GTC-XPI3576 Debian12 Software Usage Guide

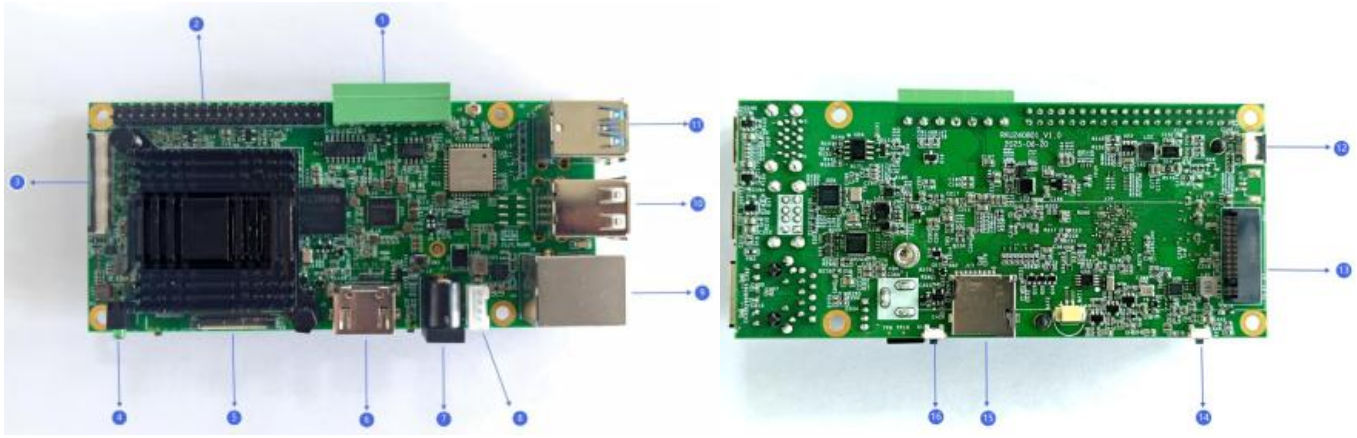
V1.0

Catalog

REVISION HISTORY

Time	Version	Purpose	Author	Comment
26/09/2025	1.0	Create a document	YYT	

1.Product interface definition:

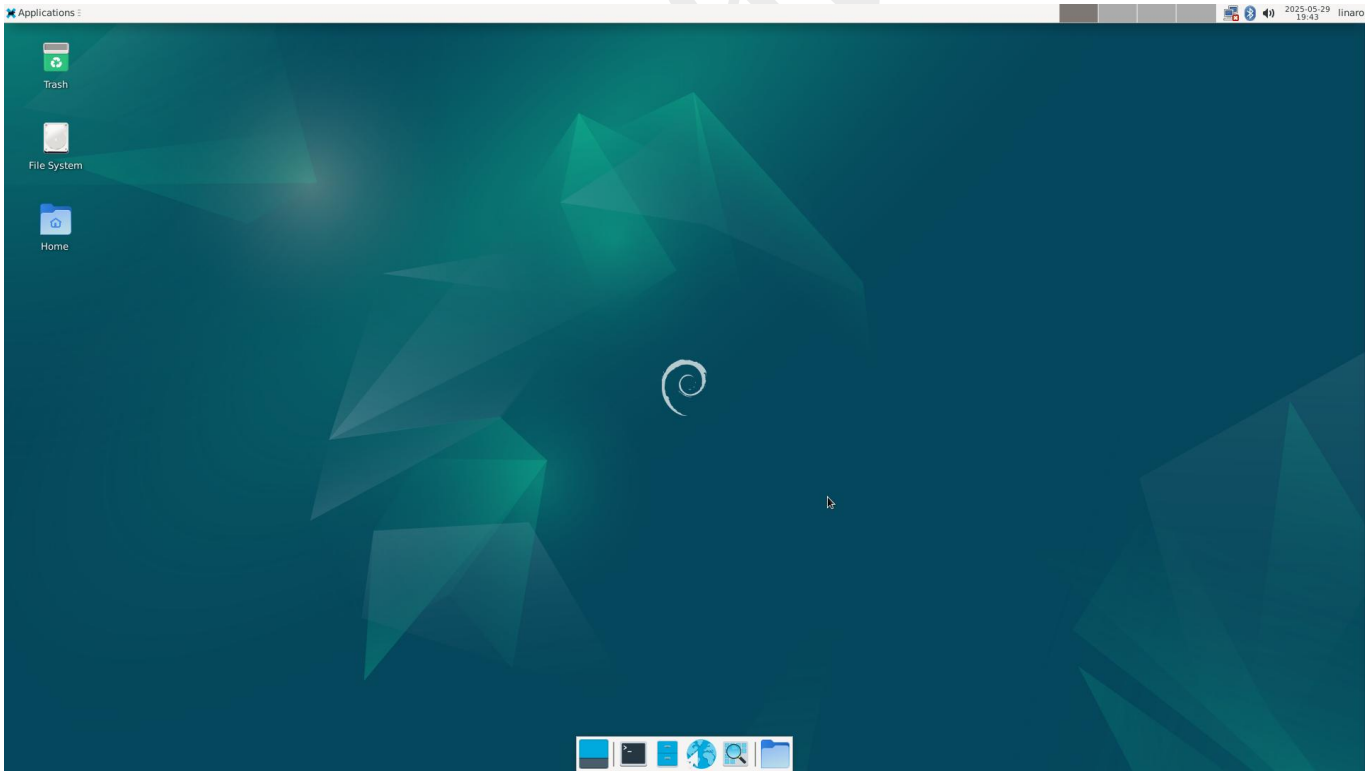


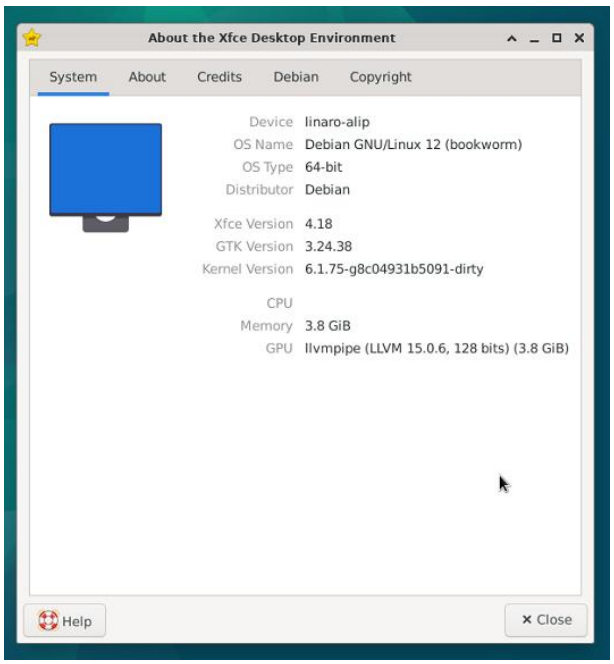
Serial number	Features	Notes
1	15 EDGRV-7p 3.81mm	1×CAN, 1× 485,1 ×232
2	GPIO	40PIN Expand I/O, 2.54mm interface
3	MIPI_DSI	4LANE DSI FPC (0.5mm FPC seat)
4	LED	5V power indicator light
5	MIPI_CSI	4LANE CSI FPC (0.5mm FPC seat)
6	HDMI OUT	HDMI V2.1
7	DC	12V/2A DC Power input
8	UART interface	DEBUG

9	RJ45	1×100M/1000M Ethernet (with POE mode)
10	USB 2.0	2×USB 2.0
11	USB 3.0	2×USB 3.0
12	FPC 6P	Touch Panel connector
13	M.2	A computing power module with Hailo H8
14	ADC	Recovery button
15	TF Card slot	TF Card Socket
16	Power key	Standby, wake switch

2.XPI3576 Debian12 OS

XPI3576 Supported OS: Debian GNU/Linux 12 \n \l



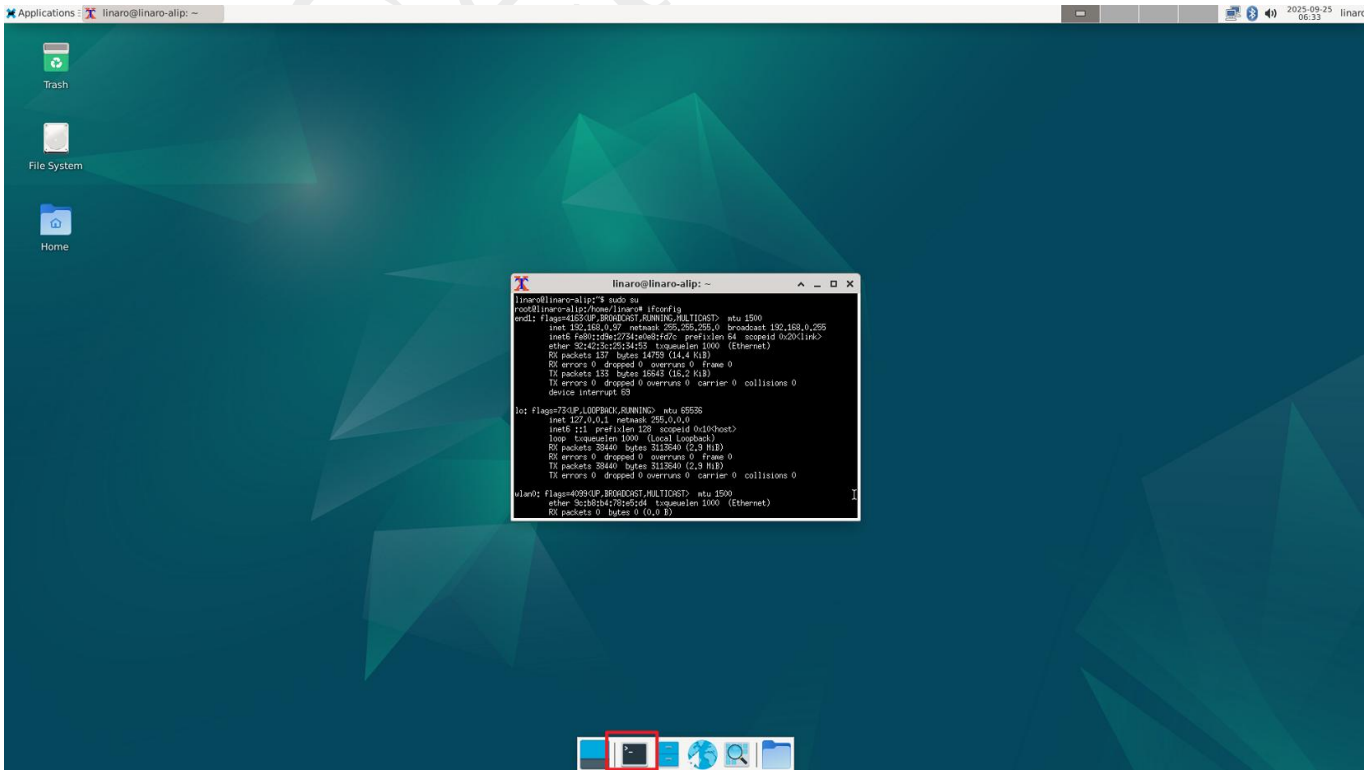


3.Function Description

3.1 How to access the OS

3.1.1 HDMI Display

XPI3576 support HDMI OUT display, default resolution 1920x1080;It can be connected to the monitor and wait for the system start up, connecting the keyboard and mouse, open the terminal emulator that comes with the system:



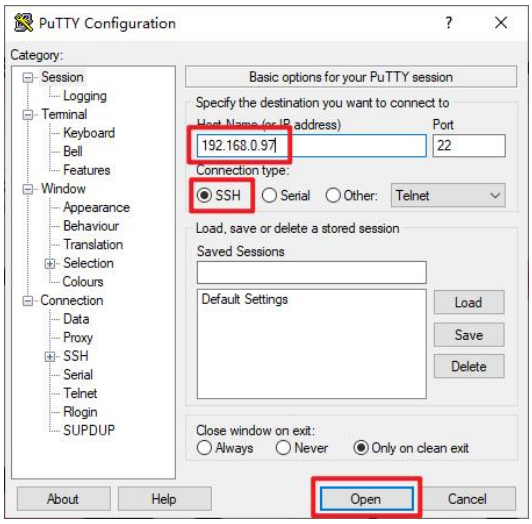
3.1.2 Remote Connection

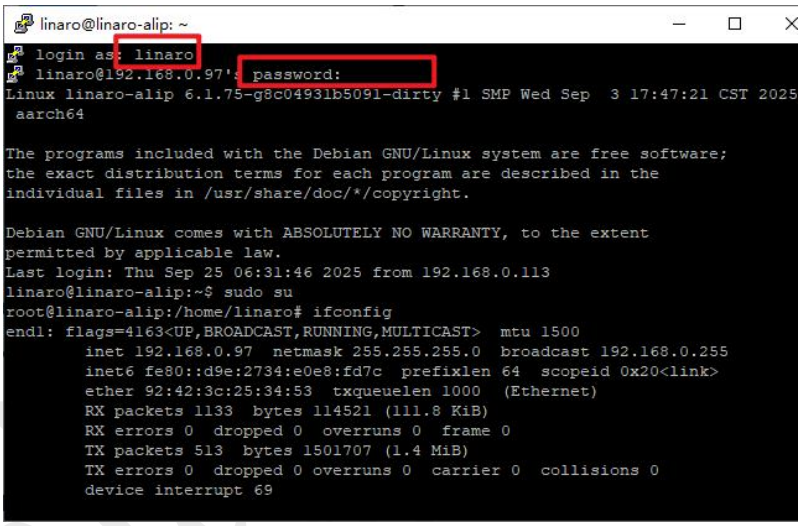
Connect the PC and XPI3576 in a local area network, get the IP address and connect via the serial tool (putty.exe) using SSH.

Protocol : SSH2

User name : linaro

Password : linaro





3.1.3 Serial Access

(1) For serial port access through USB to UART, connect the USB port to the computer, and connect the other end to the serial port conversion board.

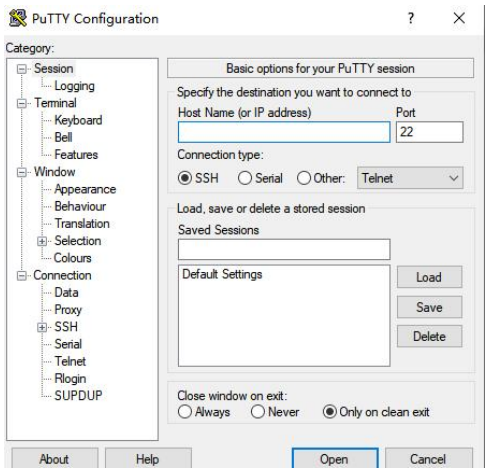
DEBUG (J12)

Table 5 J12 interface definition

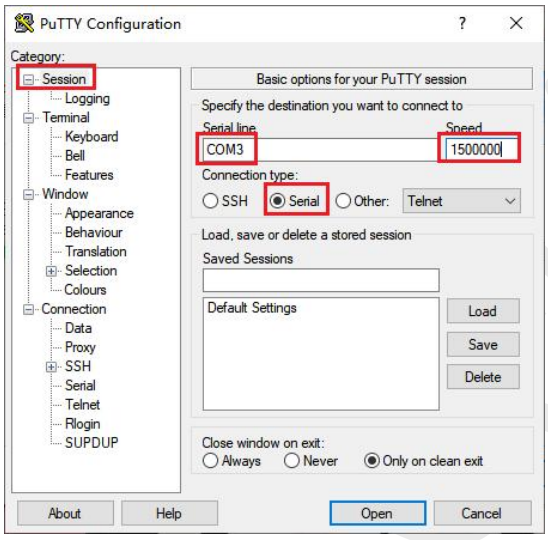
Serial number	Features	Features	Description	Illustration
1	RX	Input	UART0_RX_M0_DEBUG	
2	GND	Ground	Ground	
3	TX	Output	UART0_TX_M0_DEBUG	
4	VCC	/	empty	

(2) IN the Device Manager of your computer, view the serial port number.

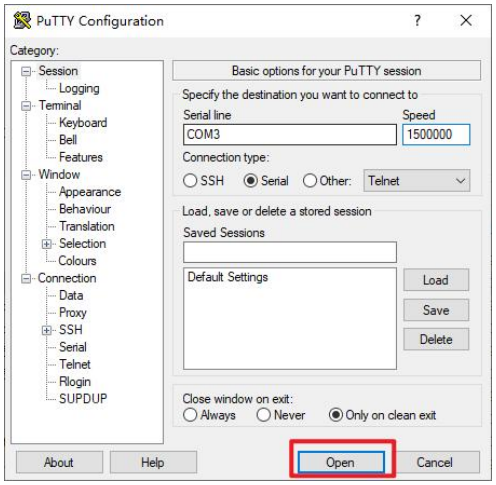
(3) Double-click putty.exe to open the serial port tool.

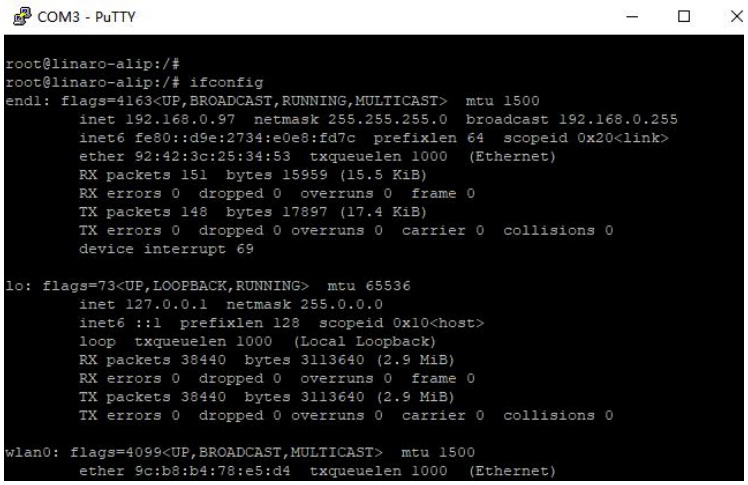


(4) Click “Session”, select the Serial port “serial”, after the computer is connected to the DEBUG port, select the new serial port name COMx, and set the corresponding serial port baud rate (1500000).



(5) Click “Open” to start the tool.



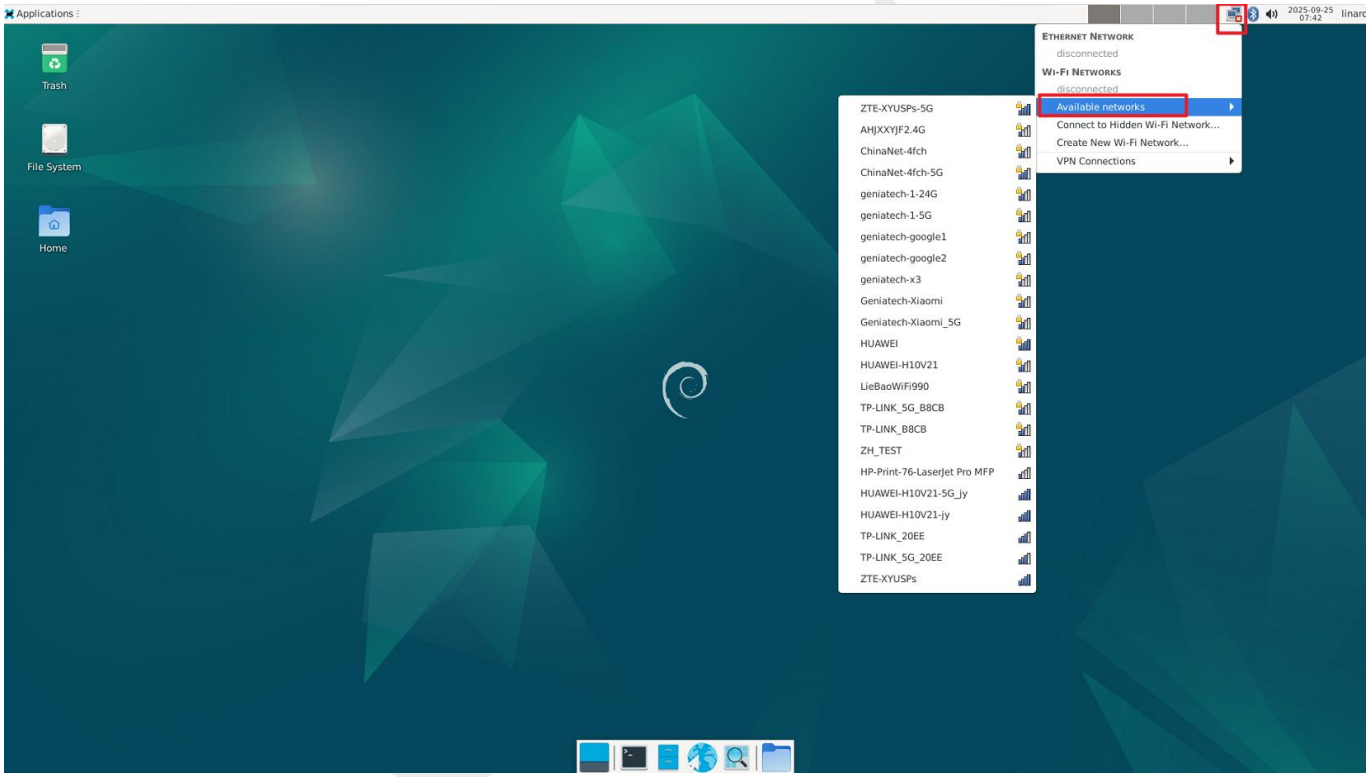


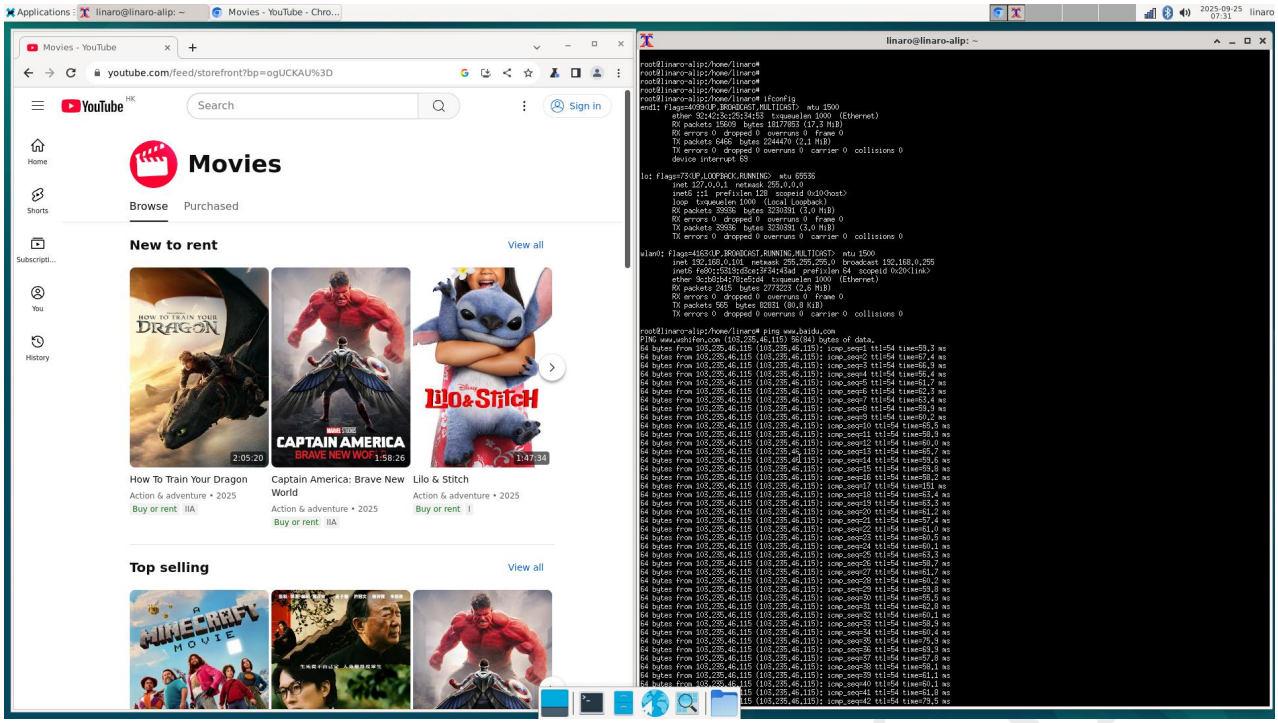
3.2 Network Function

3.2.1 WIFI

WiFi interface connection: click “Network status” icon  in the upper right corner -> click “Available networks” -> Display the searched WiFi -> Enter password/direct connection; Execute instructions, check the obtained IP address, verify that the network is normal by ping Baidu or entering a browser webpage.

Check the IP address: **ifconfig**

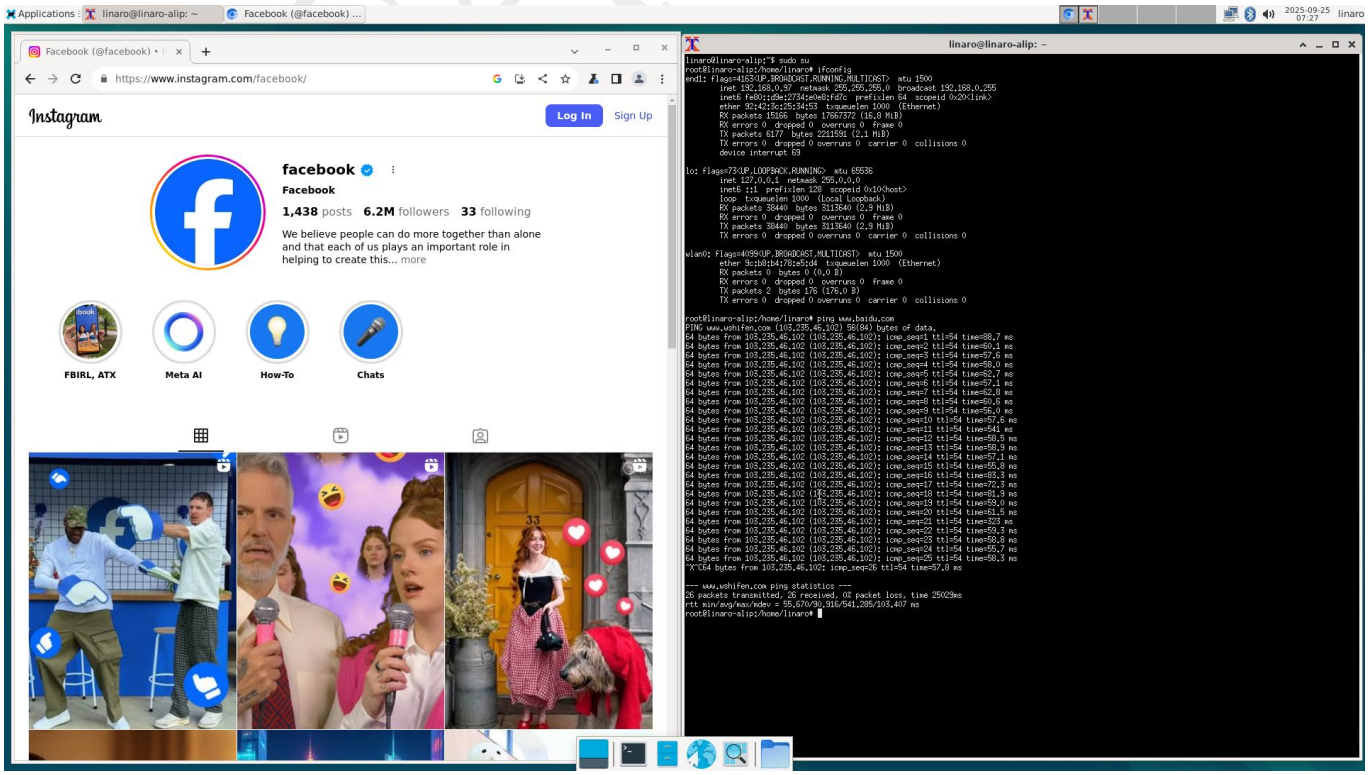




3.2.2 Wired Network

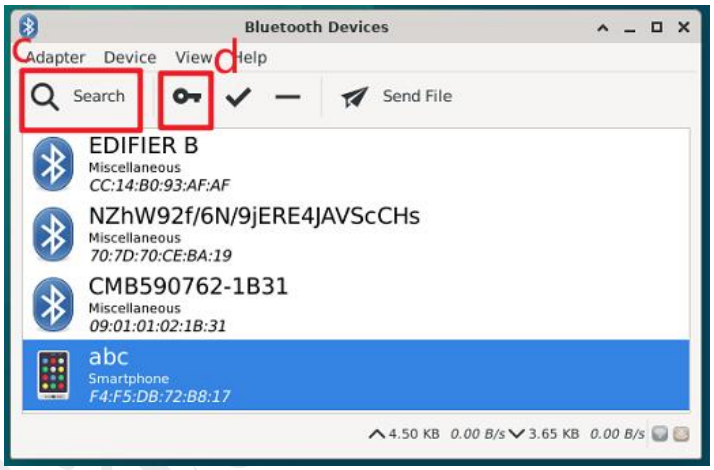
Wired connection: After connecting to the network cable, execute instructions, check the obtained IP address, verify that the network is normal by ping Baidu or entering a browser webpage.

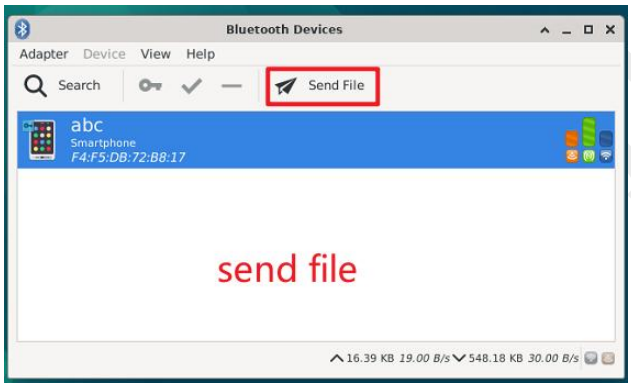
Check the IP address: **ifconfig**

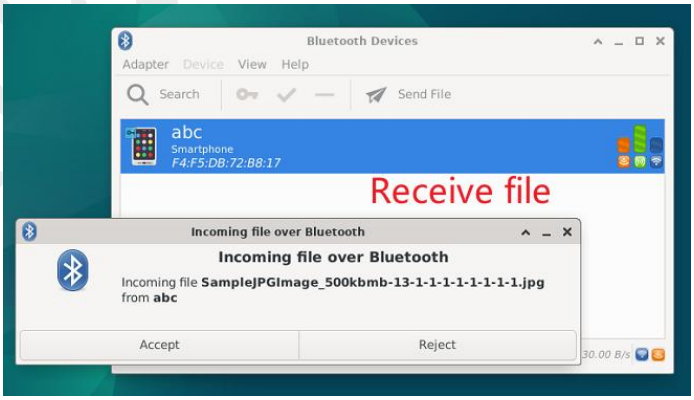


3.2.3 Bluetooth

Bluetooth interface connection: a. Click on the “Bluetooth” icon in the upper right corner -> b. Select and click “Devices...”, enter the “Bluetooth Devices” interface -> c. Click “Search”, 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 mobile phone to confirm pairing and connect -> f. Click “Send File” to send the file.



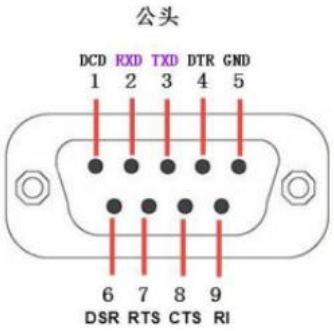




3.3 RS232

3.3.1 Preconditions:

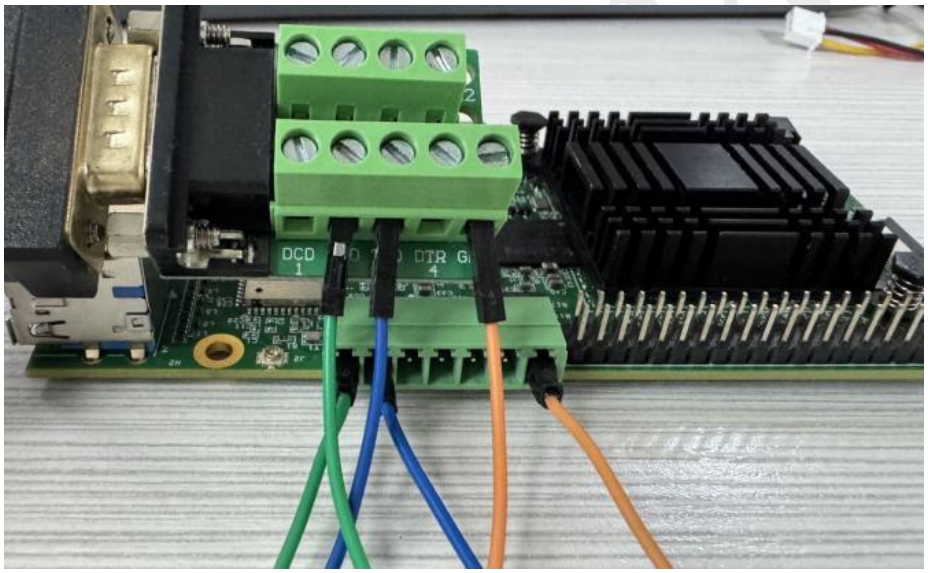
- USB to RS232 cable
- Host PC (supports 64 bit)
- Serial port tool (putty)
- 12V power cord



After the PC detects the connected USB to RS232 cable, connect the 2/3/5 pins of RS232 to the corresponding places on the board. Select the corresponding port number and baud rate (the baud rate can be switched, but please keep the Settings of the sending and receiving ends the same).

3.3.2 Terminal operation:

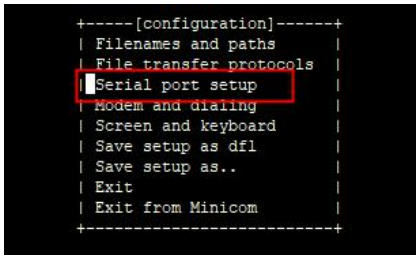
(1) Connect as shown in the figure below, and open the RS232 debugging window and DEBUG port on the PC side respectively.

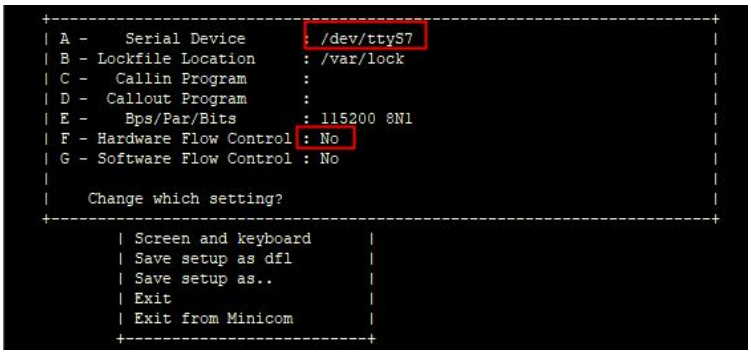


(2) Enter commands in the DEBUG port, set the port number (ttyS7) and baud rate, and save them. Test that the two debugging windows can exchange data with each other.

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.)





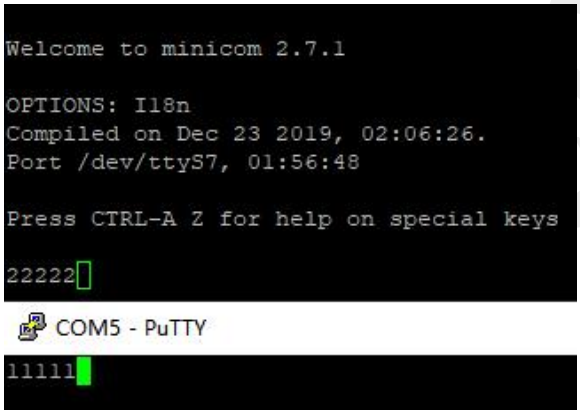
After setting up

Save setup as dfl

Exit

(3) Enter the DEBUG port and the RS232 debugging window to send and receive data to each other.

For example, enter any value (such as 11111) in the minicom interface of the DEBUG port, and the RS232 debugging window will receive 11111; Input any value (such as 22222) into the RS232 debugging window, and the DEBUG port should receive 22222.

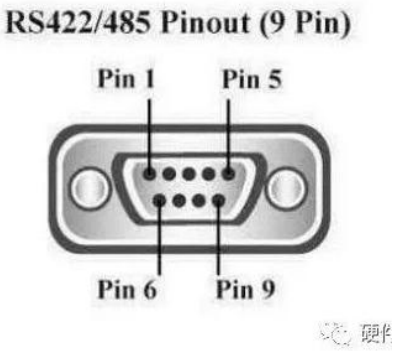


3.4 RS485

3.4.1 Preconditions:

- USB to RS485 cable
- Host PC (supports 64 bit)
- Serial port tool (putty)
- 12V power cord

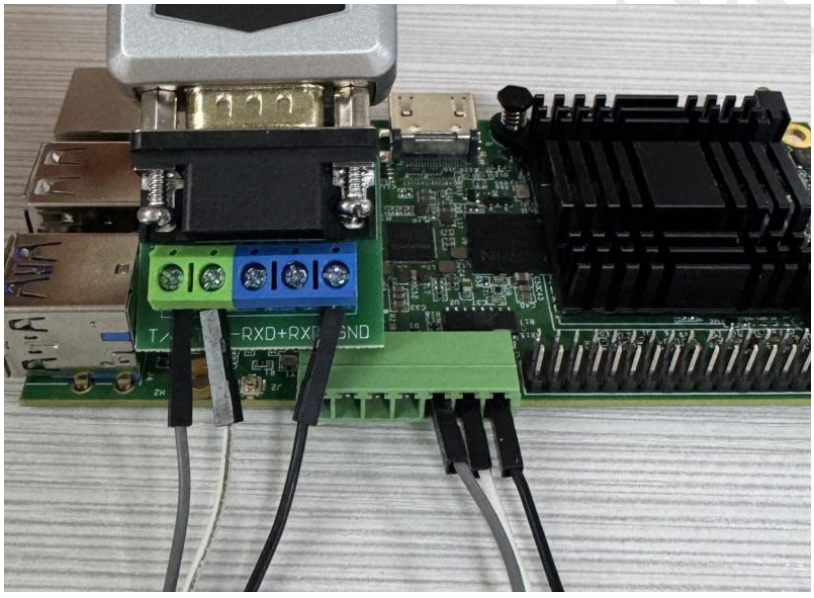
Pin 1	TXD-
Pin 2	TXD+
Pin 3	RTS-
Pin 4	RTS+
Pin 5	GND
Pin 6	RXD-
Pin 7	RXD+
Pin 8	CTS
Pin 9	CTS+



After the PC detects the connected USB to RS485 cable, connect the 1/2/5 pins of RS485 to the corresponding places on the board. Select the corresponding port number and baud rate (the baud rate can be switched, but please keep the Settings of the sending and receiving ends the same).

3.4.2 Terminal operation:

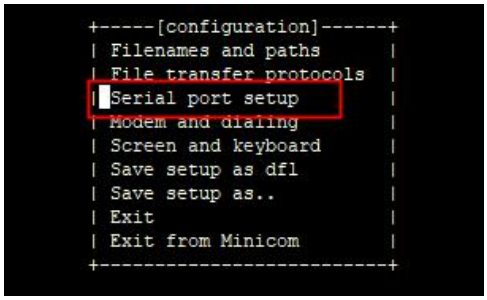
(1) Connect as shown in the figure below, and open the RS485 debugging window and DEBUG port on the PC side respectively.

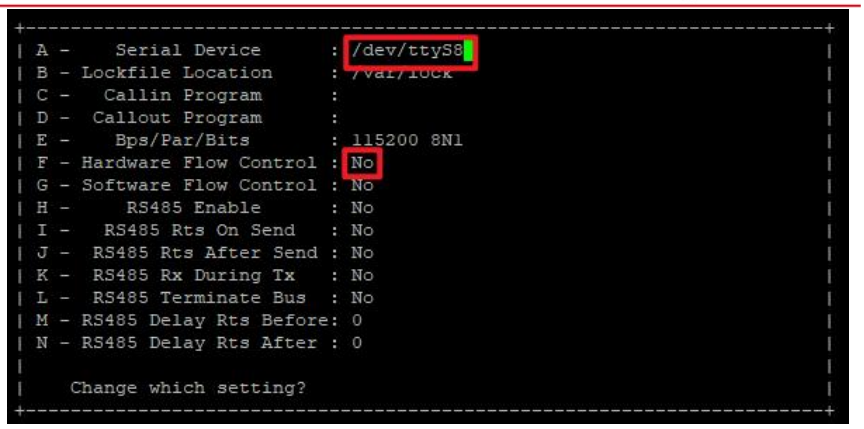


(2) Enter commands in the DEBUG port, set the port number (ttyS8) and baud rate, and save them. Test that the two debugging windows can exchange data with each other.

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.)





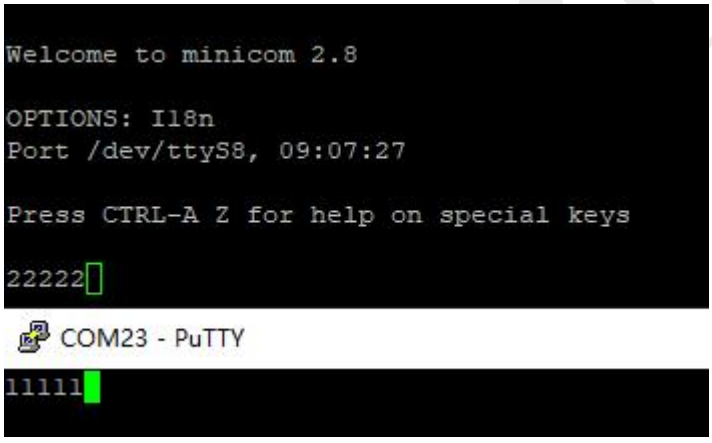
After setting up

Save setup as dfl

Exit

(3) Enter the DEBUG port and the RS485 debugging window to send and receive data to each other.

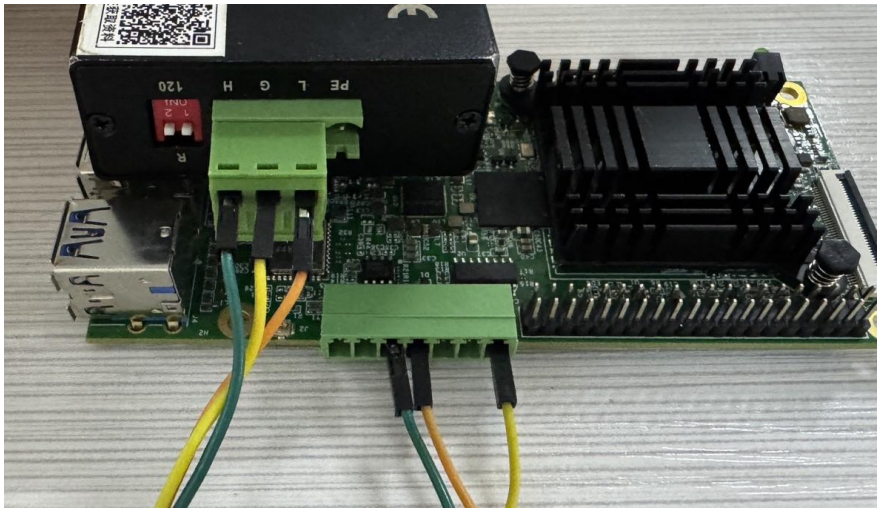
For example, enter any value (such as 11111) in the minicom interface of the DEBUG port, and the RS485 debugging window will receive 11111; Input any value (such as 22222) into the RS485 debugging window, and the DEBUG port should receive 22222.



3.5 CAN

3.5.1 Hardware connection

Connect the PC to the can0 port of the board using the CAN test box. The hardware connection is as shown in the following figure:



3.5.2 Terminal operation

(1) Execute command in serial port to check if can0 node has been recognized.

ifconfig -a

```
root@linaro-alip:/# ifconfig -a
can0: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (
UNSPEC)
        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
        device interrupt 68

end1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 92:42:3c:25:34:53  txqueuelen 1000  (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
        device interrupt 69

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 16  bytes 1296 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1296 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

(2) Execute the following command to configure the can0 node.

ip link set can0 type can bitrate 1000000 dbitrate 1000000 fd on

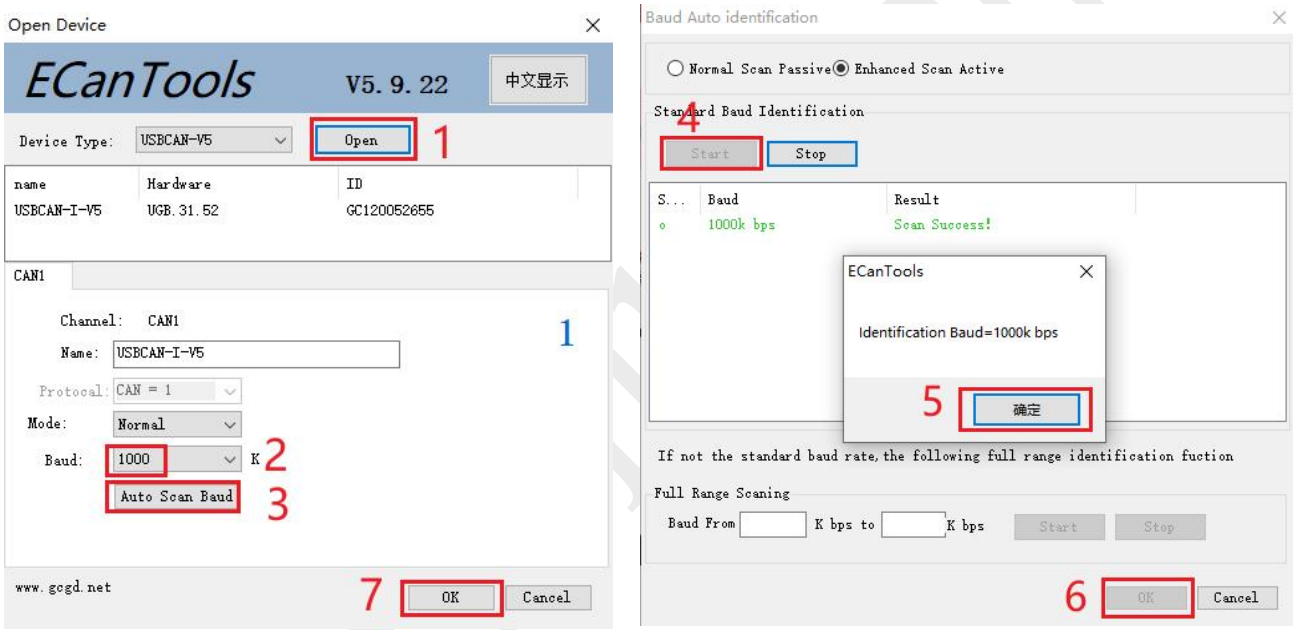
ip -details link show can0

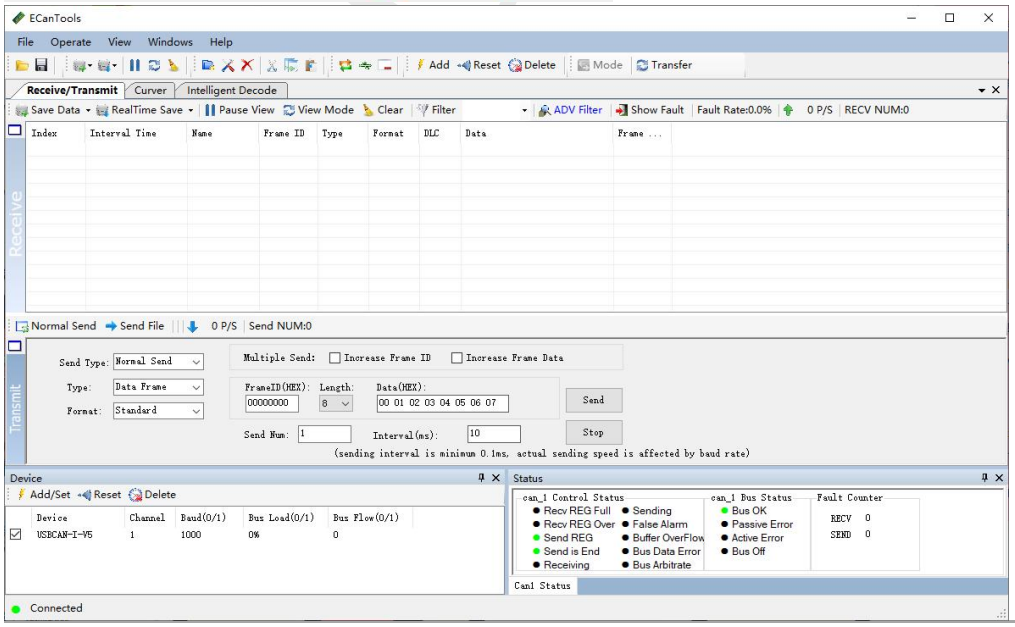
ip link set can0 up

ifconfig

```
root@linaro-alip:/#
p link set can0 type can bitrate 1000000 dbitrate 1000000 fd onbitrate 1000000 fd on
[ 1146.174874] rk3576_canfd 2ac00000.can can0: bitrate error 0.3%
[ 1146.174955] rk3576_canfd 2ac00000.can can0: bitrate error 0.3%
root@linaro-alip:/#
root@linaro-alip:/# ip -details link show can0
2: can0: <NOARP,ECHO> mtu 72 qdisc pfifo_fast state DOWN mode DEFAULT group default qlen 10
    link/can  promiscuity 0  allmulti 0 minmtu 0 maxmtu 0
    can <FD> state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 996644 sample-point 0.744
          tq 6 prop-seg 55 phase-seg1 55 phase-seg2 38 sjw 1 brp 2
          rk3576_canfd: tseg1 1..128 tseg2 1..128 sjw 1..128 brp 1..256 brp_inc 2
          dbitrate 1003378 dsample-point 0.750
          dtq 249 dprop-seg 1 dphase-seg1 1 dphase-seg2 1 dsjw 1 dbrp 74
          rk3576_canfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp_inc 2
          clock 297000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_s
ize 65536 tso_max_segs 65535 gro_max_size 65536 parentbus platform parentdev 2ac00000.can
root@linaro-alip:/#
root@linaro-alip:/# ip link set can0 up
root@linaro-alip:/#
root@linaro-alip:/# ifconfig
can0: flags=193<UP,RUNNING,NOARP>  mtu 72
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 28  bytes 28 (28.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

(3) Open ECanTools on PC, click to open the device and follow the steps below to automatically recognize the baud rate.

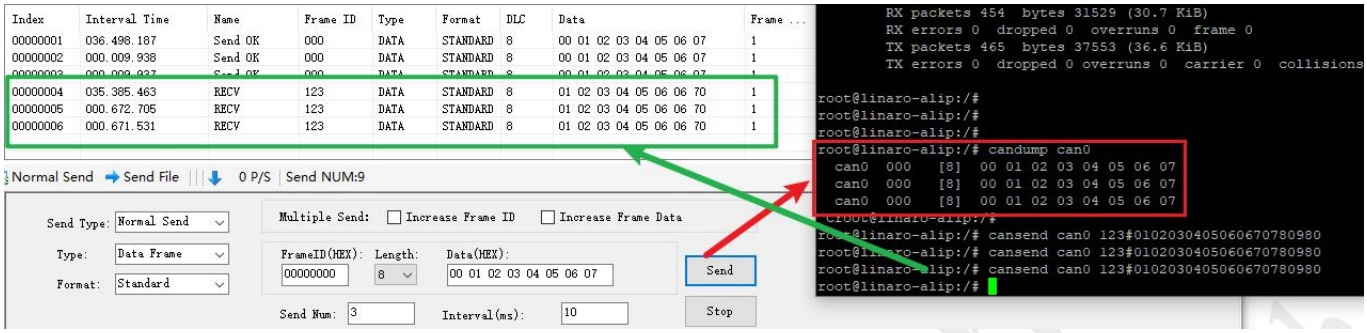




(4) After the ECanTools settings are completed, the following command is executed in the serial port, and can0 sends and receives data as shown in the following figure:

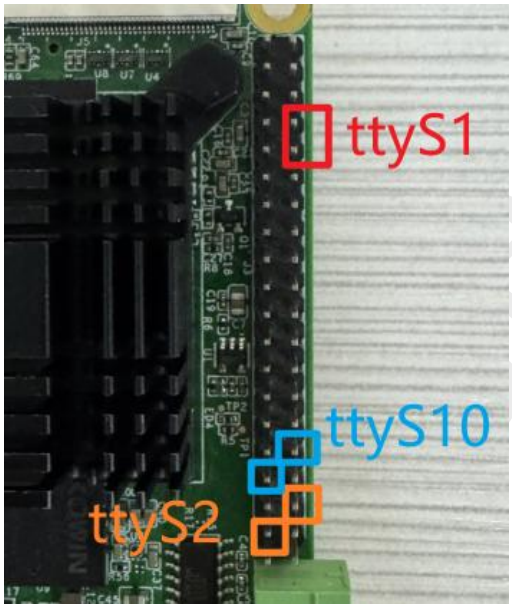
Received: **candump can0**

Send: **cansend can0 123#01020304050606707809 80**



3.6 UART*3

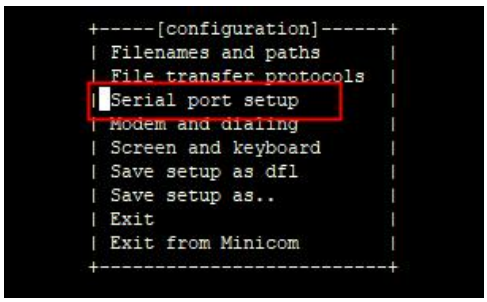
(1) The schematic diagram of the UART pin with port numbers ttyS1/ttyS2/ttyS10 is shown as follows:

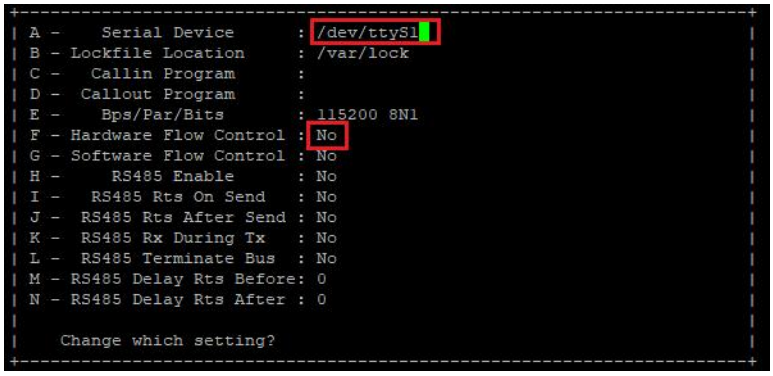


(2) Enter the command in the DEBUG port, set the port number (ttyS1) and baud rate, and save.

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.)



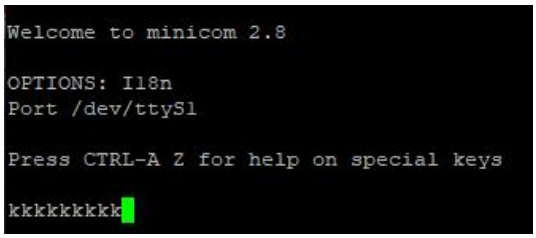


After setting up

Save setup as dfl

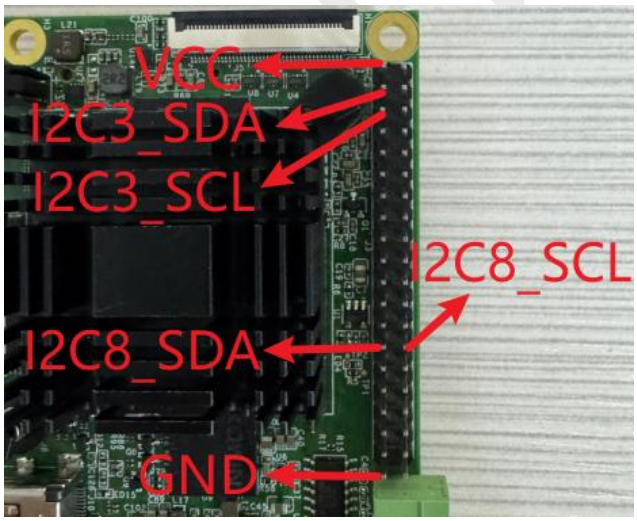
Exit

(3) Short-circuit the two pins to enter the minicom interface of the DEBUG port, send data normally and receive the same data back.



3.7 I2C*3

(1) The pin diagram of I2C in J3 and the connection diagram of I2C in J14 are shown as follows:



Note: The CLK/SDA/GND/VCC pins of the 3.3V I2C device correspond respectively to

the SDA/CLK/GND/VCC pins on the connection board card. The I2C function with bit J14 is verified through the adapter board.

(2) By entering commands in the debug port, the I2C mounting success can be recognized on the I2C bus.

Not connected to the I2C device, execute the command: **i2cdetect -y 3**

```
root@linaro-alip:/# i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@linaro-alip:/#
```

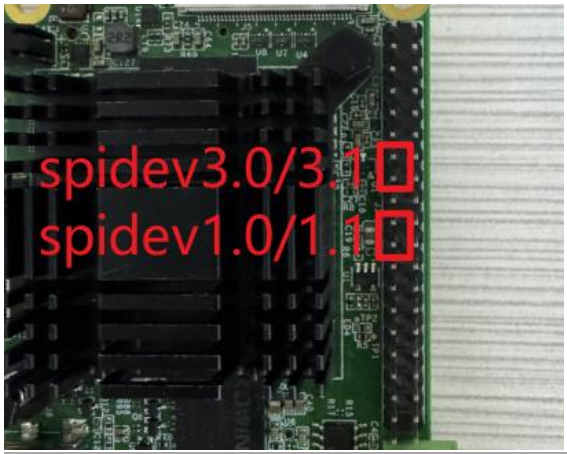
After connecting to the I2C device, execute the command: **i2cdetect -y 3**

```
root@linaro-alip:/# i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@linaro-alip:/#
```

Note: The verification method for nodes I2C8/I2C0 is the same as above, which means that after connecting the I2C device at the corresponding location, it can recognize the I2C mounting on the I2C bus.

3.8 SPI*4

(1) The schematic diagram of SPI pins is shown below:



(2) Short-circuit two pins and execute the following command through the debug port to conduct the SPI loopback test.

Create a file with custom content: **vi spi.txt**

Generate the out.txt file: **spidev_test -D /dev/spidev3.0 -i spi.txt -o out.txt**

See that the file is the same as the spi.txt created earlier: **cat out.txt**

```
root@linaro-alip:/# spidev_test -D /dev/spidev3.0 -i spi.txt -o out.txt
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
root@linaro-alip:/# cat out.txt
test123
root@linaro-alip:/# cat spi.txt
test123
root@linaro-alip:/#
```

Note: The positions of nodes spidev1.0 and spidev1.1 are the same, and the positions of nodes spidev3.0 and spidev3.1 are the same. The verification method is the same as above, just replace /dev/spidev *.*.

3.9 GPIO

(1) The schematic diagram of the GPIO pin is shown below:

(2) Enter the following command to control the voltage of GPIO port (measured with a multimeter).

echo 109 > /sys/class/gpio/export

echo out > /sys/class/gpio/gpio109/direction

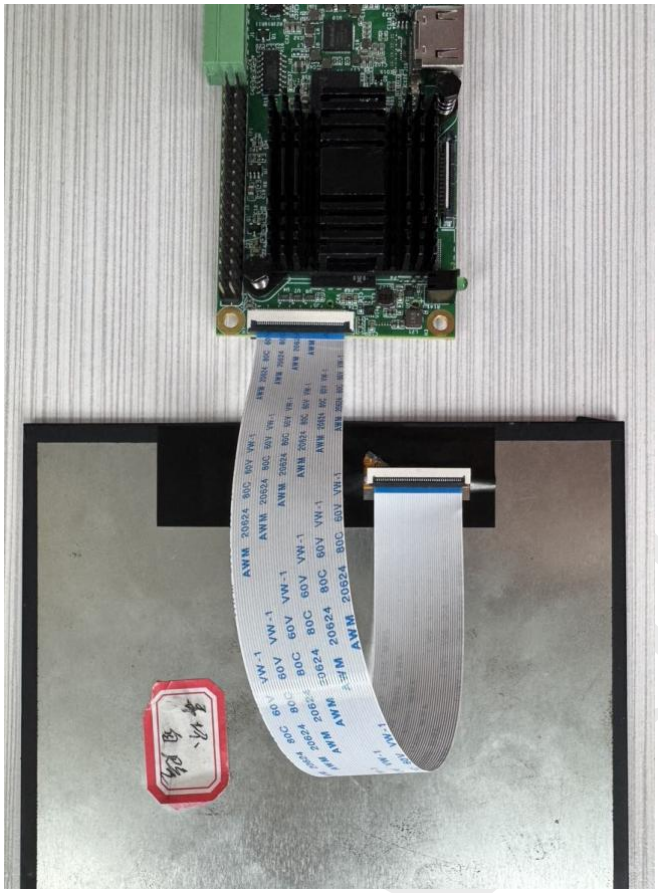
echo 1 > /sys/class/gpio/gpio109/value //Raise the voltage to 3.3V

echo 0 > /sys/class/gpio/gpio109/value //Lower the voltage to 0V

```
root@linaro-alip:/# echo 109 > /sys/class/gpio/export
root@linaro-alip:/# echo out > /sys/class/gpio/gpio109/direction
root@linaro-alip:/# echo 1 > /sys/class/gpio/gpio109/value
root@linaro-alip:/# echo 0 > /sys/class/gpio/gpio109/value
root@linaro-alip:/#
```

3.10 MIPI Screen

Do not connect HDMI OUT, only connect MIPI screen. The connection method is as follows. The MIPI screen is displaying normally and the mouse is clicking normally.



3.11 Computing Card

After the computing power card is connected to the board card and powered on, the following instructions can be executed to normally recognize and run the computing power card.

hardware identification: **lspci**

Check the hailortcli version: **hailortcli -v**

Automatic settlement: **hailortcli run /vendor/etc/firmware/hailo/resnet_v1_18.hef**

```
root@linaro-alip:/# lspci
00:00.0 PCI bridge: Rockchip Electronics Co., Ltd Device 3576 (rev 01)
01:00.0 Co-processor: Hailo Technologies Ltd. Hailo-8 AI Processor (rev 01)
root@linaro-alip:/#
root@linaro-alip:/# hailortcli -v
HailoRT-CLI version 4.22.0
root@linaro-alip:/#
root@linaro-alip:/# hailortcli run /vendor/etc/firmware/hailo/resnet_v1_18.hef
Running streaming inference (/vendor/etc/firmware/hailo/resnet_v1_18.hef):
  Transform data: true
    Type:      auto
    Quantized: true
Network resnet_v1_18/resnet_v1_18: 100% | 12635 | FPS: 2523.53 | ETA: 00:00:00
> Inference result:
 Network group: resnet_v1_18
    Frames count: 12635
    FPS: 2523.71
    Send Rate: 3039.11 Mbit/s
    Recv Rate: 20.19 Mbit/s
```

3.12 External Storage Devices

3.12.1 U disk

After connecting the USB flash drive to the board card, execute the following command to view the mount path of the USB flash drive.

View USB flash drive mounting path: **df -h**

```
root@linaro-alip:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        14G  3.8G  9.3G  30% /
devtmpfs        1.9G  8.0K  1.9G   1% /dev
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           780M  1.8M  779M   1% /run
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
/dev/mmcblk2p8   15G  288K   15G   1% /userdata
/dev/mmcblk2p7  123M   12M  108M  10% /oem
tmpfs           390M   32K  390M   1% /run/user/1000
/dev/sda1        15G  3.7G   11G  26% /mnt/sda1
root@linaro-alip:~# ls /mnt/sda1/
 1080p   720p  'System Volume Information'   music
root@linaro-alip:~#
```

3.12.2 TF card

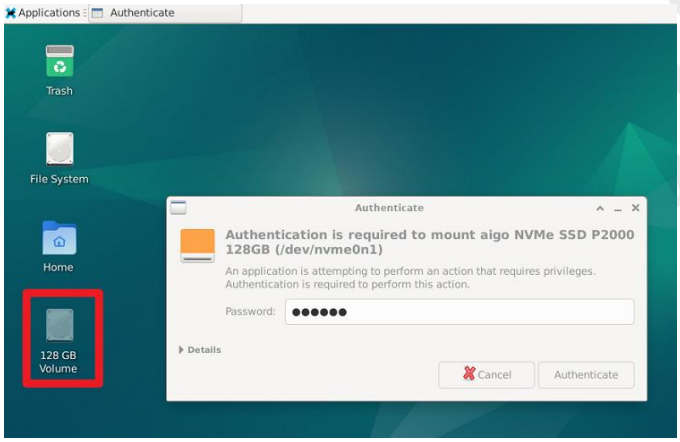
After the TF card is connected to the board card, execute the following command to view the mount path of the TF card.

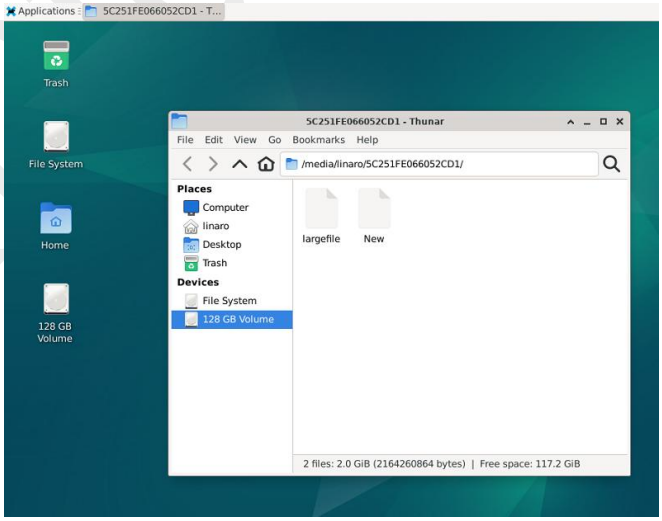
View TF card mounting path: **df -h**

```
root@linaro-alip:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        14G  3.8G  9.3G  30% /
devtmpfs        1.9G  8.0K  1.9G   1% /dev
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           780M  1.8M  779M   1% /run
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
/dev/mmcblk2p8   15G  288K   15G   1% /userdata
/dev/mmcblk2p7  123M   12M  108M  10% /oem
tmpfs           390M   32K  390M   1% /run/user/1000
/dev/mmcblk1    7.4G   16K  7.4G   1% /mnt/mmcblk1
root@linaro-alip:~# ls /mnt/mmcblk1/
'System Volume Information'
root@linaro-alip:~#
```

3.12.3 SSD

After connecting the SSD to the board and powering it on, the desktop will display the SSD icon, indicating that the system has recognized the SSD. Click on the SSD icon and enter the password “linaro” in the pop-up window to enter the SSD folder normally, as shown in the following figure.

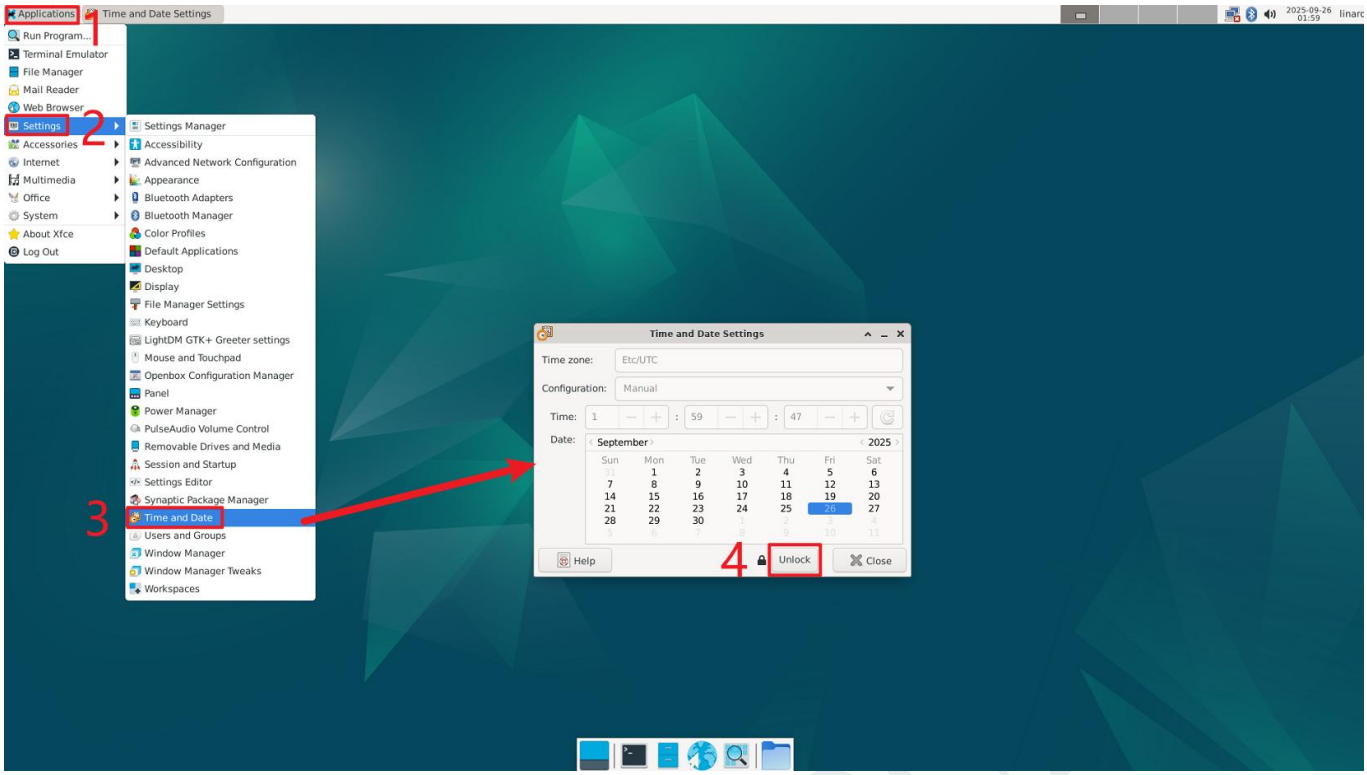




3.13 RTC

Prerequisite: Disconnect the network connection and connect the RTC battery.

Method 1: Interface settings: Click on “Application” in the upper left corner ->click on “Settings” ->click on “Time and Date” ->click on “Unlock”, enter the password: linaro, unlock and manually set the time.



Method 2: Execute the following instructions to read and write the time:

date -s "2028-05-01 20:00:00" //Set system time

hwclock -w //Write the time into the hardware clock

date //Read system time

hwclock -r //Read hardware clock time

```
root@linaro-alip:~# date -s" 2028-05-01 20:00:00"
Mon May  1 20:00:00 UTC 2028
root@linaro-alip:~# hwclock -w
root@linaro-alip:~# hwclock -r
2028-05-01 20:00:03.392036+00:00
root@linaro-alip:~# date
Mon May  1 20:00:04 UTC 2028
root@linaro-alip:~#
```