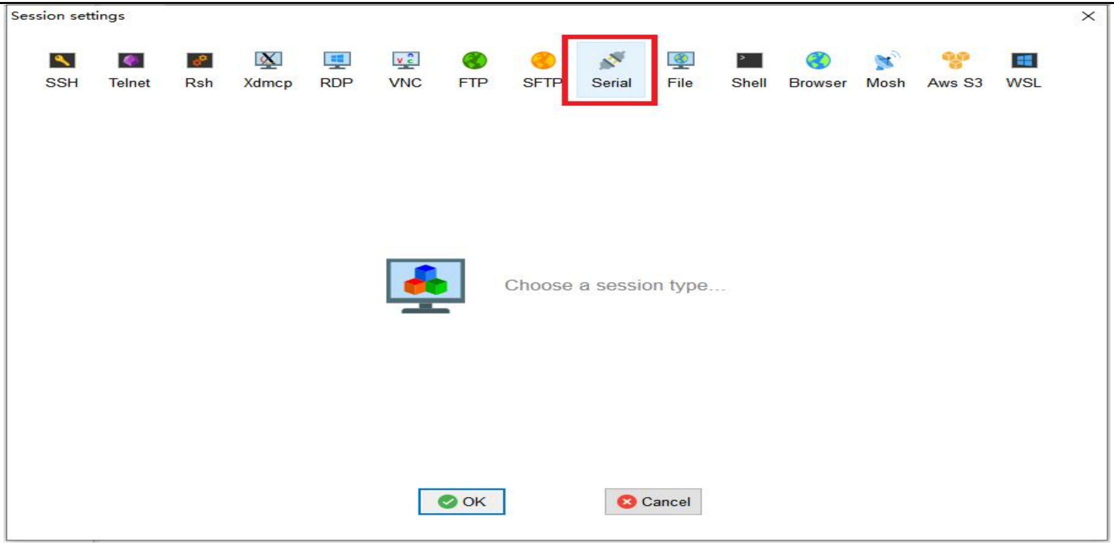
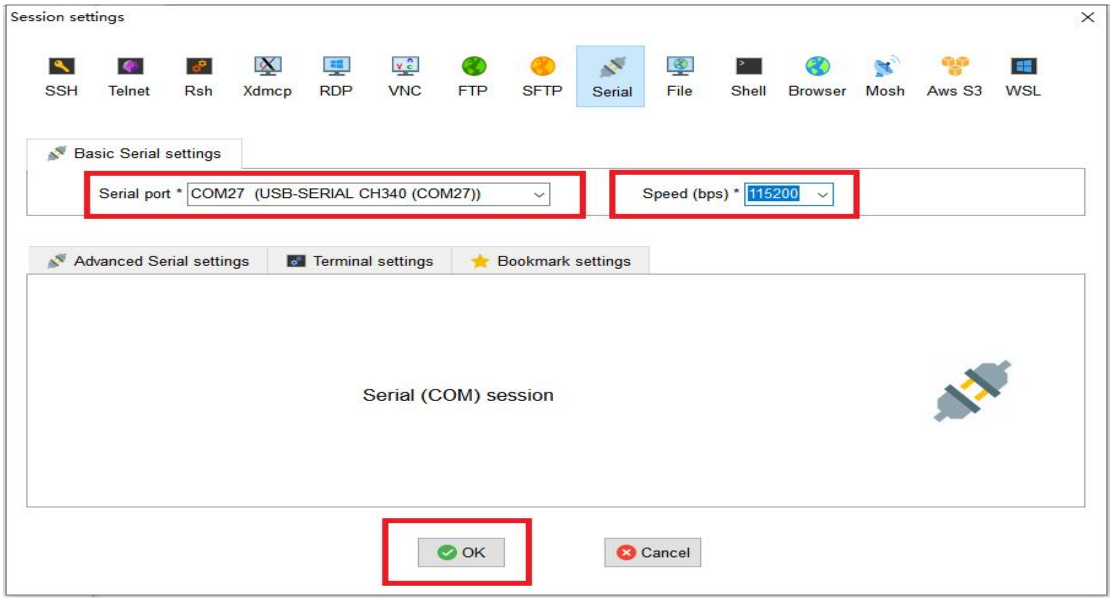


Figure 90

5. Change the port number to the COM port found in the device manager, set the Speed(bps) to **115200**, and click **OK**.

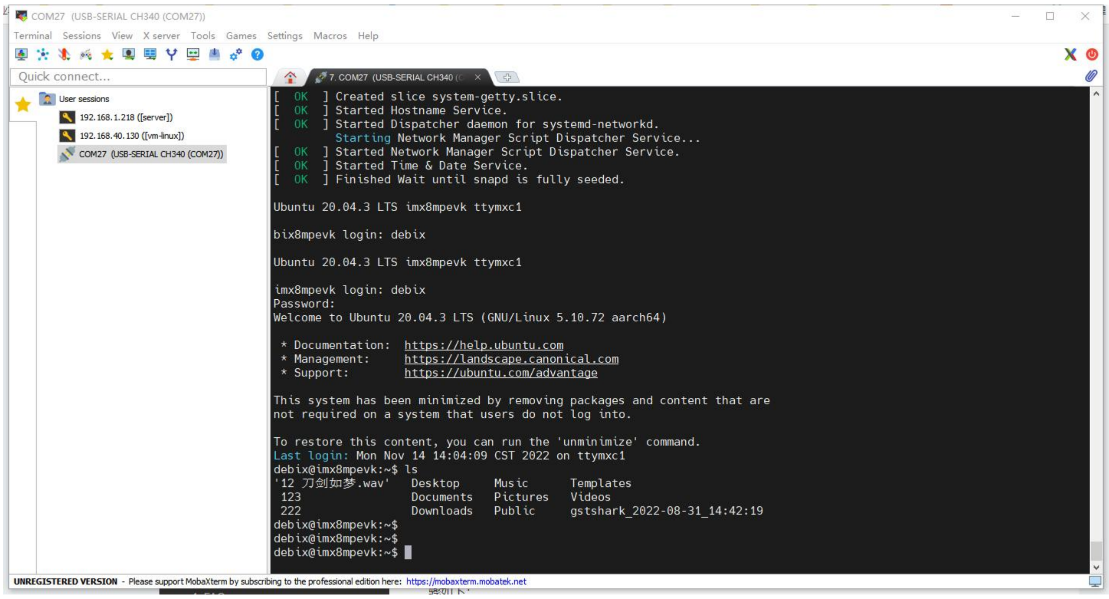


Serial port * COM27 (USB-SERIAL CH340 (COM27))

Speed (bps) * 115200

Figure 91

6. On the terminal, you can see the boot process logs output by Uboot, Kernel, and System. After the system boot is complete, type the **default user** and **password** (both **debix**) on the terminal to enter the serial console.


COM27 (USB-SERIAL CH340 (COM27))
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
7. COM27 (USB-SERIAL CH340 (C
User sessions
192.168.1.218 ([server])
192.168.40.130 ([vm-linux])
COM27 (USB-SERIAL CH340 (COM27))

```
[  OK  ] Created slice system-getty.slice.
[  OK  ] Started Hostname Service.
[  OK  ] Started Dispatcher daemon for systemd-networkd.
         Starting Network Manager Script Dispatcher Service...
[  OK  ] Started Network Manager Script Dispatcher Service.
[  OK  ] Started Time & Date Service.
[  OK  ] Finished Wait until snapd is fully seeded.

Ubuntu 20.04.3 LTS imx8mpevk ttymxc1

bix8mpevk login: debix

Ubuntu 20.04.3 LTS imx8mpevk ttymxc1

imx8mpevk login: debix
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.10.72 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Mon Nov 14 14:04:09 CST 2022 on ttymxc1
debix@imx8mpevk:~$ ls
'12 刀剑如梦.wav'   Desktop     Music      Templates
 123               Documents   Pictures   Videos
 222               Downloads   Public     gstshark_2022-08-31_14:42:19
debix@imx8mpevk:~$
debix@imx8mpevk:~$
debix@imx8mpevk:~$ 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Figure 92

4.5.2. RS232

IMPORTANT

RS232 and RS485 on the same connector, only one can used at one time.

NOTE

RS232/RS485 supports multiple baud rates, just set the baud rate of both communication parties to be the same. In this example, the baud rate of both communication parties is set to 115200.

1. **Hardware connection:** Connect a RS232 interface (J31 for example) to the USB-RS232 module, as shown in the figure below:
   - Pin3 of J31 is connected to the receiving end of USB-RS232
   - Pin4 of J31 is connected to the sending end of USB-RS232
   - Pin5 of J31 is connected to the ground terminal of USB-RS232
   - USB-RS232 is connected to the onboard USB 3.0 interface

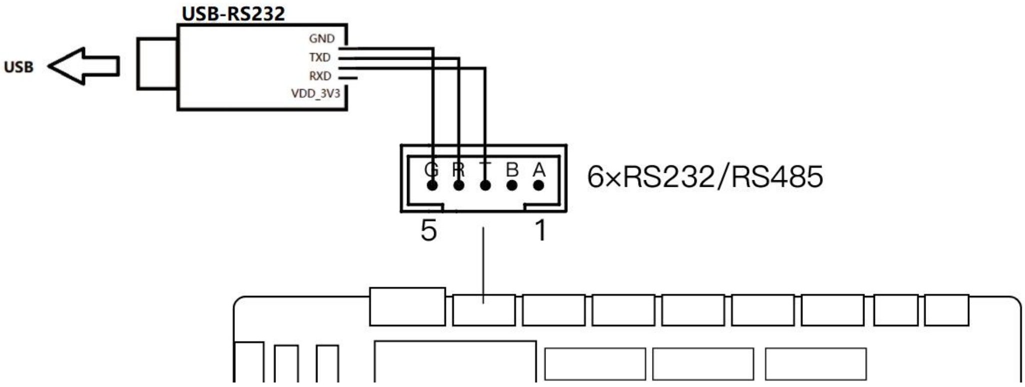


Figure 93

2. Open a Terminal on the DEBIX SOM A and run the following command to install the cutecom serial port tool:

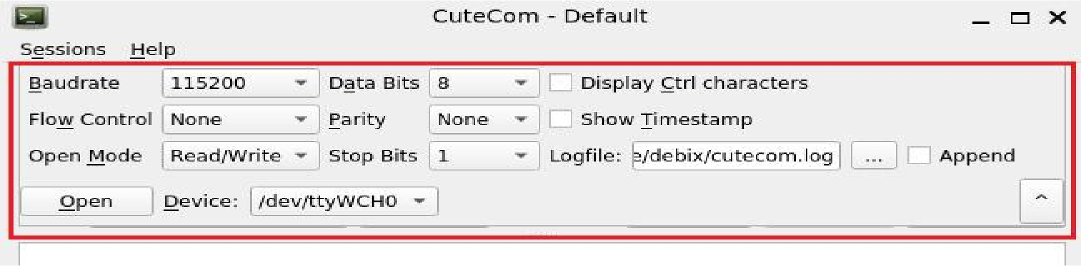
```
sudo apt update
sudo apt install cutecom qtwayland5
```

3. Open the cutecom tool and set the serial port parameters as shown in the following table:

Table 44 Cutecom parameter settings

Parameter	Value
Baudrate	115200
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

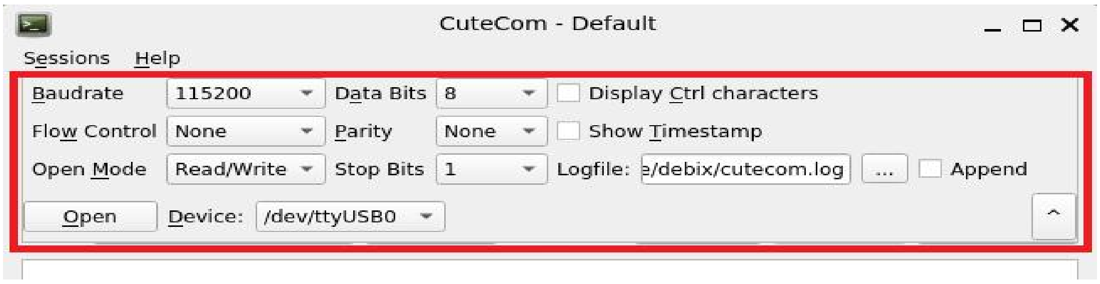
4. Run the command cutecom, and set the **Device** to /dev/ttyWCH0, set other parameters as shown in the table, click **Open**.



Baudrate 115200 Data Bits 8 □ Display Ctrl characters
Flow Control None Parity None □ Show Timestamp
Open Mode Read/Write Stop Bits 1 Logfile: e/debix/cutecom.log □ Append
Device: /dev/ttyWCH0

Figure 94

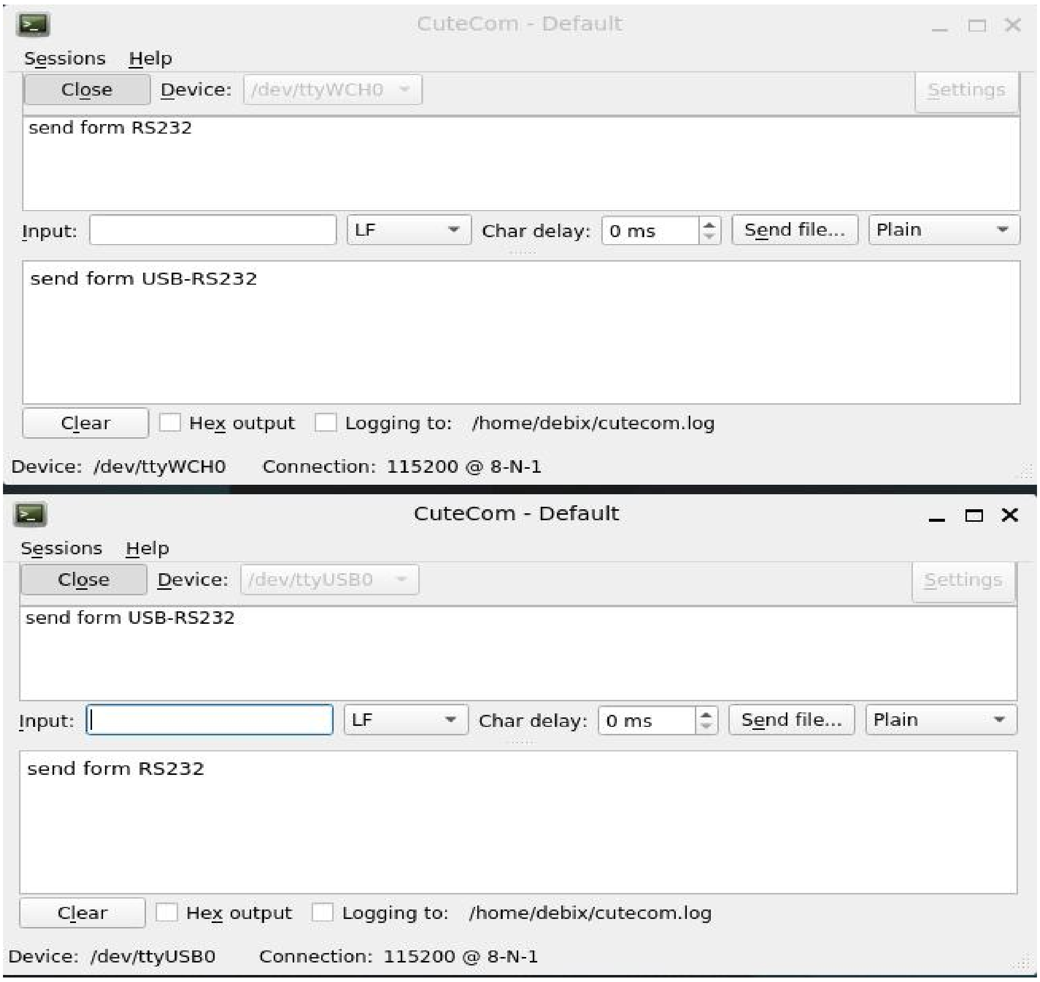
5. Open another cutecom serial port tool, set **Device** to /dev/ttyUSB0, set other parameters as follows, click **Open**.



Baudrate 115200 Data Bits 8 □ Display Ctrl characters
Flow Control None Parity None □ Show Timestamp
Open Mode Read/Write Stop Bits 1 Logfile: e/debix/cutecom.log □ Append
Device: /dev/ttyUSB0

Figure 95

6. **Send and receive data via cutecom:** Type the test string in the cutecom input box, press **Enter** to send, you can see that another cutecom receiving box receives the same message, indicating that the communication is successful, and the result is as follows:



CuteCom - Default

Device: /dev/ttyWCH0

send form RS232

Input: LF Char delay: 0 ms Plain

send form USB-RS232

□ Hex output □ Logging to: /home/debix/cutecom.log

Device: /dev/ttyWCH0 Connection: 115200 @ 8-N-1

CuteCom - Default

Device: /dev/ttyUSB0

send form USB-RS232

Input: LF Char delay: 0 ms Plain

send form RS232

□ Hex output □ Logging to: /home/debix/cutecom.log

Device: /dev/ttyUSB0 Connection: 115200 @ 8-N-1

Figure 96

4.5.3. RS485

1. **Hardware connection:** Connect two RS485 interfaces (J33 and J34 for example), as shown in the figure below:
   - Pin1 of J33 is connected to Pin1 of J34 (that is, A to A)
   - Pin2 of J33 is connected to Pin2 of J34 (that is, B to B)

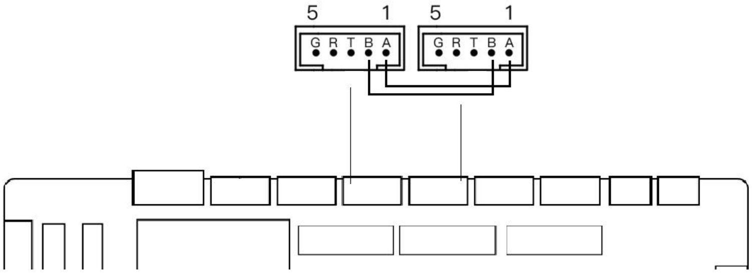
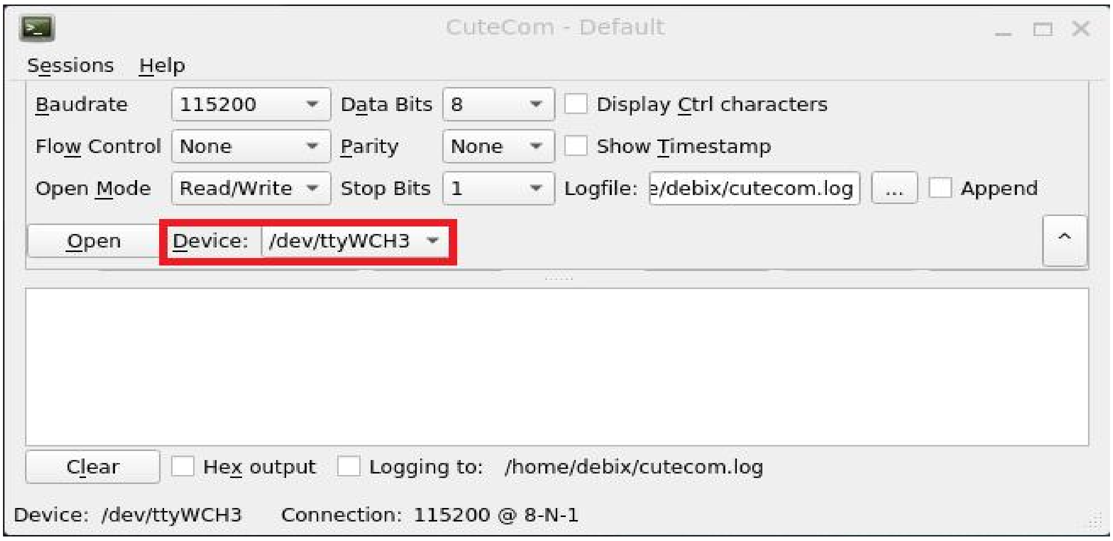


Figure 97

2. Set the **Device** of the cutecom tool to /dev/ttyWCH3, click **Open**.



Baudrate 115200 | Data Bits 8 | □ Display Ctrl characters

Flow Control None | Parity None | □ Show Timestamp

Open Mode Read/Write | Stop Bits 1 | Logfile: e/debix/cutecom.log | ... | □ Append

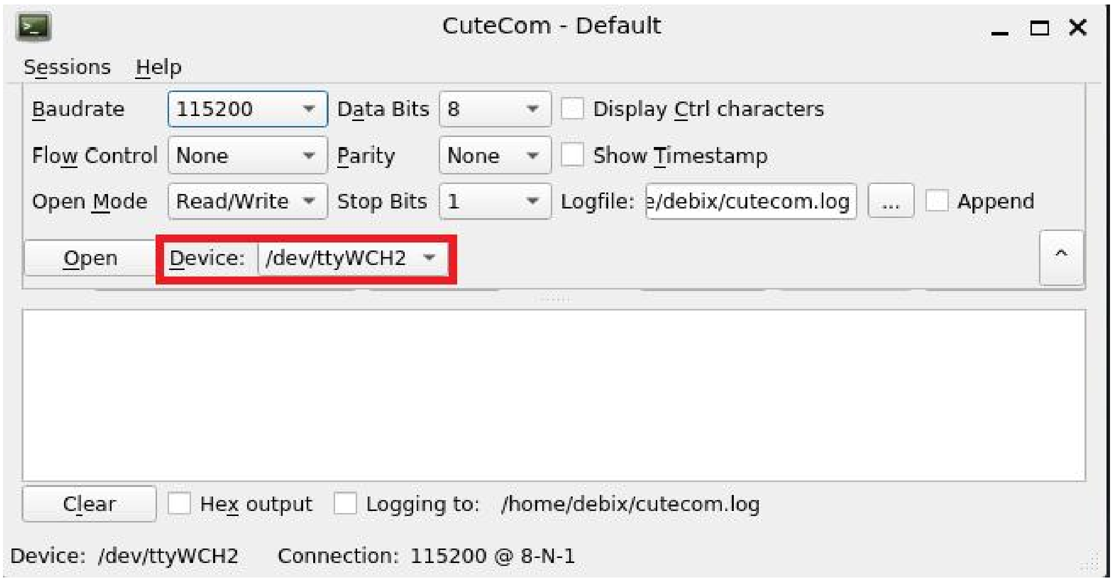
Device: /dev/ttyWCH3

□ Hex output □ Logging to: /home/debix/cutecom.log

Device: /dev/ttyWCH3 Connection: 115200 @ 8-N-1

Figure 98

3. Set the **Device** of another cutecom tool to /dev/ttyWCH2, click **Open**.



Baudrate 115200 | Data Bits 8 | □ Display Ctrl characters

Flow Control None | Parity None | □ Show Timestamp

Open Mode Read/Write | Stop Bits 1 | Logfile: e/debix/cutecom.log | ... | □ Append

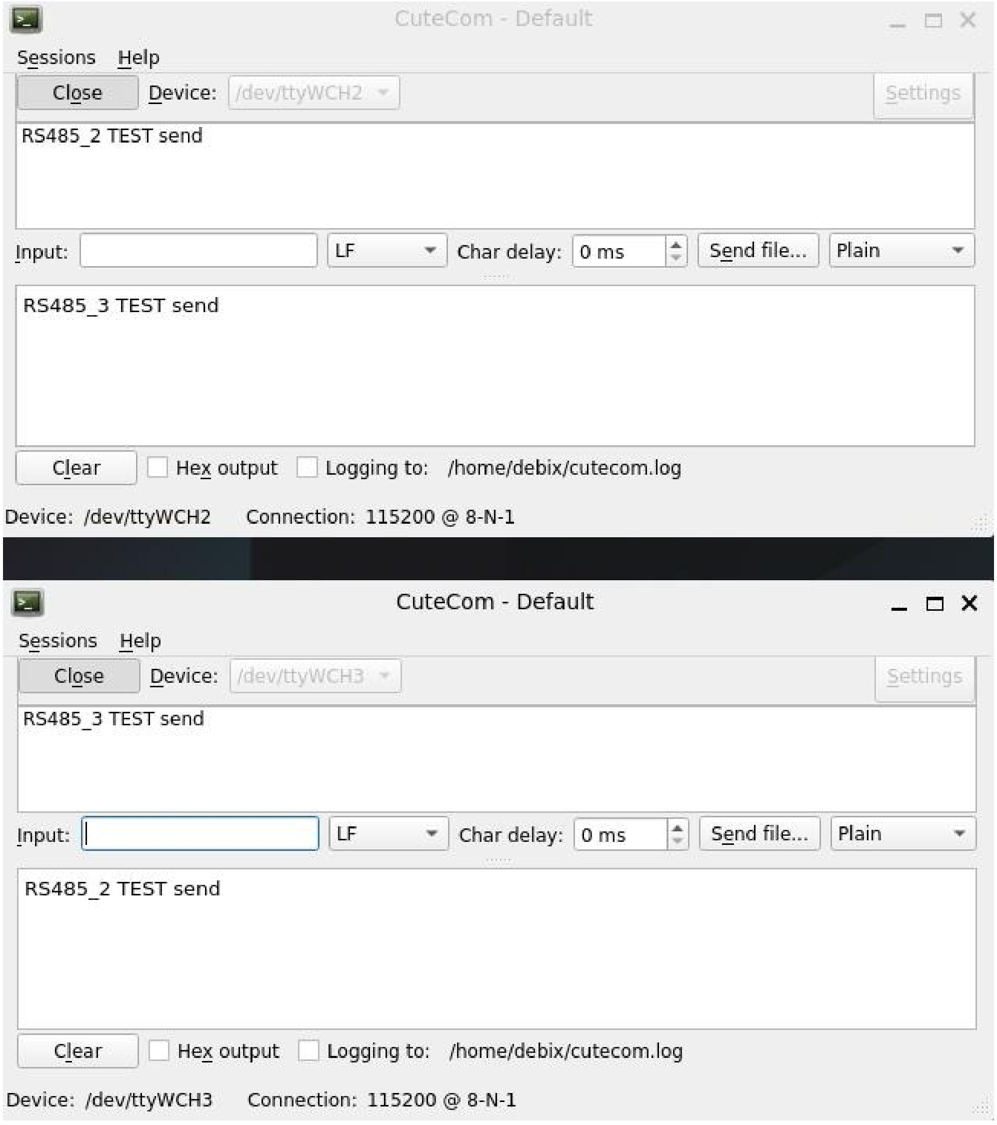
Device: /dev/ttyWCH2

□ Hex output □ Logging to: /home/debix/cutecom.log

Device: /dev/ttyWCH2 Connection: 115200 @ 8-N-1

Figure 99

4. **Send and receive data via cutecom:** Type the test string in the cutecom input box, press **Enter** to send, you can see that another cutecom receiving box receives the same message, indicating that the communication is successful, and the result is as follows:



Device: /dev/ttyWCH2

RS485_2 TEST send

Char delay: 0 ms

RS485_3 TEST send

☐ Hex output ☐ Logging to: /home/debix/cutecom.log

Device: /dev/ttyWCH2 Connection: 115200 @ 8-N-1

Device: /dev/ttyWCH3

RS485_3 TEST send

Char delay: 0 ms

RS485_2 TEST send

☐ Hex output ☐ Logging to: /home/debix/cutecom.log

Device: /dev/ttyWCH3 Connection: 115200 @ 8-N-1

Figure 100

4.5.4. CAN

1. **Hardware connection:** Connect two CAN interfaces (J23 and J24 for example), as shown in the figure below:
   - Pin1 of J23 is connected to Pin1 of J24
   - Pin2 of J23 is connected to Pin2 of J24
   - Pin3 of J23 is connected to Pin3 of J24

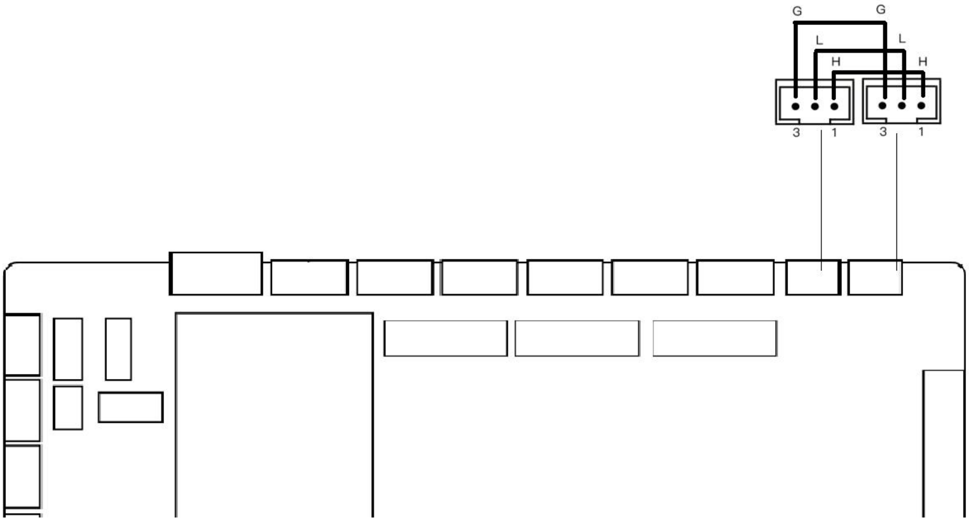
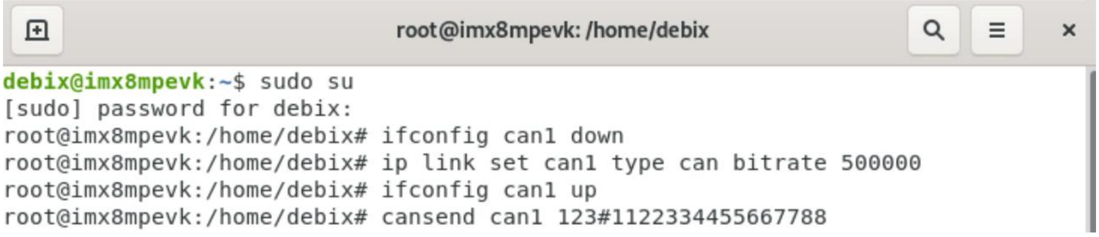


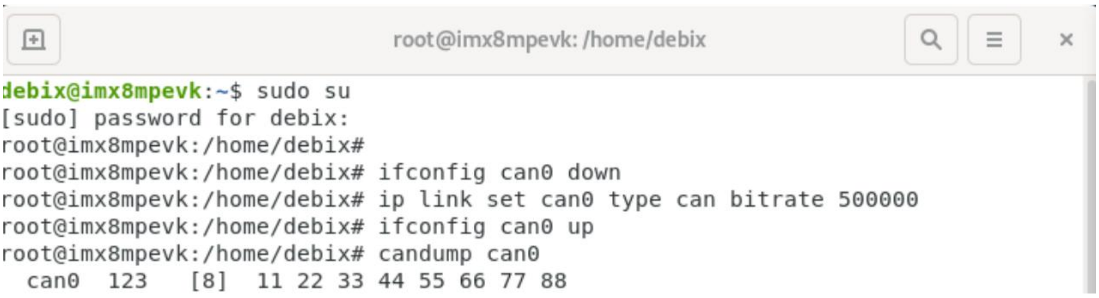
Figure 101

2. **CAN1 sends data, CAN0 receives data:** In a Terminal window, configure CAN0 to receive data and CAN1 to send data.

```
ifconfig can0 down
ip link set can0 type can bitrate 500000
ifconfig can0 up
ifconfig can1 down
ip link set can1 type can bitrate 500000
ifconfig can1 up
candump can0     #(can0 receive data in background)
cansend can1 123#1122334455667788   #(can1 send data)
```



```
debix@imx8mpevk:~$ sudo su
[sudo] password for debix:
root@imx8mpevk:/home/debix# ifconfig can1 down
root@imx8mpevk:/home/debix# ip link set can1 type can bitrate 500000
root@imx8mpevk:/home/debix# ifconfig can1 up
root@imx8mpevk:/home/debix# cansend can1 123#1122334455667788
```

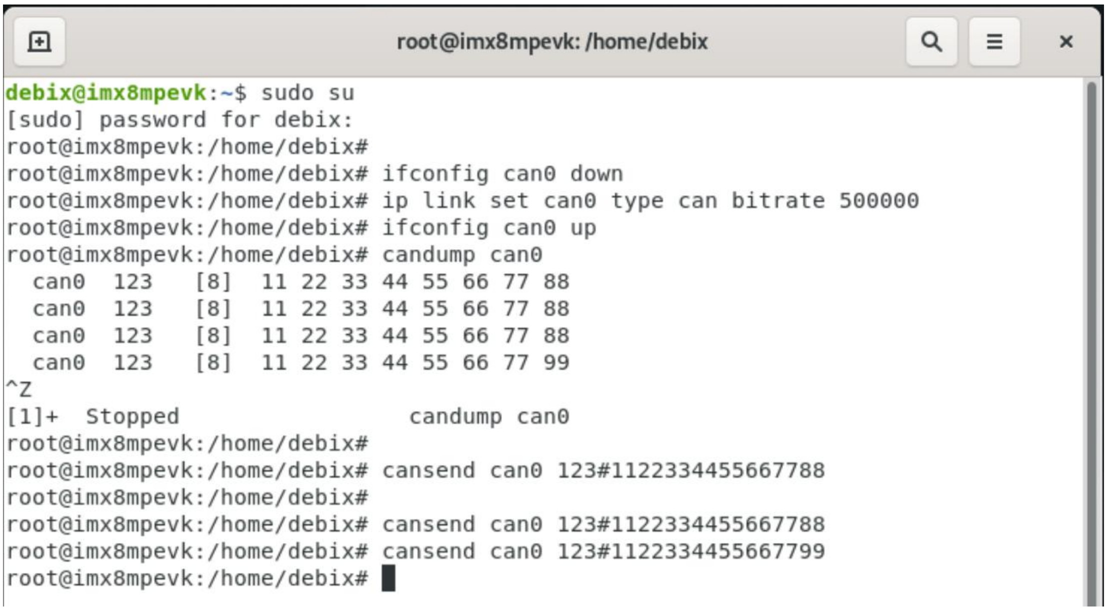


root@imx8mpevk: /home/debix

```
debix@imx8mpevk:~$ sudo su
[sudo] password for debix:
root@imx8mpevk:/home/debix#
root@imx8mpevk:/home/debix# ifconfig can0 down
root@imx8mpevk:/home/debix# ip link set can0 type can bitrate 500000
root@imx8mpevk:/home/debix# ifconfig can0 up
root@imx8mpevk:/home/debix# candump can0
  can0  123   [8]  11 22 33 44 55 66 77 88
```

3. **CAN1 receives data, CAN0 sends data:** In the Terminal window, switch CAN1 to receive state and CAN0 to send state.

```
candump can1     #(can1 receive data in background)
cansend can0 123#1122334455667788   #(can0 send data)
```



root@imx8mpevk: /home/debix

```
debix@imx8mpevk:~$ sudo su
[sudo] password for debix:
root@imx8mpevk:/home/debix#
root@imx8mpevk:/home/debix# ifconfig can0 down
root@imx8mpevk:/home/debix# ip link set can0 type can bitrate 500000
root@imx8mpevk:/home/debix# ifconfig can0 up
root@imx8mpevk:/home/debix# candump can0
  can0  123   [8]  11 22 33 44 55 66 77 88
  can0  123   [8]  11 22 33 44 55 66 77 88
  can0  123   [8]  11 22 33 44 55 66 77 88
  can0  123   [8]  11 22 33 44 55 66 77 99
^Z
[1]+  Stopped                 candump can0
root@imx8mpevk:/home/debix#
root@imx8mpevk:/home/debix# cansend can0 123#1122334455667788
root@imx8mpevk:/home/debix#
root@imx8mpevk:/home/debix# cansend can0 123#1122334455667788
root@imx8mpevk:/home/debix# cansend can0 123#1122334455667799
root@imx8mpevk:/home/debix# 
```

4.6. DO/DI

4.6.1. DO

Take DO_1 as an example, the current supported by DO_1 is 500mA.

1. **Hardware connection:** Isolated digital output connection, as shown in the following figure:

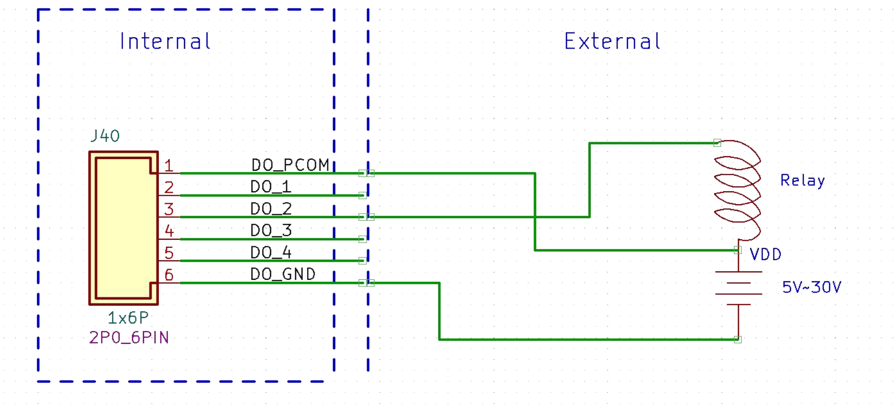
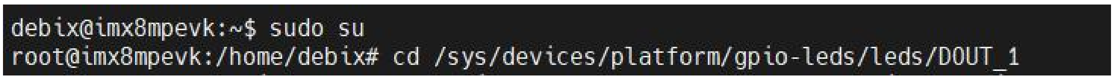


Figure 102

2. Enter the DOUT_1 control directory:

```
cd /sys/devices/platform/gpio-leds/leds/DOUT_1
```



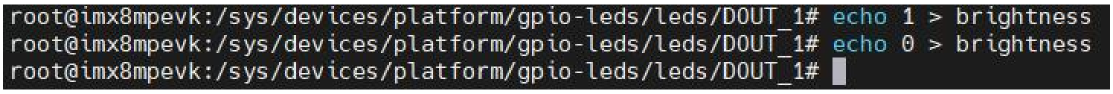
```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/devices/platform/gpio-leds/leds/DOUT_1
```

3. GPIO output is low, DO_1 open-drain output is high configuration, and the relay is turned off.

```
echo 0 > brightness
```

4. GPIO output is high, DO_1 output is low, and the relay is on.

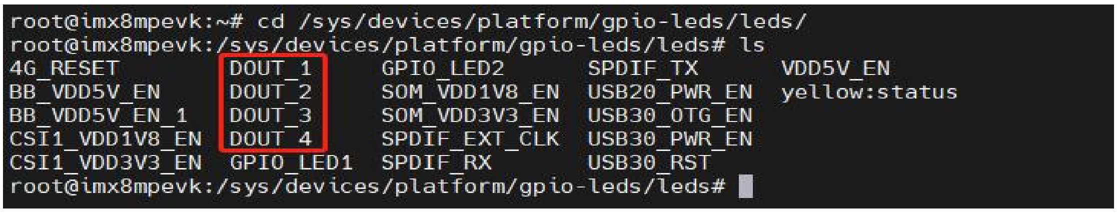
```
echo 1 > brightness
```



```
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/DOUT_1# echo 1 > brightness
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/DOUT_1# echo 0 > brightness
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds/DOUT_1# 
```

NOTE

The other DO verification methods are the same, the DOUT_1, DOUT_2, DOUT_3, and DOUT_4 files under /sys/devices/platform/gpio-leds/leds/ correspond to DO_1, DO_2, DO_3, and DO_4 ports respectively.



```
root@imx8mpevk:~# cd /sys/devices/platform/gpio-leds/leds/
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds# ls
4G_RESET        DOUT_1     GPIO_LED2      SPDIF_TX      VDD5V_EN
BB_VDD5V_EN     DOUT_2     SOM_VDD1V8_EN  USB20_PWR_EN  yellow:status
BB_VDD5V_EN_1   DOUT_3     SOM_VDD3V3_EN  USB30_OTG_EN
CSI1_VDD1V8_EN  DOUT_4     SPDIF_EXT_CLK  USB30_PWR_EN
CSI1_VDD3V3_EN  GPIO_LED1  SPDIF_RX       USB30_RST
root@imx8mpevk:/sys/devices/platform/gpio-leds/leds# 
```

4.6.2. DI

Take the dry node connection as an example, all ports of DI operate in the same way, here take DI_1 as an example, and connect the DI_1 to ground in series.

Calculation of GPIO number: GPIO number = GPIOn_IOx = (n-1)*32 + x

Table 45 Calculation of GPIO number

GPIO Number	DI Definition	Pin	IO Port
5	DI_1	2	GPIO1_IO5
6	DI_2	3	GPIO1_IO6
7	DI_3	4	GPIO1_IO7
8	DI_4	5	GPIO1_IO8

1. **Hardware connection:** Isolated digital input connection, as shown in the following figure:

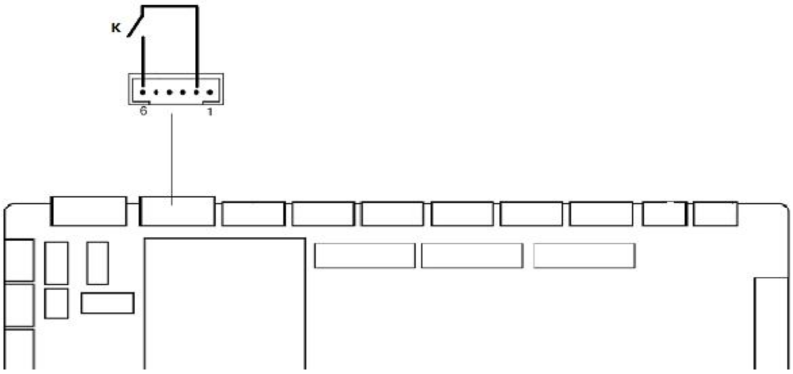
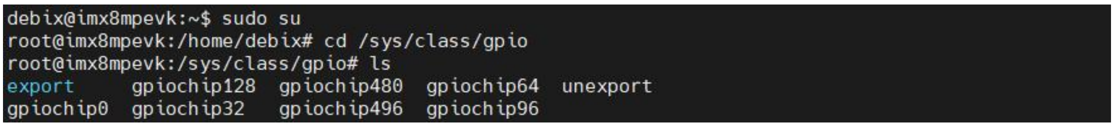


Figure 103

2. Enter the GPIO directory:

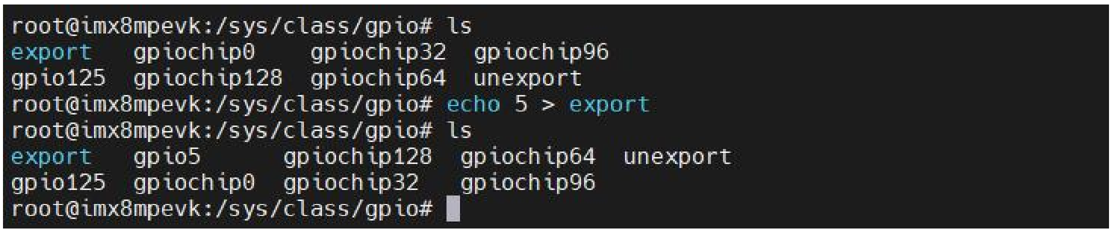
```
cd /sys/class/gpio
```



```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/class/gpio
root@imx8mpevk:/sys/class/gpio# ls
export     gpiochip128  gpiochip480  gpiochip64  unexport
gpiochip0  gpiochip32   gpiochip496  gpiochip96
```

3. Export GPIO, after successful export, it will automatically generate gpio5 directory, where 5 is calculated from GPIO number. As shown in the following figure:

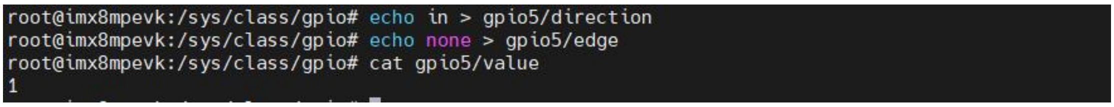
```
echo 5 > export
```



```
root@imx8mpevk:/sys/class/gpio# ls
export   gpiochip0    gpiochip32  gpiochip96
gpio125  gpiochip128  gpiochip64  unexport
root@imx8mpevk:/sys/class/gpio# echo 5 > export
root@imx8mpevk:/sys/class/gpio# ls
export   gpio5      gpiochip128  gpiochip64  unexport
gpio125  gpiochip0  gpiochip32   gpiochip96
root@imx8mpevk:/sys/class/gpio# 
```

4. Configure GPIO pins:

```
echo in > gpio5/direction          # Set pin direction to input
echo none > gpio5/edge             # Set non-interrupt pin
cat gpio5/value                    # Check DI_1 level, the default is high
```



```
root@imx8mpevk:/sys/class/gpio# echo in > gpio5/direction
root@imx8mpevk:/sys/class/gpio# echo none > gpio5/edge
root@imx8mpevk:/sys/class/gpio# cat gpio5/value
1
```

4.7. Mini PCIe

Mini PCIe interface can be connected to different modules to achieve rich functionality expansion. Here, we will introduce the integration of 4G module and SATA module as examples.

Table 46 Mini PCIe node description

Function Name	Interface	Access Module Type	Device Node
Mini PCIe	J21	4G Module	/dev/ttyUSB2
		SATA Module	/dev/sd* (* is automatically generated by the system when the device is inserted, and modified according to the actual situation)

The connection of Mini PCIe to 4G module and SATA module are shown in the figure below:

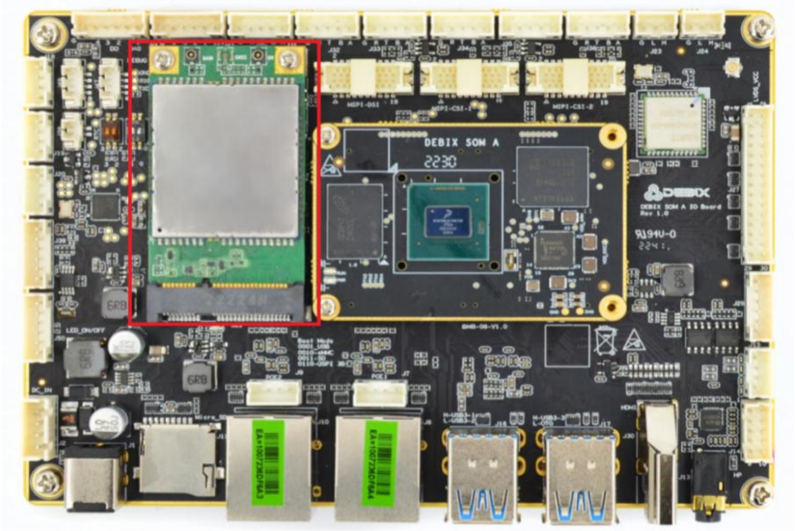


Figure 104 Connect 4G module to Mini PCIe

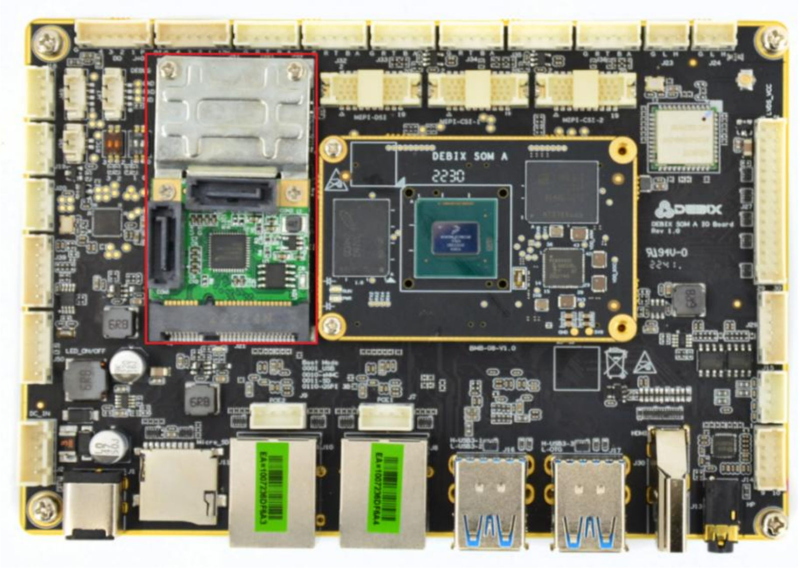
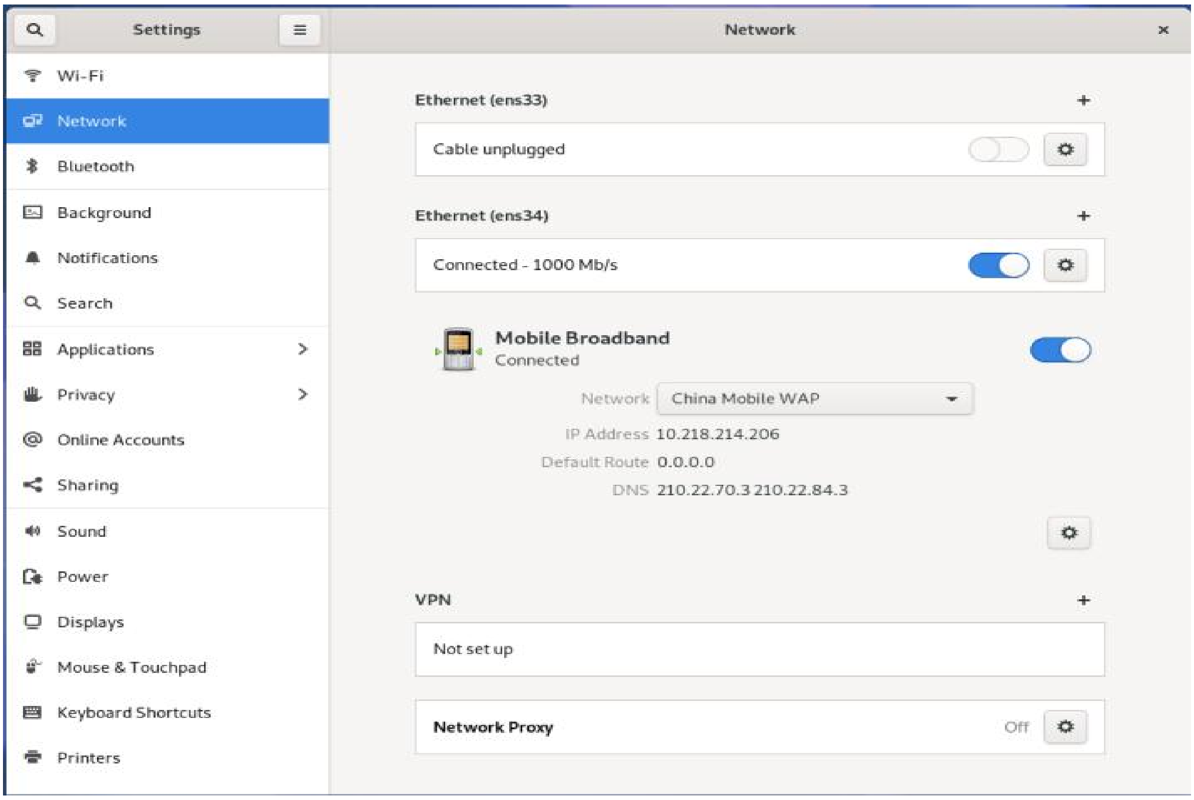


Figure 105 Connect SATA module to Mini PCIe

- Mini PCIe 4G module
  - Connect the 4G module to the Mini PCIe interface, insert a Micro SIM card, power on the device, boot into the system desktop, open the **Settings** app, select **Network** to set up the network, and after the **IP Address** is displayed, ping Ethernet to test the connectivity. For specific verification details, please refer to the Function Examples of DEBIX 4G Board User Manual.



Ethernet (ens33)

Cable unplugged

Ethernet (ens34)

Connected - 1000 Mb/s

Mobile Broadband
Connected

Network China Mobile WAP

IP Address 10.218.214.206

Default Route 0.0.0.0

DNS 210.22.70.3 210.22.84.3

VPN

Not set up

Network Proxy Off

Figure 106

- Mini PCIe SATA module
  - The verification method is the same as USB 2.0: First connect the SATA module to the Mini PCIe interface, and then power on the device.

4.8. I2C

I2C4 mounts RTC, EEPROM, and I2C external expansion interface (J39) on DEBIX SOM A I/O Board. The J39 interface can be used for TP, etc.

Table 47 Description of I2C mount module

Module	Model	Address
RTC	HYM8563S	0x50
EEPROM	BL24C02F	0x51

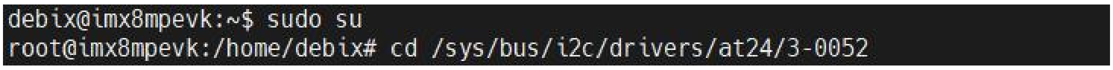
Take EEPROM as an example to verify I2C:

1. Enter the system desktop, open a terminal and switch to the root user;

```
sudo su
```

2. Enter the control directory:

```
cd /sys/bus/i2c/drivers/at24/3-0052
```



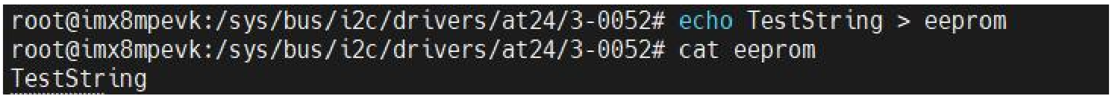
```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/bus/i2c/drivers/at24/3-0052
```

3. Write data: Create a eeprom document for write verification.

```
echo TestString > eeprom
```

4. Read data.

```
cat eeprom
```



```
root@imx8mpevk:/sys/bus/i2c/drivers/at24/3-0052# echo TestString > eeprom
root@imx8mpevk:/sys/bus/i2c/drivers/at24/3-0052# cat eeprom
TestString
```

4.9. LVDS BL CTRL

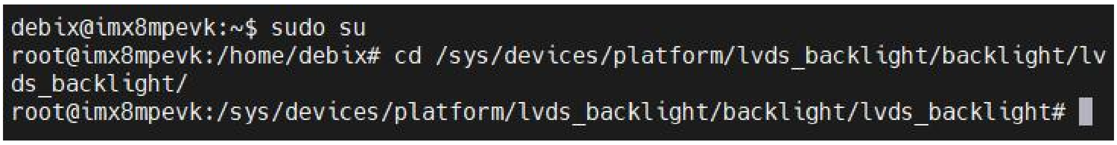
DEBIX SOM A IO Board defaults to HDMI output. To use LVDS output, you need to open the Add on Board APP in the system desktop, select a suitable LVDS screen, and then restart the system.

1. Enter the system desktop, open a terminal and switch to the root user

```
sudo su
```

2. Enter the LVDS backlight control directory:

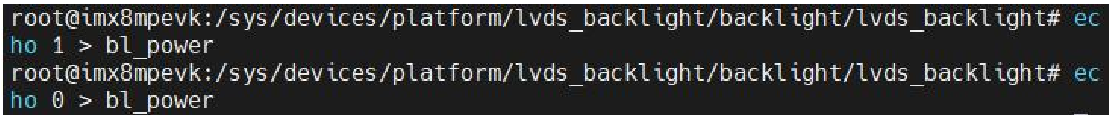
```
cd /sys/devices/platform/lvds_backlight/backlight/lvds_backlight
```



```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/devices/platform/lvds_backlight/backlight/lv
ds_backlight/
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# 
```

3. Backlight switch (on by default)

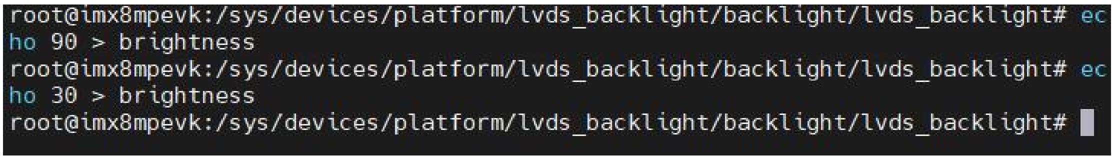
```
echo 1 > bl_power            # Turn off backlight

echo 0 > bl_power            # Turn on backlight
```



```
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 1 > bl_power
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 0 > bl_power
```

4. **Backlight brightness adjustment:** The principle is to adjust the brightness by changing the PWM duty cycle, the adjustable range (that is, the PWM duty cycle) is 0-100, the default value: 80.

```
echo 90 > brightness          # PWM duty cycle adjusted to 90%

echo 30 > brightness          # PWM duty cycle adjusted to 30%
```



```
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 90 > brightness
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 30 > brightness
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# █
```

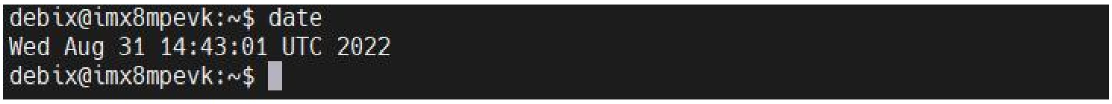
NOTE

> Parameters 30 and 90 are the PWM duty cycle, and the parameters can be adjusted according to the actual situation.

4.10. RTC

1. Query the current system time:

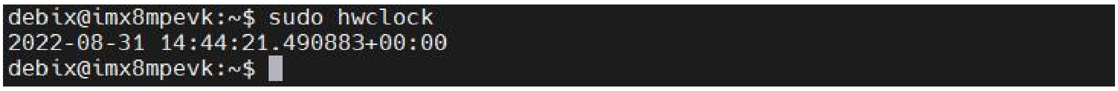
```
date
```



```
debix@imx8mpevk:~$ date
Wed Aug 31 14:43:01 UTC 2022
debix@imx8mpevk:~$ █
```

2. Query the current RTC time:

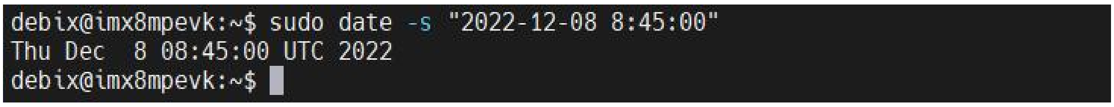
```
sudo hwclock
```



```
debix@imx8mpevk:~$ sudo hwclock
2022-08-31 14:44:21.490883+00:00
debix@imx8mpevk:~$ █
```

3. Modify the current system time:

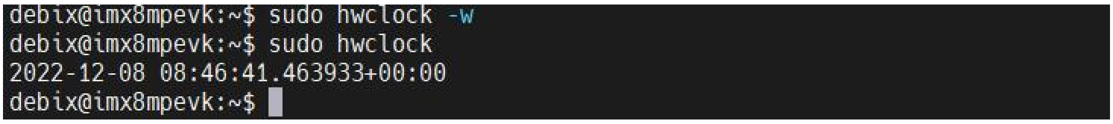
```
sudo date -s “2022-12-08 8:45:00”
```



```
debix@imx8mpevk:~$ sudo date -s "2022-12-08 8:45:00"
Thu Dec  8 08:45:00 UTC 2022
debix@imx8mpevk:~$ 
```

4. Write the system time to RTC:

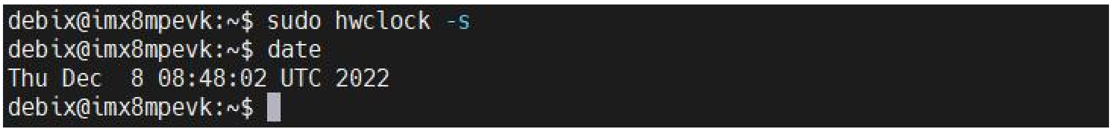
```
sudo hwclock -w

sudo hwclock                    #Check the RTC time
```



```
debix@imx8mpevk:~$ sudo hwclock -w
debix@imx8mpevk:~$ sudo hwclock
2022-12-08 08:46:41.463933+00:00
debix@imx8mpevk:~$ 
```

5. Write the RTC time to system:

```
sudo hwclock -s

date                            #Check the system time
```



```
debix@imx8mpevk:~$ sudo hwclock -s
debix@imx8mpevk:~$ date
Thu Dec  8 08:48:02 UTC 2022
debix@imx8mpevk:~$ 
```

4.11. Heat Dissipation

When DEBIX SOM A + IO Board runs for a prolonged period of time, it will result in an increase in its CPU temperature. Therefore, implementations should be considered to cool the CPU and the entire device passively. If the CPU needs to be cooled, it is recommended to use CPU aluminum alloy heatsink: paste aluminum alloy heatsink directly above the CPU for heat dissipation, as shown below:

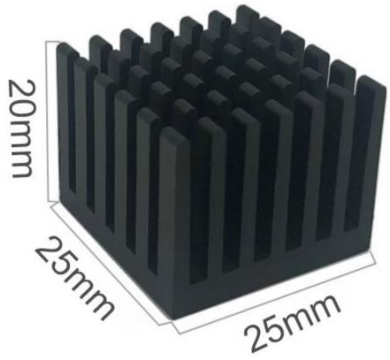


Figure 107 Aluminum alloy heatsink

4.12. Usage of Display Screen

The three screens supported by DEBIX SOM A + IO Board are as follows:

Table 48 Display Screen supported by DEBIX SOM A + IO Board

No	Screen Type	Specification Address
1	DEBIX TD050A 800x480 5-inch LVDS screen	https://debix.io/hardware/5inch-tft-lcd-monitor.html
2	DEBIX TD070A 1024x600 7-inch LVDS screen	https://debix.io/hardware/7inch-tft-lcd-monitor.html
3	DEBIX TD101A 1280x800 10.1-inch LVDS screen	https://debix.io/hardware/10inch-tft-lcd-monitor.html
4	DEBIX TD080B 8-inch MIPI DSI display	For further information, please contact DEBIX representatives (Email Address: info@debix.io). *(Note: The DEBIX SOM A IO Board must use the MIPI DSI adapter board when connecting to the DEBIX MIPI DSI display.)*

The following steps use the DEBIX TD101A as an example.

Step 1: Switching The Device Tree

- **Method 1: (switching the device tree through the debug serial port)**

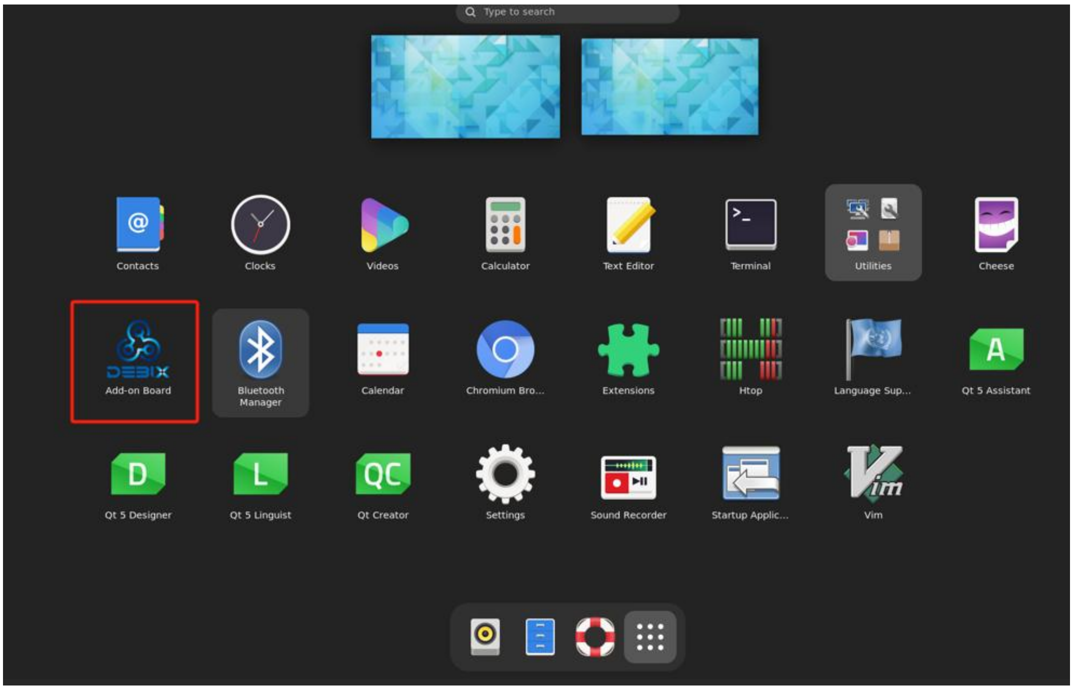
For the usage of the debug serial port, please refer to the section 4.5.1.UART.

After entering the serial console, run the following commands:

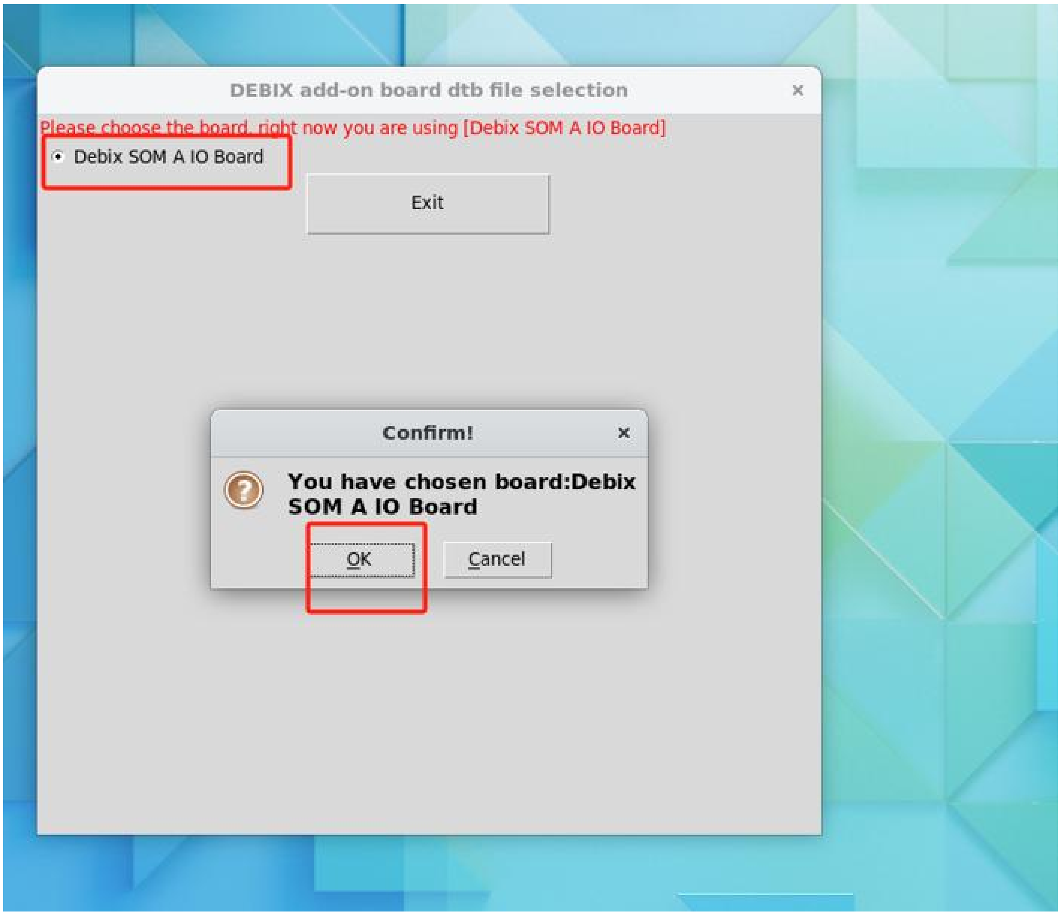
```
#sudo su

#cd /boot

#cp imx8mp-debix-core-TD101A.dtb imx8mp-evk.dtb

#reboot
```

- **Method 2:**

1) Connect the DEBIX to an HDMI display screen, a keyboard, a mouse, and power it on. Then enter the DEBIX system desktop and select the **Add-on Board** app;



2) In the pop-up "DEBIX add on board dtb file selection" window, select **Debix board** and click **OK;**

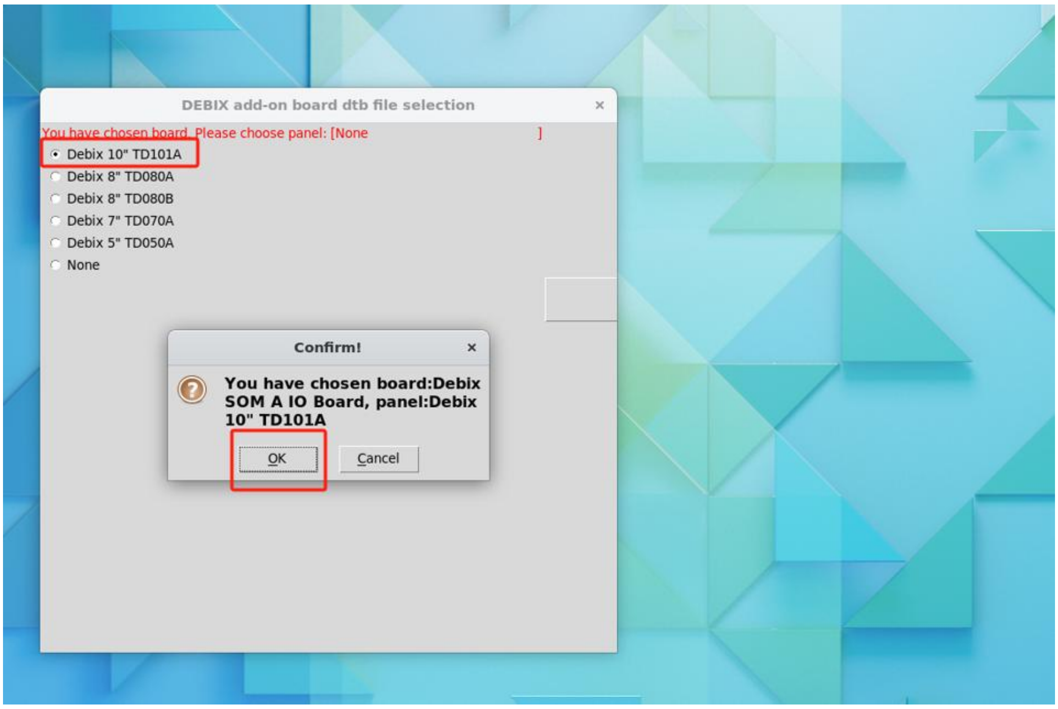


Please choose the board, right now you are using [Debix SOM A IO Board]

● Debix SOM A IO Board

You have chosen board:Debix SOM A IO Board

3) Check “Debix 10” TD101A”, click **OK;**

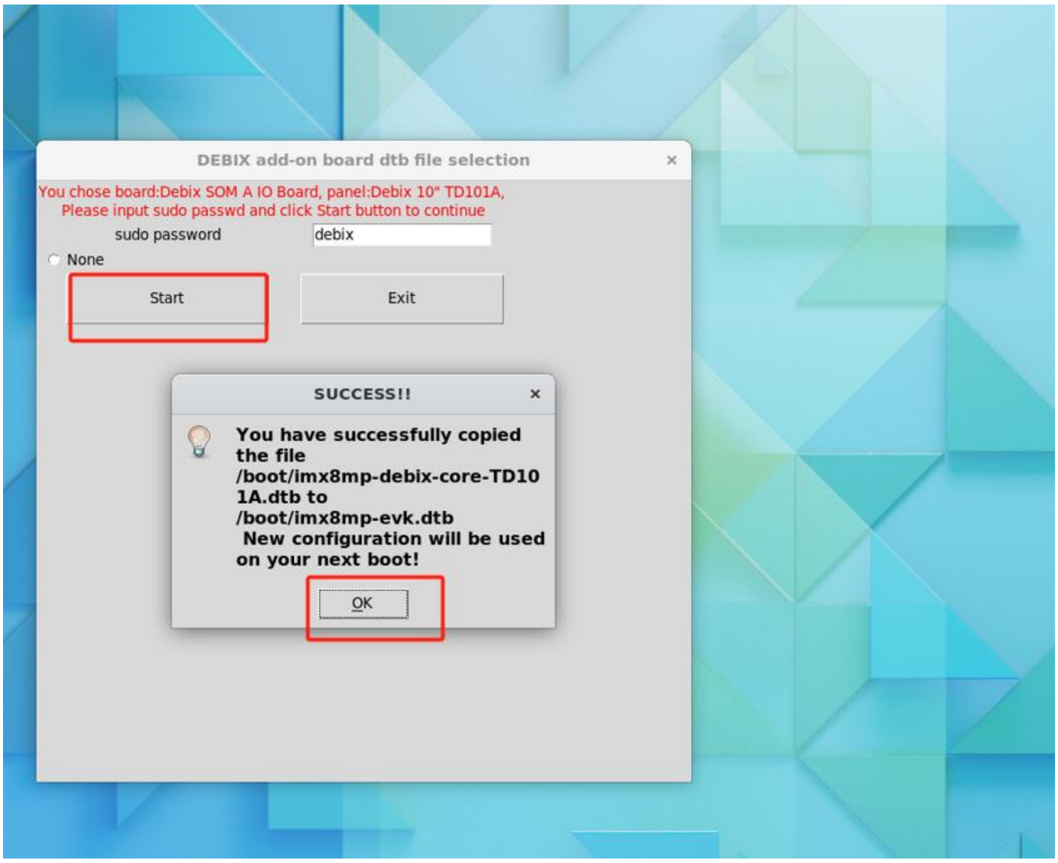


You have chosen board, Please choose panel: [None]

● Debix 10" TD101A
○ Debix 8" TD080A
○ Debix 8" TD080B
○ Debix 7" TD070A
○ Debix 5" TD050A
○ None

You have chosen board:Debix SOM A IO Board, panel:Debix 10" TD101A

4) Enter the sudo password (**debix**) and click **Start**, then click **OK;**



DEBIX add-on board dtb file selection

You chose board:Debix SOM A IO Board, panel:Debix 10" TD101A,
Please input sudo passwd and click Start button to continue

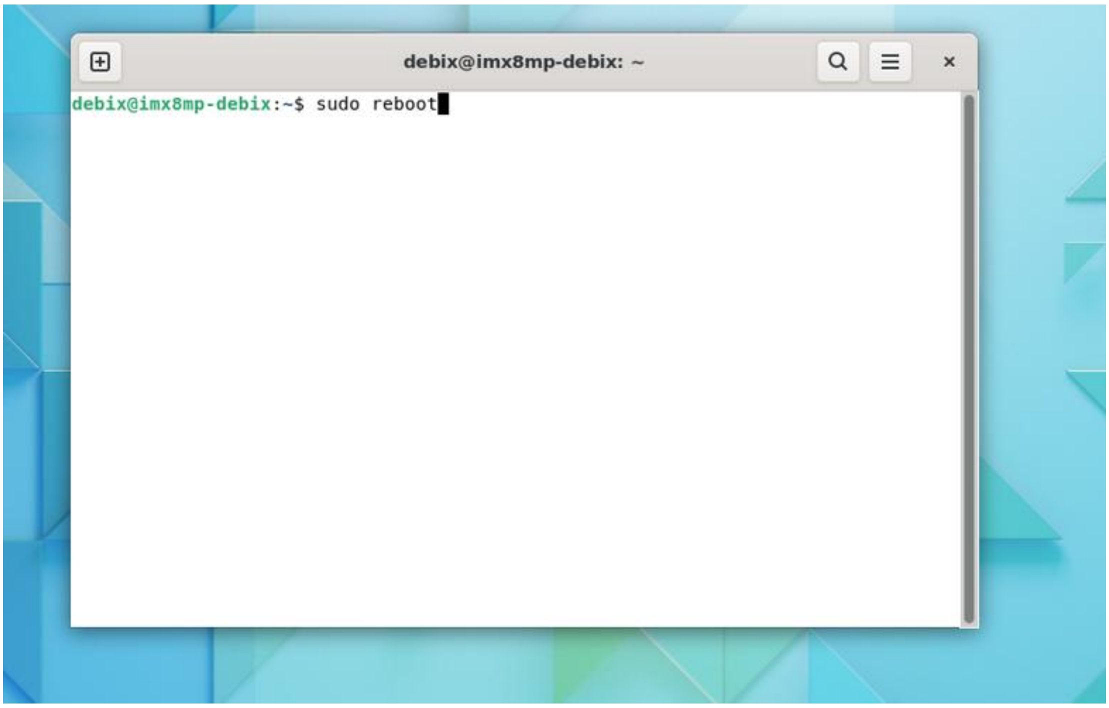
sudo password: debix

○ None

SUCCESS!!

You have successfully copied the file /boot/imx8mp-debix-core-TD101A.dtb to /boot/imx8mp-evk.dtb
New configuration will be used on your next boot!

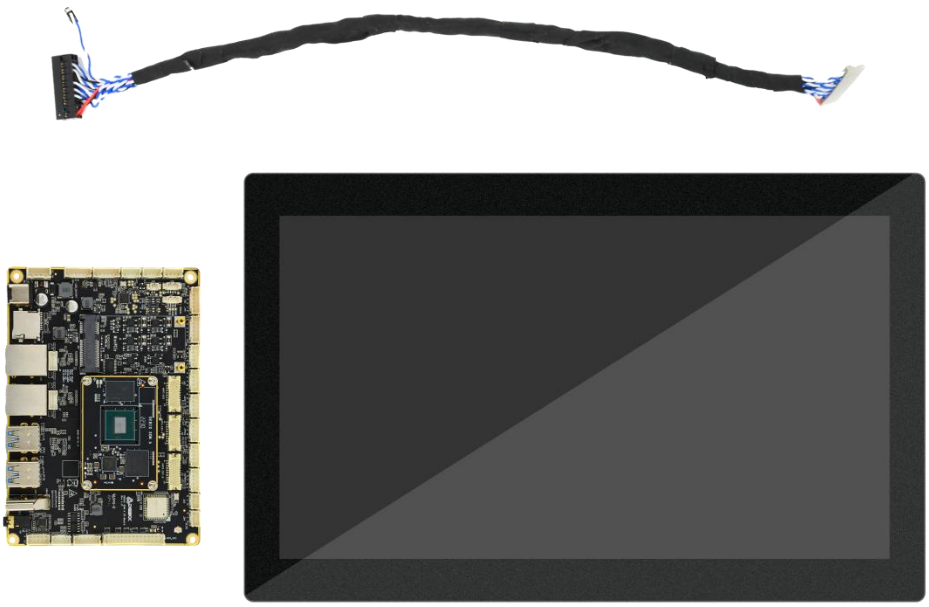
5) Reboot the device, and then disconnect the power supply.



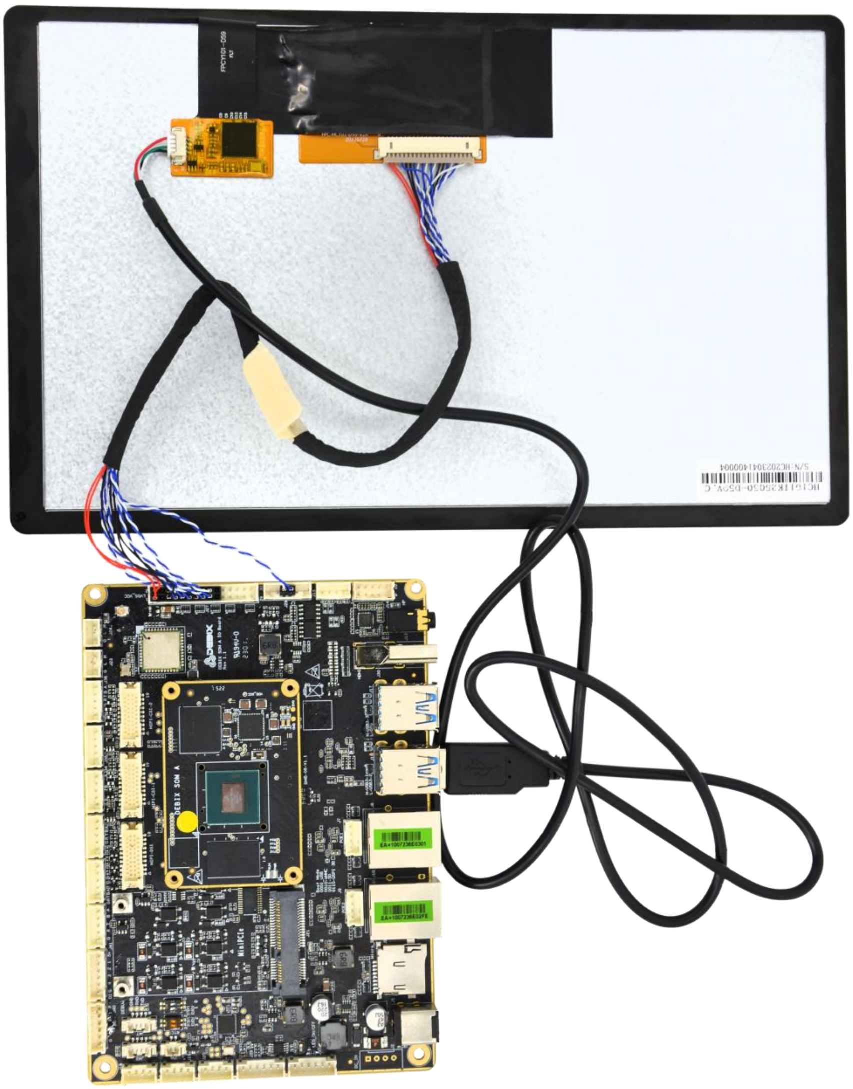
```
debix@imx8mp-debix:~$ sudo reboot
```

Step 2: Connecting the LVDS Screen

1) **Component Preparation:** LVDS screen, DEBIX SOM A + IO Board, LVDS screen cable, as shown in the figure below:



2) Plug the double-row female header of LVDS screen cable to LVDS interface (J27) of DEBIX SOM A IO Board, the red line should be connected to Pin1, Pin2; as for the sole 2Pin blue and white line, the blue line is connected to Pin4 of LVDS backlight control adjustment interface (J29), the white line is connected to Pin3 of LVDS backlight control adjustment interface (J29).



3) Connect the device to the power supply, the LVDS screen displays the following figure: