DWIN Linux Screen Development Guide

(36 Series)

1 Product Introduction

1.1 Product Feature

DWIN Linux screen(36 series), with CAN interface.

CPU： Allwinner T113-S3, Quad-core ARM Cortex-A7

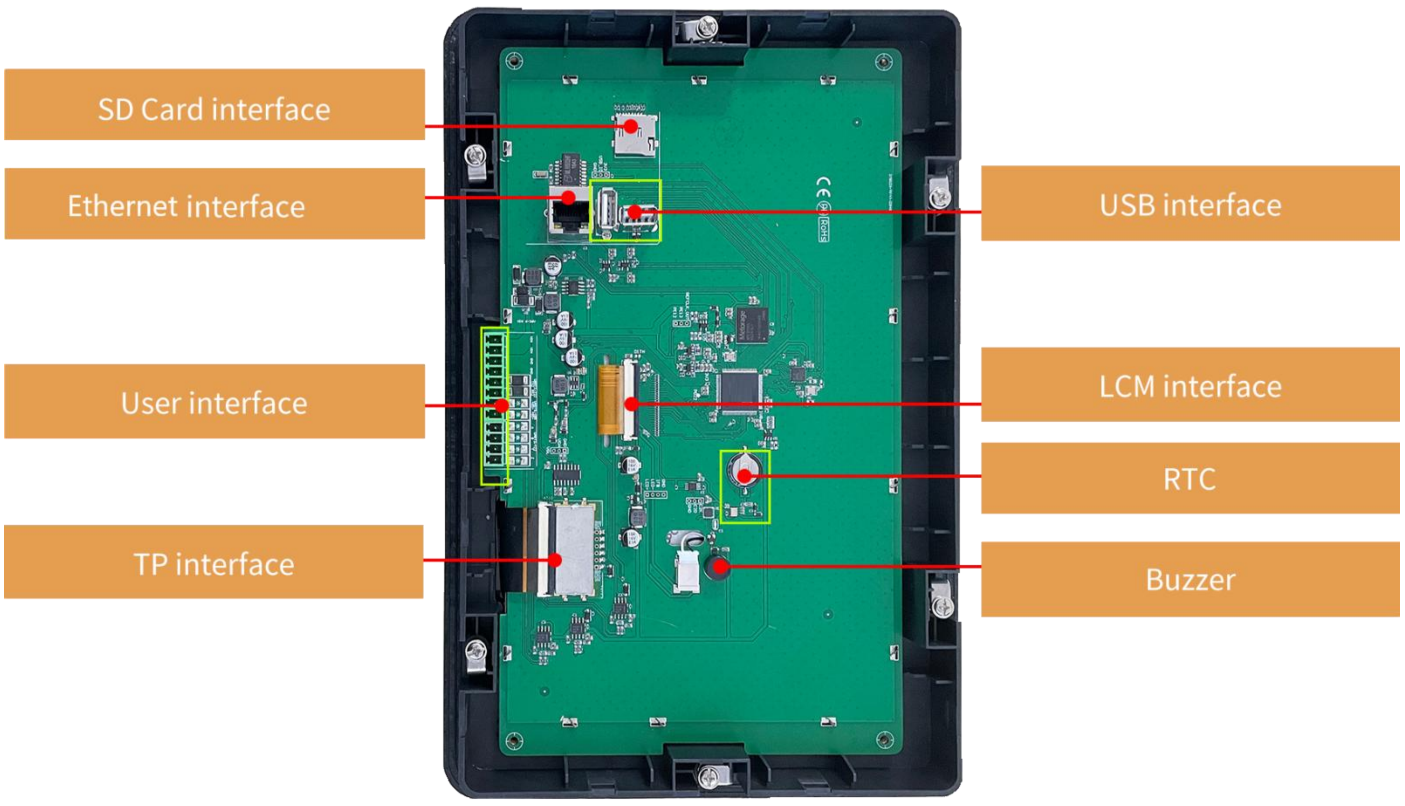
RAM: 128MB DDR3

Storage: 8GB EMMC

Main Frequency: 1GHz

Linux Version: Linux 5.4

DMT10600T101_36WTC



DMT10600T101_36WTC

1.1.2 Development Method

QT and LVGL are optional.

The development materials and tools can be referred to:

Documents: https://www.dwin-global.com/development-guide/

Tool: https://www.dwin-global.com/tool-page/

1.1.3 Shipping List(for reference)

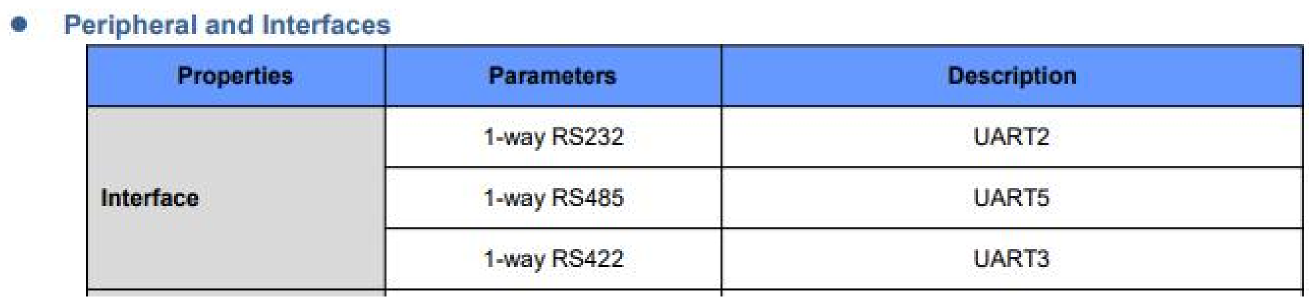
- screen ×1

1.1.4 Optional Accessories

- SD card

1.2 Wiring

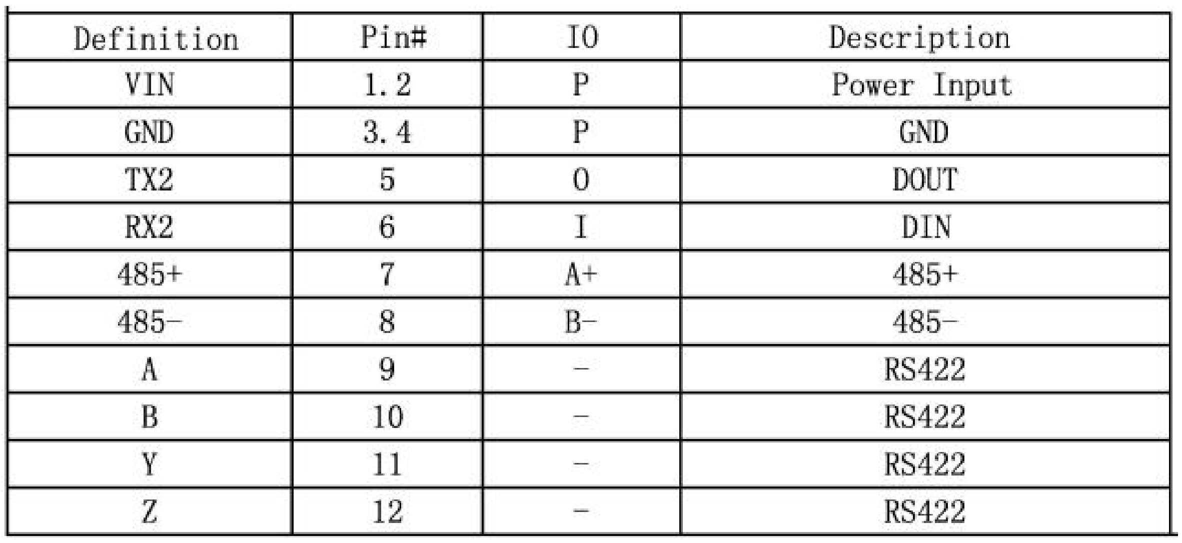
Regarding the definition of serial please refer to the related datasheet as below:



- **Peripheral and Interfaces**

Properties	Parameters	Description
Interface	1-way RS232	UART2
	1-way RS485	UART5
	1-way RS422	UART3

Datasheet_Peripheral and Interfaces



Definition	Pin#	IO	Description
VIN	1. 2	P	Power Input
GND	3. 4	P	GND
TX2	5	O	DOUT
RX2	6	I	DIN
485+	7	A+	485+
485-	8	B-	485-
A	9	-	RS422
B	10	-	RS422
Y	11	-	RS422
Z	12	-	RS422

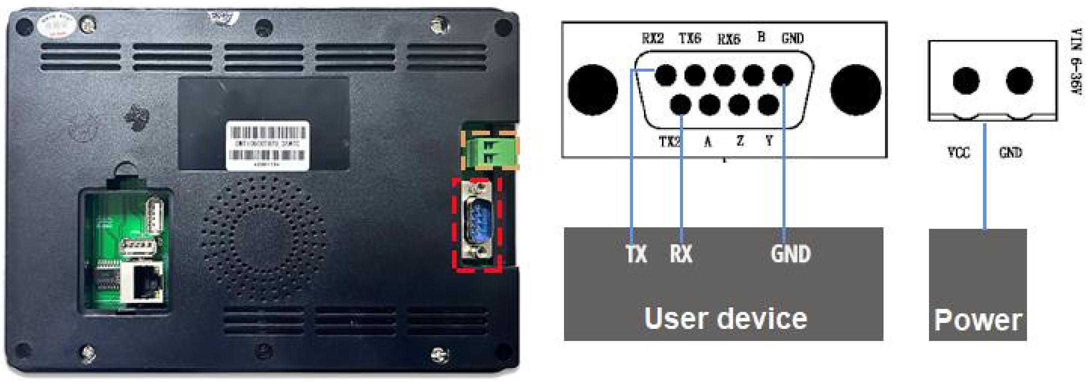
Datasheet_Interface Definition

1.2.1 Hardware Connection

GND, Ground, connect to GND pin of the user device.

TXD, Transmit, connect to RX pin of the user device.

RXD, Receive, connect to TX pin of the user device.



Wiring Schematic Diagram with a DB9 Interface

1.2.2 Serial Parameter Setting

All the defined serial port baud rates are 115200. In general, the default serial port for debugging is Serial Port 0.

1.2.3 Other Tools

DC regulated 12V power supply is recommended for testing, using SD card with 1~16 GB memory for project downloading.

2 Environment Configuration

2.1 Ubuntu16.04 Configuration

2.1.1 Introduction

This section provides a tutorial on installing a virtual machine and configuring Ubuntu 16.04 on it. If already have Ubuntu 16.04 installed, you can skip this section and refer to Section 1.2 for toolchain installation and configuration.

2.1.2 Environment Requirements

CPU: No specific requirements.

Memory: Generally, 2GB or more.

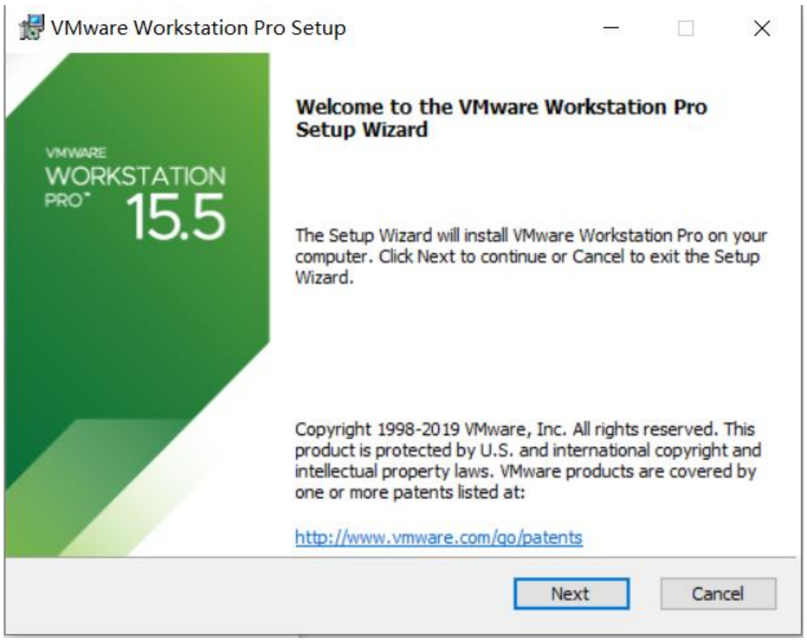
Host Operating System: Windows XP, Windows 7, and above.

Version Selection: Depending on your needs (Windows version), choose VMware Workstation 10 and above. Versions below 10 are not recommended.

Note: This example demonstrates the installation using VMware Workstation 15 Pro. If you have already installed the virtual machine and Ubuntu, you can proceed directly to Section 1.2 for toolchain installation.

2.1.3 VMware Workstation Installation

(1) Running installation package


VMware Workstation Pro Setup

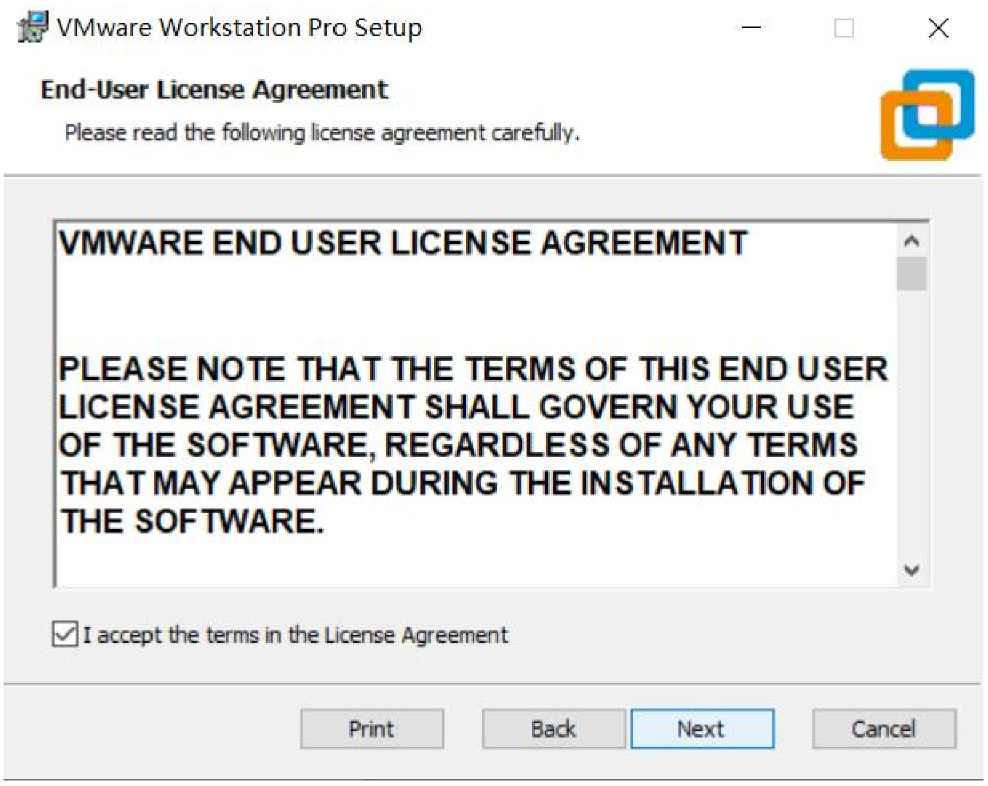
Welcome to the VMware Workstation Pro Setup Wizard

The Setup Wizard will install VMware Workstation Pro on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Copyright 1998-2019 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:

http://www.vmware.com/go/patents

(2) In the End User License Agreement interface, select the checkbox "I accept the terms in the license agreement", and then click "Next".


VMware Workstation Pro Setup

End-User License Agreement

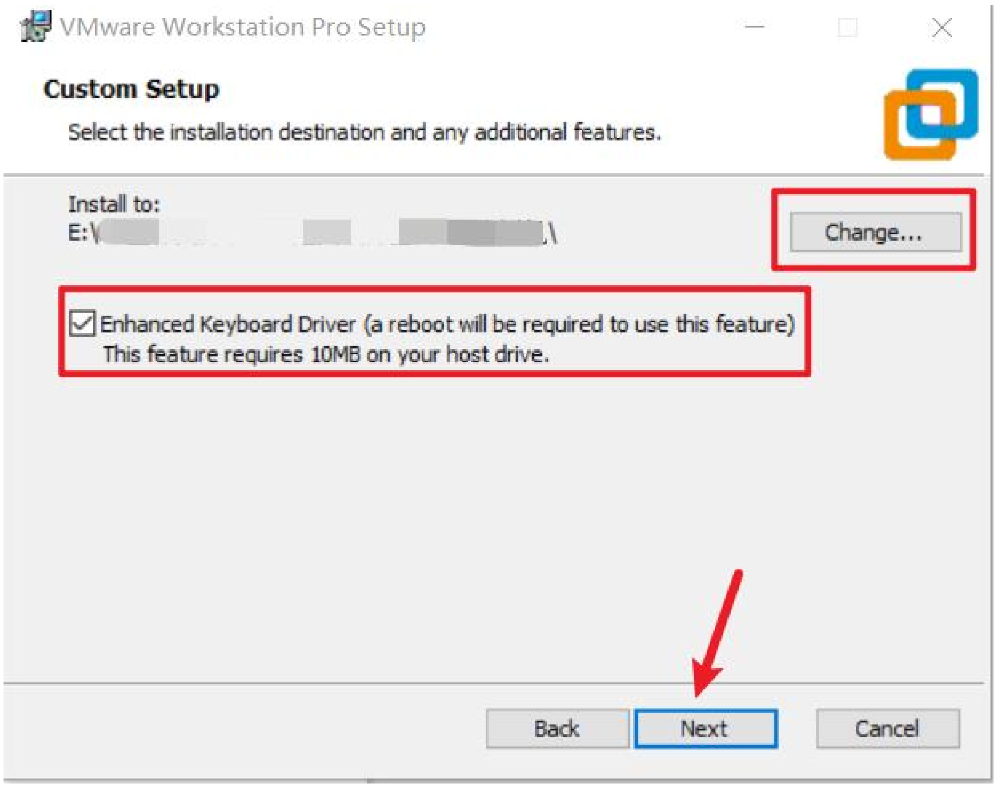
Please read the following license agreement carefully.

VMWARE END USER LICENSE AGREEMENT

PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.

☑ I accept the terms in the License Agreement

(3) Select the installation path (or choose default path), check the "Enhanced keyboard driver" option and click "Next".



VMware Workstation Pro Setup

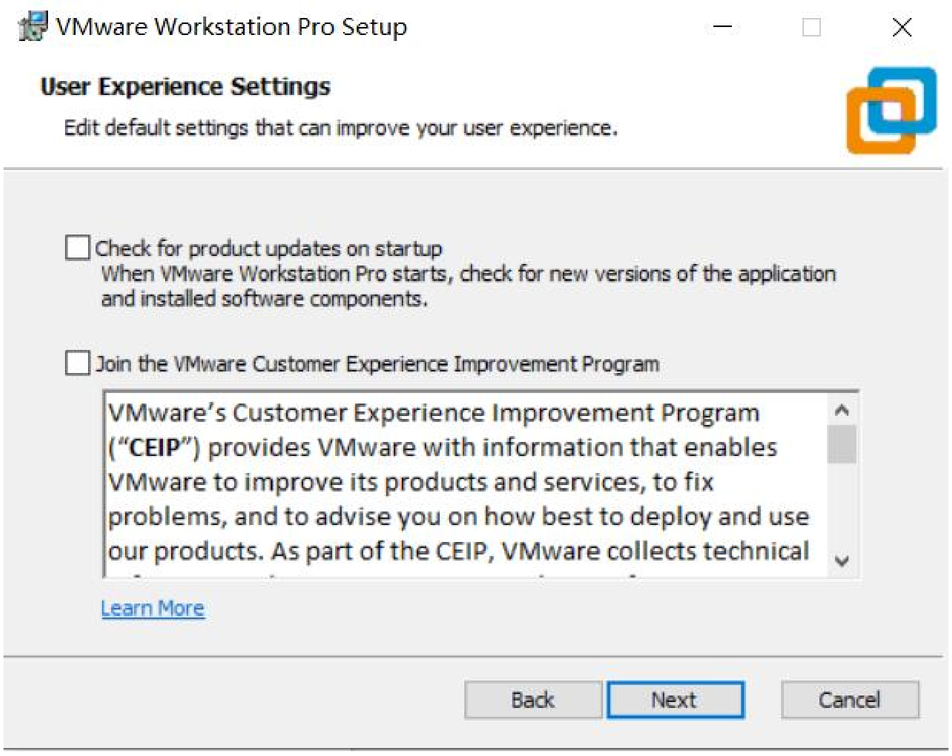
Custom Setup

Select the installation destination and any additional features.

Install to:
E:\[redacted]\

☑ Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.

(4) Appropriately select the checkboxes of "Check for product updates on startup" and "Help improve VMware Workstation Pro" according to your own situation, and then click "Next".



VMware Workstation Pro Setup

User Experience Settings

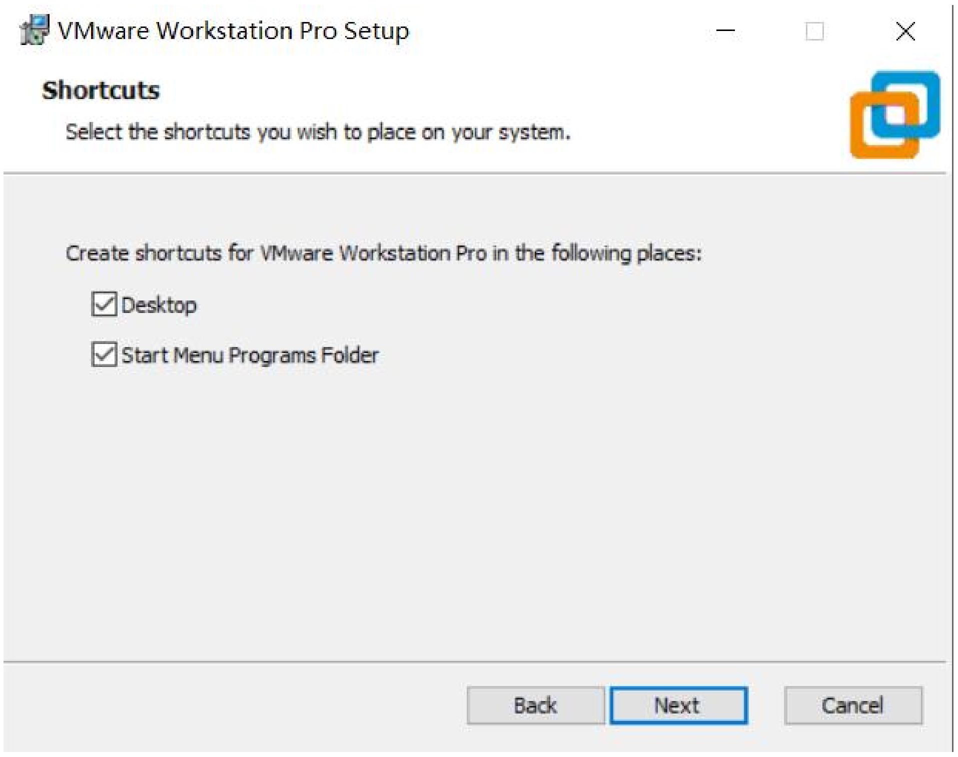
Edit default settings that can improve your user experience.

☐ Check for product updates on startup
When VMware Workstation Pro starts, check for new versions of the application and installed software components.

☐ Join the VMware Customer Experience Improvement Program

VMware's Customer Experience Improvement Program ("**CEIP**") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical

(5) Select the checkboxes for "Desktop" and "Start Menu Programs Folder", and then click "Next".


VMware Workstation Pro Setup

Shortcuts

Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Workstation Pro in the following places:

☑ Desktop

☑ Start Menu Programs Folder

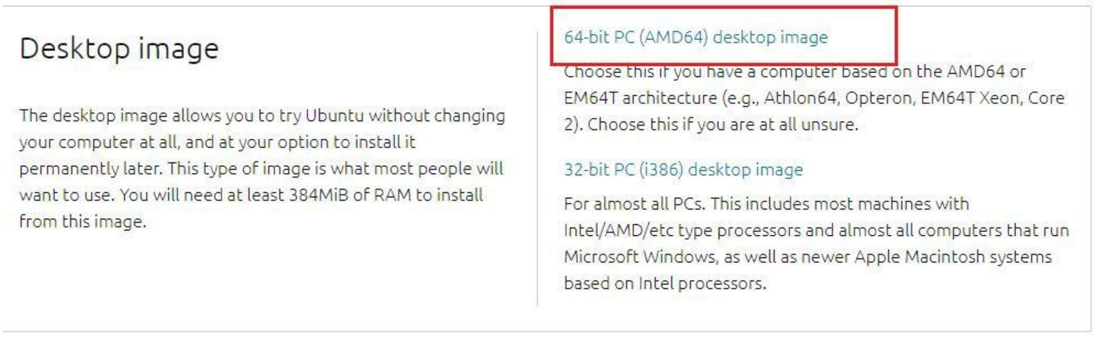
(6) Click "install".

(7) After a period of time, the installation will be completed. Click "Finish".

2.1.4 Download Ubuntu

(1) Ubuntu16.04 from official website: https://releases.ubuntu.com/16.04/

(2) Choose and download 64-bit PC desktop image "ubuntu-16.04.7-desktop-amd64.iso".


Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 384MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

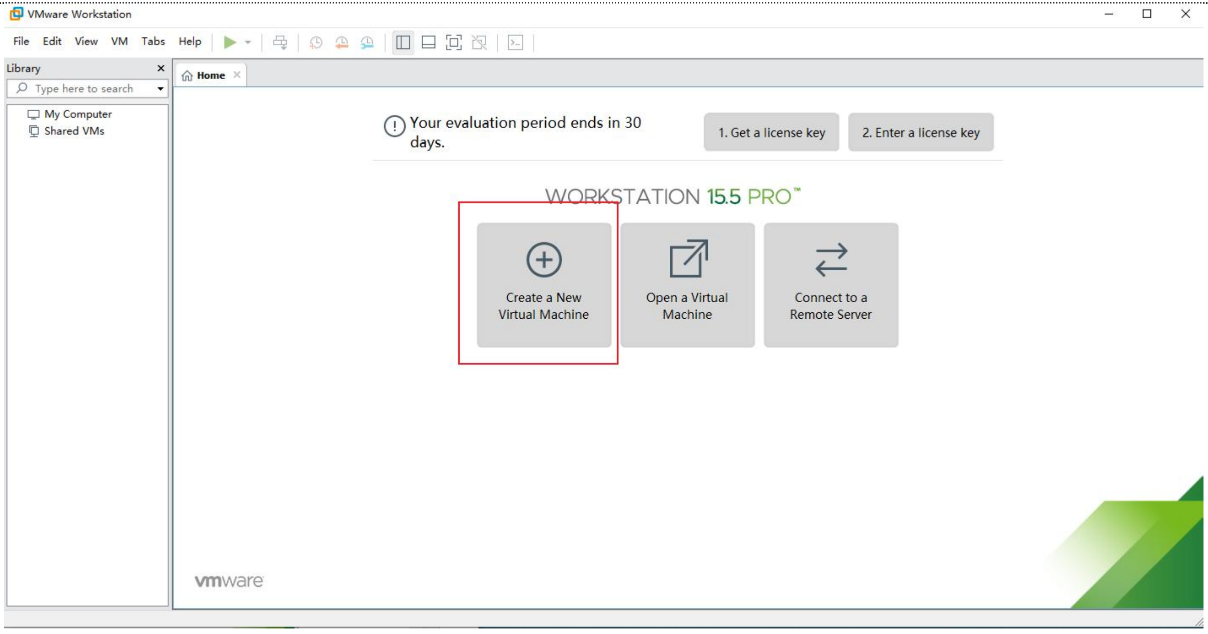
32-bit PC (i386) desktop image

For almost all PCs. This includes most machines with Intel/AMD/etc type processors and almost all computers that run Microsoft Windows, as well as newer Apple Macintosh systems based on Intel processors.

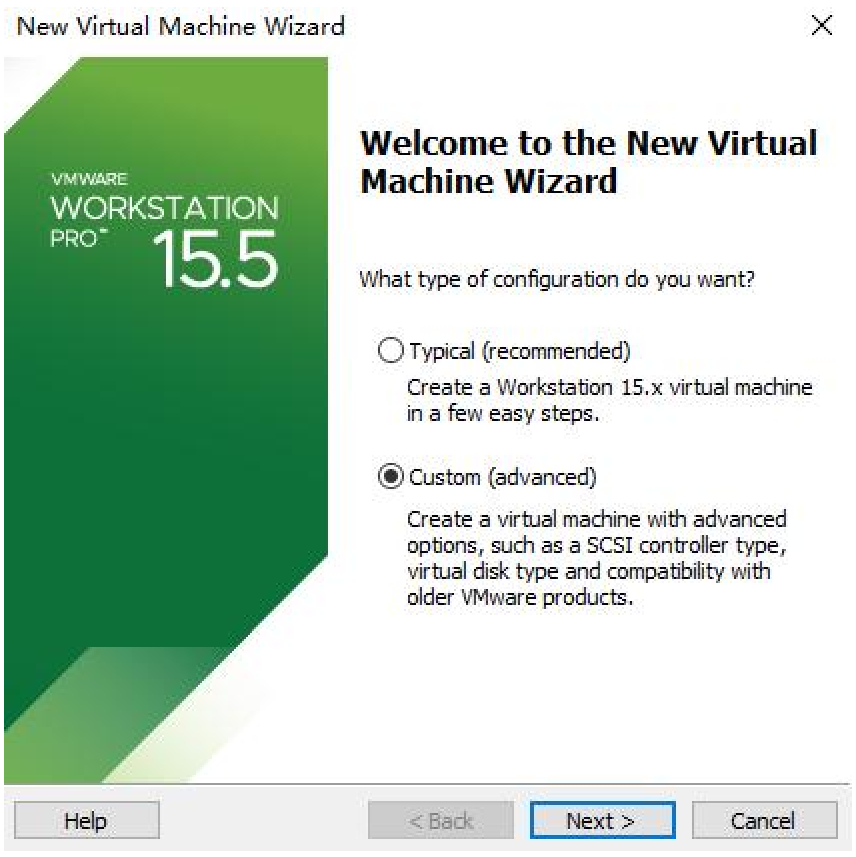
2.1.5 Ubuntu Installation

(1) Open VMware Workstation.

(2) Create a new virtual machine.



(3) Select "custom (advanced)" and click "Next".


New Virtual Machine Wizard

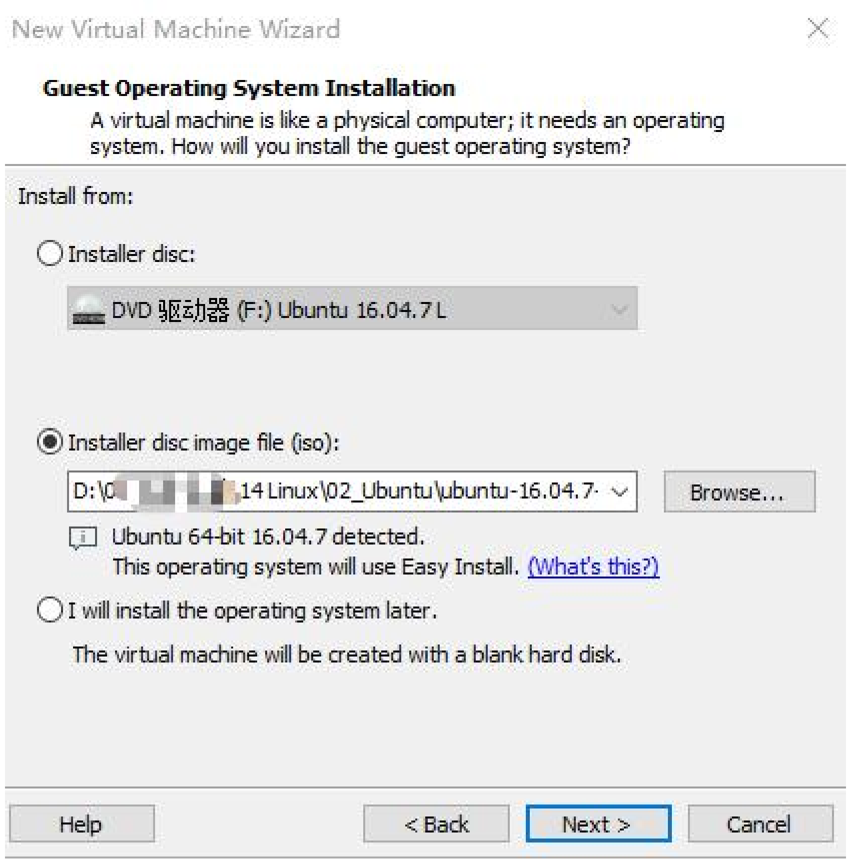
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

○ Typical (recommended)
Create a Workstation 15.x virtual machine in a few easy steps.

● Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

(4) Choose "Installer disc image file (iso)" – "Browse"– select the download file ***.iso containing Ubuntu, it will automatically recognize and read the file, then click "Next".



New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

( ) Installer disc:

DVD 驱动器 (F:) Ubuntu 16.04.7 L

(●) Installer disc image file (iso):

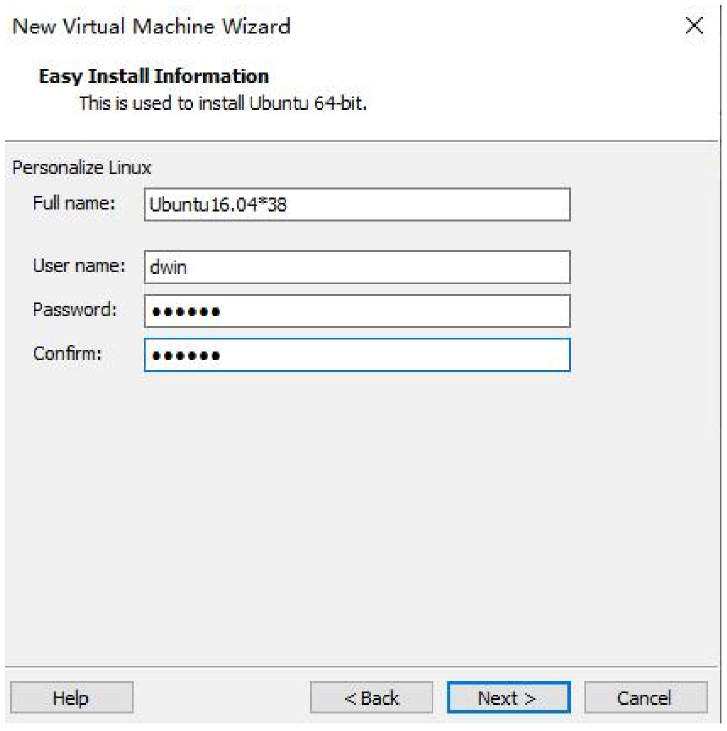
D:\0 14 Linux\02_Ubuntu\ubuntu-16.04.7-

Ubuntu 64-bit 16.04.7 detected.
This operating system will use Easy Install. (What's this?)

( ) I will install the operating system later.

The virtual machine will be created with a blank hard disk.

(5) Input custom name and password, the password will serve as the login password for Ubuntu and the sudo authorization password, then click "Next".



New Virtual Machine Wizard

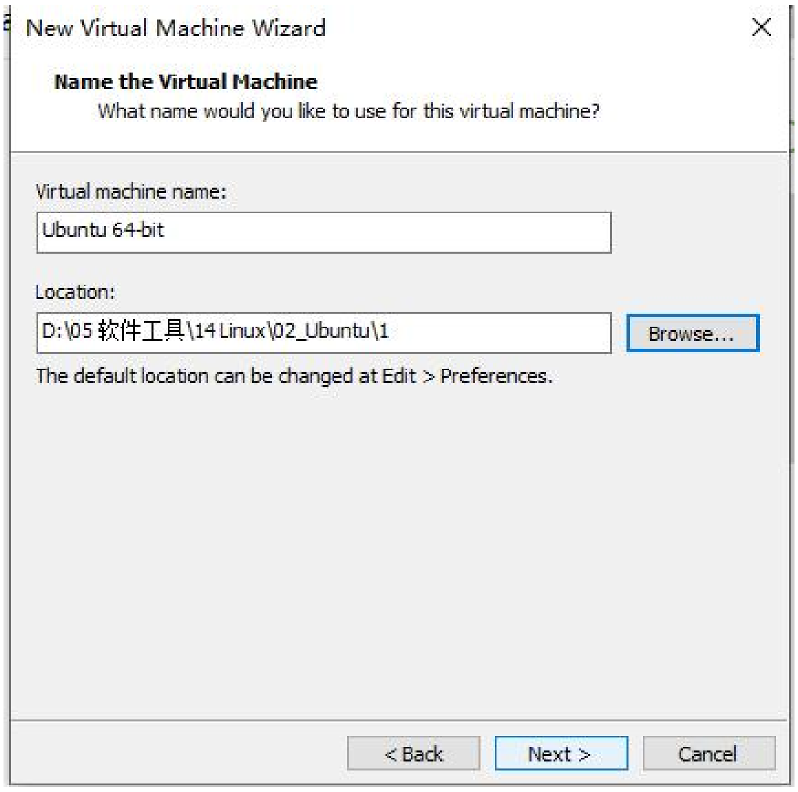
Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

Full name:	Ubuntu16.04*38
User name:	dwin
Password:	••••••
Confirm:	••••••

(6) Setup the name of Ubuntu and location, click "Next".



New Virtual Machine Wizard

Name the Virtual Machine

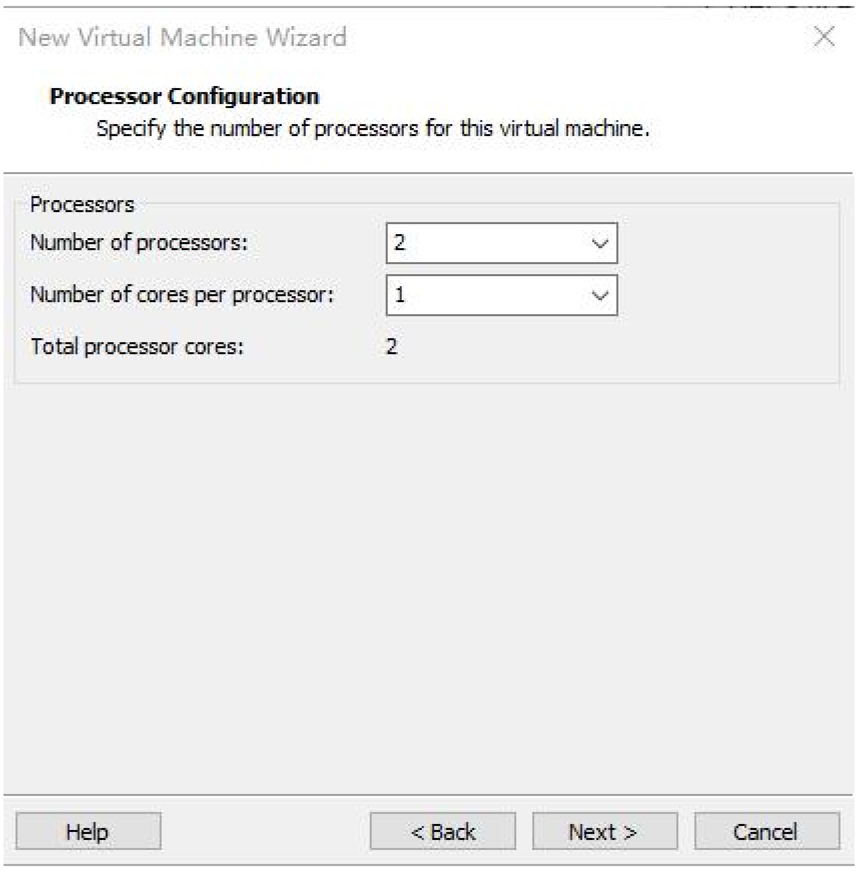
What name would you like to use for this virtual machine?

Virtual machine name: Ubuntu 64-bit

Location: D:\05 软件工具\14 Linux\02_Ubuntu\1

The default location can be changed at Edit > Preferences.

(7) Based on user's requirements and computer configuration, allocate the number of processors and cores(Here, the author sets the total number of processor cores to 2). Click "Next".



New Virtual Machine Wizard

Processor Configuration

Specify the number of processors for this virtual machine.

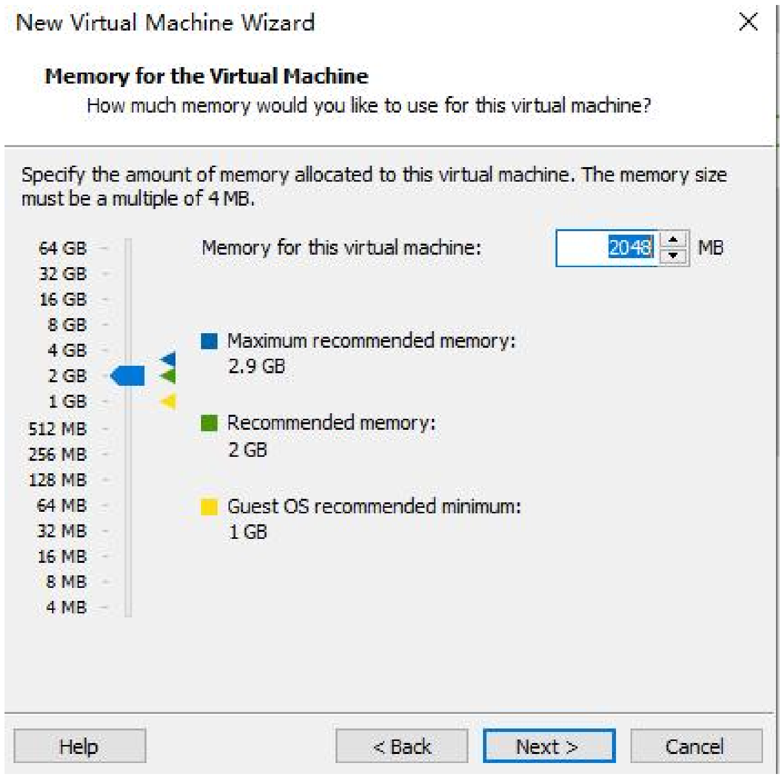
Processors

Number of processors: 2

Number of cores per processor: 1

Total processor cores: 2

(8) The default allocated memory is 2GB (sufficient, can be changed later). Click "Next".


New Virtual Machine Wizard

Memory for the Virtual Machine
How much memory would you like to use for this virtual machine?

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

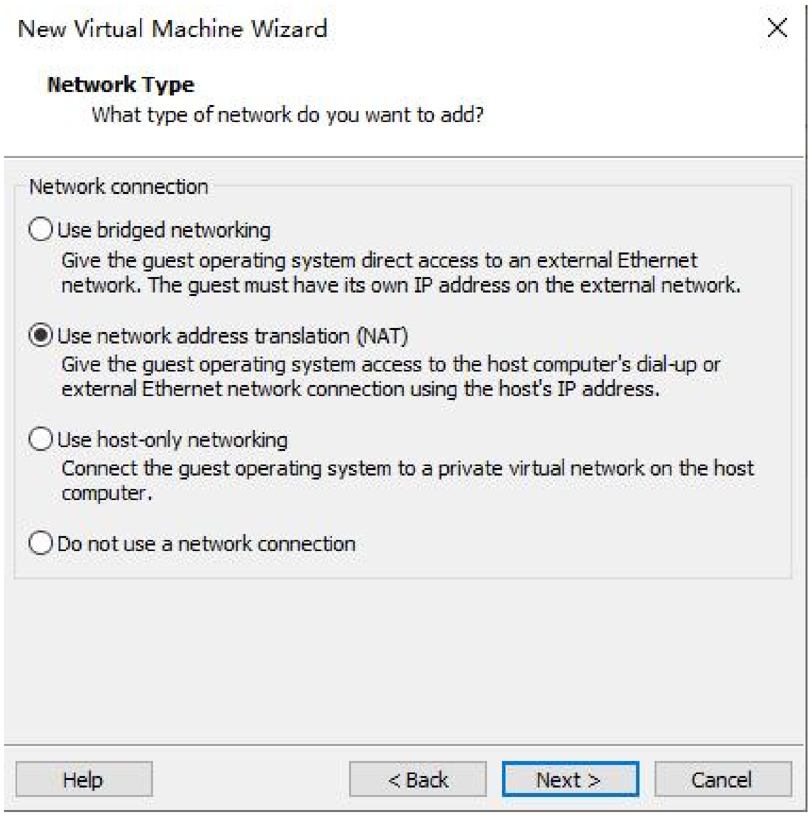
Memory for this virtual machine: 2048 MB

🟦 Maximum recommended memory: 2.9 GB

🟩 Recommended memory: 2 GB

🟨 Guest OS recommended minimum: 1 GB

(9) The default configuration is fine (however, in the network type section here, you can choose the bridged network, which can be used for tftp transmission). Click "Next" until you reach the "Select a Disk" step.

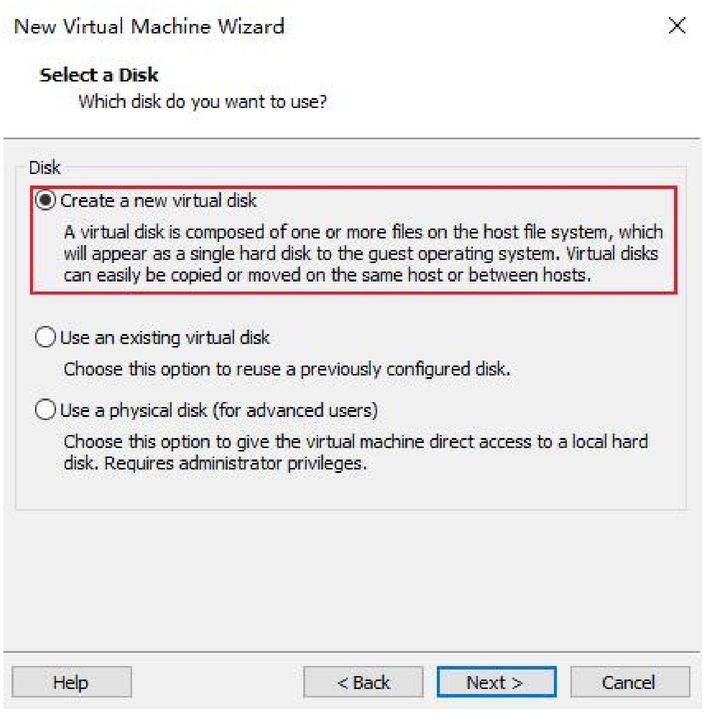

New Virtual Machine Wizard

Network Type
What type of network do you want to add?

Network connection

- ○ Use bridged networking
  Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.
- ● Use network address translation (NAT)
  Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.
- ○ Use host-only networking
  Connect the guest operating system to a private virtual network on the host computer.
- ○ Do not use a network connection

(10) Select "Create a new virtual disk", and click "Next".


New Virtual Machine Wizard

Select a Disk
Which disk do you want to use?

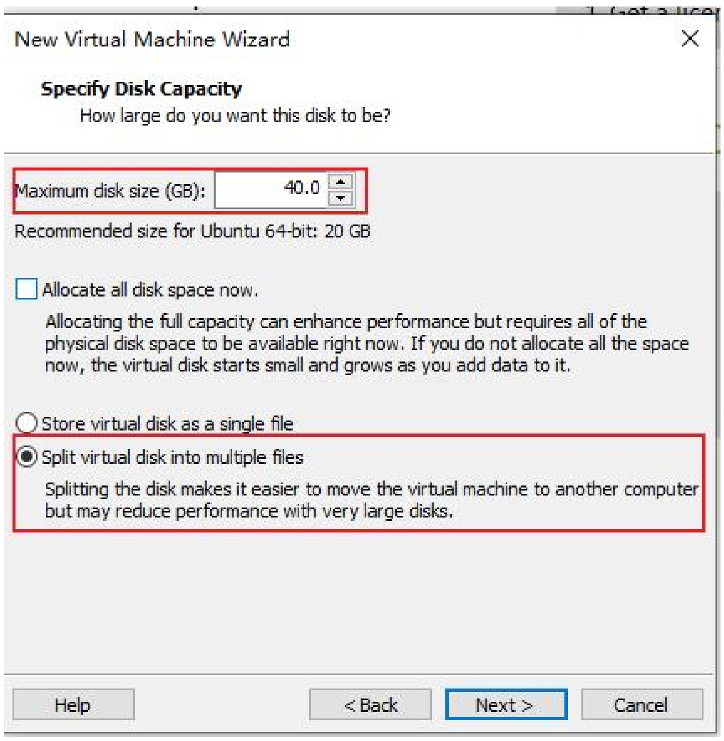
Disk

[✓ Create a new virtual disk]
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

Use an existing virtual disk
Choose this option to reuse a previously configured disk.

Use a physical disk (for advanced users)
Choose this option to give the virtual machine direct access to a local hard disk. Requires administrator privileges.

(11) Set the disk capacity: It is recommended to set the disk capacity according to the size of the computer's memory. When the memory is large, it can be set to more than 30GB. If you need to increase the disk space, please refer to the subsequent chapters for expansion. Click "Next" to continue.


New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB): 40.0
Recommended size for Ubuntu 64-bit: 20 GB

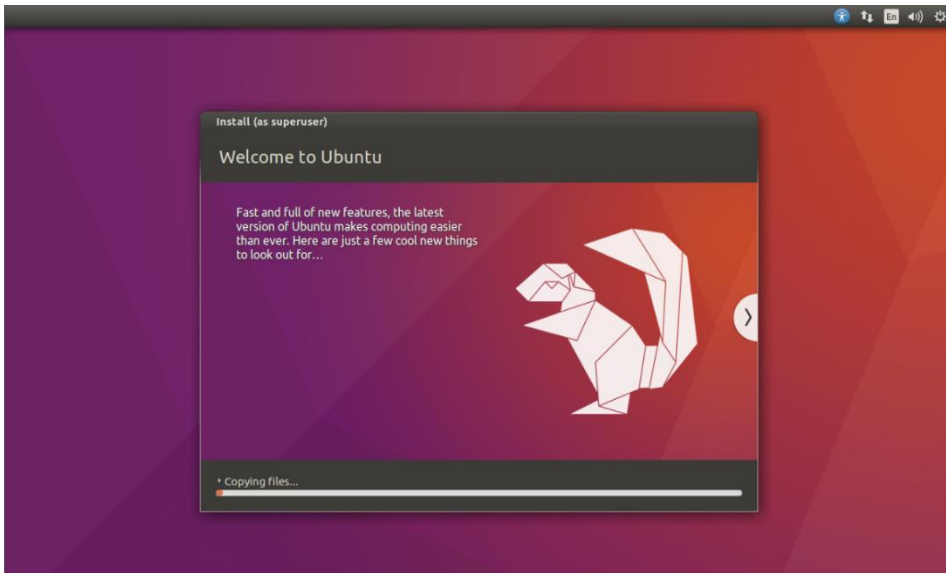
☐ Allocate all disk space now.
Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

Store virtual disk as a single file
[✓ Split virtual disk into multiple files]
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

(12) The disk will be automatically named, keep the default and click "Next".

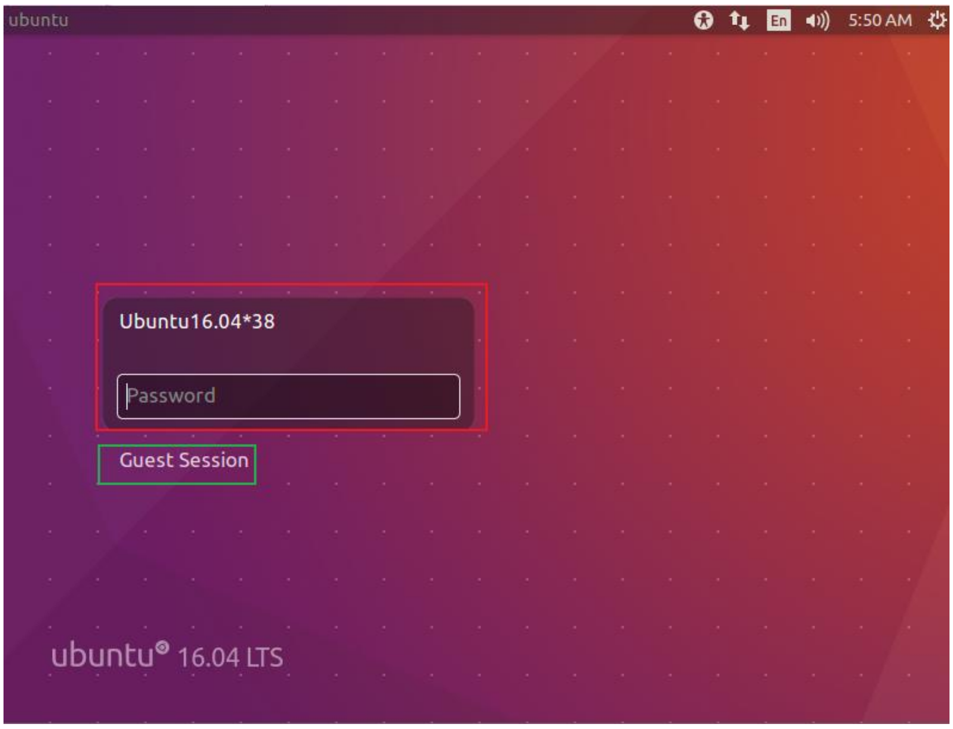
(13) After clicking "Finish", the virtual machine will start, and the installation will begin.

(14) Please wait patiently for a while.



Fast and full of new features, the latest version of Ubuntu makes computing easier than ever. Here are just a few cool new things to look out for...

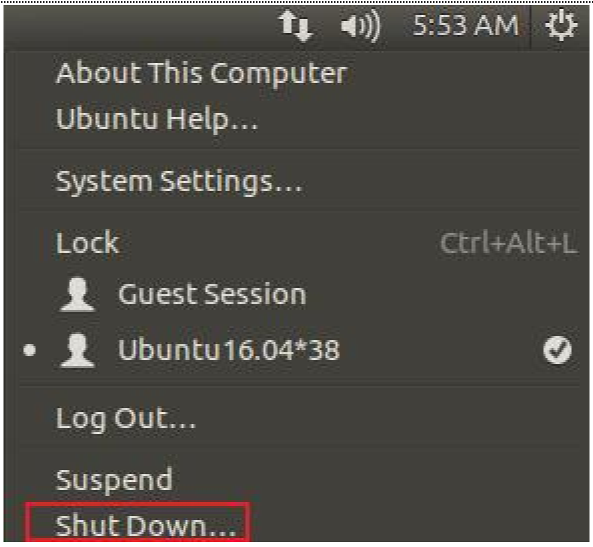
(15) When this interface appears, it indicates that the installation of Ubuntu is complete. (Note: For the two user login inputs on the login interface, the content within the red frame is user-defined, and the content within the green frame is provided by the system.)



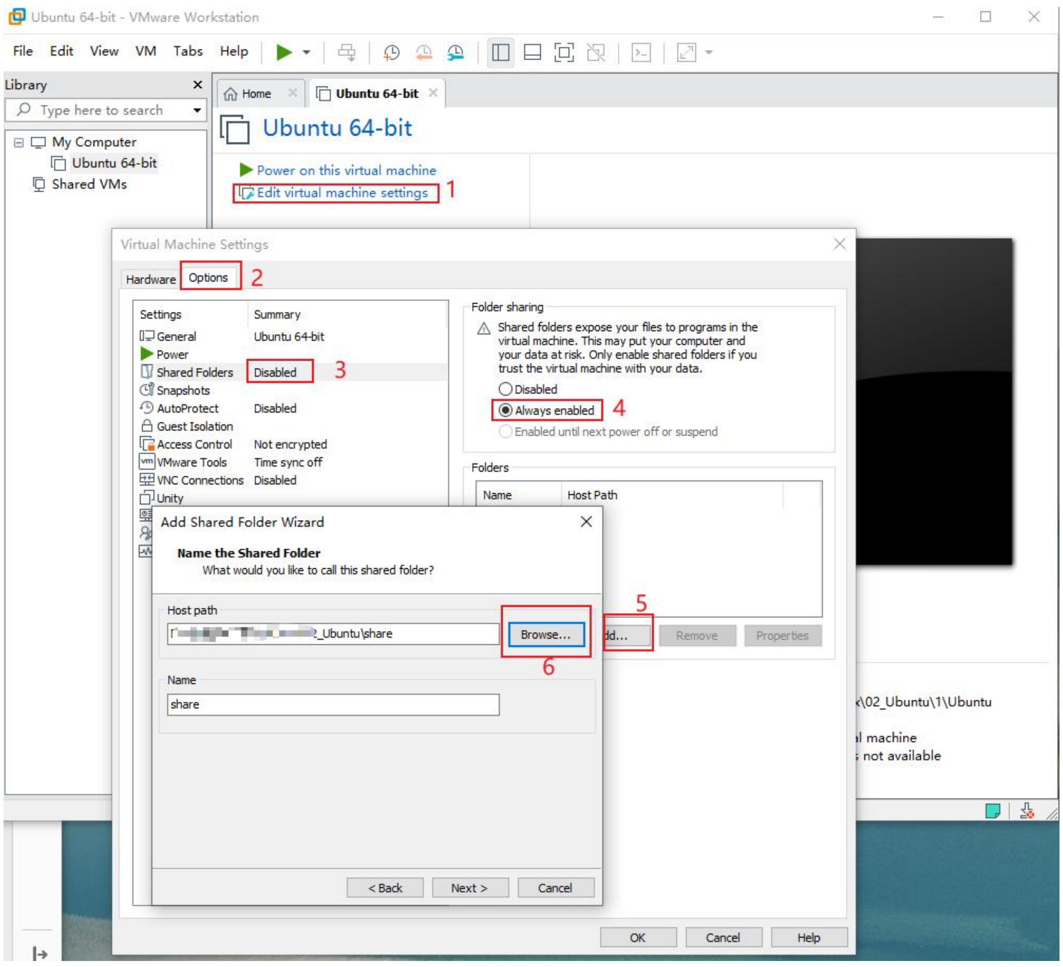
(16) Next, we will start to configure the required environment for Ubuntu.

2.1.6 Setting up the Shared Folder

(1) Shut down Ubuntu. Click "Shut Down" in the upper right corner of the desktop.



(2) After shutting down, on the corresponding virtual machine page, click "Edit virtual machine settings"→"Options"→"Shared Folders"→"Always enabled"→"Add". Add a folder as a medium for transferring files between the host machine and the virtual machine. Finally, click "OK".



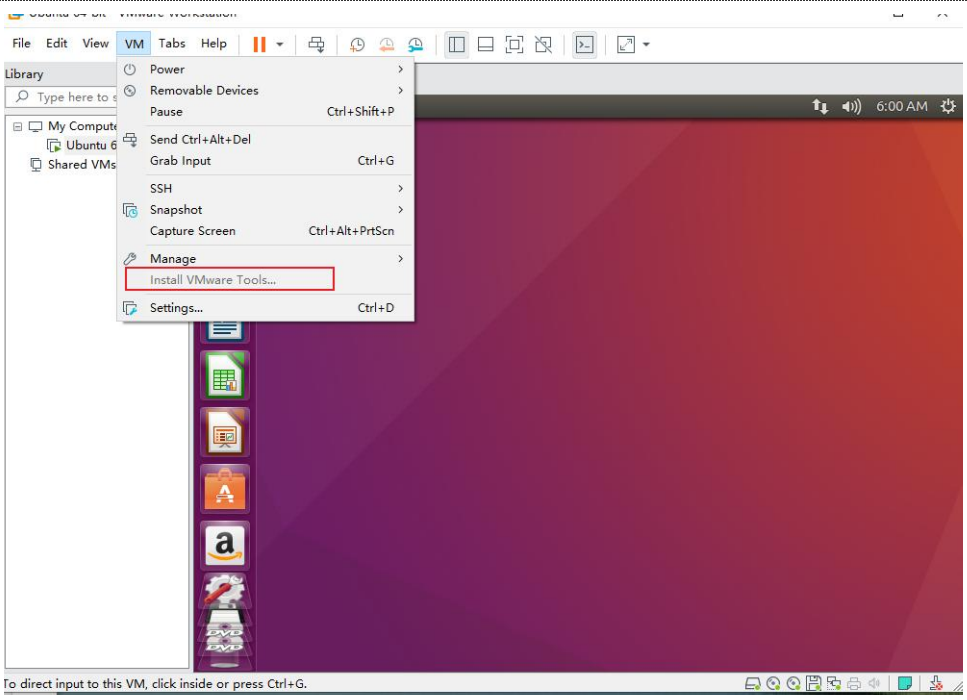
Settings	Summary
General	Ubuntu 64-bit
Power	
Shared Folders	Disabled
Snapshots	
AutoProtect	Disabled
Guest Isolation	
Access Control	Not encrypted
VMware Tools	Time sync off
VNC Connections	Disabled
Unity	

Folder sharing
Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.
Disabled [✓ Always enabled] Enabled until next power off or suspend

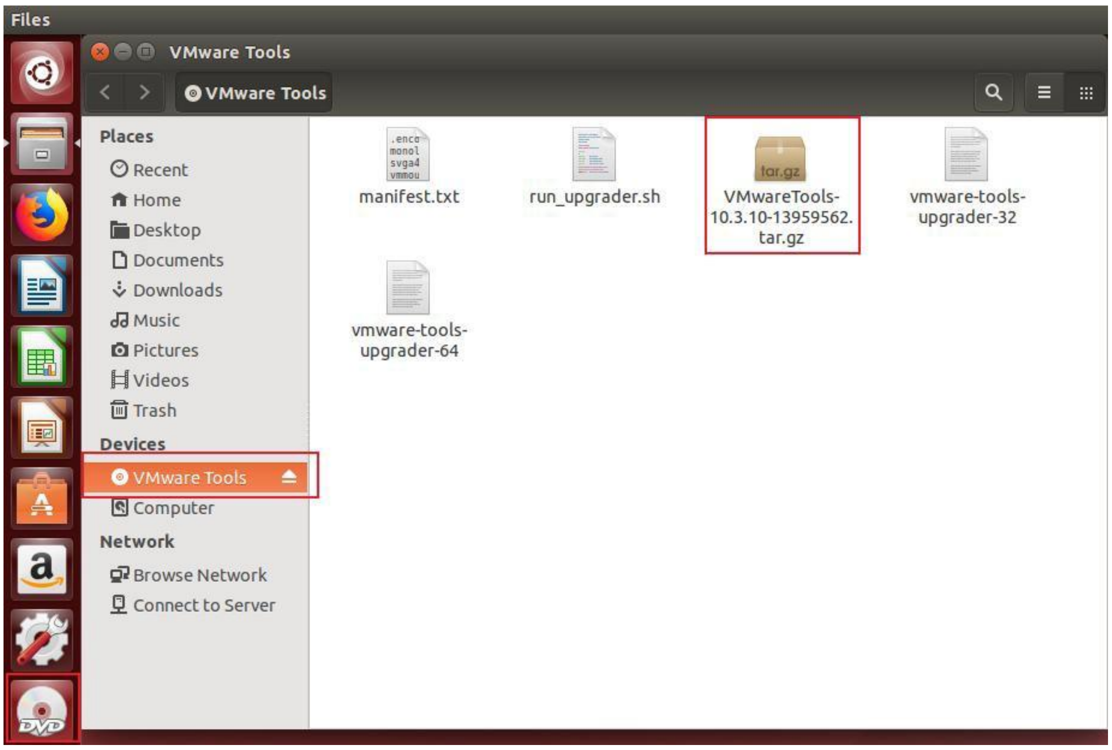
Folders
Name | Host Path
Add… Remove Properties

Name the Shared Folder
What would you like to call this shared folder?
Host path: [redacted]_Ubuntu\share Browse…
Name: share

(3) When starting up the virtual machine, click "Virtual Machine"→ "Install VMware Tools". (Note: You can only select "Install VMware Tools" when the virtual machine is starting up. Otherwise, this option will be grayed out and unavailable. Since the author has already installed VMware Tools, it will display "Reinstall VMware Tools" instead.)



(4) Click "DVD" icon and open to see a compressed file "VMwareTools-10.3.10-12406962.tar.gz".

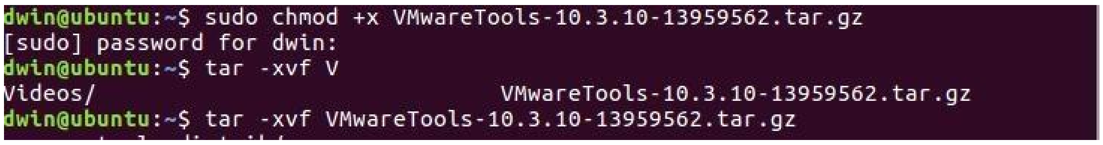


(5) Right-click and choose "Copy to" to a path with permissions, you can directly copy it to "home."

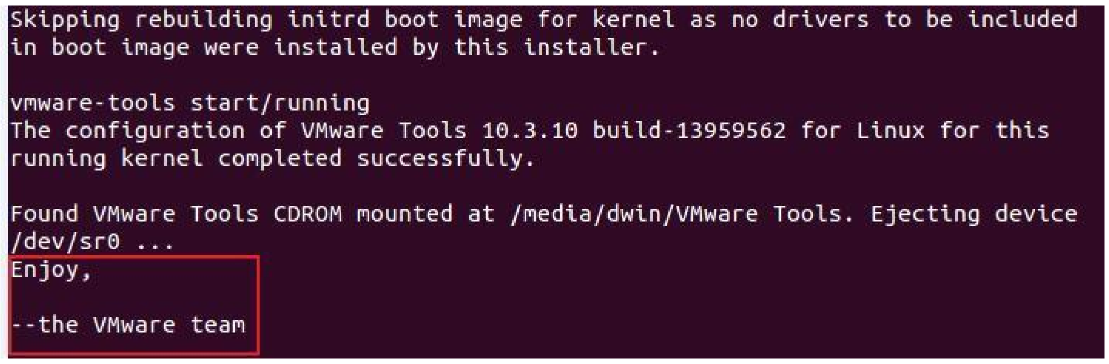
(6) At this point, open the terminal by pressing [Ctrl] + [Alt] + [T], which will open the terminal in the root directory

(7) Enter the command to add executable permissions: **sudo chmod +x VM** (use the Tab key to display the full name) (Enter) (Note: When using sudo privileges for the first time, you need to enter the password, and it won't be visible when entering the password).

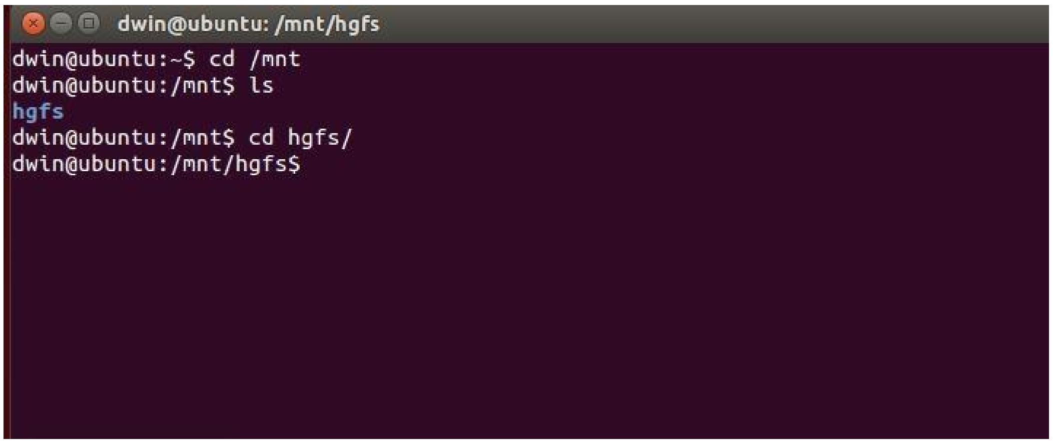
(8) Enter the decompression command: **tar -xvf VM** (Tab key) (Enter), and it will automatically decompress to generate "vmware-tools-distrib" in the current directory. Enter the command: **cd vm** (Tab key) (Press the Enter key. The subsequent steps will be omitted and not written).


```
dwin@ubuntu:~$ sudo chmod +x VMwareTools-10.3.10-13959562.tar.gz
[sudo] password for dwin:
dwin@ubuntu:~$ tar -xvf V
Videos/                              VMwareTools-10.3.10-13959562.tar.gz
dwin@ubuntu:~$ tar -xvf VMwareTools-10.3.10-13959562.tar.gz
```

(9) Enter the run command: **sudo ./vm** (Tab), and the installation will begin. When [yes] or [no] appears, just enter "y" and press Enter key (the default for enabling shared folders is no, for ease of operation, all configurations are selected with "y"). Press Enter key for the remaining cases until it shows as shown in the image, indicating that the installation is complete.


```
Skipping rebuilding initrd boot image for kernel as no drivers to be included
in boot image were installed by this installer.

vmware-tools start/running
The configuration of VMware Tools 10.3.10 build-13959562 for Linux for this
running kernel completed successfully.

Found VMware Tools CDROM mounted at /media/dwin/VMware Tools. Ejecting device
/dev/sr0 ...
Enjoy,

--the VMware team
```

(10) Enter the command: **cd /mnt** (Keep pressing the Tab key until you reach the shared folder you have set up.), the path is /mnt/hfgs/***. The shared folder is now set up, and you can proceed to install the T113 toolchain on Ubuntu.


```
dwin@ubuntu: /mnt/hgfs
dwin@ubuntu:~$ cd /mnt
dwin@ubuntu:/mnt$ ls
hgfs
dwin@ubuntu:/mnt$ cd hgfs/
dwin@ubuntu:/mnt/hgfs$
```

2.2 Installing the T113 Toolchain

(1) Move the T113 compressed package (buildroot-T113-QT5_12_5-sdk-soft20221012.tar.gz) to Ubuntu, you can use a shared folder or transfer via SFTP, etc.

buildroot-T113-QT5_12_5-sdk-soft20221012.tar.gz

(2) Move the file to the root directory (/home/dwin). When using a shared folder, enter the command: **sudo mv buil (Tab)~**, after waiting for a while, it moves to the root directory.

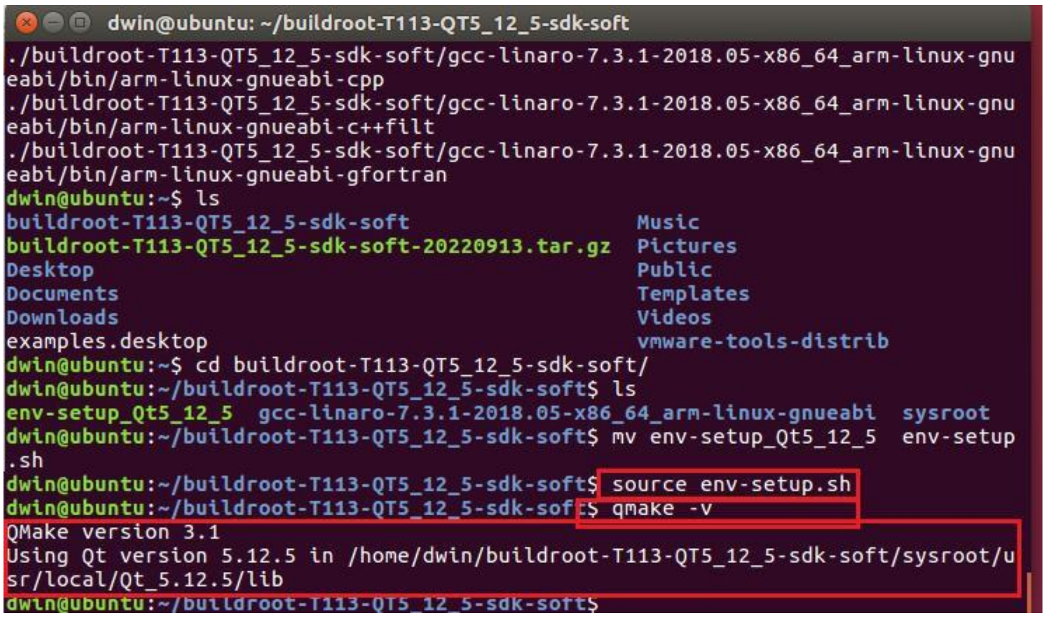
(3) Enter the command: **tar -xvf bu(TAB)** to unzip the file.

(4) Enter the following commands one by one:

cd bui(TAB)

source env-setup.sh

Then, enter "**qmake -v**" to check the version information of qmake and check if the setup is successful.



```
./buildroot-T113-QT5_12_5-sdk-soft/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnu
eabi/bin/arm-linux-gnueabi-cpp
./buildroot-T113-QT5_12_5-sdk-soft/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnu
eabi/bin/arm-linux-gnueabi-c++filt
./buildroot-T113-QT5_12_5-sdk-soft/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnu
eabi/bin/arm-linux-gnueabi-gfortran
dwin@ubuntu:~$ ls
buildroot-T113-QT5_12_5-sdk-soft                    Music
buildroot-T113-QT5_12_5-sdk-soft-20220913.tar.gz    Pictures
Desktop                                             Public
Documents                                           Templates
Downloads                                           Videos
examples.desktop                                    vmware-tools-distrib
dwin@ubuntu:~$ cd buildroot-T113-QT5_12_5-sdk-soft/
dwin@ubuntu:~/buildroot-T113-QT5_12_5-sdk-soft$ ls
env-setup_Qt5_12_5  gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi  sysroot
dwin@ubuntu:~/buildroot-T113-QT5_12_5-sdk-soft$ mv env-setup_Qt5_12_5  env-setup
.sh
dwin@ubuntu:~/buildroot-T113-QT5_12_5-sdk-soft$ source env-setup.sh
dwin@ubuntu:~/buildroot-T113-QT5_12_5-sdk-soft$ qmake -v
QMake version 3.1
Using Qt version 5.12.5 in /home/dwin/buildroot-T113-QT5_12_5-sdk-soft/sysroot/u
sr/local/Qt_5.12.5/lib
dwin@ubuntu:~/buildroot-T113-QT5_12_5-sdk-soft$
```

2.3 Screen Configuration

2.3.1 Hardware Introduction

Please refer to the related datasheet.

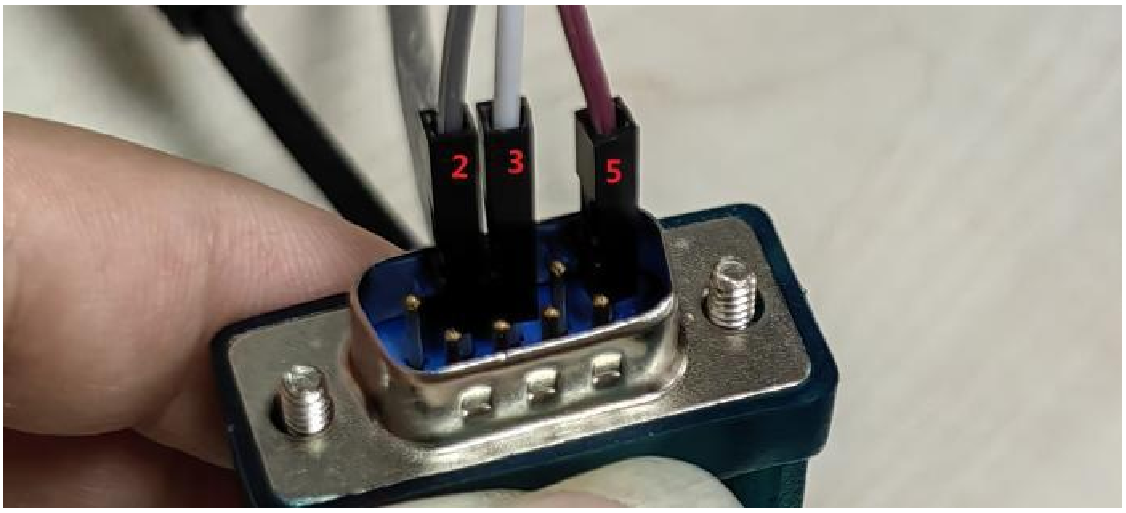
2.3.2 Terminal Software

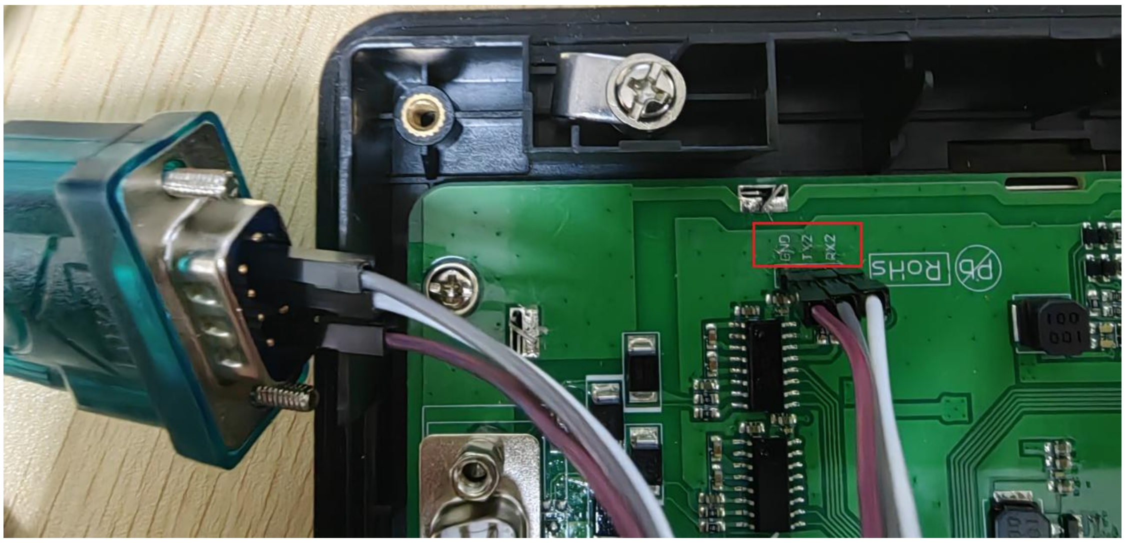
SecureCRT or MobaXterm are optional, we will introduce MobaXterm in this section.

Two types of connection: serial or Telnet.

2.3.3 Serial communication

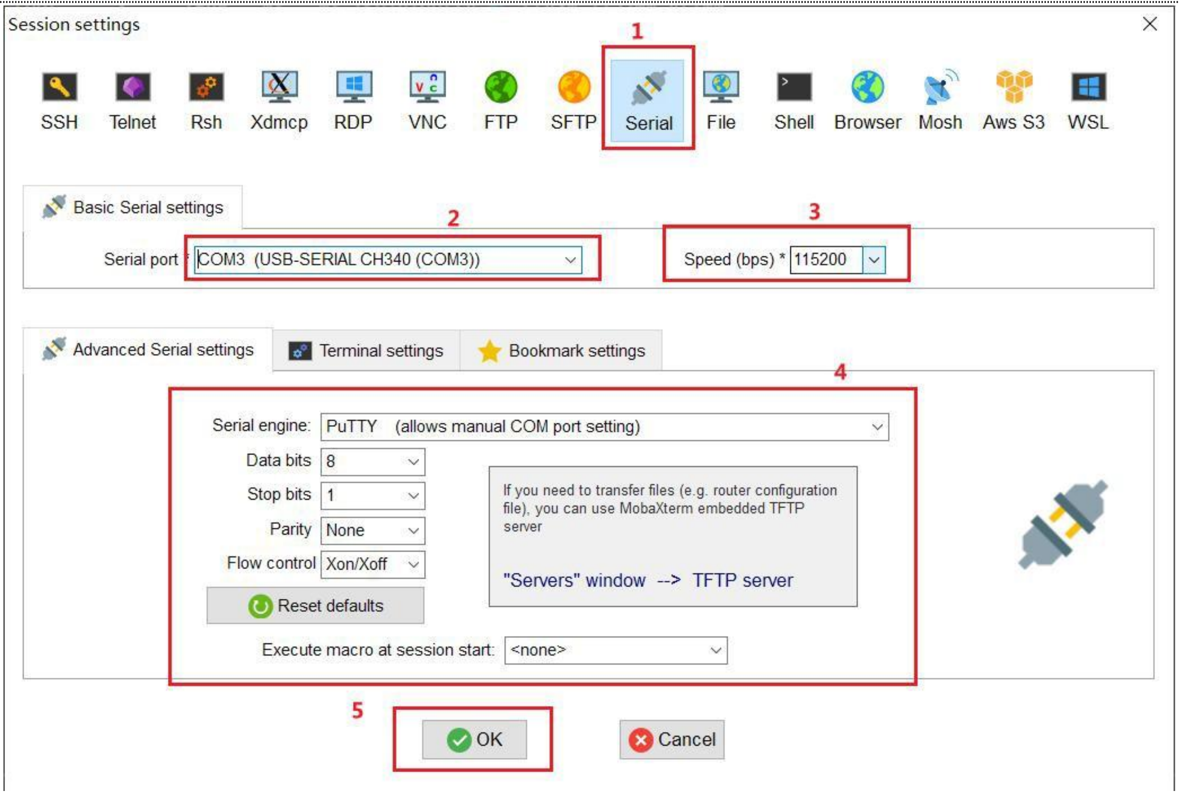
(1) Connect to Serial 2 (RX2/TX2), RS232 connection in this example.





Connect 2(RX) to TX2, 3(TX) to RX2, and 5 to GND.

(2) MobaXterm Configuration: Sessions→New session→Choose 'serial'→Choose the serial port, set the baud rate in the third step, and cross-check the information→click 'OK' to complete.



Serial port: COM3 (USB-SERIAL CH340 (COM3))
Speed (bps) * 115200

Serial engine: PuTTY (allows manual COM port setting)
Data bits: 8
Stop bits: 1
Parity: None
Flow control: Xon/Xoff

If you need to transfer files (e.g. router configuration file), you can use MobaXterm embedded TFTP server

"Servers" window --> TFTP server

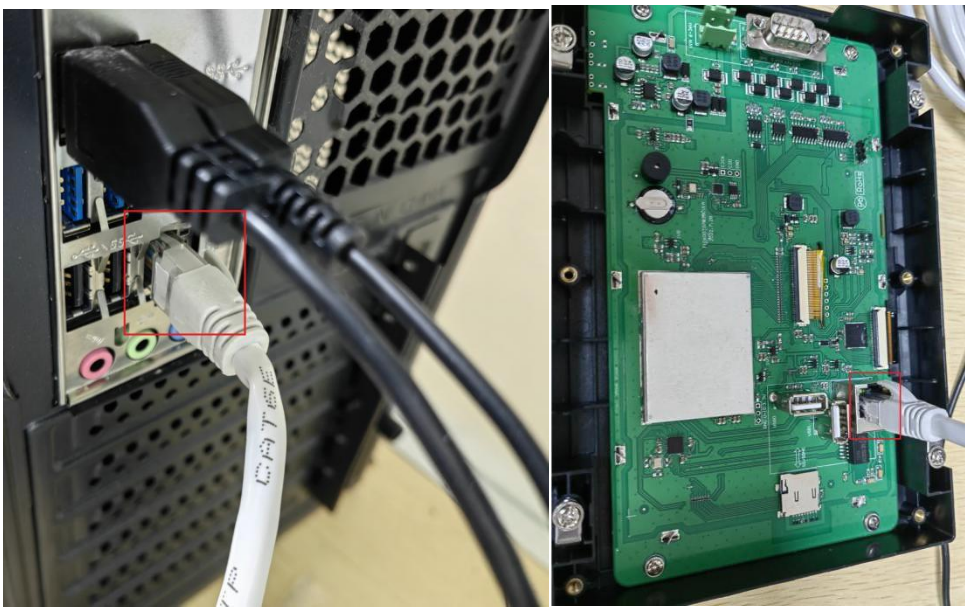
Execute macro at session start: <none>

(3) At this time, power on the screen, and enter the username "root" and the password "Dwin123", then you can start the operation.

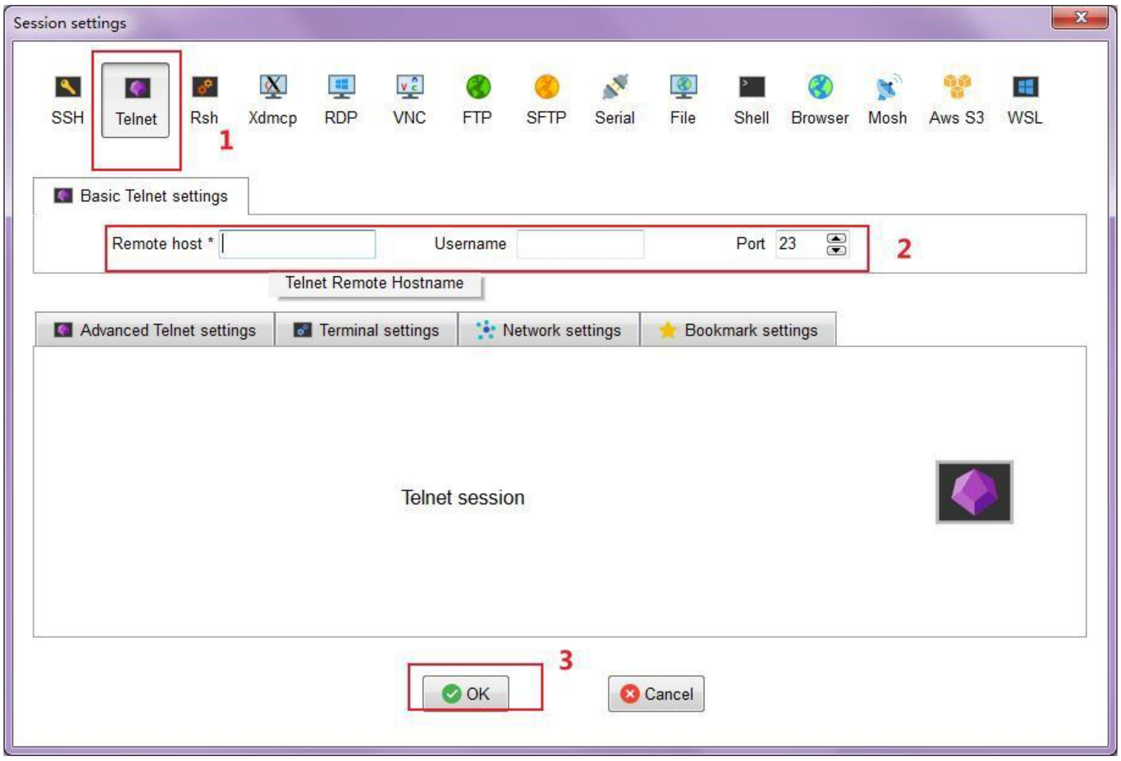
2.3.4 Telnet Connection via Ethernet

Note: Please ensure that the computer and the device are in the same network segment (the default IP address of the screen device is 192.168.10.202). Otherwise, please use a network cable to connect the computer to the device, set the computer's IP address as a static IP 192.168.10.xxx (xxx is not 202) and try to connect to the device. After a successful connection, if the device needs to be connected to the local area network, please refer to Chapter 2.3.5 to modify the device's IP address by yourself, and then use two network cables to connect the device and the computer respectively. The following operations assume that the device and the computer are in the same network segment. In the example, the computer's IP address is 192.168.10.14 and the device's IP address is 192.168.10.202.

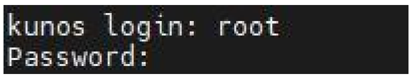
(1) Insert the network cable into the network ports of both the computer and the screen.



(2) As shown in the figure, in the first step, click "Sessions" and select "New session". In the second step, select "Telnet". In the third step, enter the IP address of the screen and click "OK". (Note: The default IP address of the screen is **192.168.10.202**, and communication can only be achieved when connected within the same local area network.)



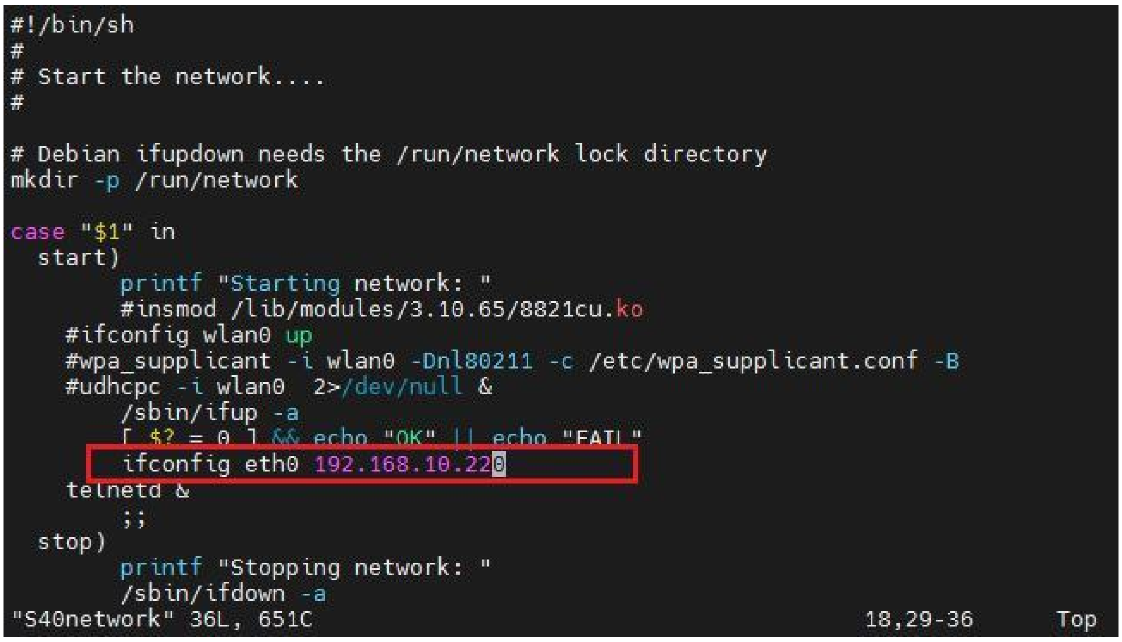
(3) At this time, power on the screen. The following interface will be displayed. Enter the account "root" and the password "Dwin123", and then you can start the operation. (The password input on the device side is not visible.)



```
kunos login: root
Password:
```

2.3.5 Screen IP Configuration

If you need to modify the IP address, after connecting to the device as described in the previous section, you can enter: **vi /etc/init.d/S40 (TAB)**. Move the cursor to the "ifconfig" line and press the "i" key to start editing. After modifying the IP address, press the "Esc" key, then enter: (colon) wq (press the Enter key), which means saving the changes and exiting, and thus the modification of the IP address is completed.



```
#!/bin/sh
#
# Start the network....
#

# Debian ifupdown needs the /run/network lock directory
mkdir -p /run/network

case "$1" in
  start)
        printf "Starting network: "
        #insmod /lib/modules/3.10.65/8821cu.ko
    #ifconfig wlan0 up
    #wpa_supplicant -i wlan0 -Dnl80211 -c /etc/wpa_supplicant.conf -B
    #udhcpc -i wlan0  2>/dev/null &
        /sbin/ifup -a
        [ $? = 0 ] && echo "OK" || echo "FAIL"
        ifconfig eth0 192.168.10.220
    telnetd &
        ;;
  stop)
        printf "Stopping network: "
        /sbin/ifdown -a
"S40network" 36L, 651C                                      18,29-36      Top
```

2.3.6 Application Upgrade Guide

2.3.6.1 Principle of the Application Upgrade Package

(1) In the running environment of the standard screen, there exists a file named /etc/emcversion, in which the current version number is stored.

(2) Only when the file name of the upgrade package is consistent with the version number can the upgrade be carried out. Generally, the version number should be modified in the upgrade file to avoid repeated upgrades.

(3) The naming rule of the upgrade package is "version number.tar". The initial version number of the 36 series is: A01-1-0.

(4) During the power-on startup process of the standard screen, it will actively detect the USB flash drive and search for the "update" subdirectory in its root directory. If there is an upgrade package, it will

automatically unzip it and execute the "install.sh" script in the upgrade package.

(5) After the "install.sh" script gains control, it can copy files, modify file attributes, and complete the upgrade function.

2.3.6.2 Application Upgrade Package Production

(1) In the Ubuntu environment, centrally store the files to be upgraded in a unified directory.

(2) Add an **install.sh** file into the directory, modifying the script for file copying and attribute changes.

(3) In the Ubuntu environment, pack this directory using the command: tar -cvf DWIN_V1.X.X.tar <INSTALL>

(4) Copy the file (e.g., DWIN_V1.X.X.tar) to the USBflash drive /update directory.

2.3.6.3 Usage of the Application Upgrade Package

(1) The standard program is burned into the standard screen, and the screen can be normally lit up.

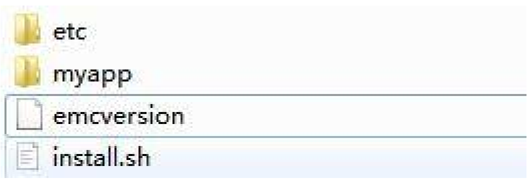
(2) According to the needs, a specific application upgrade package can be selected, and this package should be copied to the /update directory of the USB flash drive.

(3) Before powering on, insert the USB flash drive into the standard screen.

(4) After powering on, wait for the standard screen to automatically shut down, which indicates that the upgrade is successful.

2.3.6.4 Example of Upgrade Package (Modifying the Boot-up Running Program)

The composition of the folder before compression and packaging is as follows:



emcversion file stores the updated version number for device updates.

myapp folder contains files to be upgraded.

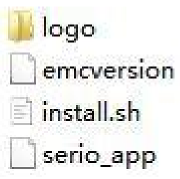
etc folder stores scripts in /etc/init.d/ that may need to be modified (it can be excluded if there are none).

The example files of **Install.sh** is as follows:

```
#!/bin/sh


copy_dir()
{
  if [ -d $1 ]; then
     for libfile in $1/*; do
         if [ -f $libfile ]; then
            cp $libfile $2/
            chmod $3 $2/${libfile##*/}
           #echo $2/${libfile##*/}
         fi
      done
  fi
}
instdir=$(cd `dirname $0`; pwd)
# update the emcversion
cp $instdir/emcversion /etc/
# copy application file
cp $instdir/myapp/myapp /usr/local/bin/myapp
# modify permission
chmod 755 /usr/local/bin/myapp
# modify runqt script file if needed
cp -a $instdir/etc/init.d/* /etc/init.d/
```

2.3.6.5 Example of Upgrade Package (Modifying the Boot-up LOGO)

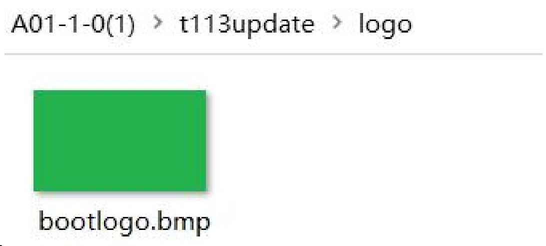
(1) The composition of the A01-1-0.tar folder before compression and packaging is as follows:



```
logo
emcversion
install.sh
serio_app
```

emcversion file stores the updated version number for device updates.

The replacement logo pictures are stored in the "logo" folder. The picture must be named "bootlogo.bmp", and the logo file must be a 24-bit BMP format image. Meanwhile, the location where the logo picture should be placed is: logoupdate/logo/bootlogo.bmp.



Example install.sh file:

```
#!/bin/sh


copy_dir()
{
  if [ -d $1 ]; then
     for libfile in $1/*; do
         if [ -f $libfile ]; then
            cp $libfile $2/
            chmod $3 $2/${libfile##*/}
           #echo $2/${libfile##*/}
         fi
      done
  fi
}
instdir=$(cd `dirname $0`; pwd)
# update the emcversion
cp $instdir/emcversion /etc/
# copy logo file
if [ -f $instdir/logo/bootlogo.bmp ]; then
   mkdir -p /extp/temp0p2
   mount /dev/mmcblk0p2 /extp/temp0p2
   cp -a $instdir/logo/bootlogo.bmp /extp/temp0p2/
   umount /extp/temp0p2
```

```
        rm -r /extp/temp0p2
        sync
    fi
    sync

    $instdir/serio_app
```

(2) Compress the "logoupdate" directory into a tar file and name it "A01-1-0.tar". Copy this tar file to the "update" directory of the USB drive. Then, power off the device, insert the USB drive, and power it on again. Once the upgrade is successful, you will hear a "beep" sound. After that, the screen will turn off. Then, remove the USB drive, power off the device and restart it, and check whether the boot logo is correct.

(3) Precautions:

1. After testing, when unzipped in the Windows environment, only replace the "bootlogo.bmp" file, and then repackage it. The update can be successful in this way. If the upgrade fails, it may be due to the lack of execution permission. Please go to the Linux environment to confirm whether the "install.sh" file has the execution permission.

2. The command to compress a tar file in Linux: tar -cvf A01-1-0.tar logoupdate

3. Don't put too many files in the USB drive (it is recommended to use a dedicated USB drive that only contains the upgrade file). Otherwise, it may cause the update to fail.

4. This upgrade package can be obtained from the sales staff.

3 Cross-Compilation of QT Project Files

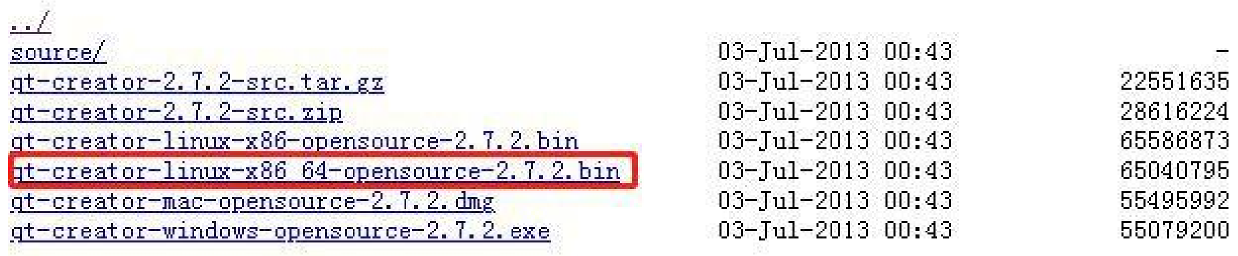
3.1 Configuration of Cross-Compilation of Qt Creator

3.1.1 System Requirements

This document has been verified based on the Ubuntu 14.04 system, and other versions of the Ubuntu system have not been verified.

3.1.2 Download the Installation Package of Qt Creator

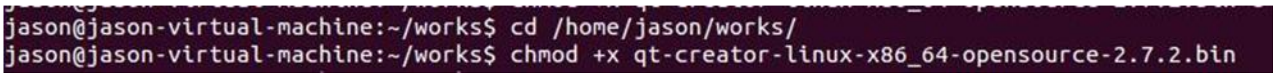
The version of QtCreator used in this document is 2.7.2. Please download the version that is compatible with your operating system.



```
../
source/                                          03-Jul-2013 00:43                   -
qt-creator-2.7.2-src.tar.gz                      03-Jul-2013 00:43            22551635
qt-creator-2.7.2-src.zip                         03-Jul-2013 00:43            28616224
qt-creator-linux-x86-opensource-2.7.2.bin        03-Jul-2013 00:43            65586873
qt-creator-linux-x86_64-opensource-2.7.2.bin     03-Jul-2013 00:43            65040795
qt-creator-mac-opensource-2.7.2.dmg              03-Jul-2013 00:43            55495992
qt-creator-windows-opensource-2.7.2.exe          03-Jul-2013 00:43            55079200
```

3.1.3 Install Qt Creator

Copy the installation package to the Ubuntu system and grant the file execution permission:



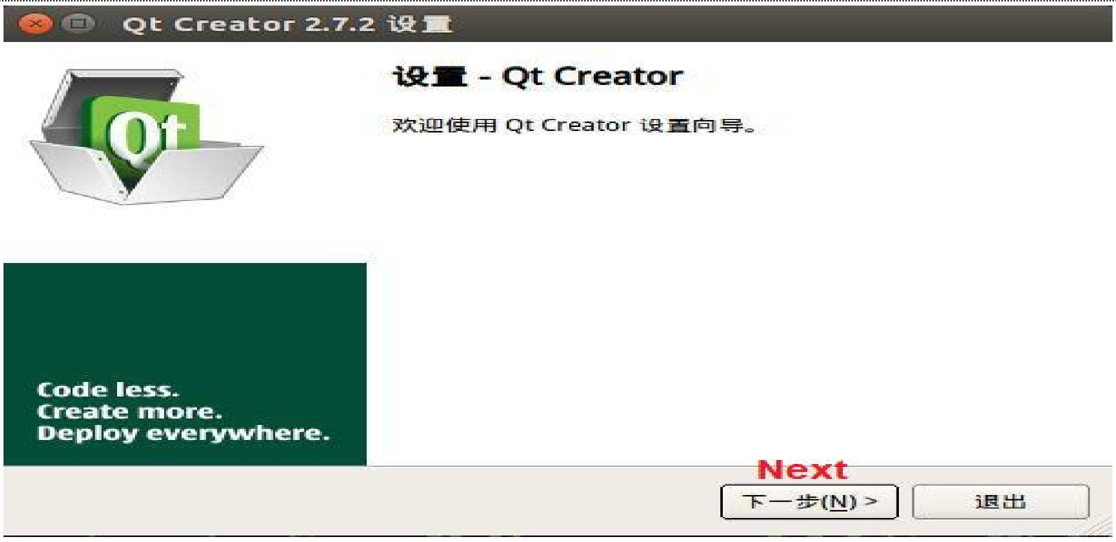
```
jason@jason-virtual-machine:~/works$ cd /home/jason/works/
jason@jason-virtual-machine:~/works$ chmod +x qt-creator-linux-x86_64-opensource-2.7.2.bin
```

# chmod +x qt-creator-linux-x86_64-opensource-2.7.2.bin

3.1.4 Run the Installation Package

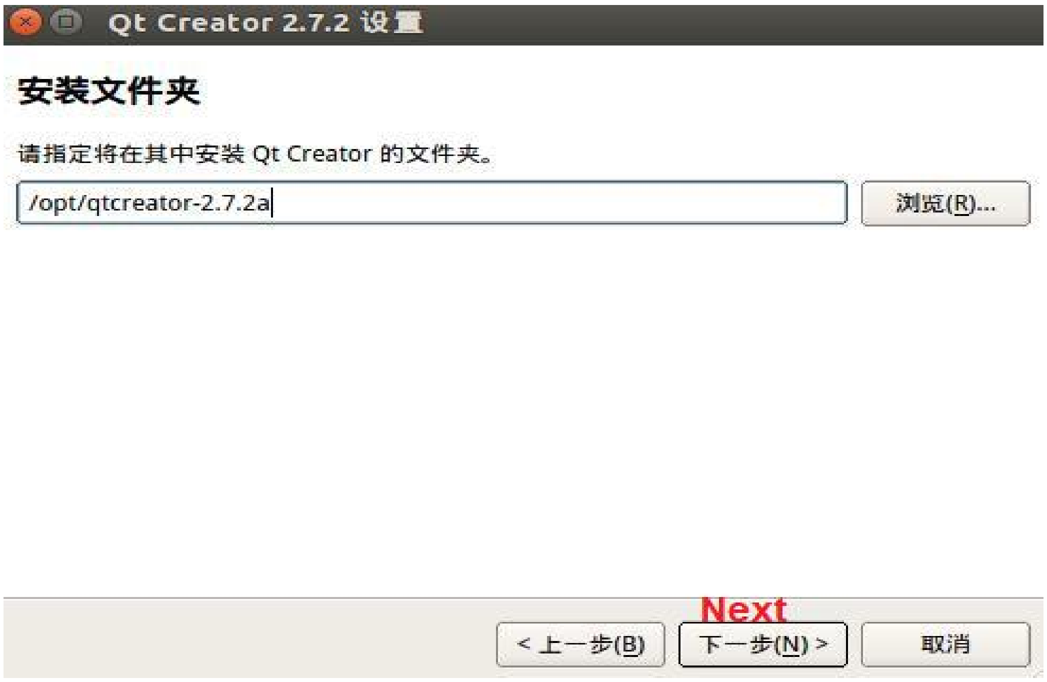
```
# sudo
./qt-creator-linux-x86_64-opensource-2.7.2
.bin
```

Just click "Next" directly.



设置 - Qt Creator

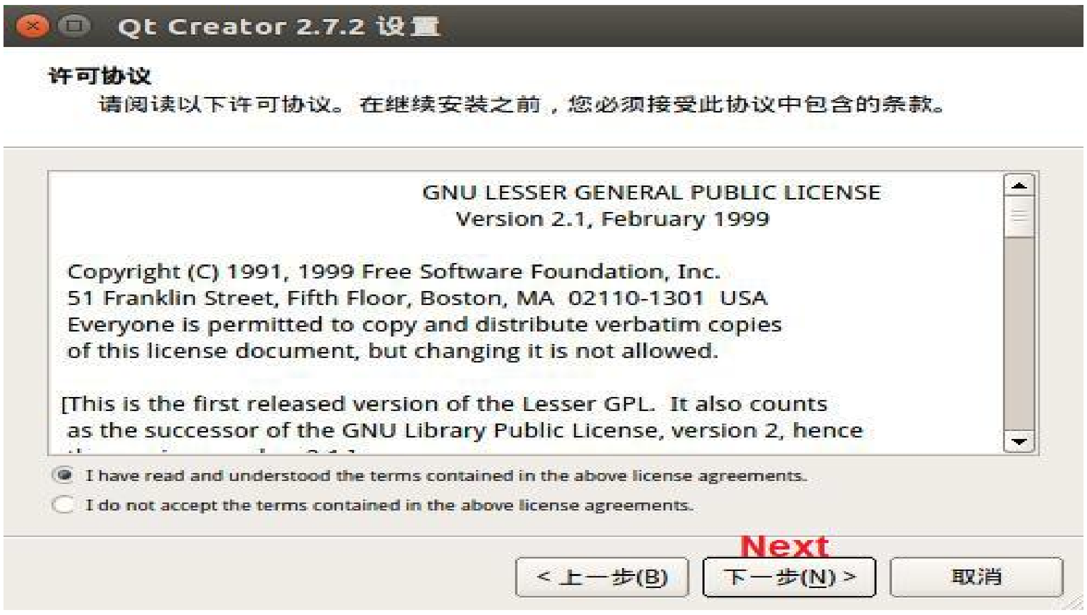
欢迎使用 Qt Creator 设置向导。



安装文件夹

请指定将在其中安装 Qt Creator 的文件夹。

/opt/qtcreator-2.7.2a



许可协议

请阅读以下许可协议。在继续安装之前，您必须接受此协议中包含的条款。

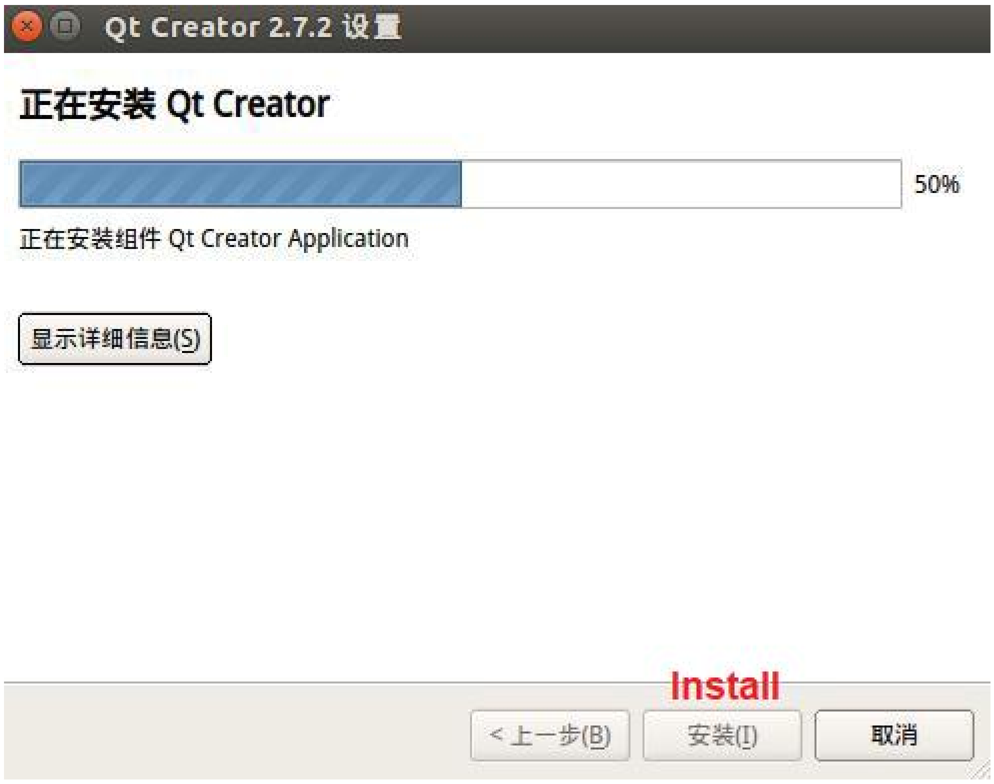
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence

[✓ I have read and understood the terms contained in the above license agreements.]

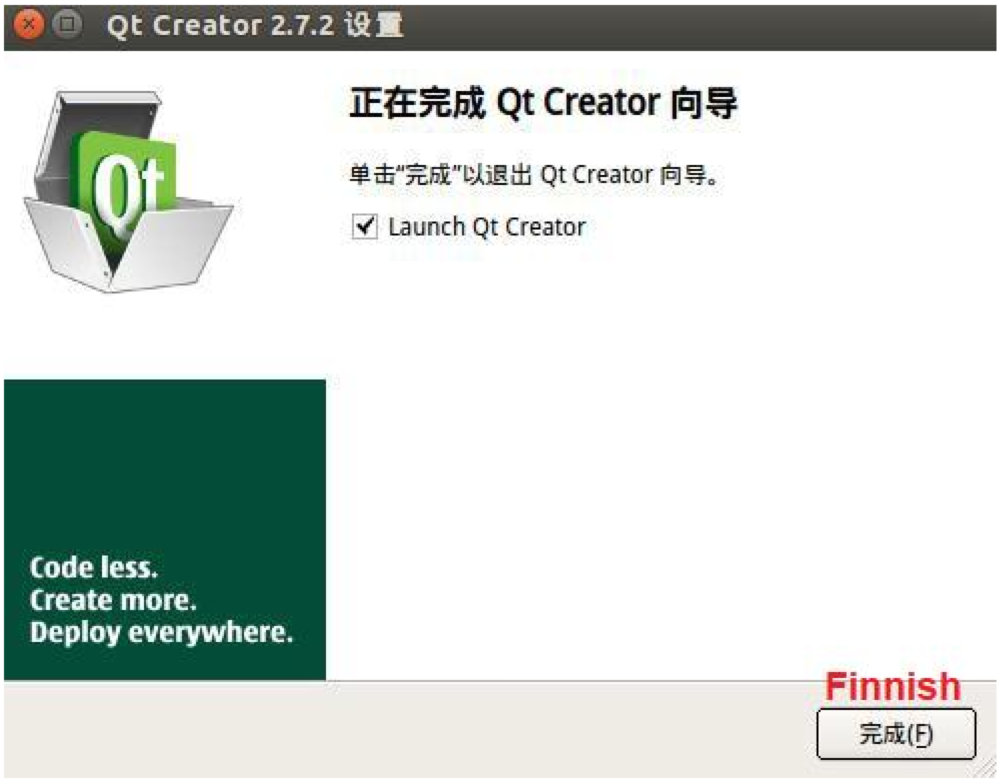
I do not accept the terms contained in the above license agreements.


Qt Creator 2.7.2 设置

正在安装 Qt Creator

50%

正在安装组件 Qt Creator Application


Qt Creator 2.7.2 设置

正在完成 Qt Creator 向导

单击"完成"以退出 Qt Creator 向导。

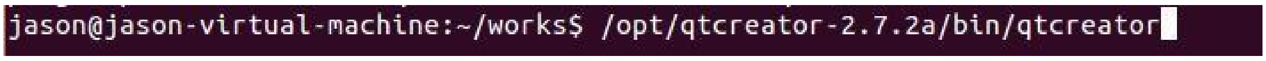
☑ Launch Qt Creator

3.1.5 Configure the Cross-compilation Environment

3.1.5.1 Run the QtCreator

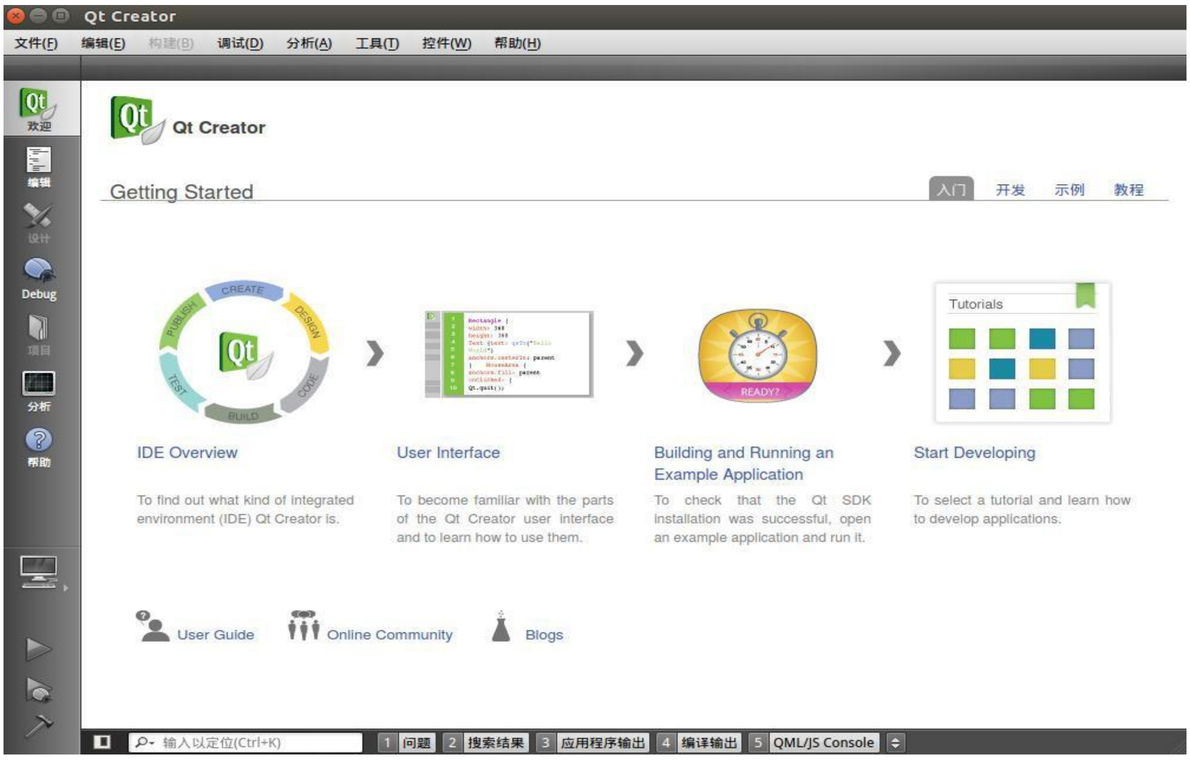
The executable program of QtCreator is located in the "bin" directory under the installation directory:

# /opt/qtcreator-2.7.2a/bin/qtcreator



```
jason@jason-virtual-machine:~/works$ /opt/qtcreator-2.7.2a/bin/qtcreator
```

The software interface is as follows:


Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) 分析(A) 工具(T) 控件(W) 帮助(H)
欢迎
编辑
设计
Debug
项目
分析
帮助
Qt Creator
Getting Started
入门 开发 示例 教程
CREATE DESIGN CODE BUILD TEST PUBLISH
READY?
Tutorials
User Guide
Online Community
Blogs
输入以定位(Ctrl+K)
1 问题 2 搜索结果 3 应用程序输出 4 编译输出 5 QML/JS Console

IDE Overview

To find out what kind of integrated environment (IDE) Qt Creator is.

User Interface

To become familiar with the parts of the Qt Creator user interface and to learn how to use them.

Building and Running an Example Application

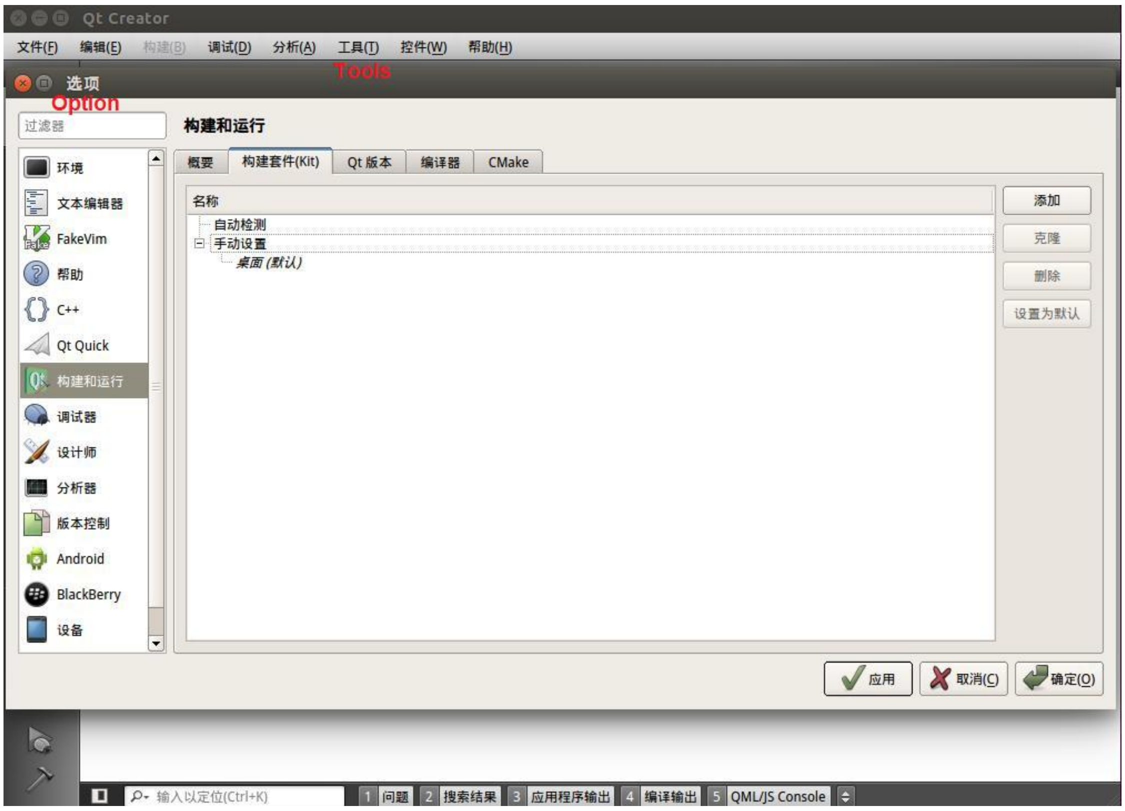
To check that the Qt SDK installation was successful, open an example application and run it.

Start Developing

To select a tutorial and learn how to develop applications.

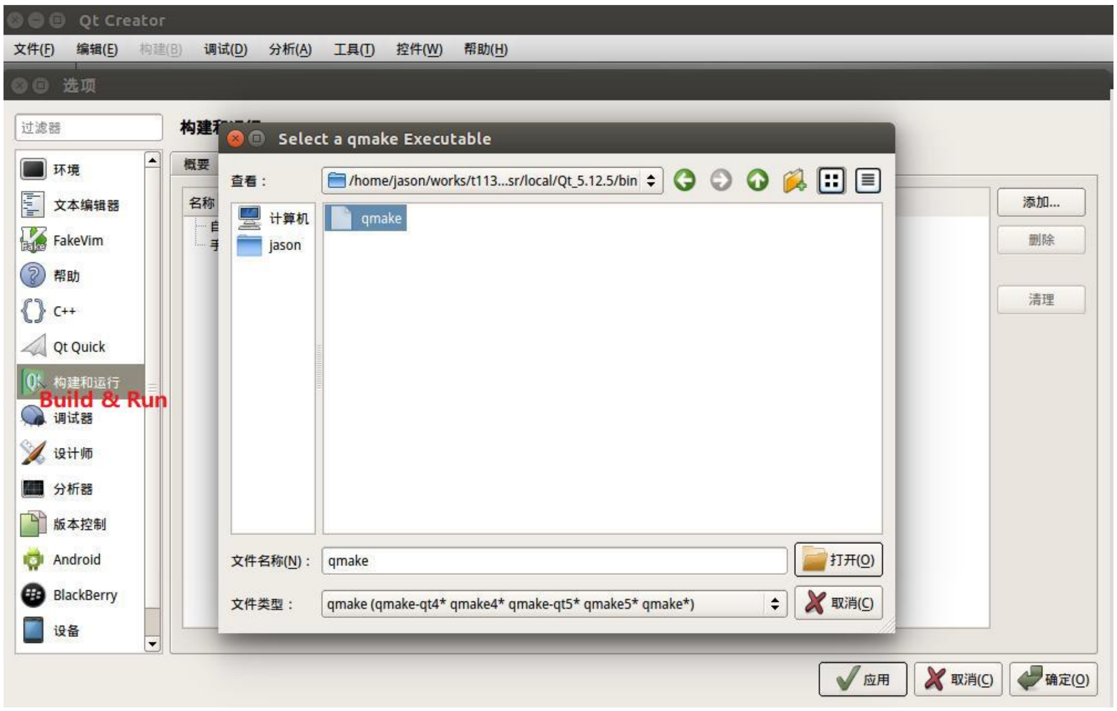
3.1.5.2 Configure the Cross-Compilation Environment

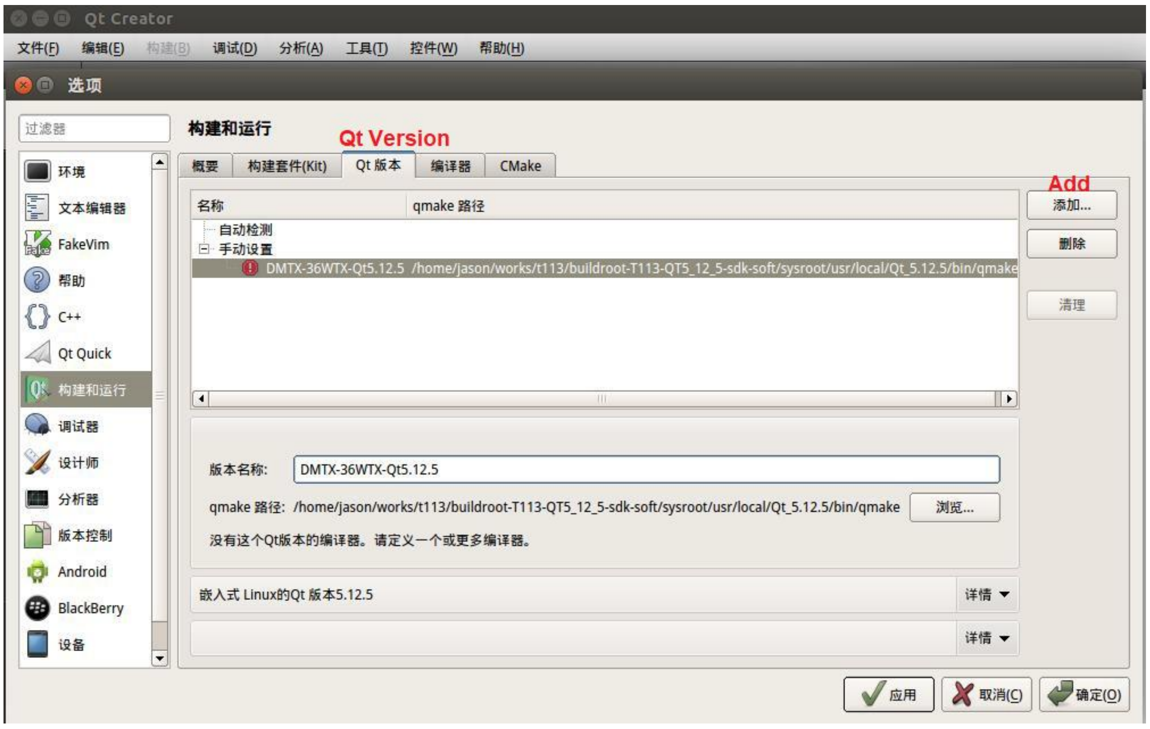
Select the "Tools"→Option


Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) 分析(A) 工具(T) 控件(W) 帮助(H)
Tools
选项
Option
过滤器
构建和运行
概要 构建套件(Kit) Qt 版本 编译器 CMake
名称
自动检测
手动设置
桌面 (默认)
添加 克隆 删除 设置为默认
环境 文本编辑器 FakeVim 帮助 C++ Qt Quick 构建和运行 调试器 设计师 分析器 版本控制 Android BlackBerry 设备
应用 取消(C) 确定(O)
输入以定位(Ctrl+K)
1 问题 2 搜索结果 3 应用程序输出 4 编译输出 5 QML/JS Console

Set qmake: Select "Build & Run"→"Qt Version"→"Add":

Qmake is located in the directory sysroot/usr/local/Qt_5.12.5/bin/ of the buildroot - T113 - QT5_12_5 - sdk - soft20221012.tar.gz package.


Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) 分析(A) 工具(T) 控件(W) 帮助(H)
选项
过滤器
Select a qmake Executable
查看：
/home/jason/works/t113...sr/local/Qt_5.12.5/bin
计算机
jason
qmake
文件名称(N)：qmake
打开(O)
文件类型：qmake (qmake-qt4* qmake4* qmake-qt5* qmake5* qmake*)
取消(C)
添加... 删除 清理
环境 文本编辑器 FakeVim 帮助 C++ Qt Quick 构建和运行 调试器 设计师 分析器 版本控制 Android BlackBerry 设备
Build & Run
应用 取消(C) 确定(O)



Qt Creator

选项

构建和运行

概要 | 构建套件(Kit) | Qt 版本 | 编译器 | CMake

名称	qmake 路径
自动检测	
手动设置	
DMTX-36WTX-Qt5.12.5	/home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot/usr/local/Qt_5.12.5/bin/qmake

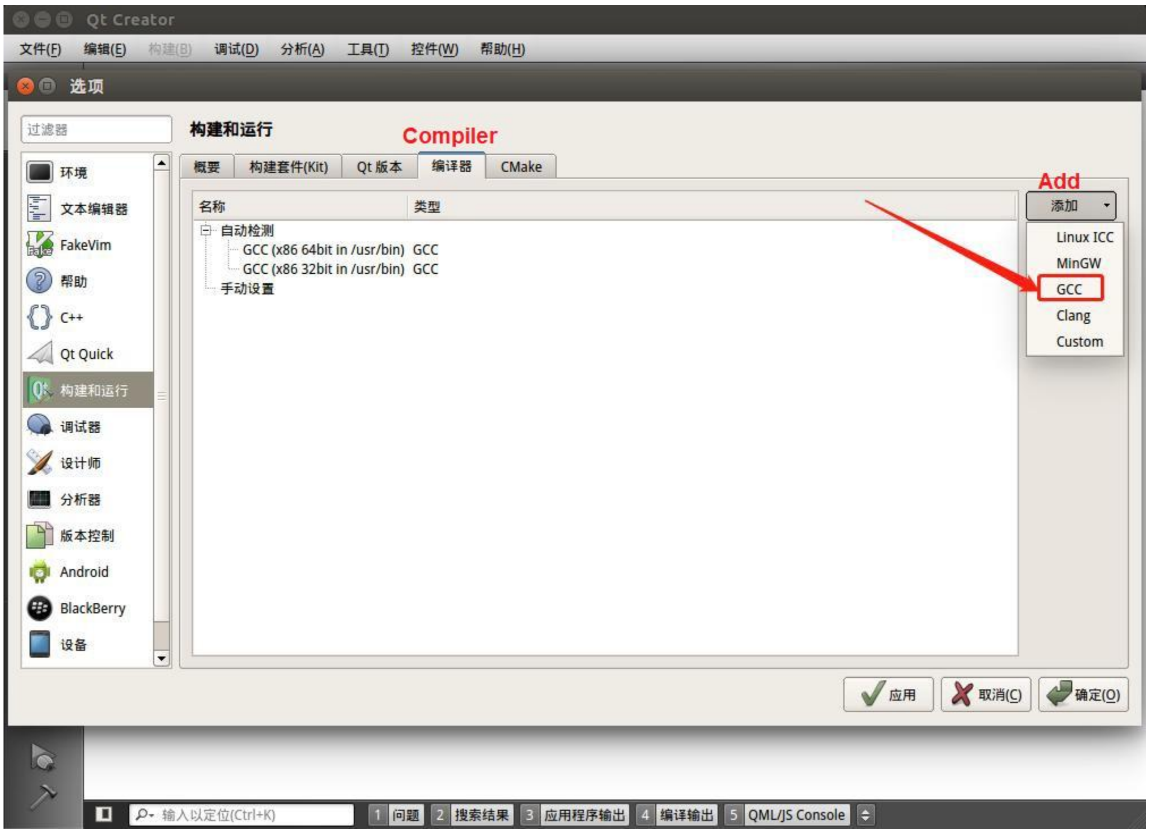
版本名称: DMTX-36WTX-Qt5.12.5

qmake 路径: /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot/usr/local/Qt_5.12.5/bin/qmake

没有这个Qt版本的编译器。请定义一个或更多编译器。

嵌入式 Linux的Qt 版本5.12.5

Set the compilation toolchain: Select "Build & Run"→"Compiler"→"Add"



Qt Creator

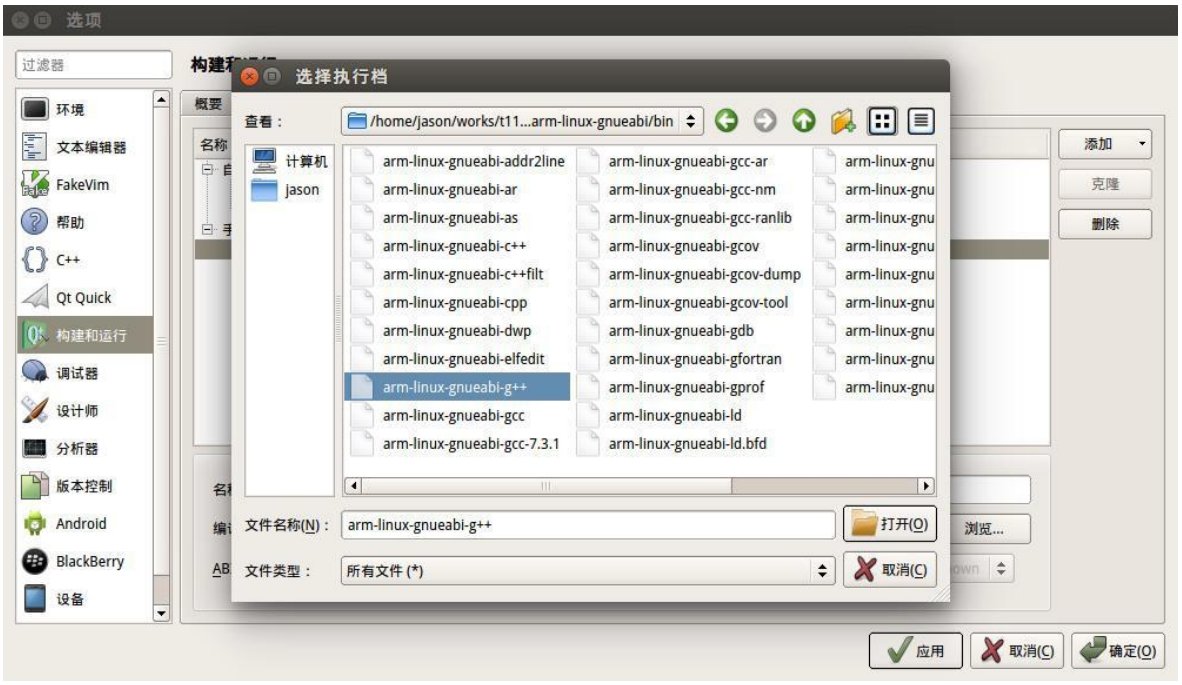
选项

构建和运行

概要 | 构建套件(Kit) | Qt 版本 | 编译器 | CMake

名称	类型
自动检测	
GCC (x86 64bit in /usr/bin)	GCC
GCC (x86 32bit in /usr/bin)	GCC
手动设置	

The compiler is located in the directory gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/bin of the buildroot-T113-QT5_12_5-sdk-soft20221012.tar.gz package.

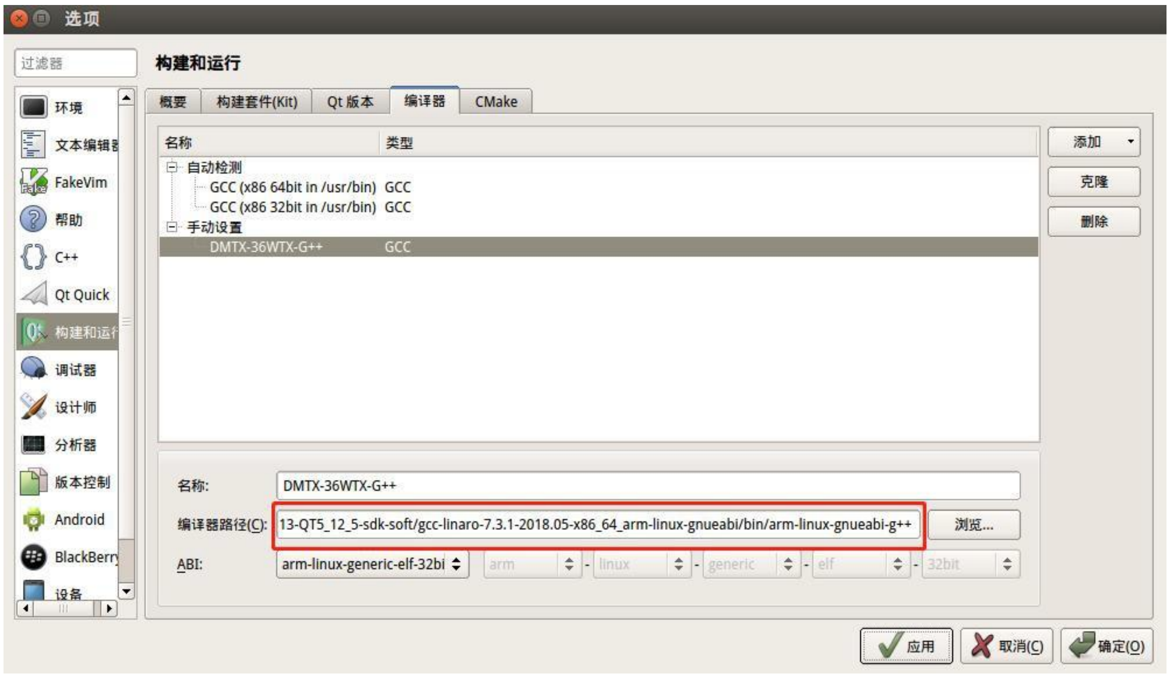


查看： /home/jason/works/t11...arm-linux-gnueabi/bin

```
arm-linux-gnueabi-addr2line    arm-linux-gnueabi-gcc-ar          arm-linux-gnu
arm-linux-gnueabi-ar           arm-linux-gnueabi-gcc-nm          arm-linux-gnu
arm-linux-gnueabi-as           arm-linux-gnueabi-gcc-ranlib      arm-linux-gnu
arm-linux-gnueabi-c++          arm-linux-gnueabi-gcov            arm-linux-gnu
arm-linux-gnueabi-c++filt      arm-linux-gnueabi-gcov-dump       arm-linux-gnu
arm-linux-gnueabi-cpp          arm-linux-gnueabi-gcov-tool       arm-linux-gnu
arm-linux-gnueabi-dwp          arm-linux-gnueabi-gdb             arm-linux-gnu
arm-linux-gnueabi-elfedit      arm-linux-gnueabi-gfortran        arm-linux-gnu
arm-linux-gnueabi-g++          arm-linux-gnueabi-gprof           arm-linux-gnu
arm-linux-gnueabi-gcc          arm-linux-gnueabi-ld
arm-linux-gnueabi-gcc-7.3.1    arm-linux-gnueabi-ld.bfd
```

文件名称(N)： arm-linux-gnueabi-g++

文件类型： 所有文件 (*)



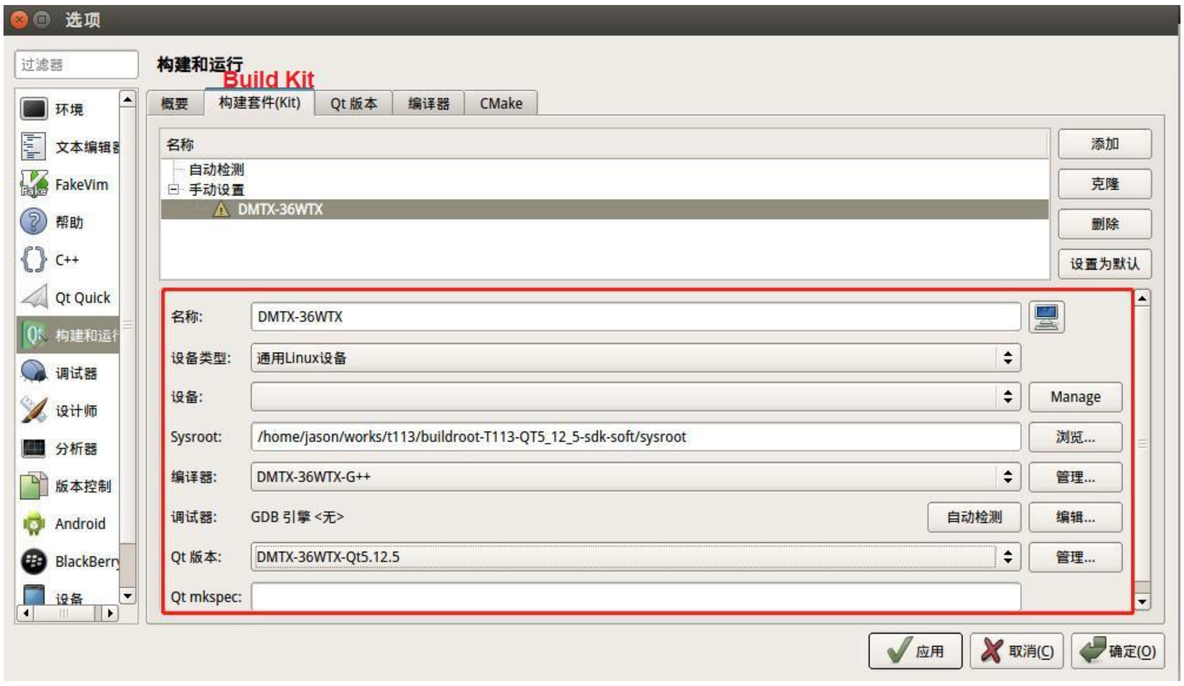
名称	类型
自动检测	
GCC (x86 64bit in /usr/bin)	GCC
GCC (x86 32bit in /usr/bin)	GCC
手动设置	
DMTX-36WTX-G++	GCC

名称: DMTX-36WTX-G++

编译器路径(C): 13-QT5_12_5-sdk-soft/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-g++

ABI: arm-linux-generic-elf-32bi - arm - linux - generic - elf - 32bit

Set up the build kit: Select "Build & Run"→"Build Kit".



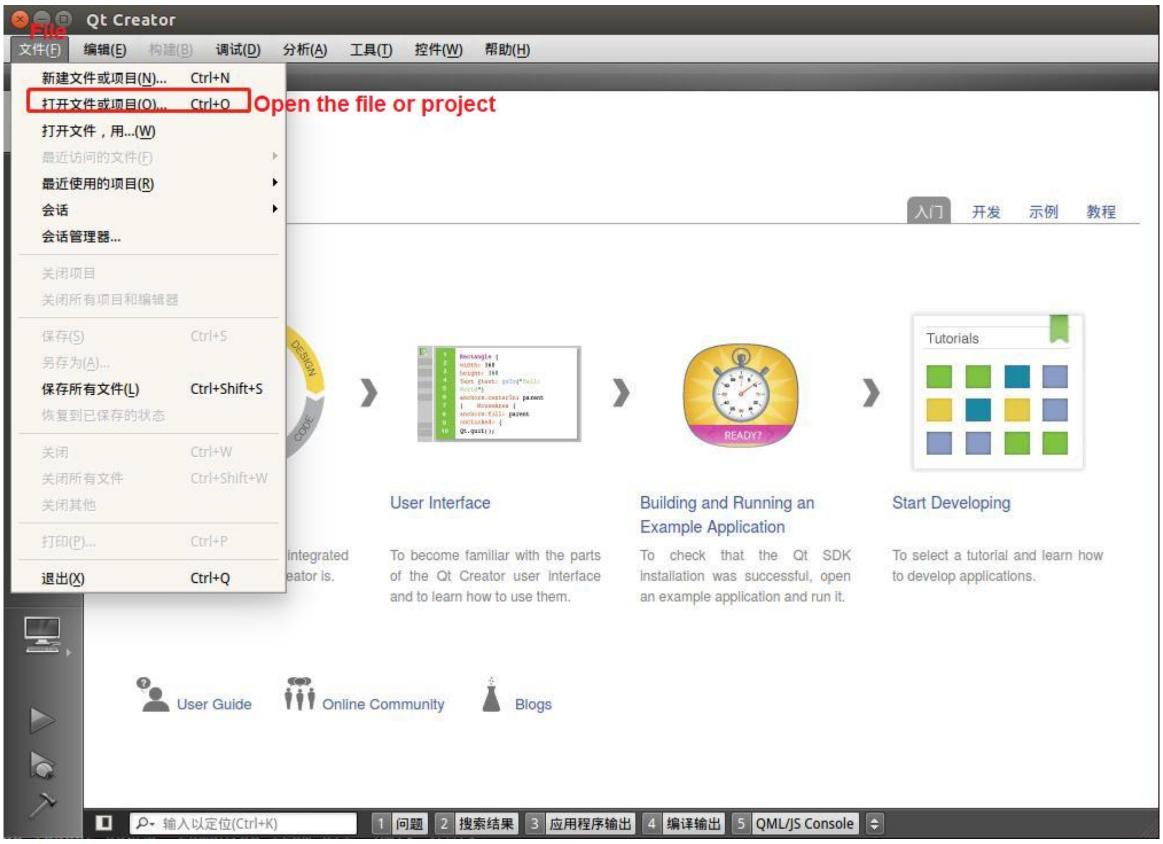
名称
- 自动检测
- 手动设置
  - DMTX-36WTX

名称: DMTX-36WTX
设备类型: 通用Linux设备
设备:
Sysroot: /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot
编译器: DMTX-36WTX-G++
调试器: GDB 引擎 <无>
Qt 版本: DMTX-36WTX-Qt5.12.5
Qt mkspec:

3.1.6 Compile Qt Project

3.1.6.1 Open the Project

File→Open the file or project:



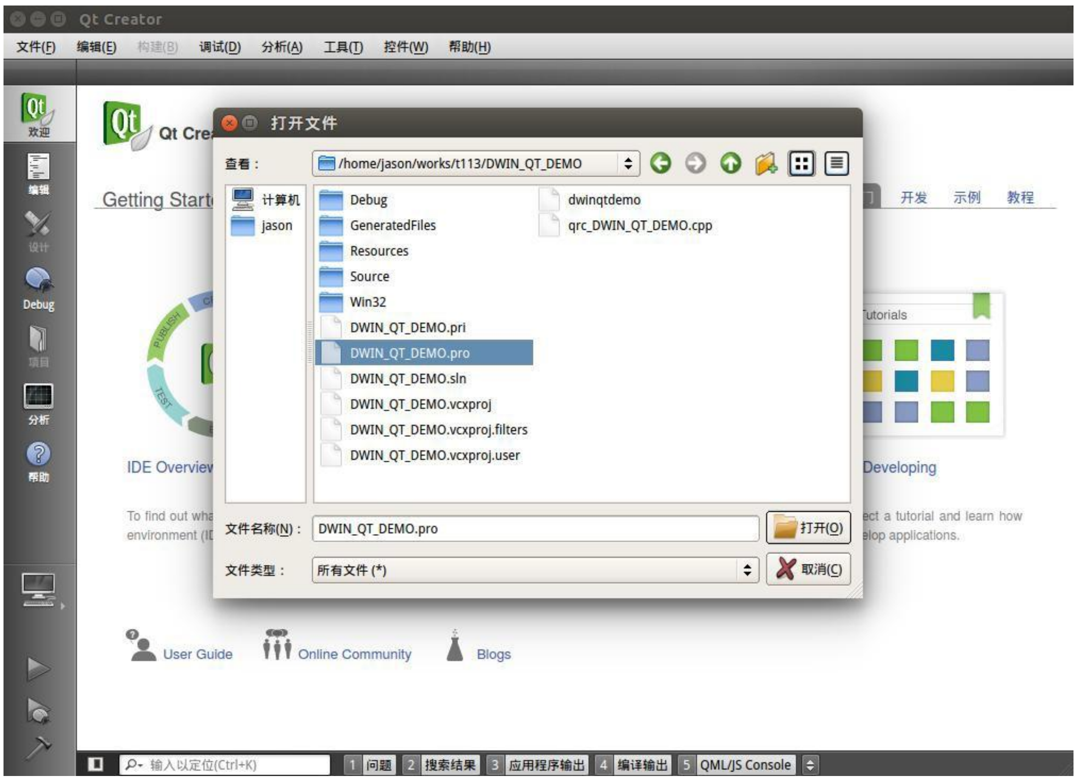
integrated ...eator is.

To become familiar with the parts of the Qt Creator user interface and to learn how to use them.

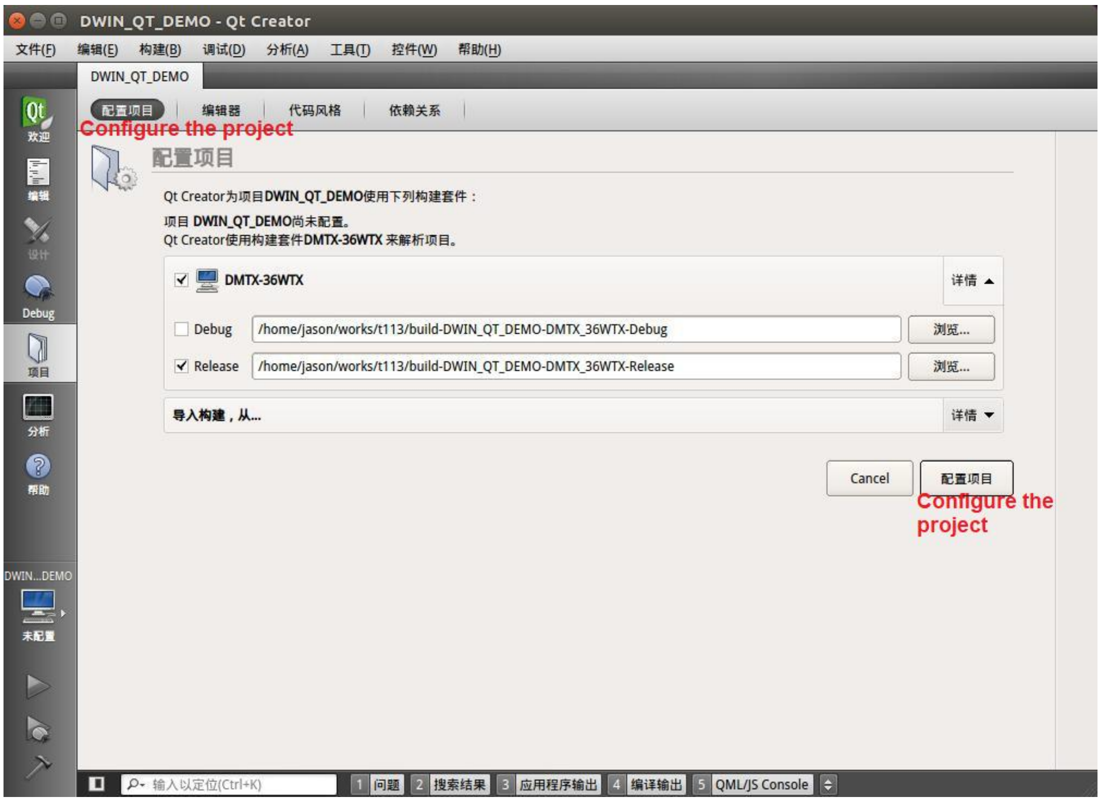
To check that the Qt SDK installation was successful, open an example application and run it.

To select a tutorial and learn how to develop applications.

Select the Qt project you want to open:



Configure the project:


DWIN_QT_DEMO - Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) 分析(A) 工具(T) 控件(W) 帮助(H)
配置项目 编辑器 代码风格 依赖关系
Configure the project
配置项目
详情
导入构建，从...
Cancel
配置项目
Configure the project

Qt Creator为项目DWIN_QT_DEMO使用下列构建套件：
项目 DWIN_QT_DEMO尚未配置。
Qt Creator使用构建套件DMTX-36WTX 来解析项目。

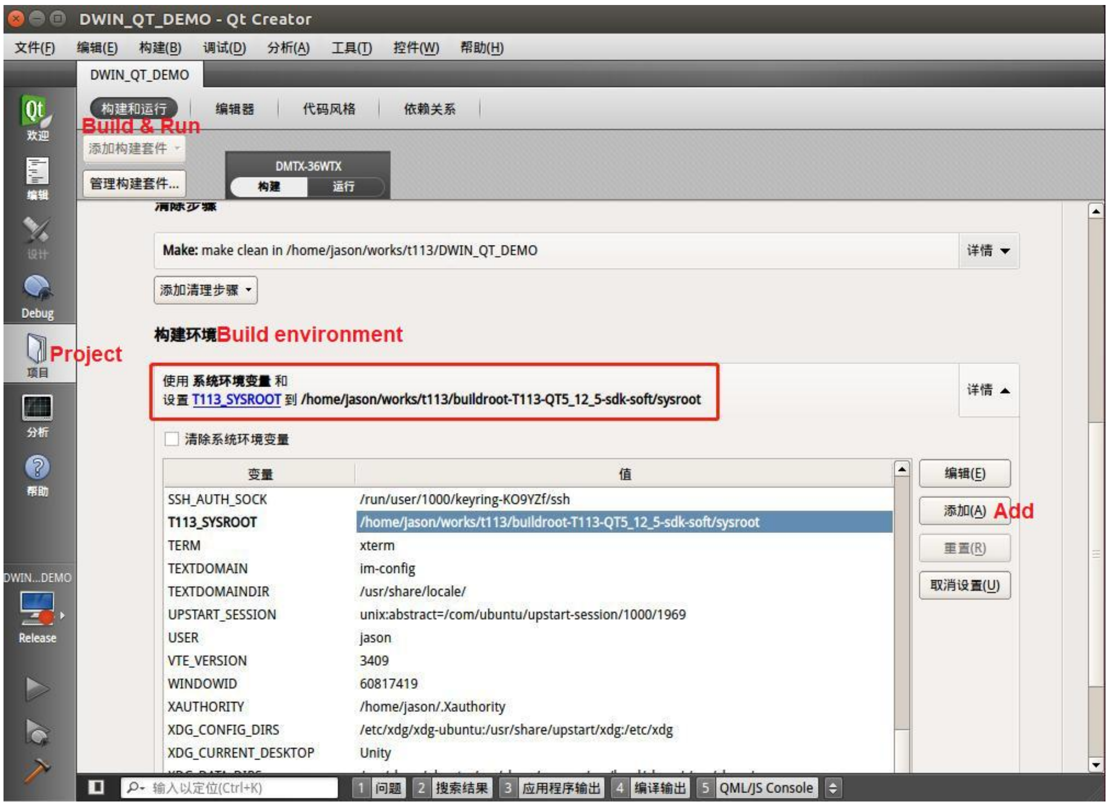
- [✓] DMTX-36WTX
- [ ] Debug /home/jason/works/t113/build-DWIN_QT_DEMO-DMTX_36WTX-Debug
- [✓] Release /home/jason/works/t113/build-DWIN_QT_DEMO-DMTX_36WTX-Release

3.1.6.2 Add Environment Variables

Project→Build & Run→Build Environment, add a variable value. Variable name: T113_SYSROOT

The value of the variable:

The "sysroot" directory of the "buildroot-T113-QT5_12_5-sdk-soft20221012.tar.gz" package.


DWIN_QT_DEMO - Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) 分析(A) 工具(T) 控件(W) 帮助(H)
DWIN_QT_DEMO
构建和运行 编辑器 代码风格 依赖关系
Build & Run
添加构建套件
管理构建套件...
DMTX-36WTX
构建 运行
清除步骤
添加清理步骤
构建环境Build environment
Project
清除系统环境变量
编辑(E)
添加(A) Add
重置(R)
取消设置(U)
详情
输入以定位(Ctrl+K)
1 问题 2 搜索结果 3 应用程序输出 4 编译输出 5 QML/JS Console

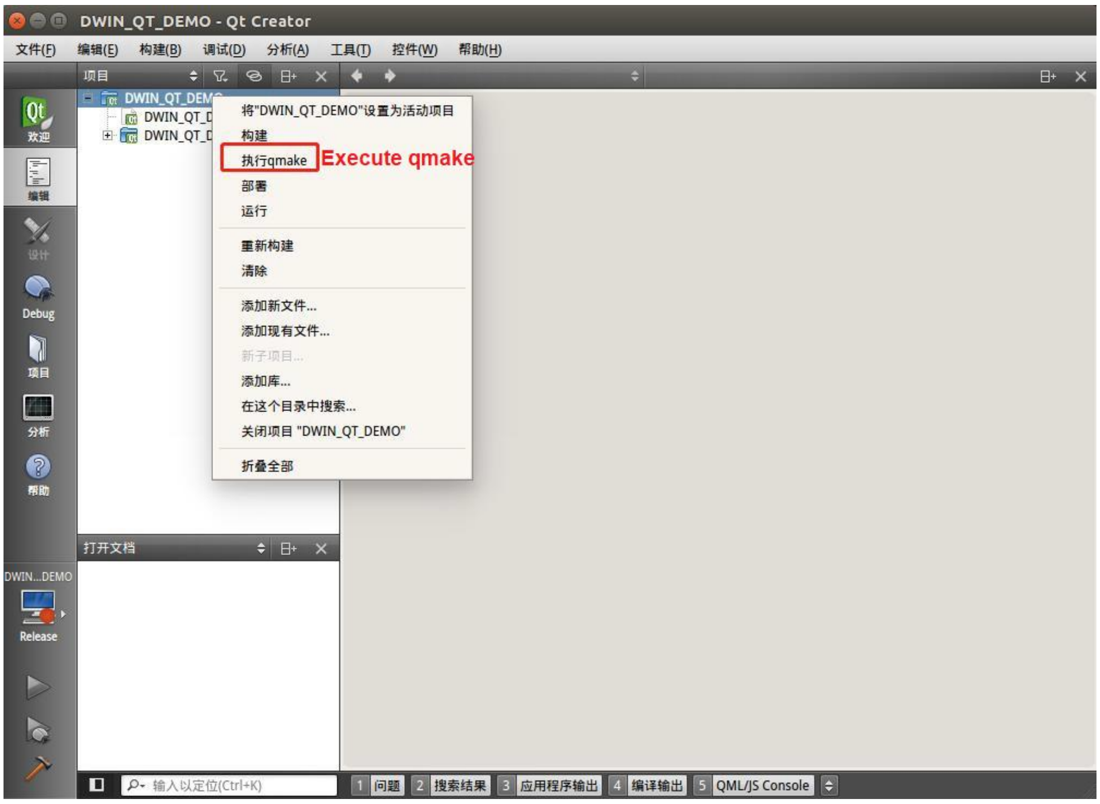
Make: make clean in /home/jason/works/t113/DWIN_QT_DEMO

使用 系统环境变量 和
设置 T113_SYSROOT 到 /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot

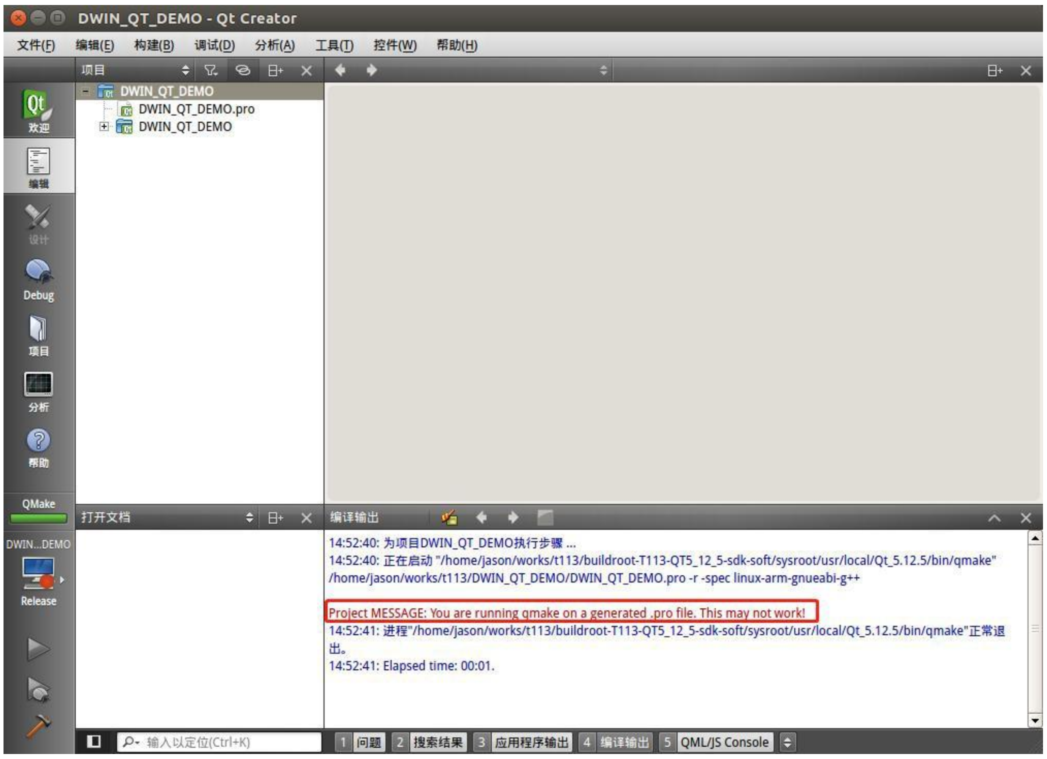
变量	值
SSH_AUTH_SOCK	/run/user/1000/keyring-KO9YZf/ssh
T113_SYSROOT	/home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot
TERM	xterm
TEXTDOMAIN	im-config
TEXTDOMAINDIR	/usr/share/locale/
UPSTART_SESSION	unix:abstract=/com/ubuntu/upstart-session/1000/1969
USER	jason
VTE_VERSION	3409
WINDOWID	60817419
XAUTHORITY	/home/jason/.Xauthority
XDG_CONFIG_DIRS	/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg
XDG_CURRENT_DESKTOP	Unity

3.1.6.3 Execute qmake

Select the project, right-click→Execute qmake



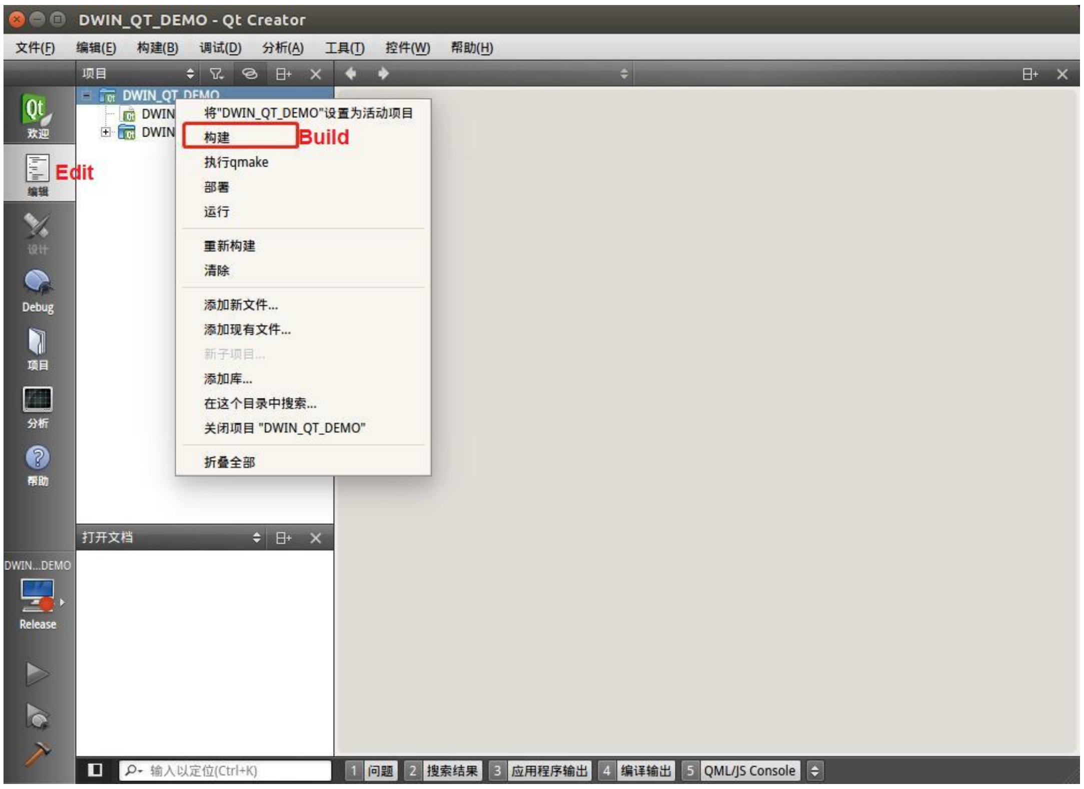
When qmake is successful, it is as shown in the following figure (the red part is the printout of DWIN_QT_DEMO.pri and does not affect the result):



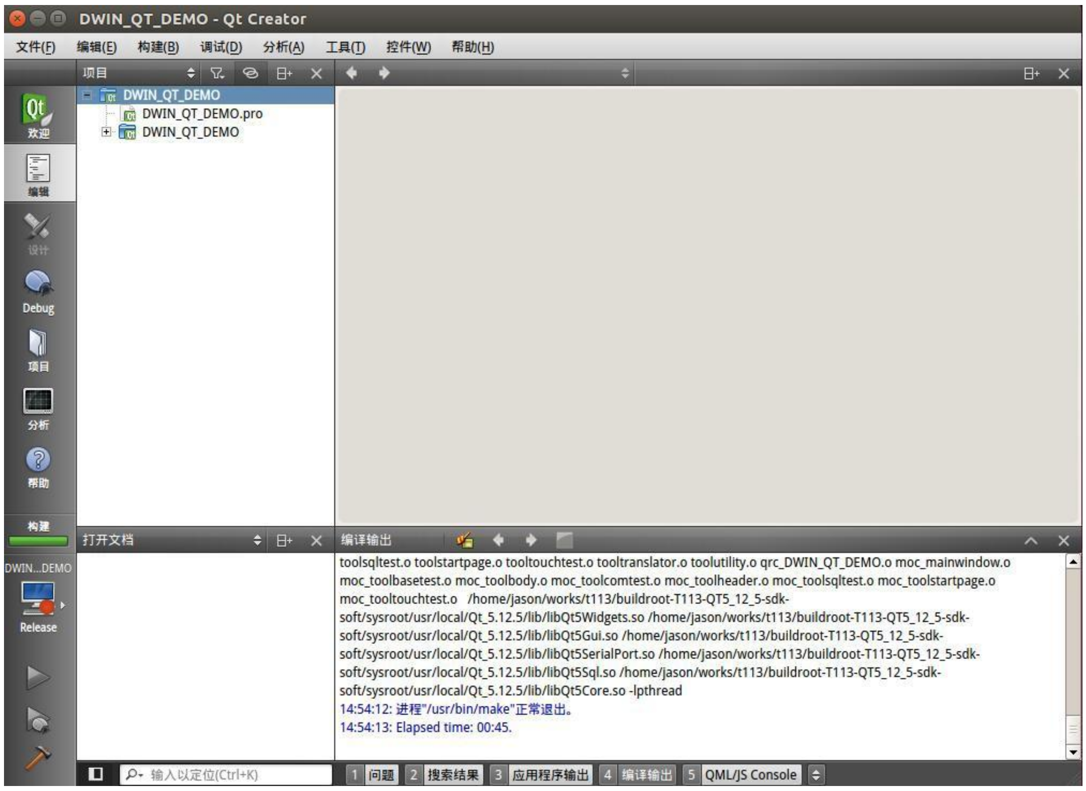
```
14:52:40: 为项目DWIN_QT_DEMO执行步骤 ...
14:52:40: 正在启动 "/home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot/usr/local/Qt_5.12.5/bin/qmake"
/home/jason/works/t113/DWIN_QT_DEMO/DWIN_QT_DEMO.pro -r -spec linux-arm-gnueabi-g++

Project MESSAGE: You are running qmake on a generated .pro file. This may not work!
14:52:41: 进程"/home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-soft/sysroot/usr/local/Qt_5.12.5/bin/qmake"正常退
出。
14:52:41: Elapsed time: 00:01.
```

3.1.6.4 Build

Edit→Select the project, right-click→Build

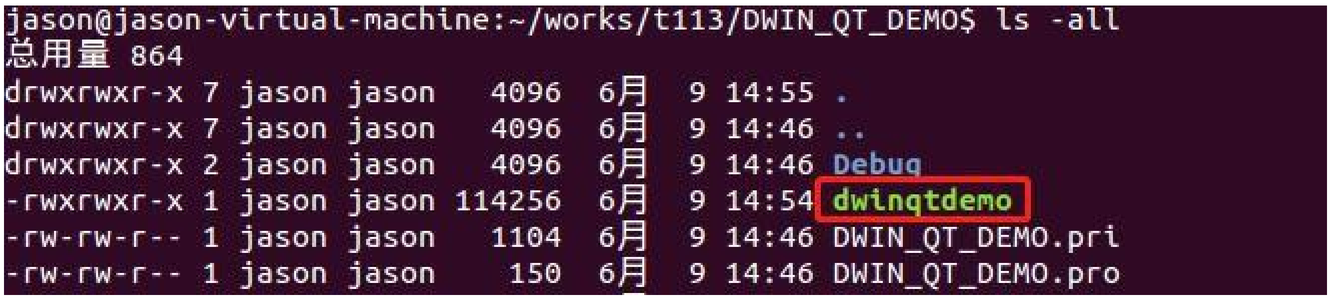


Build completed



```
toolsqltest.o toolstartpage.o tooltouchtest.o tooltranslator.o toolutility.o qrc_DWIN_QT_DEMO.o moc_mainwindow.o
moc_toolbasetest.o moc_toolbody.o moc_toolcomtest.o moc_toolheader.o moc_toolsqltest.o moc_toolstartpage.o
moc_tooltouchtest.o   /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-
soft/sysroot/usr/local/Qt_5.12.5/lib/libQt5Widgets.so /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-
soft/sysroot/usr/local/Qt_5.12.5/lib/libQt5Gui.so /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-
soft/sysroot/usr/local/Qt_5.12.5/lib/libQt5SerialPort.so /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-
soft/sysroot/usr/local/Qt_5.12.5/lib/libQt5Sql.so /home/jason/works/t113/buildroot-T113-QT5_12_5-sdk-
soft/sysroot/usr/local/Qt_5.12.5/lib/libQt5Core.so -lpthread
14:54:12: 进程"/usr/bin/make"正常退出。
14:54:13: Elapsed time: 00:45.
```

Up to this point, the target file has been generated in the project directory, and it can be copied to the screen for running.

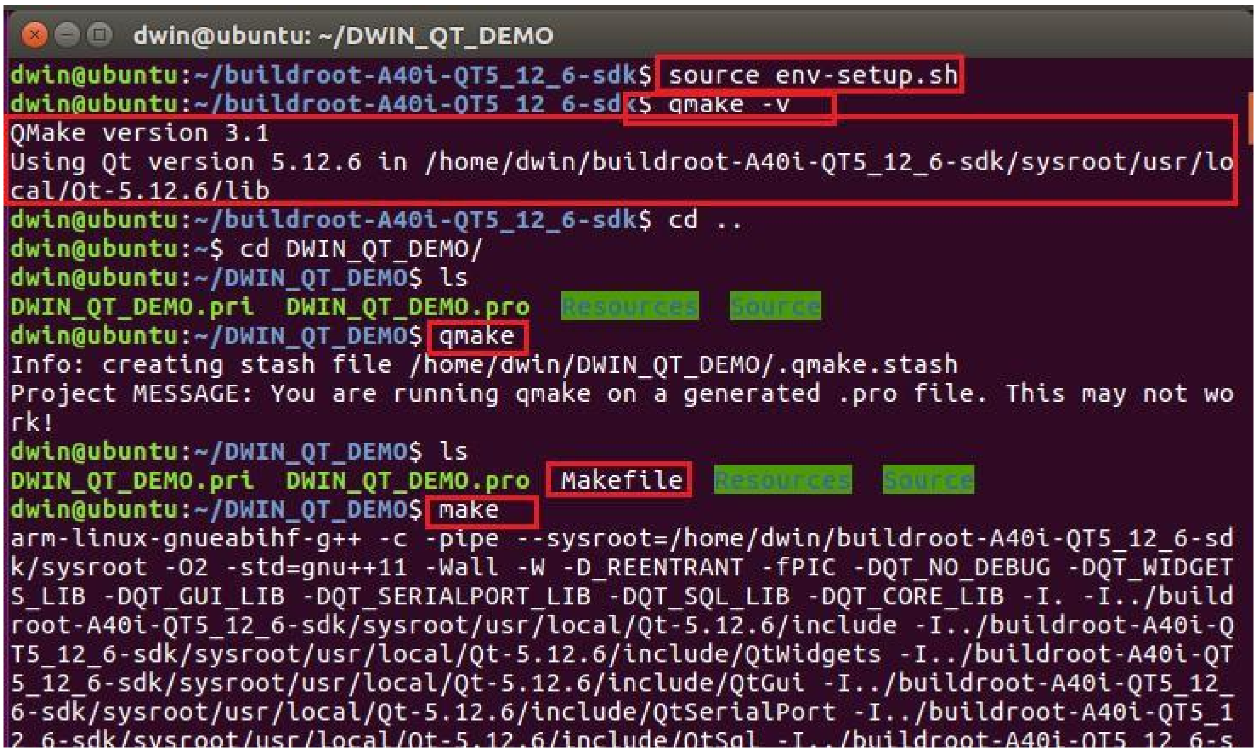


```
jason@jason-virtual-machine:~/works/t113/DWIN_QT_DEMO$ ls -all
总用量 864
drwxrwxr-x 7 jason jason   4096 6月  9 14:55 .
drwxrwxr-x 7 jason jason   4096 6月  9 14:46 ..
drwxrwxr-x 2 jason jason   4096 6月  9 14:46 Debug
-rwxrwxr-x 1 jason jason 114256 6月  9 14:54 dwinqtdemo
-rw-rw-r-- 1 jason jason   1104 6月  9 14:46 DWIN_QT_DEMO.pri
-rw-rw-r-- 1 jason jason    150 6月  9 14:46 DWIN_QT_DEMO.pro
```

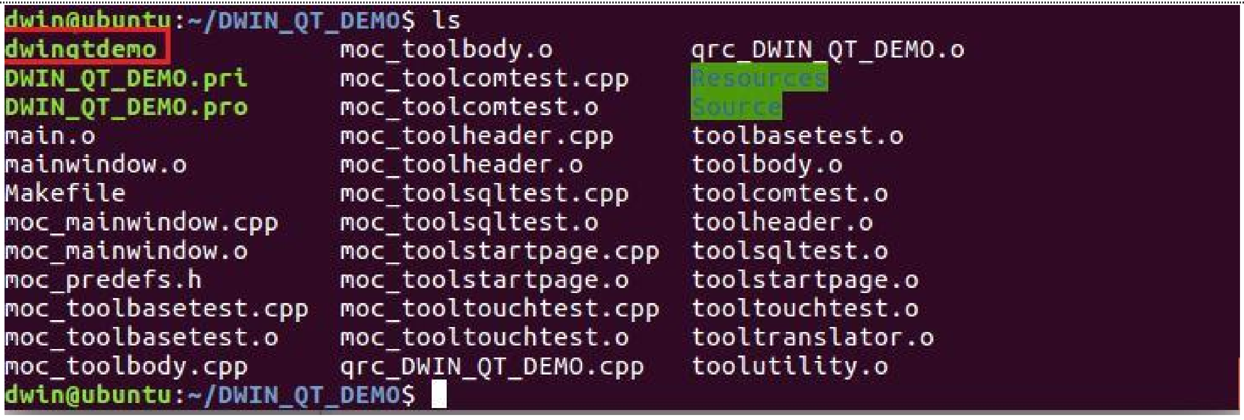
3.2 qmake

(1) After entering the environment that has been configured in Section 2.2 of Chapter 2 (that is, after running the source env-setup.sh command), you can use the **qmake -v** command to verify whether the environment is correct. Open the project folder that needs cross-compilation (here, the provided DWIN_QT_DEMO is selected, and put the folder into the /home/dwin/ directory of Ubuntu). Then, enter the instruction: **qmake** (if the .pro file has not been generated yet, you need to run **qmake -project** first). This will generate the Makefile file.

(2) Enter the command: **make**. Subsequently, a binary file named as a project will be generated. However, this file cannot be executed in Ubuntu and needs to be downloaded to the screen.



```
dwin@ubuntu: ~/DWIN_QT_DEMO
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ source env-setup.sh
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ qmake -v
QMake version 3.1
Using Qt version 5.12.6 in /home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot/usr/lo
cal/Qt-5.12.6/lib
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ cd ..
dwin@ubuntu:~$ cd DWIN_QT_DEMO/
dwin@ubuntu:~/DWIN_QT_DEMO$ ls
DWIN_QT_DEMO.pri  DWIN_QT_DEMO.pro  Resources  Source
dwin@ubuntu:~/DWIN_QT_DEMO$ qmake
Info: creating stash file /home/dwin/DWIN_QT_DEMO/.qmake.stash
Project MESSAGE: You are running qmake on a generated .pro file. This may not wo
rk!
dwin@ubuntu:~/DWIN_QT_DEMO$ ls
DWIN_QT_DEMO.pri  DWIN_QT_DEMO.pro  Makefile  Resources  Source
dwin@ubuntu:~/DWIN_QT_DEMO$ make
arm-linux-gnueabihf-g++ -c -pipe --sysroot=/home/dwin/buildroot-A40i-QT5_12_6-sd
k/sysroot -O2 -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_NO_DEBUG -DQT_WIDGET
S_LIB -DQT_GUI_LIB -DQT_SERIALPORT_LIB -DQT_SQL_LIB -DQT_CORE_LIB -I. -I../build
root-A40i-QT5_12_6-sdk/sysroot/usr/local/Qt-5.12.6/include -I../buildroot-A40i-Q
T5_12_6-sdk/sysroot/usr/local/Qt-5.12.6/include/QtWidgets -I../buildroot-A40i-QT
5_12_6-sdk/sysroot/usr/local/Qt-5.12.6/include/QtGui -I../buildroot-A40i-QT5_12_
6-sdk/sysroot/usr/local/Qt-5.12.6/include/QtSerialPort -I../buildroot-A40i-QT5_1
2_6-sdk/sysroot/usr/local/Qt-5.12.6/include/QtSql -I../buildroot-A40i-QT5_12_6-s
```



```
dwin@ubuntu:~/DWIN_QT_DEMO$ ls
dwinqtdemo             moc_toolbody.o          qrc_DWIN_QT_DEMO.o
DWIN_QT_DEMO.pri       moc_toolcomtest.cpp     Resources
DWIN_QT_DEMO.pro       moc_toolcomtest.o       Source
main.o                 moc_toolheader.cpp      toolbasetest.o
mainwindow.o           moc_toolheader.o        toolbody.o
Makefile               moc_toolsqltest.cpp     toolcomtest.o
moc_mainwindow.cpp     moc_toolsqltest.o       toolheader.o
moc_mainwindow.o       moc_toolstartpage.cpp   toolsqltest.o
moc_predefs.h          moc_toolstartpage.o     toolstartpage.o
moc_toolbasetest.cpp   moc_tooltouchtest.cpp   tooltouchtest.o
moc_toolbasetest.o     moc_tooltouchtest.o     tooltranslator.o
moc_toolbody.cpp       qrc_DWIN_QT_DEMO.cpp    toolutility.o
dwin@ubuntu:~/DWIN_QT_DEMO$
```

3.3 Download via USB Flash Drive

(1) Put the compiled file under the shared folder. You can copy the file using the command: **cp** (file name) (shared folder path). For example: **cp dwinqtdemo /mnt/hgfs/share/**

(2) Move the target file in the shared folder from the computer's files to the USB flash drive.

(3) Insert the USB flash drive into the screen.

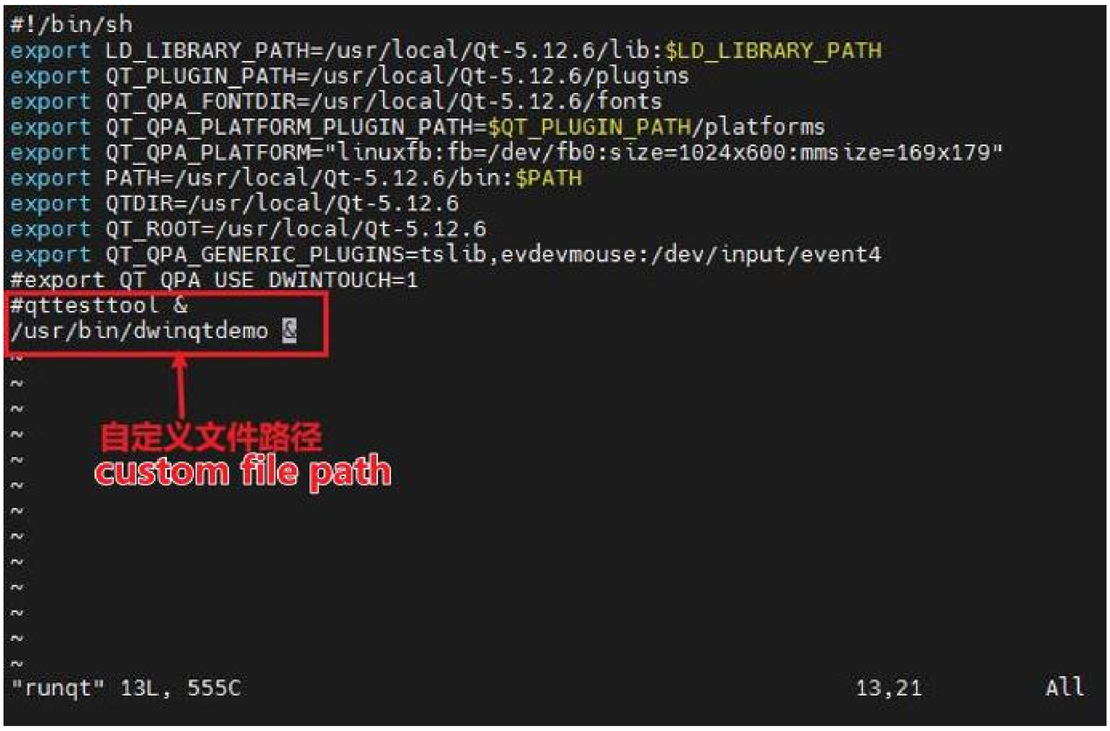
(4) Open MobaXterm and establish a connection. Then, enter the command: **cd /mnt/usb** to enter the "usb" folder. Select the "sdax" folder and copy or move the target file to the target directory (you can customize the folder at will to avoid clutter caused by too many files later). Use the command: **cp** (target file) (folder). For example: **cp dwinqtdemo /usr/bin/.**

3.4 Run dwinqtdemo

To run the above-mentioned program, need to modify the configuration file **/etc/init.d/runqt**.

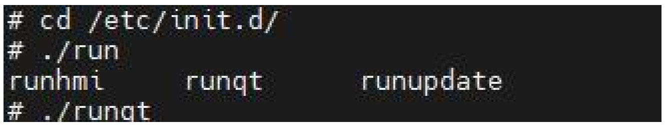
Enter: **vi /etc/init.d/runqt**.

Move the cursor to the beginning of the **qttesttool** line, press **i** to enter the input mode, type # to comment out the line. Move the cursor to the end of the line, press Enter key to go to the next line, then enter the absolute path of the dwinqtdemo program + space + &. Press Esc to exit input mode, and enter: (colon)wq to save the file modifications.

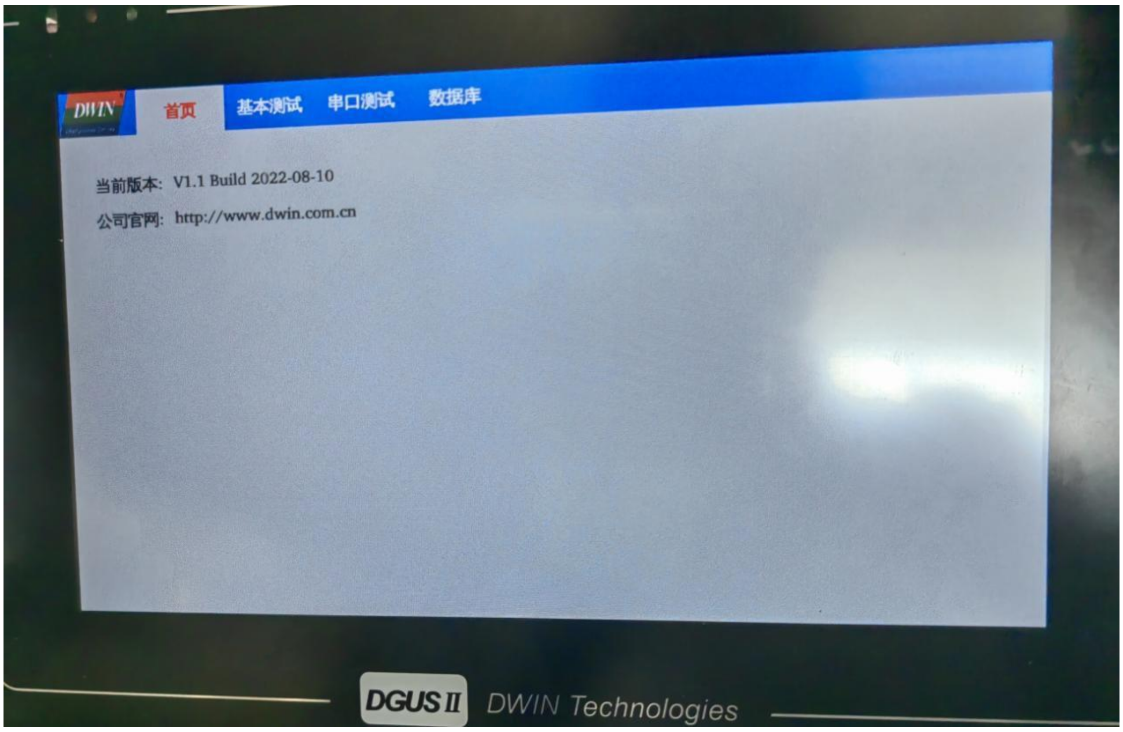


```
#!/bin/sh
export LD_LIBRARY_PATH=/usr/local/Qt-5.12.6/lib:$LD_LIBRARY_PATH
export QT_PLUGIN_PATH=/usr/local/Qt-5.12.6/plugins
export QT_QPA_FONTDIR=/usr/local/Qt-5.12.6/fonts
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_PLUGIN_PATH/platforms
export QT_QPA_PLATFORM="linuxfb:fb=/dev/fb0:size=1024x600:mmsize=169x179"
export PATH=/usr/local/Qt-5.12.6/bin:$PATH
export QTDIR=/usr/local/Qt-5.12.6
export QT_ROOT=/usr/local/Qt-5.12.6
export QT_QPA_GENERIC_PLUGINS=tslib,evdevmouse:/dev/input/event4
#export QT_QPA_USE_DWINTOUCH=1
#qttesttool &
/usr/bin/dwinqtdemo &
~
~
~
~
~
~
~
~
~
~
~
~
~
"runqt" 13L, 555C                                            13,21         All
```

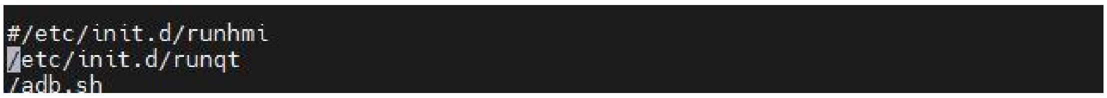
Then the program can be ran by runqt.



```
# cd /etc/init.d/
# ./run
runhmi      runqt      runupdate
# ./runqt
```


DWIN
首页
基本测试
串口测试
数据库
当前版本: V1.1 Build 2022-08-10
公司官网: http://www.dwin.com.cn
DGUS II DWIN Technologies

If the initial setting is not to run the runqt program by default, you can modify it in the /etc/init.d/rcS file. If you want to set runqt to start automatically when the system boots up, set the last three lines of /etc/init.d/rcS to the following code:



```
#/etc/init.d/runhmi
/etc/init.d/runqt
/adb.sh
```

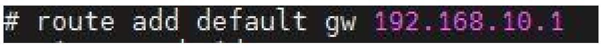
After saving, enter the "**reboot**" command and then restart the screen.

3.5 Network Function

3.5.1 Network Configuration

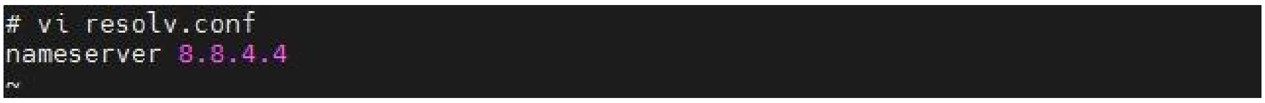
Here, it is recommended to use a serial port to connect to the device for configuration.

(1) After connecting the network cable, configure the gateway: **route add default gw** Gateway IP (in my case, it is 192.168.10.1).



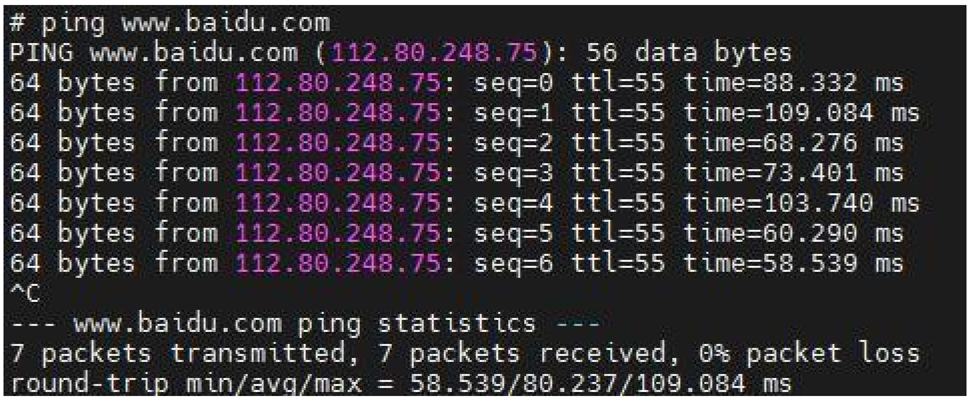
```
# route add default gw 192.168.10.1
```

(2) Configure the DNS. Use the command: **vi /etc/resolv.conf**. Press the "i" key to enter the input mode, and then enter: nameserver 8.8.4.4. After that, press the ESC key, enter: (colon)wq, save and exit.



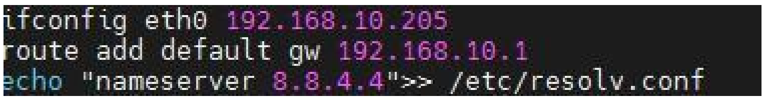
```
# vi resolv.conf
nameserver 8.8.4.4
~
```

(3) Try to ping an external network address and check the result.



```
# ping www.baidu.com
PING www.baidu.com (112.80.248.75): 56 data bytes
64 bytes from 112.80.248.75: seq=0 ttl=55 time=88.332 ms
64 bytes from 112.80.248.75: seq=1 ttl=55 time=109.084 ms
64 bytes from 112.80.248.75: seq=2 ttl=55 time=68.276 ms
64 bytes from 112.80.248.75: seq=3 ttl=55 time=73.401 ms
64 bytes from 112.80.248.75: seq=4 ttl=55 time=103.740 ms
64 bytes from 112.80.248.75: seq=5 ttl=55 time=60.290 ms
64 bytes from 112.80.248.75: seq=6 ttl=55 time=58.539 ms
^C
--- www.baidu.com ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 58.539/80.237/109.084 ms
```

(4) If you need to permanently modify the gateway and DNS, you can add the following statements after "ifconfig eth0 IP address" in the /etc/init.d/S40network file: route add default gw Gateway address; echo "nameserver 8.8.4.4" >> /etc/resolv.conf". After the modification, it is as shown in the following figure.



```
ifconfig eth0 192.168.10.205
route add default gw 192.168.10.1
echo "nameserver 8.8.4.4">> /etc/resolv.conf
```

3.6 System Time Setting

Command format:

5A	A5	08	02	Year-2000	Month-1	Day	Hour	Minute	Second	checksum

C++ example:

```
void BasicTester::SetTime(const QDateTime& time)

{

    if (!time.isValid())

        return;


    unsigned char cmd[11] = {0};

    cmd[0] = 0x5A;

    cmd[1] = 0xA5;

    cmd[2] = 0x08;

    cmd[3] = 0x02;

    cmd[4] = time.date().year() - 2000;

    cmd[5] = time.date().month() - 1;

    cmd[6] = time.date().day();

    cmd[7] = time.time().hour();

    cmd[8] = time.time().minute();

    cmd[9] = time.time().second();


    unsigned char checksum = 0;

    for (int i = 2; i < 10; i++)

        checksum += cmd[i];

    cmd[10] = checksum;


    // set time to tps02

    int fd = open(DWIN_TPS02_UART, O_RDWR|O_NONBLOCK|O_NOCTTY);

    if (fd < 0)

        return;


    ::write(fd, cmd, 11);

    ::close(fd);
    struct tm t;
    t.tm_year = time.date().year() - 1900;
    t.tm_mon = time.date().month() - 1;
    t.tm_mday = time.date().day();
    t.tm_hour = time.time().hour();
    t.tm_min = time.time().minute();
    t.tm_sec = time.time().second();

    struct timeval tv;
    tv.tv_sec  = mktime(&t);
    tv.tv_usec = 0;
    settimeofday(&tv,NULL);
}
```

3.7 Brightness Adjustment

Adjust brightness by sending message from internal serial port, the internal serial of 36 series is /dev/ttyS1.

CMD=0x04, DATA=Diming_Set(0x00-0x64), Byte5 is checksum.

5A	A5	03	04	0×00-0×64	checksum

Use command echo to test:

Off (set brightness is 0x00): echo -e -n "\x5A\xA5\x03\x04\x00\x07" > /dev/ttyS1

On (set brightness is 0x60): echo -e -n "\x5A\xA5\x03\x04\x60\x67" > /dev/ttyS1

C++ example:

```
 void BasicTester::SetBrightless(unsigned char brightless)
{
    if (brightless > 0x64)
        return;

    unsigned char cmd[6] = {0};
    cmd[0] = 0x5A;
    cmd[1] = 0xA5;
    cmd[2] = 0x03;

    cmd[3] = 0x04;

    cmd[4] = brightless;


    unsigned char checksum = 0;

    for (int i = 2; i < 5; i++)

        checksum += cmd[i];

    cmd[5] = checksum;


    // set time to tps02

    int fd = open(DWIN_TPS02_UART, O_RDWR|O_NONBLOCK|O_NOCTTY);

    if (fd < 0)

        return;


    ::write(fd, cmd, 6);

    ::close(fd);

}
```

3.8 Buzzer Switch

CMD=0x03, DATA=(0x00:OFF, 0xFF:ON), Byte5 is checksum.

5A	A5	03	03	0×00 | 0×FF	Checksum

Use command echo to test:

Buzzer on: echo -e -n "\x5A\xA5\x03\x03\xFF\x05" > /dev/ttyS1

Buzzer off: echo -e -n "\x5A\xA5\x03\x03\x00\x06" > /dev/ttyS1

C++ example:

```
void BasicTester::SetBeep(bool status)

{

    int fd = ::open(DWIN_TPS02_UART, O_RDWR | O_NONBLOCK | O_NOCTTY);

    if (fd < 0)

        return;
if (status) { // set beep on

const unsigned char cmdBeepOn[6] = {0x5A, 0xA5, 0x03, 0x03, 0xFF, 0x05};

::write(fd, cmdBeepOn, 6);

} else { // set beep off

const unsigned char cmdBeepOff[6] = {0x5A, 0xA5, 0x03, 0x03, 0x00, 0x06};

::write(fd, cmdBeepOff, 6);

}

::close(fd);

}
```

4 Cross-Compilation of LVGL Project Files

4.1 Software Requirements

LVGL supported version: 9.1.0

Download path: https://github.com/lvgl/lvgl/releases/tag/v9.1.0

Reference for LVGL development guide: https://docs.lvgl.io/master/

Compilation environment: Ubuntu 14.04 and other versions. For the environment setup, please refer to Chapter 2.

Refer to the DEMO and resources: LvglDemo,

toolchain: buildroot-T113-LVGL9_1_0-sdk-soft20241216.tar, lvgl_demo_source.

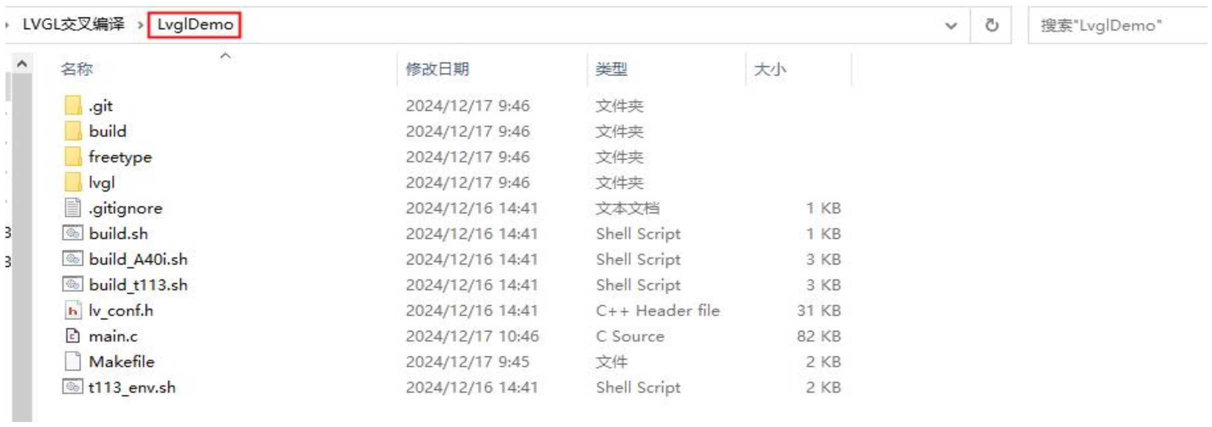
4.2 Hardware Introduction

Currently, only the capacitive touch screen models of the 36 series are supported.

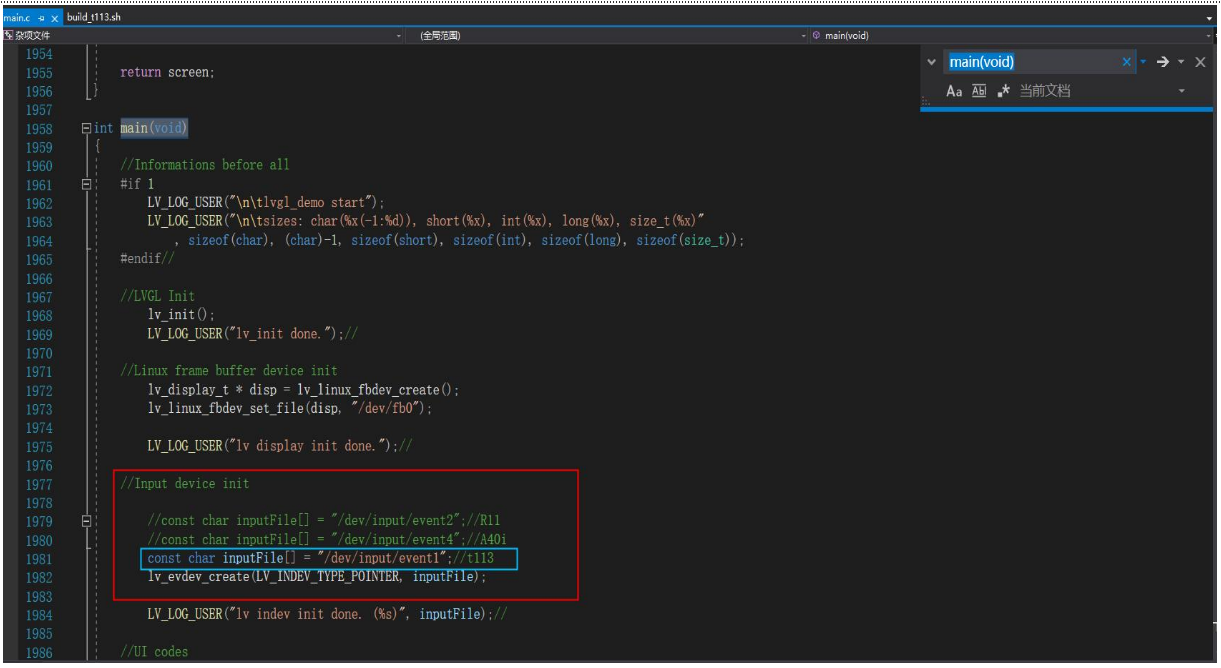
4.3 Cross Compilation

4.3.1 Preparation before Compilation

Open the main.c in LvglDemo, search for "main(void)" and perform the initialization of the input device. In this example, the 36 series is equipped with T113-S3, so select "t113".



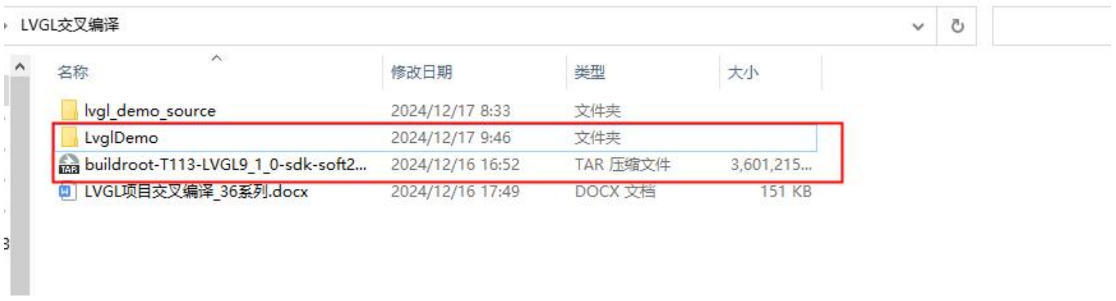
名称	修改日期	类型	大小
.git	2024/12/17 9:46	文件夹	
build	2024/12/17 9:46	文件夹	
freetype	2024/12/17 9:46	文件夹	
lvgl	2024/12/17 9:46	文件夹	
.gitignore	2024/12/16 14:41	文本文档	1 KB
build.sh	2024/12/16 14:41	Shell Script	1 KB
build_A40i.sh	2024/12/16 14:41	Shell Script	3 KB
build_t113.sh	2024/12/16 14:41	Shell Script	3 KB
lv_conf.h	2024/12/16 14:41	C++ Header file	31 KB
main.c	2024/12/17 10:46	C Source	82 KB
Makefile	2024/12/17 9:45	文件	2 KB
t113_env.sh	2024/12/16 14:41	Shell Script	2 KB



```c
    return screen;
}

int main(void)
{
    //Informations before all
    #if 1
        LV_LOG_USER("\n\tlvgl_demo start");
        LV_LOG_USER("\n\tsizes: char(%x(-1:%d)), short(%x), int(%x), long(%x), size_t(%x)"
            , sizeof(char), (char)-1, sizeof(short), sizeof(int), sizeof(long), sizeof(size_t));
    #endif//

    //LVGL Init
        lv_init();
        LV_LOG_USER("lv_init done.");//

    //Linux frame buffer device init
        lv_display_t * disp = lv_linux_fbdev_create();
        lv_linux_fbdev_set_file(disp, "/dev/fb0");

        LV_LOG_USER("lv display init done.");//

    //Input device init

        //const char inputFile[] = "/dev/input/event2";//R11
        //const char inputFile[] = "/dev/input/event4";//A40i
        const char inputFile[] = "/dev/input/event1";//t113
        lv_evdev_create(LV_INDEV_TYPE_POINTER, inputFile);

        LV_LOG_USER("lv indev init done. (%s)", inputFile);//

    //UI codes
```

4.3.2 Compilation Steps

Copy the folder LvglDemo and the toolchain buildroot-T113-LVGL9_1_0-sdk-soft20241216.tar to Ubuntu.



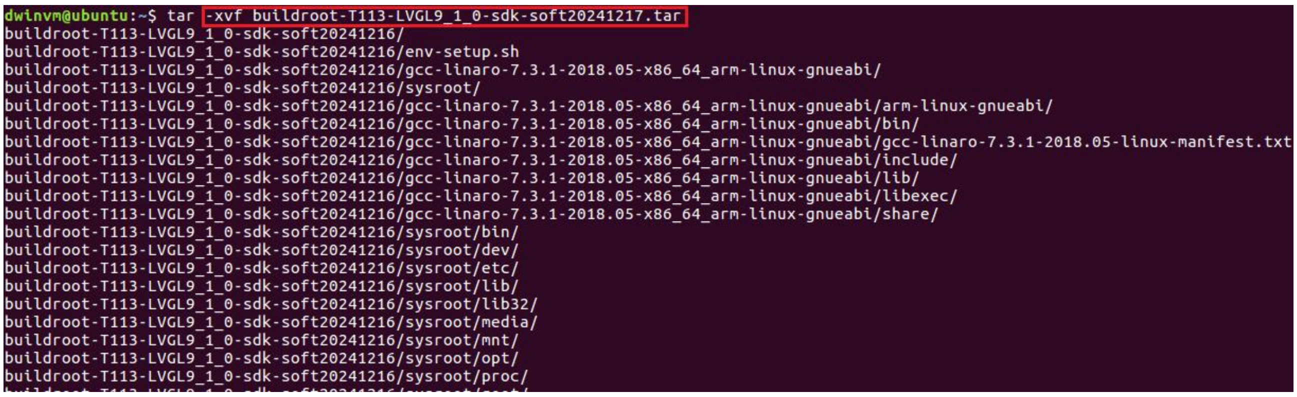
LVGL交叉编译

名称	修改日期	类型	大小
lvgl_demo_source	2024/12/17 8:33	文件夹	
LvglDemo	2024/12/17 9:46	文件夹	
buildroot-T113-LVGL9_1_0-sdk-soft2...	2024/12/16 16:52	TAR 压缩文件	3,601,215...
LVGL项目交叉编译_36系列.docx	2024/12/16 17:49	DOCX 文档	151 KB

Open the Terminal and extract the SDK (toolchain) buildroot-T113-LVGL9_1_0-sdk-soft20241216.tar.

Enter：

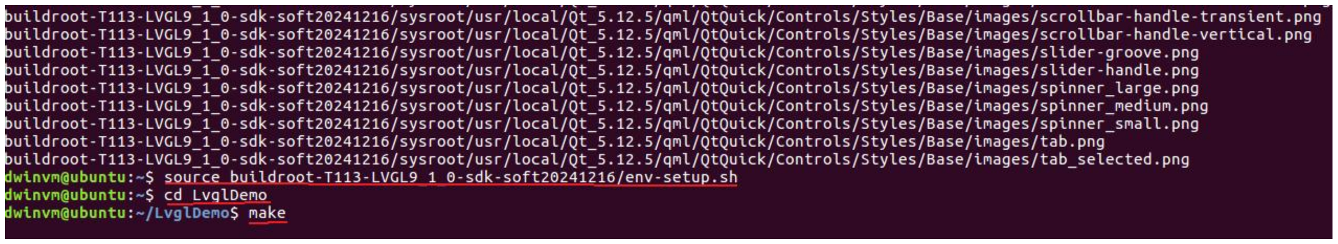
```
tar -xvf  buildroot-T113-LVGL9_1_0-sdk-soft20241216.tar
```



```
dwinvm@ubuntu:~$ tar -xvf buildroot-T113-LVGL9_1_0-sdk-soft20241217.tar
buildroot-T113-LVGL9_1_0-sdk-soft20241216/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/env-setup.sh
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/arm-linux-gnueabi/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/bin/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/gcc-linaro-7.3.1-2018.05-linux-manifest.txt
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/include/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/lib/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/libexec/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabi/share/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/bin/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/dev/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/etc/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/lib/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/lib32/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/media/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/mnt/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/opt/
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/proc/
```

After waiting for the extraction to complete, run the script→set the environment variables→open the project file→execute make to obtain the executable file.

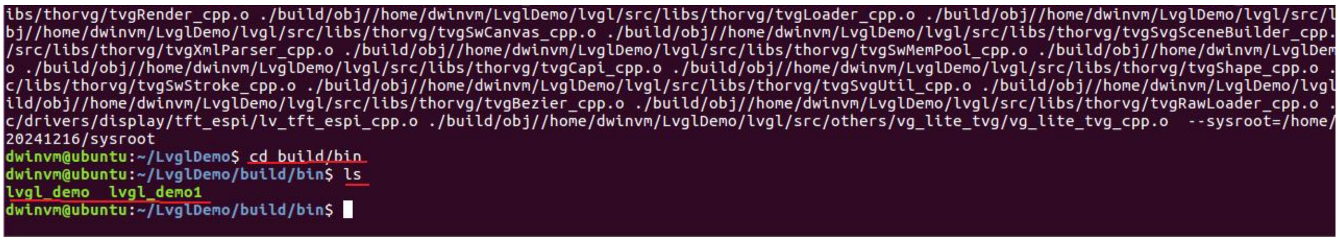
```
Source buildroot-T113-LVGL9_1_0-sdk-soft20241216/env-setup.sh
cd LvglDemo/
make
```



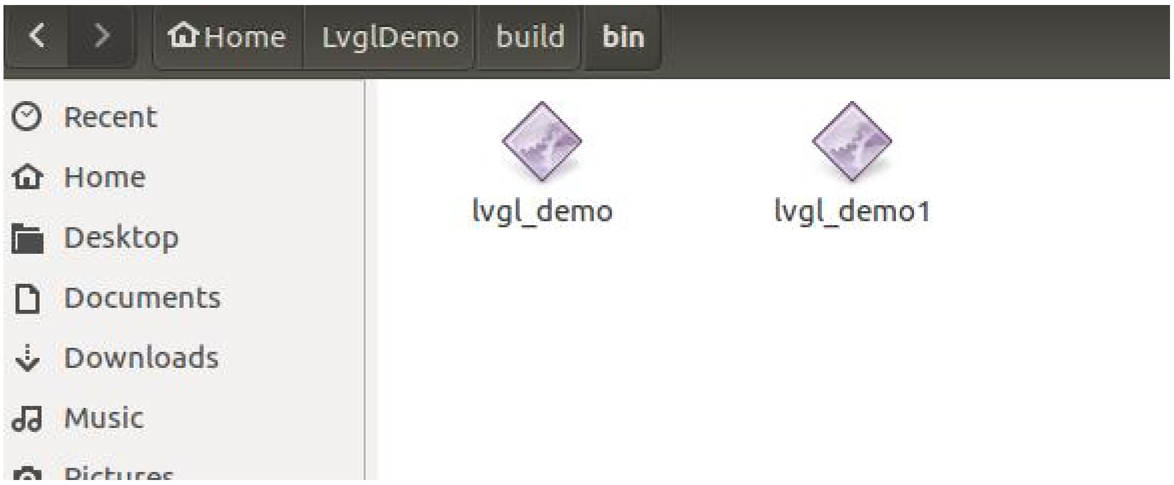
```
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/scrollbar-handle-transient.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/scrollbar-handle-vertical.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/slider-groove.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/slider-handle.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/spinner_large.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/spinner_medium.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/spinner_small.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/tab.png
buildroot-T113-LVGL9_1_0-sdk-soft20241216/sysroot/usr/local/Qt_5.12.5/qml/QtQuick/Controls/Styles/Base/images/tab_selected.png
dwinvm@ubuntu:~$ source buildroot-T113-LVGL9_1_0-sdk-soft20241216/env-setup.sh
dwinvm@ubuntu:~$ cd LvglDemo
dwinvm@ubuntu:~/LvglDemo$ make
```

Obtain the executable file lvgl_demo, and the file path is LvglDemo/build/bin.

```
cd build/bin/
```

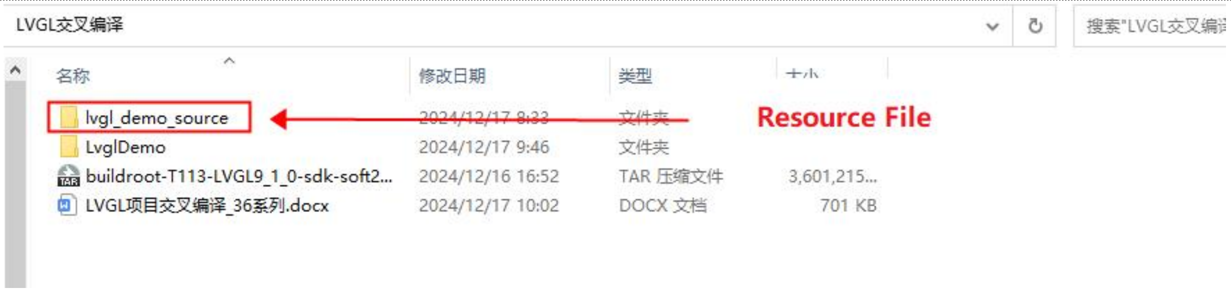


```
ibs/thorvg/tvgRender_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgLoader_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/l
bj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgSwCanvas_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgSvgSceneBuilder_cpp.
/src/libs/thorvg/tvgXmlParser_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgSwMemPool_cpp.o ./build/obj//home/dwinvm/LvglDem
o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgCapi_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgShape_cpp.o .
c/libs/thorvg/tvgSwStroke_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgSvgUtil_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl
ild/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgBezier_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/libs/thorvg/tvgRawLoader_cpp.o .
c/drivers/display/tft_espi/lv_tft_espi_cpp.o ./build/obj//home/dwinvm/LvglDemo/lvgl/src/others/vg_lite_tvg/vg_lite_tvg_cpp.o  --sysroot=/home/
20241216/sysroot
dwinvm@ubuntu:~/LvglDemo$ cd build/bin
dwinvm@ubuntu:~/LvglDemo/build/bin$ ls
lvgl_demo  lvgl_demo1
dwinvm@ubuntu:~/LvglDemo/build/bin$ 
```



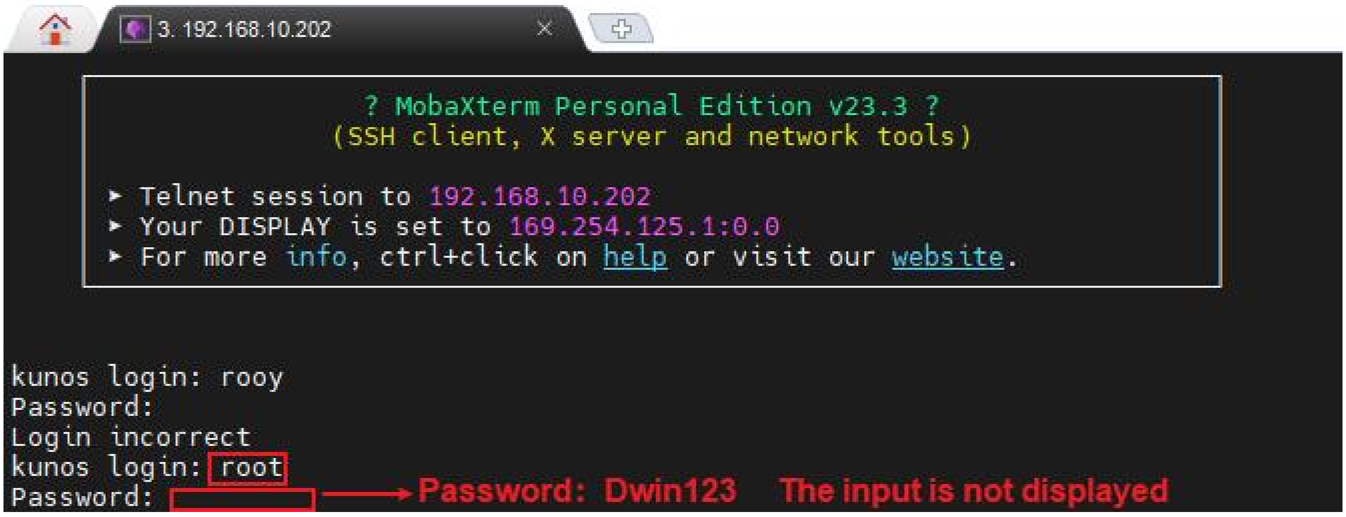
4.3.3 Run Tests

Copy the executable file lvgl_demo obtained from the compilation in the previous step and the provided lvgl_demo_source to a USB flash drive, and then insert the USB flash drive into the USB interface of the screen.


LVGL交叉编译

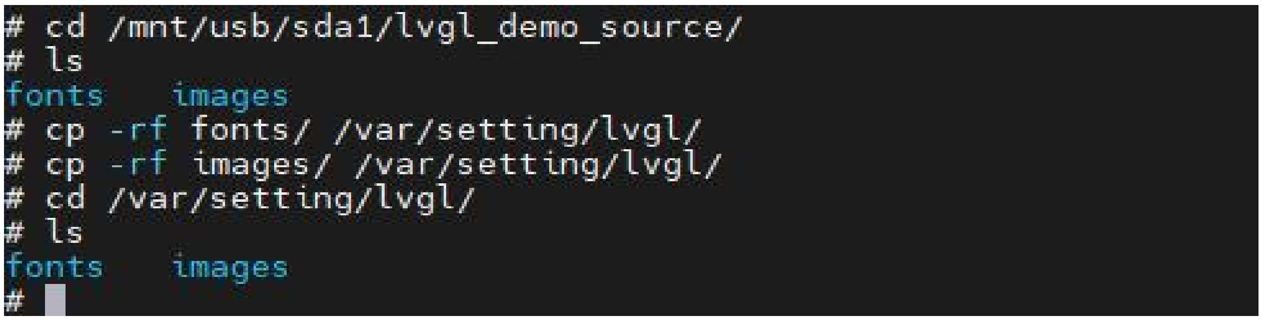
名称	修改日期	类型	大小
lvgl_demo_source	2024/12/17 8:33	文件夹	
LvglDemo	2024/12/17 9:46	文件夹	
buildroot-T113-LVGL9_1_0-sdk-soft2...	2024/12/16 16:52	TAR 压缩文件	3,601,215...
LVGL项目交叉编译_36系列.docx	2024/12/17 10:02	DOCX 文档	701 KB

Connect the screen to the computer via a network cable. For the specific operation steps, please refer to Telnet Connection via Ethernet. Use the MobaXterm tool to copy the two files, namely "fonts" and "images", from the USB flash drive to the **/var/setting/lvgl** directory.



```
? MobaXterm Personal Edition v23.3 ?
(SSH client, X server and network tools)

► Telnet session to 192.168.10.202
► Your DISPLAY is set to 169.254.125.1:0.0
► For more info, ctrl+click on help or visit our website.

kunos login: rooy
Password:
Login incorrect
kunos login: root
Password:
```

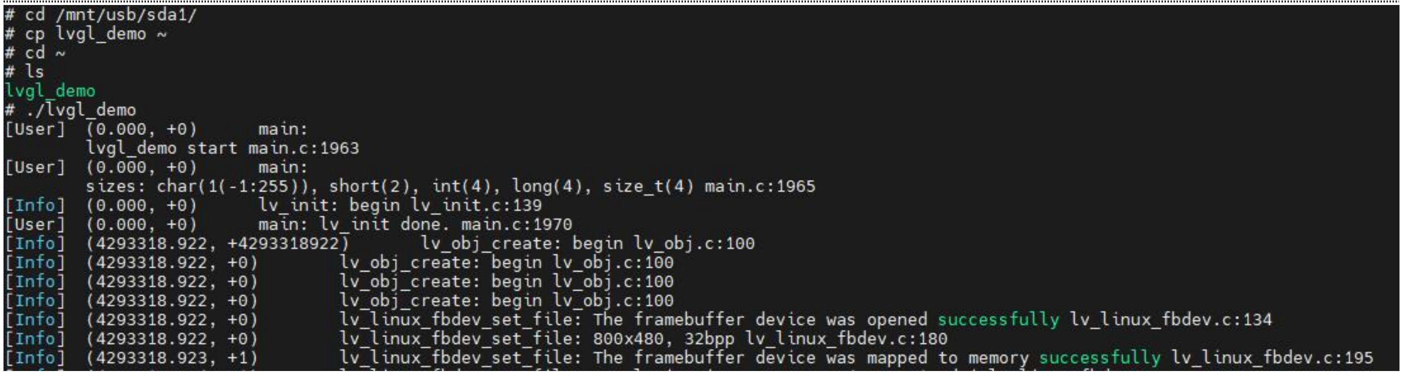
```
cd /mnt/usb/sdaXXX/lvgl_demo_source  (sdaXXX is the file location where the USB flash drive is mounted on the screen. It can be obtained by entering cd /mnt/usb/ and then pressing the Tab key.)
cp -rf fonts/ /var/setting/lvgl/
cp -rf images/ /var/setting/lvgl/
```



```
# cd /mnt/usb/sda1/lvgl_demo_source/
# ls
fonts    images
# cp -rf fonts/ /var/setting/lvgl/
# cp -rf images/ /var/setting/lvgl/
# cd /var/setting/lvgl/
# ls
fonts    images
#
```

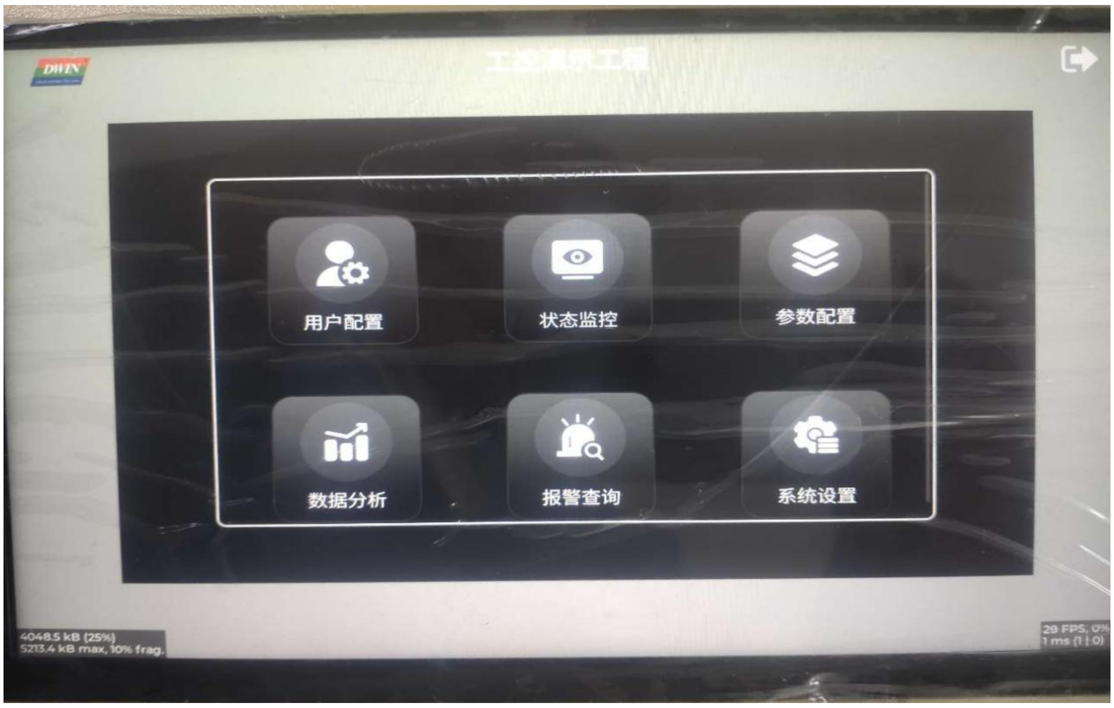
Copy the executable file to the ~ directory via a USB flash drive, and then test and run the program.

```
cd /mnt/usb/sda1/ (sda1 can be obtained through the Tab key.)
cp lvgl_demo ~
cd ~
./lvgl_demo
```



```
# cd /mnt/usb/sda1/
# cp lvgl_demo ~
# cd ~
# ls
lvgl_demo
# ./lvgl_demo
[User]  (0.000, +0)      main:
        lvgl_demo start main.c:1963
[User]  (0.000, +0)      main:
        sizes: char(1(-1:255)), short(2), int(4), long(4), size_t(4) main.c:1965
[Info]  (0.000, +0)      lv_init: begin lv_init.c:139
[User]  (0.000, +0)      main: lv_init done. main.c:1970
[Info]  (4293318.922, +4293318922)       lv_obj_create: begin lv_obj.c:100
[Info]  (4293318.922, +0)        lv_obj_create: begin lv_obj.c:100
[Info]  (4293318.922, +0)        lv_obj_create: begin lv_obj.c:100
[Info]  (4293318.922, +0)        lv_obj_create: begin lv_obj.c:100
[Info]  (4293318.922, +0)        lv_linux_fbdev_set_file: The framebuffer device was opened successfully lv_linux_fbdev.c:134
[Info]  (4293318.922, +0)        lv_linux_fbdev_set_file: 800x480, 32bpp lv_linux_fbdev.c:180
[Info]  (4293318.923, +1)        lv_linux_fbdev_set_file: The framebuffer device was mapped to memory successfully lv_linux_fbdev.c:195
```

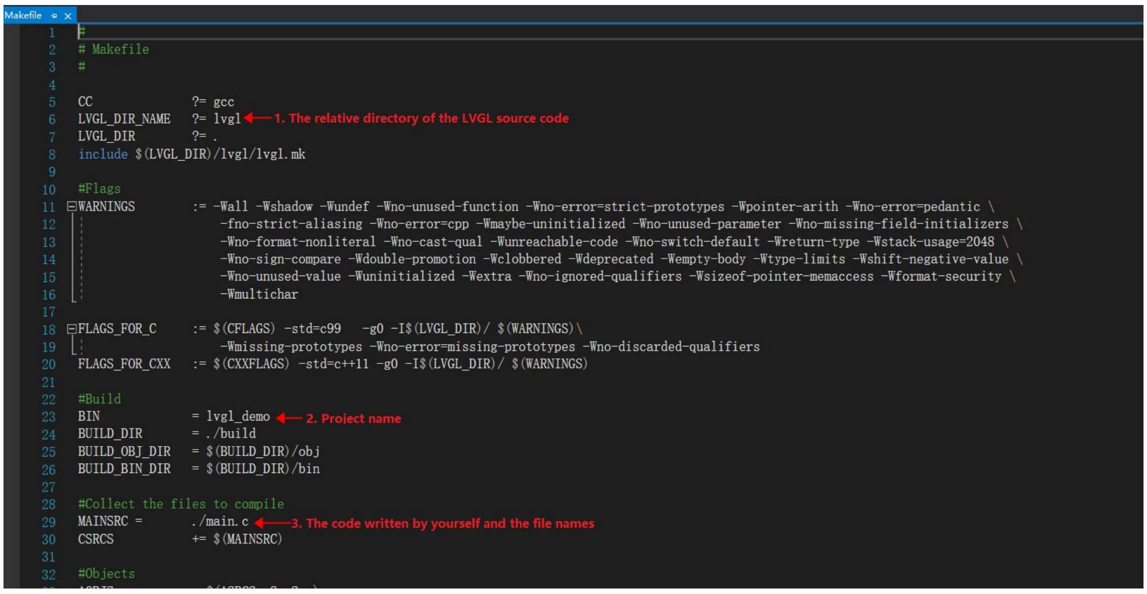
At this time, the DEMO has been installed on the screen.



4.4 Precautions

4.4.1 Makefile

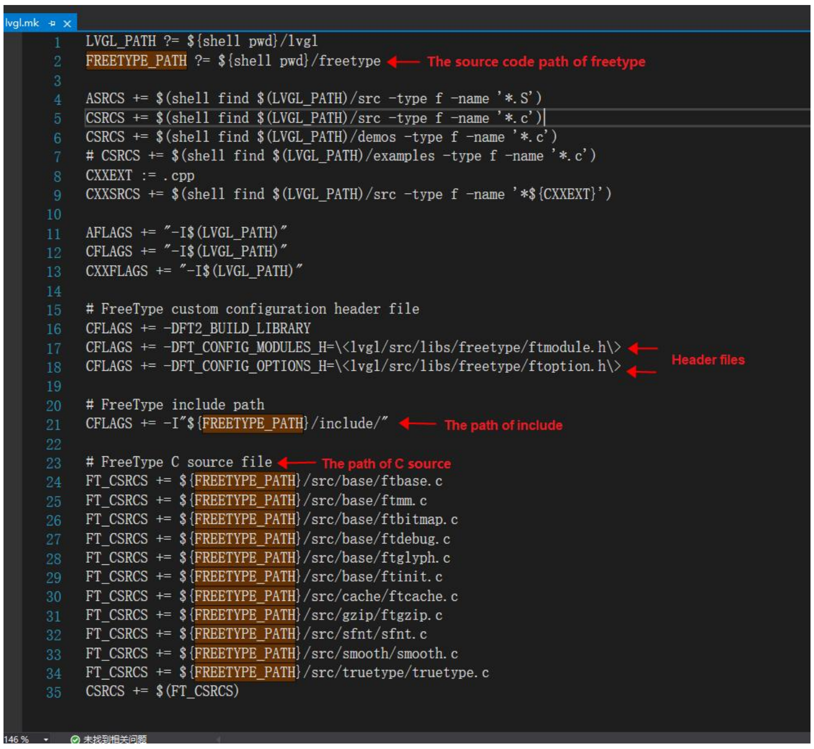
If you develop independently based on the DEMO, please modify the Makefile according to the specific situation of your own project.


Makefile
1. The relative directory of the LVGL source code
2. Project name
3. The code written by yourself and the file names

```
#
# Makefile
#

CC              ?= gcc
LVGL_DIR_NAME   ?= lvgl
LVGL_DIR        ?= .
include $(LVGL_DIR)/lvgl/lvgl.mk

#Flags
WARNINGS        := -Wall -Wshadow -Wundef -Wno-unused-function -Wno-error=strict-prototypes -Wpointer-arith -Wno-error=pedantic \
                    -fno-strict-aliasing -Wno-error=cpp -Wmaybe-uninitialized -Wno-unused-parameter -Wno-missing-field-initializers \
                    -Wno-format-nonliteral -Wno-cast-qual -Wunreachable-code -Wno-switch-default -Wreturn-type -Wstack-usage=2048 \
                    -Wno-sign-compare -Wdouble-promotion -Wclobbered -Wdeprecated -Wempty-body -Wtype-limits -Wshift-negative-value \
                    -Wno-unused-value -Wuninitialized -Wextra -Wno-ignored-qualifiers -Wsizeof-pointer-memaccess -Wformat-security \
                    -Wmultichar

FLAGS_FOR_C     := $(CFLAGS) -std=c99   -g0 -I$(LVGL_DIR)/ $(WARNINGS)\
                    -Wmissing-prototypes -Wno-error=missing-prototypes -Wno-discarded-qualifiers
FLAGS_FOR_CXX   := $(CXXFLAGS) -std=c++11 -g0 -I$(LVGL_DIR)/ $(WARNINGS)

#Build
BIN             = lvgl_demo
BUILD_DIR       = ./build
BUILD_OBJ_DIR   = $(BUILD_DIR)/obj
BUILD_BIN_DIR   = $(BUILD_DIR)/bin

#Collect the files to compile
MAINSRC =         ./main.c
CSRCS           += $(MAINSRC)

#Objects
```

4.4.2 Third-Party Library

If you need to load a third-party library (such as freetype in the sample project), open the lvgl.mk file in the lvgl directory and add the source code path, header files, etc.



```
LVGL_PATH ?= ${shell pwd}/lvgl
FREETYPE_PATH ?= ${shell pwd}/freetype

ASRCS += $(shell find $(LVGL_PATH)/src -type f -name '*.S')
CSRCS += $(shell find $(LVGL_PATH)/src -type f -name '*.c')
CSRCS += $(shell find $(LVGL_PATH)/demos -type f -name '*.c')
# CSRCS += $(shell find $(LVGL_PATH)/examples -type f -name '*.c')
CXXEXT := .cpp
CXXSRCS += $(shell find $(LVGL_PATH)/src -type f -name '*${CXXEXT}')

AFLAGS += "-I$(LVGL_PATH)"
CFLAGS += "-I$(LVGL_PATH)"
CXXFLAGS += "-I$(LVGL_PATH)"

# FreeType custom configuration header file
CFLAGS += -DFT2_BUILD_LIBRARY
CFLAGS += -DFT_CONFIG_MODULES_H=\<lvgl/src/libs/freetype/ftmodule.h\>
CFLAGS += -DFT_CONFIG_OPTIONS_H=\<lvgl/src/libs/freetype/ftoption.h\>

# FreeType include path
CFLAGS += -I"${FREETYPE_PATH}/include/"

# FreeType C source file
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftbase.c
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftmm.c
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftbitmap.c
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftdebug.c
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftglyph.c
FT_CSRCS += ${FREETYPE_PATH}/src/base/ftinit.c
FT_CSRCS += ${FREETYPE_PATH}/src/cache/ftcache.c
FT_CSRCS += ${FREETYPE_PATH}/src/gzip/ftgzip.c
FT_CSRCS += ${FREETYPE_PATH}/src/sfnt/sfnt.c
FT_CSRCS += ${FREETYPE_PATH}/src/smooth/smooth.c
FT_CSRCS += ${FREETYPE_PATH}/src/truetype/truetype.c
CSRCS += $(FT_CSRCS)
```

5 Revision Records

Rev	Revise Date	Content	Editor
00	2022-10-11	First Edition	Yu Yihe
01	2023-12-1	Update the boot startup logo	Chen Yan
02	2024-3-20	Added examples about brightness adjustment, system time settings and buzzer switch.	Chen Yan
03	2025-03-20	Add the configuration of cross-compilation between Chapter 1 and Qt Creator. Add the cross-compilation configuration for LVGL.	Chen Xian

Please contact us if you have any questions about the use of this document or our products, or if you would like to know the latest information about our products:

Customer service Tel: +86-400-018-9008

Customer service email: dwinhmi@dwin.com.cn

Website: www.dwin-global.com

DWIN Developer Forum: https://forums.dwin-global.com/index.php/forums

Thank you all for continuous support of DWIN, and your approval is the driving force of our progress!