DEBIX SOM A + I/O Board Applications Document

Version：V1.0（2022-12）

Edit: **Polyhex Technology Company Limited**（http://www.polyhex.net/）

The DEBIX SOM A I/O board is a carrier board specially designed for the DEBIX SOM A and connects to the DEBIX SOM A via 4 double board-to-board socket connectors on the front. The i.MX 8M Plus-based core board supports full-featured interfaces, providing a perfect solution combining the functionality required for industrial control, IoT connectivity and multimedia.

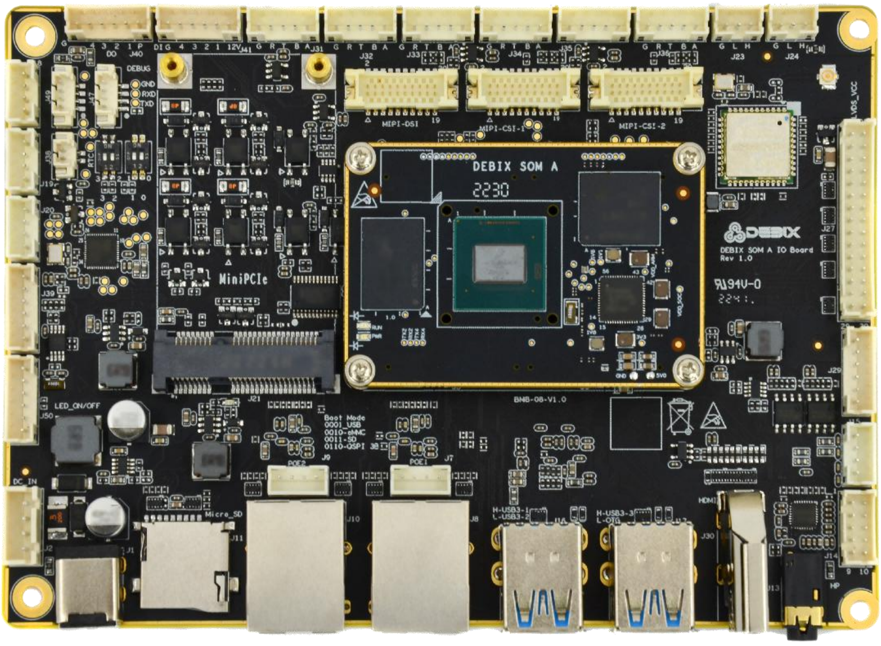


Image 1

Table of Contents

DEBIX SOM A IO Board Applications.. 3

1.1Debug serial port..3

 1.1.1 Hardware Connection...3

 1.1.2 Applications... 4

1.2 RS485/RS232... 7

 1.2.1 Hardware Connection..8

 1.2.2 Send and Receive Verification...8

1.3 CAN...12

 1.3.1 Hardware Connection..13

 1.3.2 Send and Receive Verification...13

1.4 Nor-Flash..16

 1.4.1 Authentication method...16

1.5 RTC...18

 1.5.1 Authentication Method...18

1.6 USB 2.0.. 20

 1.6.1 Hardware Connection..20

 1.6.2 Testing method.. 21

1.7 Ethernet..23

 1.7.1 Hardware Connection..23

 1.7.2 Applications... 23

1.8 Led&key..26

 1.8.1 Hardware Connection..26

 1.8.2 Led and Key Verification.. 27

1.9 DO... 29

 1.9.1 Hardware Connection..29

 1.9.2 Applications... 30

1.10 DI..31

 1.10.1 Hardware Connection... 32

 1.10.2 Applications..32

1.11 LVDS BL CTRL... 34

 1.11.1 Applications.. 34

1.12 Mini PCIe.. 36

 1.12.1 Hardware Connection... 36

 1.12.2 Application..37

1.13 I2C..40

 1.13.1 Authentication method.. 40

DEBIX SOM A IO Board Applications

1.1Debug serial port

On DEBIX SOM A I/O Board, all system debugging information is outputted through the debugging serial port (serial port 2).

The specific parameters are as follows:

Function name	Interface	Pin	Definition	Device node
Debug serial port	J47	1	VDD_3V3	/dev/ttymxc1 default baud rate 115200
		2	DEBUG_TXD	
		3	DEBUG_RXD	
		4	GND	

1.1.1 Hardware Connection

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

- 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 terminal of the USB-TTL module
- Connect the GND of the debugging serial port to the GND terminal of the USB-TTL module

As image below:

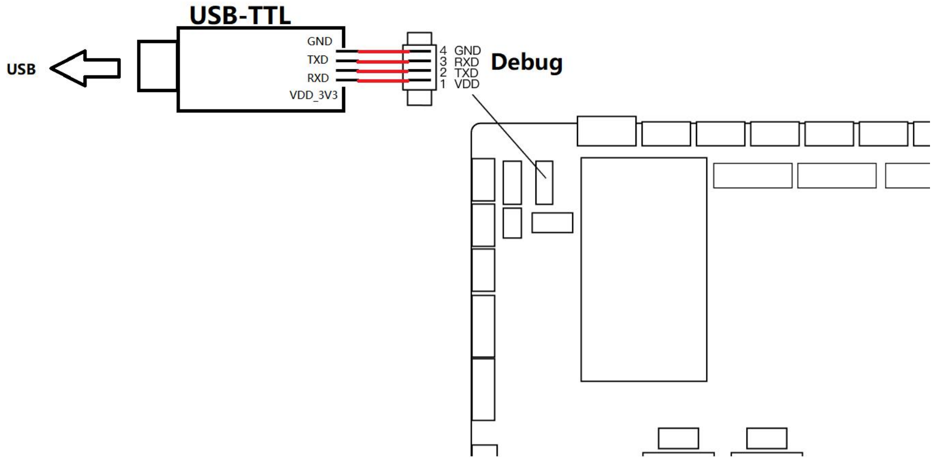


Image 2

1.1.2 Applications

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

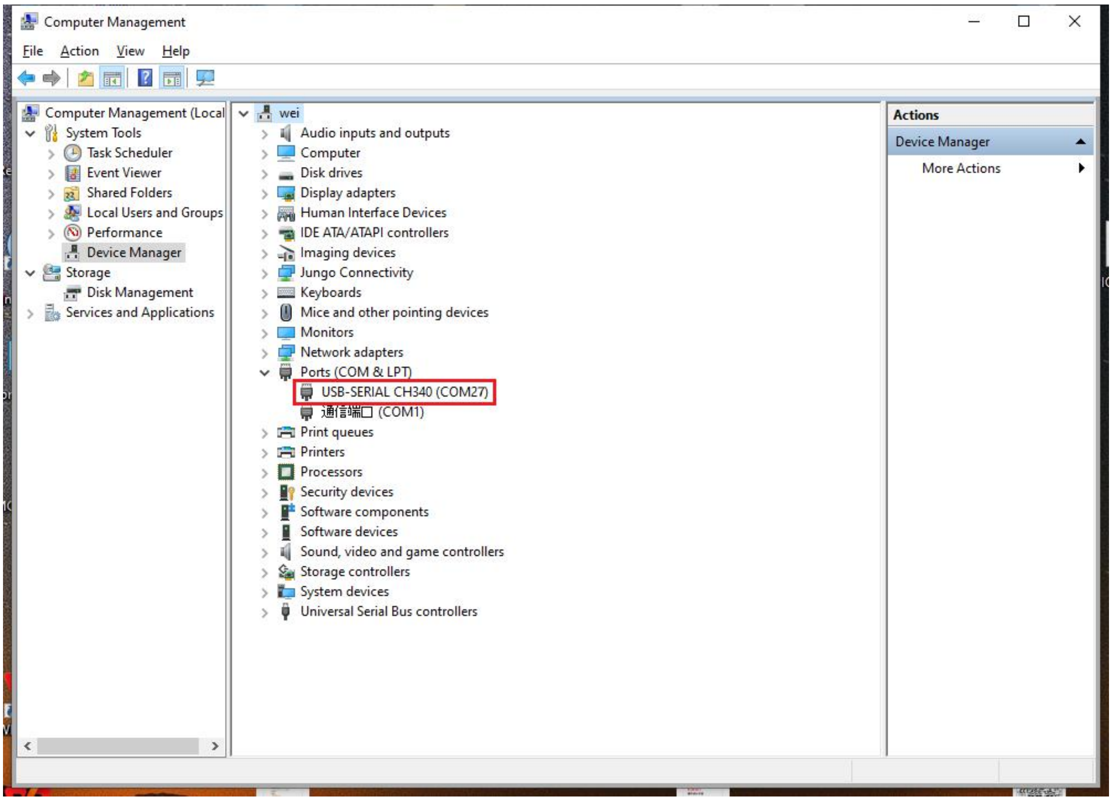


Image 3

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

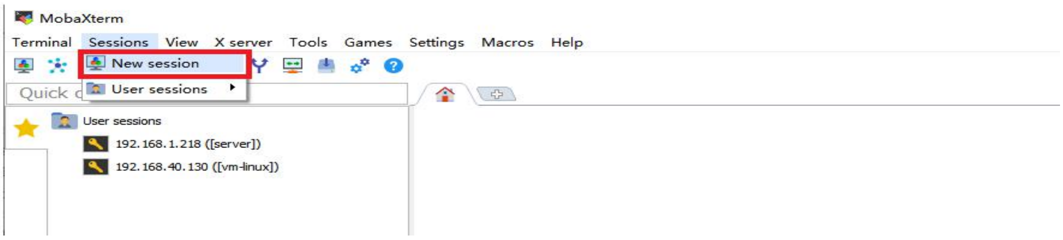

MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
New session
User sessions
Quick c
User sessions
192.168.1.218 ([server])
192.168.40.130 ([vm-linux])

Image 4

3. Select Serial in the pop-up dialog box

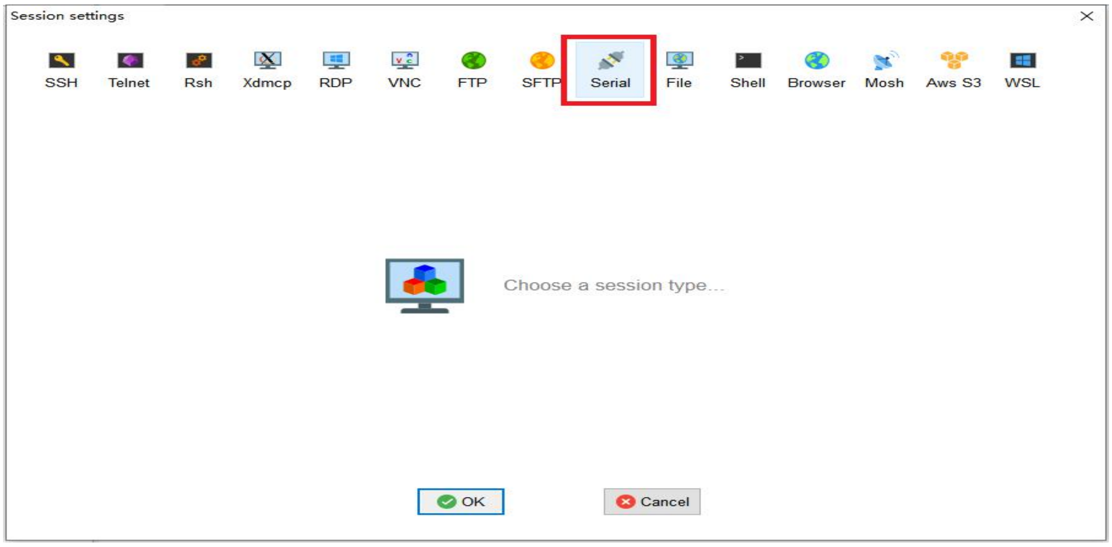
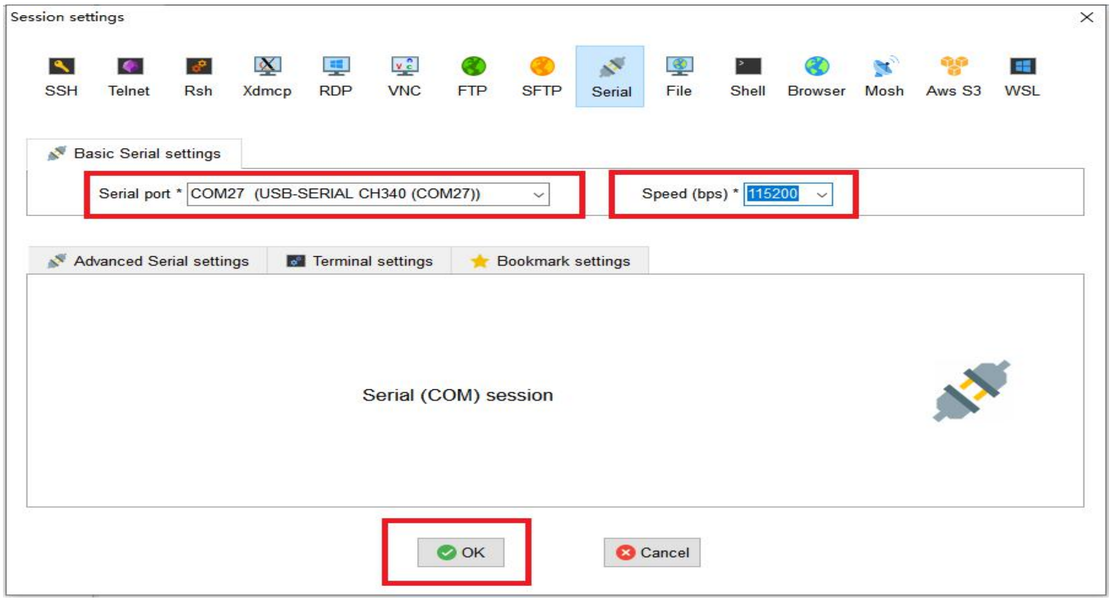


Image 5

4. Change the port number to the COM port found in the device manager, set the baud rate to 115200, and click OK



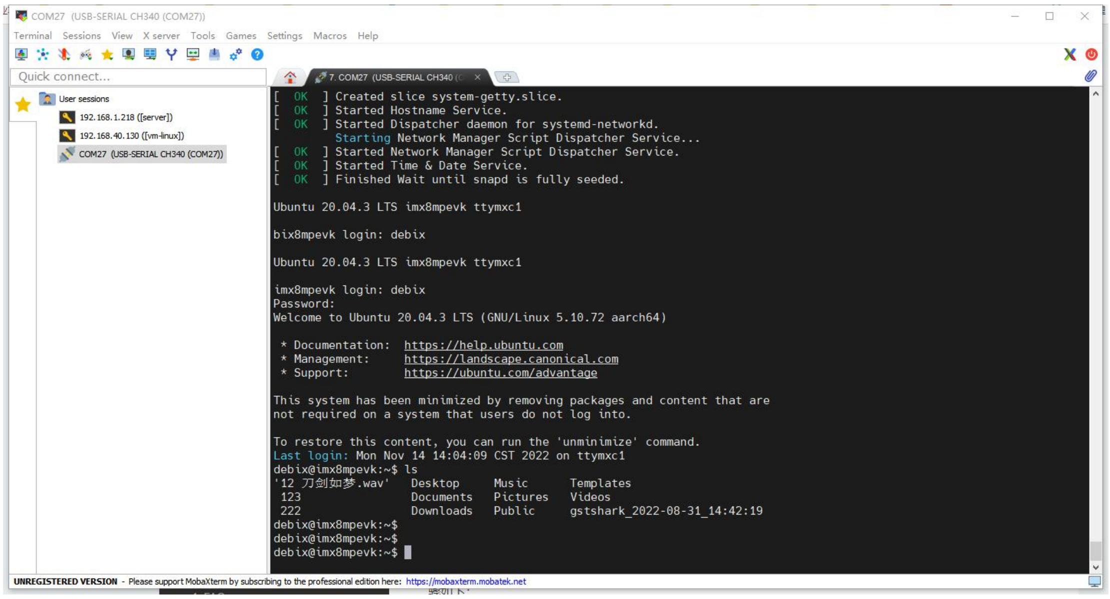
Serial port * COM27 (USB-SERIAL CH340 (COM27))

Speed (bps) * 115200

Image 6

During the boot process, you can see the logs output by Uboot, Kernel, and System on the terminal. After the system startup is complete, enter the default user and password on the terminal to enter the serial console;

- Default user: debix
- Default password: debix


COM27 (USB-SERIAL CH340 (COM27))
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
User sessions
192.168.1.218 ([server])
192.168.40.130 ([vm-linux])
COM27 (USB-SERIAL CH340 (COM27))
7. COM27 (USB-SERIAL CH340 (C
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
[  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:~$
```

Image 7

1.2 RS485/RS232

Note: With RS232 and RS485 on the same connector, only one can used at one time

DEBIX SOM A I/O Board has 6 RS485/RS232 ports on board.

The specific parameters are as follows:

Function name	Port	Pin	Definition	Device node
RS485/RS232	J31	1	RS485_AH0	/dev/ttyWCH0
		2	RS485_BL0	
		3	RS232_TXD0	
		4	RS232_RXD0	
		5	RS_GND0	
	J32	1	RS485_AH1	/dev/ttyWCH1
		2	RS485_BL1	
		3	RS232_TXD1	
		4	RS232_RXD1	
		5	RS_GND1	
	J33	1	RS485_AH2	/dev/ttyWCH2
		2	RS485_BL2	
		3	RS232_TXD2	
		4	RS232_RXD2	
		5	RS_GND2	
	J34	1	RS485_AH3	/dev/ttyWCH3
		2	RS485_BL3	
		3	RS232_TXD3	
		4	RS232_RXD3	
		5	RS_GND3	
	J35	1	RS485_AH4	/dev/ttymxc2
		2	RS485_BL4	
		3	RS232_TXD4	
		4	RS232_RXD4	
		5	RS_GND4	
	J36	1	RS485_AH5	/dev/ttymxc3
		2	RS485_BL5	
		3	RS232_TXD5	
		4	RS232_RXD5	
		5	RS_GND5	

1.2.1 Hardware Connection

RS232 Connection

- Pin3 of J31 is connected to the receiving end of USB-RS232
- Pin4 of J31 is connected to the sending end of USB-RS232
- Connect Pin5 of J31 to the ground terminal of USB-RS232
- USB-RS232 is connected to the onboard USB 3.0 interface.

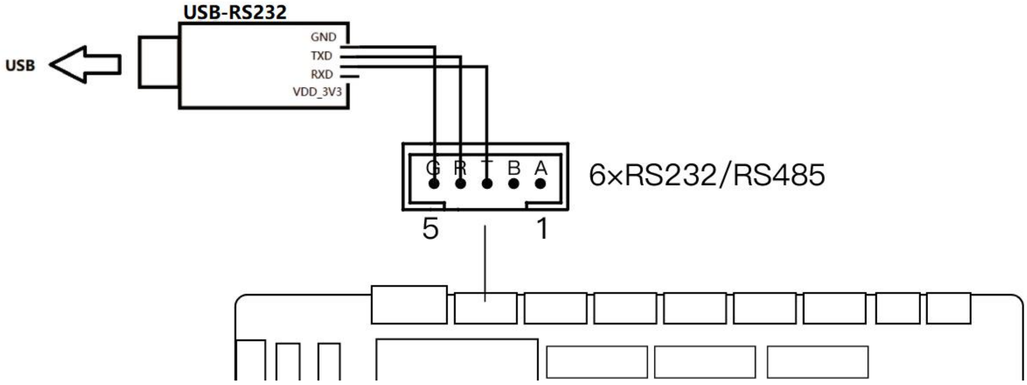


Image 8

RS485 Connection:

- Connect Pin1 of J33 to Pin1 of J34 (that is, A to A)
- Connect Pin2 of J33 to Pin2 of J34 (that is, B to B)

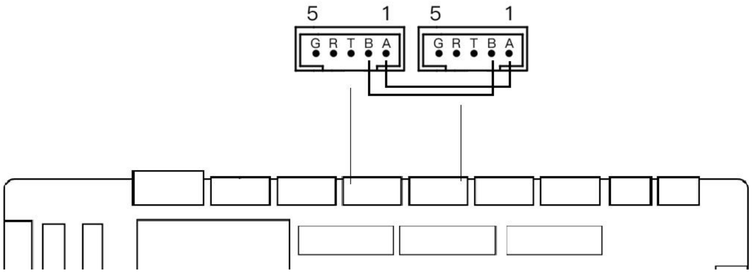


Image 9

1.2.2 Send and Receive Verification

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.

RS232 send and receive verification

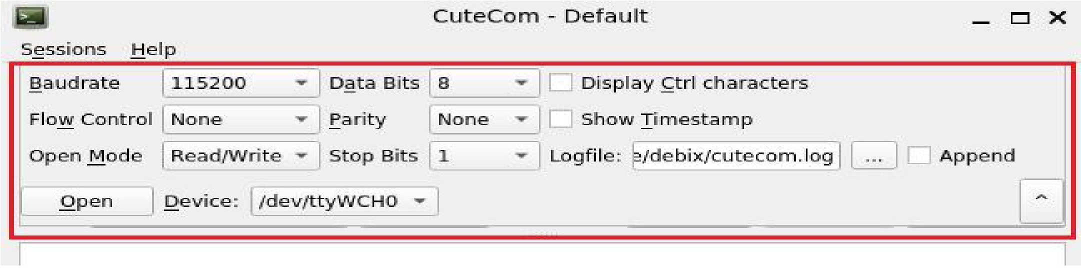
(1) Install cutecom serial port tool on DEBIX SOM A:

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

(2) The serial port parameters are set as follows:

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

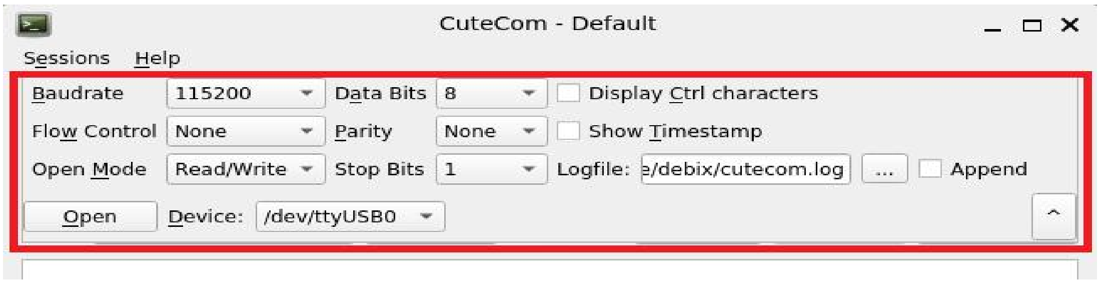
A. Run the command cutecom set the Device to /dev/ttyWCH0, set other parameters as shown in the table, click Open



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

Image 10

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

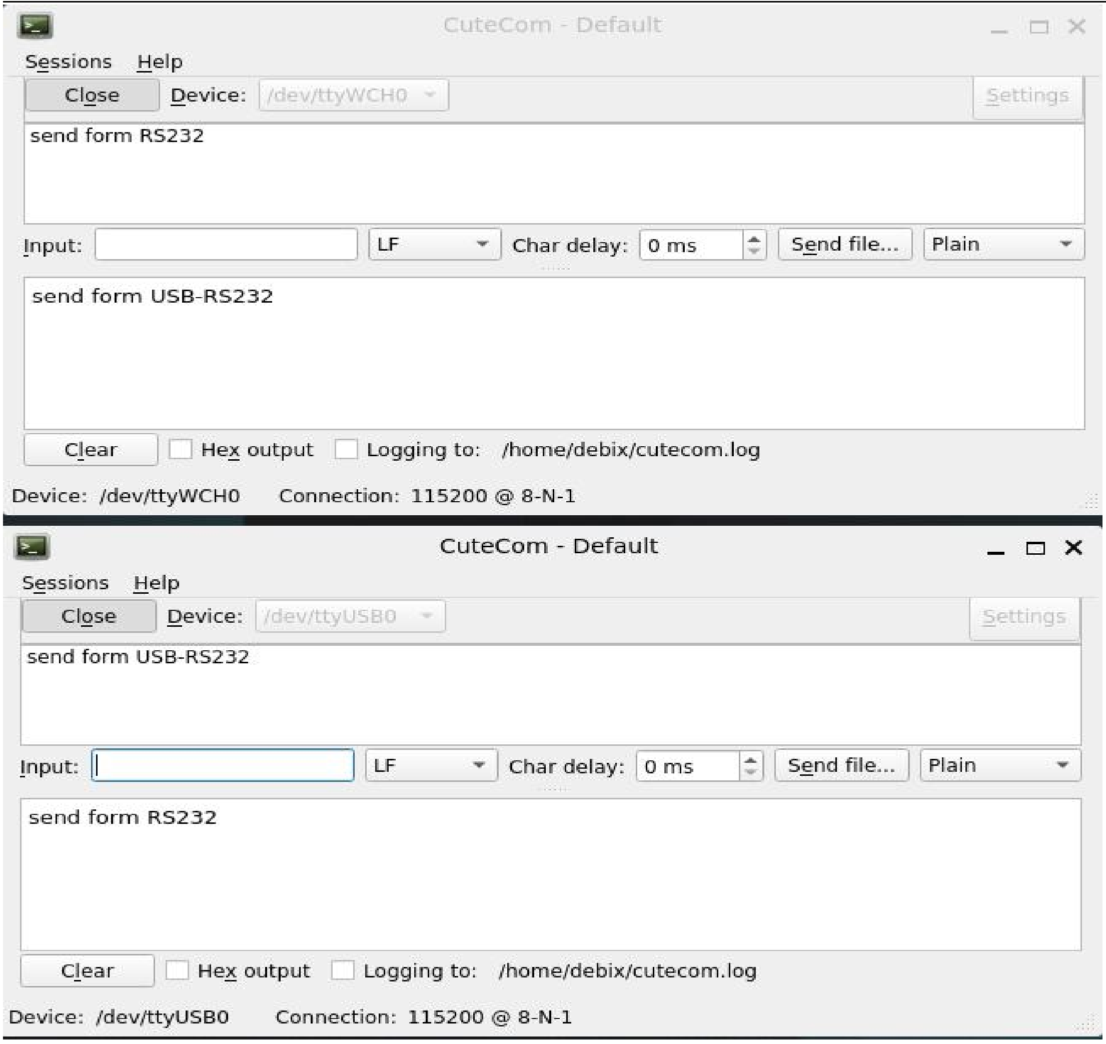


Baudrate 115200 | Data Bits 8 | □ Display Ctrl characters
Flow Control None | Parity None | □ Show Timestamp
Open Mode Read/Write | Stop Bits 1 | Logfile: e/debix/cutecom.log | □ Append
Device: /dev/ttyUSB0

Image 11

(3) Send and receive data via cutecom

Enter 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/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

Device: /dev/ttyUSB0

send form USB-RS232

Input: LF Char delay: 0 ms Plain

send form RS232

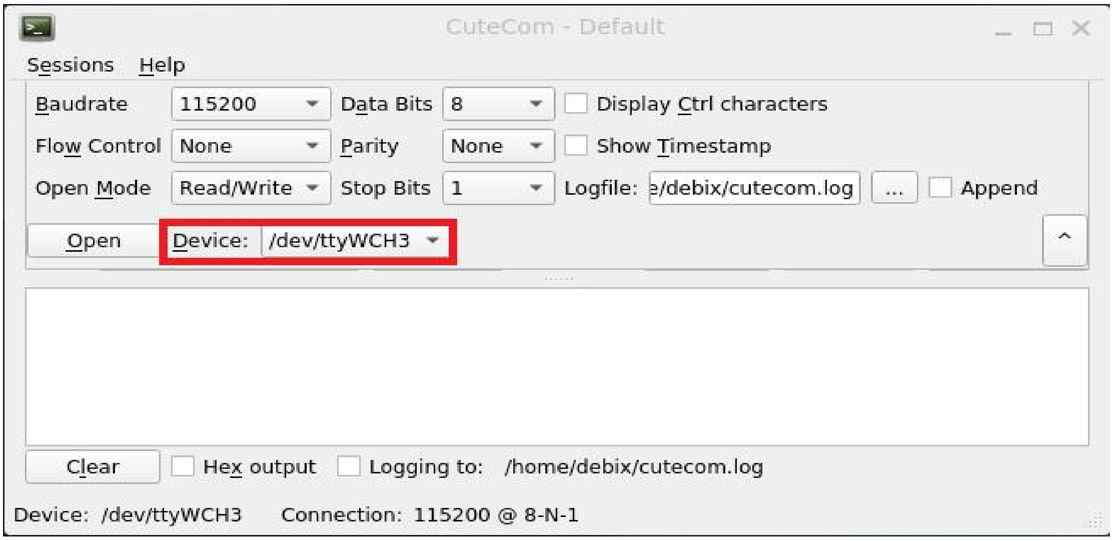
□ Hex output □ Logging to: /home/debix/cutecom.log

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

Image 12

RS485 Send and receive verification

(1) 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

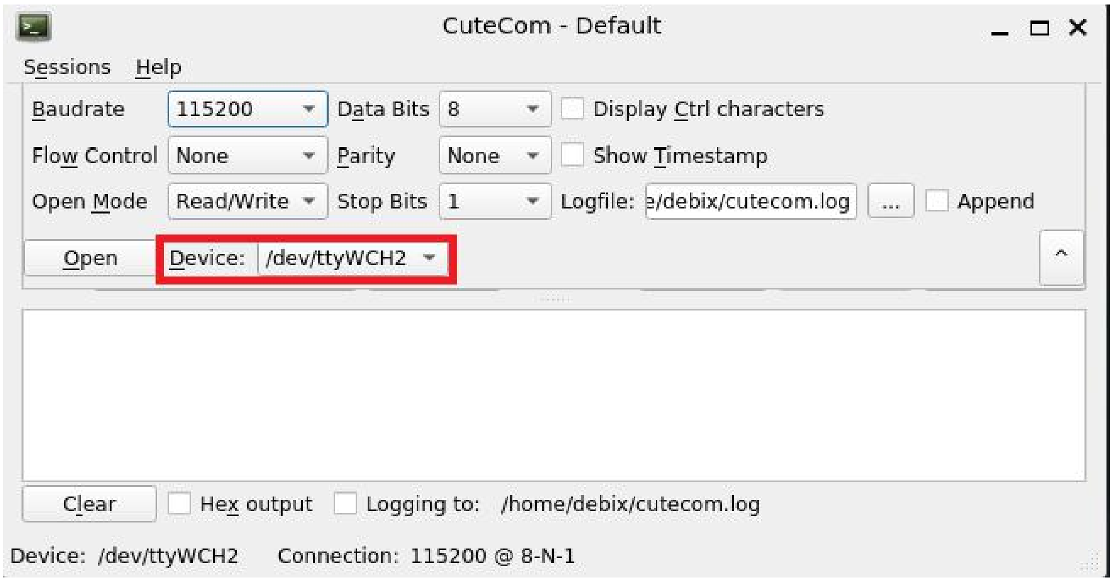
Device: /dev/ttyWCH3

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

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

Image 13

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

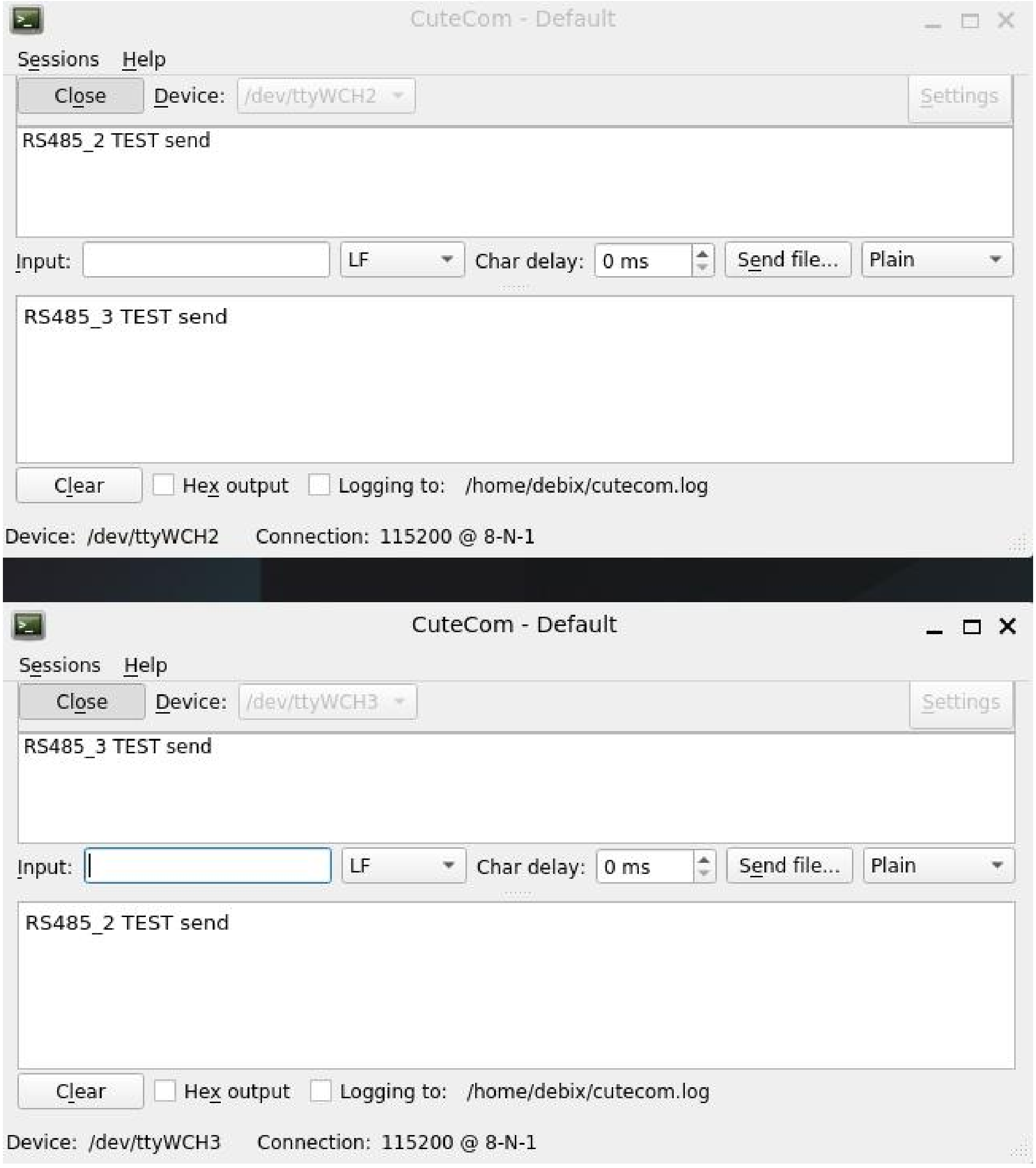
Device: /dev/ttyWCH2

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

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

Image 14

(2) Enter 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

Input:
LF
Char delay: 0 ms
Plain

RS485_3 TEST send

□ Logging to: /home/debix/cutecom.log

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

Device: /dev/ttyWCH3

RS485_3 TEST send

Input:
LF
Char delay: 0 ms
Plain

RS485_2 TEST send

□ Logging to: /home/debix/cutecom.log

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

Image 15

1.3 CAN

DEBIX SOM A I/O Board has 2 can transceivers on board.

Parameters are as follows：

Function name	Port	Pin	Definition	Device node
CAN	J23	1	CAN1H	can0
		2	CAN1L	
		3	CAN1GND	
	J24	1	CAN2H	can1
		2	CAN2L	
		3	CAN2GND	

1.3.1 Hardware Connection

- Connect Pin1 of J23 to Pin1 of J24
- Connect Pin2 of J23 to Pin2 of J24
- Connect Pin3 of J23 to Pin3 of J24

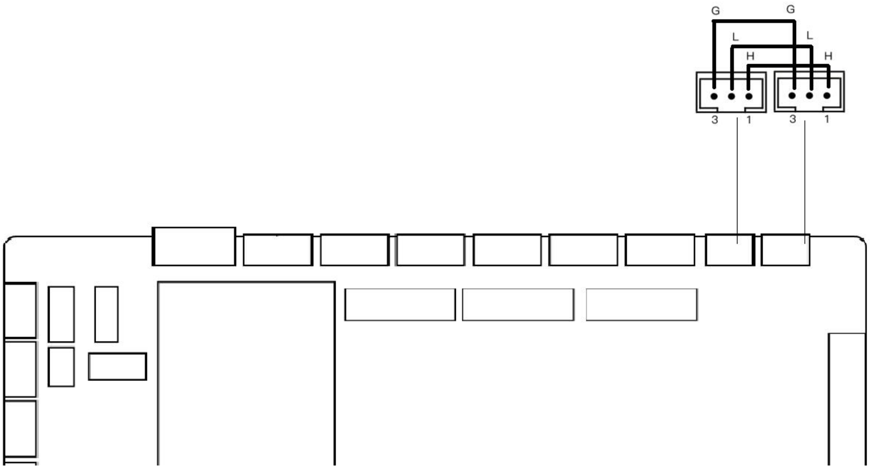


Image 16

1.3.2 Send and Receive Verification

CAN1 sends data, CAN0 receives data

Get root permission first

```
sudo su
```

(1) Open a terminal and configure CAN0

```
ifconfig can0 down
ip link set can0 type can bitrate 500000
ifconfig can0 up
```

(2) Configure CAN0 to receive

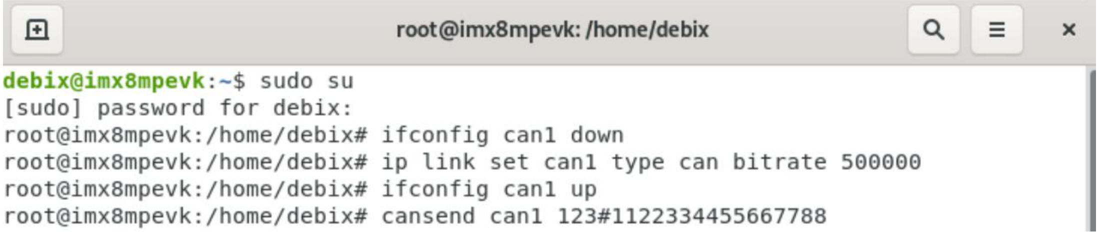
```
candump can0
```

(3) Open another terminal and configure CAN1 as sending

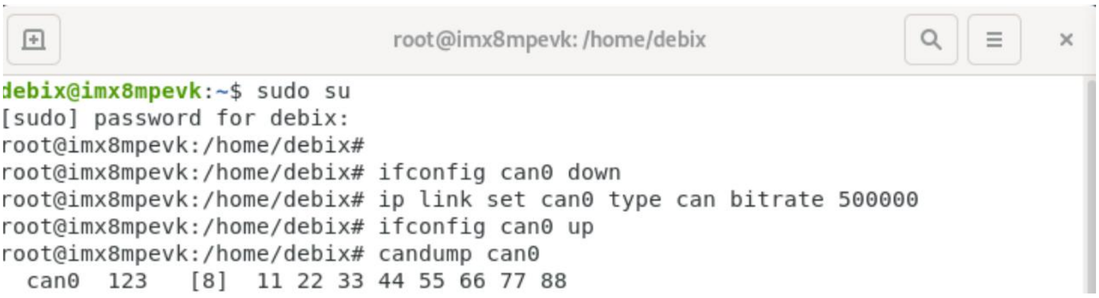
```
ifconfig can1 down
ip link set can1 type can bitrate 500000
ifconfig can1 up
```

(4) CAN1 sends data

```
cansend can1 123#1122334455667788
```



```
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
```



```
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
```

CAN1 receives data, CAN0 sends data

(1) In the terminal sent by CAN1:

```
candump can1
```

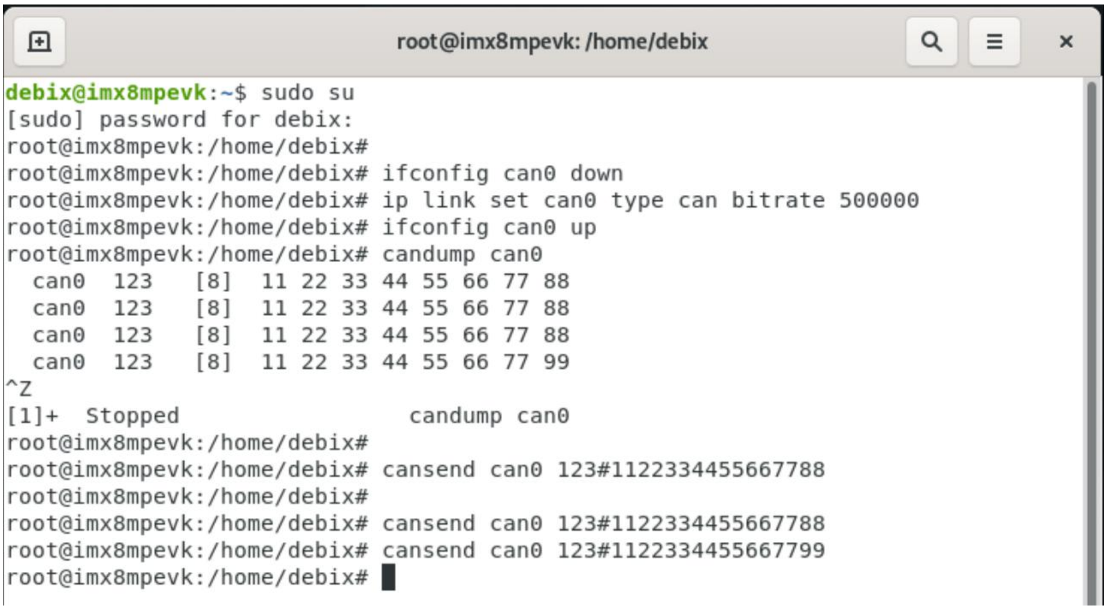
Switch can1 to receiving state.

(2) In the terminal receiving CAN0, press Ctrl+Z to end receiving.

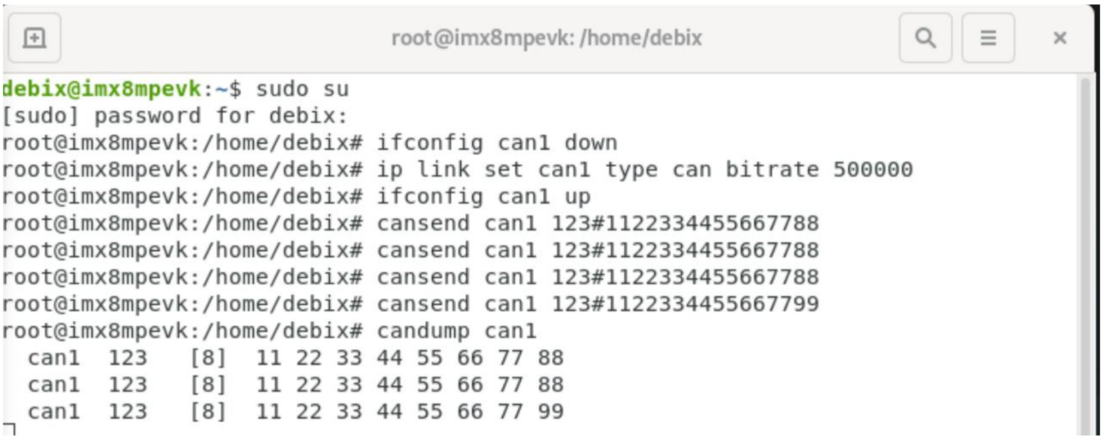
```
cansend can0 123#1122334455667788
```

Switch can0 to send state.

As shown in the image:



```
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# 
```



```
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# cansend can1 123#1122334455667788
root@imx8mpevk:/home/debix# cansend can1 123#1122334455667788
root@imx8mpevk:/home/debix# cansend can1 123#1122334455667799
root@imx8mpevk:/home/debix# candump can1
  can1  123   [8]  11 22 33 44 55 66 77 88
  can1  123   [8]  11 22 33 44 55 66 77 88
  can1  123   [8]  11 22 33 44 55 66 77 99
```

1.4 Nor-Flash

DEBIX SOM A I/O Board has a Nor-Flash on board.

Parameters as below:

Function name	Capacity	Device node
Nor-Flash	8MB	/dev/mtd0

1.4.1 Authentication method

Get root permission first

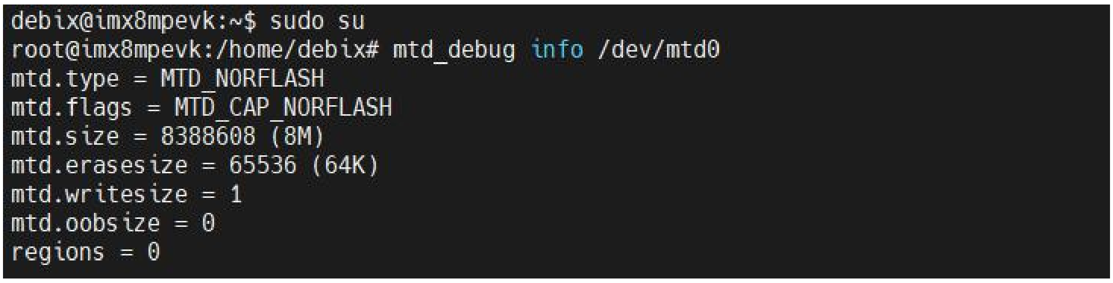
```
sudo su
```

（1）Install mtd-utils on DEBIX SOM A:

```
apt update
apt install mtd-utils
```

（1）View info command

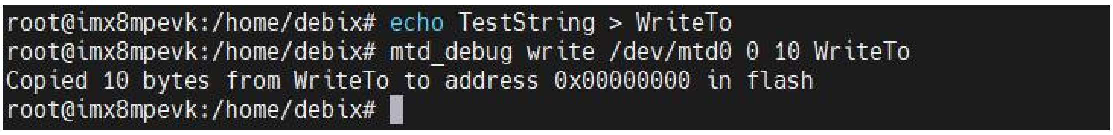
```
mtd_debug info /dev/mtd0
```



```
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
```

（2）Write verification:

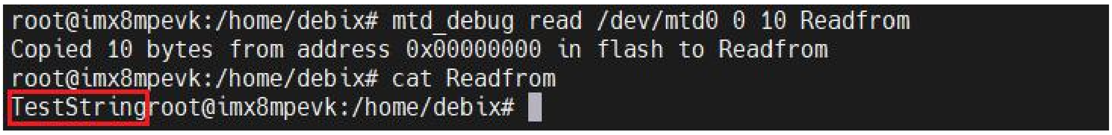
```
echo TestString > WriteTo
mtd_debug write /dev/mtd0 0 10 WriteTo
```



```
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#
```

（3）Read verification:

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

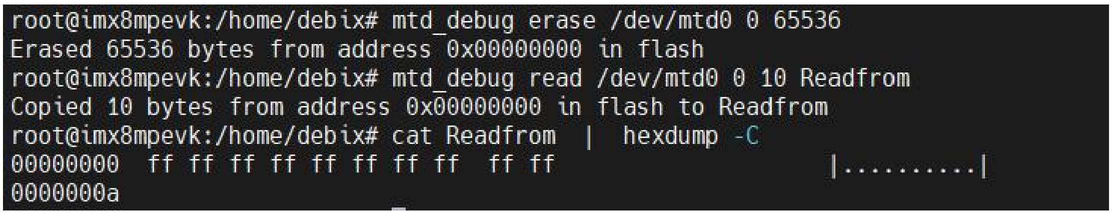


```
root@imx8mpevk:/home/debix# mtd_debug read /dev/mtd0 0 10 Readfrom
Copied 10 bytes from address 0x00000000 in flash to Readfrom
root@imx8mpevk:/home/debix# cat Readfrom
TestStringroot@imx8mpevk:/home/debix#
```

（4）Wipe Verification:

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

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



```
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
```

1.5 RTC

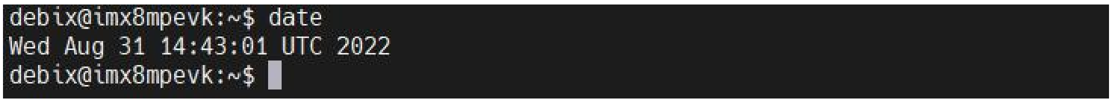
DEBIX SOM A I/O Board has an RTC interface (J38) on board.

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

1.5.1 Authentication Method

(1) View the current system time:

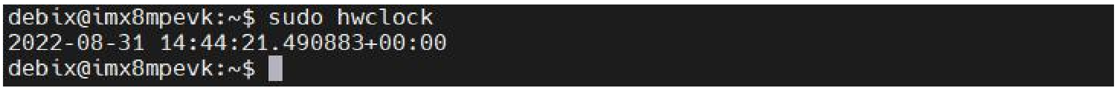
```
date
```



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

(2) View the current RTC time:

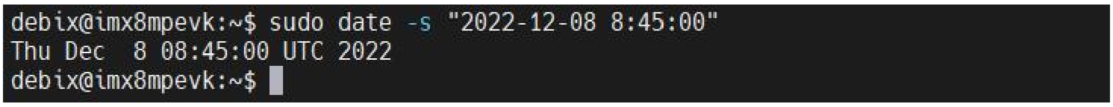
```
sudo hwclock
```



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

(3) Modify the current system time:

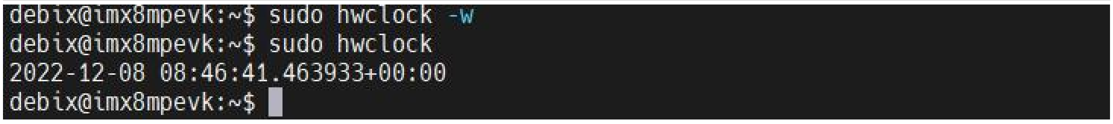
```
sudo date -s “2022-12-08 8:45:00”
```



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

(4) Write system time to RTC:

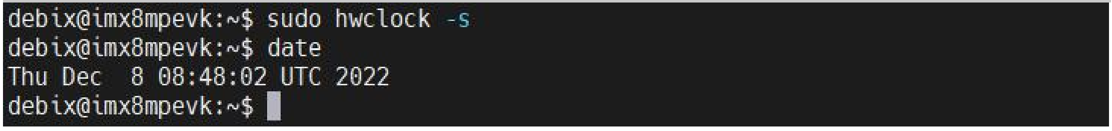
```
sudo hwclock -w
sudo hwclock                              // Check the RTC time
```



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

(5) Write RTC time to system time:

```
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:~$
```

1.6 USB 2.0

DEBIX SOM A I/O Board has 3 USB 2.0 ports on board.
Parameters as below:

Function name	Port	Pin	Definition
USB 2.0	J18	1	USB 5V
		2	USB_HUB_DM2
		3	USB_HUB_DP2
		4	GND
	J19	1	USB 5V
		2	USB_HUB_DM3
		3	USB_HUB_DP3
		4	GND
	J20	1	USB 5V
		2	USB_HUB_DM4
		3	USB_HUB_DP4
		4	GND

1.6.1 Hardware Connection

USB2.0 connects to the USB female head, as shown in the figure below:

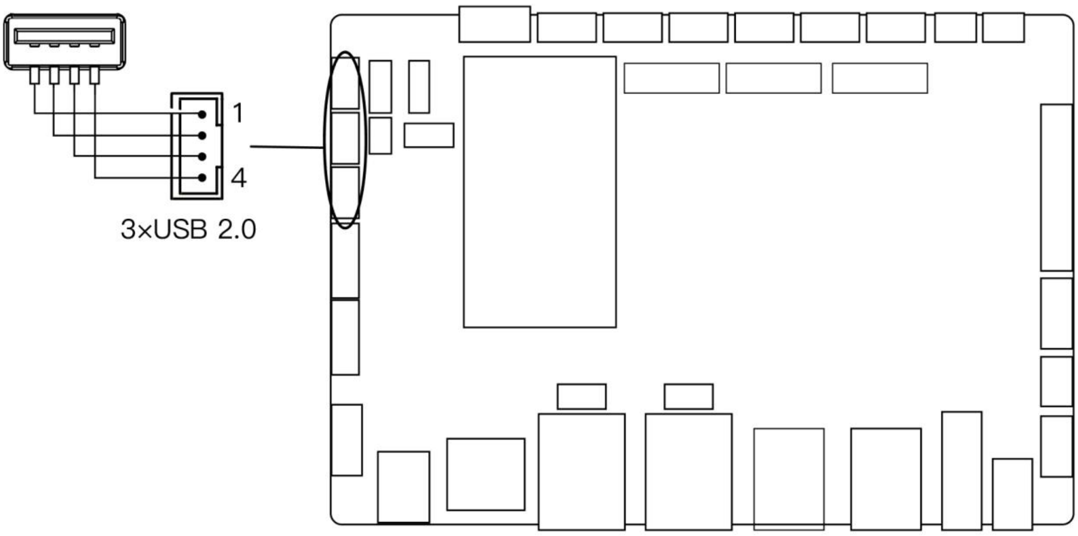
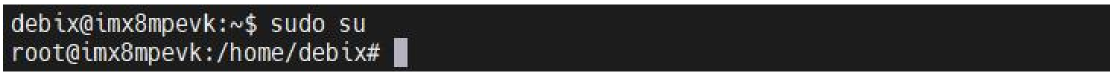


Image 17

1.6.2 Testing method

Get root permission first

```
sudo su
```



```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix#
```

(1) Mount the U disk (* is automatically generated for the device inserted into the system, modified according to the actual situation)

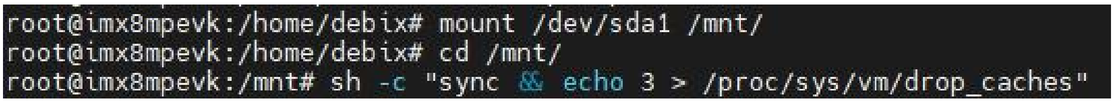
```
mount /dev/sd* /mnt
```

(2) Enter the U disk directory

```
cd /mnt
```

(3) Clear the cache, run before each read and write test command

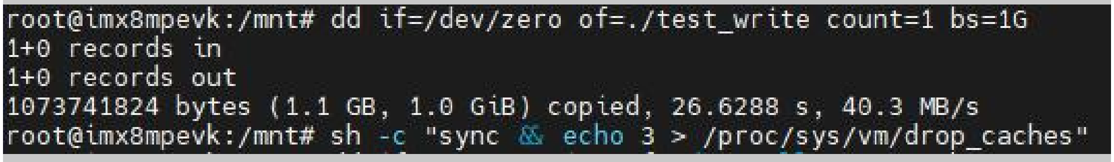
```
sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
```



```
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"
```

(4) Write speed test

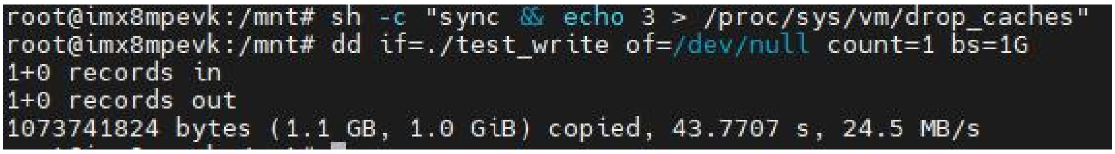
```
dd if=/dev/zero of=./test_write count=1 bs=1G
sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"        // clear cache
```



```
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"
```

(5) Reading speed test

```
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
```

1.7 Ethernet

DEBIX SOM A I/O Board has 2 Gigabit Ethernet ports on board.

1.7.1 Hardware Connection

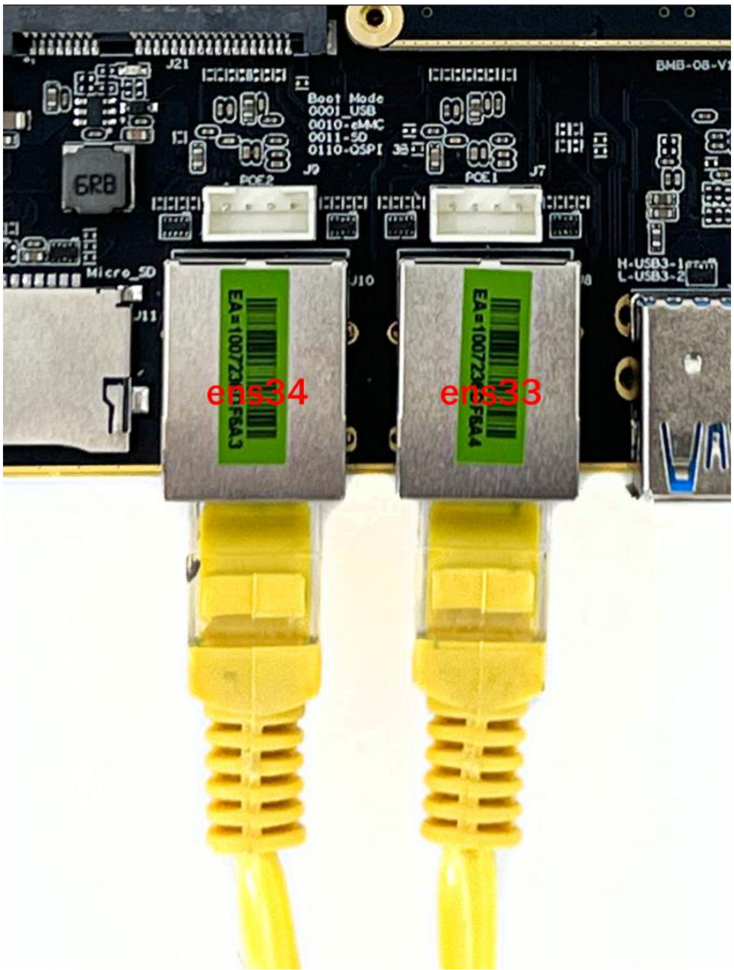
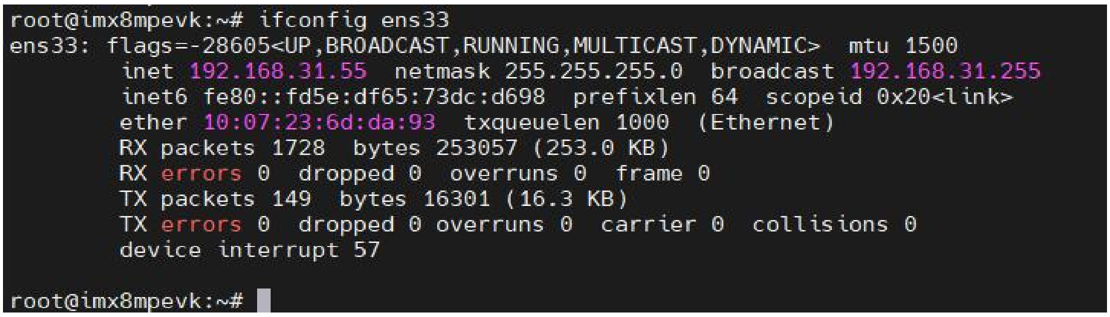


Image 18

1.7.2 Applications

Network port 1 (ENET_QOS), bit number: J8, port number: ens33:
Open a terminal and enter the command

```
ifconfig ens33
```

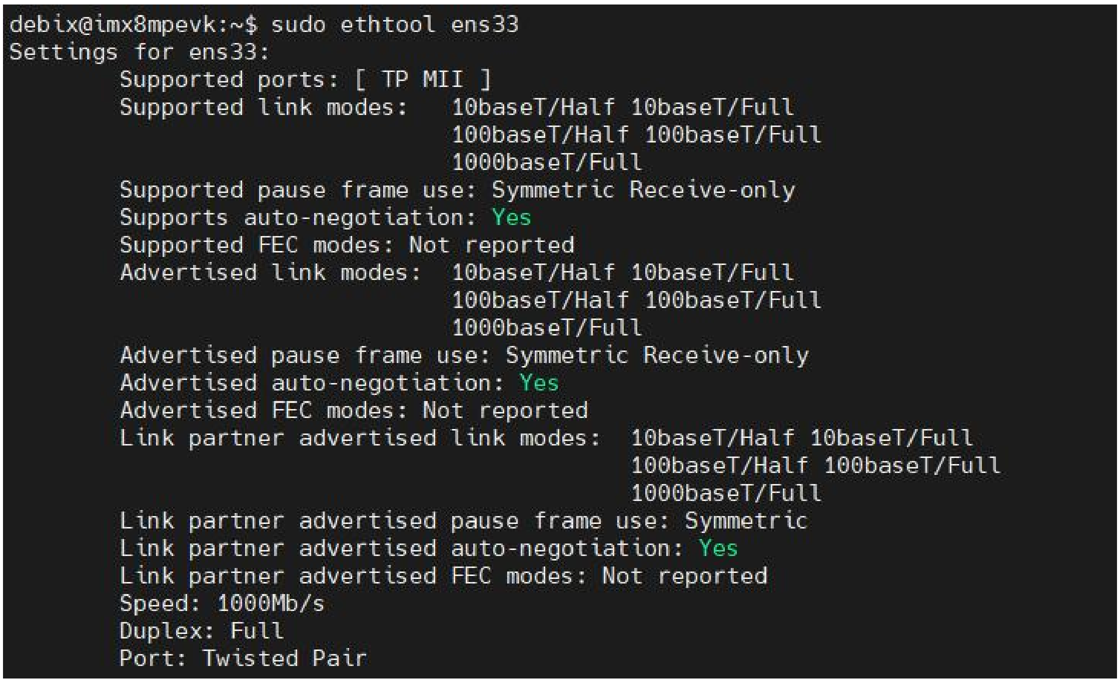


```
root@imx8mpevk:~# ifconfig ens33
ens33: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.31.55  netmask 255.255.255.0  broadcast 192.168.31.255
        inet6 fe80::fd5e:df65:73dc:d698  prefixlen 64  scopeid 0x20<link>
        ether 10:07:23:6d:da:93  txqueuelen 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:~# 
```

Query the network port speed

```
sudo ethtool ens33
```

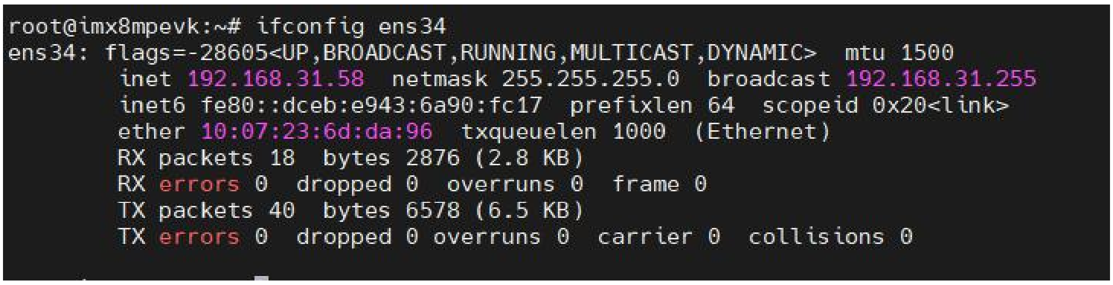


```
debix@imx8mpevk:~$ sudo ethtool ens33
Settings for ens33:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric 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, port number: ens34:

Open a terminal and enter the command

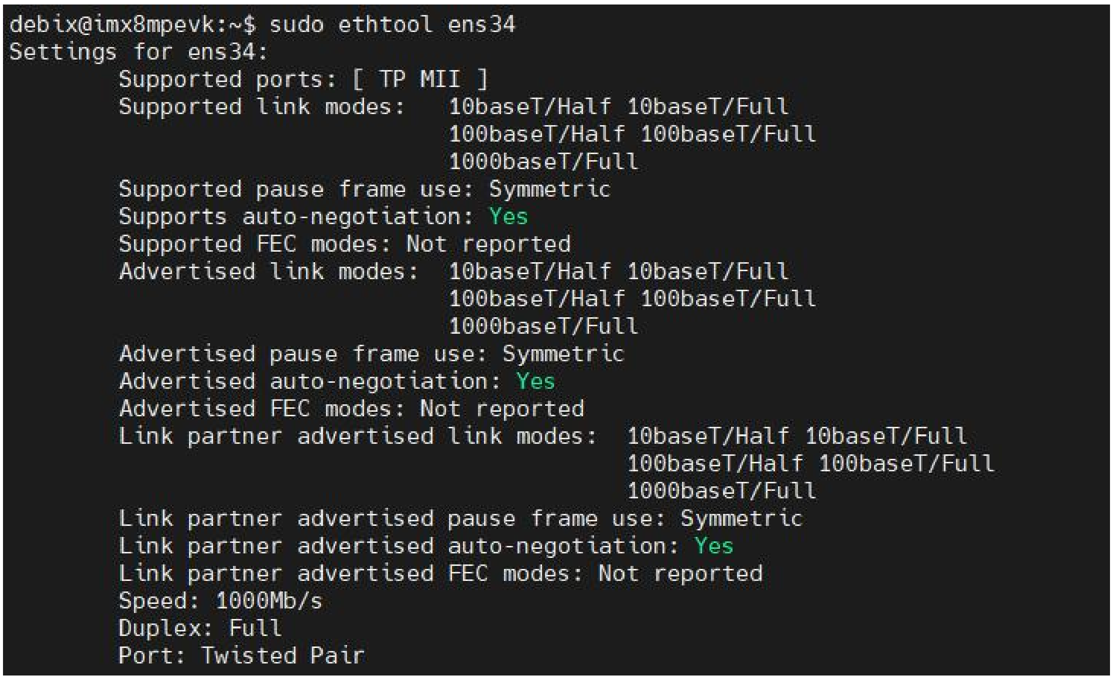
```
ifconfig ens34
```



```
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
```

Query the network port speed

```
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
```

1.8 Led&key

DEBIX SOM A I/O Board has a Led&key interface (J50) on board, including 2 LEDs and an ON/OFF button.

Parameters as below:

Function name	Port	Pin	Definition
	J50	1	DC_IN
		2	VSYS_3V8
LED		3	GPIO_LED1
		4	GPIO_LED2
key		5	ON/OFF
		6	GND

1.8.1 Hardware Connection

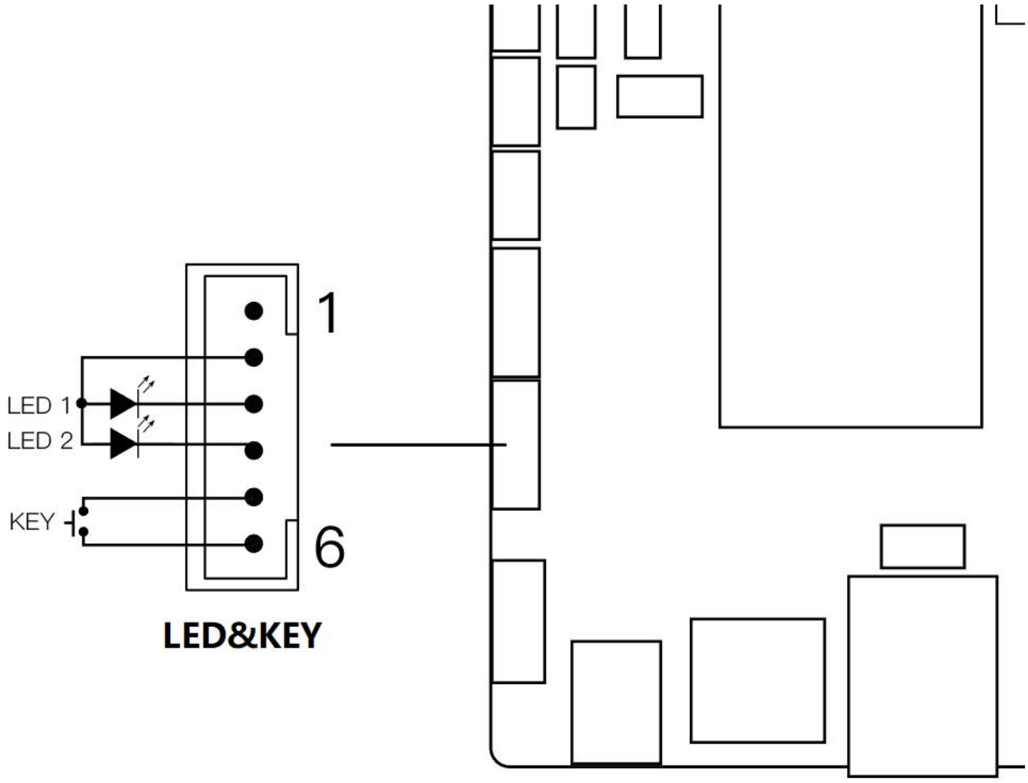


Image19

1.8.2 Led and Key Verification

GPIO_LED1 device file path

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

GPIO_LED2 device file path

```
/sys/devices/platform/gpio-leds/leds/GPIO_LED2
```

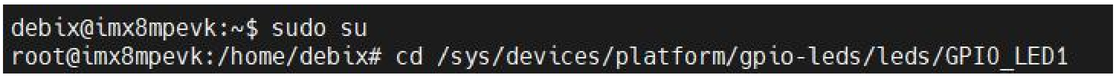
Take GPIO_LED1 as an example:

Get root permission first

```
sudo su
```

(1) 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
```

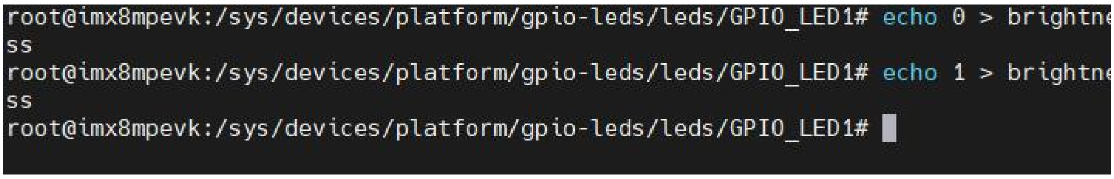
(2) Output setting

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# █
```

KEY

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

- Short press
  - When the green light is off, the system enters the sleep state.
  - Short press again, the green light flashes 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.

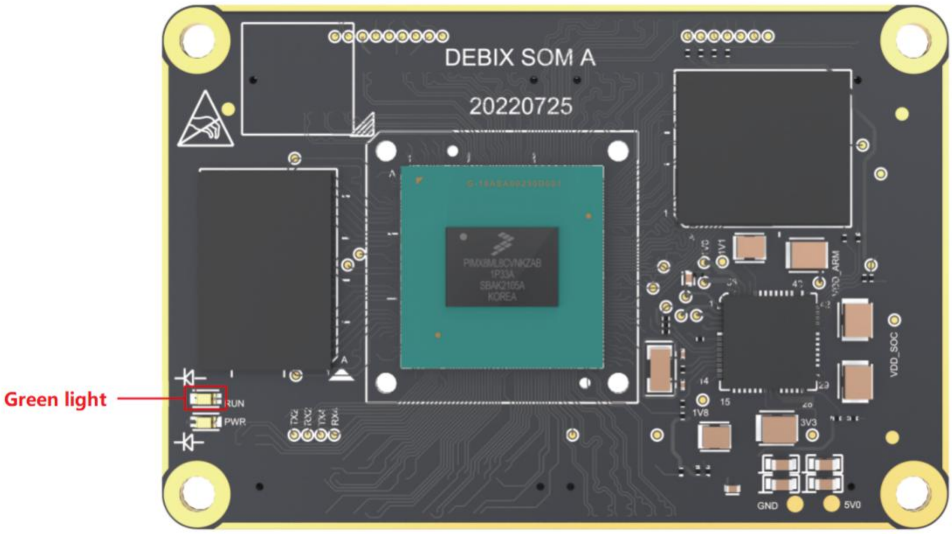

Green light
DEBIX SOM A
20220725
RUN
PWR
GND
5V0
3V3
1V8
VDD_ARM
VDD_SOC

Image 20

1.9 DO

DEBIX SOM A I/O Board has 1 DO interface (J40) on board, supports 4 physically isolated DOs, supports wet nodes, and is compatible with external relay dry nodes.

Parameters as below:

Function name	Port	Pin	Definition
DO	J40	1	DO_PCOM
		2	DO_1
		3	DO_2
		4	DO_3
		5	DO_4
		6	DO_GND

Electrical performance:

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

1.9.1 Hardware Connection

The block diagram of the isolated digital output connection is as follows:

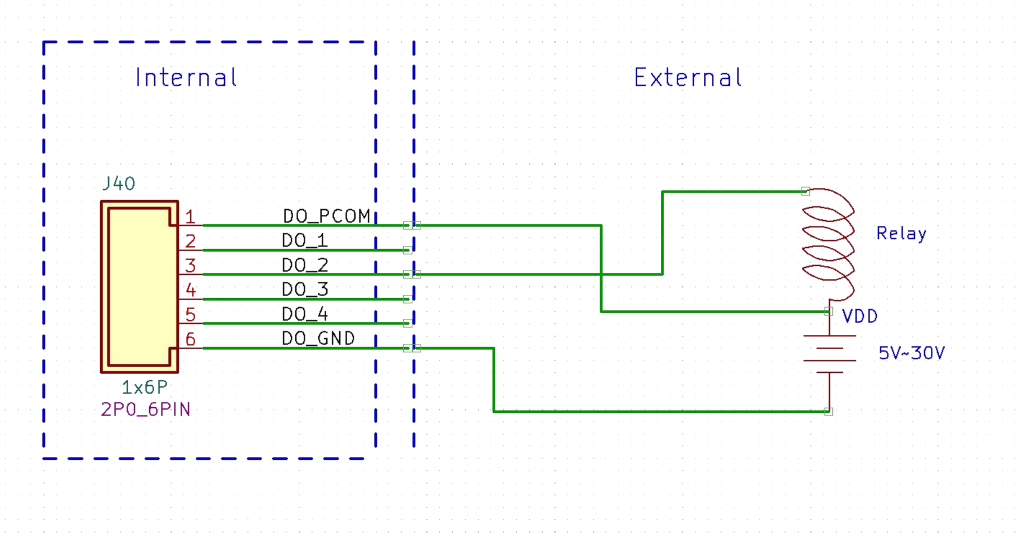


Image 21

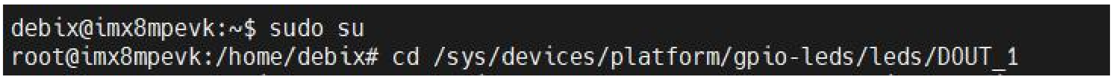
1.9.2 Applications

Get root permission first

```
sudo su
```

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
```

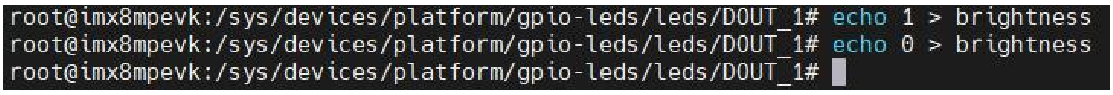
Note: GPIO output is low, DO_1 open-drain output is high configuration, and the relay is turned off.

```
echo 0 > brightness
```

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

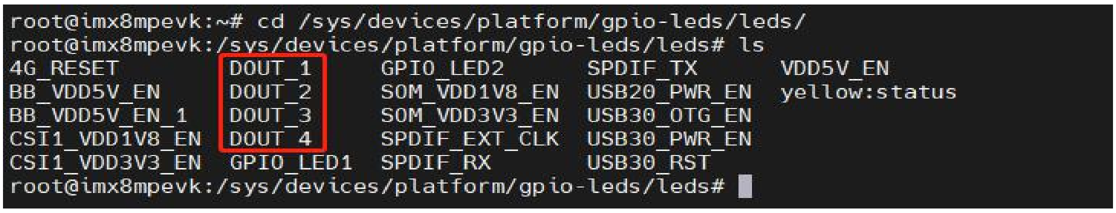
The current supported by DO_1 is 500mA.

```
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#
```

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#
```

1.10 DI

DEBIX SOM A I/O Board has 1 DI interface (J41) on board, supports 4 physically isolated DIs, and supports dry node input and wet node input.

Parameters as below:

Function name	Port	Pin	Definition
DI	J41	1	DI_ISO_12V
		2	DI_1
		3	DI_2
		4	DI_3
		5	DI_4
		6	DI_GND

Electrical performance:

- Wet contact
  - 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-circuit state with GND: low level;

1.10.1 Hardware Connection

Take the dry node connection as an example, connect the Pin2 DI_1 of J41 to the ground in series, as shown in the figure:

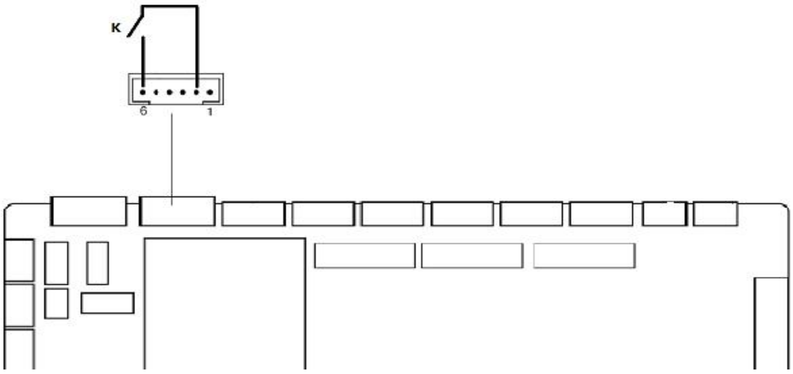


Image 22

1.10.2 Applications

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

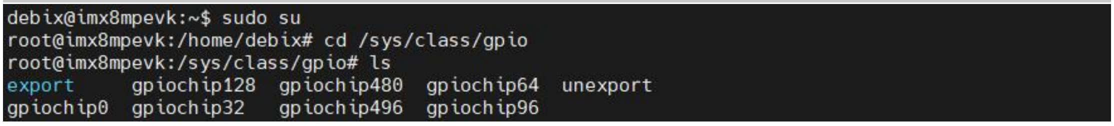
Definition	Pin number	IO Port	GPIO number
DI_1	2	GPIO1_IO5	5
DI_2	3	GPIO1_IO6	6
DI_3	4	GPIO1_IO7	7
DI_4	5	GPIO1_IO8	8

The operation method of all ports of DI is the same, here we take DI_1 as an example.
Get root permission first

```
sudo su
```

(1) Enter the GPIO directory

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



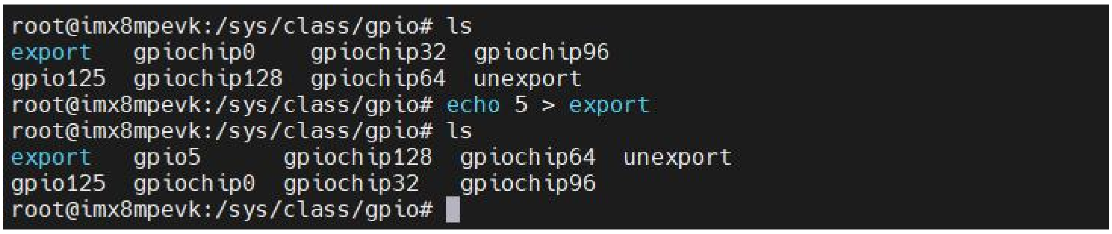
```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/class/gpio
root@imx8mpevk:/sys/class/gpio# ls
export      gpiochip128  gpiochip480  gpiochip64  unexport
gpiochip0   gpiochip32   gpiochip496  gpiochip96
```

(2) Export GPIO

```
echo 5 > export
```

Note: Parameter 5 is calculated according to the GPIO number calculation formula.

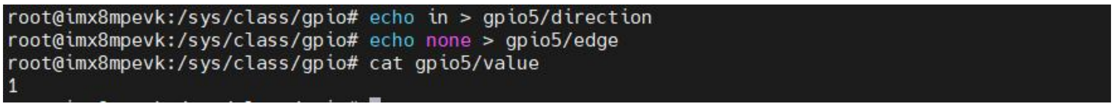
After the export is successful, the gpio5 directory will be automatically generated, as shown below:



```
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# 
```

（3）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
```

1.11 LVDS BL CTRL

There is one LVDS backlight control interface (J29) on DEBIX SOM A I/O Board.

Parameters as below:

Function name	Port	Pin	Definition
PWM backlight	J29	1	DC_IN
		2	DC_IN
		3	LVDS_BL_EN
		4	LVDS_BL_PWM
		5	GND
		6	GND

Note: DC_IN is directly connected to the backlight control interface by the DC input power supply, and the voltage range is consistent with the input voltage of the whole board (12-36V)

1.11.1 Applications

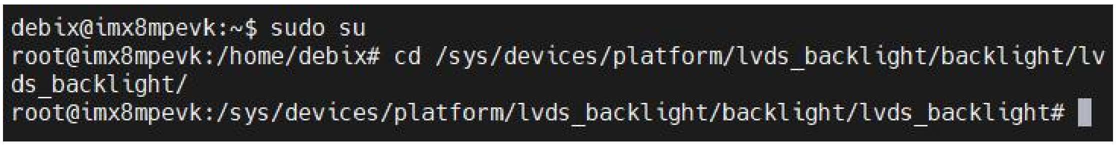
DEBIX SOM A IO Board defaults to HDMI output. To use LVDS output, you need to open the add on board application in DEBIX, select a suitable LVDS screen, and then restart the system.

Get root permission first

```
sudo su
```

(1) Switch to the LVDS backlight control directory

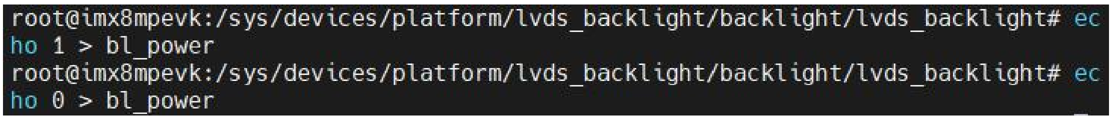
```
cd /sys/devices/platform/lvds_backlight/backlight/lvds_backlight
```



```
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# 
```

(2) Backlight switch (on by default)

```
echo 1 > bl_power               //Turn off backlight
echo 0 > bl_power               //Turn on backlight
```

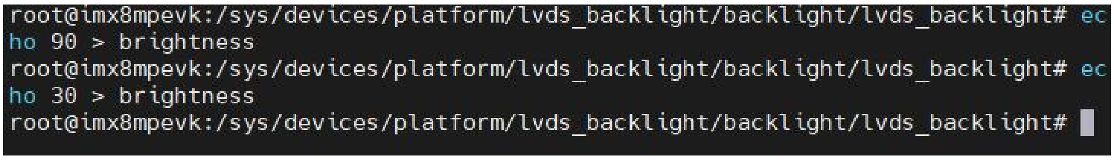


```
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 1 > bl_power
root@imx8mpevk:/sys/devices/platform/lvds_backlight/backlight/lvds_backlight# ec
ho 0 > bl_power
```

（3）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%
```

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



```
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# 
```

1.12 Mini PCIe

DEBIX SOM A I/O Board has a Mini PCIe interface onboard.

1.12.1 Hardware Connection

The connection of the Mini PCIe 4G module is as shown in the figure:

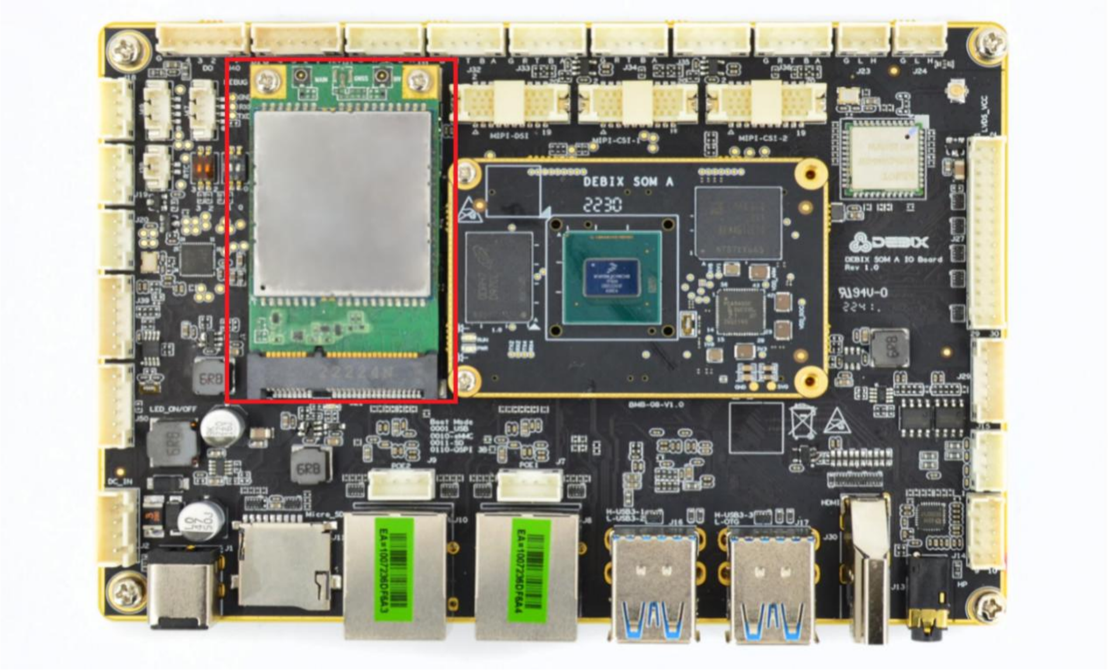


Image 23

The connection of the Mini PCIe SATA module is as shown in the figure:

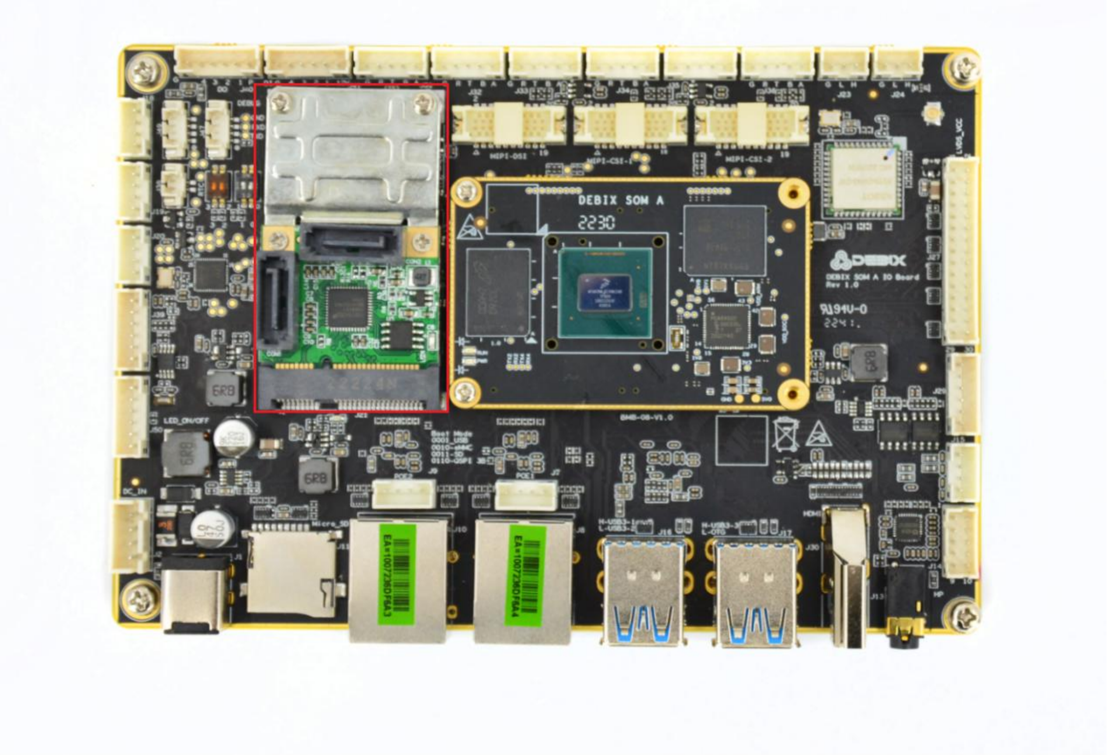
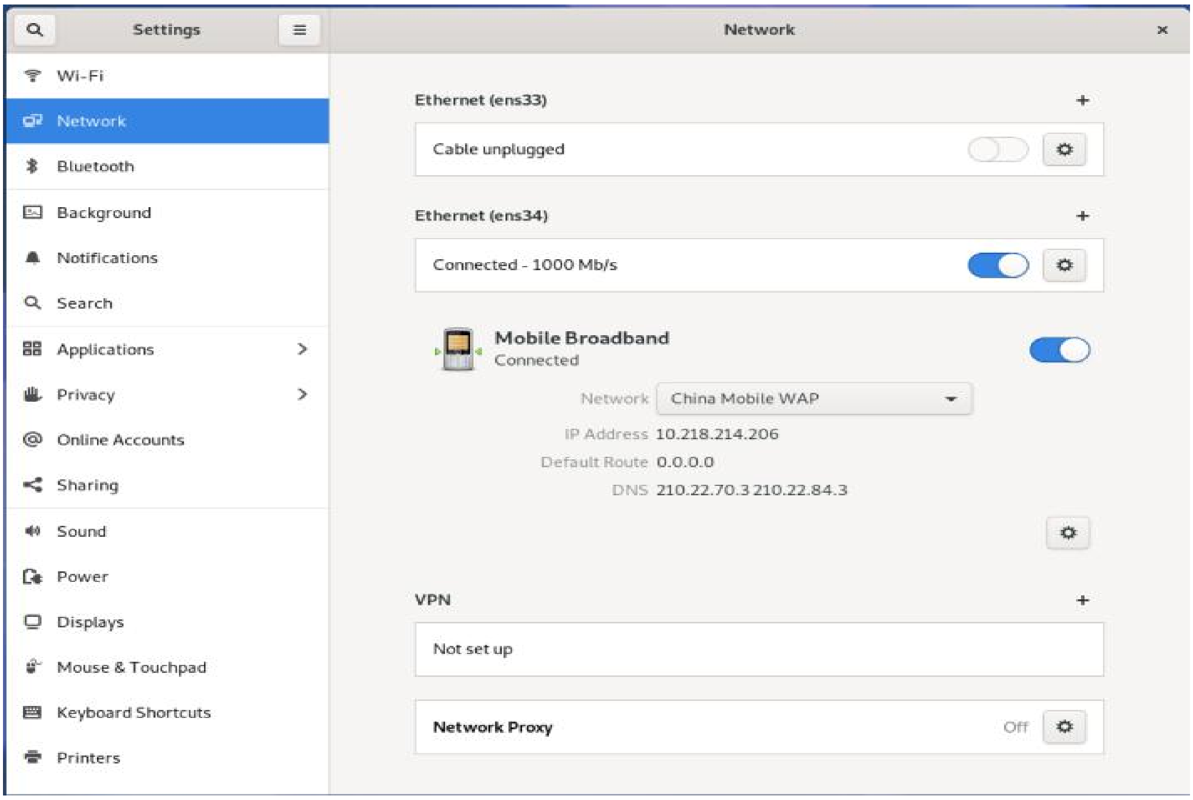


Image 24

1.12.2 Application

Mini PCIe 4G Module

After connecting the 4G module and SIM card, boot into the desktop, configure in Setting-Network, and after the IP appears, ping the Ethernet to test the connectivity.



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

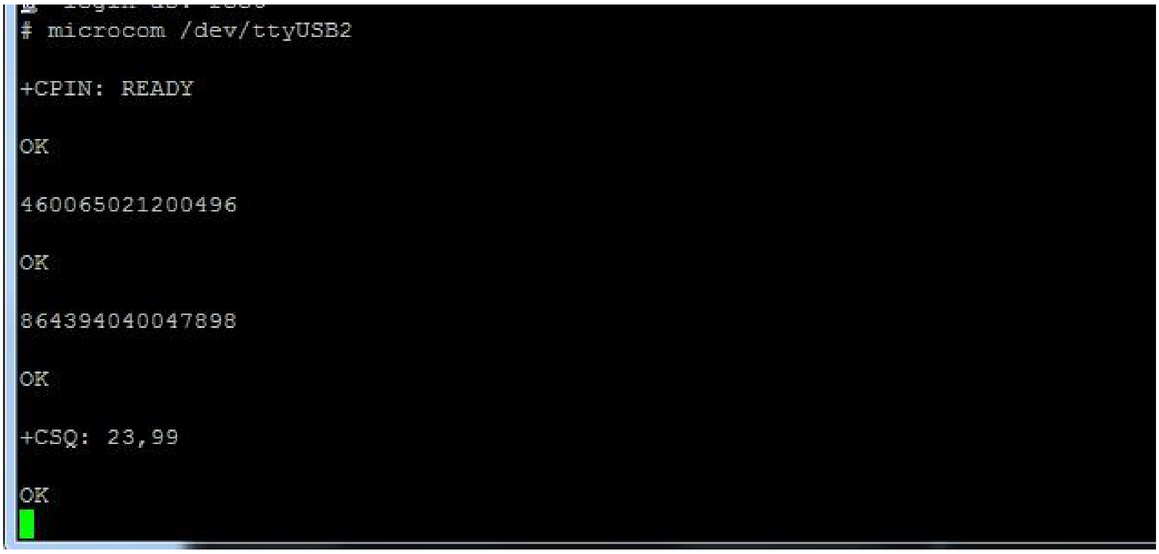
Network Proxy Off

Image 25

4G module verification

The Mini PCIe 4G device node is /dev/ttyUSB2, which can be verified by the relevant instructions of the serial port debugging tool microcom.

```
microcom /dev/ttyUSB2
AT+CPIN?             #SIM card detection
AT+CIMI          #Query SIM card number CIMI
AT+CGSN              #Query module IMEI
AT+CSQ           # query signal strength
```



```
# microcom /dev/ttyUSB2

+CPIN: READY

OK

460065021200496

OK

864394040047898

OK

+CSQ: 23,99

OK
```

Image 26

Mini PCIe SATA

The test method is the same as that of USB 2.0, the Mini PCIe SATA device node is /dev/sd*, * is automatically generated by the system when the device is inserted, and modified according to the actual situation.

1.13 I2C

DEBIX SOM A adapts to the corresponding IO Board. I2C4 mounts RTC, EEPROM, and I2C external expansion interface J39 on the bottom board. This interface can be used for TP, etc.

Equipment	Model	Address
RTC	HYM8563S	0x50
EEPROM	BL24C02F	0x51

I2C touch screen interface (J39) parameters are as follows

Function name	Port	Pin	Definition	Description
I2C	J39	1	VDD	Touch screen power supply, the default voltage is 3.3V (1.8V is optional)
		2	TP_SCL	The I2C clock port of the touch screen (controlled by I2C4) defaults to 3.3V voltage (1.8V is optional)
		3	TP_SDA	The I2C data port of the touch screen (controlled by I2C4) defaults to 3.3V voltage (1.8V is optional)
		4	TP_RESET	Touch screen reset terminal, the default voltage is 3.3V (1.8V is optional)
		5	TP_nINT	Touch screen interrupt port, the default voltage is 3.3V (1.8V is optional)
		6	GND	To ground

1.13.1 Authentication method

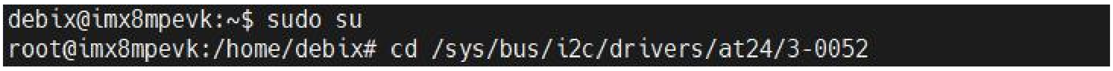
Take EEPROM as an example:

Get root permission first

```
sudo su
```

Switch to the control directory

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



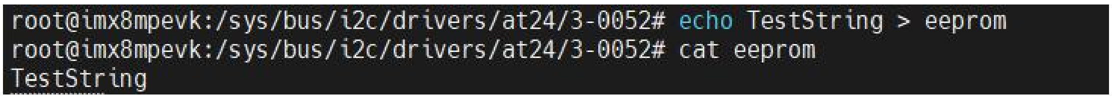
```
debix@imx8mpevk:~$ sudo su
root@imx8mpevk:/home/debix# cd /sys/bus/i2c/drivers/at24/3-0052
```

Write data

```
echo TestString > eeprom
```

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
```