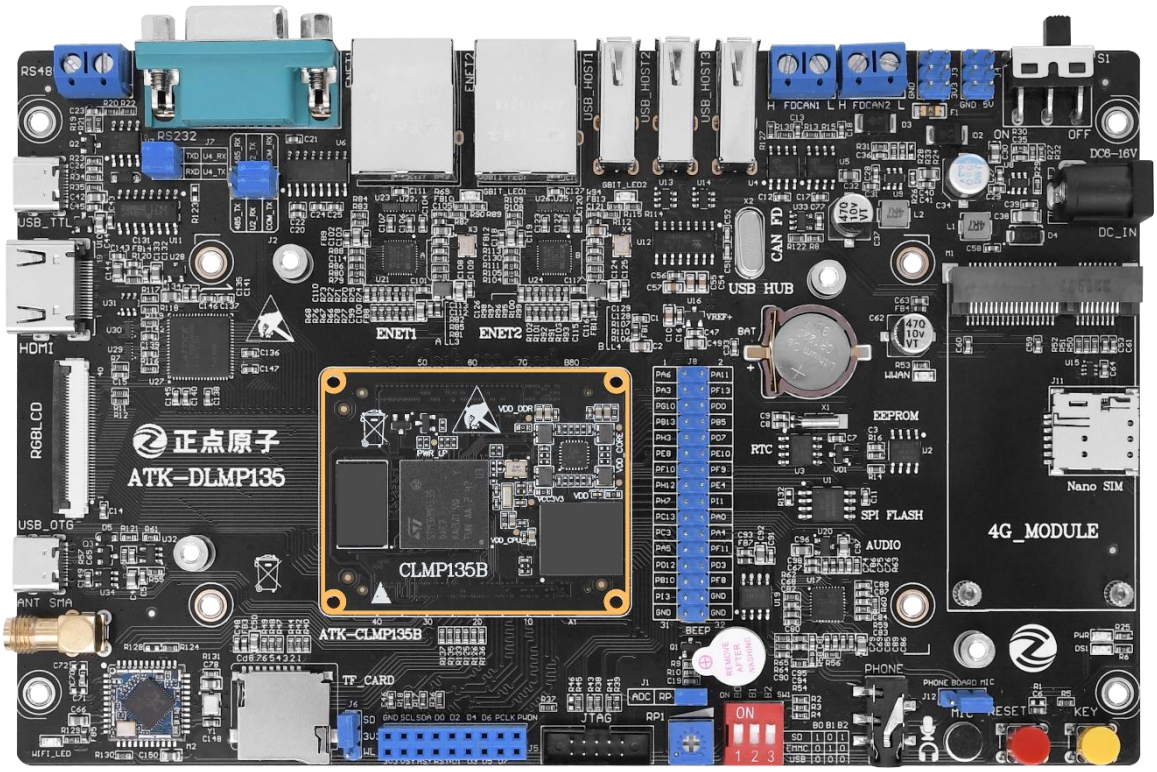
ATK-DLMP135

Quick test manual

V1.0



ALIENTEK

1. Shopping:

TMALL: https://zhengdianyuanzi.tmall.com

TAOBAO: https://openedv.taobao.com

2. Download

Address: http://www.openedv.com/docs/index.html

3. FAE

Website : www.alientek.com

Forum : http://www.openedv.com/forum.php

Videos : www.yuanzige.com

Fax : +86 - 20 - 36773971

Phone : +86 - 20 - 38271790

Disclaimer

The product specifications and instructions mentioned in this document are for reference only and subject to update without prior notice; Unless otherwise agreed, this document is intended as a product guide only, and none of the representations made herein constitutes a warranty of any kind. The copyright of this document belongs to Guangzhou Xingyi Electronic Technology Co., LTD. Without the written permission of the company, any unit or individual shall not be used for profit-making purposes in any way of dissemination.

In order to get the latest version of product information, please regularly visit the download center or contact the customer service of Taobao ALIENTEK flagship store. Thank you for your tolerance and support.

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	2023.04.14

Catalogue

Chapter 1. Introduces ATK-DLMP135 hardware resources 1
1.1 Introduction to the Base Board Resources 1
1.2 Core Board Resource Overview....2
Chapter 2. ATK-DLMP135 Programming System 3
2.1 STM32CubeProgrammer Introduction 5
2.2 Writing the system onto the development board under Windows 5
2.2.1 Installing STM32CubeProgrammer on Windows....5
2.2.2 Using STM32CubePro to burn the system to eMMC on Windows....12
2.2.3 Windows uses STM32CubePro to burn the system onto the TF (SD) card. 16
2.3 Writing the system to the development board under Linux 16
2.3.1 Installing STM32CubeProgrammer under Linux....16
2.3.2 In Linux, use the STM32CubePro to burn the system onto the eMMC. 22
2.3.3 Linux uses STM32CubePro to burn the system onto a TF (SD) card. 27
Chapter 3. Preparations Before Using ATK-DLMP135....28
3.1 Precautions Before Powering On 28
3.2 Serial Port Software Installation 28
3.3 Setting of jumpers and login to the development board....29
Chapter 4. Functional test 32
4.1 LED test, buzzer test 32
4.2 Key Test 32
4.3 LCD test 34
4.3.1 Touch test....34
4.3.2 Backlight test....35
4.4 Serial Port Test....35
4.4.1 RS232 Serial Port Test 35
4.4.2 RS485 Serial Port Test 39
4.5 USB Test....40
4.5.1 USB HOST Test....40
4.5.2 USB mouse test 42
4.6 Network testing 42
4.7 CAN Test 43
4.7.1 CAN test....44
4.7.2 CAN FD Test 45
4.8 RTC Clock Test 47
4.9 EEPROM Test....48
4.10 SPI FLASH test....48
4.11 Audio Testing....49
4.11.1 Playing Audio Testing....49
4.11.2 Recording Test 50
4.12 HDMI Test 50
4.13 Board-based SDIO WIFI test 53

4.14 Board Bluetooth test....56
4.15 Testing of 4G module ME3630-W....58
4.15.1 PPP dial-up Internet access....59
4.15.2 ECM Internet Access....62
4.16 TF(SD) Card Test....64
4.16.1 Read Speed Test....65
4.16.2 Writing speed test....65
4.17 OV5640 Camera....65
4.18 ADC test....67
4.19 CPU temperature....68
4.20 CPU frequency....68
4.21 View system information....69
4.21.1 View the system hostname....69
4.21.2 View CPU-related information....69
4.21.3 View memory-related information....69
4.22 Shutting down and restarting....70
4.23 Check the factory file system version....71
4.24 4G module EC20 test....71
4.24.1 Use quectel-CM....71

Chapter 1. Introduces ATK-DLMP135 hardware resources

This chapter will introduce the hardware resources of the ATK-DLMP135 development board.

1.1 Introduction to the Base Board Resources

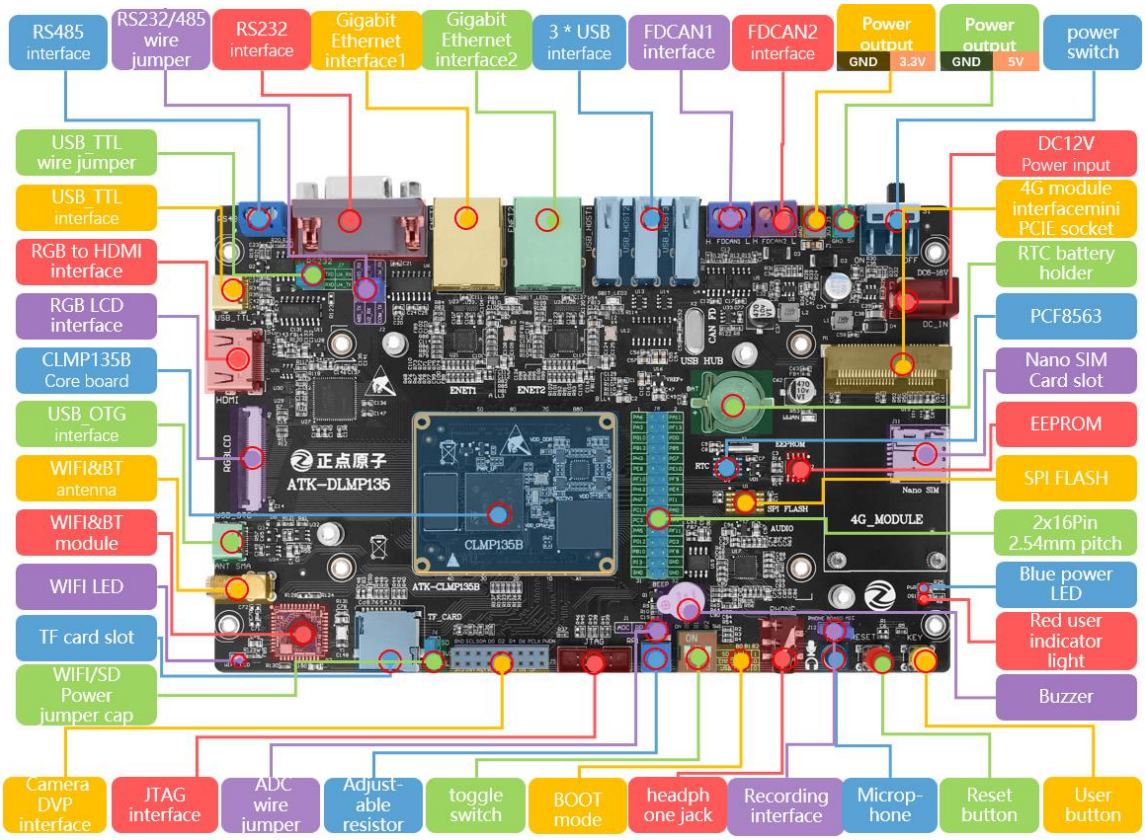


Figure 1.1-1 ATK-DLMP135 Baseboard Resource Overview

For more details, please refer to the development board CD A - Basic Materials -> 10_user_manual -> 03 [ALIENTEK] ATK-DLMP135 Hardware Reference Manual V1.x.pdf.

1.2 Core Board Resource Overview

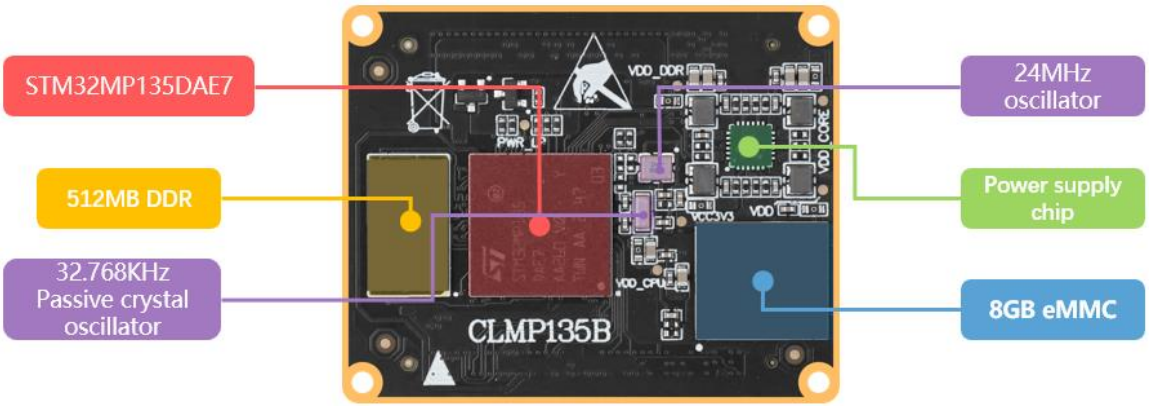


Figure 1.2-1 Core Board Resource Overview

For more details, please refer to the development board CD A - Basic Materials -> 10_user_manual -> 03 [ALIENTEK] ATK-DLMP135 Hardware Reference Manual V1.x.pdf.

Chapter 2. ATK-DLMP135 Programming System

Introduction:

The ATK-DLMP135 development board has already had the system pre-programmed onto the core board's eMMC storage when it was manufactured. Therefore, users can skip this section and do not need to re-program the system onto the development board. They can directly test it on the machine or use it directly.

However, for the following reasons, users may need to re-program the factory system:

(1) Due to firmware updates of the factory system, the system may have been optimized or updated;

(2) Due to damage to the factory system, files have been modified or deleted, making it impossible to start, and they want to restore to the factory system;

(3) Due to the development process, the user has burned their own u-boot, kernel, and system, but the hardware functionality cannot be verified, and they need to re-program it back to the factory system. They should first verify if the hardware is working properly, and then modify the source code.

Here, we provide two ways to program the system onto the development board as follows:

(1) In Windows, use STM32CubeProgrammer to program the system. A USB 3.0 interface is required. Do not use the USB 2.0 interface (because it will be slower! Although the board's OTG is 2.0).;

(2) In Linux, use STM32CubeProgrammer to program the system. A USB 3.0 interface is required.

2.1 STM32CubeProgrammer Introduction

STM32CubeProgrammer, abbreviated as STM32CubeProg, is a cross-platform, all-in-one program flashing tool suitable for STM32 series products. The ALIENTEK ATK-DLMP135 development board can use this tool for system flashing.

The following features:

(1) "Cross-platform", it means it supports Windows, macOS and Linux operating systems. The software requires a Java environment during operation.

(2) "All-in-one", it means it supports flashing firmware through USB, ST-LINK, UART, OTA and other methods.

To facilitate user usage, we have downloaded the STM32CubeProgrammer tool from the ST official website and stored it at the path of the development board CD A - Basic Materials -> 5_tools -> 2_ST_official_development_tools. ST provides three tool supports for different systems, namely Linux, macOS and Windows. Due to our limited conditions, we will only explain the installation and usage of STM32CubeProgrammer for Windows and Linux.

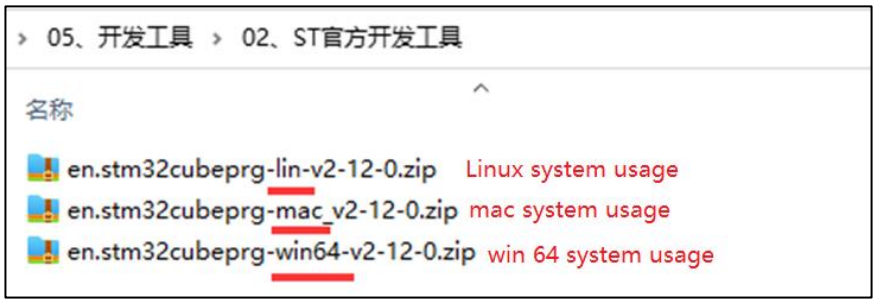


Figure 2.1-1 Official development tools of ST

2.2 Writing the system onto the development board under Windows

2.2.1 Installing STM32CubeProgrammer on Windows

Extract the en.stm32cubeprg-win64-v2-12-0.zip installation package. Double-click SetupSTM32CubeProgrammer_win64.exe to start the installation process.

The pop-up welcome installation interface. Just click Next directly; (For the subsequent operations, you can basically click Next as well.)

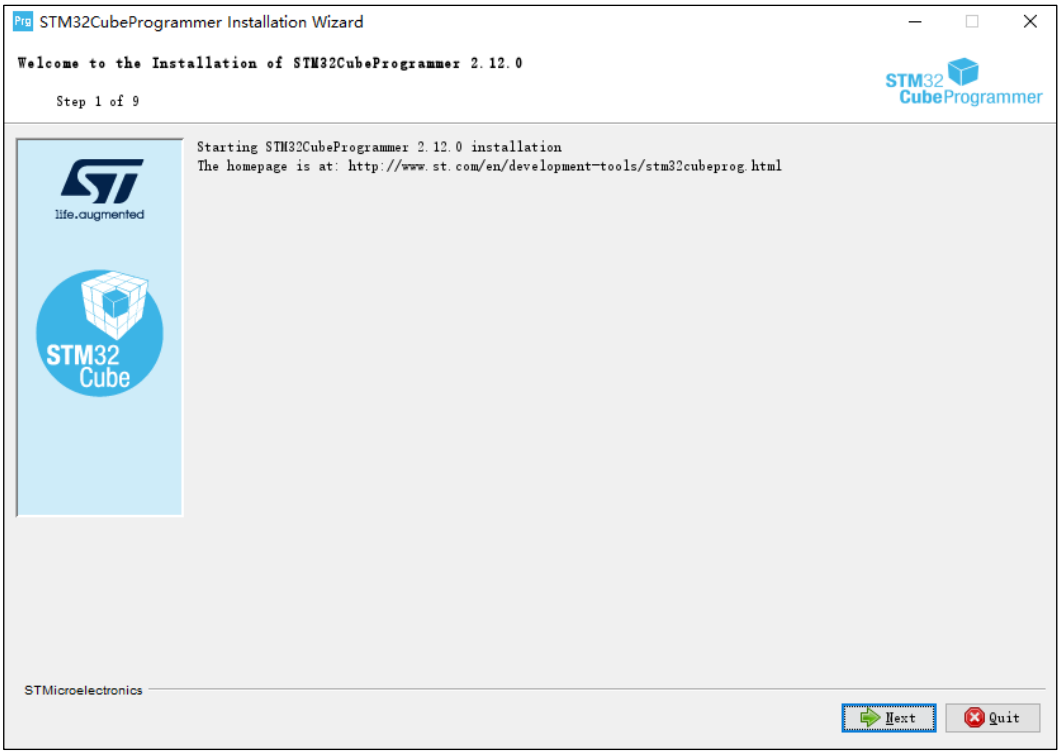


Figure 2.2-1 The installation interface of STM32CubeProgrammer

Read the information about STM32CubeProgrammer and click "Next" directly;

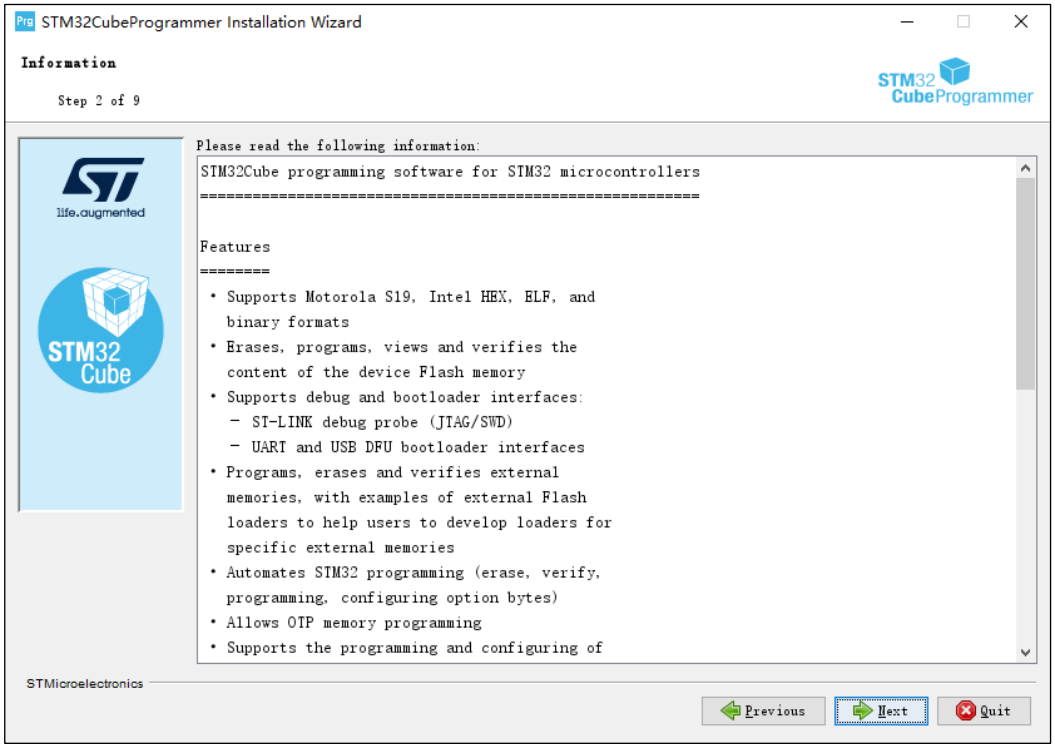


Figure 2.2-2 Read the relevant information

Check "Agree to License" and click "Next";

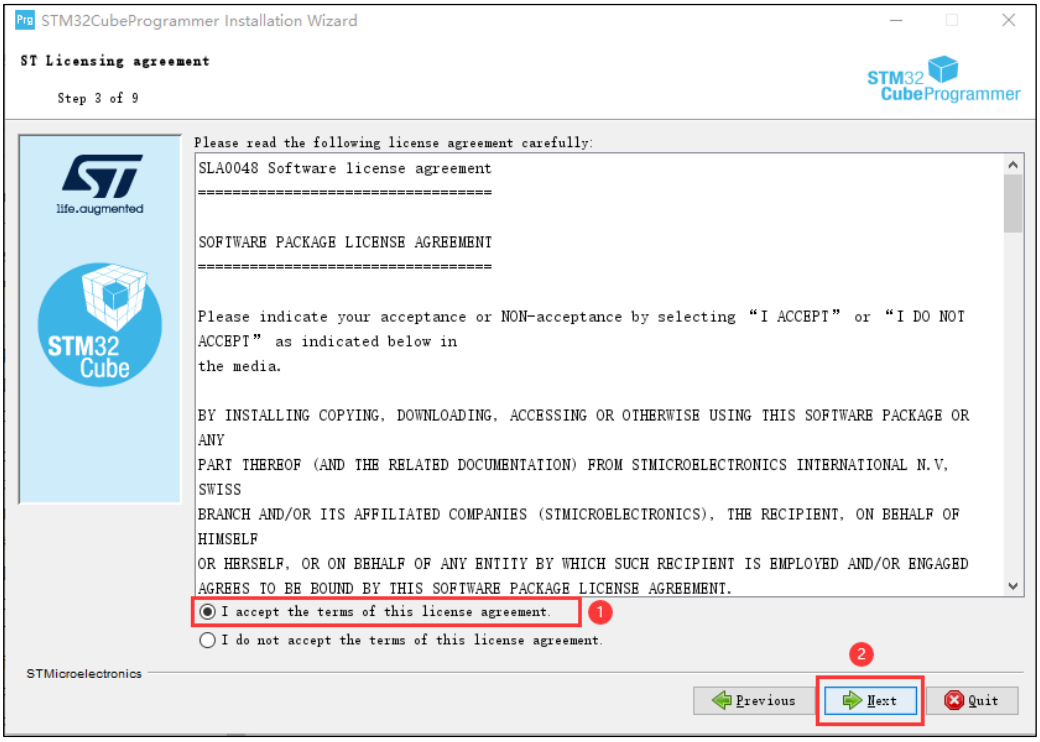


Figure 2.2-3 Agree to terms, click next

Have read terms, click next;

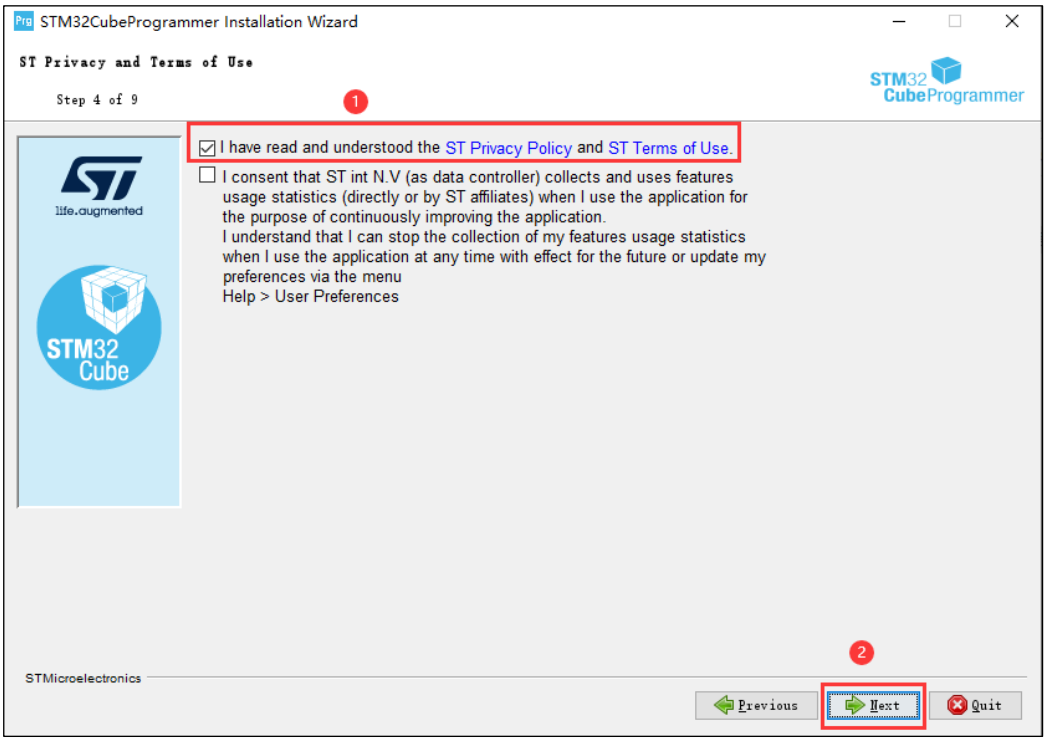


Figure 2.2-4 Read the terms, then click "Next".

Please select the installation path. You can modify the installation path as you wish, then click "Next".

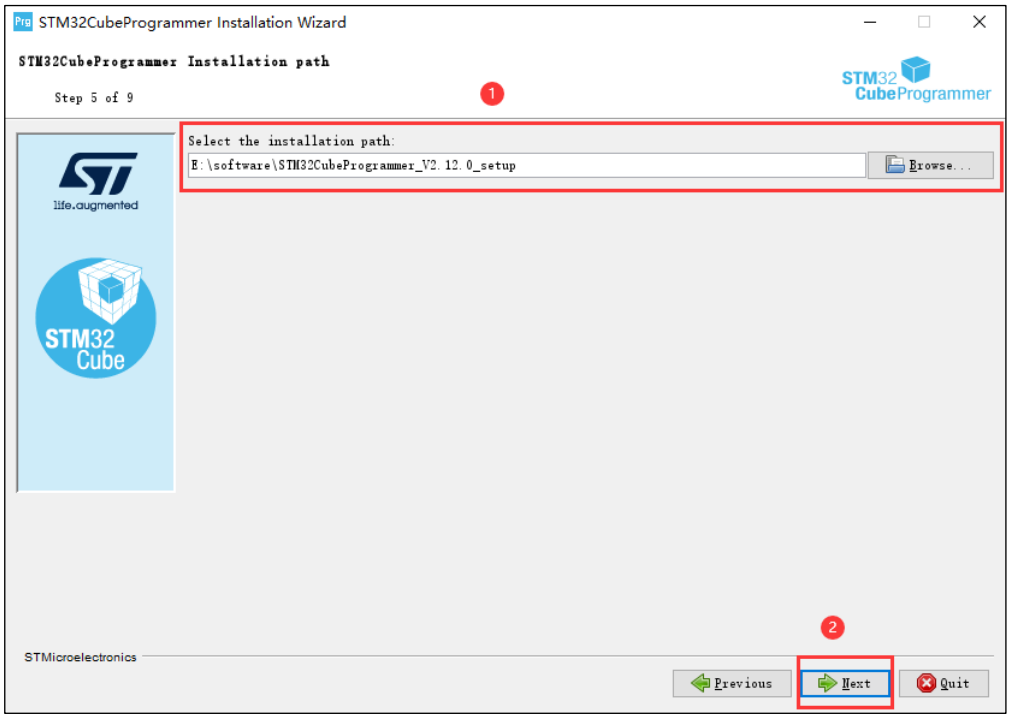


Figure 2.2-5 Select the installation path, then click "Next".

Check the options you need to install. We can simply choose the default settings and then click "Next".

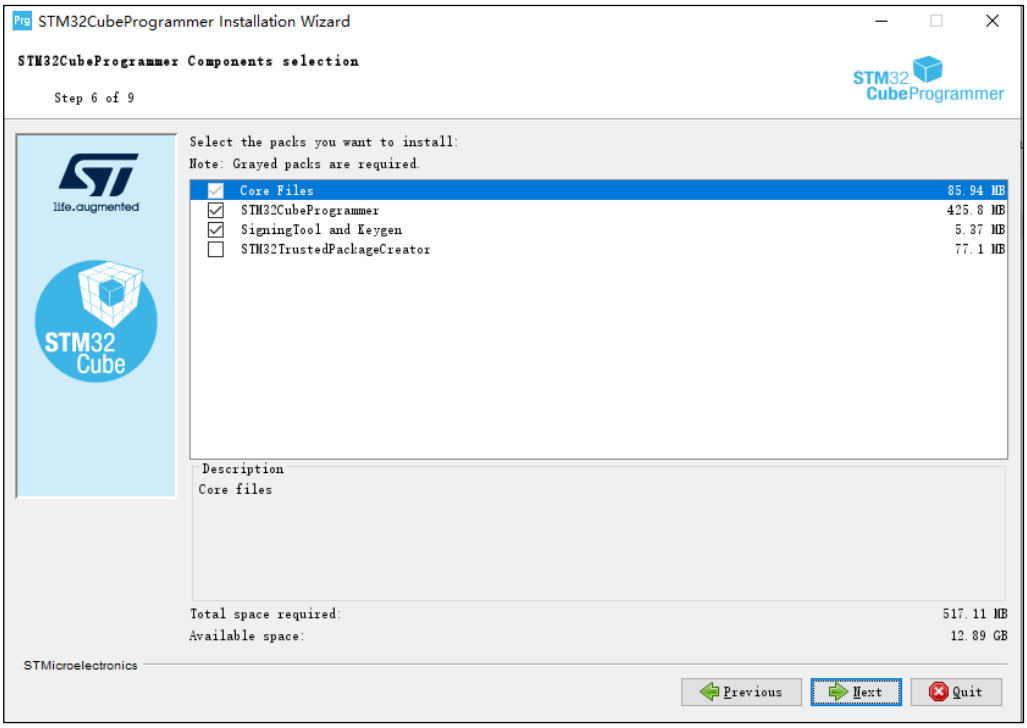


Figure 2.2-6 Default installation options, click Next

Being installed.

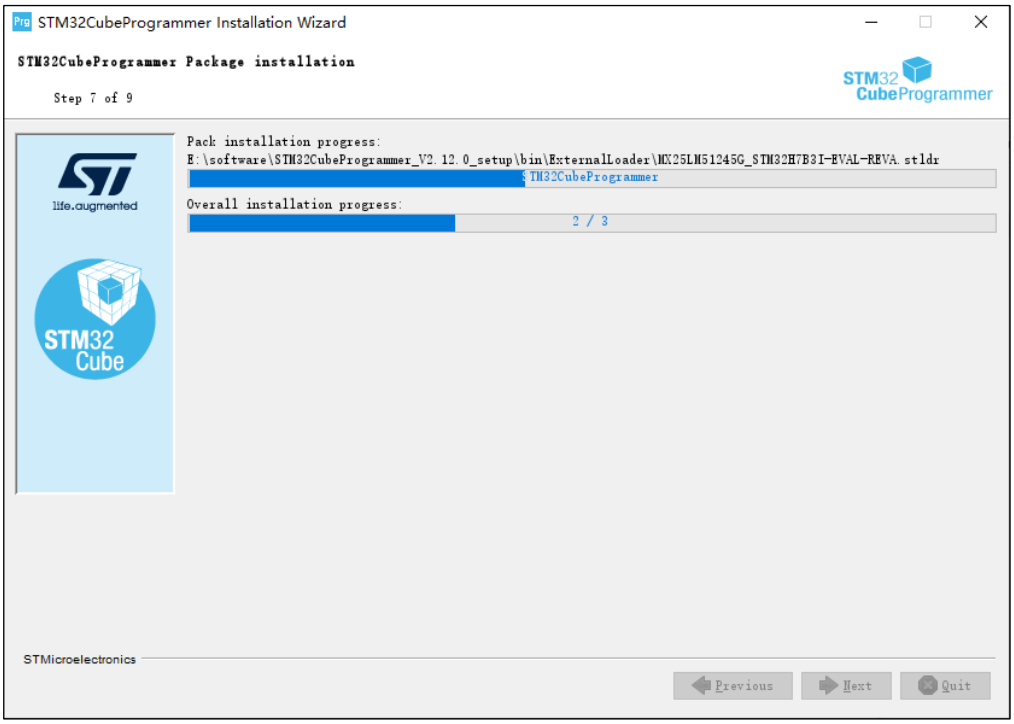


Figure 2.2-7 Being installed

During the installation process, a prompt will appear asking whether to install some drivers. We choose to always install this driver.

Click on the "Next" button on the driver installation interface.

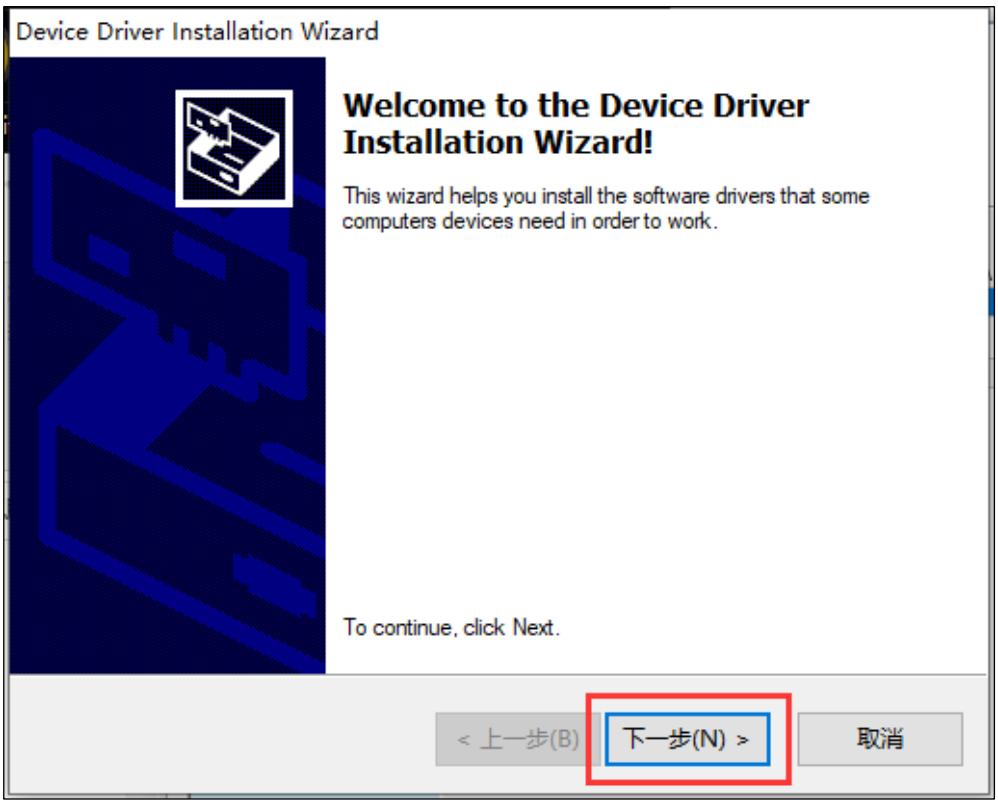


Figure 2.2-8 Click "Next" to install this driver.

The device driver installation is complete. Click "Finish".

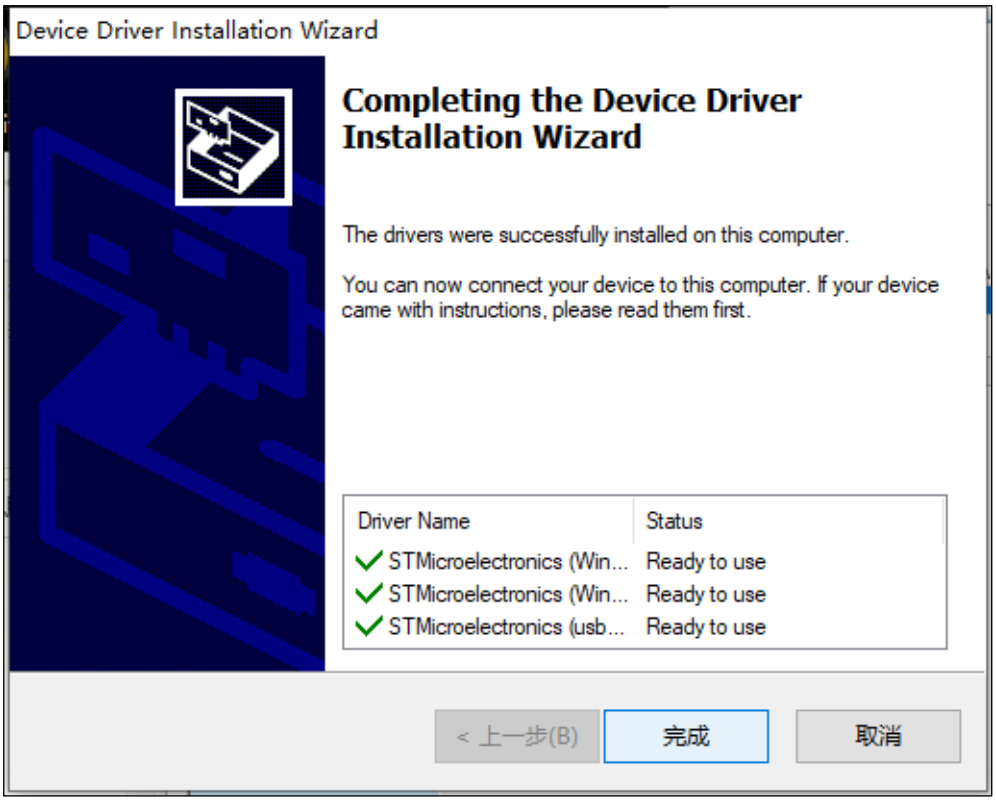


Figure 2.2-9 The installation has been completed.

After the installation is finished, click "Next".

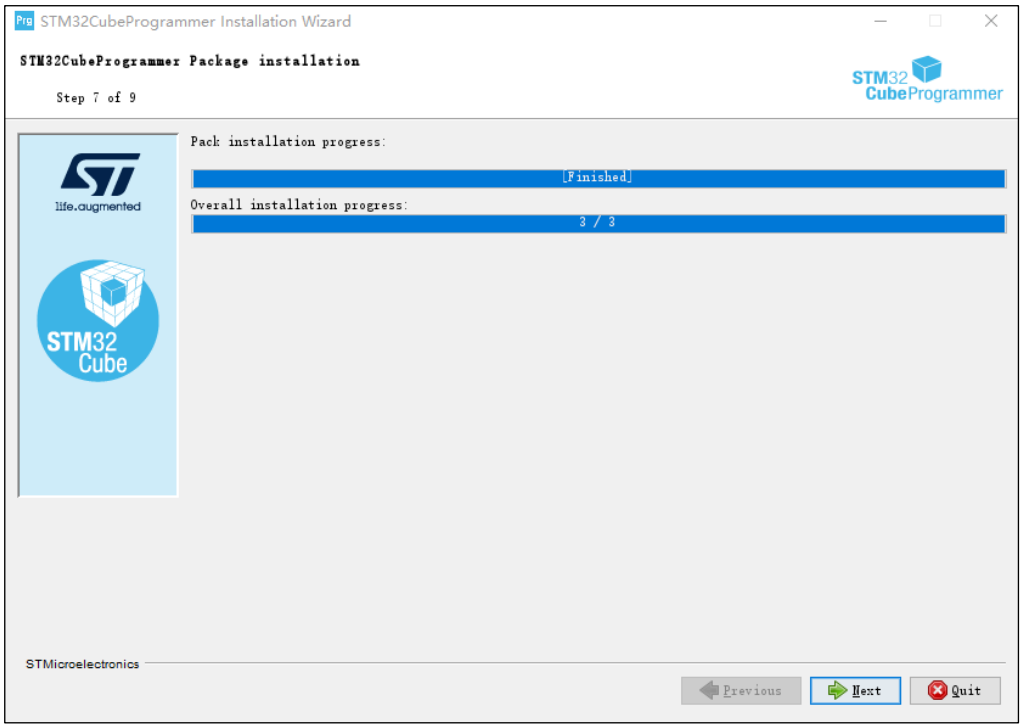


Figure 2.2-10 Installation status of STM32CubeProgrammer

By default, a desktop shortcut will be created. Click "Next".

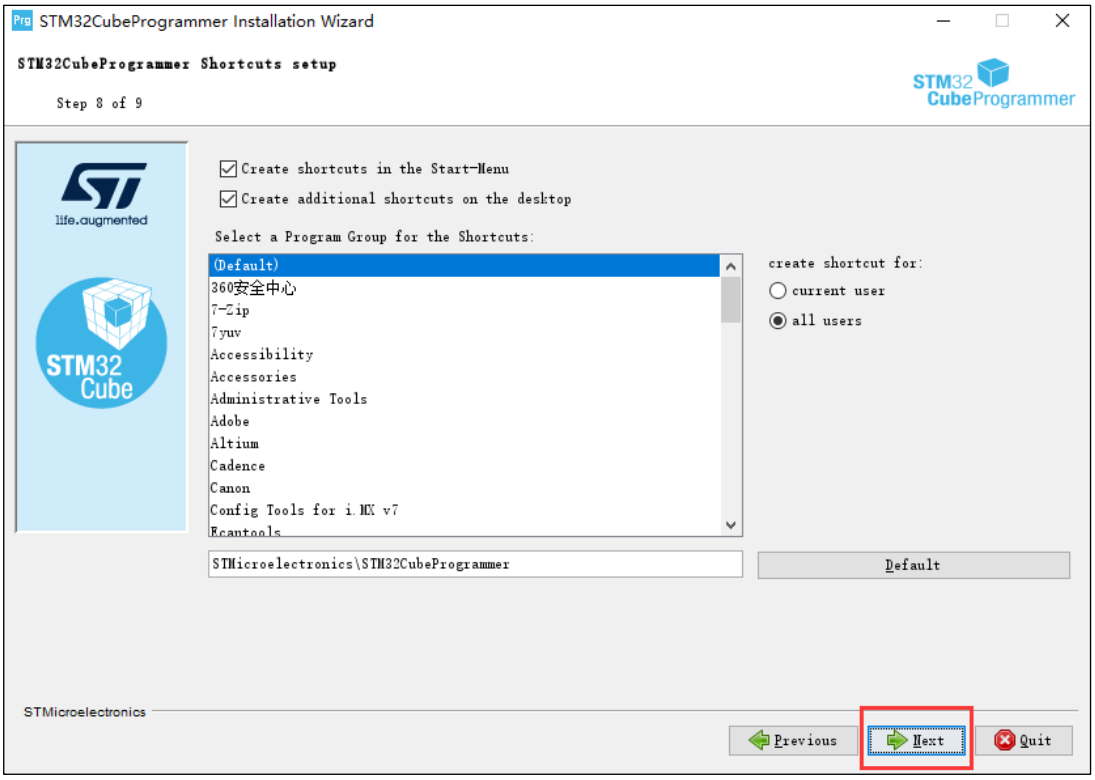


Figure 2.2-11 Click "Next" to generate the shortcut for STM32CubeProgrammer.

Click "Done" to complete the installation.

Figure 2.2-12 Installation completed

The shortcut for desktop generation is as follows.

Figure 2.2-13 The generated desktop shortcut

Double-clicking the STM32CubeProgrammer shortcut will open the following interface, indicating that the installation is complete and the software is running normally.

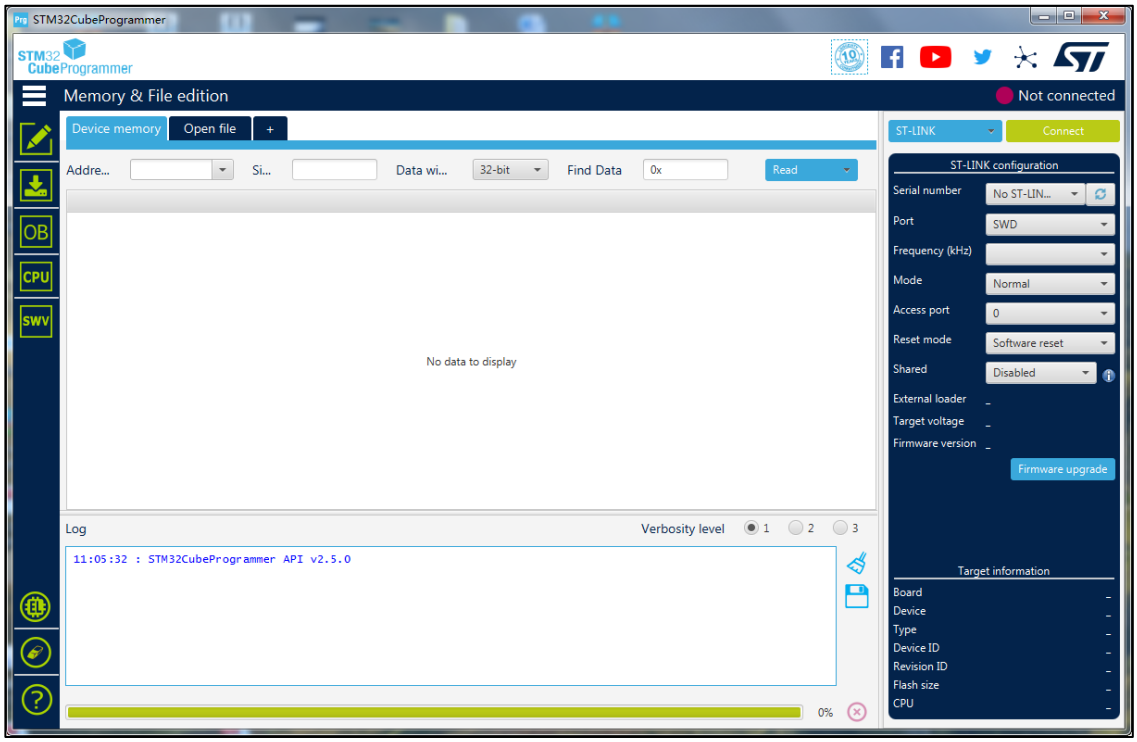


Figure 2.2-14 The interface after starting STM32CubeProgrammer

2.2.2 Using STM32CubePro to burn the system to eMMC on Windows

After plugging in the power supply for the ATK-DLMP135 development board, set the jumpers to 000 (USB mode), connect one USB Type-C data cable to the USB_OTG on the baseboard, and then connect it to the USB 3.0 interface on the PC (usually the one inside with blue color is the USB 3.0 interface, do not connect to the USB 2.0 interface, otherwise the burning process will be very slow! After actual testing, if the burning time is very long, it indicates that your USB 3.0 interface may not be a true USB 3.0 interface, it is actually a USB 2.0 interface). Connect the other USB Type-C data cable to the USB_TTL on the baseboard (serial port is not necessary to connect, the purpose is to observe some print information). Then power on the development board.

Double-click the desktop icon of STM32CubeProgrammer to open the STM32CubeProgrammer interface. Follow the steps below, select the device type as USB, click Refresh Devices, if it cannot be refreshed, power on again (note: do not connect this USB device to the Ubuntu virtual machine). As shown in the following picture, when USB1 appears, it indicates success!

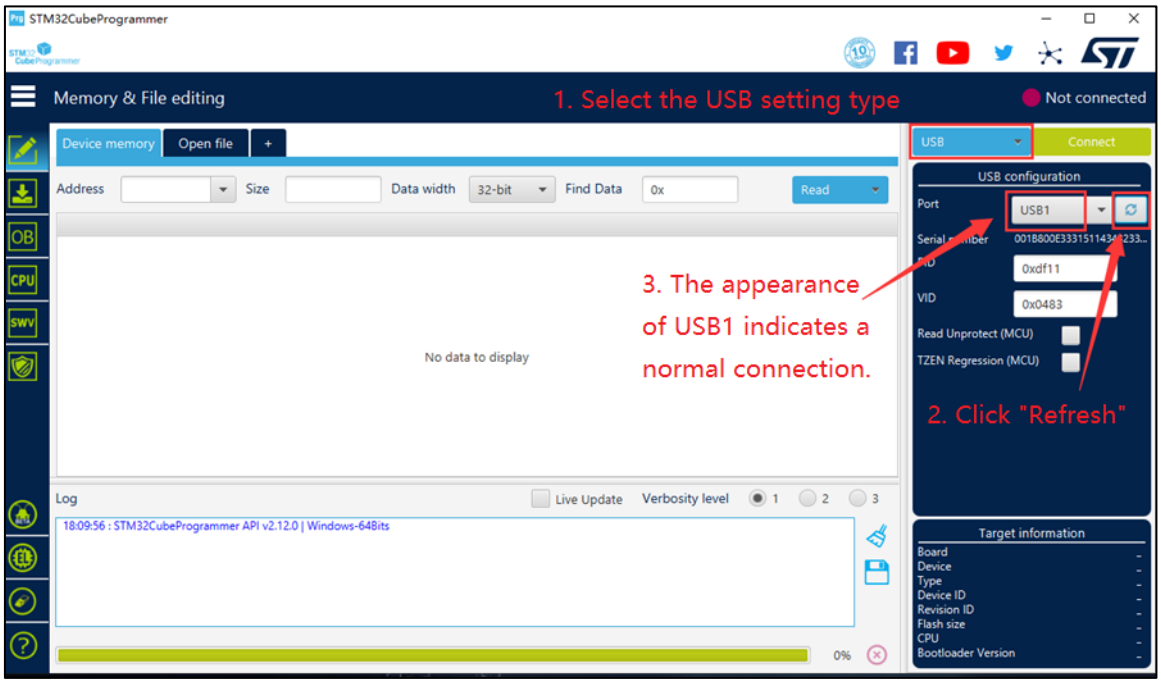


Figure 2.2-15 Identify USB1

Follow the steps in the figure to open the tsv configuration file burned onto the eMMC. The network drive path is STM32MP135 development board -> Development board CD A - Basic data -> 8_system_image -> 1_Factory_system_image -> 01 -> STM32CubeProg firmware burning package -> flashlayout_atk-image -> optee. The STM32CubeProgrammer will read this configuration file and burn the system onto the eMMC.

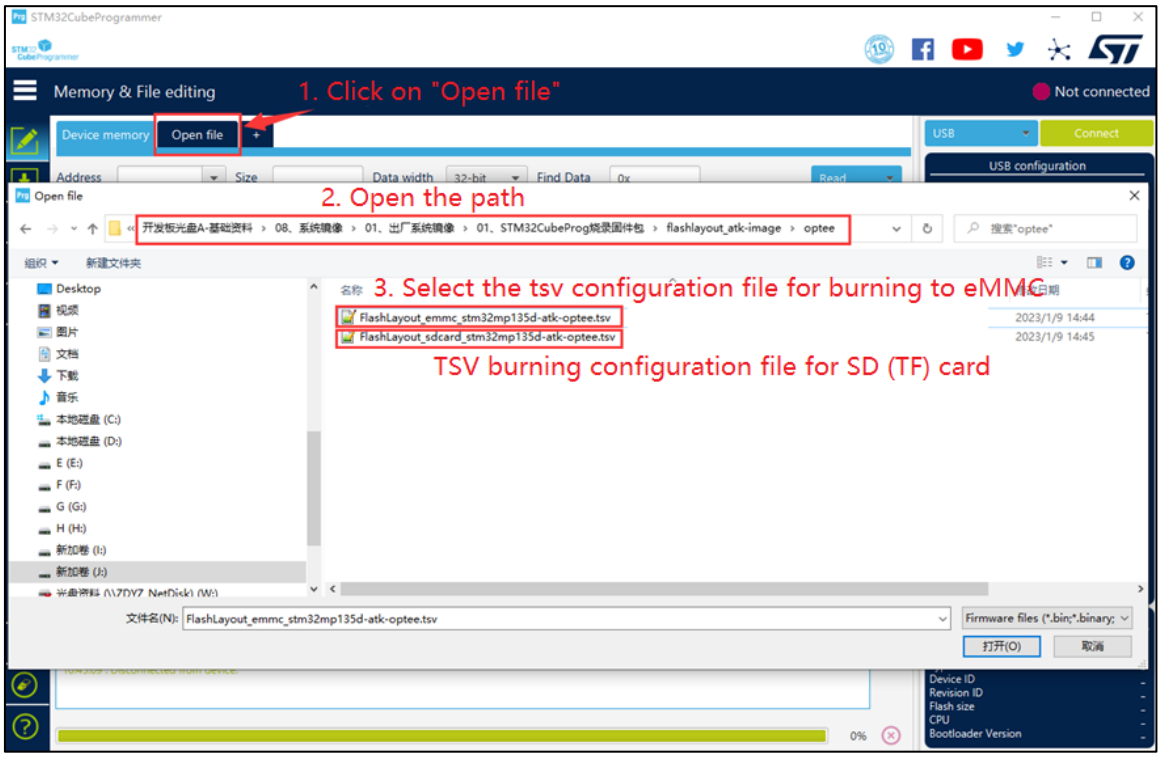


Figure 2.2-16 Select the configuration file to be burned onto the eMMC

Browse the directory where the firmware is burned. The path is: STM32MP135 development board -> Development board CD A - Basic materials -> 8_system_image -> 2. Factory system image -> 1. STM32CubeProg firmware burning package. (Users often mistakenly locate the directory. Please be careful not to select the wrong firmware path for your own.)

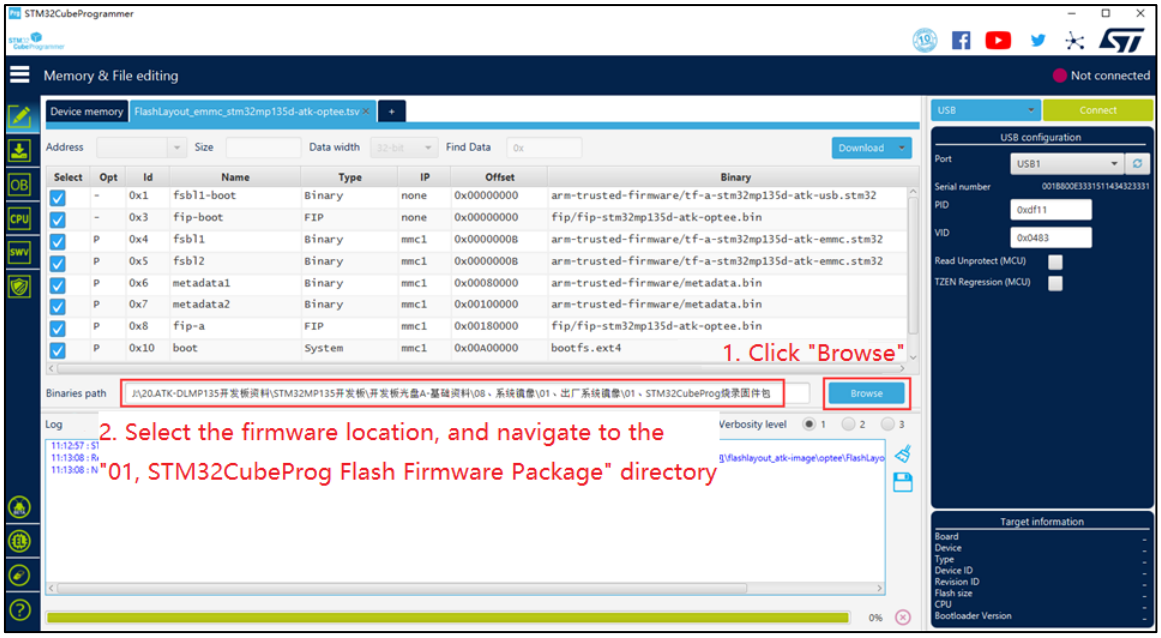


Figure 2.2-17 Browse the directory where the burning system is located.

Confirm the connection information and simply click "Download".

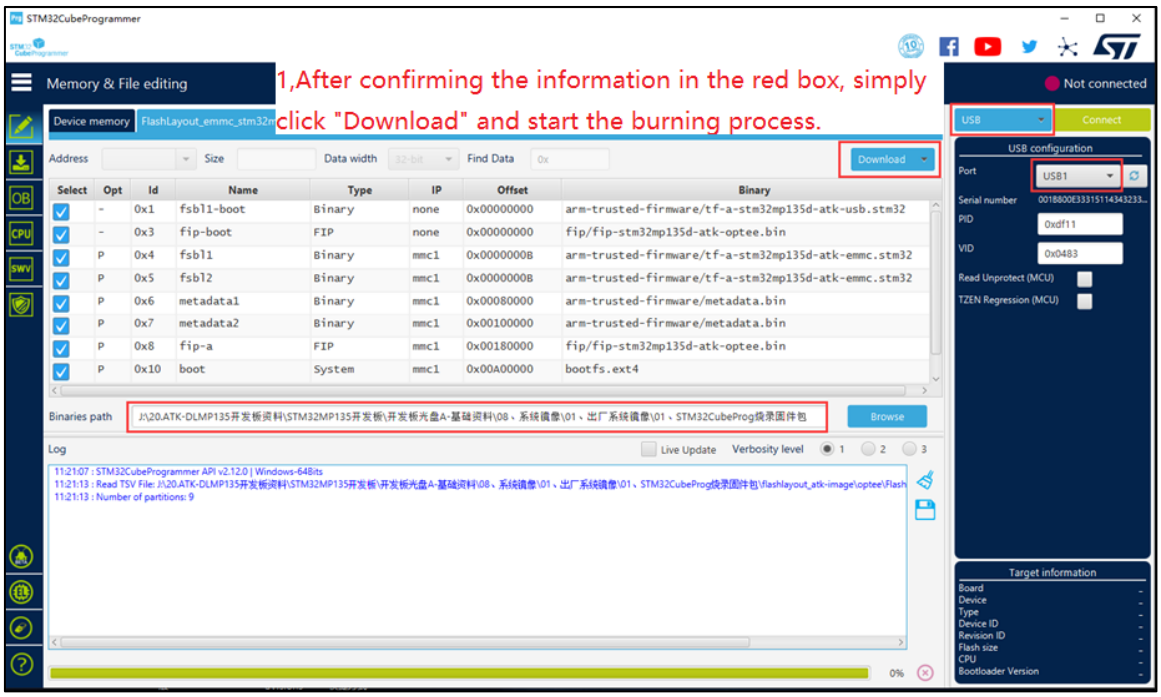


Figure 2.2-18 Click to download and start the burning process.

The burning process is in progress. During the burning process, the development board will be reconnected. Note: If you are using an Ubuntu virtual machine, do not allow the development board to connect to the Ubuntu virtual machine.

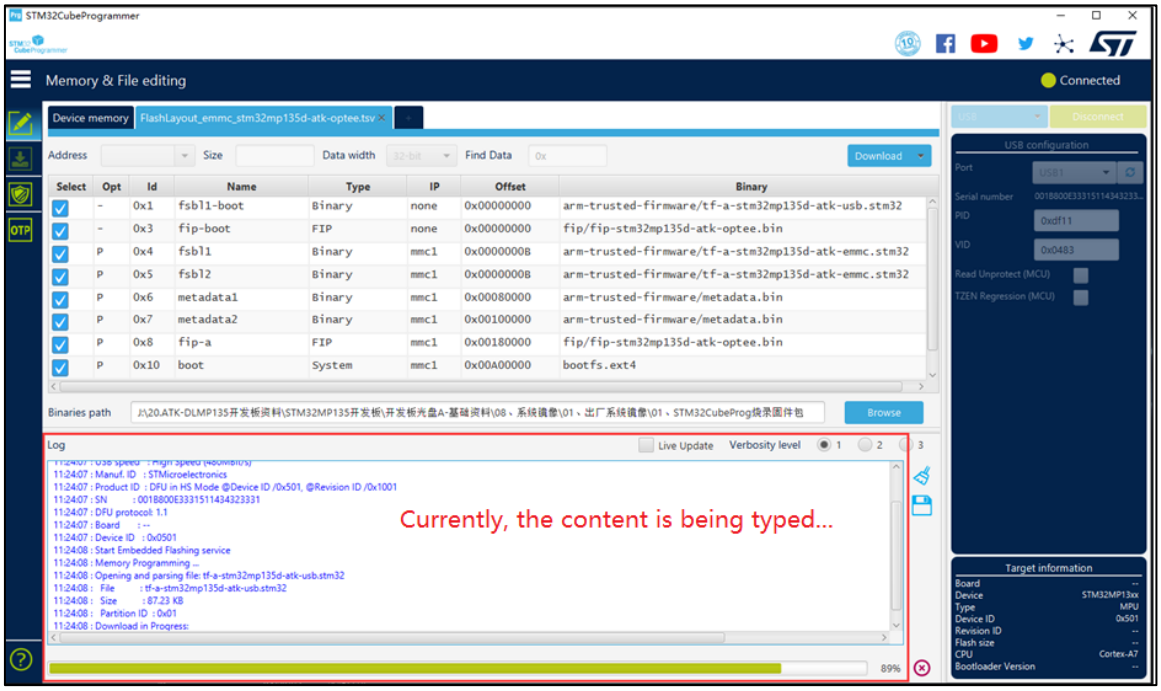


Figure 2.2-19 Currently in the process of burning

The burning process uses the PC (computer) USB 3.0 interface, and the burning speed is relatively fast. It takes approximately several minutes to burn the entire system. If the user's computer does not have a USB 3.0 interface, using USB 2.0 will be quite slow (possibly several tens of minutes).

Here is a reminder: When the system needs to be burned multiple times and the user does not want to update the board file system, we can edit the FlashLayout_emmc_stm32mp135d-atk-optee.tsv file and modify the configuration information for burning the file system (the last line) by changing "P" to "PE", which means to skip and not burn.

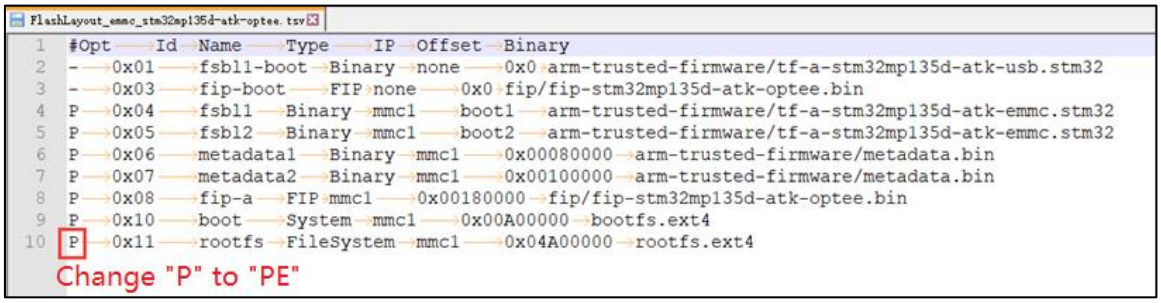


Figure 2.2-20 Skip the file system writing process.

The writing is completed as shown in the figure below. Set the dip switch to 010 and the eMMC will start.

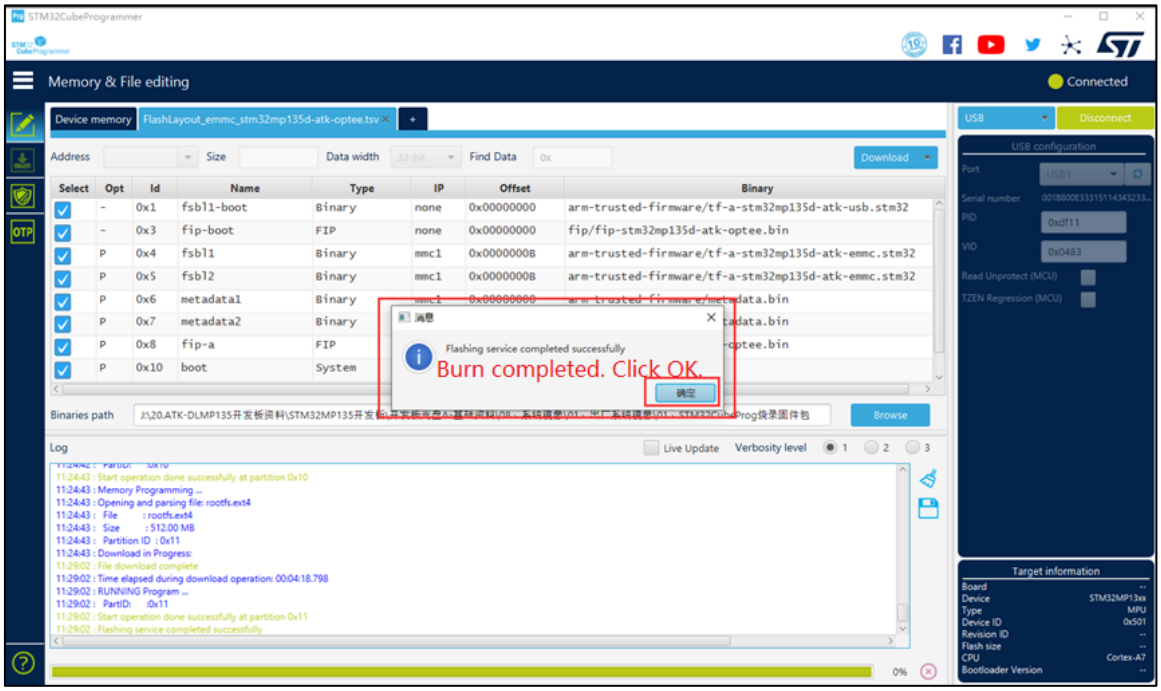


Figure 2.2-21 Burn completed

2.2.3 Windows uses STM32CubePro to burn the system onto the TF (SD) card.

Please refer to the method in Section 2.2.2 for burning onto the eMMC, and select the FlashLayout_sdcard_stm32mp135d-atk-optee.tsv configuration file for burning onto the TF card. Remember that when burning onto the TF card, you need to insert the TF card on the baseboard and ensure that the power jumper cap on the baseboard is switched to power the TF card instead of powering the WIFI module.

After the burning is completed, set the jumpers to 101, and the SD card (TF card) will boot up.

2.3 Writing the system to the development board under Linux

The previous section introduced how to use the STM32CubeProgrammer tool in the Windows system to write the system. Then, is it possible to perform system writing under the Linux system? The answer is yes. For users who need to write the system in Ubuntu, please follow the steps below.

2.3.1 Installing STM32CubeProgrammer under Linux

This section will install STM32CubeProgrammer in the Ubuntu 18.04 system. Other system versions can be tried on your own.

First, copy the 5_tools\02, and en.stm32cubeprg-lin-v2-12-0.zip compressed file from the 05, development tools, and ST official development tools to any directory in the Ubuntu system, for example, the author created a "tool" directory specifically for storing or installing tool software.

Execute the following command to decompress the file.

```
unzip en.stm32cubeprg-lin-v2-12-0.zip
```

```
atk@atk:~/Desktop/mp135/tool$ ls
en.stm32cubeprg-lin-v2-12-0.zip
atk@atk:~/Desktop/mp135/tool$
atk@atk:~/Desktop/mp135/tool$ unzip en.stm32cubeprg-lin-v2-12-0.zip
Archive:  en.stm32cubeprg-lin-v2-12-0.zip
   creating: jre/
   creating: jre/bin/
```

Figure 2.3-1 Extract the file

Just run the "SetupSTM32CubeProgrammer-2.12.0.linux" installation directly.

sudo ./SetupSTM32CubeProgrammer-2.12.0.linux

// It is recommended that the user also add "sudo". Just like the environment set by the editor, adding "sudo" will default to installing in the "/usr/local/" directory.

```
atk@atk:~/Desktop/mp135/tool$
atk@atk:~/Desktop/mp135/tool$ ls          The decompressed files
en.stm32cubeprg-lin-v2-12-0.zip  SetupSTM32CubeProgrammer-2.12.0.exe
jre                              SetupSTM32CubeProgrammer-2.12.0.linux
atk@atk:~/Desktop/mp135/tool$
atk@atk:~/Desktop/mp135/tool$ sudo ./SetupSTM32CubeProgrammer-2.12.0.linux
[sudo] password for atk:
./SetupSTM32CubeProgrammer-2.12.0.linux     Carry out the installation
/home/atk/Desktop/mp135/tool
chdir /home/atk/Desktop/mp135/tool/.
current dir /home/atk/Desktop/mp135/tool
Command line arguments:
```

Figure 2.3-2 Carry out the installation

The installation process is basically the same as that for Windows in Section 2.2.1. Please refer to Section 2.2.1. Generally, just follow the instructions to "Next", and the installation location can be set according to the default installation location of the software. Therefore, the installation steps will not be repeated here.

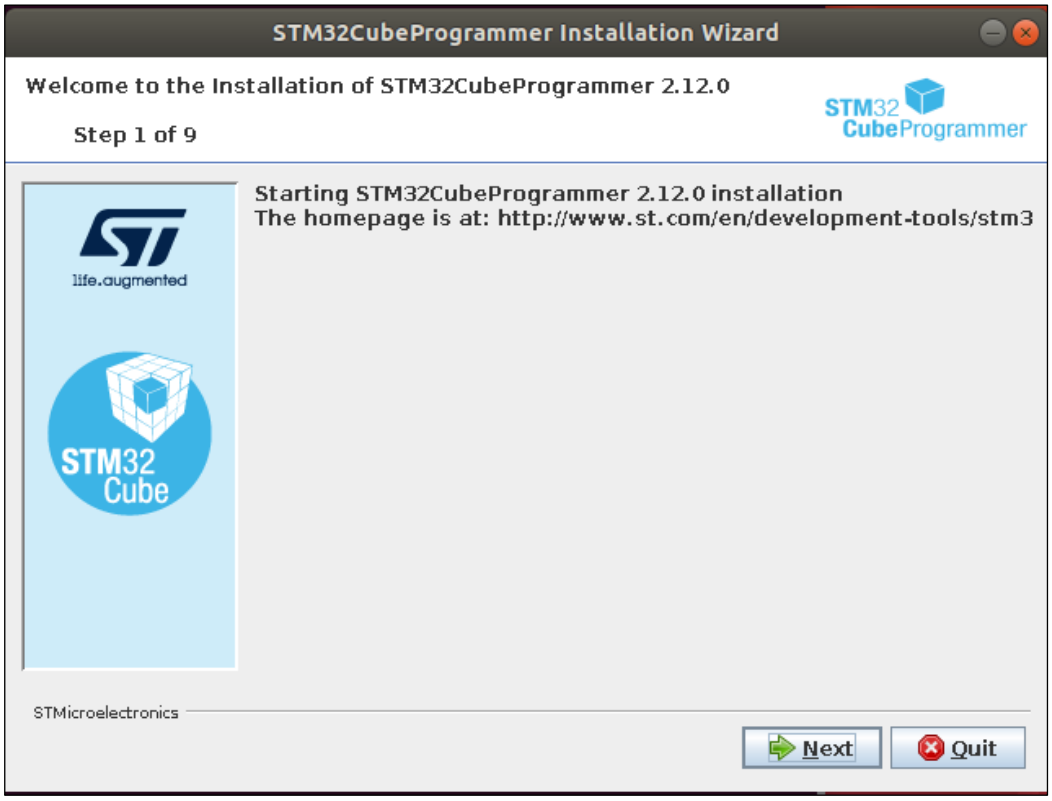


Figure 2.3-3 Installation Interface

After the installation is completed, there are two methods to open the STM32CubeProgrammer interface.

Method 1: Open STM32CubeProgrammer via the icon

After the installation, an icon will be generated on the desktop as shown in the figure below. Double-click this icon (in the Ubuntu 18.04 environment), and a pop-up window will appear, prompting us to select "Trust & Launch" (Trust and Launch).

Figure 2.3-4 Unchanged desktop icons

The icons will change as follows. Please keep reading.

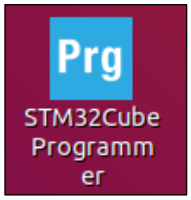


Figure 2.3-5 Changed desktop icons

Then a "Usage statistics" window will pop up. Select "No, thanks" and it will be done. Then the STM32CubeProgrammer interface will automatically pop up.

Figure 2.3-6 Use statistics

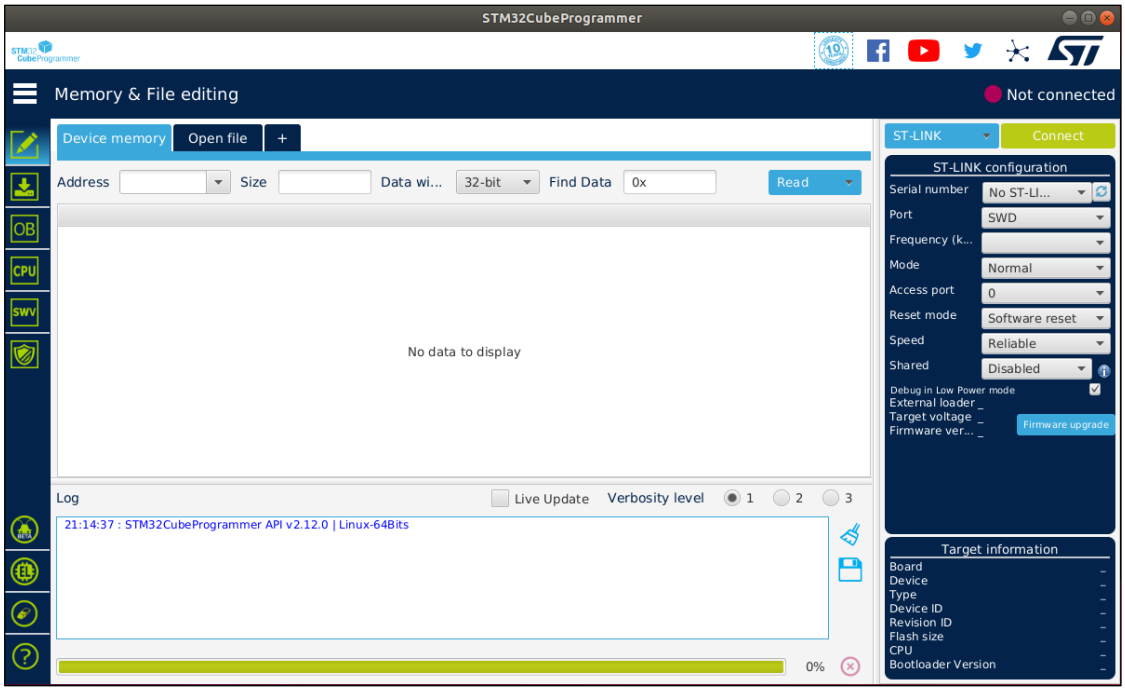


Figure 2.3-7 STM32CubeProgrammer interface

Method 2: Open STM32CubeProgrammer through the command line

Configure the environment variables of STM32CubeProgrammer so that we can use commands to open STM32CubeProgrammer from the command line.

Edit the /etc/profile file.

```
sudo vi /etc/profile
```

Add the following content at the end of /etc/profile, then press Esc after inputting, and finally press :wq to save and exit.

```
export PATH=$PATH:/usr/local/STMicroelectronics/STM32Cube/STM32CubeProgrammer/bin/
```

```
atk@atk: ~/Desktop/mp135/tool
File  Edit  View  Search  Terminal  Help

if [ "${PS1-}" ]; then
  if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

export PATH=$PATH:/usr/local/STMicroelectronics/STM32Cube/STM32CubeProgrammer/bin/
:wq
```

Figure 2.3-8 Write the environment variables in the last line.

After the configuration is completed, enable the environment variables (you need to source /etc/profile before each startup).

```
source /etc/profile
```

```
atk@atk:~/Desktop/mp135/tool$ source /etc/profile
atk@atk:~/Desktop/mp135/tool$
```

Figure 2.3-9 Enable the environment variables

Then we can use the command in the terminal to open STM32CubeProgrammer.

```
STM32CubeProgrammer
```

At this point, the installation of STM32CubeProgrammer is complete.

When using the USB port on Ubuntu to connect the STM32 device, the libusb1.0 software package needs to be installed as well.

```
sudo apt-get install libusb-1.0.0-dev
```

To use the ST-LINK tool or connect via USB DFU to the target, it is necessary to copy the rule file located in the Drivers/rules directory within the STM32CubeProgrammer installation directory to the /etc/udev/rules.d directory in Ubuntu. If the user installed STM32CubeProgrammer using the default installation directory, simply use the following command to copy the rule file to the /etc/udev/rules.d directory in Ubuntu.

```
sudo cp /usr/local/STMicroelectronics/STM32Cube/STM32CubeProgrammer/Drivers/rules/*.* /etc/udev/rules.d/
```

By following these two steps, the STM32CubeProgrammer tool will be able to properly identify the USB target device.

Plug in the power supply for the ATK-DLMP135 development board, set the dip switch to 000 (USB mode), connect a USB Type-C data cable to the USB_OTG interface on the bottom board, and then connect it to the USB 3.0 interface on the PC (computer) (usually the one with blue inside is the USB 3.0 interface, do not connect it to the USB 2.0 interface, otherwise the burning process will be very slow!) 。 Another USB Type-C data cable is connected to the USB_TTL on the baseplate (the

serial port is not necessary for connection, but it is used to observe some printing information). Then power on the development board.

Note: The Ubuntu virtual machine needs to be set to support USB 3.0. Follow the steps in the following picture, select USB 3.0.

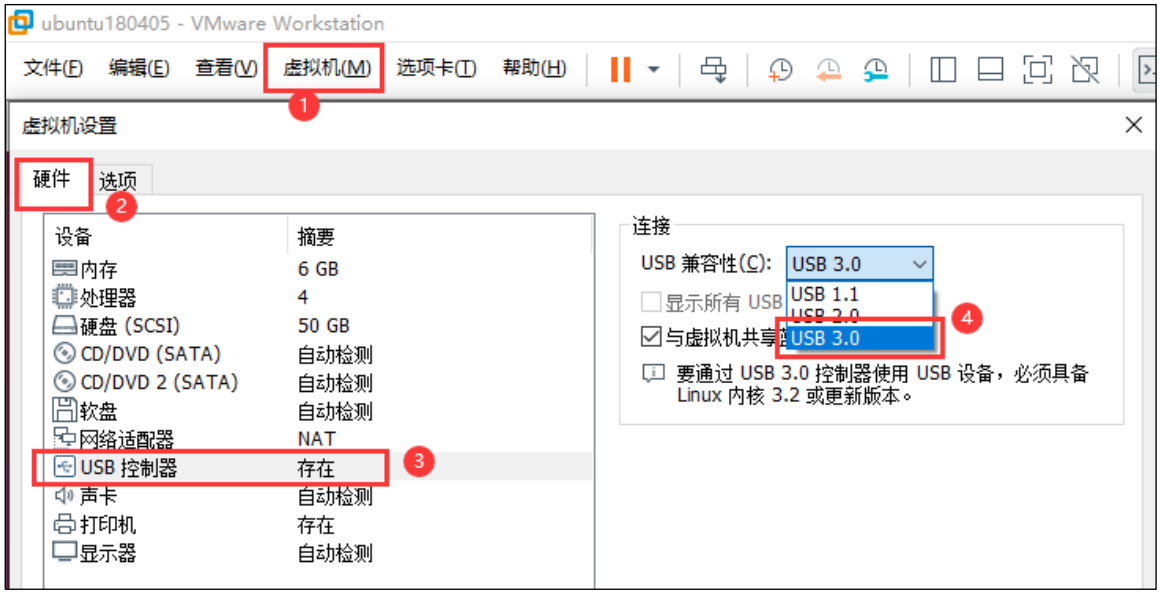


Figure 2.3-10 The virtual machine is selected to be compatible with USB 3.0.

When the Ubuntu 18.04 virtual machine pops up the following window, we need to connect the STMicroelectronic DFU in HS... (The editor has installed the DFU driver of STM32CubeProgrammer on Windows. If the user has not installed it, it may not be named this way) to the virtual machine, select to connect to Ubuntu 180405 (your Ubuntu host name). Then click Confirm.

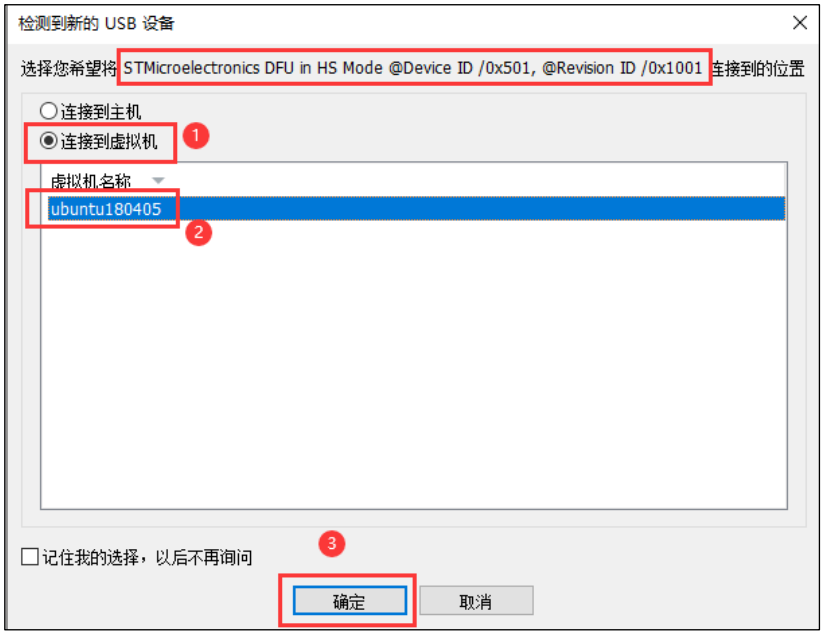


Figure 2.3-11 Connect the device to the Ubuntu virtual machine

If the above window doesn't pop up, it's okay. Just click the "Connect (Disconnect from Host Machine) (C)" option for the STMicroelectronic DFU in HS... device in the virtual machine (M), and the development board device can be connected to the virtual machine.

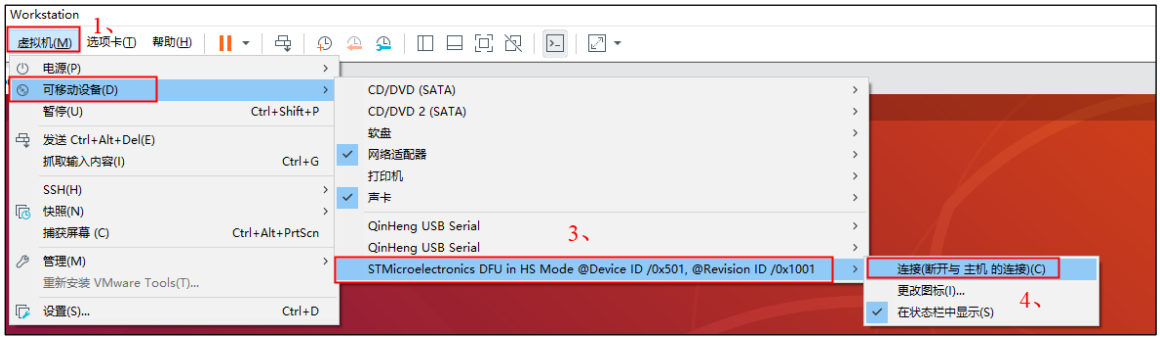


Figure 2.3-12 Connect the device to the Ubuntu virtual machine.

Double-click the desktop icon of STM32CubeProgrammer to open it, or enable the environment variables in the terminal and then input "STM32CubeProgrammer" to open STM32CubeProgrammer. Follow the steps below: Select the device type as USB, click Refresh Devices. If it fails to refresh, you can power on the development board again. As shown in the following picture, if USB1 appears, it indicates that the development board has been successfully connected!

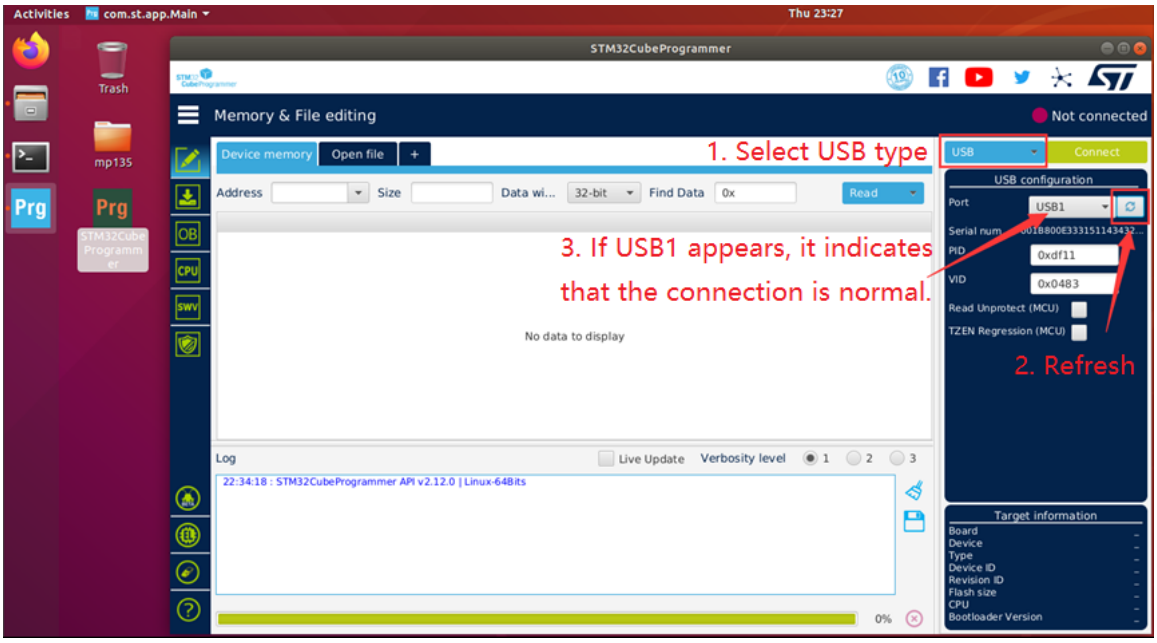


Figure 2.3-13 The connection of the development board was successful, and USB1 was displayed.

2.3.2 In Linux, use the STM32CubePro to burn the system onto the eMMC.

After connecting to the development board according to the above operation, copy the network drive path to the development board CD-A - basic data -> 8_system_image -> 01, factory system image -> 01, and STM32CubeProg firmware package to any directory in Ubuntu. As shown in the following figure, the editor copied it to the custom directory in Ubuntu.

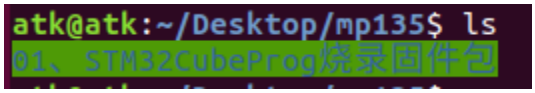


Figure 2.3-14 Copy the burn package to Ubuntu

Open the burn configuration file and click on the "Open file" button as shown in the picture.

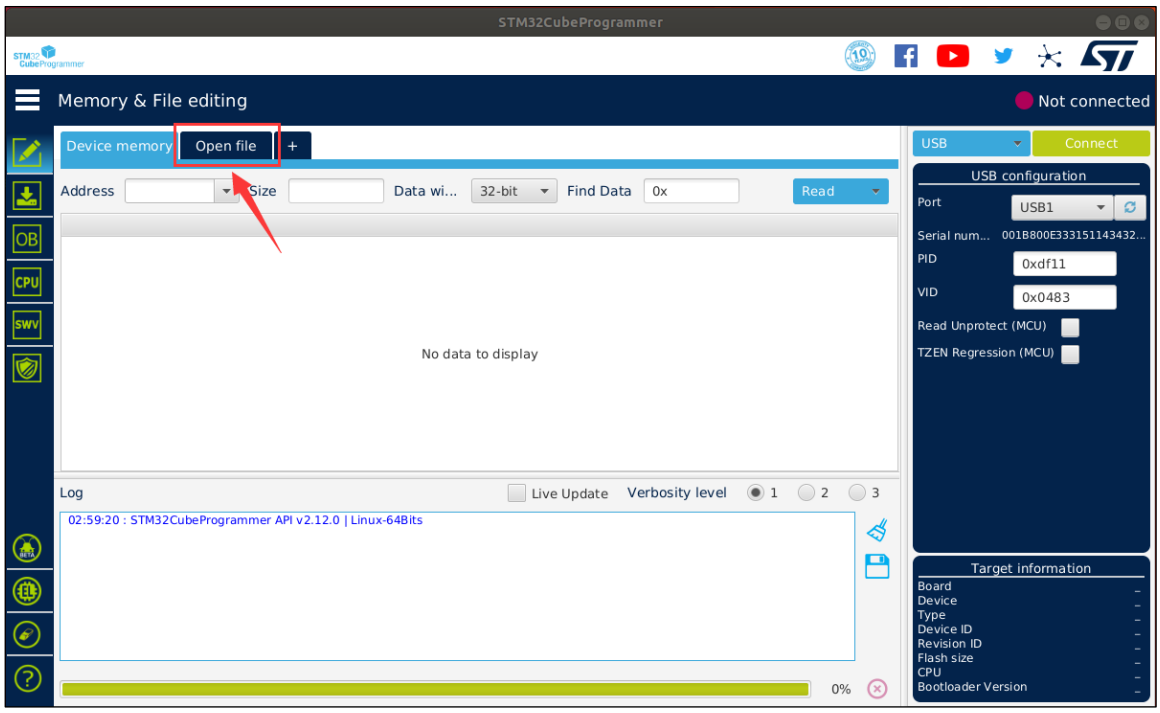


Figure 2.3-15 Open the burn configuration file

Locate the folder "01, STM32CubeProg Burn Firmware Package" that was copied above. Enter the flashlayout/optee directory, select the FlashLayout_emmc_stm32mp135d-atk-optee.tsv file, and then click "Open".

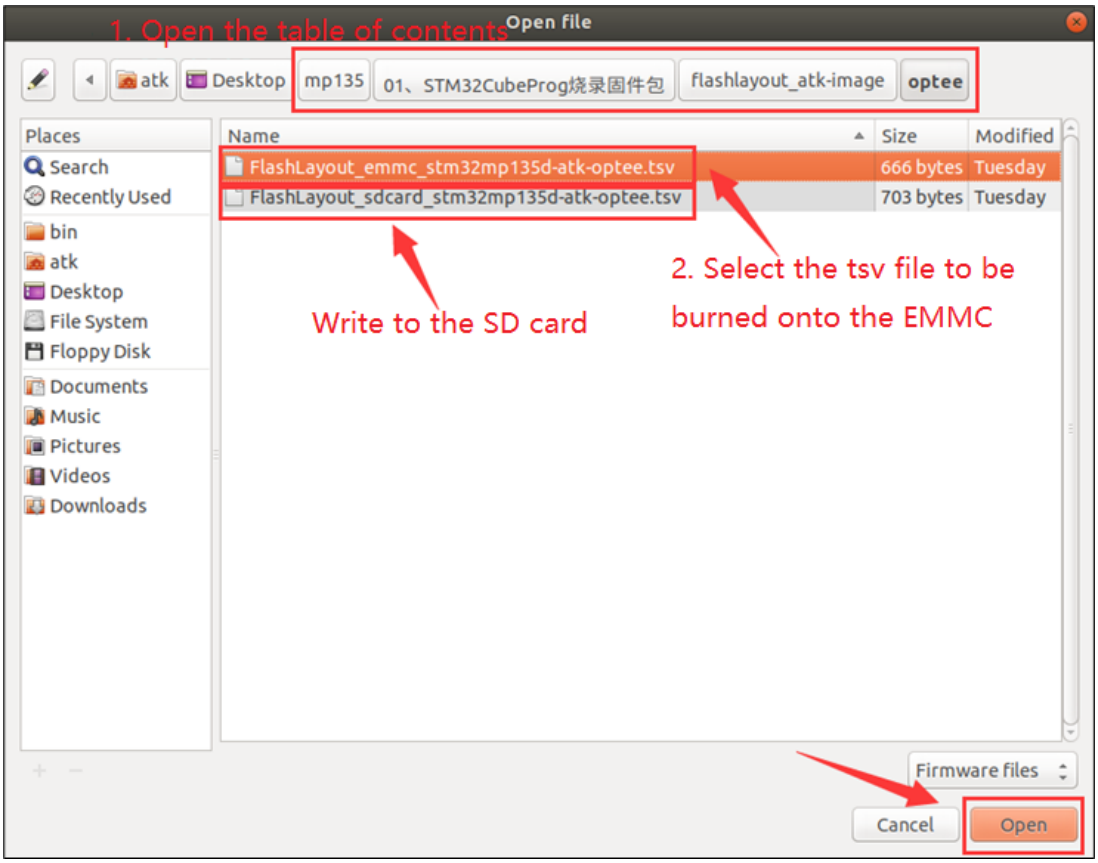


Figure 2.3-16 Select the tsv configuration file to be burned onto the eMMC.

After opening the tsv file, click "Browse" again and locate the "01, STM32CubeProg Burn Firmware Package" directory.

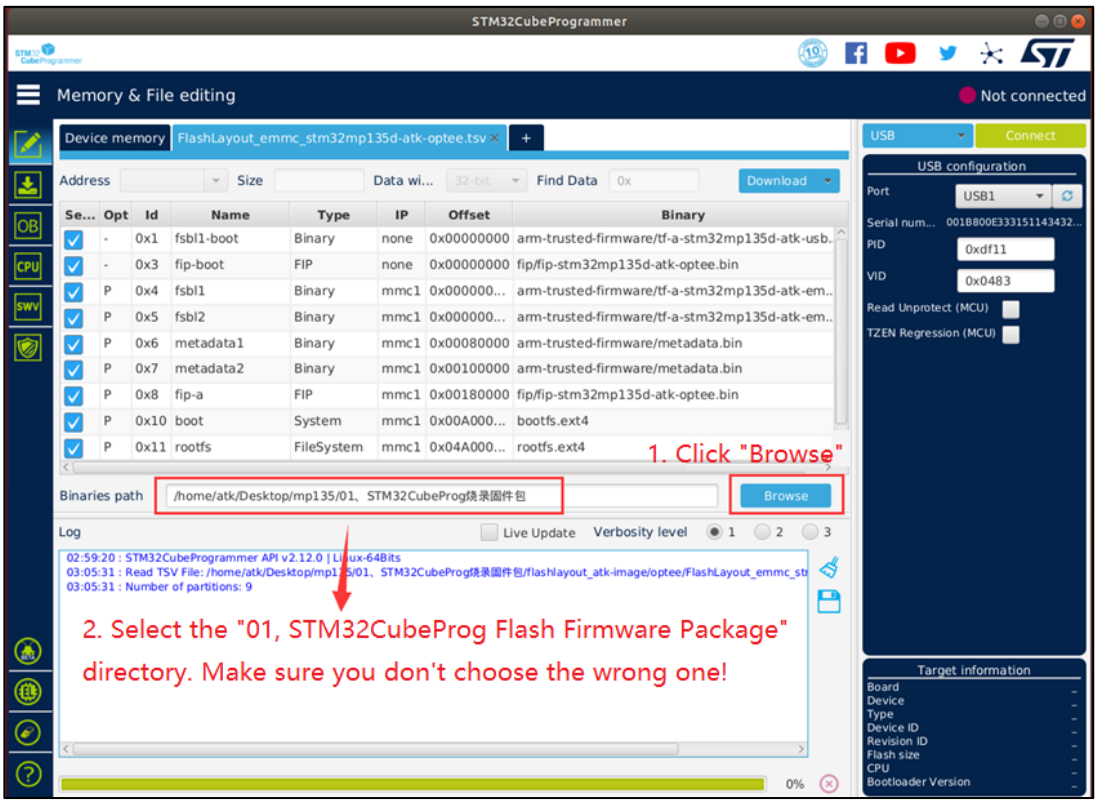


Figure 2.3-17 Select the location for the burning system

Confirm that the USB type has been selected, that the development board has been connected to the Ubuntu virtual machine, and then click "Download".

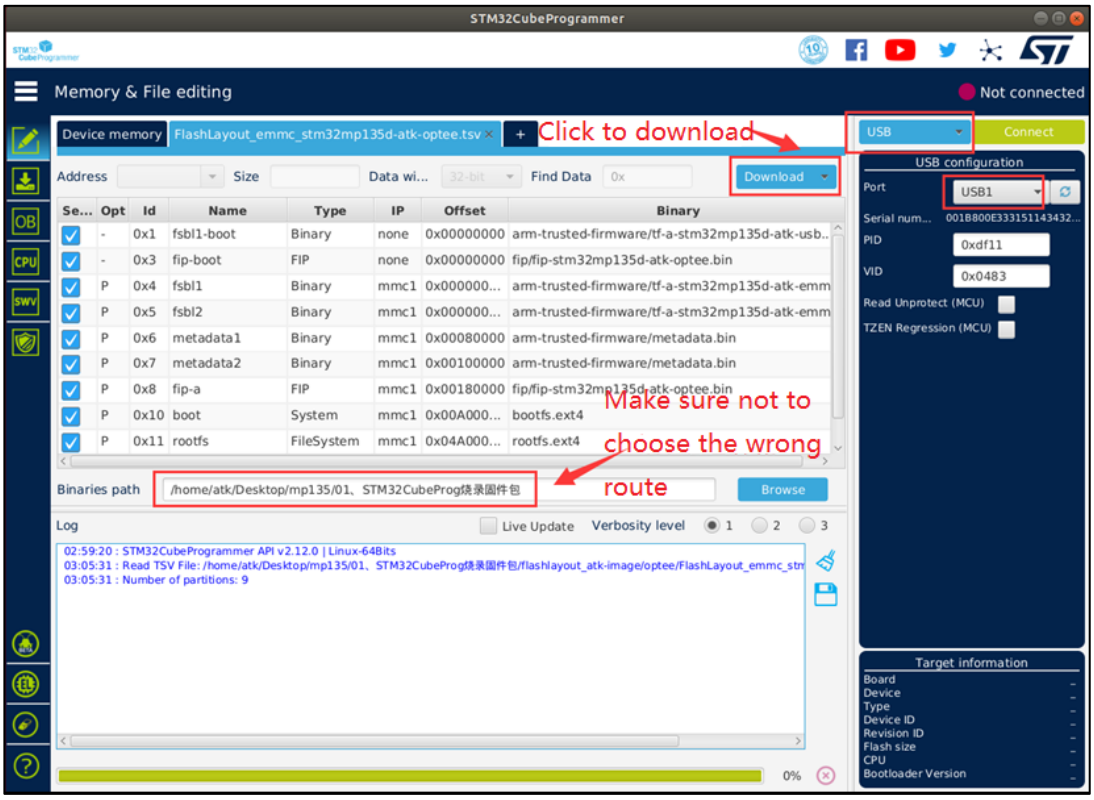


Figure 2.3-18 Click "Download" and start the burning process.

The burning process is underway and the log information of the burning process is being displayed.

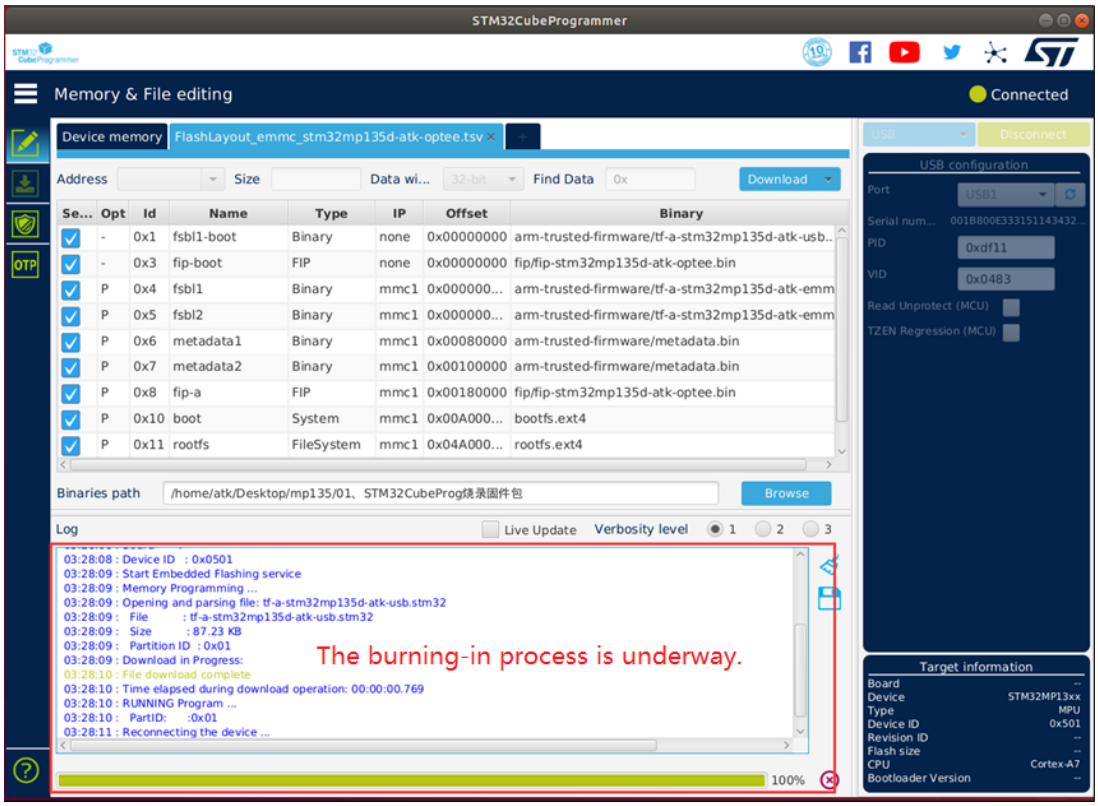


Figure 2.3-19 The burning process is in progress.

The burning is completed as shown in the figure below. Set the jumpers of the development board to 010, and select eMMC booting to proceed.

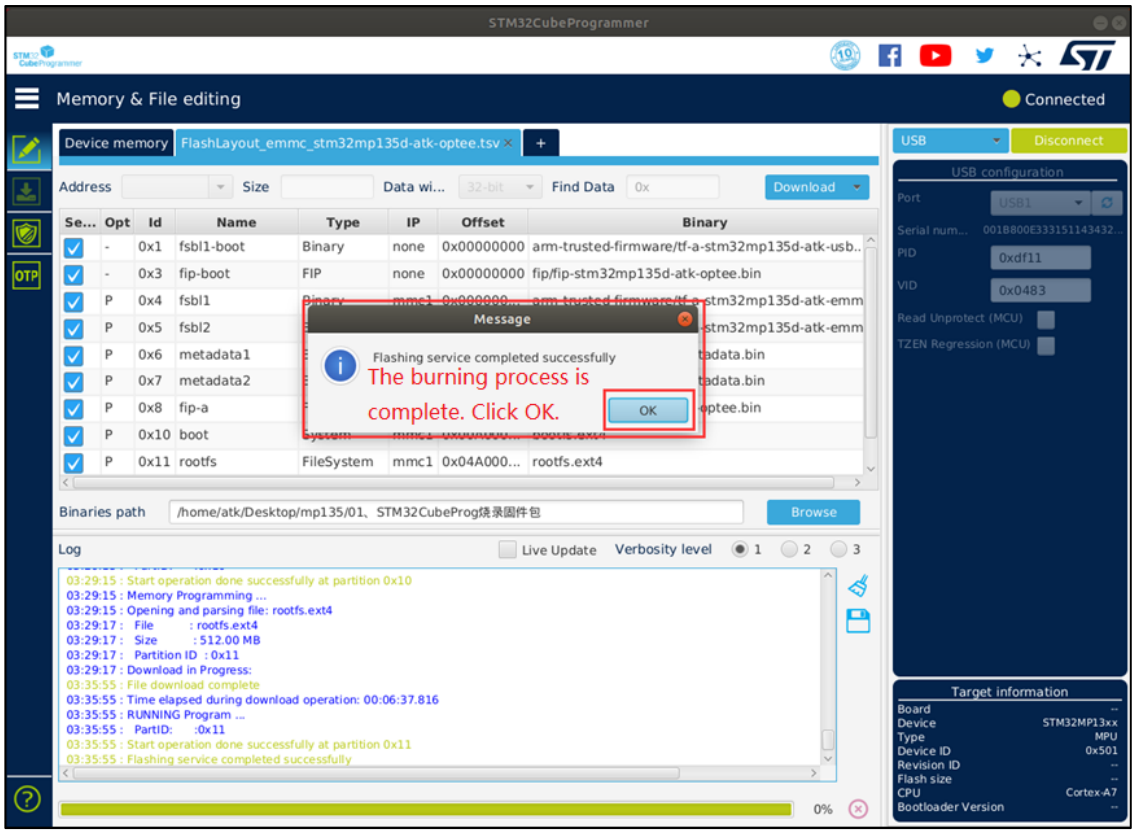


Figure 2.3-20 Burn completed

2.3.3 Linux uses STM32CubePro to burn the system onto a TF (SD) card.

Please refer to the method in Section 2.3.2 for burning onto eMMC, and select the FlashLayout_sdcard_stm32mp135d-atk-optee.tsv configuration file for burning onto the TF card. Remember that when burning onto the TF card, you need to insert the TF card on the baseboard and ensure that the power supply jumper cap on the baseboard is switched to power the TF card instead of powering the WIFI module.

After the burning is completed, set the jumpers to 101, and the SD card (TF card) will boot up.

Chapter 3. Preparations Before Using ATK-DLMP135

Here, we will introduce the precautions for using the development board. If these precautions are not followed, it may lead to the failure of your experiments or damage to the development board. Therefore, you must pay attention to them.

3.1 Precautions Before Powering On

- First, check if the DC power supply is plugged in. The development board has removed the USB_TTL serial port power supply function. If the power supply is not plugged in, only being powered by USB_TTL may result in insufficient power supply for the entire development board, causing problems such as LCD or HDMI not being displayed, LCD not being touchable, and 4G module not working properly.
- Before leaving the factory, the entire board was protected by acrylic board. However, in some cases, users may remove the acrylic board. At this time, it is necessary to pay attention to observing that the bottom board should not be placed on the miscellaneous table. Instead, a protective film should be placed under the bottom board. Otherwise, it is easy to come into contact with metal foreign substances, causing short circuits between two of the contacts under the bottom board. When using it for a long time, also pay attention to observing whether any foreign substances have fallen on the development board to prevent short circuits. Pay attention to waterproofing, moisture-proofing, dust-proofing, etc.

3.2 Serial Port Software Installation

Introduction:

Here, we will briefly introduce the installation of the CH340 USB serial port driver and the installation of the MobaXterm terminal. We will not provide a detailed installation tutorial as it is relatively basic.

- Install the CH340 USB serial port driver (for the computer to communicate with the development board via serial port, we need to install this driver on the development board's CD-ROM A - Basic Data -> 3_softwares -> CH340 Driver (USB Serial Port Driver) - Files for XP_WIN7 Common Use). Locate the file SETUP.EXE in the folder "03, Software -> CH340 Driver (USB Serial Port Driver) - Files for XP_WIN7 Common Use" on the development board's CD-ROM A. Double-click to run it, and then follow the pop-up window. Simply click "Install" and wait for the pre-installation or the "Installation Successful" window to appear. If the installation fails, please refer to the "Installation Failure Solution" folder.
- Install the MobaXterm terminal software (or install terminal software such as Xshell, SecureCRT). For MobaXterm, the network drive path is "STM32MP135 Development Board -> Development Board CD-ROM A - Basic Materials -> 3_softwares -> MobaXterm_Installer_v12.3.zip". Double-click to open this compressed file, wait for the decompression, and then simply double-click "MobaXterm_installer_12.3.msi" to install. The MobaXterm installation program will guide you through the installation process.

3.3 Setting of jumpers and login to the development board

The jumpers of the ATK-DLMP135 development board support the following boot modes. The following figure is the BOOT schematic diagram of the bottom board of the development board.

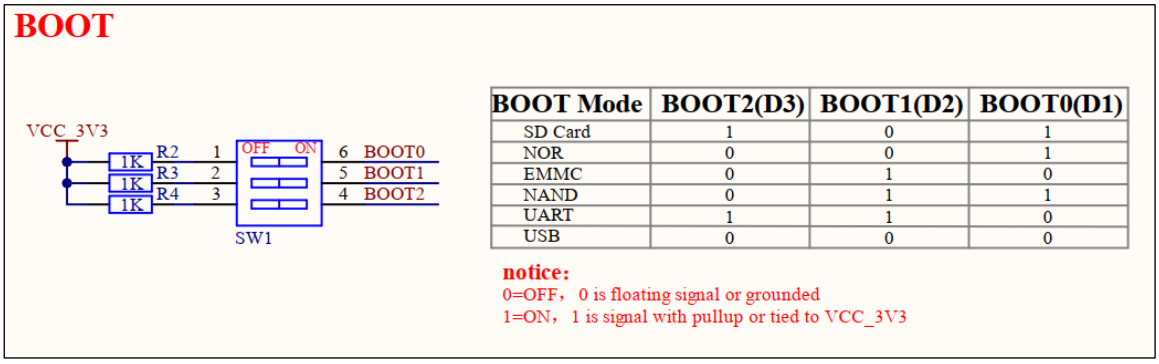


Figure 3.3-1 BOOT startup mode

As can be seen from the schematic diagram of the development board:

(1) The jumper for the SD Card startup mode is set to 101;

(2) The jumper for the EMMC startup mode is set to 010;

(3) The USB mode jumper is set to 000.

The ATK-DLMP135 development board uses a USB Type-C connection cable to connect to the USB_TTL interface on the baseboard, and the other end is connected to the USB interface of the PC (computer). The Linux system has been pre-written onto the eMMC storage on the core board when it was manufactured. Therefore, we will set the jumper to 010, as shown in the figure. The upward position of the jumper indicates "1" (ON), and the downward position indicates "1" (OFF).

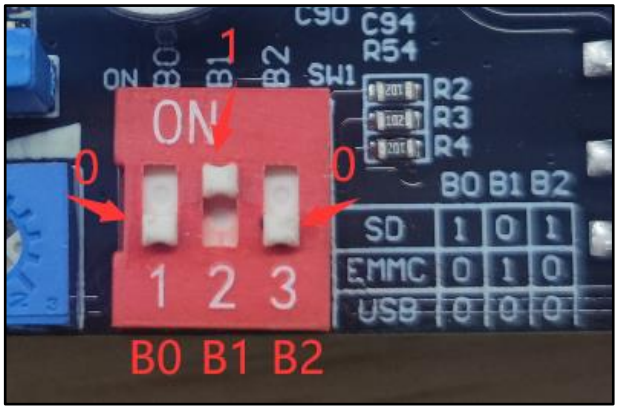


Figure 3.3-2 eMMC boot jumper method

Connect the DC12V DC power supply to the development board. In MobaXterm, select the serial port type and make the following settings. Please select the corresponding COM port (the COM port is the serial port number recognized by the PC for the development board USB_TTL).

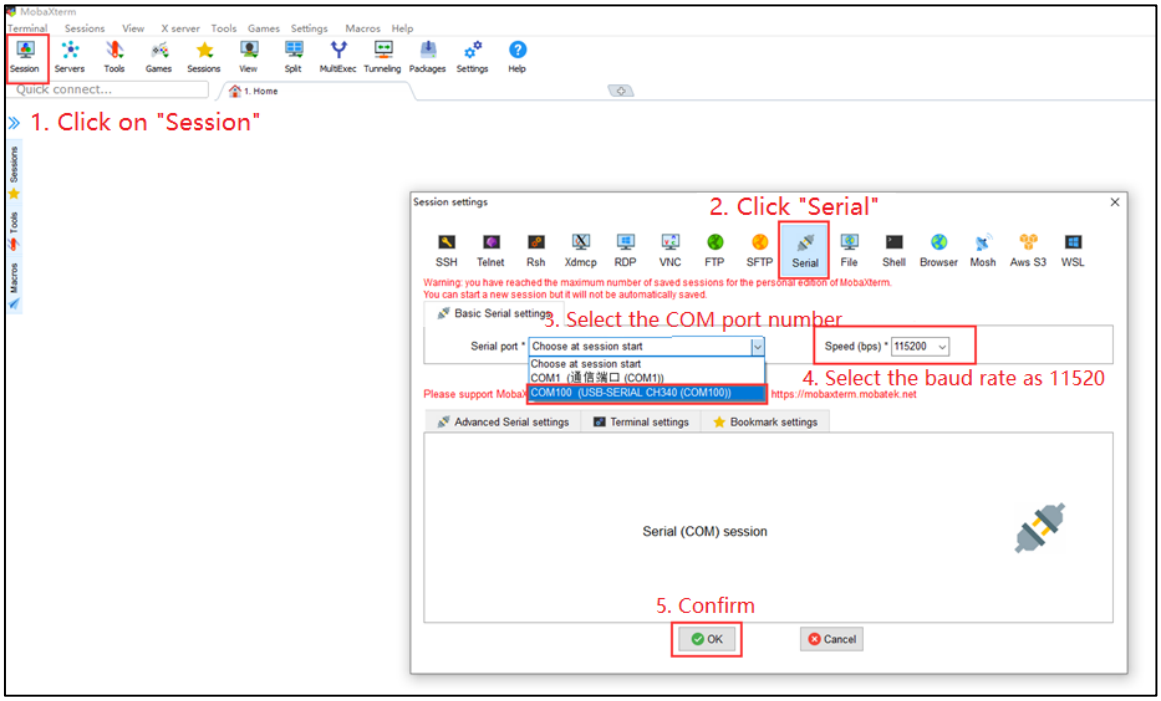


Figure 3.3-3 MobaXterm Serial Port Settings

After the development board starts in eMMC mode, the serial port terminal prints the running information of TF-A, OP-TEE, U-Boot and the kernel as follows.

```
U-Boot 2021.10-stm32mp-r1 (Feb 28 2023 - 15:31:27 +0800)

CPU: STM32MP135D Rev.Z
Model: STM32MP135D-ATK Discovery Board
Board: stm32mp1 in trusted mode (st,stm32mp135d-atk)
DRAM:  512 MiB
optee optee: OP-TEE: revision 3.16 (72845af2)
Clocks:
- MPU : 1000 MHz
- AXI : 266.500 MHz
- PER : 24 MHz
- DDR : 533 MHz
WDT:   Started with servicing (32s timeout)
NAND:  0 MiB
MMC:   STM32 SD/MMC: 0, STM32 SD/MMC: 1
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
invalid MAC address 0 in OTP 00:00:00:00:00:00
Net:
Error: eth2@5800e000 address not set.

Error: eth1@5800a000 address not set.

Error: eth1@5800a000 address not set.

Error: eth2@5800e000 address not set.
No ethernet found.

lcd_id  = 00
Hit any key to stop autoboot:  0
Boot over mmc1!
switch to partitions #0, OK
mmc1(part 0) is current device
261192 bytes read in 14 ms (17.8 MiB/s)
Scanning mmc 1:4...
Found /mmc1_extlinux/stm32mp135d-atk_extlinux.conf
Retrieving file: /mmc1_extlinux/stm32mp135d-atk_extlinux.conf
648 bytes read in 8 ms (79.1 KiB/s)
Select the boot mode
1:      stm32mp135d-atk
2:      stm32mp135d-atk-hdmi
3:      stm32mp135d-atk-wifi-bluetooth
Enter choice: 1:        stm32mp135d-atk
Retrieving file: /uInitrd
4588714 bytes read in 107 ms (40.9 MiB/s)
Retrieving file: /uImage
7782088 bytes read in 176 ms (42.2 MiB/s)
append: root=/dev/mmcblk1p5 rootwait rw console=ttySTM0,115200
Retrieving file: /stm32mp135d-atk.dtb
58851 bytes read in 10 ms (5.6 MiB/s)
## Booting kernel from Legacy Image at c2000000 ...
   Image Name:   Linux-5.15.24-g8e961c093355
   Created:      2023-02-28   7:35:43 UTC
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    7782024 Bytes = 7.4 MiB
   Load Address: c2000040
   Entry Point:  c2000040
```

Because the MAC address of the network interface was not set, U-Boot printed a prompt indicating that the network could not be found.

1. Use the standard atomic RGB LCD panel device tree for startup. The LCD panel is enabled by default when present, and disabled by default when absent.

2. To boot using the HDMI device tree, if the user connects an HDMI device, they need to select "2" on the terminal to load the HDMI device tree.

3. Use the WiFi & Bluetooth device tree to start, which is used to test the SDIO WiFi and Bluetooth modules. Select "3"

Figure 3.3-4 Serial port-printed U-Boot operation information

The factory file system was built by buildroot. After logging into the development board, the Qt interface still needs a few seconds to complete startup.

By default, there is no need to enter the login account password; directly access the system root directory. After logging in, the serial port terminal is as shown in the figure below.

```
Starting input-event-daemon: done
[root@ATK-DLMP135 /]#
[root@ATK-DLMP135 /]#
[root@ATK-DLMP135 /]#
```

Figure 3.3-5 Serial port terminal prints the information of logging into the development board.

Note: The factory file system defaults to supporting SSH service. When users log in using SSH, they need to enter the account as root and the password as root.

Chapter 4. Functional test

This chapter will introduce how to test the various interface functions of the ATK-DLMP135 development board. The interface function test instructions are based on the Buildroot root file system.

4.1 LED test, buzzer test

The ATK-DLMP135 development board is equipped with 1 user LED and 1 buzzer.

Enter the following instructions in the factory file system to control the status of the LED and the buzzer.

```
cat /sys/class/leds/sys-led/trigger
// View the current triggering mode of the LED and the supported triggering modes
echo none > /sys/class/leds/sys-led/trigger    // Change the trigger mode of the LED to "none".
echo 1 > /sys/class/leds/sys-led/brightness    // Light up LED0
echo 0 > /sys/class/leds/sys-led/brightness    // Turn off LED0
echo heartbeat > /sys/class/leds/sys-led/trigger
// Change the triggering method of the LED and set it to "heartbeat".
echo 1 > /sys/class/leds/beep/brightness       // Turn on the buzzer
echo 0 > /sys/class/leds/beep/brightness       //Turn off the buzzer
echo heartbeat > /sys/class/leds/beep/trigger
// Change the triggering method of the buzzer and set it to "heartbeat" (which represents the heartbeat sound).
echo none > /sys/class/leds/beep/trigger
// Change the triggering mode of the buzzer and set it to "none".
echo 0 > /sys/class/leds/beep/brightness       // Turn off the buzzer
```

4.2 Key Test

The ATK-DLMP135 development board is equipped with 1 user key named KEY.

Enter the following command in the factory file system to view the devices corresponding to the input events.

```
cat /proc/bus/input/devices
```

```
[root@ATK-DLMP135 ~]#cat /proc/bus/input/devices
I: Bus=0018 Vendor=0000 Product=0000 Version=0000
N: Name="generic ft5x06 (82)"
P: Phys=
S: Sysfs=/devices/platform/soc/4c004000.i2c/i2c-1/1-0038/input/input0
U: Uniq=
H: Handlers=event0
B: PROP=2
B: EV=b
B: KEY=400 0 0 0 0 0 0 0 0 0 0
B: ABS=2608000 3

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="gpio-keys"
P: Phys=gpio-keys/input0
S: Sysfs=/devices/platform/gpio-keys/input/input1
U: Uniq=
H: Handlers=kbd event1
B: PROP=0
B: EV=100003
B: KEY=40000 0 0 0

[root@ATK-DLMP135 ~]#
```

Figure 4.2-1 Button event

During the testing, the development board was connected to an LCD screen. From the above picture, we can see that at this time, event0 is a touchscreen event (such as the ft5x06 touch chip), and event1 is a key input event for the key. If the user does not connect a touchscreen, then event0 will be a key event. Therefore, event1 is not necessarily always a key event, and this is related to the devices connected to the user board.

If a screen is connected, you can use the od -x or hexdump command to view the key input information. Press Ctrl+c to terminate the command.

```
od -x /dev/input/event1
or: hexdump /dev/input/event1
```

If the user is not connected to an LCD screen, the instructions for testing the buttons are as follows:

```
od -x /dev/input/event0
or: hexdump /dev/input/event0
```

Pressing the "KEY" will display the following information, indicating that the key is functioning properly. If there is no response, you can try pressing the key several more times.

```
[root@ATK-DLMP135 ~]#od -x /dev/input/event1
0000000 51c6 63ad 2b6f 000f 0001 0072 0001 0000
0000020 51c6 63ad 2b6f 000f 0000 0000 0000 0000
0000040 51c7 63ad 7a9c 0002 0001 0072 0000 0000
0000060 51c7 63ad 7a9c 0002 0000 0000 0000 0000
0000100 51c8 63ad 191d 0006 0001 0072 0001 0000
0000120 51c8 63ad 191d 0006 0000 0000 0000 0000
0000140 51c8 63ad 9e2a 0008 0001 0072 0000 0000
0000160 51c8 63ad 9e2a 0008 0000 0000 0000 0000
^C
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#hexdump /dev/input/event1
0000000 51ee 63ad 76b7 000b 0001 0072 0001 0000
0000010 51ee 63ad 76b7 000b 0000 0000 0000 0000
0000020 51ee 63ad 45a0 000e 0001 0072 0000 0000
0000030 51ee 63ad 45a0 000e 0000 0000 0000 0000
0000040 51ef 63ad 1bc2 000f 0001 0072 0001 0000
0000050 51ef 63ad 1bc2 000f 0000 0000 0000 0000
0000060 51f0 63ad 088f 0003 0001 0072 0000 0000
0000070 51f0 63ad 088f 0003 0000 0000 0000 0000
^C
[root@ATK-DLMP135 ~]#
```

Figure 4.2-2 Printed key event

4.3 LCD test

This section is only applicable to users of development boards that are equipped with a screen.

Since the ALIENTEK screen has pull-up resistors at the highest bits R7\G7\B7 (used as ID identification codes) on the R,G,B data lines, the factory system will identify the resolution of different screens based on the default high or low level states of R7\G7\B7 during the startup process, and then load the corresponding touch driver. The RGB screens used by ALIENTEK have five types, as shown in the table below:

Screen size	Touch chip	Screen ID identification code
4.3 inch screen（480x272）	gt9xx	0x00
4.3 inch screen（800x480）	gt9xx/gt1151	0x04
7 inch screen（800x480）	ft5x06/cst340(cst340 compatible ft5x06)/gt911	0x01
7 inch screen（1024x600）	ft5x06/cst340(cst340 compatible ft5x06)/gt911	0x02
10 inch screen（1280x800）	gt9xx	0x05

4.3.1 Touch test

Just like the method in Section 4.2, after the development board is powered on, we input the command "cat /proc/bus/input/devices" to check the touch events corresponding to the touch screen. We can also directly touch and click, or slide on the screen to test if the screen is functioning properly.

4.3.2 Backlight test

In the default factory system, the LCD screen backlight supports 256 levels of PWM adjustment, with the brightness level ranging from 0 to 255. The default setting is 255, which is the brightest state. When set to 0, the backlight is turned off. The higher the value, the brighter the screen.

```
cat /sys/class/backlight/panel-backlight/max_brightness
// Check the maximum brightness level of the backlight
cat /sys/class/backlight/panel-backlight/brightness          // Check the current brightness level
echo 128 > /sys/class/backlight/panel-backlight/brightness
// Modify the current brightness level and observe the change in screen brightness
cat /sys/class/backlight/panel-backlight/brightness // Then check the current brightness level again
```

4.4 Serial Port Test

4.4.1 RS232 Serial Port Test

The ATK-DLMP135 development board is equipped with one RS232 and one RS485 port. Since they share the same USART2 serial port, the jumpers on the development board need to be adjusted to ensure the selection of the RS232 function. Additionally, before the test, a USB-to-RS232 male serial port cable (from different brands are acceptable) needs to be prepared.

Figure 4.4-1 USB to RS232 Serial Cable (Male Connector)

Connect the USB to RS232 serial cable to the RS232 interface, then connect U2_TX and COM_RX, U2_RX and COM_TX of the J2 pin group together through jumper caps, thereby connecting the USART2 serial port to the RS232 function. As shown in Figure 4.4.1.2.

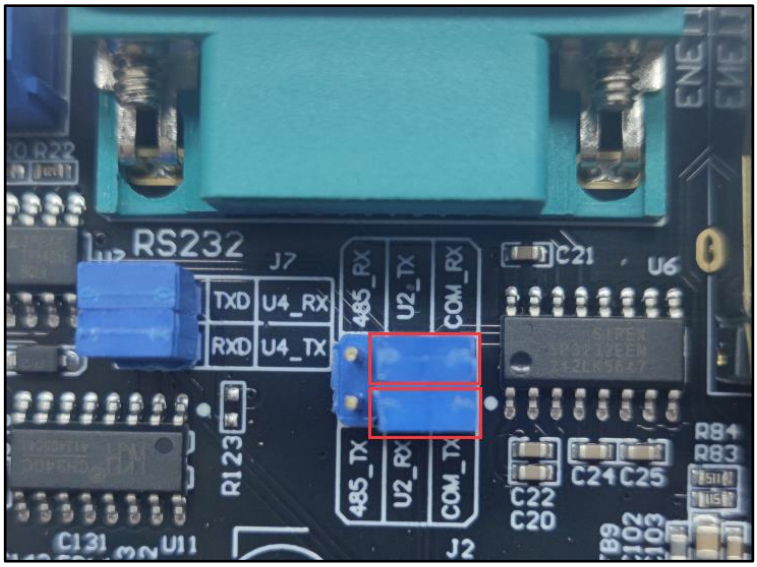


Figure 4.4-2 RS232 jumper cap connection diagram

Check the port numbers in the device manager of the computer. The editor has two port numbers. One is the USB debugging serial port of the development board, and the other is the RS232 port number.

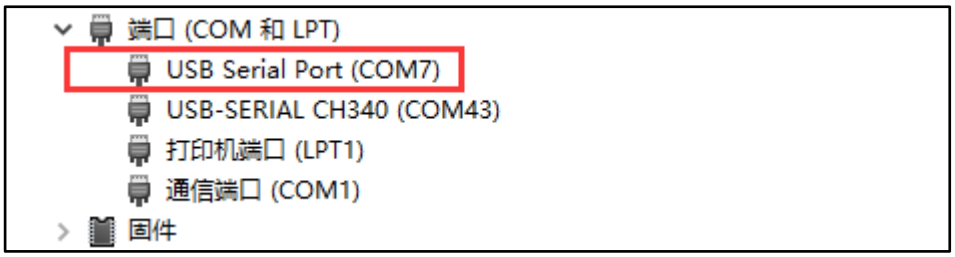


Figure 4.4-3 Viewing port numbers in the device manager

You can see that COM43 is the debugging serial port (UART4) for the author, and COM7 is the RS232 (USART2) serial port on the bottom board of the development board. Next, we will use the minicom serial port debugging tool to conduct data transmission and reception tests.

Execute the following command to enter the minicom configuration interface (the usage method of minicom is not detailed here)

```
minicom -s
```

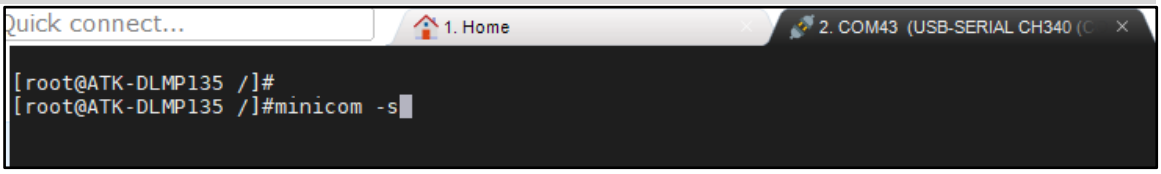


Figure 4.4-4 Enter the minicom configuration interface

Enter the minicom configuration interface. Use the " ↓ " key on the keyboard to select the third item "Serial port setup" to configure the RS232 serial port.

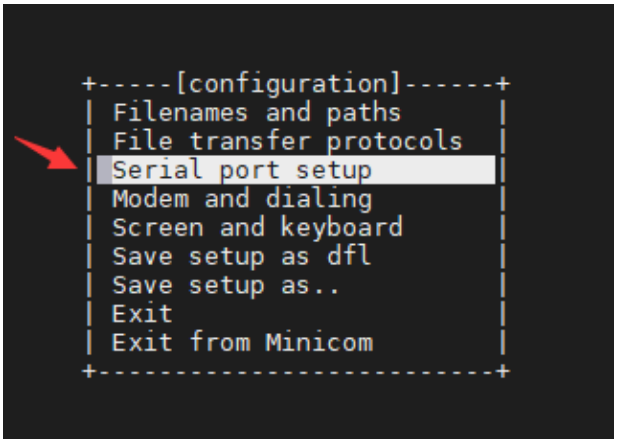


Figure 4.4-5 Minicom configuration interface

In the factory system, the device corresponding to USART2 serial port is /dev/ttySTM1. Therefore, we will configure it as /dev/ttySTM1, 115200, 8N1, without hardware flow control.

The operation method is as follows: On the interface below, press the "A" key to modify the "Serial Device" option content. We change it to "/dev/ttySTM1", and press Enter to confirm; then press the "F" key to turn off the hardware flow control function, that is, change the default "Yes" to "No". Press the "E" key to adjust the baud rate, but it is already set as "115200 8N1", so we don't need to adjust it.

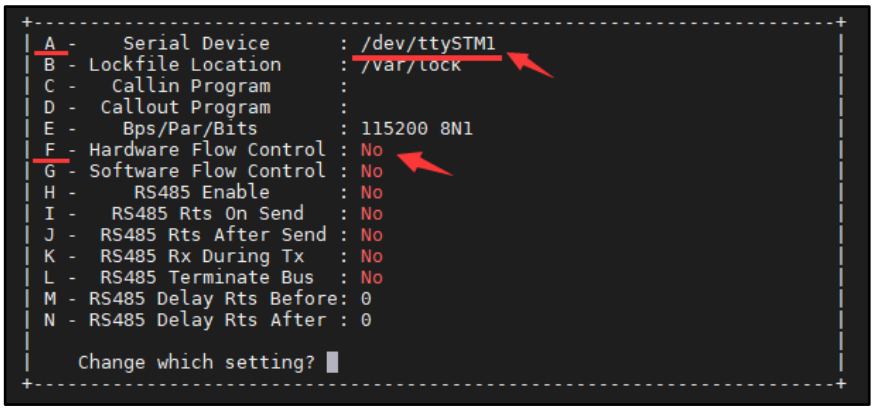


Figure 4.4-6 Serial port configuration interface

Finally, press Enter to confirm the final configuration, which will exit the current configuration interface. In the following interface, select "Exit" to exit the configuration interface and then you can enter the data transmission/reception window.

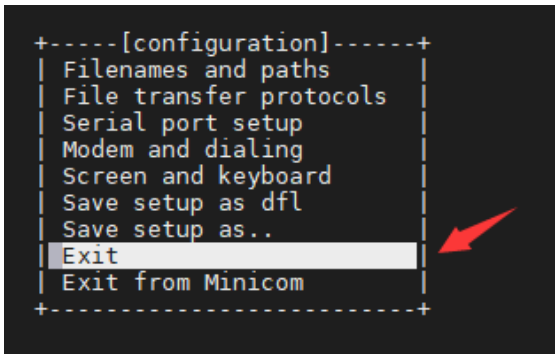


Figure 4.4-7 Exit the configuration interface

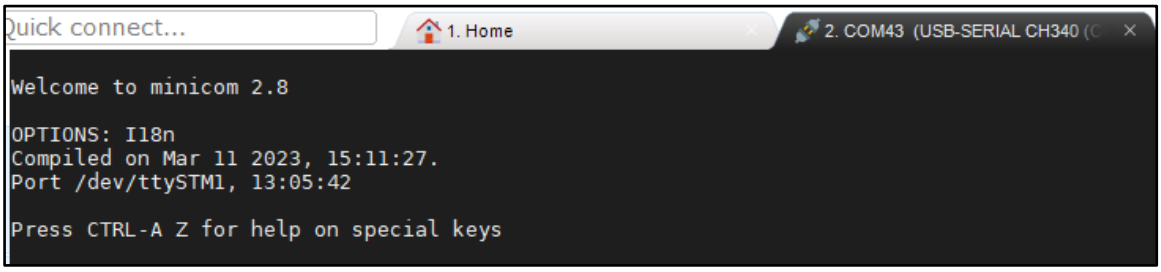


Figure 4.4-8 Enter the data transmission/reception window

Before conducting data transmission and reception, first turn on the minicom data echo function to display the data to be sent and the received data. Press Ctrl+A, then press "Z" to open the minicom command interface. Press the "E" key to turn on the echo function, and then return to the data transmission/reception window. You can now conduct data transmission and reception through the RS232 serial port and the PC port.

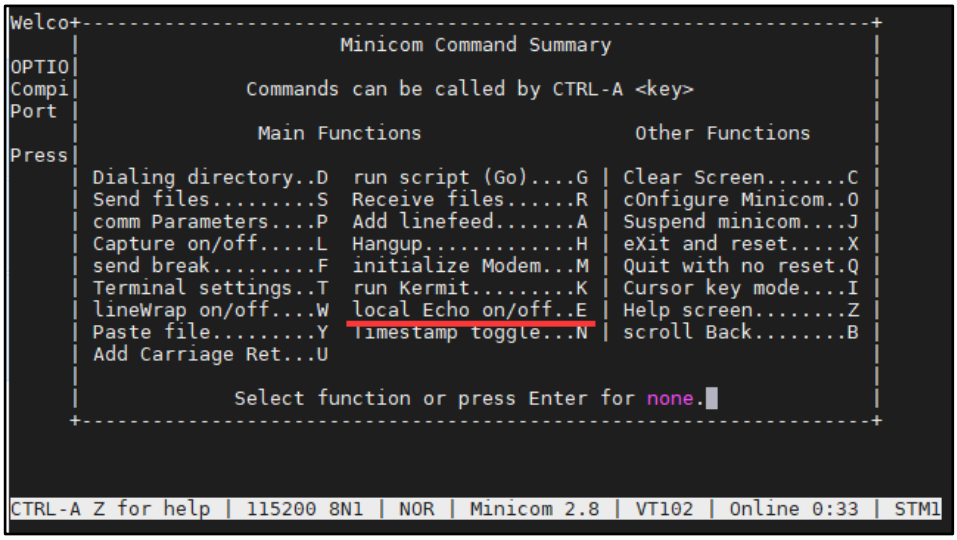


Figure 4.4-9 Press "E" to activate the echo function.

The computer will open the COM7 serial port with a baud rate of 115200, 8 data bits, 1 stop bit, and 1 parity bit. Thus, data can be transmitted and received through the RS232 serial port with the development board.

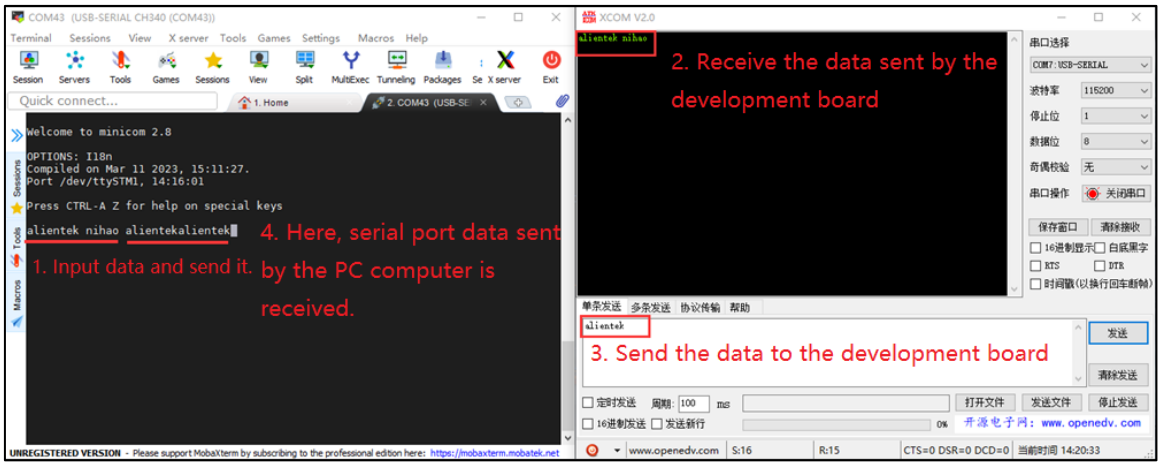


Figure 4.4-10 Serial port data transmission and reception

After the test is completed, press Ctrl+A, then press "Z" to open the minicom command interface. Press the "X" key to exit the minicom tool, and press Enter to confirm the exit. The test process is completed.

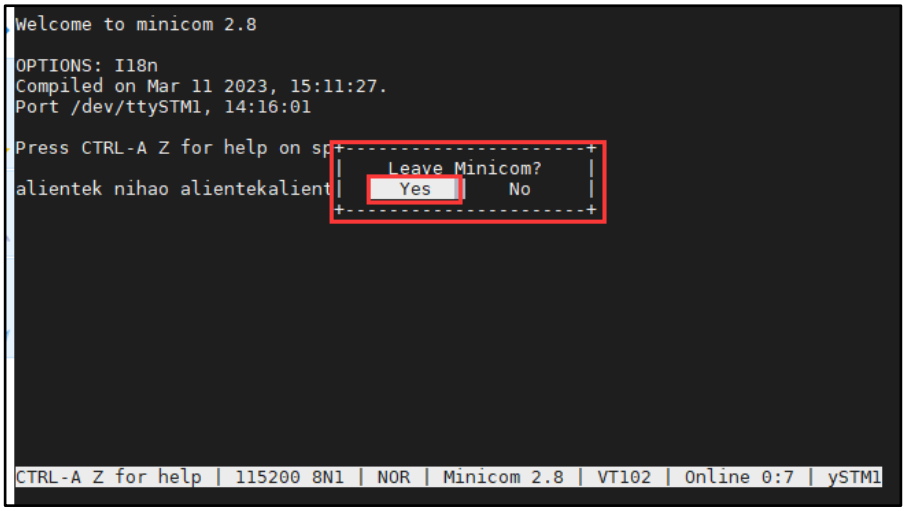


Figure 4.4-11 Exit the minicom tool

4.4.2 RS485 Serial Port Test

As we learned in Section 4.4.1, RS232 and RS485 can be switched using jumper caps. Before testing RS485, we first switch the jumper cap to connect RS485 to USART2. Additionally, we need to use the ALIENTEK USB Converter Module to complete the test. If the user has other RS485-related testing tools, they can test it themselves instead of using this module. As shown in the following figure.

Figure 4.4-12 ALIENTEK all-in-one USB converter module

The U2_TX and 485_RX of the development board J2 pin group, as well as U2_RX and 485_TX, will be connected together through jumper caps, thereby connecting USART2 to the RS485 function. As shown in Figure 4.4.2.2.

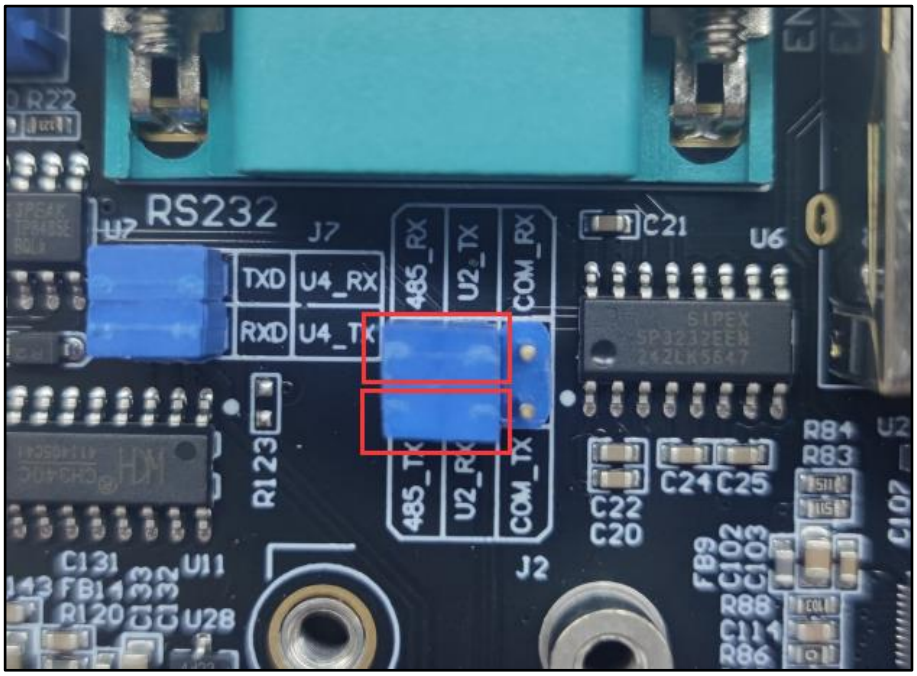


Figure 4.4-13 RS485 jumper cap connection diagram

At the same time, connect the RS485 A terminal of the development board to the RS485 A terminal of the ALIENTEK USB converter module via a DuPont cable, and connect the B terminal to the B terminal. Using the USB converter module from ALIENTEK, a serial port will be visible on the PC (computer end). The testing method can be tested according to the steps in Section 4.4.1.

4.5 USB Test

4.5.1 USB HOST Test

Insert a FAT32 formatted USB drive or a card reader with a TF card directly into the USB_HOST1 or USB_HOST2, USB_HOST3 USB interface of the ATK-DLMP135 development board. The factory system will display the following identification information.

```
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#[ 1659.042275] usb 2-1.2: new high-speed USB device number 6 using ehci-platform
[ 1659.299894] usb-storage 2-1.2:1.0: USB Mass Storage device detected
[ 1659.312879] scsi host0: usb-storage 2-1.2:1.0
[ 1660.323763] scsi 0:0:0:0: Direct-Access     Generic  STORAGE DEVICE   1532 PQ: 0 ANSI: 6
[ 1660.340733] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 1660.559215] sd 0:0:0:0: [sda] 30277632 512-byte logical blocks: (15.5 GB/14.4 GiB)  Capacity
[ 1660.566701] sd 0:0:0:0: [sda] Write Protect is off
[ 1660.571314] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 1660.589730]  sda: sda1   Node name
[ 1660.602602] sd 0:0:0:0: [sda] Attached SCSI removable disk
[ 1660.908233] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

[root@ATK-DLMP135 ~]#
```

Figure 4.5-1 USB drive identification information

Enter the command "df -h" to check the current mount path of the USB drive.

```
df -h
```

```
[root@ATK-DLMP135 ~]#df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        152M     0  152M   0% /dev
/dev/mmcblk1p5  7.2G  435M  6.4G   7% /
tmpfs           219M     0  219M   0% /dev/shm
tmpfs           219M   64K  219M   1% /tmp
tmpfs           219M  172K  218M   1% /run
/dev/mmcblk1p4   45M   39M  2.1M  95% /lib/modules
/dev/sda1        15G   96K   15G   1% /media/sda1
[root@ATK-DLMP135 ~]#
```

USB flash drive usage information and mounting path

Figure 4.5-2 Check the mount path of the USB drive

As shown in the above figure, the current mount path of the USB drive is /media/sda1. Now, let's conduct a test for the read and write speeds of the USB drive.

Note: The interface read and write speed of the card reader itself, the type of TF card, and the type of USB drive also affect the speed value. The user's test value may be different from the author's test value. The speed value is for reference only.

Write speed test:

Enter the following command to write a test file to the USB drive:

```
time dd if=/dev/zero of=/media/sda1/test bs=1024k count=100 conv=fdatasync
```

```
[root@ATK-DLMP135 ~]#time dd if=/dev/zero of=/media/sda1/test bs=1024k count=100
 conv=fdatasync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 8.76041 s, 12.0 MB/s
real    0m 8.78s
user    0m 0.00s
sys     0m 1.28s
[root@ATK-DLMP135 ~]#
```

Figure 4.5-3 Write speed test

It indicates that a 100MiB "test" file is written, and the write speed is **12MB/s**. In fact, the larger the file written, the closer the average write speed is to the actual value. Execute the "**sync**" command once before removing the USB drive to synchronize it with the storage disk.

Read speed test:

Enter the following command to clear the running cache.

```
echo 3 > /proc/sys/vm/drop_caches
```

Tips: Due to the kernel mechanism of LINUX, in general, there is no need to deliberately release the used cache. These cache contents can increase the reading and writing speed of files. Here, in order to test the reading and writing speed, the release operation is required to be performed.

Enter the following command to read the test file that was previously written using the dd command.

```
time dd if=/media/sda1/test of=/dev/null bs=1024k
```

```
[root@ATK-DLMP135 ~]#echo 3 > /proc/sys/vm/drop_caches
[ 3242.445845] sh (571): drop_caches: 3
[root@ATK-DLMP135 ~]#time dd if=/media/sda1/test of=/dev/null bs=1024k
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 6.20074 s, 16.9 MB/s
real    0m 6.20s
user    0m 0.00s
sys     0m 0.63s
[root@ATK-DLMP135 ~]#
```

Figure 4.5-4 Read speed test

This indicates a read operation of 100 MiB data, with a read speed of 16.9 MB/s. The read and write speeds of the card reader also affect this speed value, for reference only.

4.5.2 USB mouse test

After the factory system boots up, plug in the USB mouse, then connect the LCD screen or HDMI to the display. The mouse pointer will be displayed on the screen, and you can operate the Qt interface using the mouse.

4.6 Network testing

The ATK-DLMP135 development board is equipped with 2 gigabit Ethernet interfaces, which can adapt to 10/100/1000M rates.

First, insert the network cable into any of the network ports to ensure that the network can access the internet. The factory system of the development board will automatically obtain an IP address based on the local network environment (the factory system supports DHCP). Enter the ifconfig command to view the obtained network IP.

Here is an example of connecting to the network port ENET1, which corresponds to the network node eth0. For the network port ENET2, which corresponds to the network node eth1, the testing method is the same.

```
ifconfig
```

```
[root@ATK-DLMP135 ~]#ifconfig
eth0      Link encap:Ethernet  HWaddr 88:A7:0E:29:AF:4E
          inet addr:192.168.6.33  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::d2ea:2e6f:2377:3905/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:54 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5360 (5.2 KiB)  TX bytes:3272 (3.1 KiB)
          Interrupt:46 Base address:0xc000
```

Figure 4.6-1 eth0 obtains the external network IP address

Check if the network can access the internet. If there is a data response, it indicates that the network function is normal. Press Ctrl+c to terminate the command.

```
ping www.baidu.com
```

```
[root@ATK-DLMP135 ~]#ping www.baidu.com
PING www.baidu.com (220.181.38.149): 56 data bytes
64 bytes from 220.181.38.149: seq=0 ttl=52 time=35.035 ms
64 bytes from 220.181.38.149: seq=1 ttl=52 time=34.743 ms
64 bytes from 220.181.38.149: seq=2 ttl=52 time=34.782 ms
64 bytes from 220.181.38.149: seq=3 ttl=52 time=35.061 ms
64 bytes from 220.181.38.149: seq=4 ttl=52 time=34.823 ms
64 bytes from 220.181.38.149: seq=5 ttl=52 time=34.859 ms
64 bytes from 220.181.38.149: seq=6 ttl=52 time=35.096 ms
64 bytes from 220.181.38.149: seq=7 ttl=52 time=35.179 ms
^C
--- www.baidu.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 34.743/34.947/35.179 ms
[root@ATK-DLMP135 ~]#
```

Figure 4.6-2 ping Baidu

If the local network is not an internet environment (local area network), then test the network communication by configuring a static IP. For example, configure the network static IP for the network interface ENET1 on the development board: ifconfig eth0 192.168.1.100. Configure the static IP for the PC (computer end): 192.168.1.200. Conduct a local network ping communication test between the two. Note that the PC (computer end) needs to disable the firewall.

Test whether it is a gigabit network. Pay attention to using a gigabit network cable, a gigabit router or a gigabit switch. The network card of the PC (computer end) must be a gigabit network card; otherwise, the speed measured or identified may be 100 Mbps.

As shown in the figure below, when the system identifies as 1Gbps/Full, it indicates that it is a gigabit network.

```
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#[ 2645.606380] stm32-dwmac 5800a000.eth1 eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 2645.613277] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

[root@ATK-DLMP135 ~]#
```

Figure 4.6-3 Identified as a gigabit network

The connection speed of the gigabit network can be tested using the iperf3 command. Before the test, please ensure that the network of the development board can ping through to Ubuntu normally.

Execute the following command in the Ubuntu system environment to set Ubuntu as the server;

```
iperf3 -s
```

On the command line of the ATK-DLMP135 development board, by using the iperf3 command, simply input the server IP.

Note that the server IP should be set according to the actual IP of the user's Ubuntu system, not a uniform IP.

```
iperf3 -c 192.168.1.10 -i 10
```

Explanation:

(1) -c: Client.

(2) 192.168.1.10: The IP address of the user ubuntu. This is only an example.

(3) -i: Trigger period.

The measured Bitrate is approximately 800 Mbits/sec, which is the speed of a gigabit network (please do not run other applications before the test to avoid affecting the test speed).

4.7 CAN Test

The ATK-DLMP135 development board is equipped with 2 CAN interfaces (both supporting CAN FD). Before testing CAN, you need to prepare the testing instruments for CAN, such as Zhou Liguang's CAN analyzer, Chuangxin Technology's CAN analyzer, and Guangcheng Technology's CAN analyzer, etc. Otherwise, you can only use the CAN interfaces of two different development boards or other CAN devices for testing. Regarding the use of CAN instruments and CAN upper computers, please refer to the user manuals of each manufacturer's products.

The main characteristics of the CAN used by the development board are as follows:

- Supports CAN FD, which can be regarded as an upgraded version of the CAN protocol. Only the protocol has been upgraded, while the physical layer remains unchanged.
- Main differences between CAN and CAN FD: transmission rate is different, data length is different, frame format is different, and ID length is different.
- The maximum data bit rate is 5 Mbps.

Before testing, connect the CAN analyzer or the CAN testing device to the CAN interface of the ATK-DLMP135 development board. Connect the CANH port to the CANH of the CAN analyzer, and the CANL port to the CANL of the CAN analyzer.

Here is an example of using the FDCAN1 interface to test the CAN function. The same applies to FDCAN2. The CAN interface of the development board is shown in the following figure.

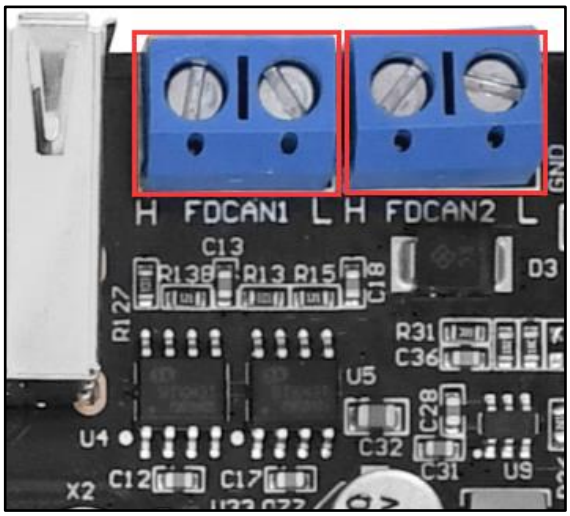


Figure 4.7-1 CAN interface position

4.7.1 CAN test

The minimum data bit rate of CAN is 10 kBit/s, and the maximum data bit rate is 1000 kBit/s.

Before the test, please note that the following example instructions are based on the FDCAN1 interface test, and the corresponding system device name is can0. If the user needs to test the FDCAN2 interface, then the corresponding system device name is can1, and simply modify the instructions.

Example for FDCAN1 interface, set the data bit rate of can0 to 500 kBit/s.

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

Use the "cansend" command to send data.

```
cansend can0 123#01.02.03.04.05.06.07.08
```

Explanation:

(1) can0: can0 represents the FDCAN1 interface device, and can1 represents the FDCAN2 interface device.

(2) 123: Frame ID

(3) 01.02.03.04.05.06.07.08: Frame data

Receive data using the candump command. Terminate the command by pressing Ctrl+c.

```
candump -ta can0
```

Explanation:

(1) -ta: t stands for printing time, and a stands for enabling ASCII output.

Using the Crecent Technology CAN analyzer, the CAN rate was set to 500,000. The test results are shown in the following figure.

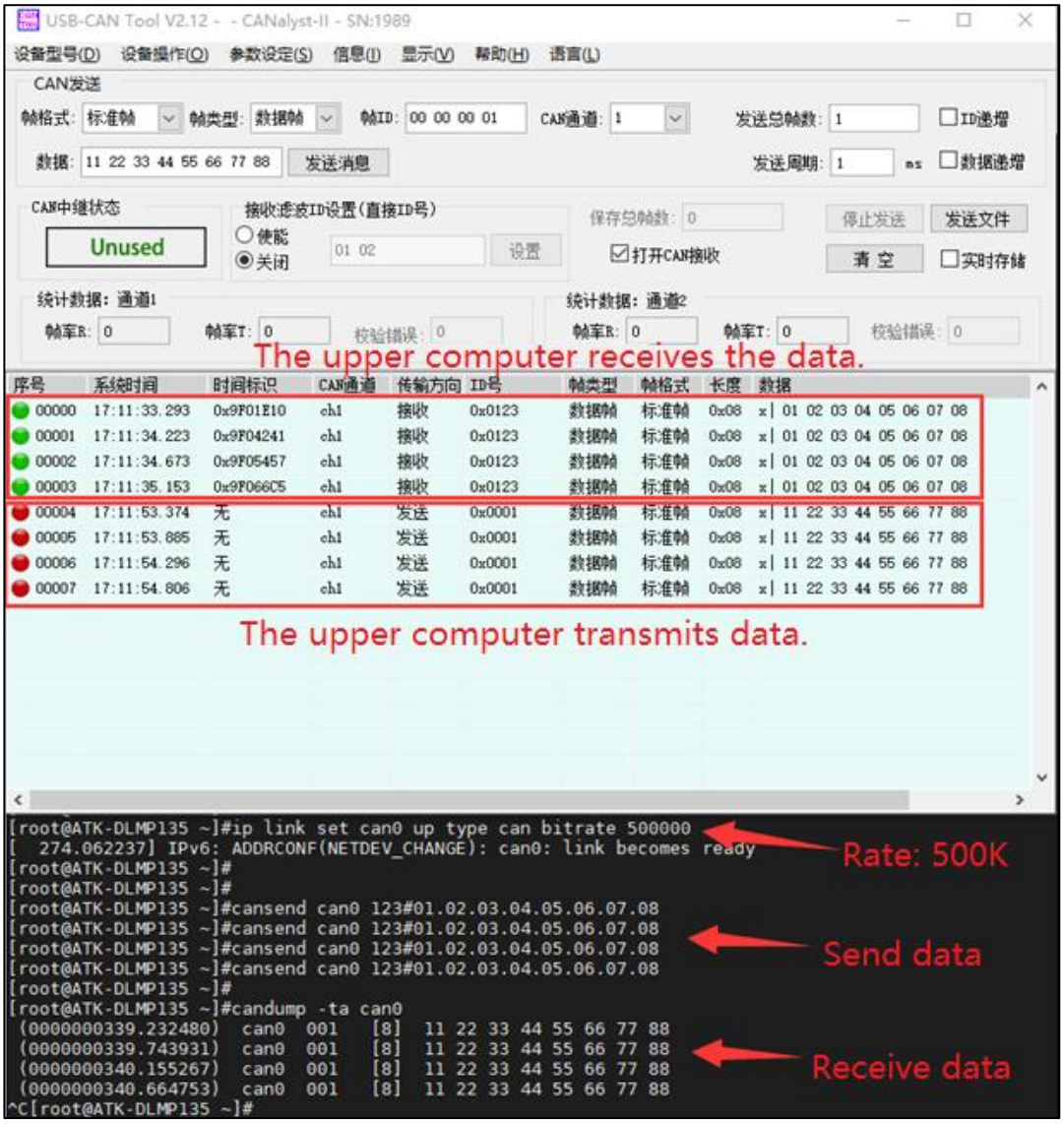


Figure 4.7-2 Perform CAN receive and transmit communication using the host computer and the development board.

4.7.2 CAN FD Test

The following uses the CAN test equipment from Guangcheng Technology (which supports CAN FD) to conduct CAN FD tests. If the CAN test equipment does not support the CAN FD function, it cannot be used for testing CAN FD.

The maximum CAN FD baud rate of the development board is 1000 kBit/s, and the data baud rate is 5000 kBit/s. The available rates are as follows. If a lower baud rate is needed, please use CAN instead of CAN FD.

Before the test, please remind that the following example instructions are based on the FDCAN1 interface test, and the corresponding system device name is can0. If the user needs to test the FDCAN2 interface, the corresponding system device name is can1. Just modify the instructions accordingly.

When setting different rates, it is necessary to first turn off the CAN device, then set the CAN baud rate. If the setting is unsuccessful, please restart the board or the upper computer software.

ifconfig can0 down

Here is an example of configuring the rate for CAN FD. You only need to select one of the following instructions to configure it.

ip link set can0 up type can bitrate 1000000 dbitrate 5000000 fd on

ip link set can0 up type can bitrate 200000 dbitrate 1000000 fd on

ip link set can0 up type can bitrate 100000 dbitrate 500000 fd on

Here, the baud rate is set to 1000 kBit/s and the data baud rate is set to 5000 kBit/s for the test.

ip link set can0 up type can bitrate 1000000 dbitrate 5000000 fd on

```
[root@ATK-DLMP135 ~]#ip link set can0 up type can bitrate 1000000 dbitrate 50000
00 fd on
[  361.206484] m_can_platform 4400f000.can can0: bitrate error 4.0%
[  361.214129] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
[root@ATK-DLMP135 ~]#
```

Figure 4.7-3 Set the CAN FD rate for the development board

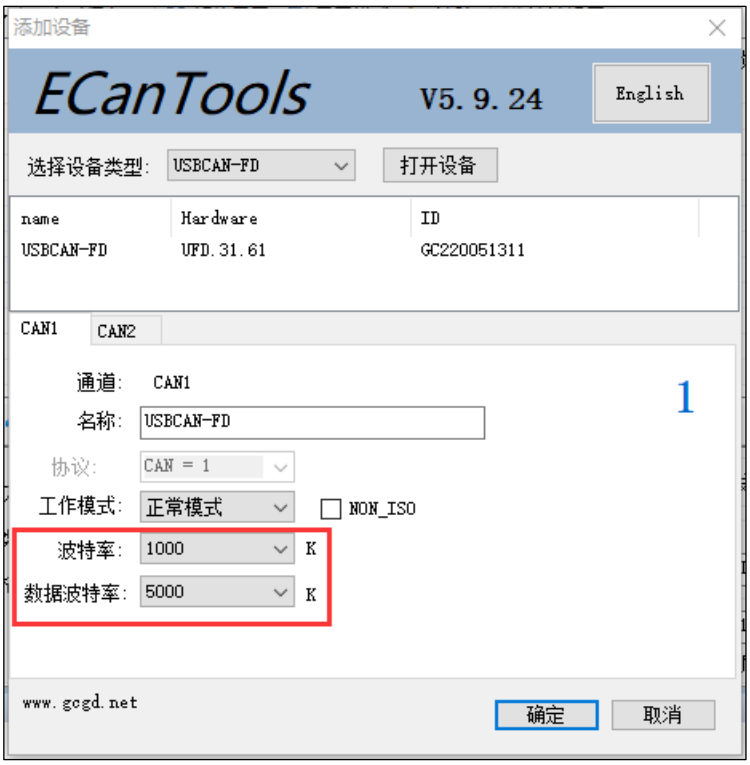


Figure 4.7-4 Configuration of the host computer for Guangcheng Technology's CAN FD testing instrument

To send data using the cansend command, the following is a single instruction. Please copy it line by line!

cansend can0 123##300.11.22.33.44.55.66.77.88.99.aa.bb.cc.dd.ee.ff.00.11.22.33.44.55.66.77.88.99.aa.bb.cc.dd.ee.ff.00.11.22.33.44.55.66.77.88.99.aa.bb.cc.dd.ee.ff.00.11.22.33.44.55.66.77.88.99.aa.bb.cc.dd.ee.ff

Explanation:

(1) can0: can0 represents the FDCAN1 interface device, and can1 represents the FDCAN2 interface device.

(2) 123: Frame ID

(3) 3: Flag (flags)

(4) 00.11.22…: Frame data

Receive data using the candump command. Terminate the command by pressing Ctrl+c.

```
candump -ta can0
```

Explanation:

(1) -ta: t represents the printing time, a represents the activation of ASCII output.

The test results of the development board's CAN FD are shown in the following figure.

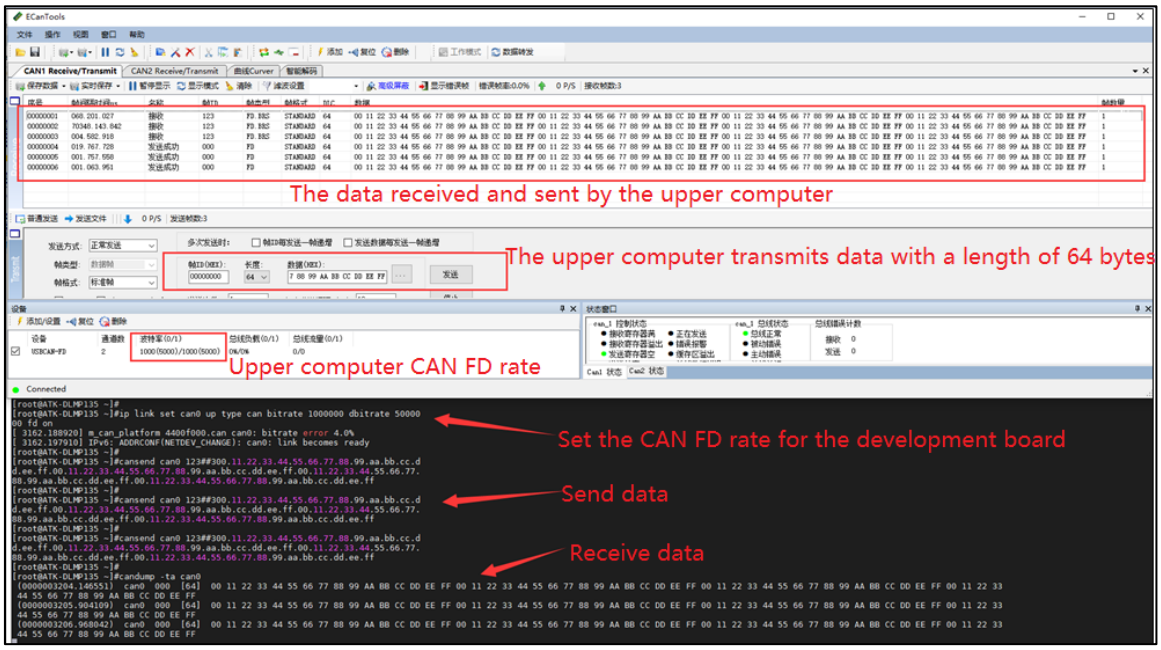


Figure 4.7-5 Perform CAN FD transmission and reception communication using the host computer and the development board.

4.8 RTC Clock Test

The ATK-DLMP135 development board is equipped with an RTC clock chip PCF8563, which is an external RTC clock.

Check if the development board base board has installed an RTC button battery. You can also use a multimeter to check if the RTC battery is powered. Prevent the system from losing time due to a dead RTC battery. Note: The battery is a consumable item. If it is dead, please replace the button battery.

The Linux system has two clocks: one is system time (software clock), and the other is hardware clock (hardware clock). The date and hwclock commands can be used to view and set the system time and hardware time respectively. The system clock disappears when powered off, while the RTC clock will operate for a long time with a battery. The system clock will synchronize with the RTC clock during system restart.

To view the system clock, use the command date.

```
date
```

Set the current system clock.

```
date -s "2023-1-1 10:00:00"
```

Check the system clock.

```
date
```

Use hwclock to write to the hardware clock.

```
hwclock -w
```

Check the hardware clock to see if it is the same as the one set by the command "hwclock -w".

4.9 EEPROM Test

The ATK-DLMP135 development board is equipped with an EEPROM chip, model AT24C64, which can be read and written via the I2C bus and has 8KB of storage space.

The factory system provides a simple read-write test tool named eeprom-rw for testing the AT24C64. Just execute the following instructions.

```
eeprom-rw
```

Return the following information, indicating normal reading and writing.

```
[root@ATK-DLMP135 ~]#eeprom-rw
write data: http://www.openedv.com/forum.php
read data: http://www.openedv.com/forum.php
result: eeprom readwrite is ok!
```

Figure 4.9-1 EEPROM read/write test

4.10 SPI FLASH test

The ATK-DLMP135 development board is equipped with a SPI NOR FLASH chip of the model W25Q128, which performs read/write operations via the SPI bus and has a storage capacity of 16MB.

When the kernel starts, it will automatically load the W25Q128 driver. When the chip is working properly, the kernel will display the following driver loading information:

```
[    3.668124] panel-simple panel-rgb: Specify missing connector_type
[    3.675544] spi-nor spi0.0: w25q128 (16384 Kbytes)
[    3.683233] spi_stm32 4c003000.spi: driver initialized
[    3.696124] st,stm32-i2s 44004000.audio-controller: No cache defaults, reading back from HW
[    3.707635] [drm] Initialized stm 1.0.0 20170330 for 5a001000.display-controller on minor 0
[    3.719490] hub 2-1:1.0: USB hub found
[    3.719735] hub 2-1:1.0: 4 ports detected
```

Figure 4.10-1 SPI FLASH driver loading

As can be seen below, the factory system registers the SPI Flash as a MTD device. Therefore, we can perform file reading and writing operations on it through the MTD subsystem.

```
ls /dev/mtd*
```

```
[root@ATK-DLMP135 ~]#ls /dev/mtd*
/dev/mtd0  /dev/mtd0ro  /dev/mtdblock0
[root@ATK-DLMP135 ~]#
```

Figure 4.10-2 Check the MTD device

The steps for file reading and writing test are as follows:

0. For the first use, you need to format the /dev/mtdblock0 partition. Note that if it is not the first use, you do not need to execute the formatting command; otherwise, the previously written file data will be cleared and needs to be rewritten again.

```
mkfs.vfat /dev/mtdblock0
```

1. Create a custom mount directory and mount the /dev/mtdblock0 partition to this directory in VFAT format.

```
mkdir /root/w25q128
mount -t vfat /dev/mtdblock0 /root/w25q128
```

2. Enter the mounted directory and conduct a file writing test;

```
cd /root/w25q128
```

```
echo "W25Q128 TEST!" > file.txt
sync
```

3. Read the file and confirm whether the writing was successful;

```
cat file.txt
```

4. Finally, exit the mounted directory and unmount the /dev/mtdblock0 partition.

```
cd ../
umount /dev/mtdblock0
```

At this point, the written file data has been stored in the SPI FLASH chip. The file can be read or written by mounting the directory again or restarting the board and then mounting it again.

```
[root@ATK-DLMP135 ~]#mkfs.vfat /dev/mtdblock0
mkfs.fat 4.2 (2021-01-31)
Cannot initialize conversion from codepage 850 to UTF-8: Invalid argument
Cannot initialize conversion from UTF-8 to codepage 850: Invalid argument
Using internal CP850 conversion table
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#mkdir /root/w25q128
[root@ATK-DLMP135 ~]#mount -t vfat /dev/mtdblock0 /root/w25q128
[root@ATK-DLMP135 ~]#cd /root/w25q128
[root@ATK-DLMP135 ~/w25q128]#echo "W25Q128 TEST!" > file.txt
[root@ATK-DLMP135 ~/w25q128]#sync
[root@ATK-DLMP135 ~/w25q128]#cat file.txt
W25Q128 TEST!
[root@ATK-DLMP135 ~/w25q128]#cd ../
[root@ATK-DLMP135 ~]#umount /dev/mtdblock0
[root@ATK-DLMP135 ~]#
```

Figure 4.10-3 SPI FLASH Read-Write File Testing Process

4.11 Audio Testing

The ATK-DLMP135 development board is equipped with a high-performance audio codec chip CS42L51. The onboard microphone can record audio, and a small-power speaker is connected on the back of the baseboard, facilitating users to conduct audio testing.

4.11.1 Playing Audio Testing

Using system instructions aplay or gst-play-1.0, audio files can be played. Users can also copy audio files to the file system and use instructions aplay or gst-play-1.0 for playback. Plug in headphones or listen to whether there is sound from the speaker on the baseboard. Note: When wearing headphones, first place them close to your ear and listen, do not directly put them on to avoid possible excessive volume that may affect the experience.

Set the playback volume.

```
amixer -c STM32MP1DK cset name='PCM Playback Switch' 'on','on'
amixer -c STM32MP1DK cset name='PCM Playback Volume' '85','85'
amixer -c STM32MP1DK cset name='Analog Playback Volume' '170','170'
amixer -c STM32MP1DK cset name='PCM channel mixer' 'L R'
```

Play the audio test files provided by the factory system. You can also replace the audio files for testing. Press Ctrl+C to terminate the instruction.

```
gst-play-1.0 /opt/qtapp/src/audio/0.mp3
```

4.11.2 Recording Test

There are two recording devices on the ATK-DLMP135 baseplate: one is the microphone MIC and the other is the headphone jack PHONE. Note that there is a jumper cap next to the headphones, which is used to switch between the headphone MIC input and the microphone MIC input on the baseplate. When PHONE is connected to the BOARD, it indicates the headphone MIC input; when MIC is connected to the BOARD, it indicates the microphone MIC input on the baseplate.

Set the recording gain.

```
amixer -c STM32MP1DK cset name='PGA-ADC Mux Left' '3'
amixer -c STM32MP1DK cset name='PGA-ADC Mux Right' '3'
amixer -c STM32MP1DK cset name='Mic Boost Volume' '1','1'
amixer -c STM32MP1DK cset name='ADC Boost Volume' '1','1'
```

Record, and carry out the following instructions.

```
arecord -r 44100 -f S16_LE -c 2 -d 10 -D hw:0,1 record.wav
```

Explanation:

(1) -r 44100: Sampling rate 44.1K.

(2) -f S16_LE: Sampling in S16_LE format.

(3) -c 2: 2 channels.

(4) -d 10: Recording duration 10 seconds.

(5) -D hw: 0, 1: Specify the hardware sound card, 0 is the card number, 1 is the device number.

(6) record.wav: The audio file generated by the recording.

When playing the audio file recorded above, the size of the generated audio file is related to the format set by the user and the duration of the recording.

```
aplay record.wav
```

Note: The CS42L51 chip itself only has an output interface for playing through headphones, and does not have a dedicated speaker interface. The speaker on the ATK-DLMP135 development board is directly connected to the right channel of the headphones. Therefore, the headphones and the speaker cannot be automatically switched, and there will be sound simultaneously when playing audio.

4.12 HDMI Test

- Overview of the maximum supported resolution for onboard HDMI

Since the STMP32MP135 CPU does not have an HDMI controller, only RGB to HDMI conversion can be used. The ATK-DLMP135 board has an onboard HDMI Sil902x chip (acting as an HDMI transmitter), and the Sil902x chip supports 1080p output. However, the STM32MP135 CPU can only output to the LTDC signal at WXGA (1366×768) @ 60 fps. Therefore, the onboard HDMI is determined by the CPU's LTDC signal, which means it can only output WXGA (1366×768) @ 60 fps at most.

The default configuration of the HDMI output resolution in the factory system is 1280×720@60 fps. There is no need to modify it to WXGA (1366×768) @ 60 fps because some displays may not support this resolution and may not be able to display it. After testing, the factory system uses a larger resolution of Qt will not be smooth, because the larger Qt interface resolution consumes more performance. While the ALIENTEK RGB LCD screen feels more smooth when sliding the interface because the RGB LCD screen has a smaller resolution. It should be noted that the STM32MP135 is just a single-core A7, and its performance cannot be compared to that of a mobile phone.

Since the HDMI signal is converted from the RGB signal, HDMI and RGB LCD share the same interface. Therefore, theoretically, only one of them can be used.

First, use the HDMI cable (the user needs to have an HDMI cable; if not, the user needs to purchase one themselves) to connect to the HDMI interface on the development board, and connect the other end to the monitor. Power on the monitor. The development board's jumpers are set to boot from eMMC. In the serial terminal, when you see the countdown for U-Boot startup ends, it enters the device tree loading mode. Press the number key "2" to select the device tree using HDMI, then press Enter to load the HDMI device tree for startup.

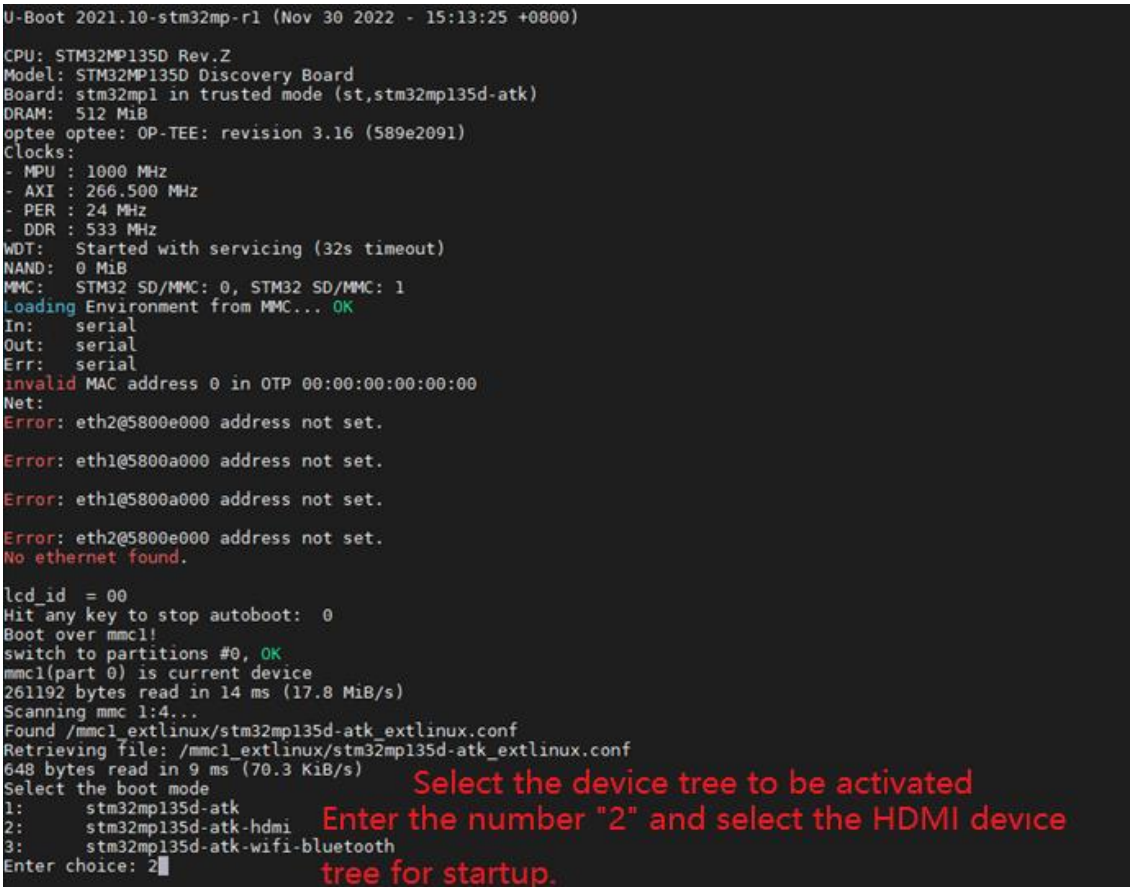


Figure 4.12-1 HDMI device tree selection

After successful startup, the factory system's Qt UI can be displayed on the screen, as shown in the following figure.

Figure 4.12-2 HDMI display shows

➢ Permanently modify the loading of the HDMI device tree for booting

The default booting of the factory system starts with the LCD's stm32mp135d-atk.dtb device tree. After booting from eMMC, you can modify the default boot device tree in the /lib/modules/mmc1_extlinux/stm32mp135d-atk_extlinux.conf file of the factory file system. If the user is booting the system using a TF(SD) system card, then the /lib/modules/mmc0_extlinux/stm32mp135d-atk_extlinux.conf file of the factory file system should be modified.

Enter the command in the serial terminal, edit the boot configuration file, and change stm32mp135d-atk to stm32mp135d-atk-hdmi. Press Esc and then press :wq to save and exit. Thus, the development board will start by default with the HDMI device tree. Users no longer need to select the device tree for booting in U-Boot.

```
vi /lib/modules/mmc1_extlinux/stm32mp135d-atk_extlinux.conf
```

```
# Generic Distro Configuration file generated by OpenEmbedded
menu title Select the boot mode
TIMEOUT 20
DEFAULT stm32mp135d-atk-hdmi
LABEL stm32mp135d-atk
        KERNEL /uImage
        FDTDIR /
        INITRD /uInitrd
        APPEND root=/dev/mmcblk1p5 rootwait rw console=ttySTM0,115200
LABEL stm32mp135d-atk-hdmi
        KERNEL /uImage
        FDT /stm32mp135d-atk-hdmi.dtb
        INITRD /uInitrd
        APPEND root=/dev/mmcblk1p5 rootwait rw console=ttySTM0,115200
LABEL stm32mp135d-atk-wifi-bluetooth
        KERNEL /uImage
        FDT /stm32mp135d-atk-wifi-bluetooth.dtb
        INITRD /uInitrd
        APPEND root=/dev/mmcblk1p5 rootwait rw console=ttySTM0,115200
~
```

Change the default content to "stm32mp135d-atk-hdmi"

Figure 4.12-3 Modify the default loading of HDMI device tree

4.13 Board-based SDIO WIFI test

The ATK-DLMP135 development board uses a WIFI&Bluetooth dual-module, with the chip being RTL8723DS.

When starting the experiment, please check the antenna on the board, that is, the antenna next to the USB interface, whether it is tightened. A 12V power supply needs to be plugged in.

In addition, since SDIO WIFI and TF card share the same set of SDMMC signal communication, when testing WIFI, the bottom board power supply jumper cap needs to be switched to connect 3V3 and WL to supply power to SDIO WIFI. At this time, the blue power indicator light of WIFI_LED lights up.

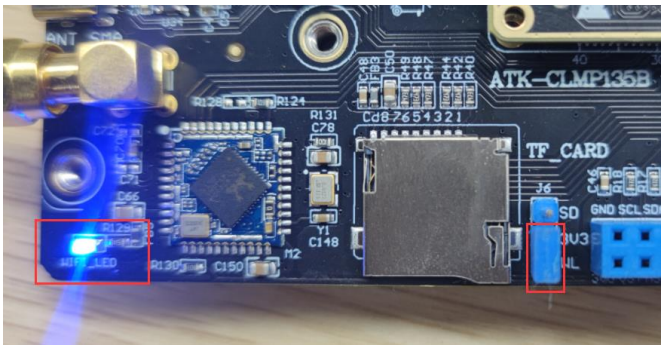


Figure 4.13-1 SDIO WIFI&BT Module Power Supply Connection Diagram

In addition to switching power supply, the corresponding device tree needs to be loaded.

When the development board is powered on and U-Boot is running, in the device tree selection terminal, input the number "3", and the SDIO WIFI&Bluetooth device tree can be loaded.

```
U-Boot 2021.10-stm32mp-r1 (Feb 28 2023 - 15:31:27 +0800)

CPU: STM32MP135D Rev.?
Model: STM32MP135D-ATK Discovery Board
Board: stm32mp1 in trusted mode (st,stm32mp135d-atk)
DRAM:  512 MiB
optee optee: OP-TEE: revision 3.16 (72845af2)
Clocks:
- MPU : 1000 MHz
- AXI : 266.500 MHz
- PER : 24 MHz
- DDR : 533 MHz
WDT:   Started with servicing (32s timeout)
NAND:  0 MiB
MMC:   STM32 SD/MMC: 0, STM32 SD/MMC: 1
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
invalid MAC address 0 in OTP 00:00:00:00:00:00
Net:
Error: eth2@5800e000 address not set.

Error: eth1@5800a000 address not set.

Error: eth1@5800a000 address not set.

Error: eth2@5800e000 address not set.
No ethernet found.

lcd_id  = 00
Hit any key to stop autoboot:  0
Boot over mmc1!
switch to partitions #0, OK
mmc1(part 0) is current device
261192 bytes read in 14 ms (17.8 MiB/s)
Scanning mmc 1:4...
Found /mmc1_extlinux/stm32mp135d-atk_extlinux.conf
Retrieving file: /mmc1_extlinux/stm32mp135d-atk_extlinux.conf
648 bytes read in 9 ms (70.3 KiB/s)
Select the boot mode
1:      stm32mp135d-atk
2:      stm32mp135d-atk-hdmi
3:      stm32mp135d-atk-wifi-bluetooth
Enter choice: 3
```

U-Boot operation information

Input 3, select to load the WIFI & BT device tree

Figure 4.13-2 Load the SDIO WIFI & BT module device tree.

After entering the file system, input the following command to conduct WIFI testing.

rfkill unblock all	// Unlock radio frequency
connmanctl	// Enter the WIFI operation terminal
connmanctl> enable wifi	// Enable WIFI
connmanctl> scan wifi	// Start WIFI scanning, can be scanned multiple times
connmanctl> agent on	// Register the agent
connmanctl> services	// List the WIFI list scanned

The command execution result is as follows.

```
[root@ATK-DLMP135 ~]#connmanctl
connmanctl> enable wifi
wifi is already enabled
connmanctl> scan wifi
Scan completed for wifi
connmanctl> agent on
Agent registered
connmanctl> services
    Linux-WIFI6          wifi_94a4082f8e0e_4c696e75782d5749464936_managed_psk
    ALIENTEK-Linux       wifi_94a4082f8e0e_414c49454e54454b2d4c696e7578_managed_psk
                         wifi_94a4082f8e0e_hidden_managed_psk
    GZXH HUAWEI_Wi-Fi5   wifi_94a4082f8e0e_475a5848204855415745495f57692d466935_managed_psk
    RICH901              wifi_94a4082f8e0e_52494348393031_managed_psk
    GZXH HUAWEI          wifi_94a4082f8e0e_475a584820485541574549_managed_psk
    ALIENTEK-WX          wifi_94a4082f8e0e_414c49454e54454b2d5758_managed_psk
    ChinaNet-ymyk        wifi_94a4082f8e0e_4368696e614e65742d796d796b_managed_psk
    ChinaNet-kss2        wifi_94a4082f8e0e_4368696e614e65742d6b737332_managed_psk
    ALIENTEK-SYS         wifi_94a4082f8e0e_414c49454e54454b2d535953_managed_psk
    ATK-HUAWEI           wifi_94a4082f8e0e_41544b2d485541574549_managed_psk
    ALIENTEK-XHYS        wifi_94a4082f8e0e_414c49454e54454b2d58485953_managed_psk
    ALIENTEK-7机房     wifi_94a4082f8e0e_414c49454e54454b2d37e69cbae688bf_managed_psk
    ALIENTEK-HYS         wifi_94a4082f8e0e_414c49454e54454b2d485953_managed_psk
    ChinaNet-egTF        wifi_94a4082f8e0e_4368696e614e65742d65675446_managed_psk
    ALIENTEK-7办公     wifi_94a4082f8e0e_414c49454e54454b2d37e58a9ee585ac_managed_psk
    ALIENTEK-YF-SYS      wifi_94a4082f8e0e_414c49454e54454b2d59462d535953_managed_psk
    C21                  wifi_94a4082f8e0e_433231_managed_psk
connmanctl>
```

Figure 4.13-3 Scanned hotspots

From the list of scanned WIFI hotspots, use the command "connect XXXX" to connect, then enter the password and press Enter to confirm. For example, if the example is to connect to ALIENTEK-Linux, the result will be as follows:

```
connmanctl> services
    Linux-WIFI6          wifi_94a4082f8e0e_4c696e75782d5749464936_managed_psk
    ALIENTEK-Linux       wifi_94a4082f8e0e_414c49454e54454b2d4c696e7578_managed_psk   Examples
                         wifi_94a4082f8e0e_hidden_managed_psk
    GZXH HUAWEI_Wi-Fi5   wifi_94a4082f8e0e_475a5848204855415745495f57692d466935_managed_psk
    RICH901              wifi_94a4082f8e0e_52494348393031_managed_psk
    GZXH HUAWEI          wifi_94a4082f8e0e_475a584820485541574549_managed_psk
    ALIENTEK-WX          wifi_94a4082f8e0e_414c49454e54454b2d5758_managed_psk
    ChinaNet-ymyk        wifi_94a4082f8e0e_4368696e614e65742d796d796b_managed_psk
    ChinaNet-kss2        wifi_94a4082f8e0e_4368696e614e65742d6b737332_managed_psk
    ALIENTEK-SYS         wifi_94a4082f8e0e_414c49454e54454b2d535953_managed_psk
    ATK-HUAWEI           wifi_94a4082f8e0e_41544b2d485541574549_managed_psk
    ALIENTEK-XHYS        wifi_94a4082f8e0e_414c49454e54454b2d58485953_managed_psk
    ALIENTEK-7机房     wifi_94a4082f8e0e_414c49454e54454b2d37e69cbae688bf_managed_psk
    ALIENTEK-HYS         wifi_94a4082f8e0e_414c49454e54454b2d485953_managed_psk
    ChinaNet-egTF        wifi_94a4082f8e0e_4368696e614e65742d65675446_managed_psk
    ALIENTEK-7办公     wifi_94a4082f8e0e_414c49454e54454b2d37e58a9ee585ac_managed_psk
    ALIENTEK-YF-SYS      wifi_94a4082f8e0e_414c49454e54454b2d59462d535953_managed_psk
    C21                  wifi_94a4082f8e0e_433231_managed_psk
connmanctl> connect wifi_94a4082f8e0e_414c49454e54454b2d4c696e7578_managed_psk   Connection
Agent RequestInput wifi_94a4082f8e0e_414c49454e54454b2d4c696e7578_managed_psk   example hotspot
  Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[ WPS ] ]
  WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? 15902[redacted]3   Enter the hotspot password and press Enter to confirm.
connmanctl> [  221.968605] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
Connected wifi_94a4082f8e0e_414c49454e54454b2d4c696e7578_managed_psk   Indicates successful
connmanctl>                                                             connection
connmanctl>
connmanctl> exit   Exit the connmand command terminal
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#
```

Figure 4.13-4 Connect to the hotspot

As per the above result, you have connected to the WIFI hotspot. You can use the "ifconfig" command to view the IP address obtained by our wlan0.

```
[root@ATK-DLMP135 ~]#ifconfig
eth0      Link encap:Ethernet  HWaddr 88:CF:5A:FD:D4:9F
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:46 Base address:0xc000

eth1      Link encap:Ethernet  HWaddr 88:FB:B5:79:79:4E
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:55 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:220 errors:0 dropped:0 overruns:0 frame:0
          TX packets:220 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12437 (12.1 KiB)  TX bytes:12437 (12.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 94:A4:08:2F:8E:0E
          inet addr:192.168.6.247  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::e57e:e22:1a53:ae46/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1497 errors:0 dropped:126 overruns:0 frame:0
          TX packets:123 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:155027 (151.3 KiB)  TX bytes:14758 (14.4 KiB)

[root@ATK-DLMP135 ~]#
```

Figure 4.13-5 Check the obtained IP address

Ping Baidu to test connectivity. You can also ping the gateway to test the connectivity of the WIFI. Note: If you cannot ping Baidu, please restart the development board. Do not plug in the network cable and reconnect the WIFI again! Because the system will default to allowing only one network device to connect to the external network.

```
ping www.baidu.com -I wlan0
```

```
[root@ATK-DLMP135 ~]#ping www.baidu.com -I wlan0
PING www.baidu.com (14.215.177.38): 56 data bytes
64 bytes from 14.215.177.38: seq=0 ttl=55 time=49.107 ms
64 bytes from 14.215.177.38: seq=1 ttl=55 time=28.592 ms
64 bytes from 14.215.177.38: seq=2 ttl=55 time=19.973 ms
64 bytes from 14.215.177.38: seq=3 ttl=55 time=37.498 ms
64 bytes from 14.215.177.38: seq=4 ttl=55 time=67.250 ms
64 bytes from 14.215.177.38: seq=5 ttl=55 time=37.036 ms
64 bytes from 14.215.177.38: seq=6 ttl=55 time=66.721 ms
64 bytes from 14.215.177.38: seq=7 ttl=55 time=76.500 ms
^C
--- www.baidu.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 19.973/47.834/76.500 ms
[root@ATK-DLMP135 ~]#
```

Figure 4.13-6 Ping Baidu test

4.14 Board Bluetooth test

The ATK-DLMP135 development board uses a WIFI&Bluetooth dual-module, with the chip being RTL8723DS.

Just like the SDIO WIFI test in the previous chapter, first ensure the power supply and device tree loading switch are completed.

To prevent the Bluetooth from not fully resetting, we request a gpio from the kernel (this gpio is hardware-connected to the power-on enable pin of the Bluetooth). Set it to output mode, give it a low level first, then a high level, so that the Bluetooth can be reset.

Editor Bluetooth initialization script atk-bluetooth-init.sh, placed in /root/shell/bluetooth. We can enter the /root/shell/bluetooth directory and run the initialization script.

```
cd /root/shell/bluetooth/
./atk-bluetooth-init.sh
```

```
[root@ATK-DLMP135 ~]#cd /root/shell/bluetooth/
[root@ATK-DLMP135 ~/shell/bluetooth]#
[root@ATK-DLMP135 ~/shell/bluetooth]#./atk-bluetooth-init.sh
[   32.972947] Bluetooth: HCI UART driver ver 2.3
[   32.976023] Bluetooth: HCI UART protocol H4 registered
[   32.983579] Bluetooth: HCI UART protocol Three-wire (H5) registered
[   33.323297] Bluetooth: : Non-link packet received in non-active state
[   33.519908] cryptd: max_cpu_qlen set to 1000
bluetooth init finished!
[root@ATK-DLMP135 ~/shell/bluetooth]#
```

Figure 4.14-1 Run the initialization script

Use the following command to turn on Bluetooth.

```
hciconfig hci0 up
hciconfig hci0 piscan // Bluetooth is turned on and being scanned.
```

```
[root@ATK-DLMP135 ~/shell/bluetooth]#hciconfig hci0 up
[root@ATK-DLMP135 ~/shell/bluetooth]#hciconfig hci0 piscan
```

Figure 4.14-2 Turn on Bluetooth

Explanation:

(1) hciconfig: Bluetooth utility, used for configuring Bluetooth

(2) hci0: Name of the Bluetooth network node

(3) up: Turn on Bluetooth. If set to down, it indicates that Bluetooth is turned off.

Use the following command to scan for Bluetooth devices. The user should first turn on the Bluetooth on the mobile phone, and then use the following command for scanning. You can try scanning several times.

```
hcitool scan
```

```
[root@ATK-DLMP135 ~/shell/bluetooth]#hcitool scan
Scanning ...
        48:95:07:54:ED:D0       n/a
        00:13:EF:A0:04:F1       n/a
        6C:F7:84:02:D3:F0       flyzeng-bt-atk
        D4:1A:3F:6F:97:CC       OPPO R15
        7C:2A:DB:03:DF:77       bluetooth
        80:CF:A2:74:0E:72       n/a
        AC:D6:18:5C:C2:28       SHELLY.
        68:B9:D3:AE:F7:31       MP135
[root@ATK-DLMP135 ~/shell/bluetooth]#
```

Figure 4.14-3 Scan to the mobile phone Bluetooth device

The mobile phone needs to connect to the development board's Bluetooth. The system must have Bluetooth services enabled, but the default file system of the development board does not have the corresponding services enabled. There are many ways to connect the mobile phone to Bluetooth, such as Bluetooth audio, Bluetooth file transfer OBEX, etc. The Buildroot file system does not have corresponding service configurations by default. The Bluetooth SPP protocol can be used for serial data transmission between the mobile phone Bluetooth and the development board.

4.15 Testing of 4G module ME3630-W

The ATK-DLMP135 development board is equipped with a 4G module interface, which supports the installation of ME3630-W, EC20, etc. 4G modules. The ME3630-W 4G module is ready and is sold by the ALIENTEK store. Insert the ME3630-W 4G module into the 4G module interface and tighten the screw. Ensure that the 4G module is properly connected and aligned with the interface of the base. Please use the original antenna and connect it to the MAIN interface of the 4G module. (Warm reminder: After opening the 4G module, the gold fingers are prone to oxidation. Over time, there will be an oxide film on them. If the experiments are not successful after multiple attempts, please check if the 4G module is properly inserted or if there is an oxide film. Please save the 4G module after use!)

Insert the 4G card (supporting mobile, unicom, and telecom 4G cards) and the module properly, and the WWAN LED on the base board will light up green after the development board starts. If the WWAN LED does not light up, please check if the module is properly connected and inserted, if the 4G card is inserted, if the antenna is connected, and if the 12V power supply provided with it is plugged in.

Figure 4.15-1 4G Module Antenna SIM Card Connection Diagram

After the 4G module is properly loaded, three ttyUSB* files can be seen under /dev. The 4G module communicates via AT commands to access /dev/ttyUSB2.

```
ls /dev/ttyUSB*
```

```
[root@ATK-DLMP135 ~]#ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2
[root@ATK-DLMP135 ~]#
```

Figure 4.15-2 Check the generated /dev/ttyUSB* nodes

4.15.1 PPP dial-up Internet access

The relevant scripts for testing the 4G module have already been stored in the /root/shell/4G directory of the factory file system.

Enter the /root/shell/4G/ directory.

```
cd /root/shell/4G/
```

By using the "ls" command, you can see that there are multiple script files in this directory. If there are no such files, please update the system firmware promptly.

```
[root@ATK-DLMP135 ~/shell/4G]#ls
ECM_DEMO        gosuncn_options            gosuncn_ppp_dialer_10086  ppp-on-10086
ECM_DEMO_AUTO   gosuncn_ppp_dialer_10000   ppp-on-10000
disconnect      gosuncn_ppp_dialer_10010   ppp-on-10010
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-3 View the script

The suffix 10000 indicates that it is applicable to the telecom card, 10010 indicates that it is applicable to the unicom card, and 10086 indicates that it is applicable to the mobile card.

Here, we take the mobile card as an example. Execute the following ppp-on script to configure the 4G module, and the & symbol indicates that it runs in the background.

```
./ppp-on-10086 &
```

```
[root@ATK-DLMP135 ~/shell/4G]#./ppp-on-10086 &
[root@ATK-DLMP135 ~/shell/4G]#
pppd options in effect:
debug           # (from gosuncn_options)
nodetach                # (from gosuncn_options)
persist         # (from gosuncn_options)
dump            # (from gosuncn_options)
noauth          # (from gosuncn_options)
user Anyname            # (from gosuncn_options)
password ??????         # (from gosuncn_options)
/dev/ttyUSB2            # (from gosuncn_options)
115200          # (from gosuncn_options)
lock            # (from gosuncn_options)
connect chat -v -f /root/shell/4G/gosuncn_ppp_dialer_10086              # (from command line)
crtscts         # (from gosuncn_options)
modem           # (from gosuncn_options)
novj            # (from gosuncn_options)
ipcp-accept-local               # (from gosuncn_options)
ipcp-accept-remote              # (from gosuncn_options)
noipdefault             # (from gosuncn_options)
defaultroute            # (from gosuncn_options)
usepeerdns              # (from gosuncn_options)
noccp           # (from gosuncn_options)
nobsdcomp               # (from gosuncn_options)
ATE
OK
ATH
OK
ATP
OK
AT+CGDCONT=1,"IP","CMNET"
OK
ATD*99#
CONNECT
Script chat -v -f /root/shell/4G/gosuncn_ppp_dialer_10086 finished (pid 610), status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB2
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xe436172a> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xdfe20c24> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xdfe20c24> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xe436172a> <pcomp> <accomp>]
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPV6CP ConfReq id=0x1 <addr fe80::0905:73cf:2dff:0e59>]
rcvd [IPCP ConfReq id=0x1 <addr 10.159.122.208>]
sent [IPCP ConfAck id=0x1 <addr 10.159.122.208>]
rcvd [IPV6CP ConfReq id=0x1 <addr fe80::79ef:3a18:b73d:e8ef>]
sent [IPV6CP ConfAck id=0x1 <addr fe80::79ef:3a18:b73d:e8ef>]
rcvd [IPCP ConfNak id=0x1 <addr 10.159.122.207> <ms-dns1 221.179.38.7> <ms-dns2 221.179.38.7>]
sent [IPCP ConfReq id=0x2 <addr 10.159.122.207> <ms-dns1 221.179.38.7> <ms-dns2 221.179.38.7>]
rcvd [IPV6CP ConfAck id=0x1 <addr fe80::0905:73cf:2dff:0e59>]
local  LL address fe80::0905:73cf:2dff:0e59
remote LL address fe80::79ef:3a18:b73d:e8ef
rcvd [IPCP ConfAck id=0x2 <addr 10.159.122.207> <ms-dns1 221.179.38.7> <ms-dns2 221.179.38.7>]
local  IP address 10.159.122.207
remote IP address 10.159.122.208
primary   DNS address 221.179.38.7
secondary DNS address 221.179.38.7
```

The obtained IP and DNS

Figure 4.15-4 PPP dial-up access obtained IP and DNS

Press Enter key to return to the command line. Use ifconfig to view the dial-up node name of the 4G module.

```
[root@ATK-DLMP135 ~/shell/4G]#ifconfig
eth0      Link encap:Ethernet  HWaddr 88:02:6B:0B:38:F9
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:46 Base address:0xc000

eth1      Link encap:Ethernet  HWaddr 88:1C:28:9E:68:04
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:55 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5766 (5.6 KiB)  TX bytes:5766 (5.6 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.159.122.207  P-t-P:10.159.122.208  Mask:255.255.255.255
          inet6 addr: fe80::905:73cf:2dff:e59/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:5936 (5.7 KiB)  TX bytes:4201 (4.1 KiB)

usb0      Link encap:Ethernet  HWaddr D2:5C:FC:65:55:3A
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[root@ATK-DLMP135 ~/shell/4G]#
```

"ppp0" is the dialing node name for the 4G network module, and it also obtains the IP address.

Figure 4.15-5 View the name of 4G network nodes

Ping Baidu to test connectivity. -I refers to the specified network card. A data reply indicates connectivity. Press Ctrl+c to terminate the command.

```
ping www.baidu.com -I ppp0
```

```
[root@ATK-DLMP135 ~/shell/4G]#ping www.baidu.com -I ppp0
PING www.baidu.com (183.232.231.172): 56 data bytes
64 bytes from 183.232.231.172: seq=0 ttl=55 time=44.375 ms
64 bytes from 183.232.231.172: seq=1 ttl=55 time=43.727 ms
64 bytes from 183.232.231.172: seq=2 ttl=55 time=43.319 ms
64 bytes from 183.232.231.172: seq=3 ttl=55 time=53.152 ms
64 bytes from 183.232.231.172: seq=4 ttl=55 time=42.757 ms
64 bytes from 183.232.231.172: seq=5 ttl=55 time=42.479 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 42.479/44.968/53.152 ms
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-6 Ping Baidu to test 4G network connection.

If you get the prompt "bad address 'www.baidu.com'", then please check if the DNS server address is set correctly. Open the /etc/resolv.conf file and then add the obtained DNS address "221.179.38.7" to it.

```
echo nameserver 221.179.38.7 >> /etc/resolv.conf
```

```
[root@ATK-DLMP135 ~/shell/4G]#ping www.baidu.com -I ppp0
ping: bad address 'www.baidu.com'
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-7 Lack of DNS

```
[root@ATK-DLMP135 ~/shell/4G]#echo nameserver 221.179.38.7 >> /etc/resolv.conf
[root@ATK-DLMP135 ~/shell/4G]#cat /etc/resolv.conf
# Generated by dhcpcd
# /etc/resolv.conf.head can replace this line
# /etc/resolv.conf.tail can replace this line
nameserver 221.179.38.7
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-8 Add DNS to the /etc/resolv.conf file

4.15.2 ECM Internet Access

From the perspective of the operating system, the CDC ECM device is just a virtual Ethernet card, containing the MAC address and IP address required by a standard network card.

ECM_DEMO_AUTO and ECM_DEMO are the ECM internet access programs of Gosuncn (we don't need to know the source code as it is just a tool), and the instruction summary is as follows:

```
ECM_DEMO -t up              // Enable ecm internet access
ECM_DEMO -t down            // Disable ecm internet access
ECM_DEMO -t up -p /dev/ttyUSB1 -a 3gnet
// Enable it and specify the corresponding APN and dialing port at the same time.
ECM_DEMO -t down -p /dev/ttyUSB1        // Disable ecm internet access
```

Detailed parameters can be viewed by executing "ECM_DEMO -h".

The ECM_DEMO_AUTO and ECM_DEMO parameters are the same. The difference is that ECM_DEMO exits the process once it completes the flow, while ECM_DEMO_AUTO runs continuously and checks every certain period of time whether there is a network disconnection. If there is a disconnection, it will automatically reconnect. ECM_DEMO_AUTO is set to automatically execute the dialing process by default, so it is suitable for programs that need to start automatically at boot.

To configure ECM mode for internet access, if you are using pppd for internet access, please first execute the disconnect script to disconnect the pppd dial-up internet access, and then execute the following instructions to configure it as an ECM mode connection to the network.

```
./disconnect
./ECM_DEMO -t up
```

```
[root@ATK-DLMP135 ~/shell/4G]#./ECM_DEMO -t up
[01_03_18:02:34] ECM_DEMO success
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-9 First disconnect the PPP-over-ON connection, then execute the ECM-over-ON connection.

Use the ifconfig command to check the IP address obtained by usb0. If no IP address is obtained, please use udhcpc -i usb0 to obtain it.

```
ifconfig
```

```
[root@ATK-DLMP135 ~/shell/4G]#ifconfig
eth0      Link encap:Ethernet  HWaddr 88:C1:D3:8F:05:2B
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:46 Base address:0x4000

eth1      Link encap:Ethernet  HWaddr 88:A5:5F:E6:F0:A4
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:53 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:620 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31766 (31.0 KiB)  TX bytes:31766 (31.0 KiB)

usb0      Link encap:Ethernet  HWaddr 0E:C4:19:65:1D:D2
          inet addr:10.21.158.221  Bcast:10.21.158.223  Mask:255.255.255.252
          inet6 addr: fe80::1e4c:6609:9ee0:cf0c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2426 (2.3 KiB)  TX bytes:7826 (7.6 KiB)

[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-10 Check IP address

Ping Baidu to test connectivity. "-I" refers to the specified network card. If there is a data reply, it indicates connectivity. Press Ctrl+c to terminate the command.

```
ping www.baidu.com -I usb0
//-The -I parameter specifies the network card name as usb0 (for 4G network)
```

```
[root@ATK-DLMP135 ~/shell/4G]#ping www.baidu.com -I usb0
PING www.baidu.com (183.232.231.174): 56 data bytes
64 bytes from 183.232.231.174: seq=0 ttl=55 time=38.689 ms
64 bytes from 183.232.231.174: seq=1 ttl=55 time=28.425 ms
64 bytes from 183.232.231.174: seq=2 ttl=55 time=28.515 ms
64 bytes from 183.232.231.174: seq=3 ttl=55 time=25.528 ms
64 bytes from 183.232.231.174: seq=4 ttl=55 time=28.113 ms
64 bytes from 183.232.231.174: seq=5 ttl=55 time=27.884 ms
64 bytes from 183.232.231.174: seq=6 ttl=55 time=27.414 ms
64 bytes from 183.232.231.174: seq=7 ttl=55 time=27.319 ms
64 bytes from 183.232.231.174: seq=8 ttl=55 time=27.057 ms
^C
--- www.baidu.com ping statistics ---
9 packets transmitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 25.528/28.771/38.689 ms
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-11 ping Baidu to test internet connection

If you need to check if 4G is online (reconnect after disconnection), it is recommended to use the command "./ECM_DEMO_AUTO" and run it in the background. After connecting, if you don't obtain an IP address, please use "udhcpc -i usb0" to get it.

```
[root@ATK-DLMP135 ~/shell/4G]#./ECM_DEMO_AUTO &
[root@ATK-DLMP135 ~/shell/4G]#[01_03_18:23:29] ECM_DEMO_AUTO VERSION:ECM_CALLV1.0.1B05 start...
[01_03_18:23:33] ECM_DEMO_AUTO GET IMEI=[869980056950523]
[01_03_18:23:33] ECM_DEMO_AUTO GET ICCID=[89860000192080744 68]
[01_03_18:23:33] ECM_DEMO_AUTO GET SIM_STATUS=[SIM_READY]
[01_03_18:23:37] ECM_DEMO_AUTO Network Connected,NetType[LTE],SignalStrength[88]

[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.15-12 ECM_DEMO_AUTO Internet connection

4.16 TF(SD) Card Test

Instruction Prompt:

The "time" command is commonly used to measure the running time of a command. "dd" is used for copying, reading from the "if" (input file) file and writing to the "of" (output file) specified file. "bs" is the size of each write block, and "count" is the number of read/write blocks. "if=/dev/zero" does not generate I/O, meaning it can continuously output data, so it can be used to test pure write speed.

Before the development board starts, insert the TF card into the TF card slot of the development board. One thing to note is that since the TF card circuit and the SDIO WIFI circuit share the same SDMMC1 bus for communication, and the development board uses a power switching method to use different functions. Therefore, when testing the TF card, the power jumper caps of the TF card and SDIO WIFI need to be switched to the TF card power supply position, and the pin connector interface is marked as J6. Additionally, when testing the TF card, in U-Boot, the wifi&bluetooth device tree cannot be selected to be loaded, and the two cannot be shared.

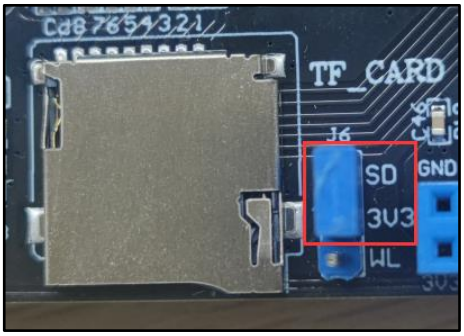


Figure 4.16-1 TF card power supply connection diagram

The author started from eMMC. It can be seen that /dev/mmcblk0 is the device node of the TF card, and /dev/mmcblk0p1 is the default partition of the TF card. Note that a TF card in FAT32 format is required, not NTFS format.

```
[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#[  112.522889] mmc0: host does not support reading read-only switch, assuming write-enable
[  112.534388] mmc0: new high speed SDHC card at address 1234
[  112.543018] mmcblk0: mmc0:1234 SA16G 14.4 GiB        Insert the TF card and check the capacity.
[  112.551264]  mmcblk0: p1
[  112.832715] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.

[root@ATK-DLMP135 ~]#
[root@ATK-DLMP135 ~]#ls /dev/mmcblk*        View the TF card device node and partitions
/dev/mmcblk0    /dev/mmcblk1boot0  /dev/mmcblk1p2  /dev/mmcblk1p5
/dev/mmcblk0p1  /dev/mmcblk1boot1  /dev/mmcblk1p3  /dev/mmcblk1rpmb
/dev/mmcblk1    /dev/mmcblk1p1     /dev/mmcblk1p4
[root@ATK-DLMP135 ~]#
```

Figure 4.16-2 View the TF card device node and partitions

Use the "df -h" command to check the mount path of the TF card. As shown in the following picture, the mount directory of the TF card is "/media/mmcblk0p1".

```
df -h
```

```
[root@ATK-DLMP135 ~]#df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        152M     0  152M   0% /dev
/dev/mmcblk1p5  7.2G  435M  6.5G   7% /
tmpfs           219M     0  219M   0% /dev/shm
tmpfs           219M   68K  219M   1% /tmp
tmpfs           219M  176K  218M   1% /run
/dev/mmcblk1p4   45M   39M  2.1M  95% /lib/modules
/dev/mmcblk0p1   15G  101M   15G   1% /media/mmcblk0p1
[root@ATK-DLMP135 ~]#
```

Use the "df -h" command to check the mount path of the TF card.

Figure 4.16-3 Mount the partition of the TF card to the /mnt directory

4.16.1 Read Speed Test

Commands commonly used to test the reading speed of hard drives include hdparm and dd. Here, only the speed of reading the TF card using hdparm is demonstrated. (Read/write speed is related to the type of the TF card used)

```
hdparm -t /dev/mmcblk0
```

```
[root@ATK-DLMP135 ~]#hdparm -t /dev/mmcblk0

/dev/mmcblk0:
Timing buffered disk reads:   68 MB in 3.01 seconds = 23091 kB/s
[root@ATK-DLMP135 ~]#
```

Figure 4.16-4 Reading speed test

4.16.2 Writing speed test

Use the dd command to write 100MB of data to the mounted directory of the TF card, namely /media/mmcblk0p1. The larger the written data, the closer it is to the actual value.

```
time dd if=/dev/zero of=/media/mmcblk0p1/test bs=1024k count=100 conv=fdatasync
```

```
[root@ATK-DLMP135 ~]#time dd if=/dev/zero of=/media/mmcblk0p1/test bs=1024k coun
t=100 conv=fdatasync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 7.43522 s, 14.1 MB/s
real    0m 7.45s
user    0m 0.00s
sys     0m 1.20s
[root@ATK-DLMP135 ~]#
```

Figure 4.16-5 Write speed test

After the test is completed, you can delete the "test" test file in the mounted directory /media/mmcblk0p1 to complete the write speed test of the TF card.

4.17 OV5640 Camera

- OV5640 Camera Test

Before the experiment, please prepare the OV5640 camera module (5 million pixels), any resolution RCB LCD capacitor screen of our company.

Camera insertion method:

The camera lens is inserted towards the outside of the development board and directly into the J5 connector port, which is the CAMERA interface. Due to the lack of anti-reverse insertion design, when inserting the camera, you need to pay attention to the silk screen on the baseboard and insert according to the pin number, that is, the lens is facing the outside of the board.

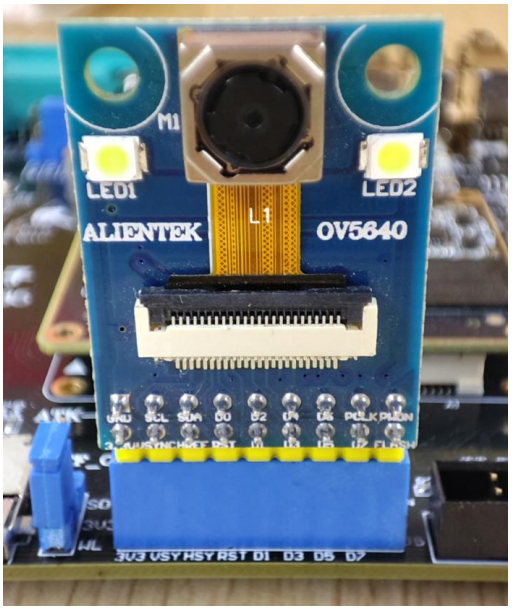


Figure 4.17-1 Physical connection of the camera

Before powering on, insert the camera and LCD screen (hot swapping is not supported), connect the equipment properly, and then start the development board.

```
[    8.003706] ov5640 1-003c: supply AVDD not found, using dummy regulator
[    8.043218] dcmipp 5a000000.dcmipp: Probe done
[    8.100284] ov5640 1-003c: supply DVDD not found, using dummy regulator
[    8.771823] dcmipp 5a000000.dcmipp: bound dcmipp-parallel.0.auto (ops dcmipp_par_pdrv_exit [dcmipp_parallel])
[    8.785979] dcmipp 5a000000.dcmipp: bound dcmipp-byteproc.1.auto (ops dcmipp_byteproc_pdrv_exit [dcmipp_byteproc])
[    8.796384] dcmipp 5a000000.dcmipp: bound dcmipp-bytecap.2.auto (ops dcmipp_bytecap_comp_ops [dcmipp_bytecap])
done
```

Figure 4.17-2 The information printed by the ov5640 driver during startup

Check the nodes generated by ov5640, and confirm that the node is video0.

```
ls /dev/video0
```

```
[root@ATK-DLMP135 ~]#ls /dev/video0
/dev/video0
[root@ATK-DLMP135 ~]#
```

Figure 4.17-3 OV5640 node

Open the "Camera" application in the QT interface, and you can access the OV5640 camera to capture images. As shown in the figure below.

Figure 4.17-4 QT interface "Camera" application

4.18 ADC test

Note: The absolute value of the ADC acquisition voltage is the maximum 3.3V. Please do not exceed 3.3V, otherwise it may cause damage to the chip. There is a single ADC path at J1 on the ATK-DLMP135 baseboard for users to test. The baseboard schematic is as follows.

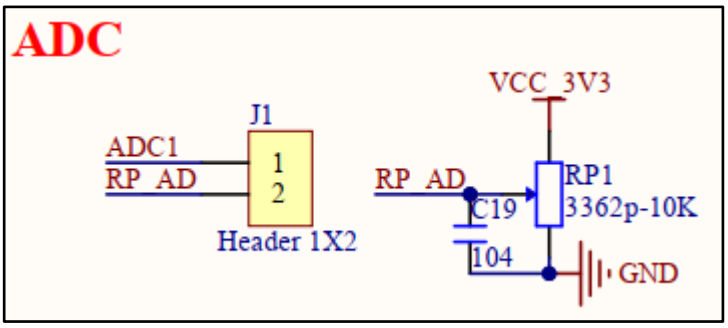


Figure 4.18-1 ADC Board Schematic Diagram

The following figure shows the location of the ADC on the ATK-DLMP135 development board.

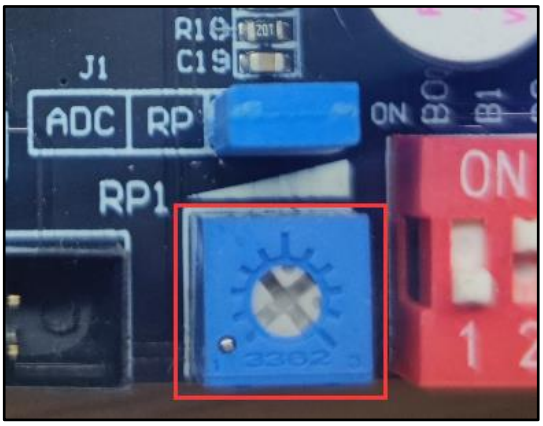


Figure 4.18-2 Development board ADC position

Main characteristics of ADC:

- STM32MP135D supports ADC functionality, providing two ADC controllers, with sampling accuracy configurable as 12, 10 and 8.

In the Linux system, ADC sampling belongs to the IIO subsystem. It can be accessed through the interface provided by sysfs, using 12-bit precision.

Use the following command to obtain the data of ADC1 sampling. The value read out here is the raw value of "raw_value".

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
```

```
[root@ATK-DLMP135 ~]#cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
2507
[root@ATK-DLMP135 ~]#
```

Figure 4.18-3 Check the value of "raw"

The following calculation formula needs to be used to convert it into the actual value.

```
real_value = (raw_value + offset_value) * scale
```

The value of "offset" can be obtained through the following instruction.

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage_offset
```

```
[root@ATK-DLMP135 ~]#cat /sys/bus/iio/devices/iio\:device0/in_voltage_offset
0
[root@ATK-DLMP135 ~]#
```

Figure 4.18-4 Check the value of offset

The value of scale can be obtained through the following instruction. This value is related to the reference voltage.

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage_scale
```

```
[root@ATK-DLMP135 ~]#cat /sys/bus/iio/devices/iio\:device0/in_voltage_scale
0.805664062
[root@ATK-DLMP135 ~]#
```

Figure 4.18-5 Check the scale value

4.19 CPU temperature

Use the following command to check the CPU temperature. The temperature level is related to the ambient temperature and the operating condition of the CPU. For example, when running large applications or playing videos and audio, the temperature will be higher. It is not recommended to touch the surface of the CPU with your hands to prevent static electricity from causing the chip to burn out.

```
cat /sys/class/hwmon/hwmon0/temp1_input
```

```
[root@ATK-DLMP135 ~]#cat /sys/class/hwmon/hwmon0/temp1_input
41406
[root@ATK-DLMP135 ~]#
```

Figure 4.19-1 Check the CPU temperature

4.20 CPU frequency

The processor of the ALIENTEK ATK-DLMP135 development board can support a maximum frequency of 1GHz. However, the factory system kernel configures the CPU frequency adjustment mode as "ondemand". This mode will periodically check the load situation and automatically adjust the frequency based on the load level. The system runs at 650MHz under light load and at 1GHz under heavy load (such as playing videos, etc.).

Under the default factory system's light load condition, the CPU will work at 650MHz for most of the time. The CPU may operate between 650MHz and 1GHz throughout the entire period.

Enter the following command to view the default CPU frequency of the factory system.

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
```

```
[root@ATK-DLMP135 ~]#cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
650000
[root@ATK-DLMP135 ~]#
```

Figure 4.20-1 Check the current CPU main frequency

Enter the following command to view the available main frequency. This processor provides two main frequencies by default: 650 MHz and 1000 MHz.

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

```
[root@ATK-DLMP135 ~]#cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_
frequencies
650000 1000000
[root@ATK-DLMP135 ~]#
```

Figure 4.20-2 Check the available CPU frequency

Then, is it possible to modify the CPU frequency without recompiling the Linux source code? The answer is yes!

Enter the following command to apply for user-space control of the CPU frequency.

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Enter the following instruction to modify the CPU frequency to 1000 MHz, which is 1 GHz. Note that not all frequencies are supported; it must be one of the available frequencies supported above!

```
echo 1000000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

Check the current CPU clock frequency again.

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
```

```
[root@ATK-DLMP135 ~]#cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
1000000
[root@ATK-DLMP135 ~]#
```

Figure 4.20-3 Check the current CPU frequency

4.21 View system information

4.21.1 View the system hostname

```
cat /etc/hostname
```

```
[root@ATK-DLMP135 ~]#cat /etc/hostname
ATK-DLMP135
[root@ATK-DLMP135 ~]#
```

Figure 4.21-1 View the system hostname

4.21.2 View CPU-related information

```
cat /proc/cpuinfo
```

```
[root@ATK-DLMP135 ~]#cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 48.00
Features        : half thumb fastmult vfp edsp thumbee neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

Hardware        : STM32 (Device Tree Support)
Revision        : 0000
Serial          : 0005800B3331511434323331
[root@ATK-DLMP135 ~]#
```

Figure 4.21-2 View CPU-related information

The BogoMIPS value shown in the figure is 48, which is the value based on the CPU's main frequency of 1GHz. When the CPU's main frequency is 650MHz, the BogoMIPS value is 31.2.

4.21.3 View memory-related information

```
cat /proc/meminfo
```

```
[root@ATK-DLMP135 ~]#cat /proc/meminfo
MemTotal:         447684 kB
MemFree:          338744 kB
MemAvailable:     390400 kB
Buffers:            4008 kB
Cached:            52412 kB
SwapCached:            0 kB
Active:             8352 kB
Inactive:          68820 kB
Active(anon):        116 kB
Inactive(anon):    21996 kB
Active(file):       8236 kB
Inactive(file):    46824 kB
Unevictable:           0 kB
Mlocked:               0 kB
HighTotal:             0 kB
HighFree:              0 kB
LowTotal:         447684 kB
LowFree:          338744 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                 4 kB
Writeback:             0 kB
AnonPages:         20780 kB
Mapped:            32128 kB
Shmem:              1360 kB
KReclaimable:       7744 kB
Slab:              18568 kB
SReclaimable:       7744 kB
SUnreclaim:        10824 kB
KernelStack:         568 kB
PageTables:          408 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:      223840 kB
Committed_AS:      48544 kB
VmallocTotal:     557056 kB
VmallocUsed:        2140 kB
VmallocChunk:          0 kB
Percpu:               64 kB
CmaTotal:         131072 kB
CmaFree:          128628 kB
[root@ATK-DLMP135 ~]#
```

Figure 4.21-3 View memory-related information

The memory-related information shown in the figure is based on the actual system version of the user.

4.22 Shutting down and restarting

- Shutting down

To shut down the system, simply enter the "poweroff" command in the terminal. Before shutting down, please save important data. Enter "sync" to synchronize the data. You can also save the data and then directly turn off the power to shut down the system.

```
sync
poweroff
```

- Reboot

To restart, simply enter the "reboot" command in the terminal. Before shutting down, please save important data. Enter "sync" to synchronize the data. You can also save the data and then directly press the reset button (the red button on the board) to restart the board.

```
sync
```

```
reboot
```

4.23 Check the factory file system version

To view the factory file system version, execute the following command. The file system will be updated and iterated according to the actual system version of the user. This is for reference only. If the file system version cannot be obtained, please update the file system to obtain the version information of the file system.

```
cat /etc/version
```

```
[root@ATK-DLMP135 /]#cat /etc/version
初始版本-V1.0.0-2023.03.21
[root@ATK-DLMP135 /]#
```

Figure 4.23-1 Check the factory system version

4.24 4G module EC20 test

The 4G module interface is reserved on the ALIENTEK ATK-DLMP135 baseboard. The installation of 4G modules such as ME3630-W, EC20, etc. is prepared. Prepare the EC20 module. Please purchase it yourself online. Note that when purchasing, you need to buy an antenna. Just the module alone cannot work properly! (Note: There are many types of EC20 modules. Currently, the tested type is EC20-CE module. Among the EC20-CE series, there are various modules, and different modules have different functions. For example, the supported operators are different. For details, please consult the seller). Other EC20 series modules should be tested by yourself. Theoretically, the drivers are the same. If there is a need, contact Miao Yuan for technical support.) 。 Insert the EC20 4G module into the 4G module interface and screw it in. Ensure that the 4G module is properly connected and aligned with the interface of the base. Please use the original antenna and connect it to the MAIN interface of the 4G module.

Insert the 4G card correctly (for supported operators, please consult the seller of the corresponding module. Note that some modules may not support IoT cards. Please use a regular 4G card for testing). Plug in the module and after the development board starts, the WWAN LED on the base will light up green. If the WWAN LED does not light up, please check if the module is properly connected and inserted, if the 4G card is inserted, if the antenna is connected, and the development board must be plugged into the 12V power supply provided.

After the 4G module is properly loaded, you can see 4 ttyUSB* generated under /dev. The 4G module communicates through AT commands with /dev/ttyUSB2.

```
ls /dev/ttyUSB*
```

```
[root@ATK-DLMP135 ~]#ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2  /dev/ttyUSB3
[root@ATK-DLMP135 ~]#
```

Figure 4.24-1 Check the interface of the 4G module

4.24.1 Use quectel-CM

Use the quectel-CM dialing program tool (this tool has been pre-cross-compiled and placed in the directory /usr/sbin of the file system), which is convenient for users to use.

Enter quectel-CM -h to view the usage.

```
quectel-CM -h
```

```
[root@ATK-DLMP135 ~/shell/4G]#quectel-CM -h
[01-03_19:36:58:883] Quectel_QConnectManager_Linux_V2.0.0
[01-03_19:36:58:884] Usage: quectel-CM [options]
[01-03_19:36:58:884] -s [apn [user password auth]]          Set apn/user/password/auth get from your network provider
[01-03_19:36:58:884] -p pincode                             Verify sim card pin if sim card is locked
[01-03_19:36:58:884] -f logfilename                         Save log message of this program to file
[01-03_19:36:58:884] -i interface                           Specify network interface(default auto-detect)
[01-03_19:36:58:884] -4                                     IPv4 protocol
[01-03_19:36:58:884] -6                                     IPv6 protocol
[01-03_19:36:58:884] -m muxID                               Specify muxid when set multi-pdn data connection.
[01-03_19:36:58:884] -n channelID                           Specify channelID when set multi-pdn data connection(default 1).
[01-03_19:36:58:884] [Examples]
[01-03_19:36:58:884] Example 1: quectel-CM
[01-03_19:36:58:884] Example 2: quectel-CM -s 3gnet
[01-03_19:36:58:884] Example 3: quectel-CM -s 3gnet carl 1234 0 -p 1234 -f gobinet_log.txt
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.24-2 Check the quectel-CM tool instructions.

You can see that the -s parameter specifies the APN type. The APN for mobile SIM cards is usually cmnet, for China Unicom SIM cards it is usually 3gnet, and for China Telecom SIM cards it is usually ctnet. Note: APN refers to a network access technology, which is usually a parameter that must be configured when using a mobile phone to access the internet. It determines the access method through which the mobile phone accesses the network.

```
quectel-CM &
// If you are not sure, just enter "quectel-CM,&". The function of this is to run in the background.
```

```
[root@ATK-DLMP135 ~/shell/4G]#quectel-CM &
[root@ATK-DLMP135 ~/shell/4G]#[01-03_19:37:45:337] Quectel_QConnectManager_Linux_V2.0.0
[01-03_19:37:45:341] Find /sys/bus/usb/devices/2-1.4 idVendor=0x2c7c idProduct=0x125
[01-03_19:37:45:341] Auto find qmichannel = /dev/qcqmi2
[01-03_19:37:45:341] Auto find usbnet_adapter = eth2
[01-03_19:37:45:342] qmichannel=/dev/qcqmi2, usbnet_adapter=eth2
[01-03_19:37:45:342] Modem works in QMI mode
[01-03_19:37:45:369] Get clientWDS = 7
[01-03_19:37:45:401] Get clientDMS = 8
[01-03_19:37:45:433] Get clientNAS = 9
[01-03_19:37:45:465] Get clientUIM = 10
[01-03_19:37:45:498] Get clientWDA = 11
[01-03_19:37:45:530] requestBaseBandVersion EC20CEFAGR06A10M4G
[01-03_19:37:45:658] requestGetSIMStatus SIMStatus: SIM_READY
[01-03_19:37:45:690] requestGetProfile[1] CMNET///0
[01-03_19:37:45:722] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE
[01-03_19:37:45:753] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[01-03_19:37:45:753] ifconfig eth2 down
[01-03_19:37:45:766] ifconfig eth2 0.0.0.0
[ 1629.406428] IPv6: ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready
[01-03_19:37:45:850] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE
[01-03_19:37:46:810] requestSetupDataCall WdsConnectionIPv4Handle: 0x87353690
[01-03_19:37:46:906] requestQueryDataCall IPv4ConnectionStatus: CONNECTED
[01-03_19:37:46:939] ifconfig eth2 down
[01-03_19:37:46:967] busybox udhcpc -f -n -q -t 5 -i eth2
udhcpc: started, v1.35.0
udhcpc: broadcasting discover
udhcpc: broadcasting select for 10.226.105.20, server 10.226.105.21
udhcpc: lease of 10.226.105.20 obtained from 10.226.105.21, lease time 7200
[01-03_19:37:47:287] deleting routers
[01-03_19:37:47:327] adding dns 221.179.38.7
[01-03_19:37:47:327] adding dns 120.196.165.7
[ 1631.632678] IPv6: ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready

[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.24-3 Access the internet via the quectel-CM tool

Press Enter to return to the command line, then input the "ifconfig" command to view the network node name of the EC20 4G module. "eth0" and "eth1" are the names of the two Ethernet network ports on the board, so "eth2" is the network node name of the EC20 4G module.

```
[root@ATK-DLMP135 ~/shell/4G]#ifconfig
eth0      Link encap:Ethernet  HWaddr 88:C1:D3:8F:05:2B
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1326 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:102625 (100.2 KiB)  TX bytes:6438 (6.2 KiB)
          Interrupt:46 Base address:0x4000

eth1      Link encap:Ethernet  HWaddr 88:A5:5F:E6:F0:A4
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:53 Base address:0xa000

eth2      Link encap:Ethernet  HWaddr EA:1B:D1:81:E6:6F
          inet addr:10.226.105.20  Bcast:10.226.105.23  Mask:255.255.255.248
          inet6 addr: fe80::f087:84bb:28de:5c57/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:192 errors:0 dropped:8 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1636 (1.5 KiB)  TX bytes:63353 (61.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:64 errors:0 dropped:0 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4673 (4.5 KiB)  TX bytes:4673 (4.5 KiB)

[root@ATK-DLMP135 ~/shell/4G]#
```

EC20 network node name and the obtained IP

Figure 4.24-4 The obtained IP

"ping baidu" to test connectivity. The "-I" indicates the specific network card. A data reply means connectivity.

```
ping www.baidu.com -I eth2
```

```
[root@ATK-DLMP135 ~/shell/4G]#ping www.baidu.com -I eth2
PING www.baidu.com (183.232.231.172): 56 data bytes
64 bytes from 183.232.231.172: seq=0 ttl=56 time=123.166 ms
64 bytes from 183.232.231.172: seq=1 ttl=56 time=36.615 ms
64 bytes from 183.232.231.172: seq=2 ttl=56 time=36.613 ms
64 bytes from 183.232.231.172: seq=3 ttl=56 time=41.846 ms
64 bytes from 183.232.231.172: seq=4 ttl=56 time=41.342 ms
64 bytes from 183.232.231.172: seq=5 ttl=56 time=35.981 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 35.981/52.593/123.166 ms
[root@ATK-DLMP135 ~/shell/4G]#
```

Figure 4.24-5 Ping Baidu test