XPI-3566-Zero Software Usage Guide

V1.0

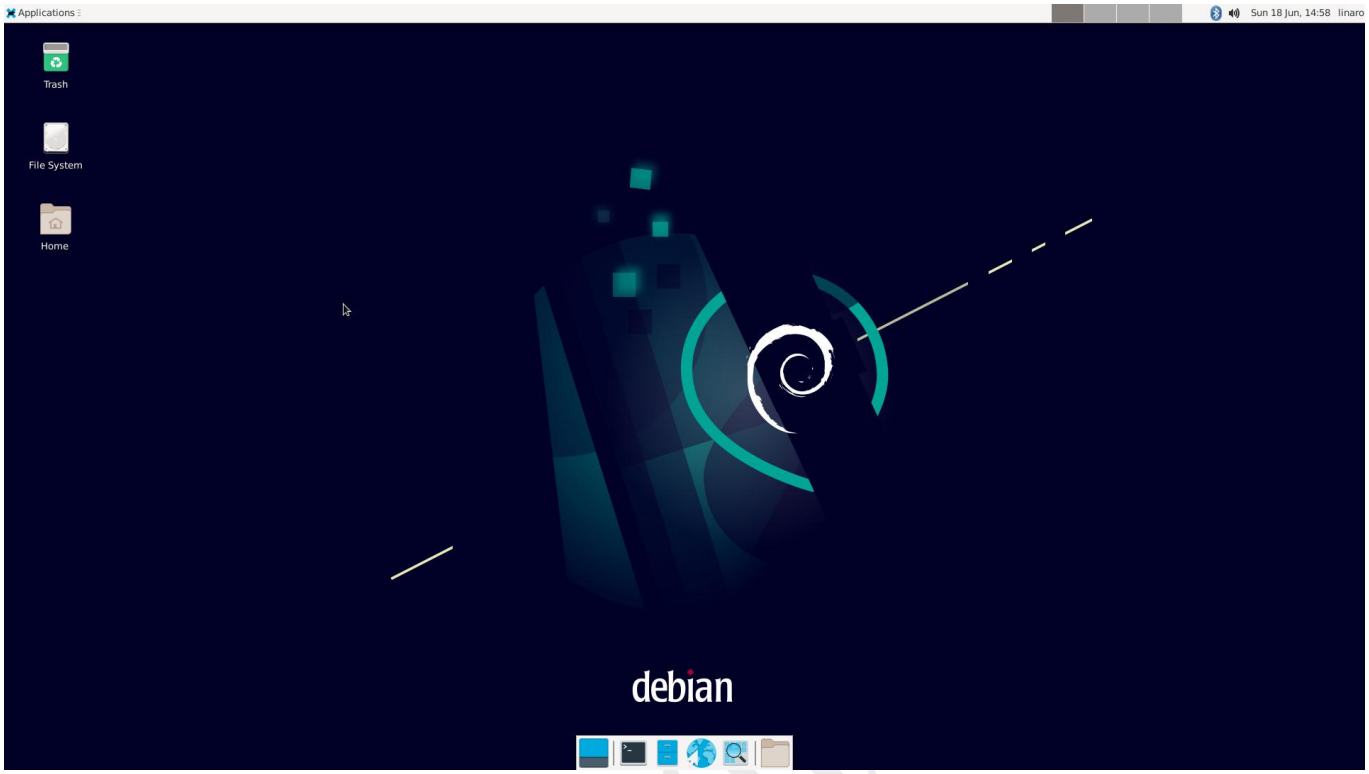
Catalog

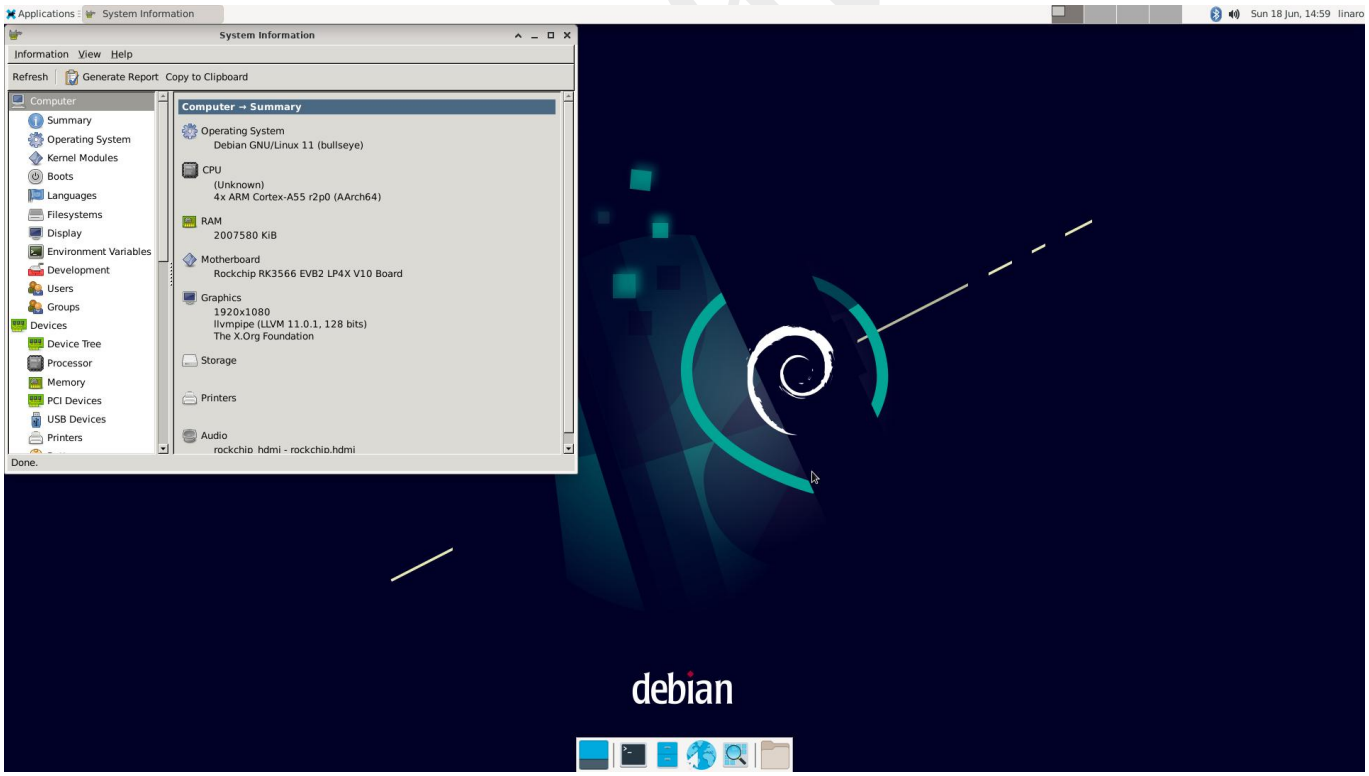
REVISION HISTORY

Time	Version	Purpose	Author	Comment
21/12/2024	1.0	Create a document	YYT	

1.XPI-3566-Zero Debian11 OS

XPI-3566-Zero Supported OS: Debian GNU/Linux 11 \n \l



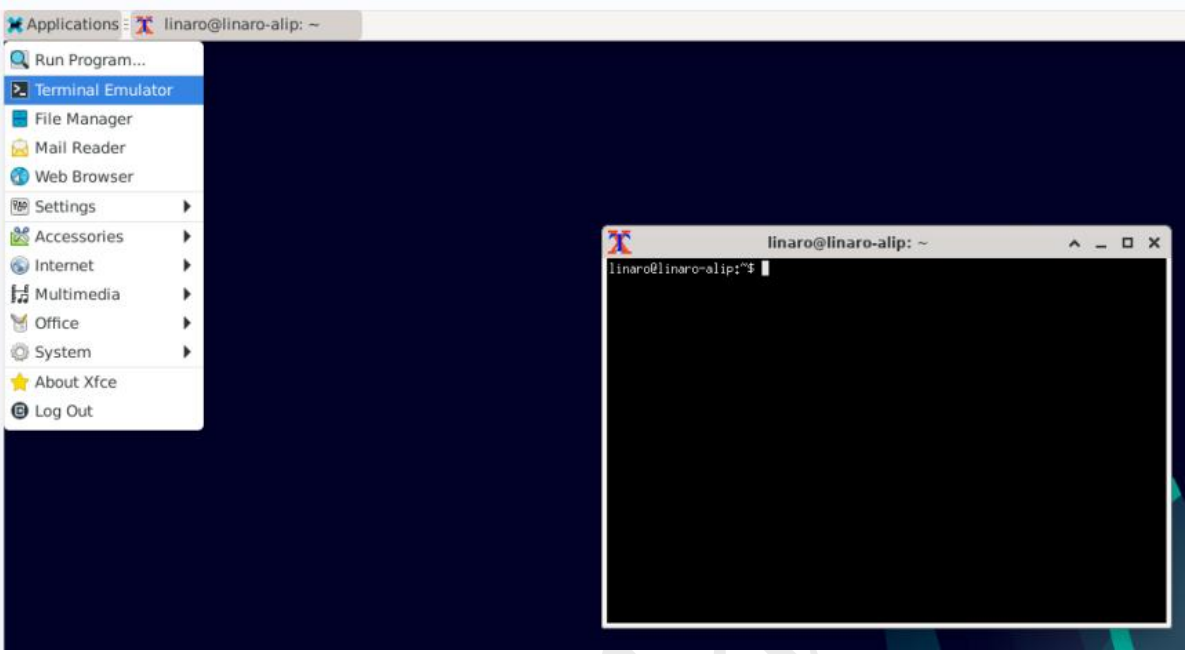


2. Function Description

2.1 How to access the OS

2.1.1 HDMI Display

XPI-3566-Zero supports HDMI OUT display, default resolution 1920x1080. It can be connected to the monitor and wait for the system start up, connecting the keyboard, open a terminal via the main menu -> “Application ”-> “Terminal Emulator”.



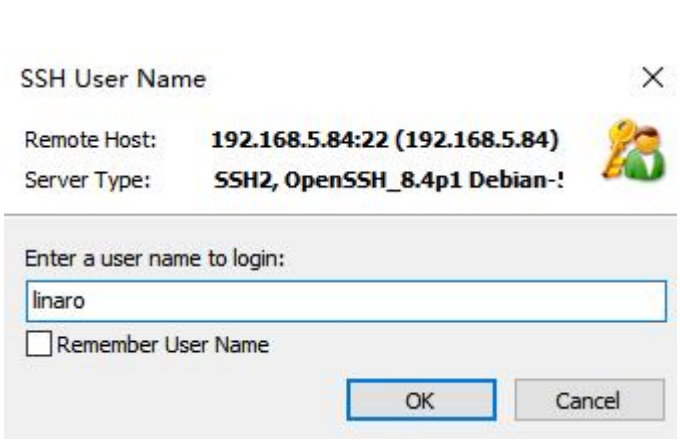
2.1.2 Remote Connection

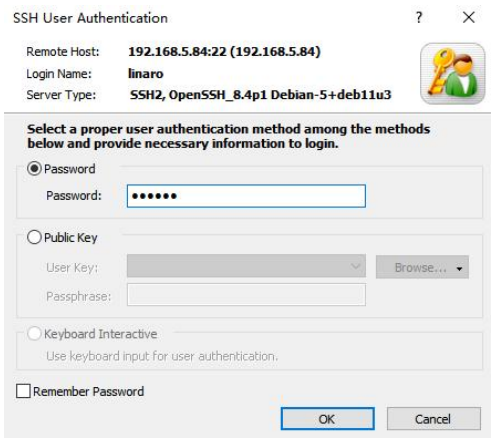
Connect the PC and XPI-3566-Zero in a local area network, get the IP address and connect via the serial tool using SSH.

Protocol : SSH2

User name: linaro

Password : linaro



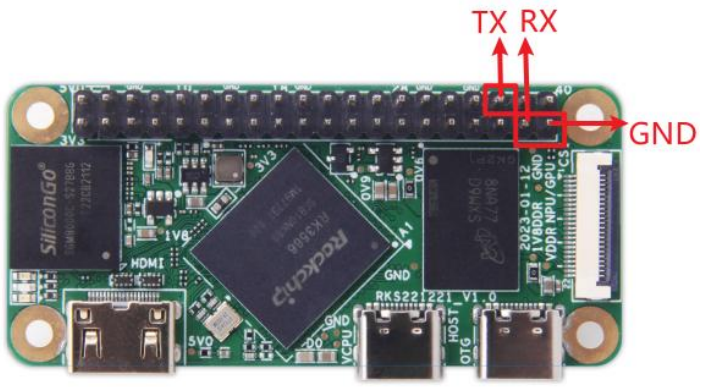


```
192.168.5.84
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
linaro@linaro-alip:~$
linaro@linaro-alip:~$
linaro@linaro-alip:~$ sudo su
root@linaro-alip:/home/linaro# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:44:c5:30:84  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.5.84  netmask 255.255.255.0  broadcast 192.168.5.255
        inet6 fe80::81bf:d0de:7d3f:953a  prefixlen 64  scopeid 0x20<link>
        ether b8:2d:28:57:61:00  txqueuelen 1000  (Ethernet)
        RX packets 449  bytes 39009 (38.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 491  bytes 407004 (397.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@linaro-alip:/home/linaro#
root@linaro-alip:/home/linaro#
```

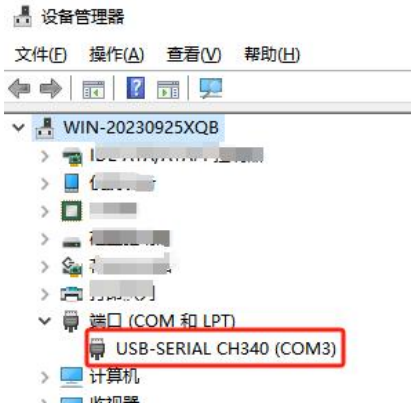
2.1.3 Serial Access

(1) Connect the USB port to the computer and the other end to the UART port on the board.

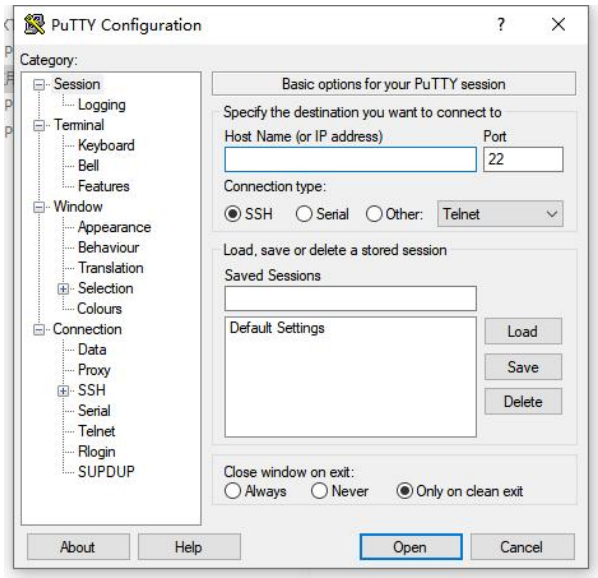
Note: Only connect TX/RX/GND.



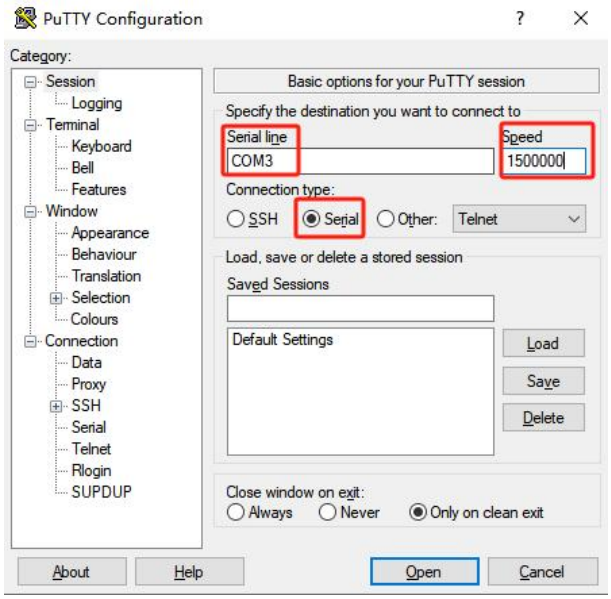
(2) In the device manager on the PC, view the serial port identification number.



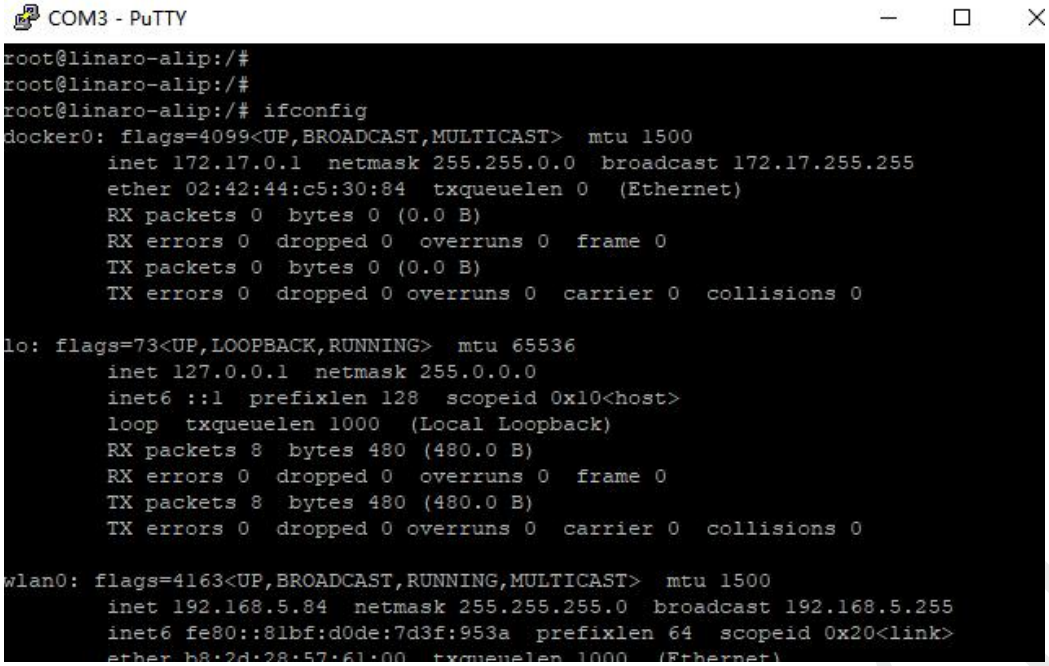
(3) Double-click putty.exe to open it.



(4) Click "session", select serial port "serial", select the new serial port name comX after the computer is connected to the UART, and set the corresponding serial port baud rate.



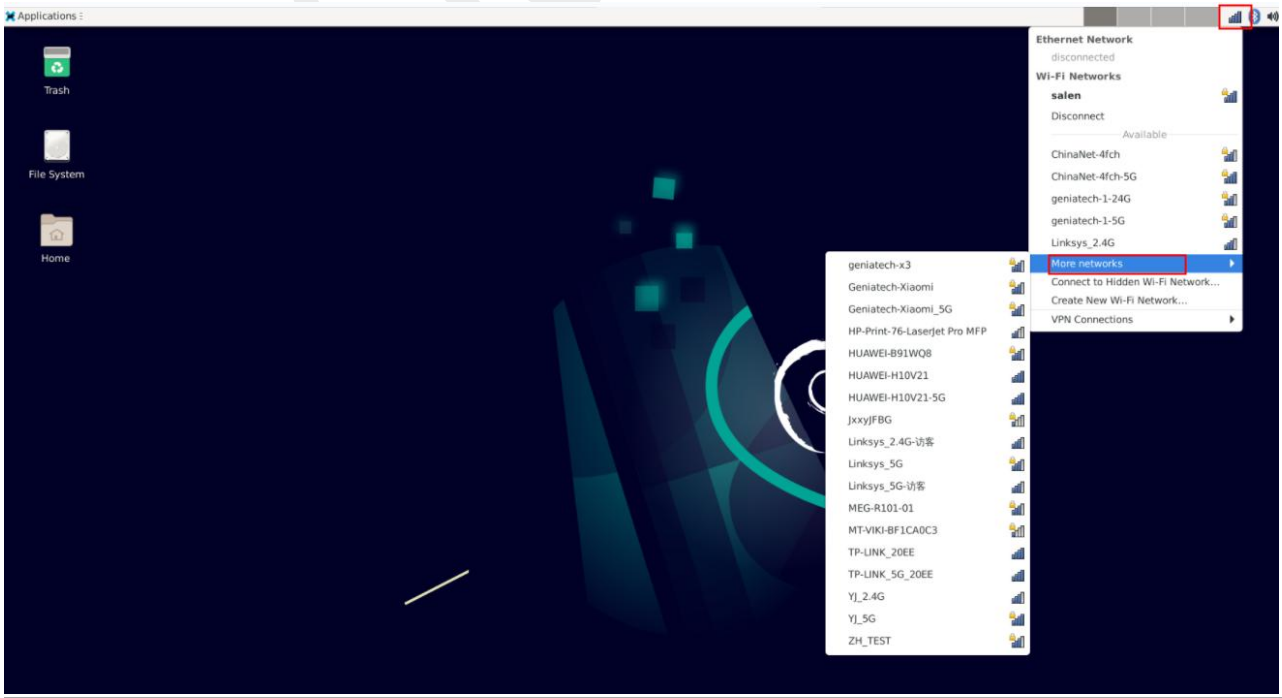
(5) Click "open" to launch the tool.

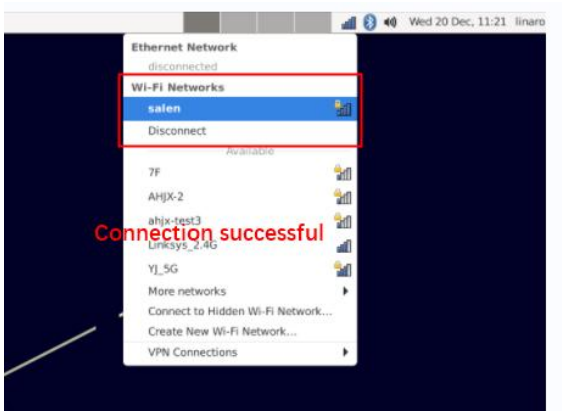


2.2 Network Function

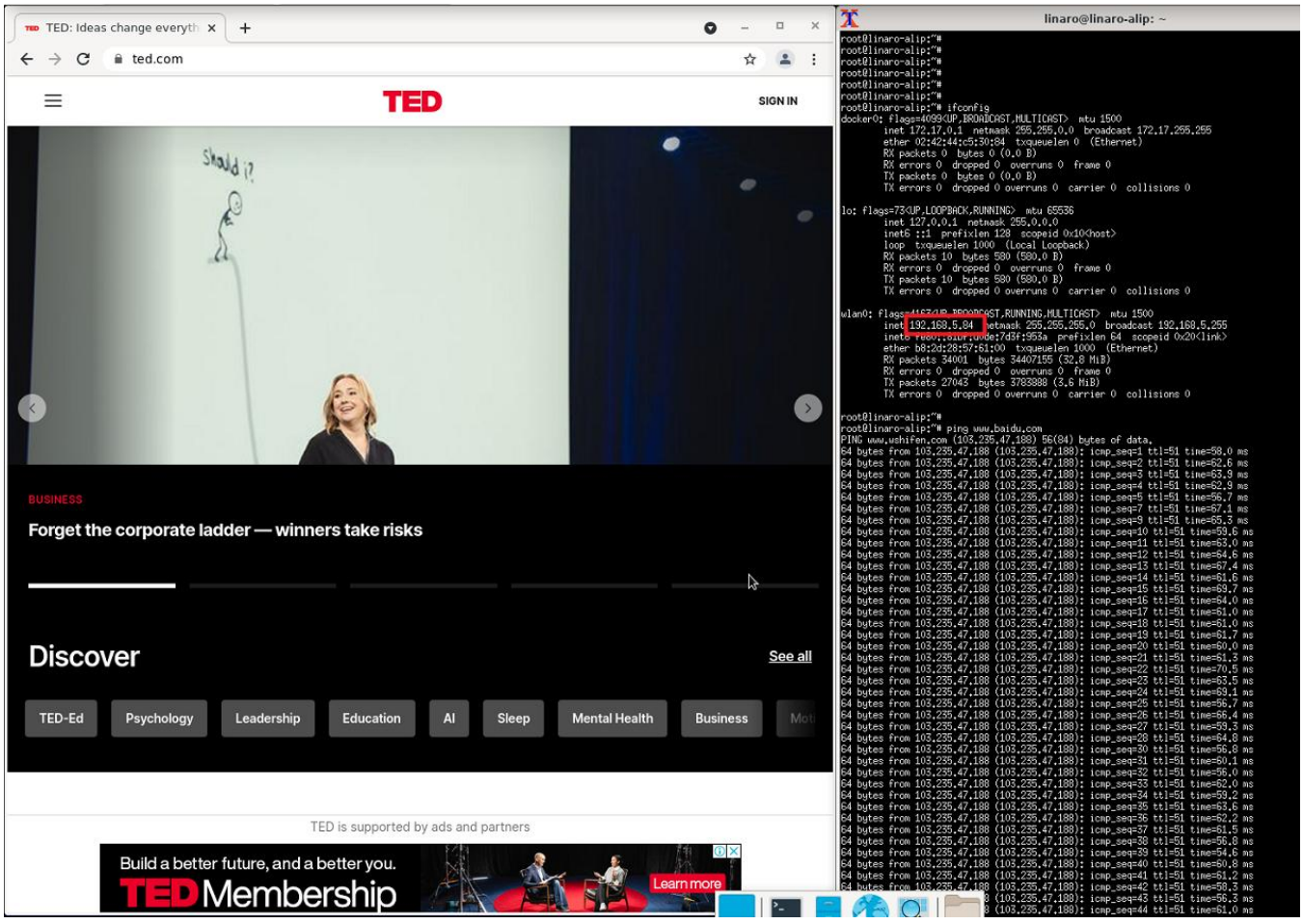
2.2.1 WIFI

WiFi interface connection: click “Network status”in the upper right corner -> click “More networks” -> select search WiFi -> Enter password/direct connection; Execute instructions, check the obtained IP address, ping through Baidu or through the browser, verify the normal network.





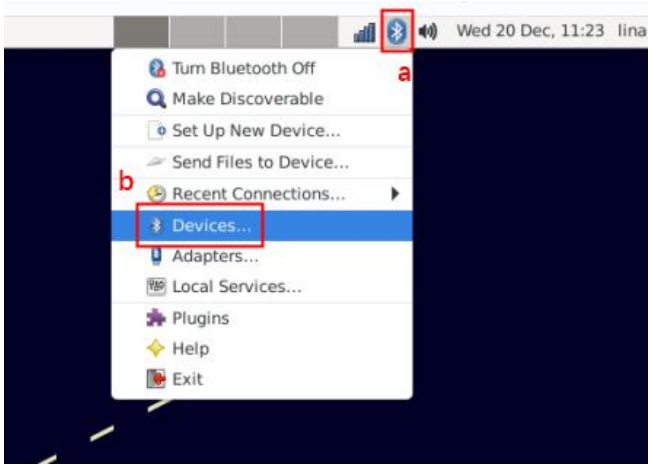
ifconfig //Check IP address

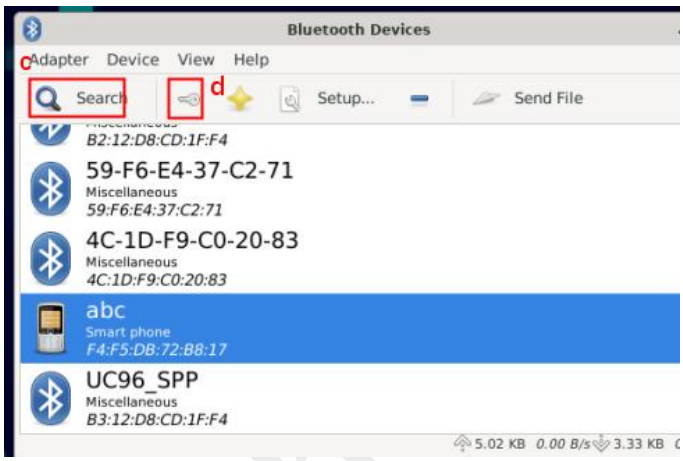


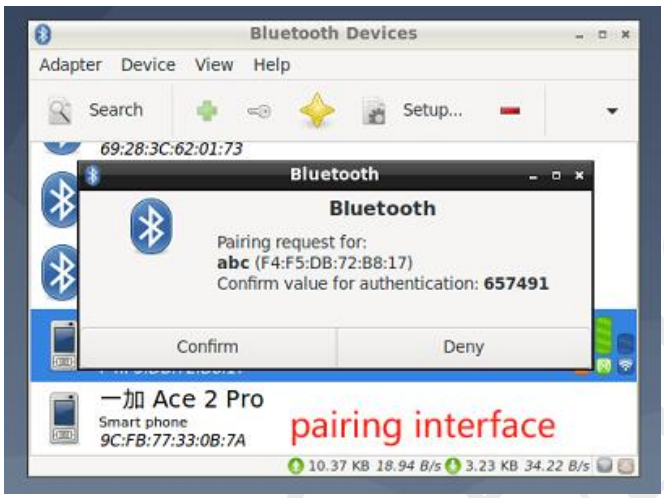
2.2.2 Bluetooth

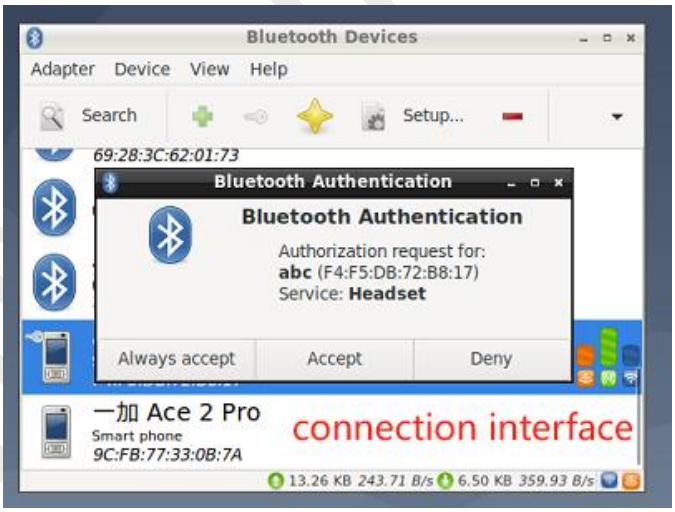
Bluetooth interface connection: a. Right click on the “Bluetooth” icon in the bottom right corner ->b. Select and click on “Devices” to enter the “Bluetooth Devices” interface ->c. Click on “Search” to search for Bluetooth devices in the environment ->d. Select and click on the “Key” icon to pair( ) ->e. Click on the paired board Bluetooth on the phone to

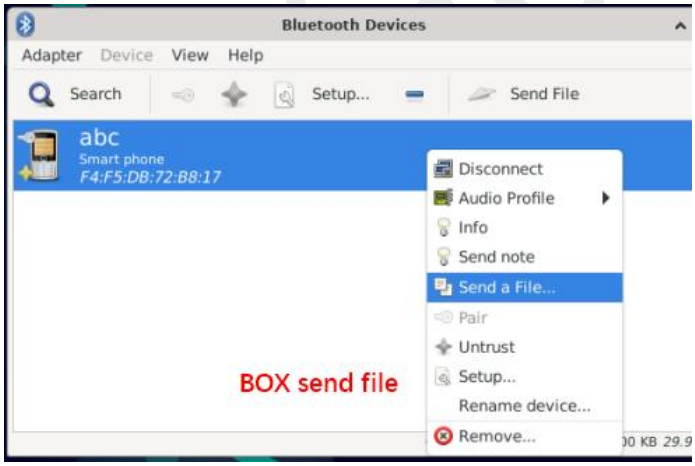
connect ->f. Right click on the connected Bluetooth device, select and click “Send a File” to send the file. As shown in the following figure:

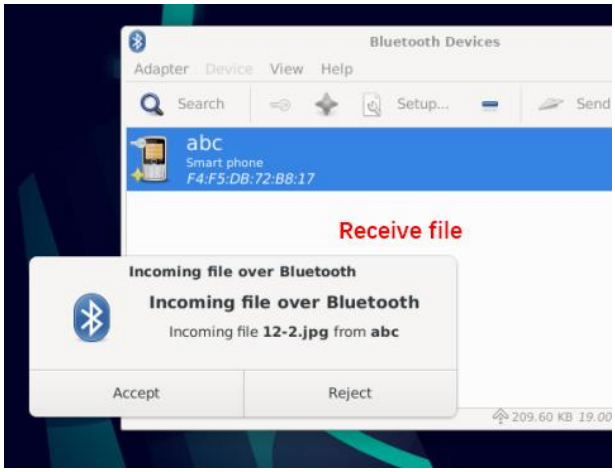








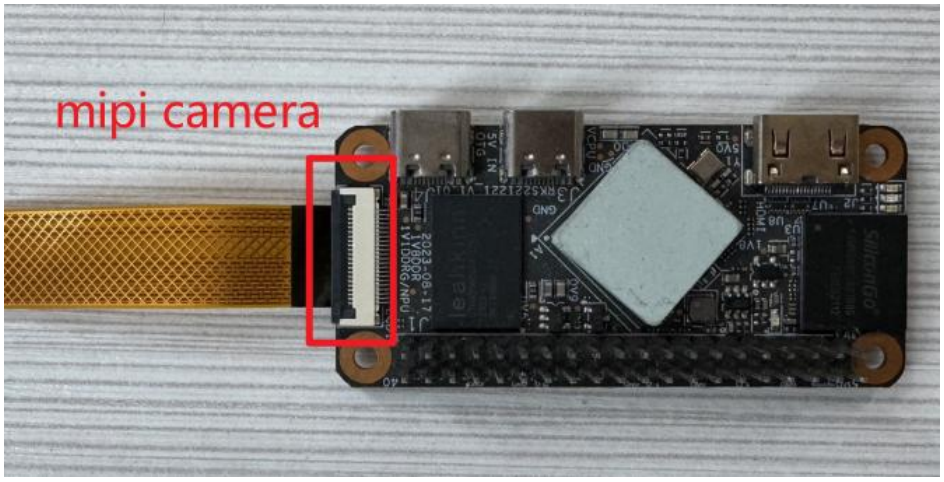




2.3 MIPI camera

2.3.1 Hardware Preparation:

Connect Camera as shown in the following figure:



2.3.2 Testing Procedure:

Execute the following command to open the camera screen:

v4l2-ctl --list-devices //View camera node

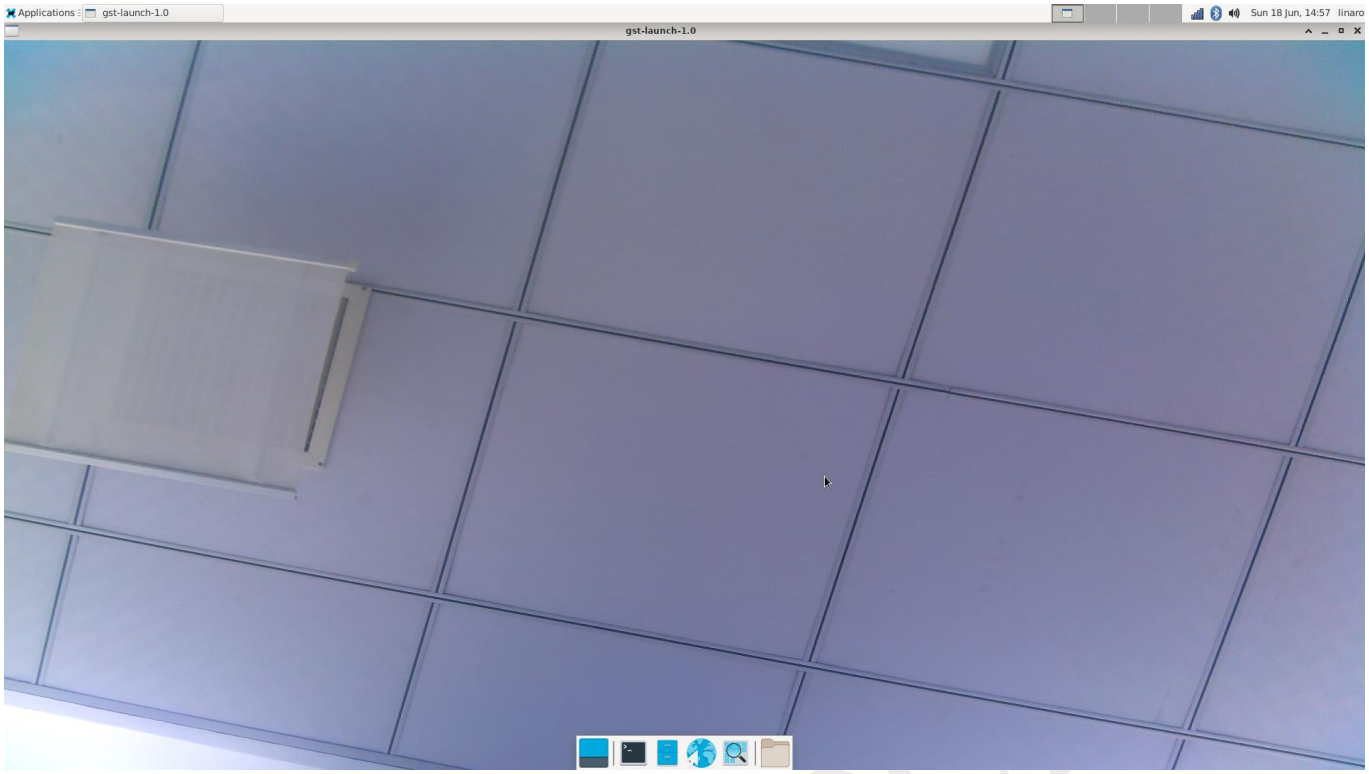
gst-launch-1.0 v4l2src io-mode=2 device=/dev/video0 !

video/x-raw,format=NV12,width=1920,height= 1080,framerate=30/1 !

xvimagesink //Open the camera screen

```
root@linaro-alip:/# v4l2-ctl --list-devices
rkisp-statistics (platform: rkisp):
        /dev/video7
        /dev/video8

rkisp_mainpath (platform:rkisp-vir0):
        /dev/video0
        /dev/video1
        /dev/video2
        /dev/video3
        /dev/video4
        /dev/video5
        /dev/video6
        /dev/media0

root@linaro-alip:/# gst-launch-1.0 v4l2src io-mode=2 device=/dev/video0 ! video/
x-raw,format=NV12,width=1920,height= 1080,framerate=30/1 ! xvimagesink
Setting pipeline to PAUSED ...
Using mplane plugin for capture
Pipeline is live and does not need PREROLL ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  505.560383] rkisp_hw fdff0000.rkisp: set isp clk = 297000000Hz
[  505.563855] rockchip-csi2-dphy1: dphy1, data_rate_mbps 300
[  505.564194] rockchip-csi2-dphy csi2-dphy1: csi2_dphy_s_stream stream on:1, dp
hy1, ret 0
[  505.604911] rkisp-vir0: MIPI error: packet: 0x00000010
0:00:09.8 / 99:99:99.
```

mipi camera node



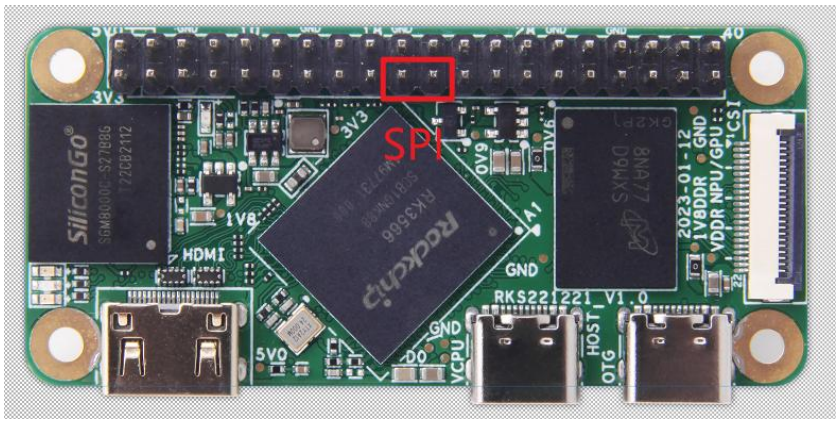
2.4 SPI

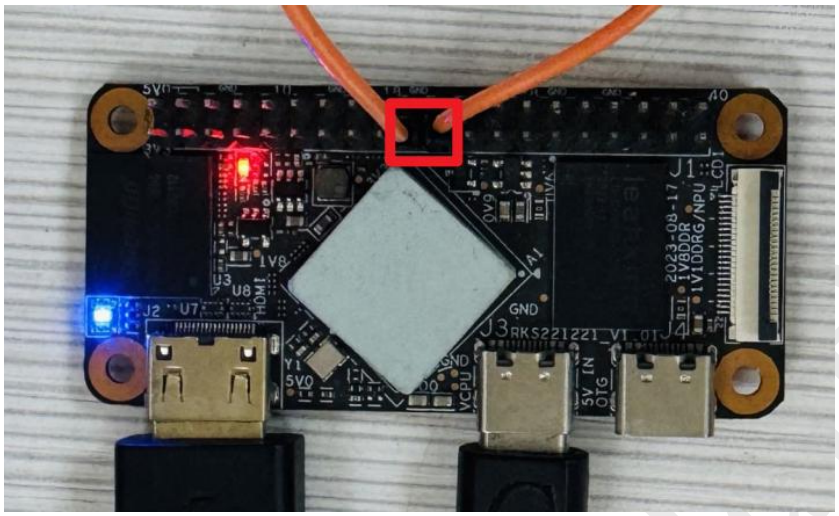
2.4.1 Preparation:

(1) View the configuration file of the SPI function in the system and copy the corresponding spidev_test.c file into the system (contact employee for spidev_test.c file).

```
root@linaro-alip:~#
root@linaro-alip:~# ls /dev/
block            loop-control  rfkill      tty3   tty60        vcsa1
bus              loop0         rga         tty30  tty61        vcsa2
cec0             loop1         rk_cec      tty31  tty62        vcsa3
char             loop2         shm         tty32  tty63        vcsa4
console          loop3         snd         tty33  tty7         vcsa5
cpu_dma_latency  loop4         spidev3.0   tty34  tty8         vcsa6
crypto           loop5         stderr      tty35  tty9         vcsa7
disk             loop6         stdin       tty36  ttyFIQ0      vcsu
dma_heap         loop7         stdout      tty37  ttyS1        vcsu1
dri              mali0         sw_sync     tty38  ttyS4        vcsu2
fb0              mapper        tty         tty39  ttyS9        vcsu3
fd               media0        tty0        tty4   ubi_ctrl     vcsu4
full             mem           tty1        tty40  uhid         vcsu5
fuse             mmcblk0       tty10       tty41  uinput       vcsu6
gpiochip0        mmcblk0boot0  tty11       tty42  urandom      vcsu7
gpiochip1        mmcblk0boot1  tty12       tty43  usb-ffs      vendor_storage
gpiochip2        mmcblk0p1     tty13       tty44  usbmon0      vhci
gpiochip3        mmcblk0p2     tty14       tty45  usbmon1      video-camera0
gpiochip4        mmcblk0p3     tty15       tty46  usbmon2      video-dec0
hdmi_hdcp1x      mmcblk0p4     tty16       tty47  usbmon3      video-enc0
hidraw0          mmcblk0p5     tty17       tty48  usbmon4      video0
hidraw1          mmcblk0p6     tty18       tty49  v4l          video1
hidraw2          mmcblk0p7     tty19       tty5   v4l-subdev0  video2
hwrng            mmcblk0p8     tty2        tty50  v4l-subdev1  video3
i2c-0            mmcblk0rpmb   tty20       tty51  v4l-subdev2  video4
i2c-1            mpp_service   tty21       tty52  vcs          video5
i2c-2            mqueue        tty22       tty53  vcs1         video6
i2c-3            net           tty23       tty54  vcs2         video7
i2c-6            null          tty24       tty55  vcs3         video8
iio:device0      port          tty25       tty56  vcs4         watchdog
initctl          ptmx          tty26       tty57  vcs5         watchdog0
input            pts           tty27       tty58  vcs6         zero
kmsg             ram0          tty28       tty59  vcs7         zram0
log              random        tty29       tty6   vcsa
root@linaro-alip:~# ls /home/linaro/
Desktop  spidev_test.c
root@linaro-alip:~#
```

(2)Short circuit pins 19 and 21, as shown in the following diagram:





2.4.2 Terminal Operation:

Execute the following command to verify whether SPI function is normal:

gcc spidev_test.c -o spidev_test //Load spidev

vi spi.txt //Create files, customize content

./spidev_test -l -i spi.txt -o out.txt

cat out.txt //Check that the file content matches the content of the created spi.txt file

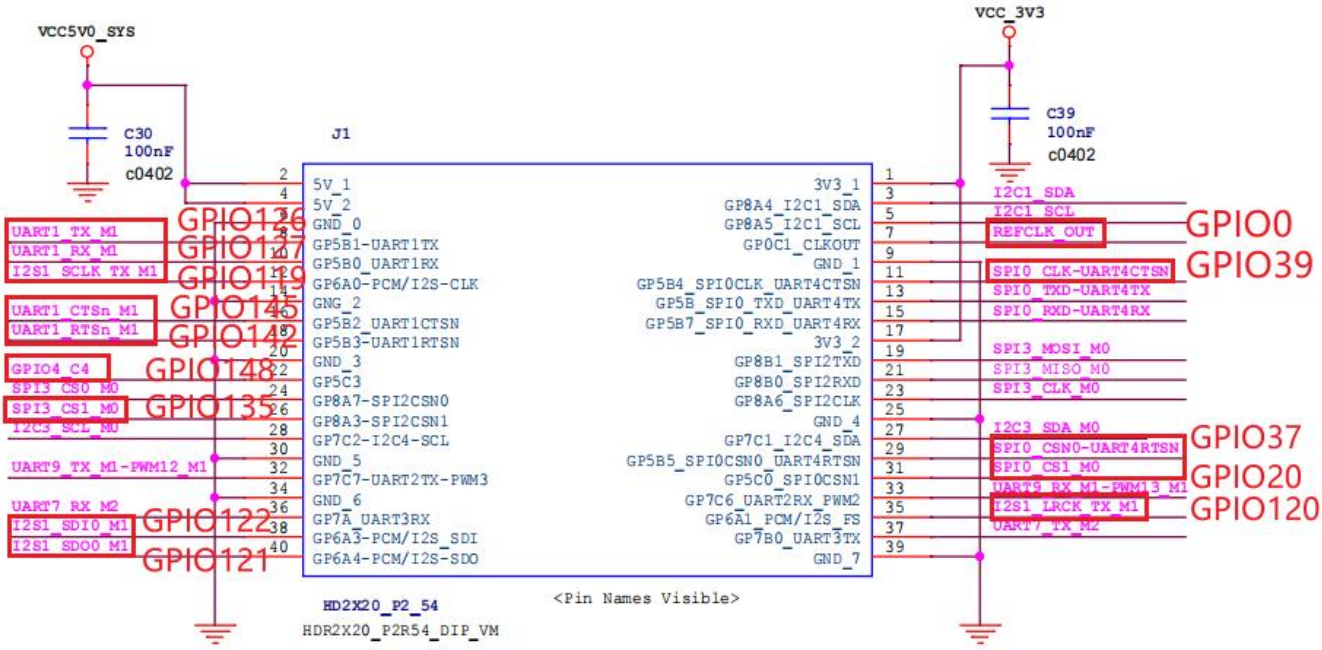
Usage of Vi Editor: Press "i" to enter editing mode, start inputting characters, press "Esc" to return to command mode, enter ": wq" to save the modified file and exit, or ": q!" to exit without saving the file.

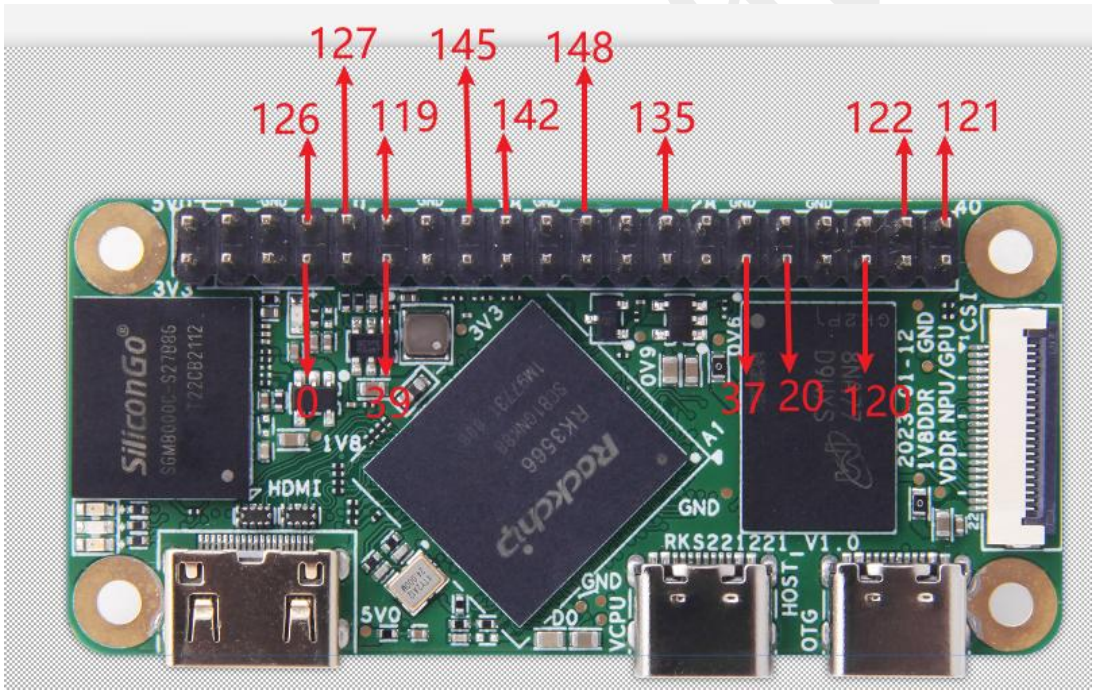
```
root@linaro-alip:/home/linaro# ./spidev_test -l -i spi.txt -o out.txt
spi mode: 0x24
bits per word: 8
max speed: 500000 Hz (500 KHz)
root@linaro-alip:/home/linaro# cat out.txt
geniatech2024
root@linaro-alip:/home/linaro#
```

2.5 GPIO*14

2.5.1 Hardware Preparation:

GPIO pin schematic and actual pin conversion.





2.5.2 Testing Procedure:

Manually execute instructions to raise and lower the voltage of the GPIO pins; Multimeter test voltage.

echo 0 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio0/direction

echo 1 >/sys/class/gpio/gpio0/value //raise the voltage to 3.4V

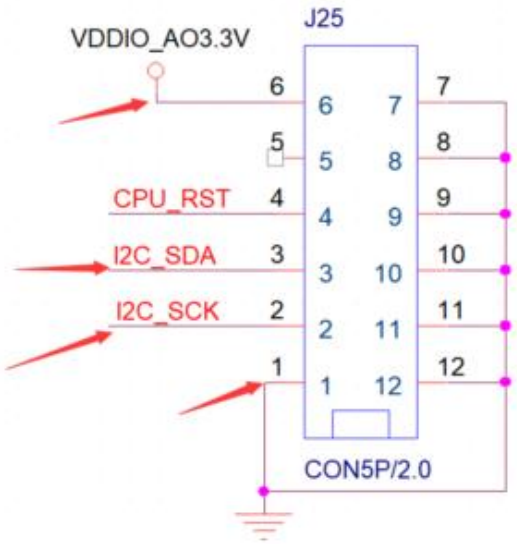
echo 0 >/sys/class/gpio/gpio0/value //lower the voltage to 0V

Note: The verification method for GPIO39/GPIO37/GPIO20/GPIO120/GPIO126/GPIO127/GPIO119/GPIO145/GPIO142/GPIO148/GPIO135/GPIO122/GPIO121 is the same as above.

2.6 I2C*2

2.6.1 Preconditions:

- I2C equipment
- PC (supports 64 bit)
- Serial port tool (putty)
- 12V power cord

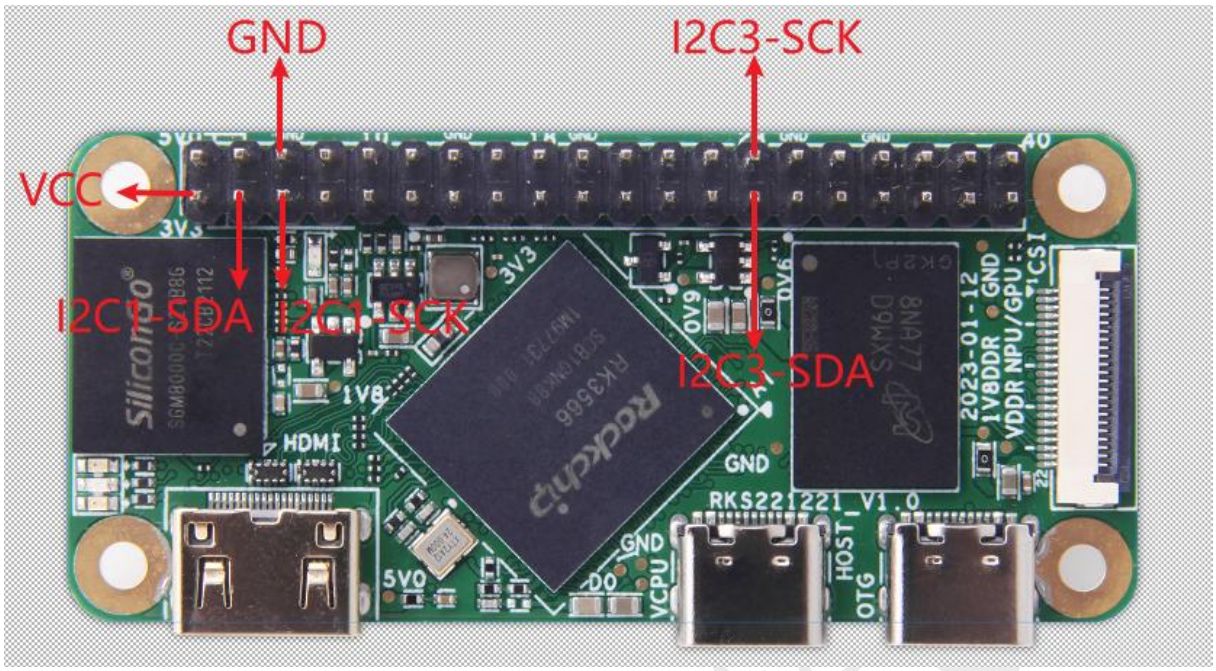


I2C device 3.3V/GND/SDA/SCK are respectively connected to the 3.3V/GND/SDA/SCK pins on the board.

Note: The current I2C device is only suitable for our company's verification use.

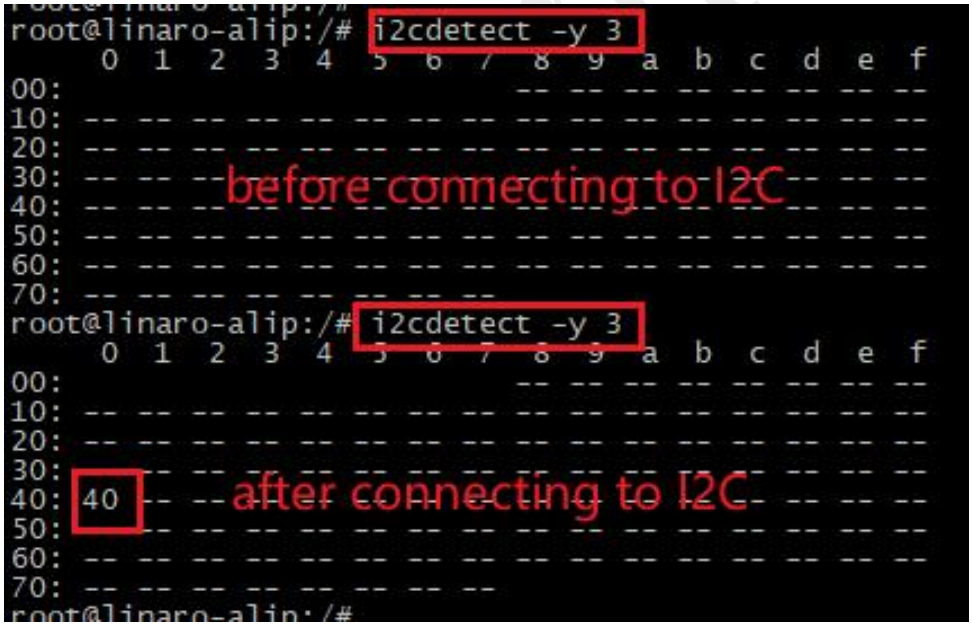
2.6.2 Terminal Operation:

(1) Connect as shown in the figure below and open the Debug port on the PC.

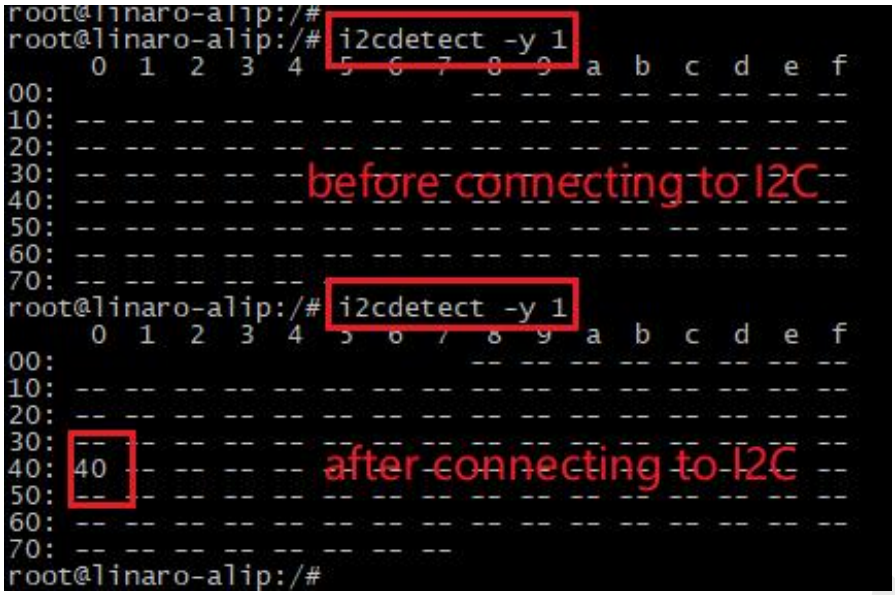


(2) By entering a command in the Debug port, you can identify an I2C successfully mounted on the I2C bus.

i2cdetect -y 3 //view device addresses under I2C3



i2cdetect -y 1 //view device addresses under I2C1



2.7 UART*2

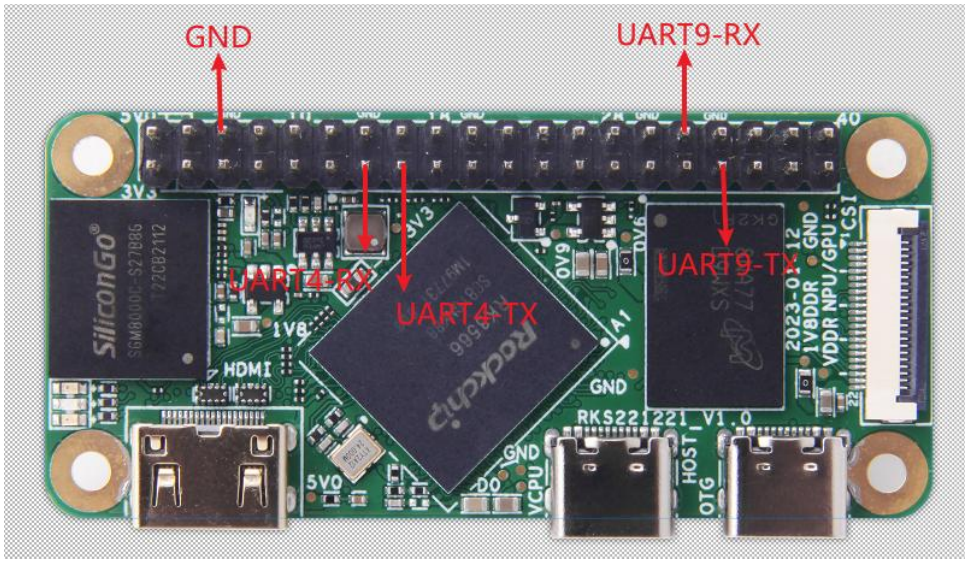
2.7.1 Preconditions:

- Serial port board
- PC (supports 64 bit)
- Serial port tool (putty)
- 12V power cord

The serial port tool on the computer has detected a serial port board. Connect the RXD/TXD/GND pins of the serial port board to the corresponding RXD/TXD/GND pins on the board. Please select the corresponding port number, Bps/Par/Bits: 115200 8N1. (Baud rate can be switched, but please keep the sender and receiver settings the same)

2.7.2 Terminal operation:

(1) Connect as shown in the figure below, and open the debugging windows for DEBUG port and UART on the PC side respectively.



(2) After networking the board and installing the Minicom, enter instructions in the Debug port, set the port number (ttyS4) and baud rate, and save them. Test that the two debugging windows can send and receive data from each other.

apt-get update //Update data source

apt-get install minicom //Install minicom

minicom -s //Set port number and baud rate

Minicom usage method: Use the up, down, left, and right keys to enter different options. In the serial port setup, you can set the port number and baud rate by changing the front letters in the options (A: change port number, E: change baud rate,etc.)

```
+-----[configuration]------+
| Filenames and paths      |
| File transfer protocols  |
| Serial port setup        |
| Modem and dialing        |
| Screen and keyboard      |
| Save setup as dfl        |
| Save setup as..          |
| Exit                     |
| Exit from Minicom        |
+--------------------------+
```

```
| A -    Serial Device      : /dev/ttyS4
| B - Lockfile Location     : /var/lock
| C -   Callin Program      :
| D -  Callout Program      :
| E -    Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
| H -     RS485 Enable      : No
| I -   RS485 Rts On Send   : No
| J -  RS485 Rts After Send : No
| K -  RS485 Rx During Tx   : No
| L -  RS485 Terminate Bus  : No
| M - RS485 Delay Rts Before: 0
| N - RS485 Delay Rts After : 0

|    Change which setting?
```

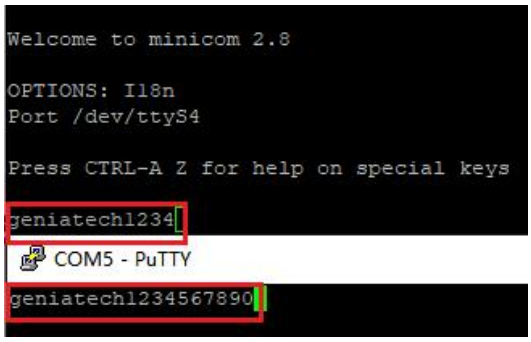
After setting up

Save setup as df1

Exit

Entering the receive command window will display the terminal sending data.

For example, open the COM port corresponding to UART, enter any value (such as geniatech1234567890) in the serial port interface, and the COM port corresponding to UART will receive geniatech1234567890; Enter any value (such as geniatech1234) into the UART COM port, and the serial port should receive geniatech1234.



Note:The verification method for UART9 (ttyS9) is the same as above.

2.8 Watch dog

Execute the watchdog test using the following command. The system will automatically restart approximately 45 seconds after stopping feeding the dog.

echo V > /dev/watchdog //Continue feeding the dog

echo A > /dev/watchdog //Stop feeding the dog