

IdeaT153 Linux5.10 User Manual

V1.0



Boardcon Embedded Design

Overview

This document applies only to the IdeaT153 development board. It is intended to help users quickly understand the hardware interfaces of the board and provides guidance for environment setup, source code compilation, firmware flashing, and functional testing of onboard hardware interfaces.

System Support

Development Board	Buildroot
CMT153 V1 + IdeaT153_V2	Y

Revision History

Version	Date	Author	Revision History
V1.0	2026-09-24	Liu Yuan	Initial version

Disclaimer

The information in this manual is provided for reference only. While Boardcon has made every effort to ensure the accuracy and reliability of the content, no guarantee is given regarding its completeness or correctness. All information in this manual is subject to change without prior notice.

Boardcon reserves the right to revise or update the contents of this manual at any time without prior notification. Boardcon shall not be held liable for any direct or indirect loss or damage arising from the use of this document or the products described herein.

Boardcon embedded design limited

2007-11 Haofang Tianji Plaza, 11008 Beihuan Avenue, Nanshan District,
Shenzhen, Guangdong, China. 518051

URL: www.armdesigner.com | www.boardcon.com

Email: market@armdesigner.com

Technical Support Inquiries: support@armdesigner.com

Tel: +86-755-26481393 | +86-755-27571591

Content

1. Introduction.....	4
1.1 Overview.....	4
1.2 Product Parameters	5
1.3 Hardware Interface Introduction.....	6
2. Install Drivers and Tools	7
2.1 Install CH9102X Driver.....	8
2.1.1 How to Connect the Serial Port Tool	8
2.1.2 Install Driver	8
2.2 Install Serial Terminal Tool.....	9
3 Firmware Flashing	12
4. Development Environment.....	15
4.1 Preparing the Development Environment.....	15
4.2 Installing Libraries and Toolkits	16
5. Compile Source Code	16
6. Buildroot Test.....	19
6.1 Serial Terminal.....	19
6.2 Display	20
6.3 Audio I/O	21
6.3.1 Audio Input (Recording).....	21
6.3.2 Audio Output (Playback)	22
6.4 USB Host.....	23
6.5 Ethernet.....	23
6.6 ADB	25
6.7 SD Card.....	26
6.8 GPADC	26
6.9 RS485.....	27
6.10 UART.....	29



6.11 SPI.....	29
6.12 GPIO	30
6.12.1 GPIO Number Calculation.....	30
6.12.2 GPIO Output Test	31
6.13 PWM.....	31

1. Introduction

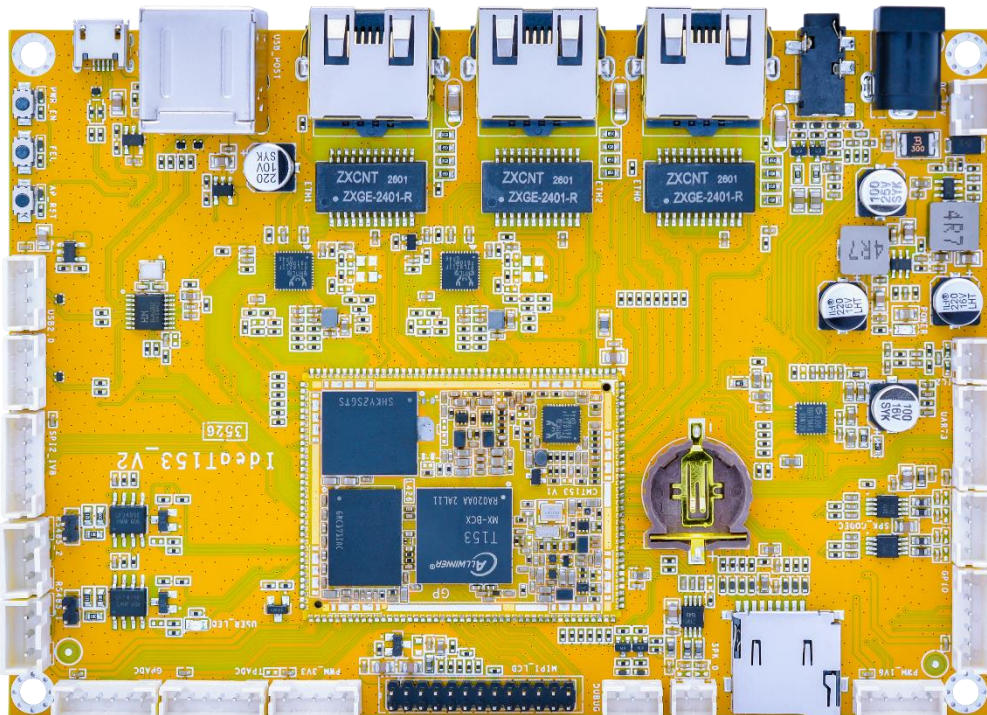
1.1 Overview

IdeaT153 is an industrial automation development board based on the Allwinner T153 quad-core high-performance processor. It is designed for industrial control, HMI, edge computing, and intelligent automation applications.

Powered by a quad-core Arm Cortex-A7 processor and one RISC-V E907 core, the T153 provides high-performance computing and real-time control capability. IdeaT153 integrates rich industrial interfaces, including 3x Ethernet, 2x RS485, UART, GPADC, and USB 2.0, enabling flexible connectivity for industrial devices.

The board supports MIPI DSI display output, speaker audio output, SD card storage, and ADB debugging, providing a complete platform for embedded Linux development, hardware evaluation, and product prototyping.

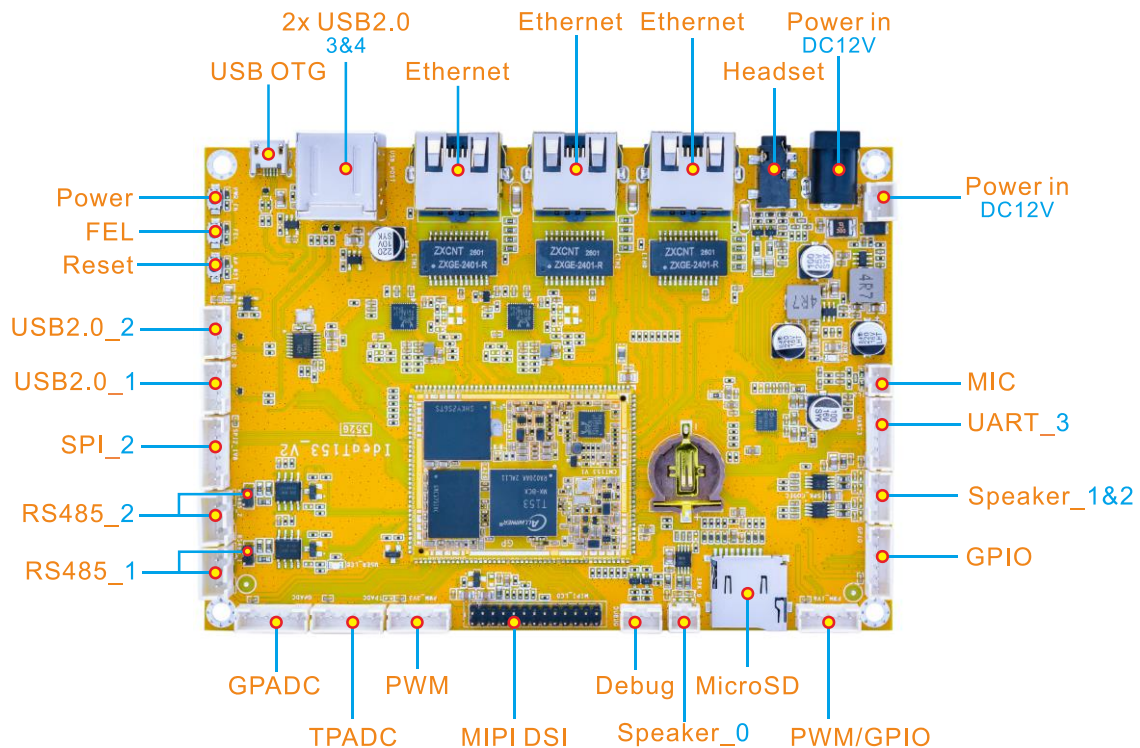
IdeaT153 is an ideal solution for industrial controllers, automation gateways, smart terminals, and embedded applications.



1.2 Product Parameters

Basic Parameters	
SOC	• Allwinner T153
CPU	• Quad-core Arm Cortex-A7 processor + RISC-V E907 real-time control core
RAM	• 512MB DDR3
ROM	• 8GB eMMC
Support systems	Buildroot
Hardware Parameters	
Extended Storage	• Support MicroSD card
Display	• Support MIPI DSI display
Audio	• Support headset input/output • Support MIC audio input • Support 2x speaker audio output
USB	• Support 4x USB2.0 host
Network	• Support 3x Gigabit Ethernet
Peripheral communication	• Support 2x RS485 • Support 1x UART
Other parameters	Support Debug, GPADC, SPI, ADB.
Electrical Parameters	
Power supply input voltage	12V/3A
RTC input voltage	3V/0.6uA
Operating temperature	0°C to 70°C
Storage temperature	-40°C to 85°C
Structural Parameters	
Core board dimensions	33.5mm x 43.0mm
Motherboard dimensions	135.0mm x 95.0mm

1.3 Hardware Interface Introduction



Interface parameters	
Power IN DC 12v	12V DC power input interface
Headset	Headset audio input/output interface
Ethernet	Gigabit Ethernet RJ45 interface
2x USB2.0 3&4	Dual-layer USB2.0 HOST interface
USB OTG	OTG download interface
Power	Power key
FEL	FEL key
Reset	Reset key
USB2.0_2	USB2.0_1 USB 2.0 host interface via 4-pin header
USB2.0_1	USB2.0_2 USB 2.0 host interface via 4-pin header
SPI_2	SPI interface

RS485_2	RS485 communication interface
RS485_1	RS485 communication interface
GPADC	GPADC interface
TPADC	TPADC interface
PWM	PWM output interface
MIPI DSI	MIPI DSI display interface
Debug	Debug UART interface
Speaker_0	Speaker output interface (audiocodec)
Micro SD	MicroSD card slot
PWM/GPIO	PWM/GPIO interface
GPIO	GPIO interface
Speaker_1&2	Speaker output interface (es8390)
UART_3	UART interface

2. Install Drivers and Tools

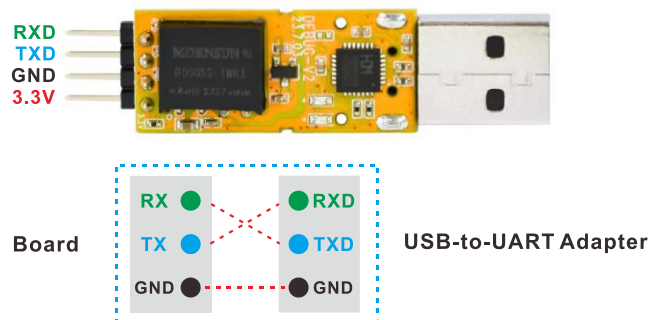
To download firmware and debug in the terminal, the following drivers and software need

to be installed (for Windows computers):

Number	Driver name	Driver	Use
2	CH9102x	SETUP.EXE	Serial port debugging driver
3	Serial Terminal Tool	SecureCRT.exe	Debugging tool

2.1 Install CH9102X Driver

2.1.1 How to Connect the Serial Port Tool



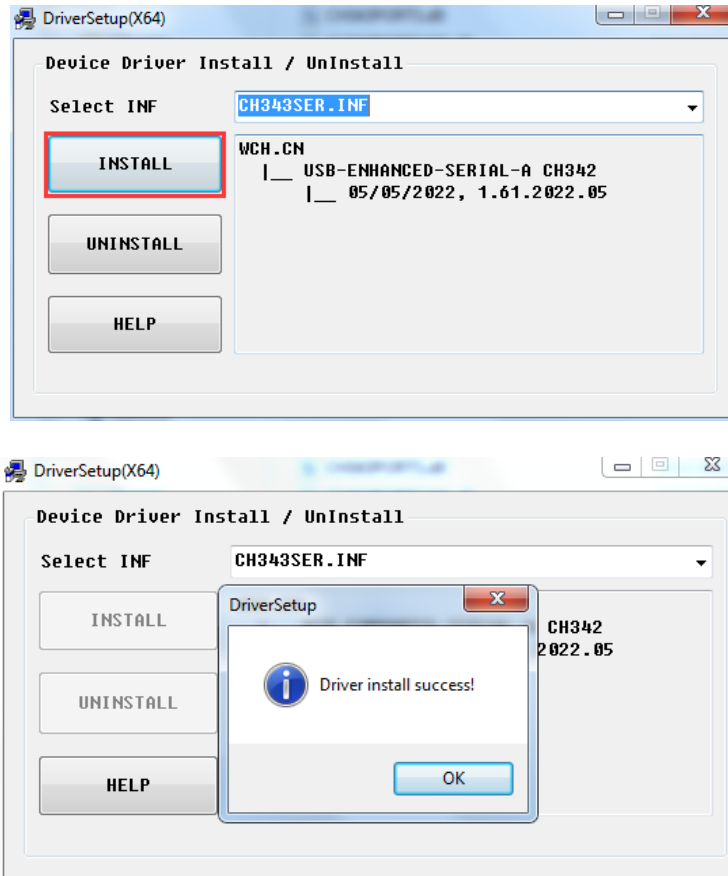
Pin	Connection Description
RXD	Receive, connect to TX pin of the board.
TXD	Transmit, connect to RX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.

2.1.2 Install Driver

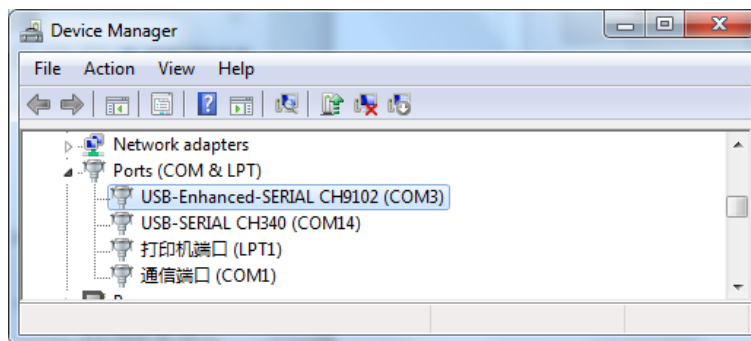
Step 1: Connect the CH9102X module to the PC.

Step 2: Unzip [CH343SER.ZIP](#) on Windows.

Step 3: Select and install the corresponding [SETUP.EXE](#) according to the computer properties.



Step 4: After the installation is completed, the device will be listed under **Device Manager** ports with unique serial port assigned.

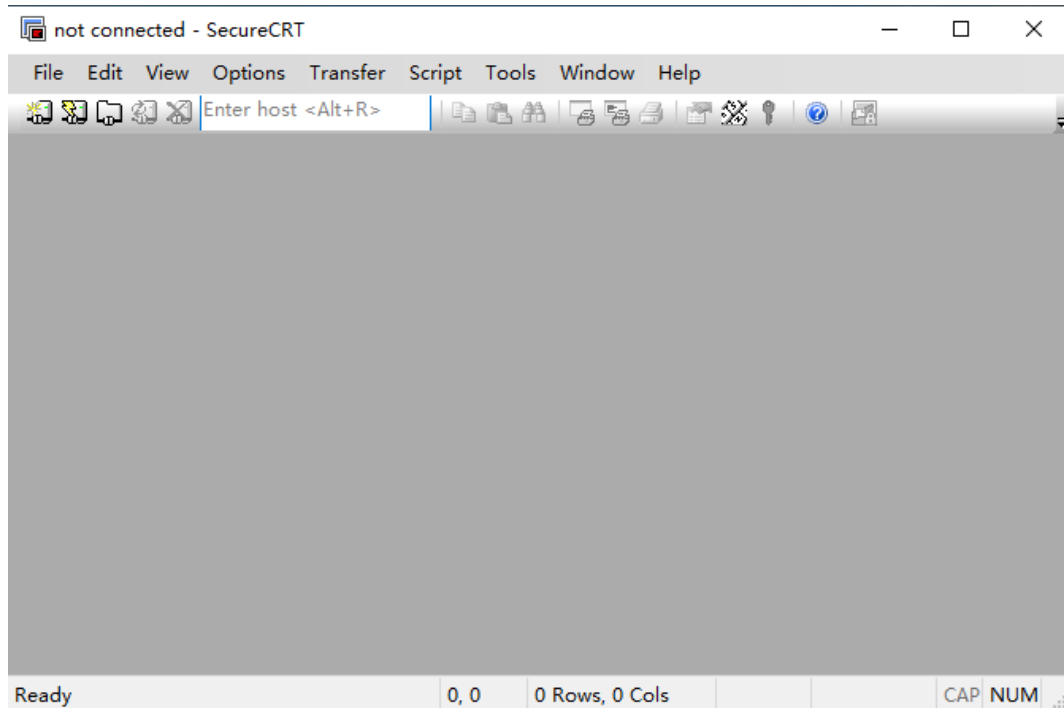


2.2 Install Serial Terminal Tool

The serial terminal SecureCRT is used for debugging in Windows. It can be used directly after decompression.

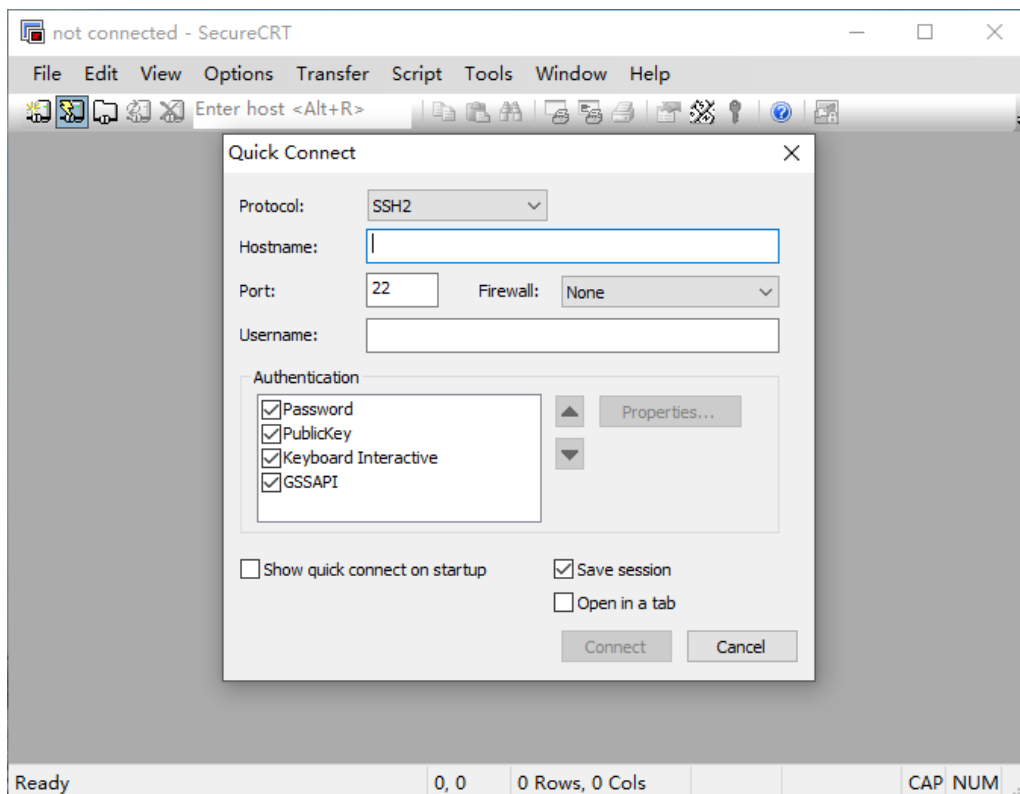
Step 1: Unzip *Platform/SecureCRT.rar* on PC.

Step 2: Click *SecureCRT/SecureCRT.exe* open to the SecureCRT.

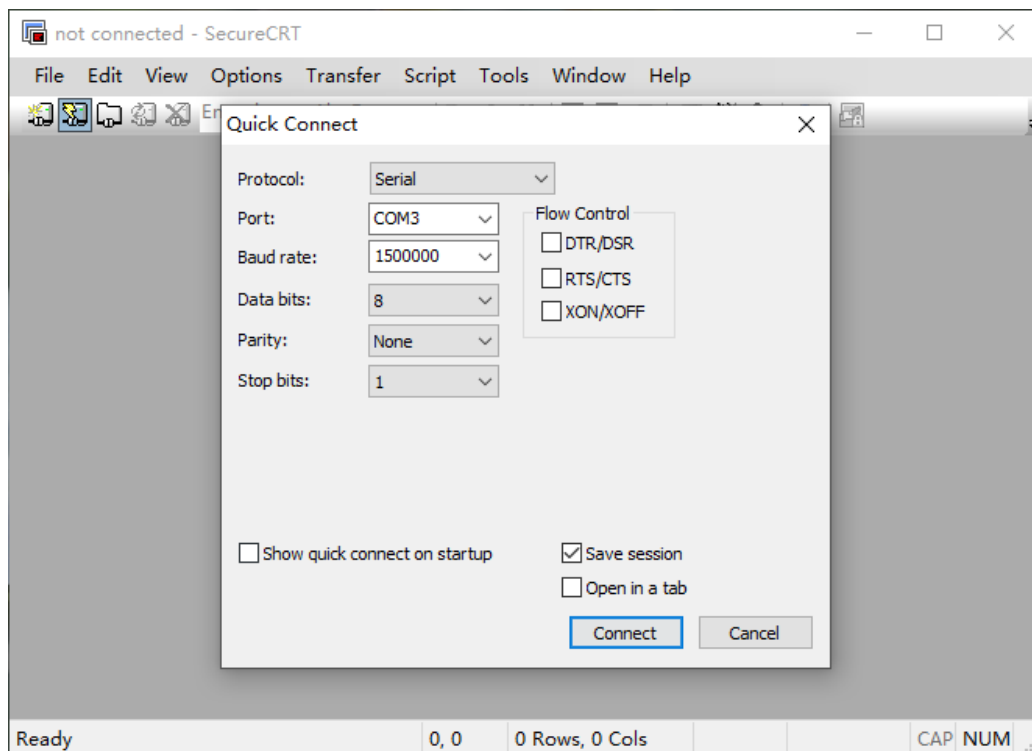


Step 3: Confirm that the CH9102X driver has been installed and the CH9102X module is connected to the PC.

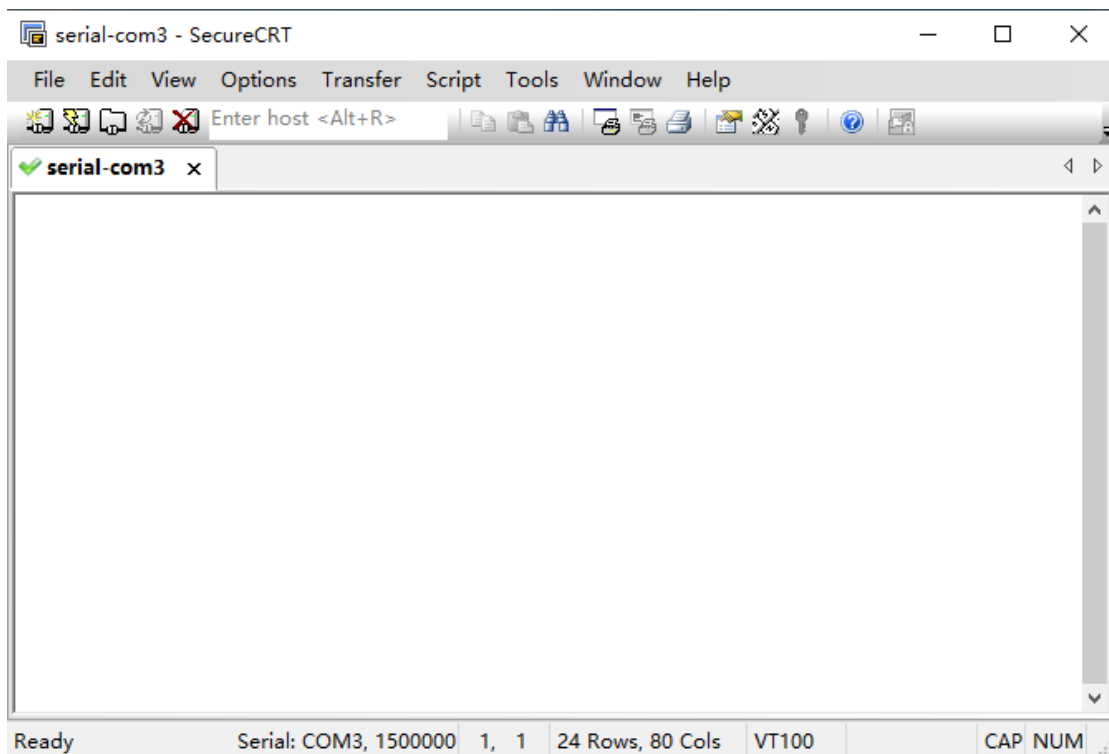
Step 4: Click the “**Quick Connect**” button to go to the Quick Connect configuration screen.



Step 5: Configure as shown in the following figure.



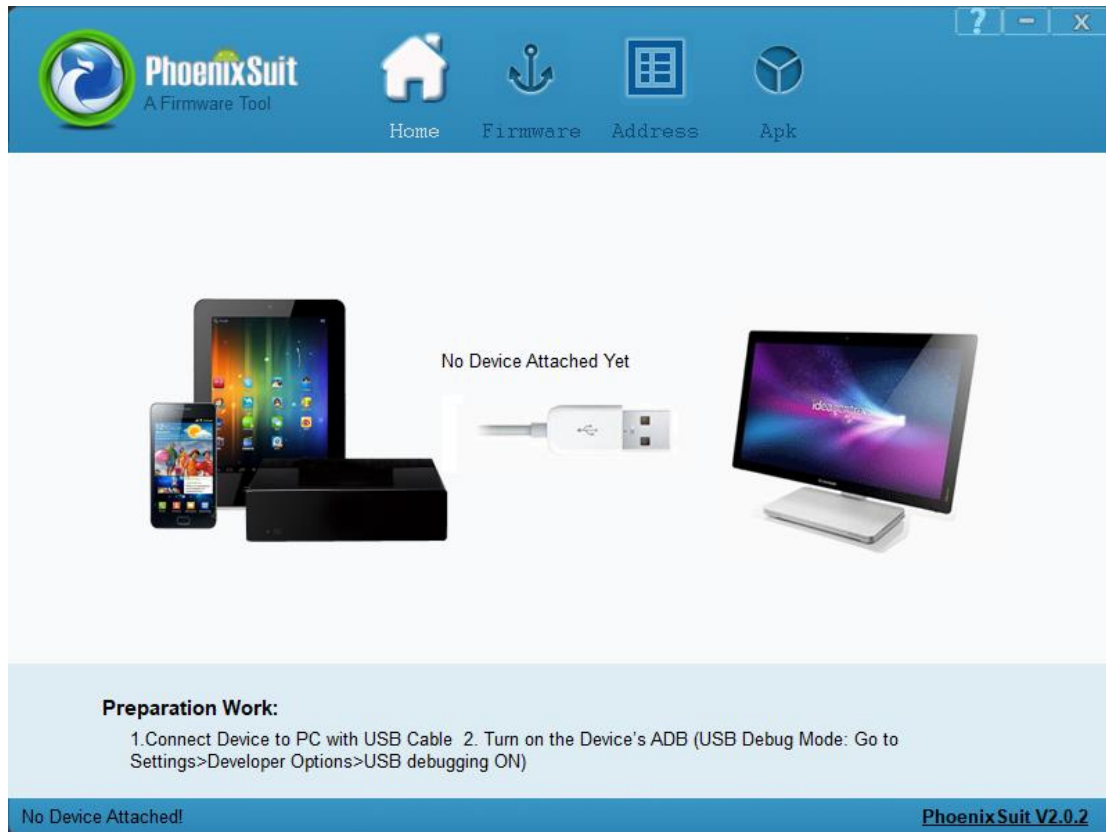
Step 6: After clicking the “**Connect**” button, the terminal serial interface will be successfully accessed.



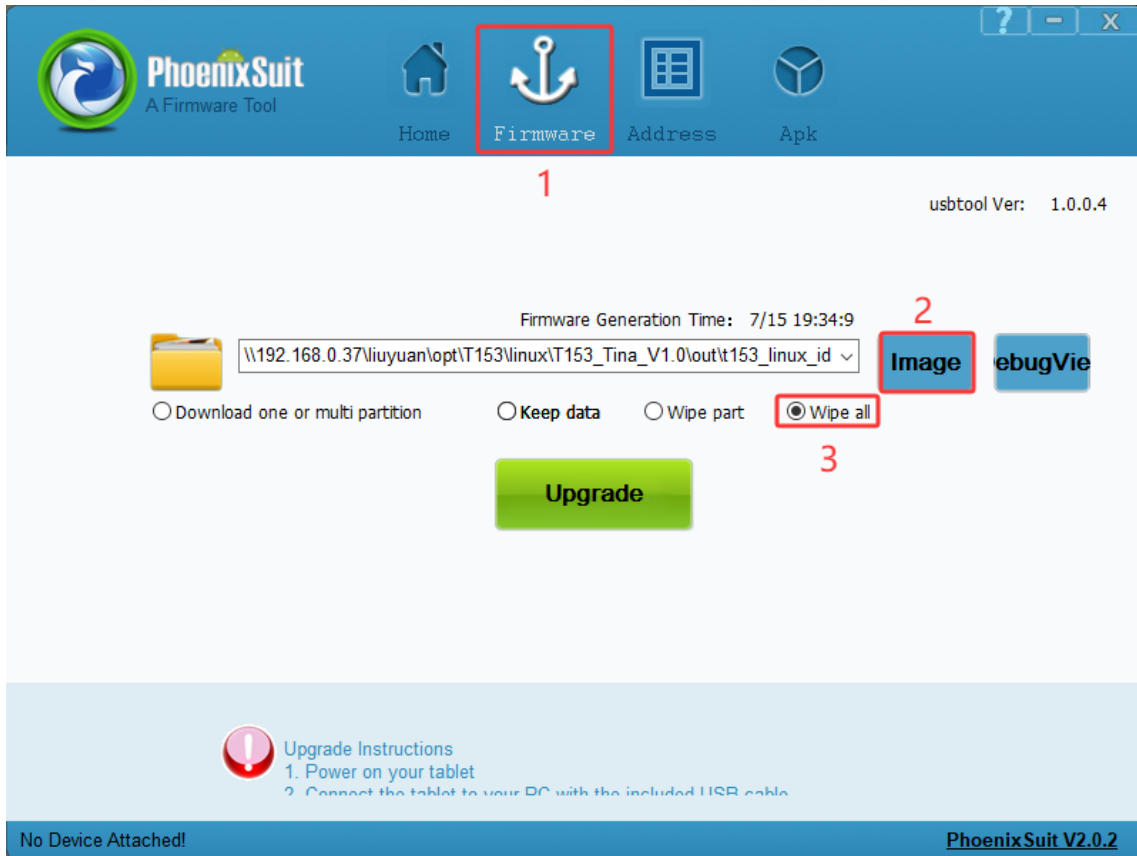
3 Firmware Flashing

Environment: Windows OS

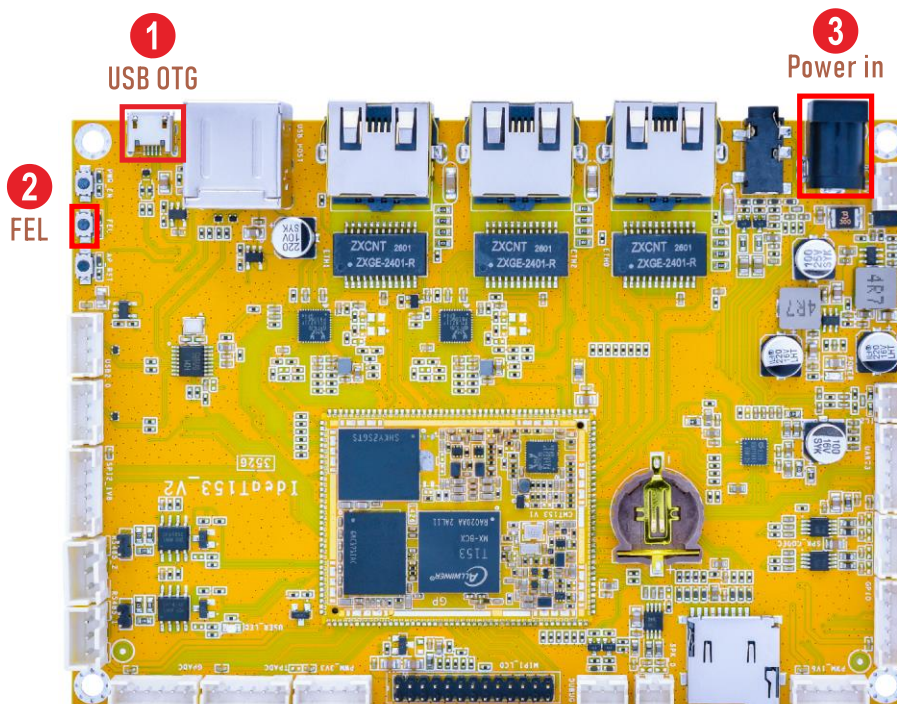
Step 1: Open *PhoenixSuit V2.0.2\PhoenixSuit.exe*.

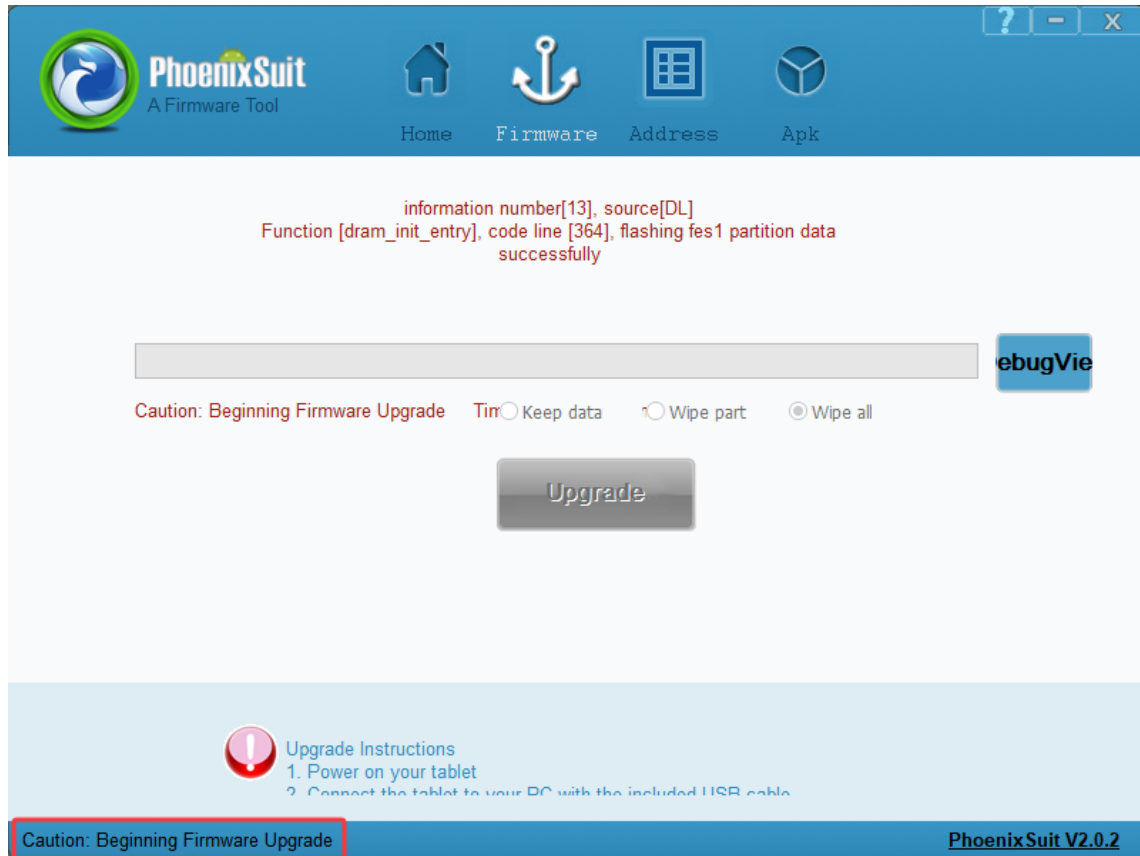


Step 2: Click **Firmware** -> **Image**, select **Image**, and choose “**Wipe all**” as the upgrade mode.

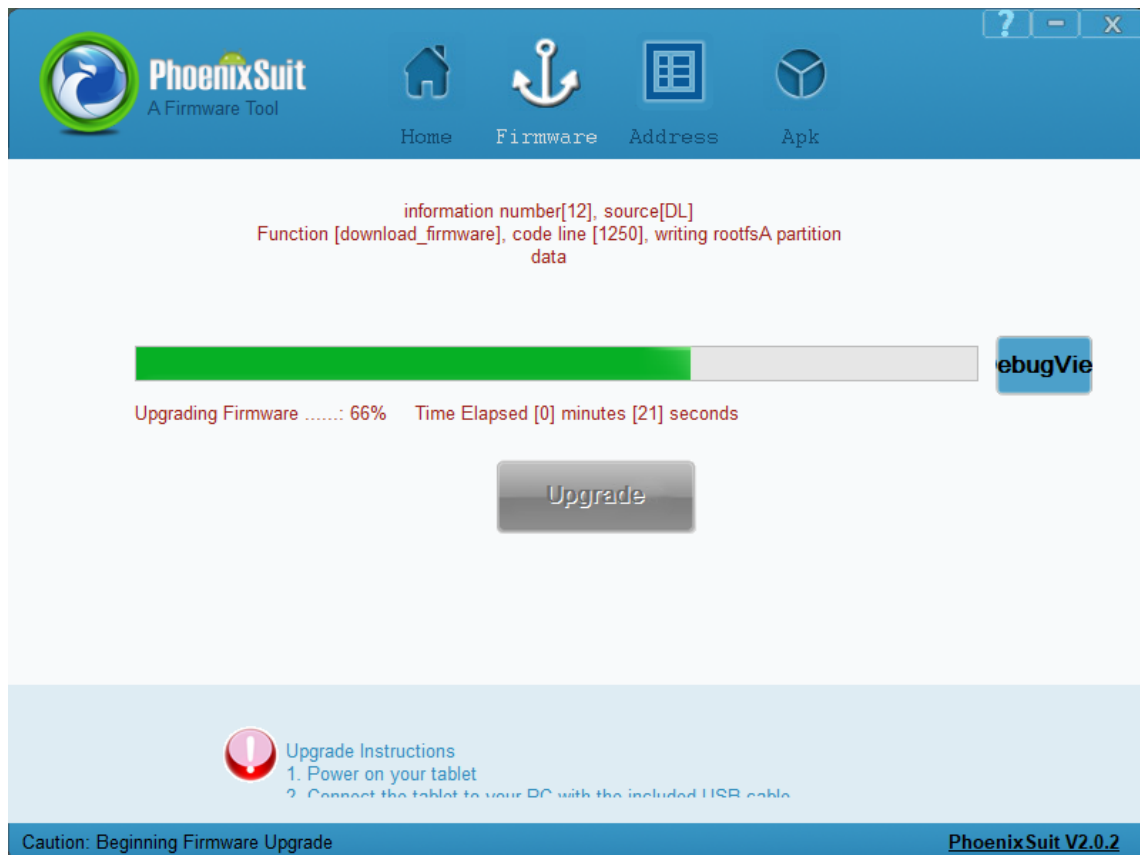


Step 3: Connect PC and development board with Micro USB cable, keep pressing the **FEL Key**, Power on the board until the pop-up window appears as shown below, or the flashing tool displays “**Beginning Firmware Upgrade**”.

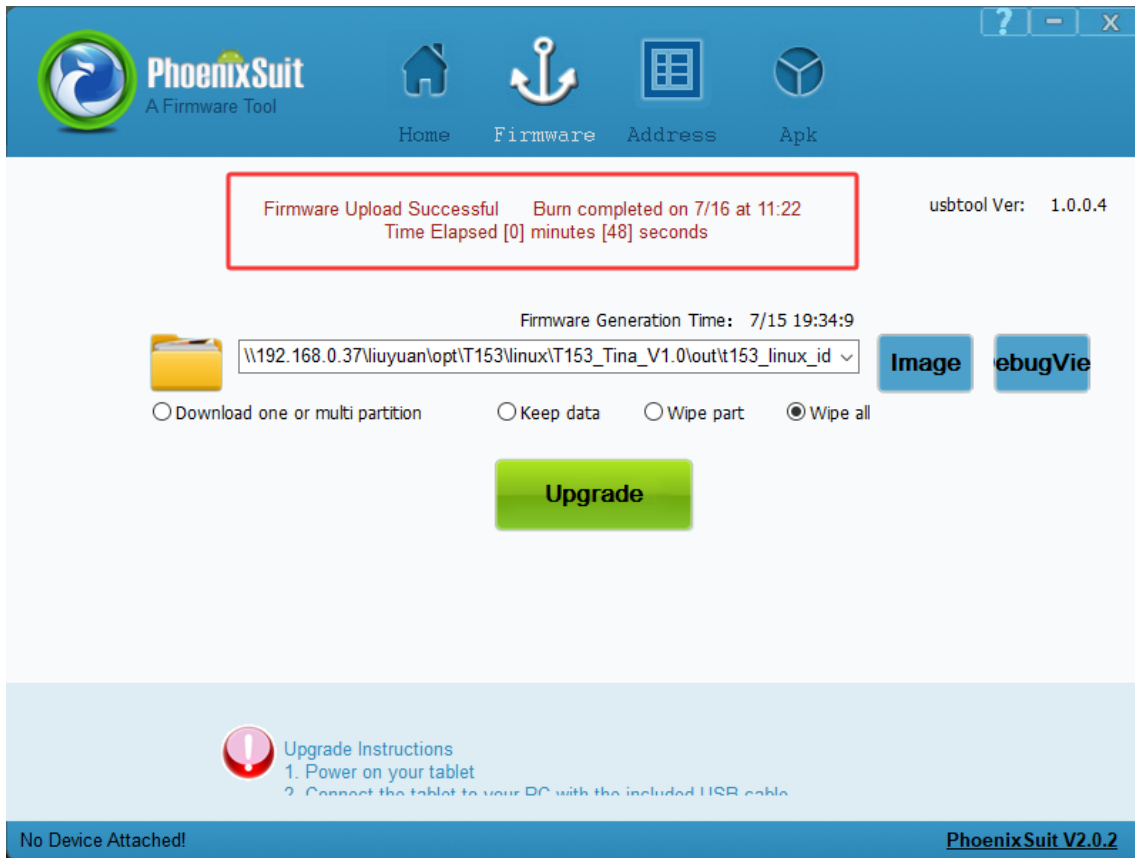




Step 4: Release the button, the download will start automatically.



Step 5: After the upgrade is complete, it will be displayed as follows.



4. Development Environment

4.1 Preparing the Development Environment

Ubuntu 22.04 or later is recommended for SDK compilation. If an error occurs during compilation, check the error message and install the required software packages accordingly. For other Linux distributions, the package names or installation commands may need to be adjusted.

In addition to the operating system requirements, the following hardware and software requirements should be met:

Hardware requirements	Software requirements
64-bit system, hard disk space should be greater than 200G. If you do multiple builds, you will need more hard drive space.	Ubuntu 22.04

4.2 Installing Libraries and Toolkits

This section provides the commands for installing the software packages required to build the SDK compilation environment. Other tools, such as Samba and SSH, should be installed as needed.

PC OS	Network	Permission
Ubuntu 22.04	online	root

To install the required tools, execute the following commands:

```
$ sudo apt-get install git ssh make gcc libssl-dev liblz4-tool libmpc-dev
$ sudo apt-get install expect g++ patchelf chrpath gawk texinfo chrpath diffstat
$ sudo apt-get install binfmt-support live-build bison flex fakeroot libgmp-dev
$ sudo apt-get install cmake gcc-multilib g++-multilib unzip device-tree-compiler
$ sudo apt-get install ncurses-dev libgucharmap-2-90-dev bzip2 expat gpgv2
$ sudo apt-get install cpp-aarch64-linux-gnu g++-aarch64-linux-gnu
$ sudo apt install python2 python-is-python3
```

5. Compile Source Code

Step 1: Extract the Source Code

To extract the source files, execute the following commands:

```
$ tar xvf IdeaT153_Tina_V*.tar.bz2
$ cd IdeaT153_Tina_V1.0
```

Step 2: Configure the Target Board

To configure the target board, execute the following command:

```
$ ./build.sh config
```

Configure the options according to the IdeaT153 board configuration. The corresponding settings are highlighted below.

```
liuyuan@boardcon:~/opt/T153/linux/IdeaT153_Tina_V1.0$ ./build.sh config
07-16 11:34:48.689 993348 D mkcommon : =====ACTION List: mk_config ;=====
07-16 11:34:48.693 993348 D mkcommon : options :
All available platform:
  0. android
  1. linux
Choice [linux]: 1
All available linux_dev:
  0. bsp
  1. buildroot
Choice [buildroot]: 1
All available ic:
  0. t153
Choice [t153]: 0
All available board:
  0. bga_demo
  1. bga_demo_amp_nand
  2. bga_demo_nand
  3. bga_demo_nor
  4. demo
  5. demo2_nand
  6. demo_amp_nand
  7. demo_nand
  8. demo_nor
  9. demo_qa
 10. ideat153_mipi
 11. okt153_emmc
 12. okt153_nand
Choice [ideat153_mipi]: 10
All available flash:
  0. default
  1. nor
Choice [default]: 0
All available kern_name:
  0. linux-5.10-origin
  1. linux-5.10-rt
  2. linux-5.10-xenomai
Choice [linux-5.10-origin]: 0
07-16 11:35:12.514 995380 D bsp : Setup BSP files
```

Step 3: Compile

To start compile, execute the following command:

```
$ ./build.sh
```

Step 4: Pack Image

To pack image, execute the following command:

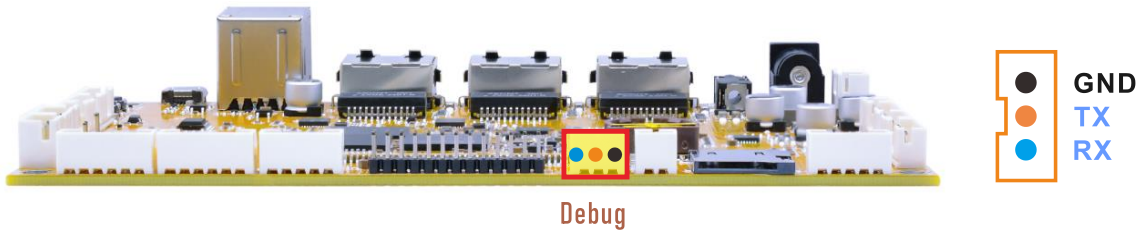
```
$ ./build.sh pack
```

After compiling all, the image will be generated in *out/*.

6. Buildroot Test

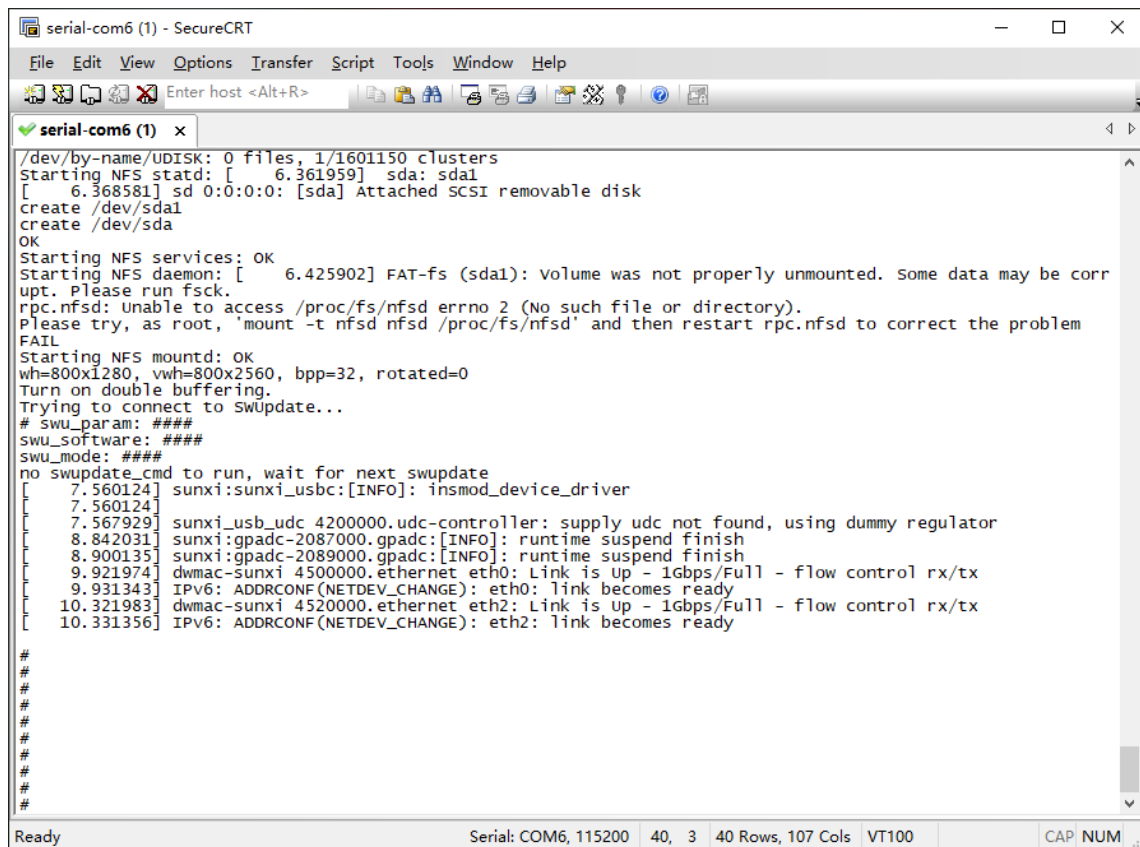
6.1 Serial Terminal

Step 1: Connect the USB-to-serial module to the PC and connect the other end to the Debug UART interface on the board.



Step 2: Open a serial terminal tool on the PC, select the corresponding serial port, and set the baud rate to 115200.

Step 3: Power on the board and check whether the boot log is displayed in the serial terminal.



```

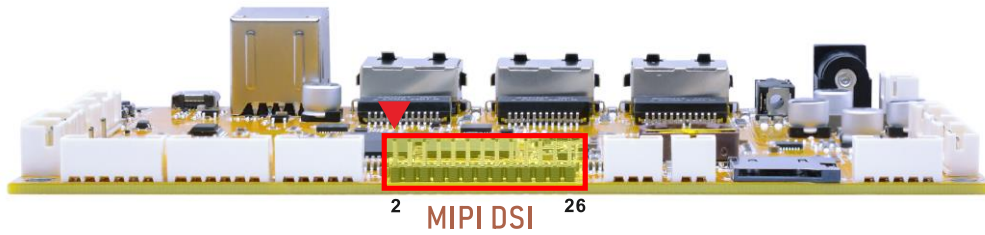
serial-com6 (1) - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
serial-com6 (1) x
/dev/by-name/UDISK: 0 files, 1/1601150 clusters
Starting NFS statd: [ 6.361959] sda: sda1
[ 6.368581] sd 0:0:0:0: [sda] Attached SCSI removable disk
create /dev/sda1
create /dev/sda
OK
Starting NFS services: OK
Starting NFS daemon: [ 6.425902] FAT-fs (sda1): volume was not properly unmounted. Some data may be corrupt. Please run fsck.
rpc.nfsd: unable to access /proc/fs/nfsd errno 2 (No such file or directory).
Please try, as root, 'mount -t nfsd nfsd /proc/fs/nfsd' and then restart rpc.nfsd to correct the problem
FAIL
Starting NFS mountd: OK
wh=800x1280, vwh=800x2560, bpp=32, rotated=0
Turn on double buffering.
Trying to connect to SWupdate...
# swu_param: ####
SWU_Software: ####
SWU_Mode: ####
no swupdate_cmd to run, wait for next swupdate
[ 7.560124] sunxi:sunxi_usbc:[INFO]: insmod_device_driver
[ 7.560124]
[ 7.567929] sunxi_usb_udc 4200000.udc-controller: supply udc not found, using dummy regulator
[ 8.842031] sunxi:gpadc-2087000.gpadc:[INFO]: runtime suspend finish
[ 8.900135] sunxi:gpadc-2089000.gpadc:[INFO]: runtime suspend finish
[ 9.921974] dwmac-sunxi 4500000.ethernet eth0: Link is up - 1Gbps/Full - flow control rx/tx
[ 9.931343] IPV6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 10.321983] dwmac-sunxi 4520000.ethernet eth2: Link is up - 1Gbps/Full - flow control rx/tx
[ 10.331356] IPV6: ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready
#
#
#
#
#
#
#
Ready Serial: COM6, 115200 40, 3 40 Rows, 107 Cols VT100 CAP NUM
  
```

For details about installing and using the serial terminal tool, please refer to Section [2.2](#)

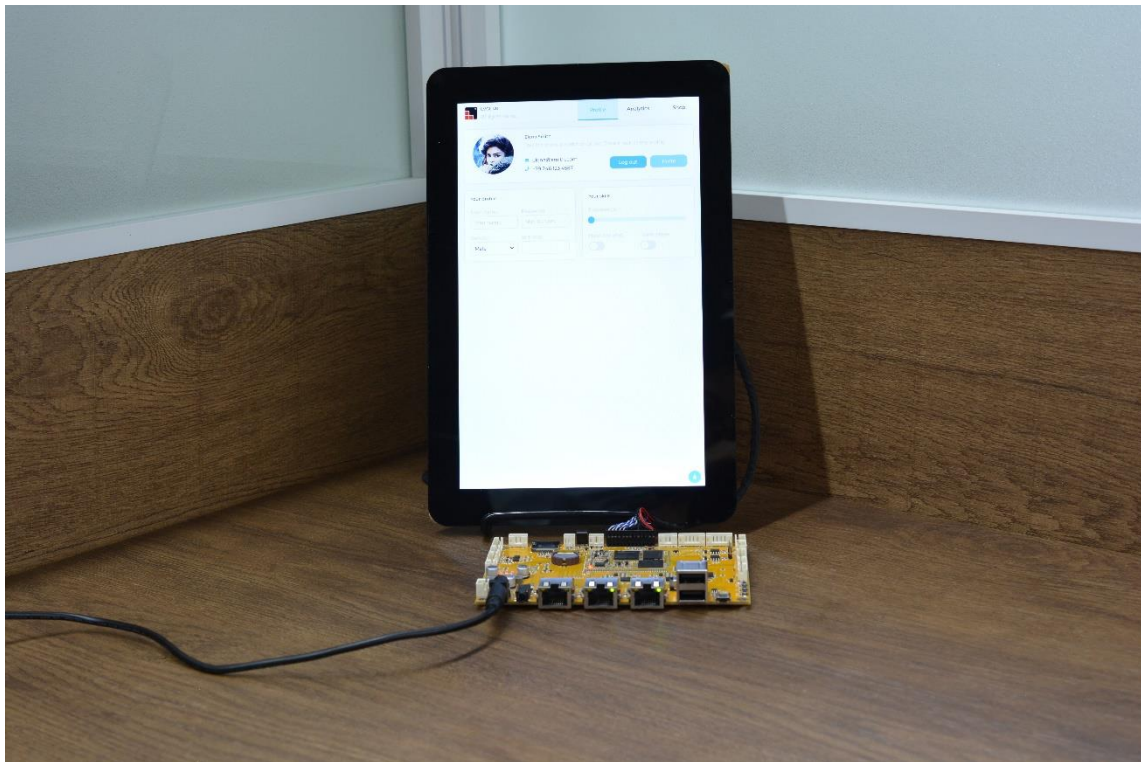
[Install CH9102X Driver](#) and [2.3 Install Serial Terminal Tool](#).

6.2 Display

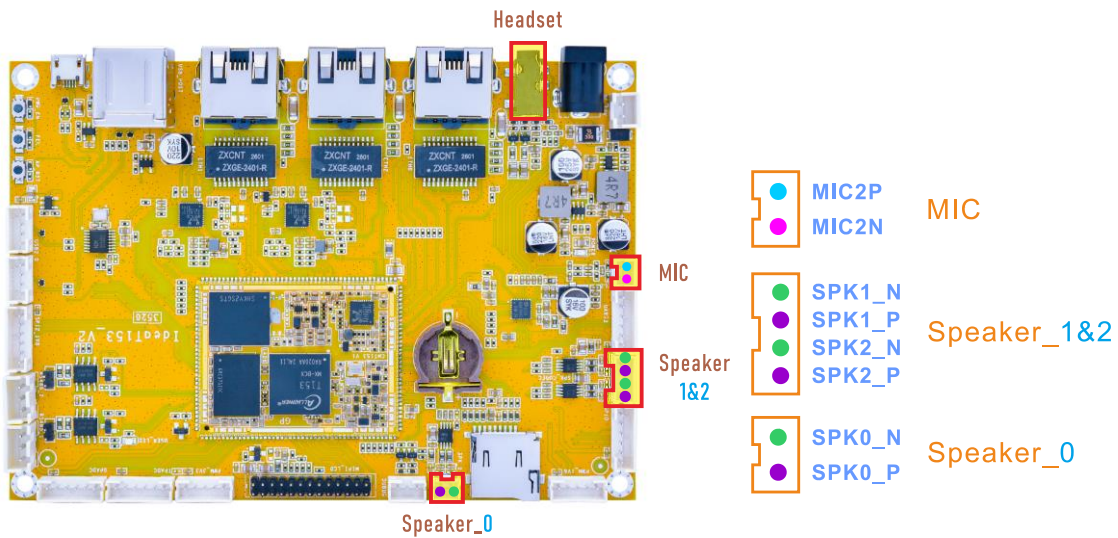
IdeaT153 is configured by default to support 800×1280 resolution display via the MIPI DSI interface.



MIPI DSI Display Effect



6.3 Audio I/O



6.3.1 Audio Input (Recording)

Step 1: Connect the headset to the audio jack and MIC

Note: The headset microphone takes priority over the onboard MIC. If no headset is connected, audio recording uses the onboard MIC.

Step 2: Check the audio devices.

Run the following command:

```
# cat /proc/asound/cards
```

```
# cat /proc/asound/cards
0 [audiocodec ]: audiocodec - audiocodec
audiocodec
1 [es8390 ]: es8390 - es8390
es8390
```

Step 3: Start audio recording.

```
# arecord -Dhw:1,0 -f cd test.wav
```

```
# arecord -Dhw:1,0 -f cd test.wav
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

Step 4: Play back the recorded audio file.

```
# aplay -Dhw:1,0 test.wav
```

```
# aplay -Dhw:1,0 test.wav
Playing WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

6.3.2 Audio Output (Playback)

Step 1: Connect the headset or speaker to the corresponding audio output interface.

The audio devices are mapped to different sound cards:

- Speaker_0: **audiocodec** sound card.
- Speaker_1&2 and headset: **es8390** sound card.

Note: When the headset is disconnected, the system automatically switches the audio output to the speaker.

Step 2: Check the audio devices.

Run the following command:

```
# cat /proc/asound/cards
```

```
# cat /proc/asound/cards
0 [audiocodec ]: audiocodec - audiocodec
audiocodec
1 [es8390 ]: es8390 - es8390
es8390
```

Step 3: Play audio files for output testing.

- **Speaker_0:**

```
# aplay /mnt/sdcard/mmcblk1p1/Music/2.wav
```

```
# aplay /mnt/sdcard/mmcblk1p1/Music/2.wav
Playing WAVE '/mnt/sdcard/mmcblk1p1/Music/2.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

- **Speaker_1&2 or headset:**

```
# aplay -Dhw:1,0 /mnt/sdcard/mmcblk1p1/Music/2.wav
```

```
# aplay -Dhw:1,0 /mnt/sdcard/mmcblk1p1/Music/2.wav
Playing WAVE '/mnt/sdcard/mmcblk1p1/Music/2.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

6.4 USB Host

IdeaT153 provides 4 USB 2.0 Host interfaces, including 2 standard USB Type-A ports and 2 pin-header interfaces for USB expansion. These interfaces can be used to connect USB peripherals, such as USB mice, USB keyboards, USB flash drives, and other USB devices.



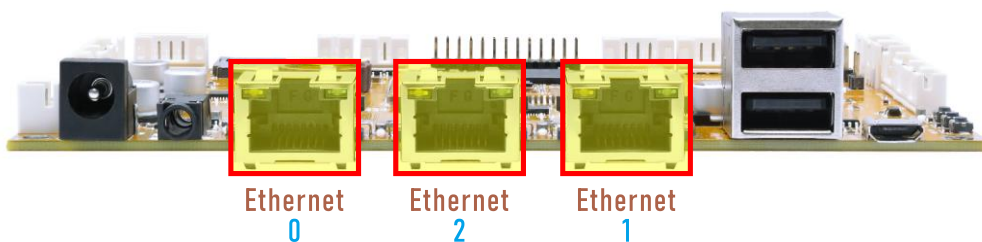
After a USB flash drive is inserted, the system automatically mounts it. Use the following command to check the device mount path:

```
# df -h
```

```
# df -h
Filesystem      Size    Used Available Use% Mounted on
/dev/root        495.6M  130.0M   355.3M   27% /
df: /dev/shm: No such file or directory
tmpfs           234.1M   84.0K   234.0M    0% /tmp
tmpfs           234.1M  252.0K   233.8M    0% /run
devtmpfs        201.6M     0    201.6M    0% /dev
/dev/mmcblk1p1   30.0G    4.5G   25.4G   15% /mnt/sdcard/mmcblk1p1
/dev/by-name/UDISK 6.1G    4.0K    6.1G    0% /mnt/UDISK
/dev/sda1       58.0G   37.5G   20.4G   65% /mnt/usb/sda1
```

6.5 Ethernet

Step 1: Connect the network cable to the Ethernet port.



After the network cable is connected, check the log. The log shows that the Gigabit Ethernet link is detected successfully.

```
# [ 132.402005] dwmac-sunxi 4500000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 132.411374] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 139.842006] dwmac-sunxi 4510000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx
[ 139.851368] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
[ 147.361970] dwmac-sunxi 4520000.ethernet eth2: Link is Up - 1Gbps/Full - flow control rx/tx
[ 147.371333] IPv6: ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready
```

Step 2: Check the network interface information.

Execute the following command:

```
# ifconfig
```

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:2A:AA:B2:E0
          inet addr:192.168.0.27  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::6689:f4cd:fb19:be82/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1184 errors:0 dropped:1 overruns:0 frame:0
          TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:120968 (118.1 KiB)  TX bytes:3998 (3.9 KiB)
          Interrupt:61 Base address:0x8000

eth1      Link encap:Ethernet  HWaddr 12:00:2A:AA:B2:E0
          inet addr:192.168.0.126  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a56e:c308:ad9d:b8f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1132 errors:0 dropped:1 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:116471 (113.7 KiB)  TX bytes:1620 (1.5 KiB)
          Interrupt:62 Base address:0x8000

eth2      Link encap:Ethernet  HWaddr 22:00:2A:AA:B2:E0
          inet addr:192.168.0.153  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::53a6:9368:b98d:b847/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:114359 (111.6 KiB)  TX bytes:1488 (1.4 KiB)
          Interrupt:63 Base address:0x8000
```

Step 3: Test the network connection.

Execute the following command:

```
# ping -I eth0 www.armdesigner.com
# ping -I eth1 www.armdesigner.com
# ping -I eth2 www.armdesigner.com
```

```
# ping -I eth0 www.armdesigner.com
PING www.armdesigner.com (13.32.99.13): 56 data bytes
64 bytes from 13.32.99.13: seq=0 ttl=246 time=186.159 ms
64 bytes from 13.32.99.13: seq=1 ttl=246 time=184.649 ms
64 bytes from 13.32.99.13: seq=2 ttl=246 time=185.975 ms
64 bytes from 13.32.99.13: seq=3 ttl=246 time=186.214 ms
64 bytes from 13.32.99.13: seq=4 ttl=246 time=185.883 ms
^C
--- www.armdesigner.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 184.649/185.776/186.214 ms
#
# ping -I eth1 www.armdesigner.com
PING www.armdesigner.com (13.32.99.13): 56 data bytes
64 bytes from 13.32.99.13: seq=0 ttl=246 time=188.461 ms
64 bytes from 13.32.99.13: seq=2 ttl=246 time=185.675 ms
64 bytes from 13.32.99.13: seq=3 ttl=246 time=186.704 ms
64 bytes from 13.32.99.13: seq=4 ttl=246 time=186.154 ms
64 bytes from 13.32.99.13: seq=5 ttl=246 time=186.499 ms
^C
--- www.armdesigner.com ping statistics ---
6 packets transmitted, 5 packets received, 16% packet loss
round-trip min/avg/max = 185.675/186.698/188.461 ms
#
# ping -I eth2 www.armdesigner.com
PING www.armdesigner.com (13.32.99.13): 56 data bytes
64 bytes from 13.32.99.13: seq=0 ttl=246 time=188.264 ms
64 bytes from 13.32.99.13: seq=1 ttl=246 time=185.977 ms
64 bytes from 13.32.99.13: seq=2 ttl=246 time=186.068 ms
64 bytes from 13.32.99.13: seq=3 ttl=246 time=185.923 ms
64 bytes from 13.32.99.13: seq=4 ttl=246 time=186.076 ms
^C
--- www.armdesigner.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 185.923/186.461/188.264 ms
```

6.6 ADB

Step 1: Connect the board to the PC with a Micro USB cable.



Step 2: Install the ADB driver on the Windows PC.

Step 3: Press **Windows + R** to open the Run dialog box. Type “cmd” and press Enter.

Step 4: Execute the following command to enable ADB:

```
# adb devices
# adb shell
```

```
C:\Windows\system32\cmd.exe - adb shell

C:\Users\15405>adb devices
List of devices attached
0402101560    device

C:\Users\15405>adb shell
#
#
#
#
#
#
```

6.7 SD Card

Step 1: Insert the micro SD card into the card slot.

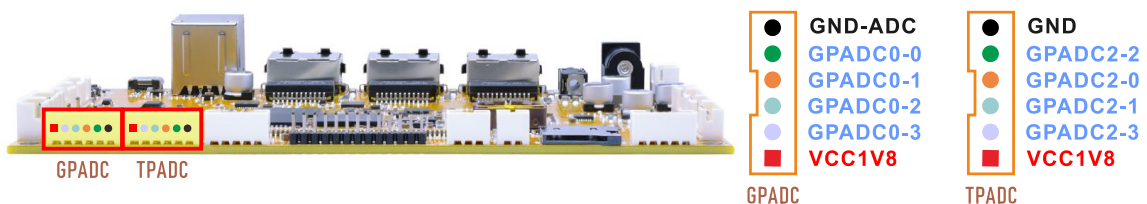


Step 2: After inserting and successfully recognizing the micro SD card, the system automatically mounts it. Use the following command to check the mount path:

```
# df -h
```

```
# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root        495.6M    130.0M    355.3M   27% /
df: /dev/shm: No such file or directory
tmpfs           234.1M     84.0K    234.0M    0% /tmp
tmpfs           234.1M    256.0K    233.8M    0% /run
devtmpfs        201.6M      0      201.6M    0% /dev
/dev/mmcblk1p1   30.0G      4.5G     25.4G   15% /mnt/sdcard/mmcblk1p1
/dev/by-name/UDISK 6.1G      4.0K      6.1G    0% /mnt/UDISK
/dev/sda1       58.0G     37.5G     20.4G   65% /mnt/usb/sda1
```

6.8 GPADC



Note: The ADC input voltage must not exceed 1.8V.

GPADC test command:

```
Single-channel test:
GPADC0-0 : cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
GPADC0-1 : cat /sys/bus/iio/devices/iio\:device0/in_voltage1_raw
GPADC0-2 : cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw
GPADC0-3 : cat /sys/bus/iio/devices/iio\:device0/in_voltage3_raw
GPADC2-0 : cat /sys/bus/iio/devices/iio\:device1/in_voltage0_raw
GPADC2-1 : cat /sys/bus/iio/devices/iio\:device1/in_voltage1_raw
GPADC2-2 : cat /sys/bus/iio/devices/iio\:device1/in_voltage2_raw
GPADC2-3 : cat /sys/bus/iio/devices/iio\:device1/in_voltage3_raw

Multi-channel test:
cat /sys/bus/iio/devices/iio\:device*/in_voltage*_raw
```

The following logs use GPADC0-0 as an example and show the ADC readings when the input is connected to 0V and 1.8V, respectively:

```
# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
1795
# cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
0
```

6.9 RS485



Step 1: Connect the RS485 test tool to the development board as shown in the diagram.



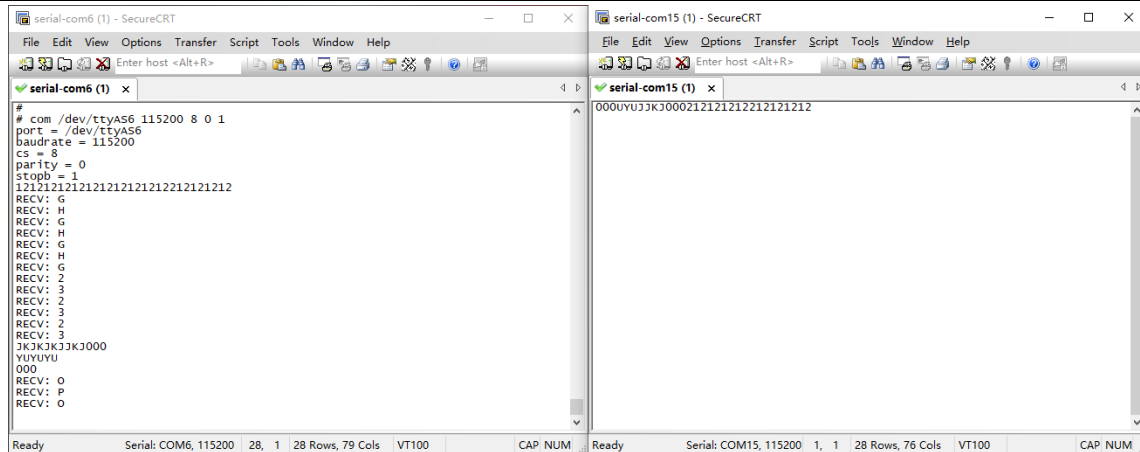
Step 2: Open two serial terminal windows. Set the debug serial port baud rate to 115200, and set the RS485 serial port baud rate to 115200.

Step 3: Execute the following command on the board to test the RS485-1 data transmission and reception:

```
# com /dev/ttyAS8 115200 8 0 1
```

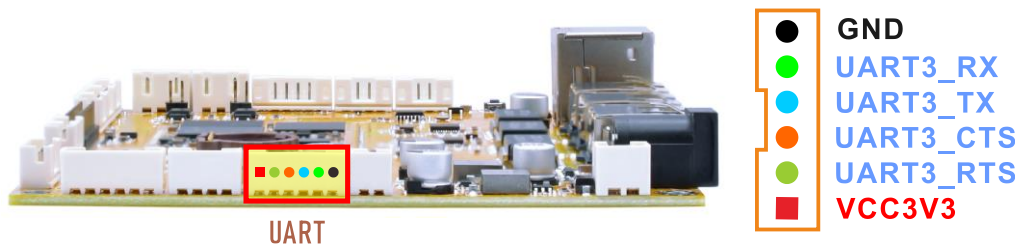
Step 4: Execute the following command on the board to test the RS485-2 data transmission and reception:

```
# com /dev/ttyAS6 115200 8 0 1
```



6.10 UART

Step 1: Connect the RX and TX pins of the UART interface together to create a loopback connection.



Step 2: Test UART3 loopback communication.

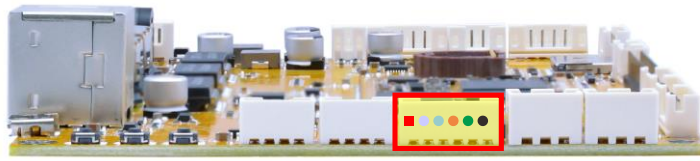
Execute the following command:

```
# com /dev/ttyAS3 115200 8 0 1
```

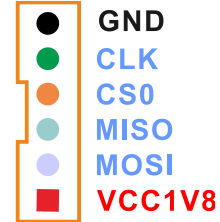
```
# com /dev/ttyAS3 115200 8 0 1  
port = /dev/ttyAS3  
baudrate = 115200  
cs = 8  
parity = 0  
stopb = 1  
klk1klk1klk1klk1klk1  
RECV: klk1klk1klk1klk1klk1kl  
565656565656565  
RECV: 565656565656565  
ioioioioioioi  
RECV: ioioioioioioi  
jkjk2323  
RECV: jkjk2323
```

6.11 SPI

Step 1: short circuit MISO and MOSI pins of SPI.



SPI

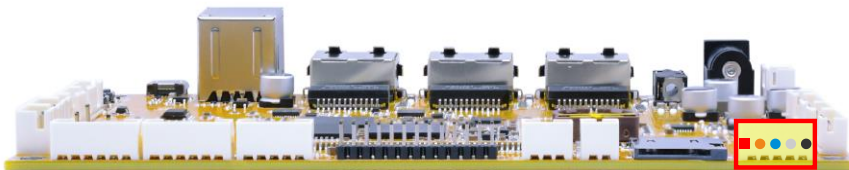


Step 2: Test spi.

```
# spidev_test -v
```

```
# spidev_test -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D
|.....@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D
|.....@.....|
```

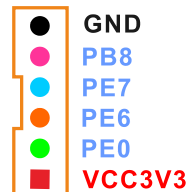
6.12 GPIO



PWM/GPIO



GPIO



6.12.1 GPIO Number Calculation

The GPIO number is calculated as follows:

```
GPIO_NUM = GPIO Port Base + Pin Number
```

The GPIO Port Base values are defined as follows:

GPIO Port	Base
PA	0
PB	32

PC	64
PD	96
PE	128
PF	160
PG	192

For example, PE0 is calculated as:

$$PE0 = 128 + 0 = 128$$

6.12.2 GPIO Output Test

The GPIO output function can be tested using GPIO PE0 as an example.

Execute the following commands:

```
# echo 128 > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio128/direction
# echo 1 > /sys/class/gpio/gpio128/value
# echo 0 > /sys/class/gpio/gpio128/value
```

```
# echo 128 > /sys/class/gpio/export: Export GPIO PE0
# echo out > /sys/class/gpio/gpio128/direction: Configure GPIO as output
# echo 1 > /sys/class/gpio/gpio128/value: Output high level
# echo 0 > /sys/class/gpio/gpio128/value: Output low level
```

Use a multimeter or oscilloscope to measure the GPIO pin voltage level changes and verify that the GPIO output function works properly.

6.13 PWM



PWM1-6 Test

Execute the following commands:

```
# echo 6 > /sys/class/pwm/pwmchip8/export
# echo 1000000 > /sys/class/pwm/pwmchip8/pwm6/period
# echo 500000 > /sys/class/pwm/pwmchip8/pwm6/duty_cycle
# echo 1 > /sys/class/pwm/pwmchip8/pwm6/enable
```

Test Parameters:

Period: 1 ms

Frequency: 1 / Period = 1 kHz

Duty cycle: 500000 / 1000000 = 50%

After executing the commands, use an oscilloscope to measure the PWM output waveform.



PWM1-7 Test

Execute the following commands:

```
# echo 7 > /sys/class/pwm/pwmchip8/export
# echo 100000 > /sys/class/pwm/pwmchip8/pwm7/period
# echo 20000 > /sys/class/pwm/pwmchip8/pwm7/duty_cycle
# echo 1 > /sys/class/pwm/pwmchip8/pwm7/enable
```

Test Parameters:

Period: $100000 \text{ ns} = 100 \text{ us}$

Frequency: $1 / 100 \text{ us} = 10 \text{ kHz}$

Duty cycle: $20000 / 100000 = 20\%$

After executing the commands, use an oscilloscope to measure the PWM output waveform:

