# DWIN Linux Screen Development Guide (40 Series & 40ZOS-1 Series)

# Contents

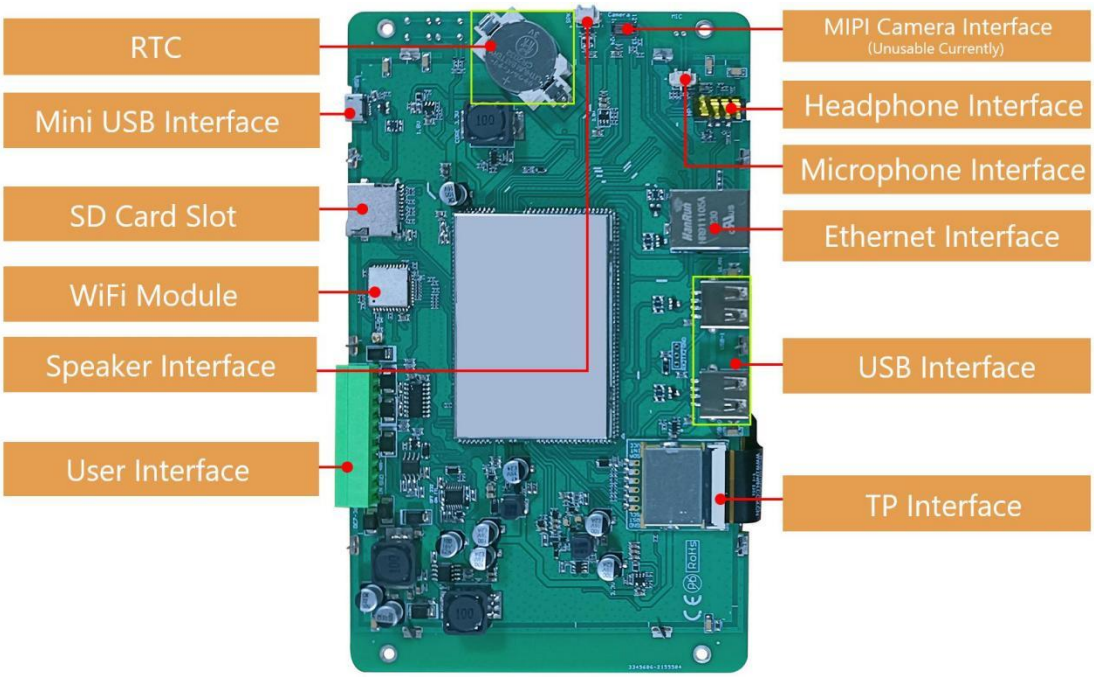
# 1 Product Introduction

## 1.1 Product Feature

DWIN Linux screen 40 series:
CPU：RK3566, Quad-core ARM Cortex-A55, 1.8GHz
RAM：2GB LPDDR4
Flash: 8GB EMMC5.0
Linux Version：Linux 4.19

Debian version (module suffix is ZOS-1)


(DMG12800T070_40WTC front)




（DMG12800T070_40WTC back)

### 1.1.1 Development Method

QT and LVGL options.

### 1.1.2 Documentation

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

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

Tutorial on YouTube:
https://youtube.com/playlist?list=PLKfWyFPPaoDr3Vq98orVxJqKA5MDaliN&si=BVVDmdfCopcH--nK

### 1.1.3 Shipping List (for reference)

- screen ×1 piece
- antenna × 1 piece

### 1.1.4 Optional Accessories

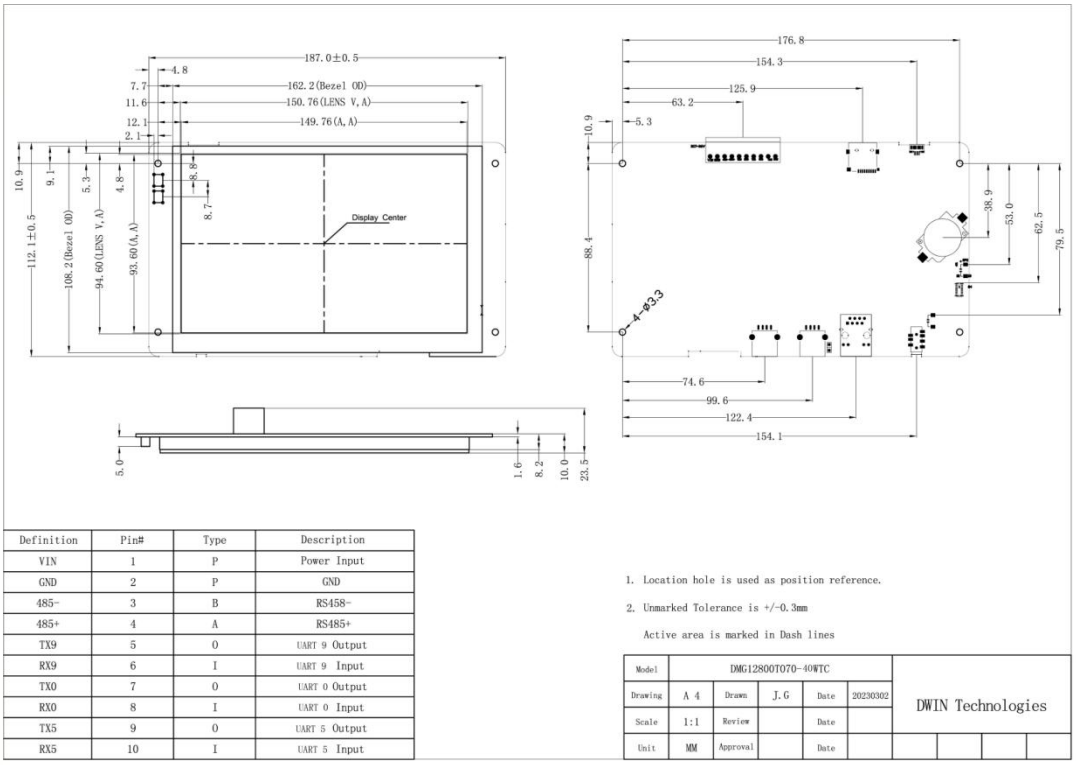
- **Speaker**
  DWIN material code B01851, cable length 180 mm, with socket 2PIN_1.25，88±3dB， 8Ω，0.8W
- **SD card**
- **4G module**
  China and India: LUAT Air780EI
  Europe: QUECTEL EC200A-EU
  Australia: QUECTEL EC200A-AU
- **Camera**
  Support camera with USB interface

## 1.2 Wiring

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

**Peripherals and Interfaces**

| Properties | Parameters | Description |
|---|---|---|
| COM | 2-way RS232 | UART5 & UART9 |
| | 1-way RS485 | UART8 |
| | 1-way TTL/COMS | UART0 |




| Definition | Pin# | Type | Description |
|---|---|---|---|
| VIN | 1 | P | Power Input |
| GND | 2 | P | GND |
| 485- | 3 | B | RS458- |
| 485+ | 4 | A | RS485+ |
| TX9 | 5 | O | UART 9 Output |
| RX9 | 6 | I | UART 9 Input |
| TX0 | 7 | O | UART 0 Output |
| RX0 | 8 | I | UART 0 Input |
| TX5 | 9 | O | UART 5 Output |
| RX5 | 10 | I | UART 5 Input |

1. Location hole is used as position reference.
2. Unmarked Tolerance is +/-0.3mm
   Active area is marked in Dash lines

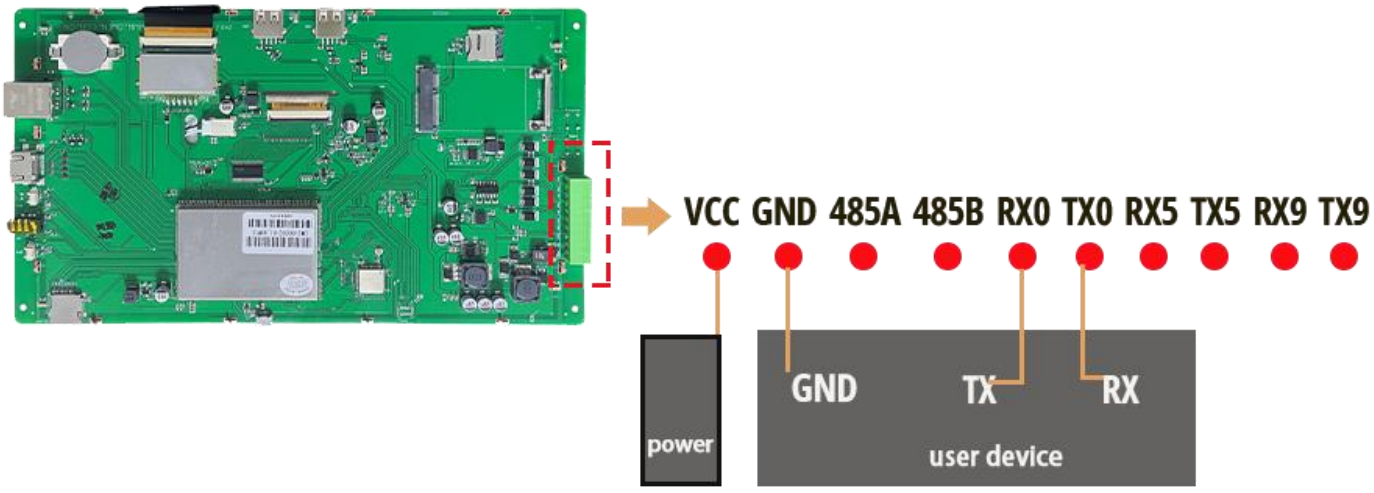
| | | | | | | |
|---|---|---|---|---|---|---|
| Model | DMG12800T070-40WTC | | | | | DWIN Technologies |
| Drawing | A 4 | Drawn | J.G | Date | 20230302 | |
| Scale | 1:1 | Review | | Date | | |
| Unit | MM | Approval | | Date | | |

## 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.

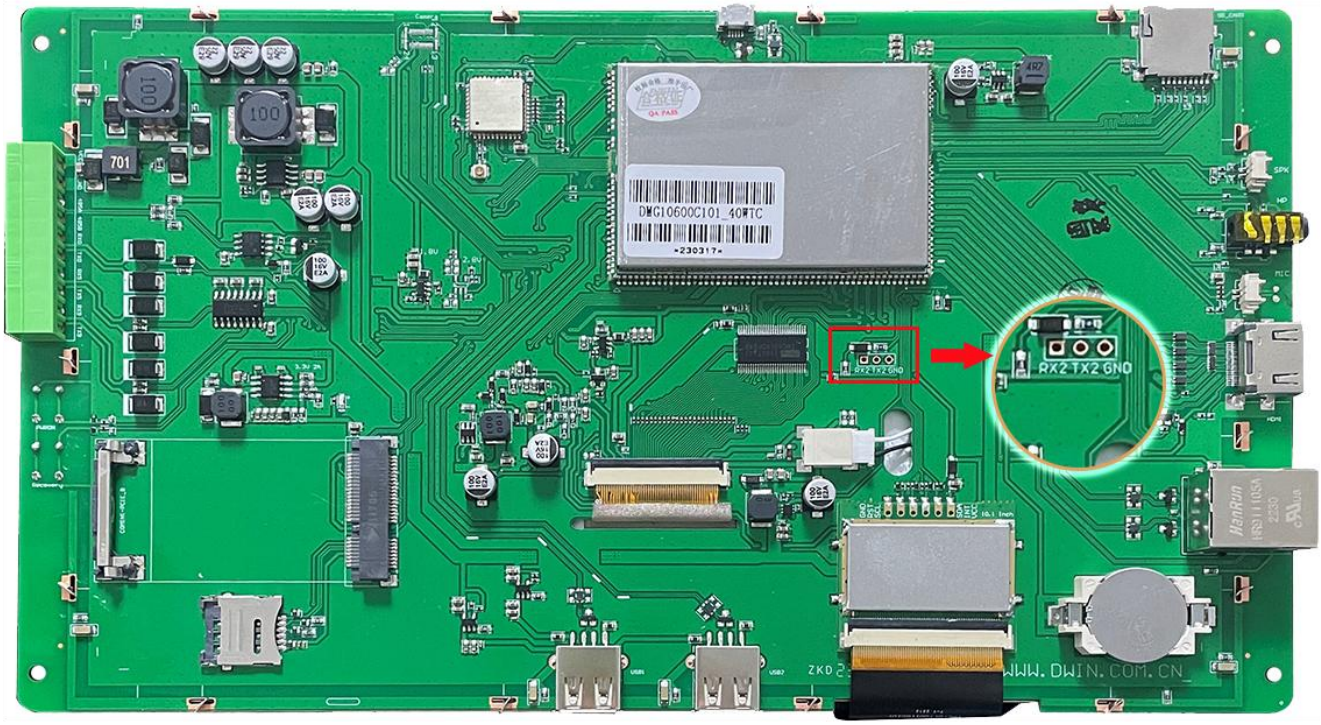
RXD, Receive, connect to TX pin of the user device.



### 1.2.2 Serial Parameter Setting

Regarding baud rate, UART 2 is 1500000, else are 115200.

For programming purposes, serial ports are identified by names that follow the pattern ttyS, such as ttyS0 or ttyS1.




### 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 Setup

## 2.1 Ubuntu16.04 Configuration

### 2.1.1 Introduction

This chapter will introduce the installation of a virtual machine and the configuration of Ubuntu16.04. If you have already installed Ubuntu16.04, you can click here.

### 2.1.2 Environment Requirements

CPU: no specific requirement Memory: generally over 2G.

Host machine OS: Windows XP, Windows 7 and above.

Software version: you can choose VMware workstation 10 and above for Windows according to your needs, it is not recommended to use previous versions.

**Note:**

**This example will use VMware Workstation 15 Pro for installation demonstration.**

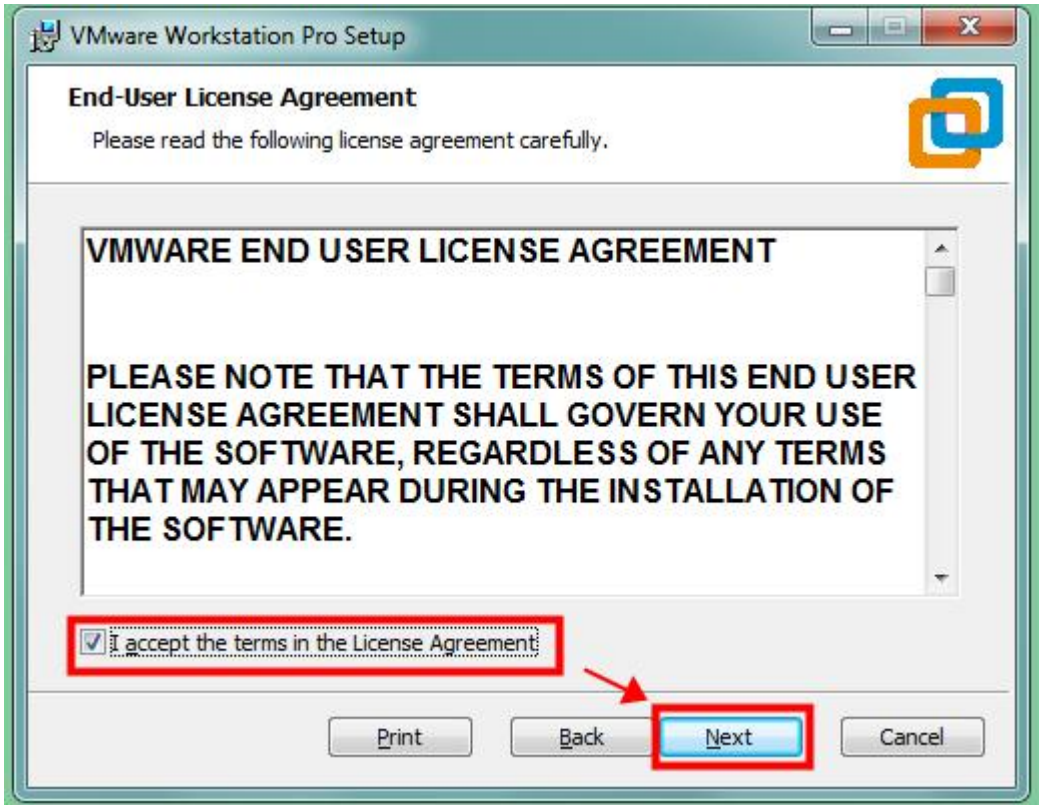
### 2.1.3 VMware Workstation Installation

（1）Download VMware Workstation pro installation package on the official website below.

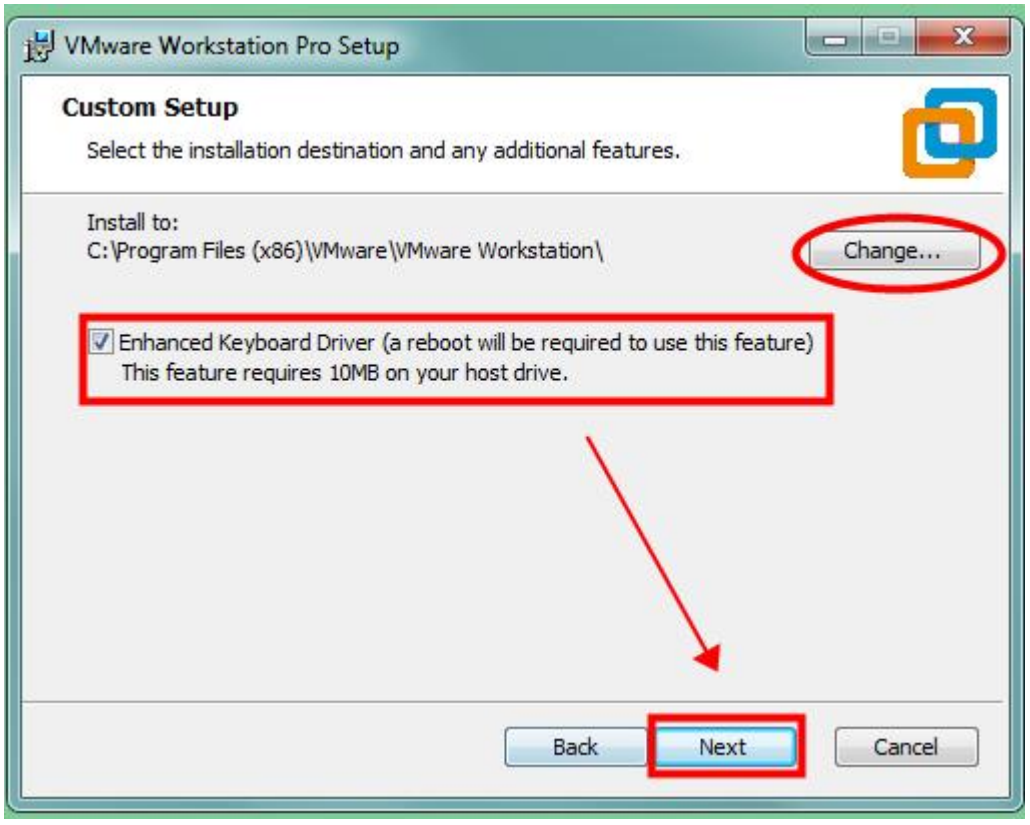
https://www.vmware.com/products/workstation-pro.html


（2）Double clicks on downloaded exe file to start the installation and click “**Next**”.

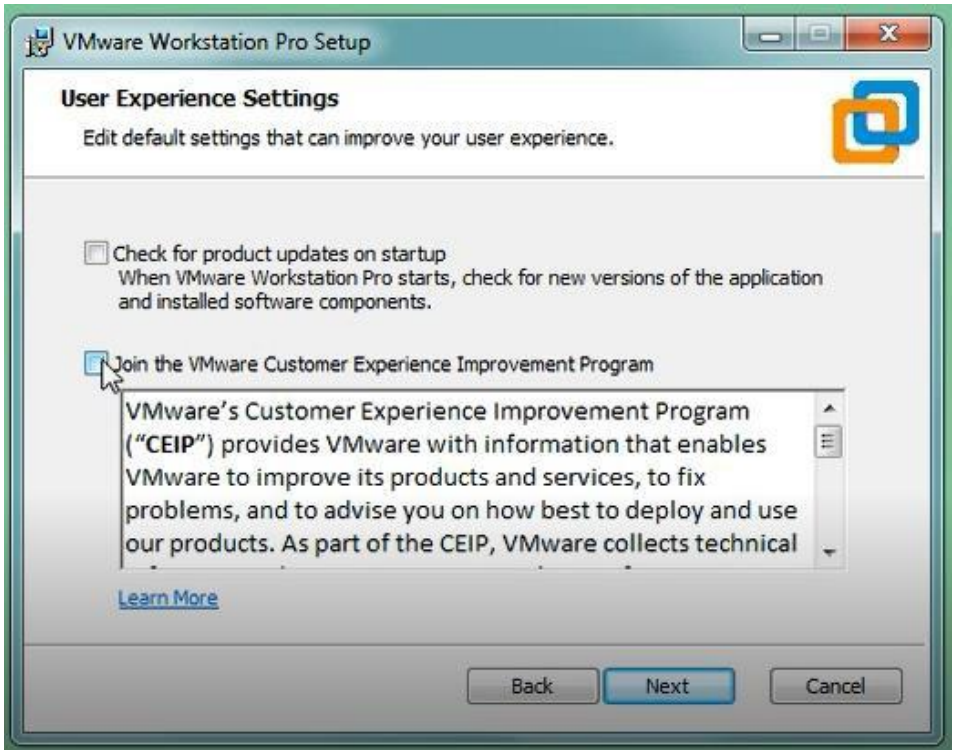
（3）Select “I accept the terms in the License Agreement” and click “Next”.



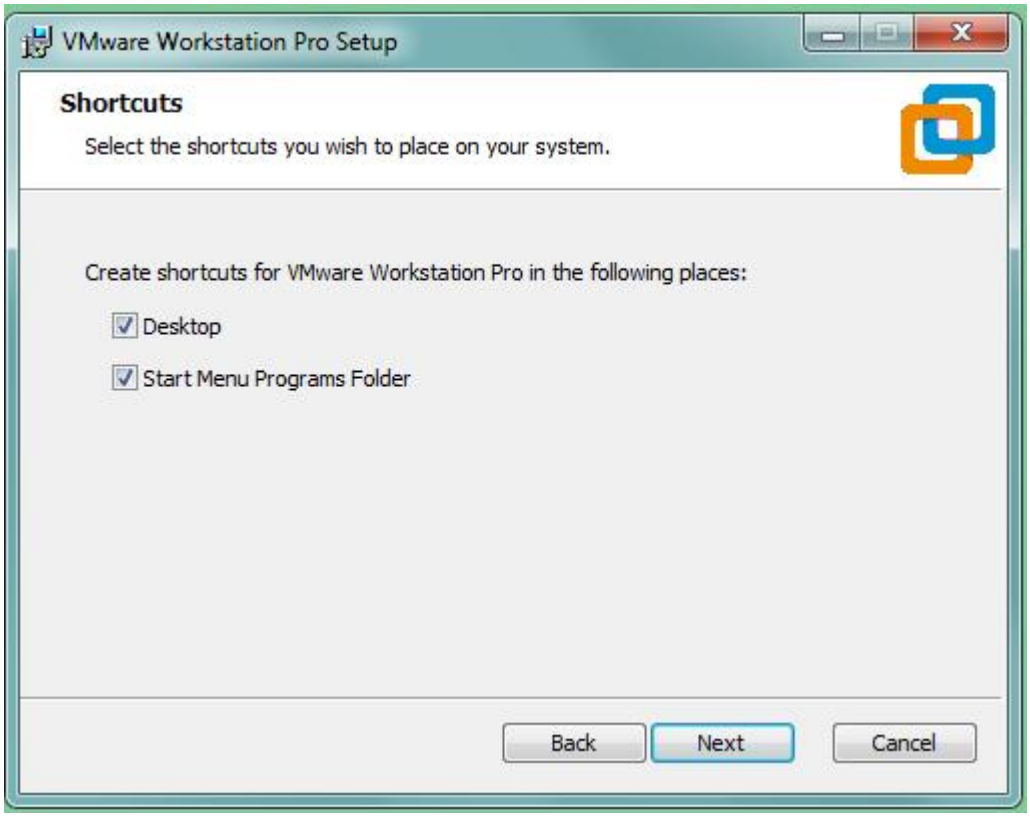

（4）Select the installation destination. Click "**Change**" if you want to install on another destination. Select "**Enhanced Keyboard**..." and then click "**Next**".



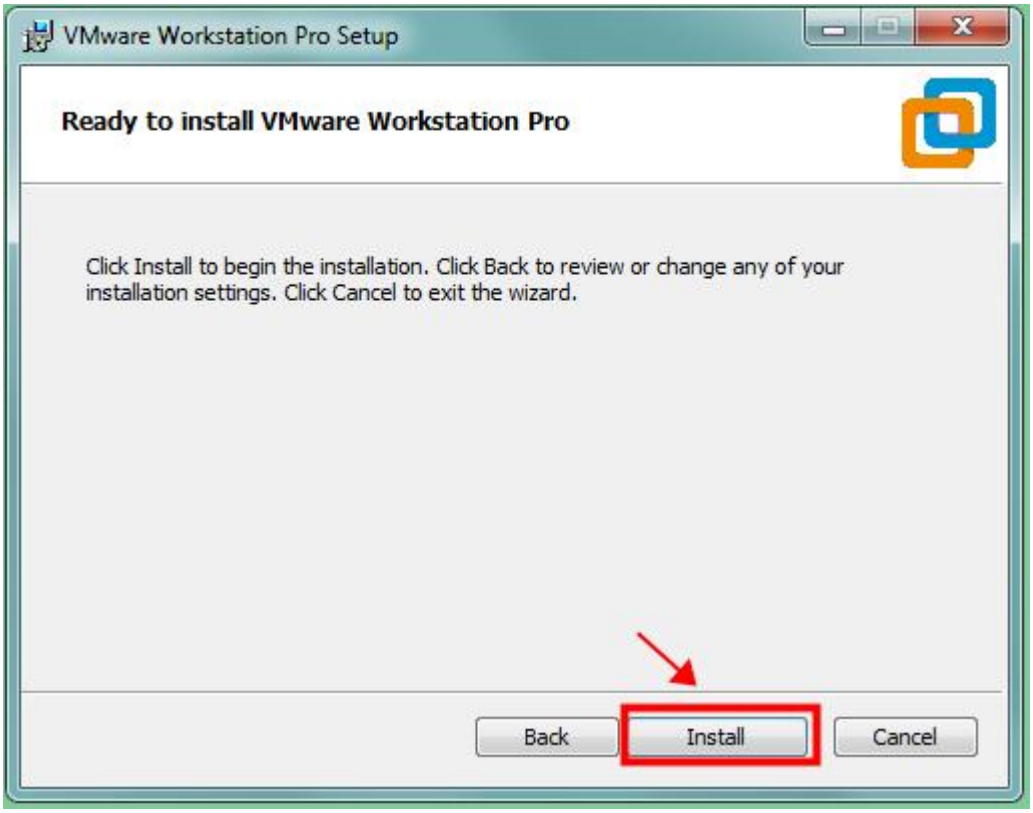
（5）Select “Check for product updates on startup” and “Join the VMware Customer Experience Improvement Program” based on what you need. Then click “Next”.




（6）Select the shortcuts you wish to place on your system. It's recommended to select both. Click "**Next**".




（7）Click "**Install**" to start the installation.

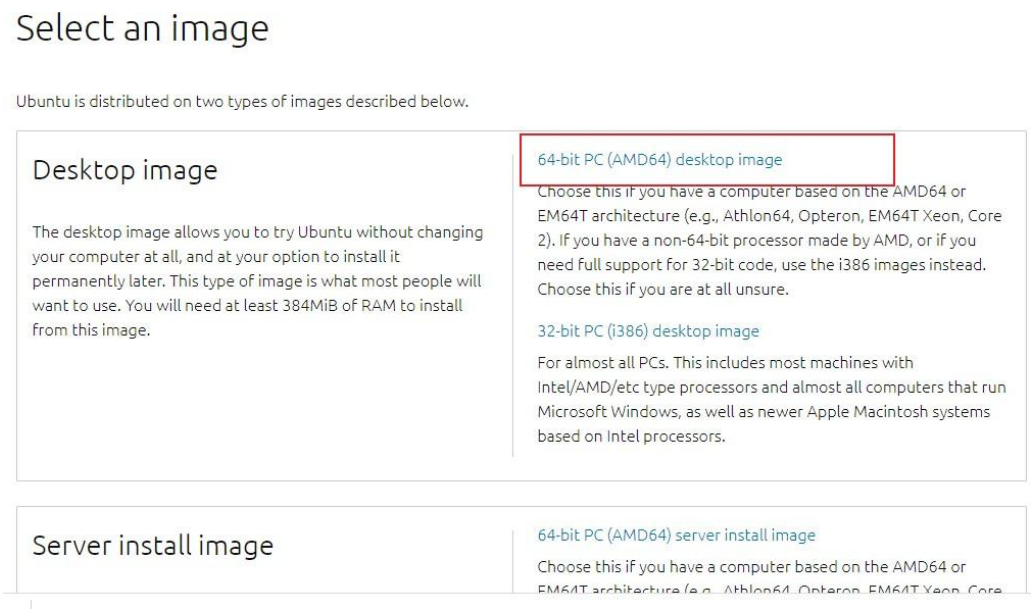



(8) The installation is completed. Click "**Finish**" to exit the Setup Wizard.


## 2.1.4 Download Ubuntu

（1）Download Ubuntu 16.04 from the official website below. https://releases.ubuntu.com/16.04/

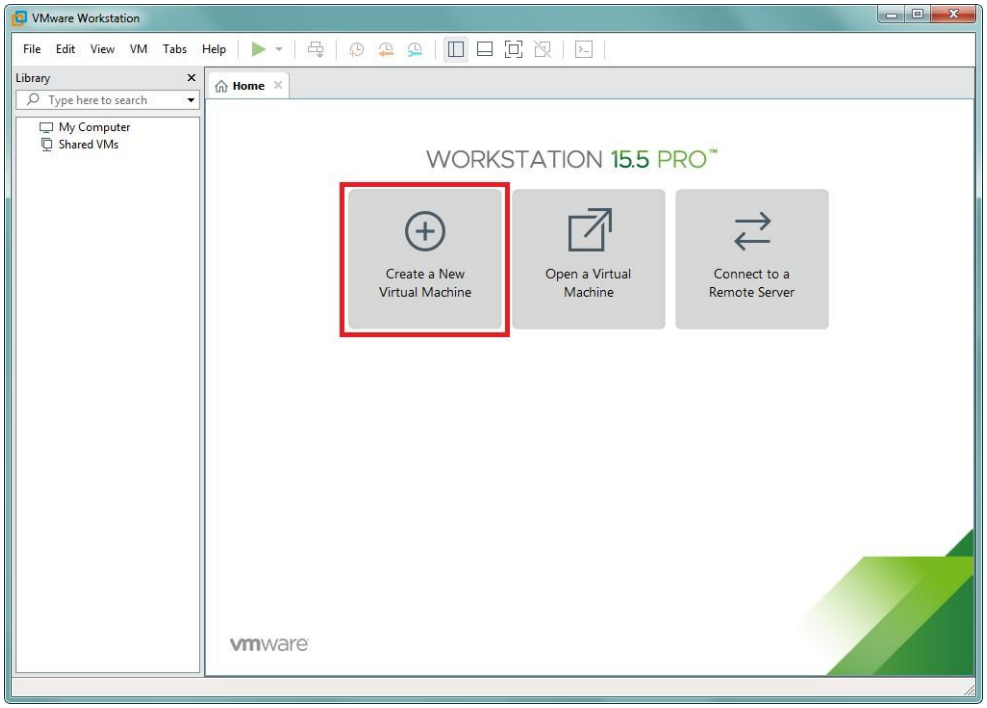
（2）Select “64-bit PC(AMD64) desktop image” to download “ubuntu-16.04.7-desktop-amd64.iso”.



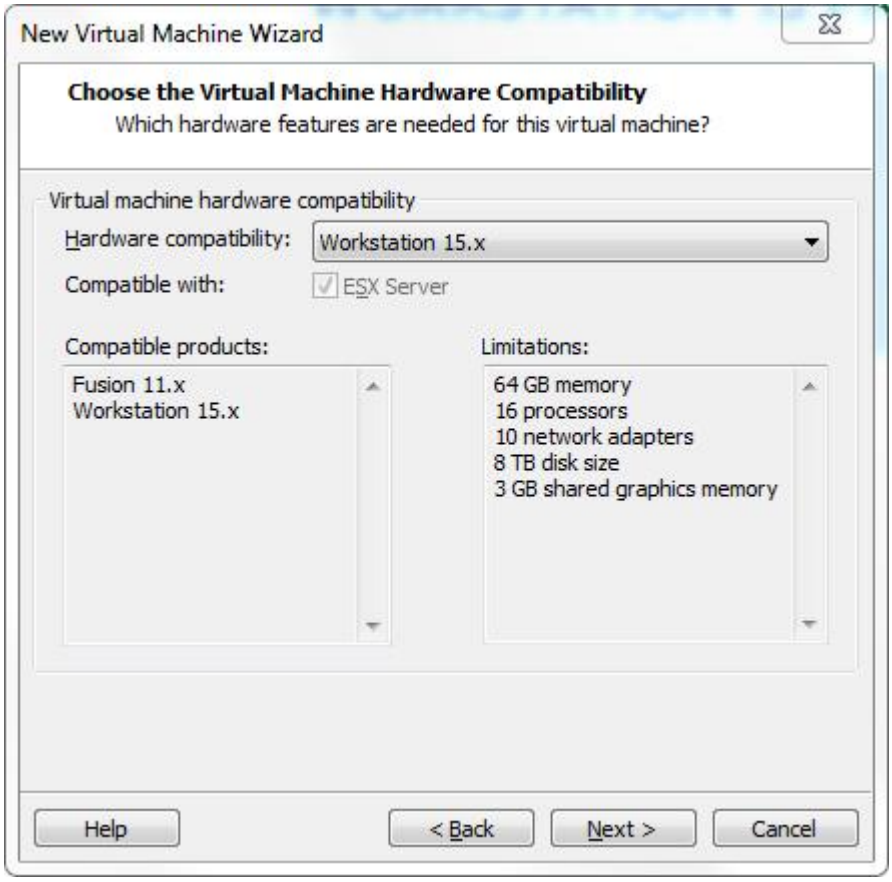

## 2.1.5 Install Ubuntu

（1）Open VMware Workstation Pro.


（2）Click “Create a new virtual machine”.

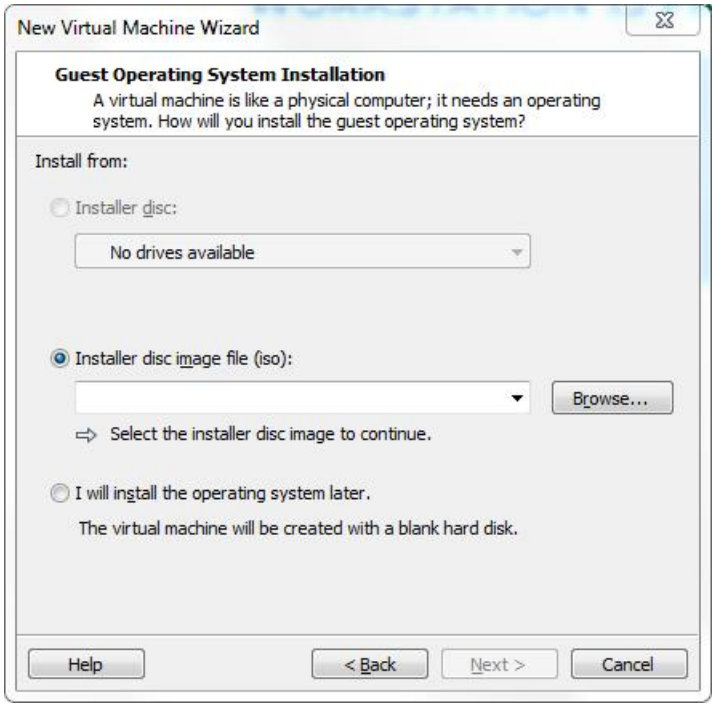


（3）Select “**Custom (advanced)**” and click “**Next**”.

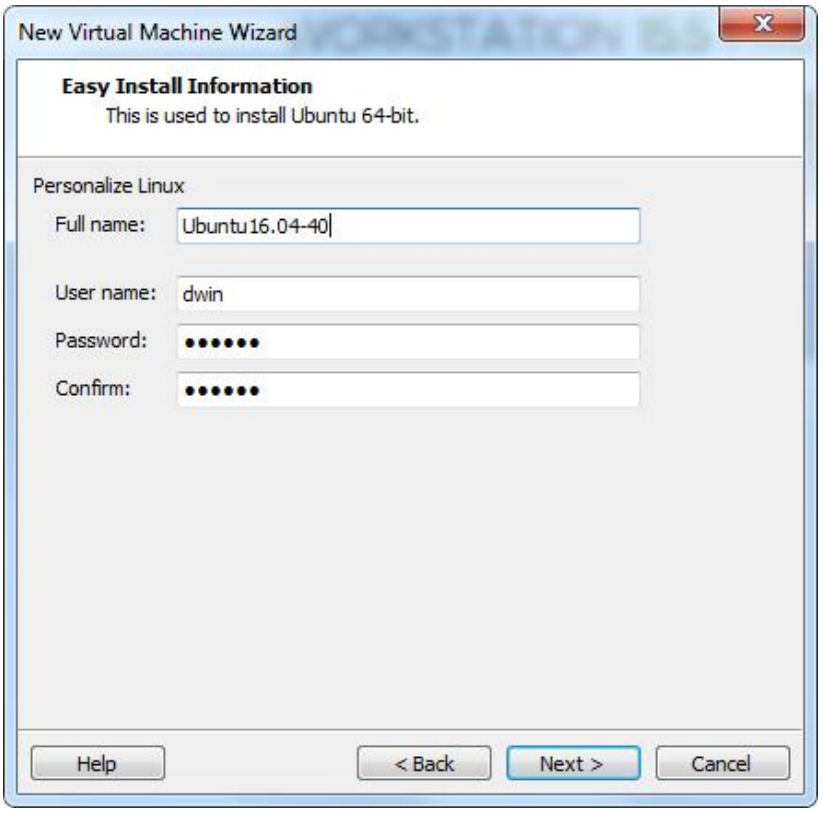




（5）Click “**Next**”

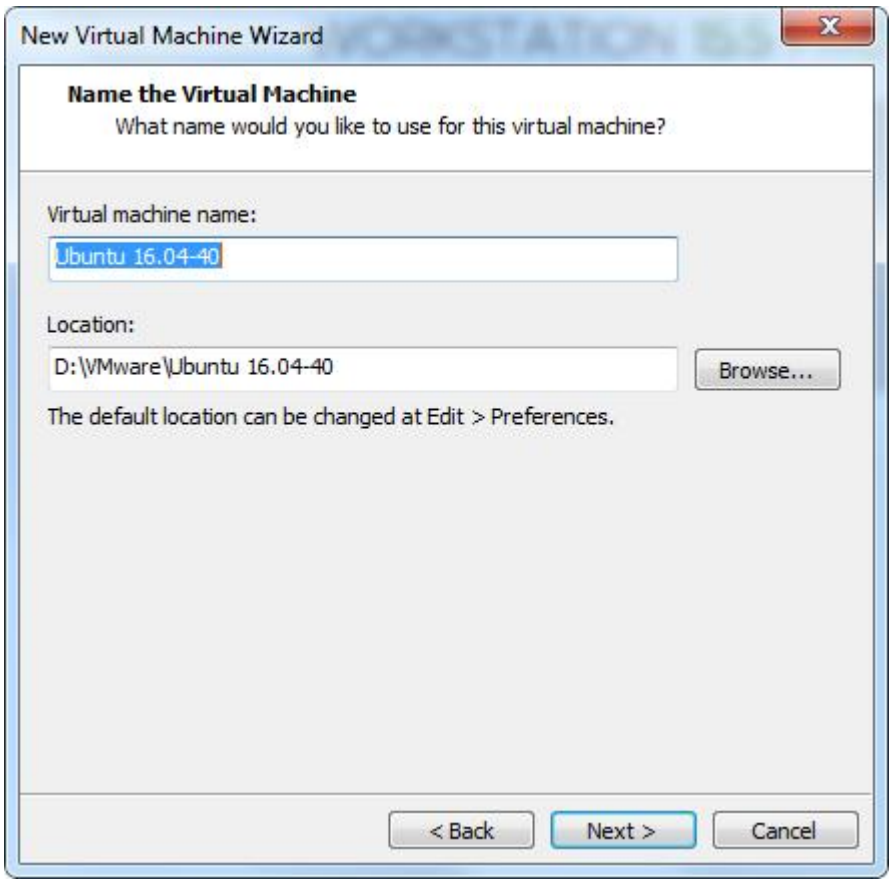
（6）Select “**Installer disc image file (iso)**”, next click “**Browse**…”, and select the downloaded Ubuntu ***.iso file. This installer will automatically recognize and read file. Click“**Next**”.



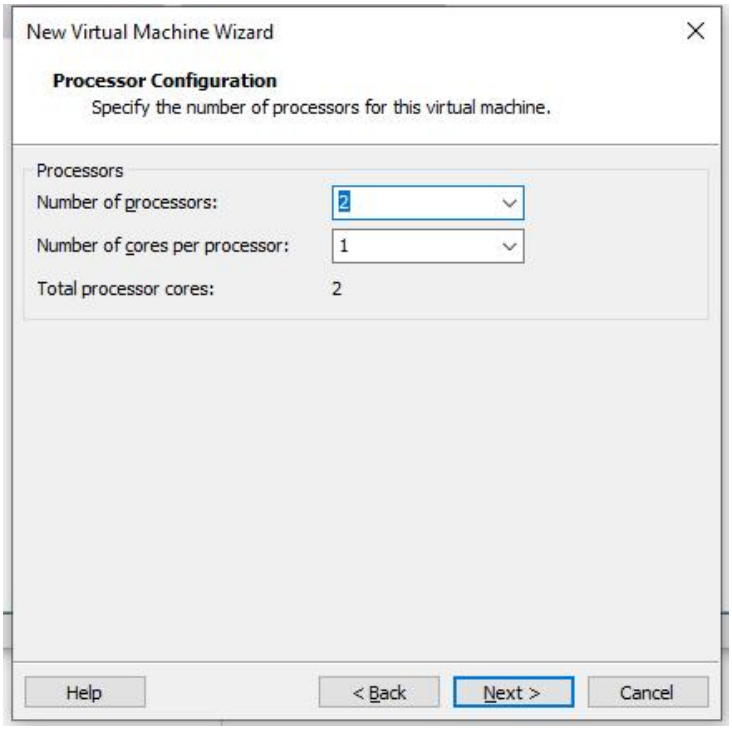

（7）Enter the custom name and password. The password is the Ubuntu login password and sudo privilege password. Click “**Next**”.



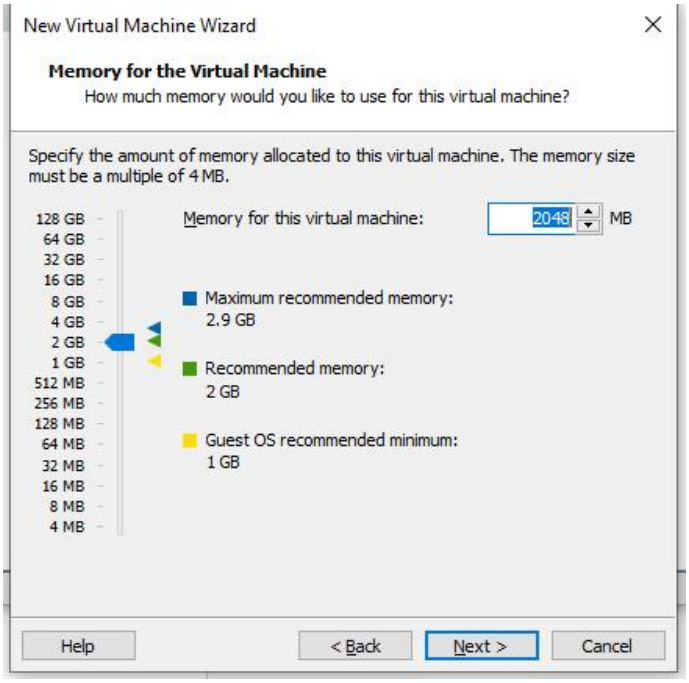

（8）Set the Ubuntu name and location, and click “**Next**”.



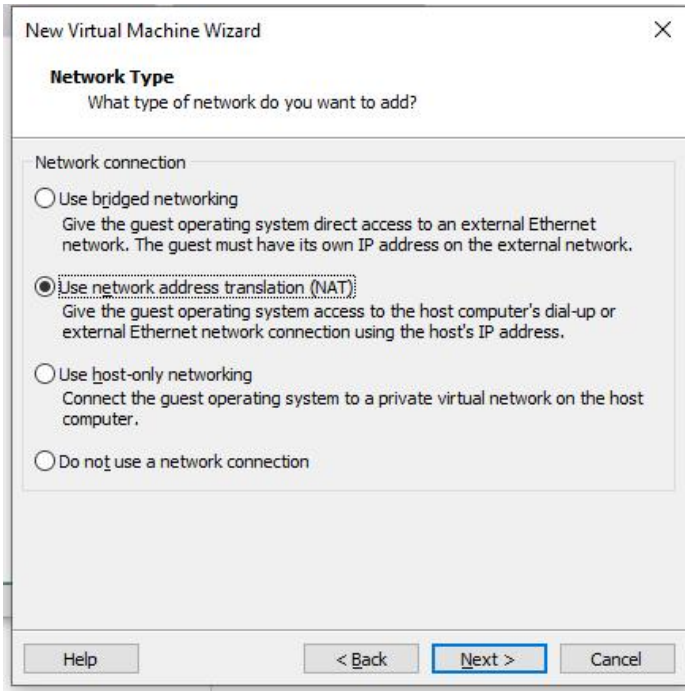

（8）According to needs and the computer configuration, allocate processors and cores (here the total number of processor cores is set to 2). Then click "**Next**".



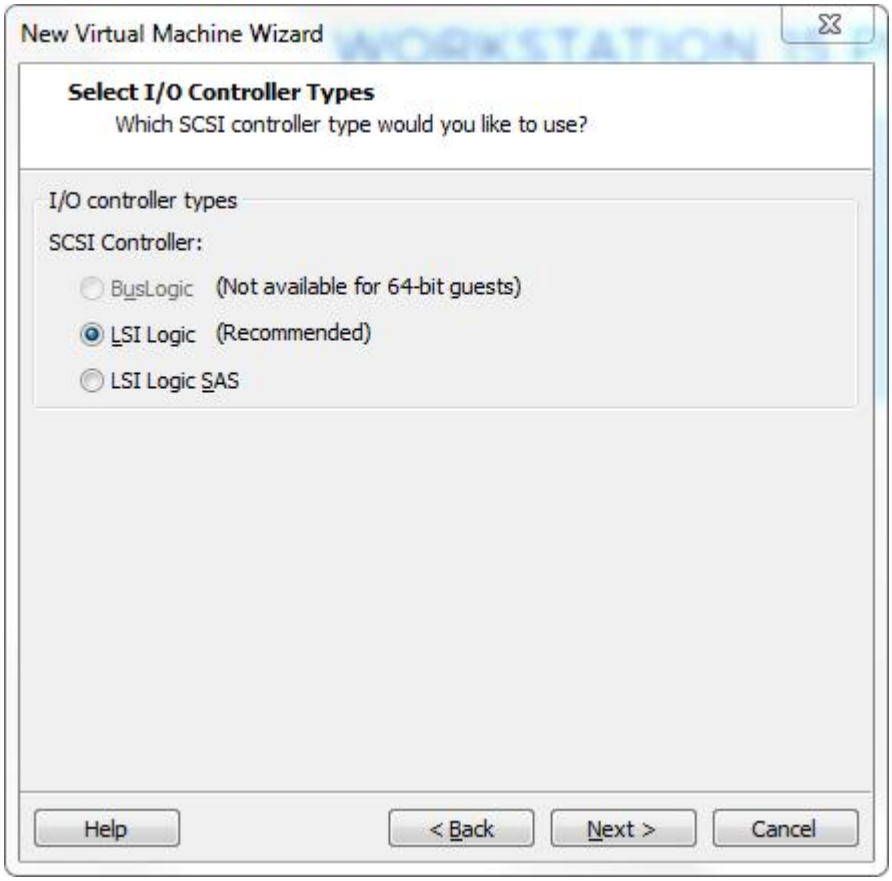

（9）The default operating memory is 2G (enough and changeable), click "**Next**".

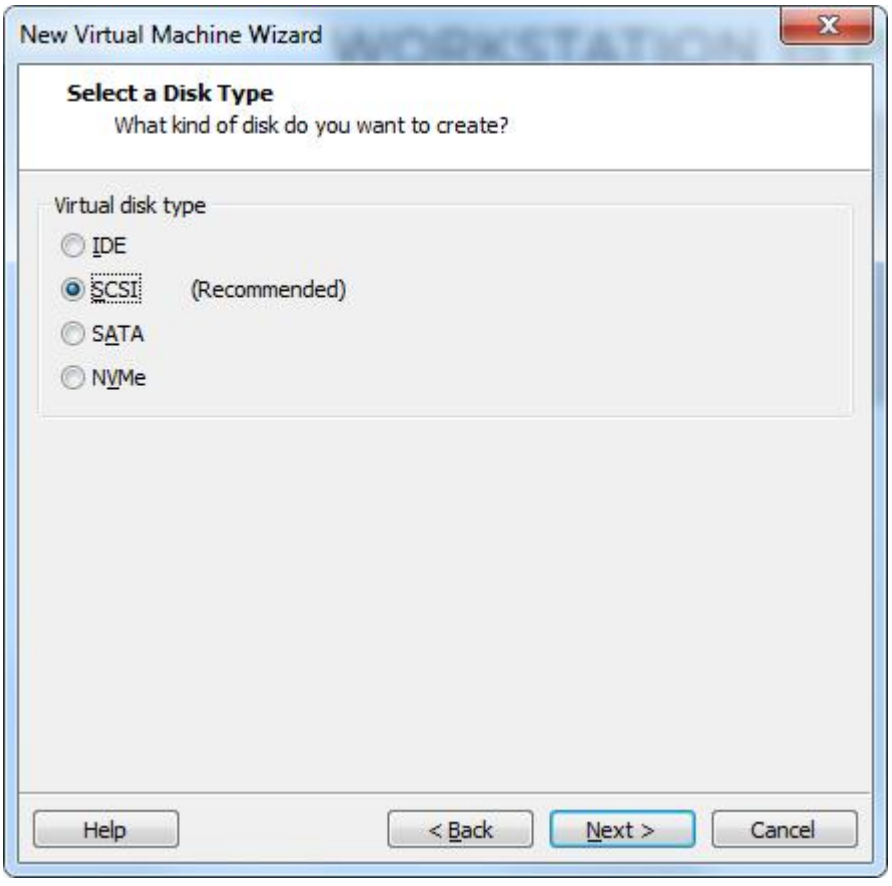



（10）Keep default configuration (or choose bridge network for tftp transfer). Click “Next”.



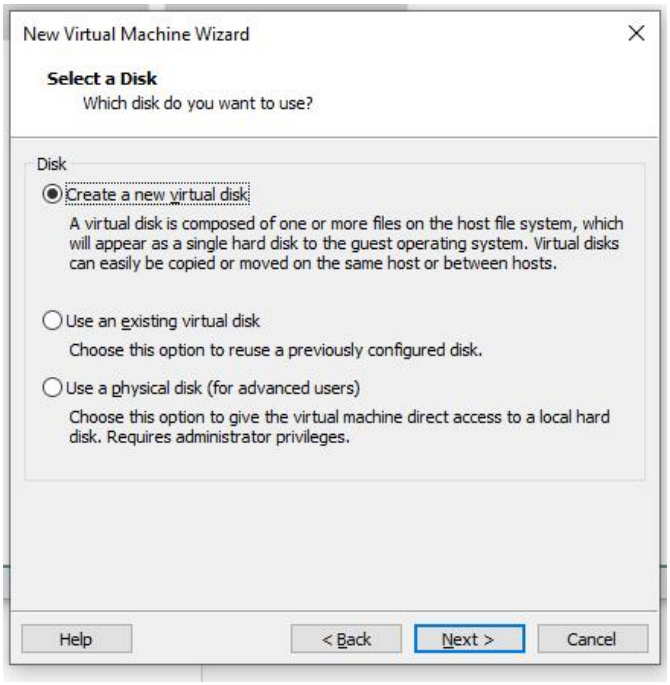

（11）Keep default and click “**Next**”.



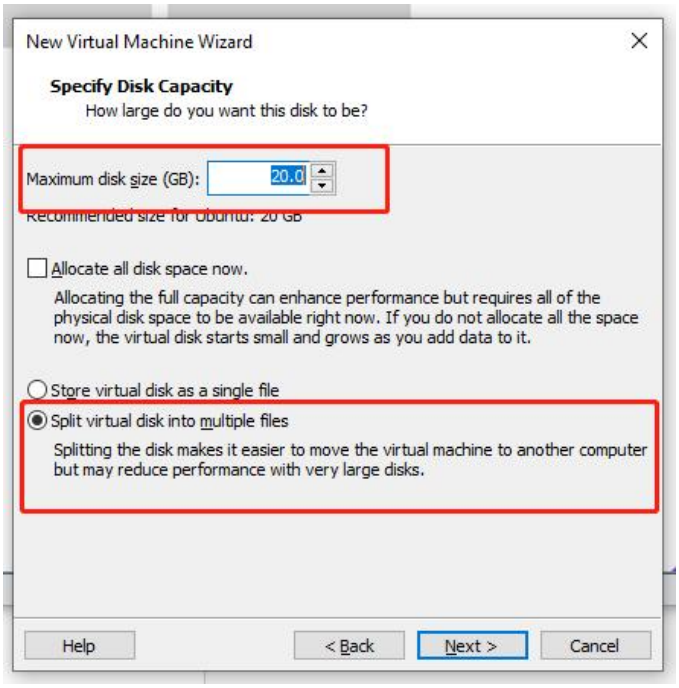




（12）Keep default and click "**Next**".

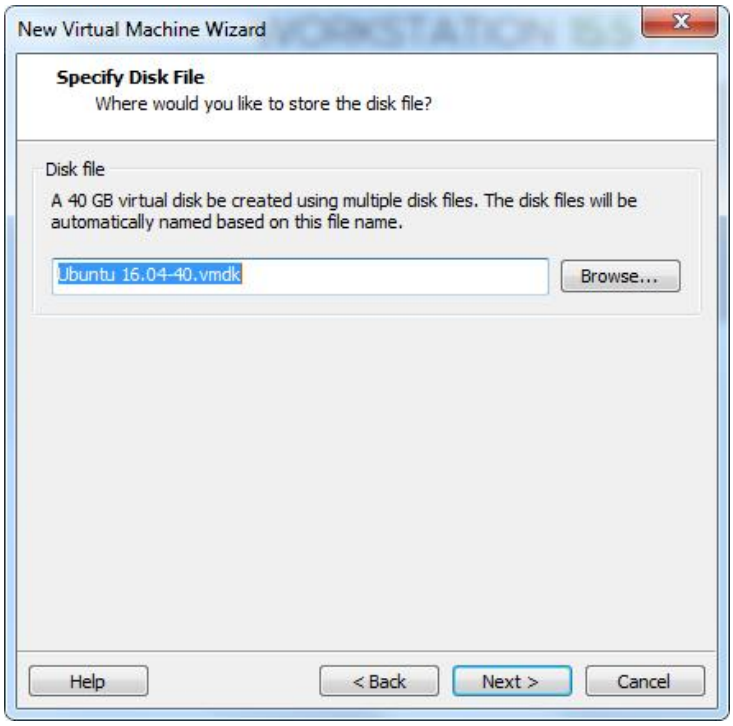
（13）Select “**Create a new virtual disk**” and click “**Next**”.



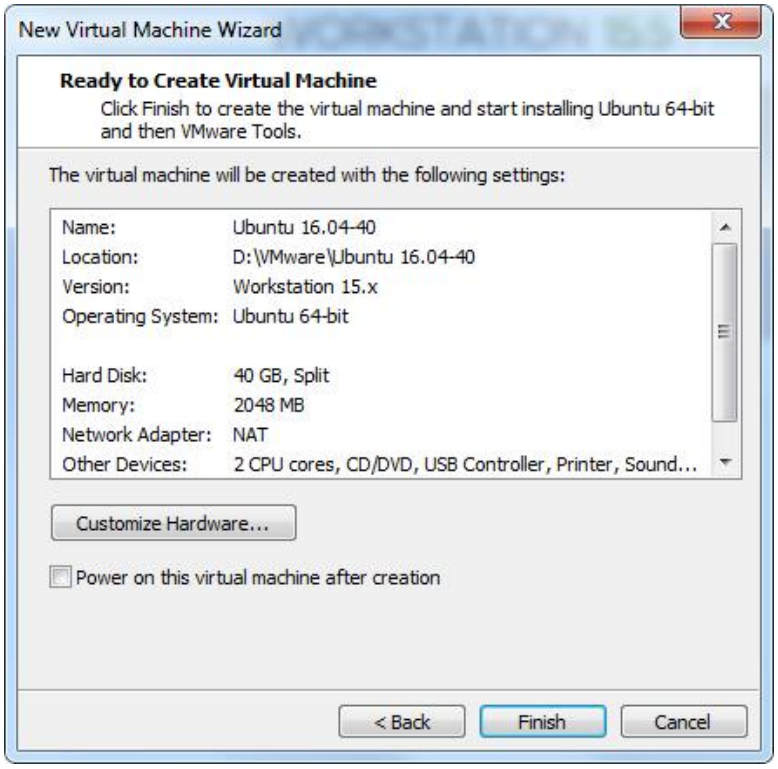

（14）Specify the disk capacity. If there is enough computer memory, it is recommended to set 30G or more because small memory may not be able to meet the subsequent demand. Select “**Split virtual disk into multiple files**”. Click “**Next**”. If the disk capacity is small, you can expand it (see subsequent sections).




（15）The disk will be named automatically. Keep the default and click “**Next**”.

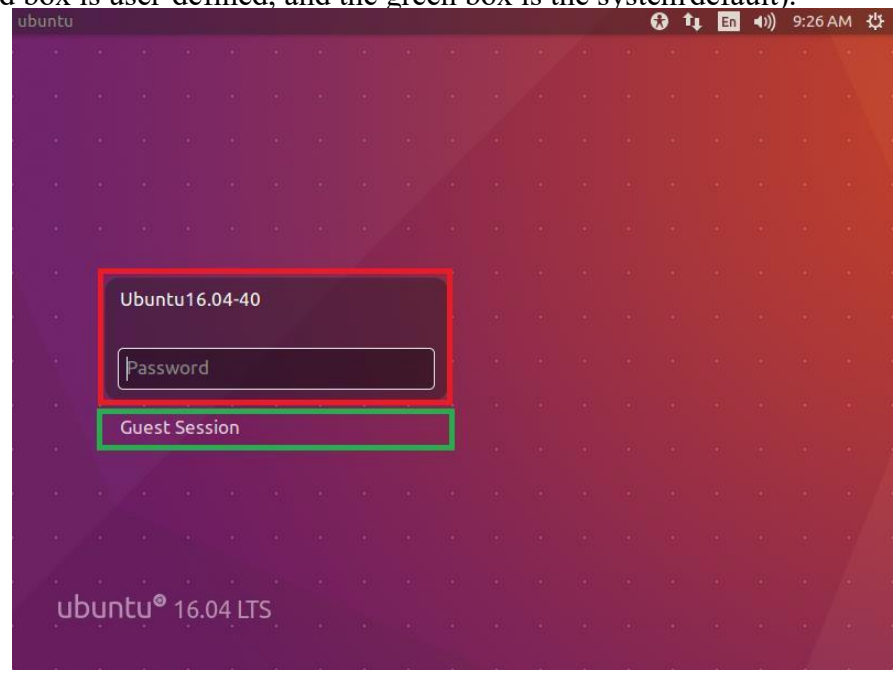



（16）Click “**Finish**” and the virtual machine will be opened and installed.




（17）Wait for a while.

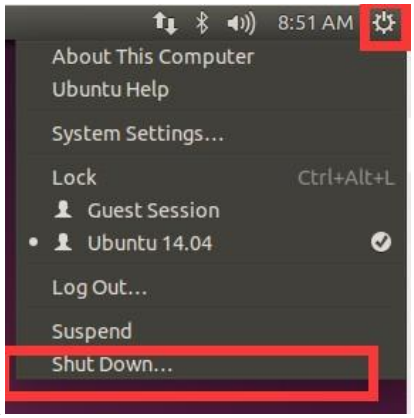

（18）When this page appears, the installation of Ubuntu is complete (Note: The login interface has two user login entries. The red box is user-defined, and the green box is the system default).



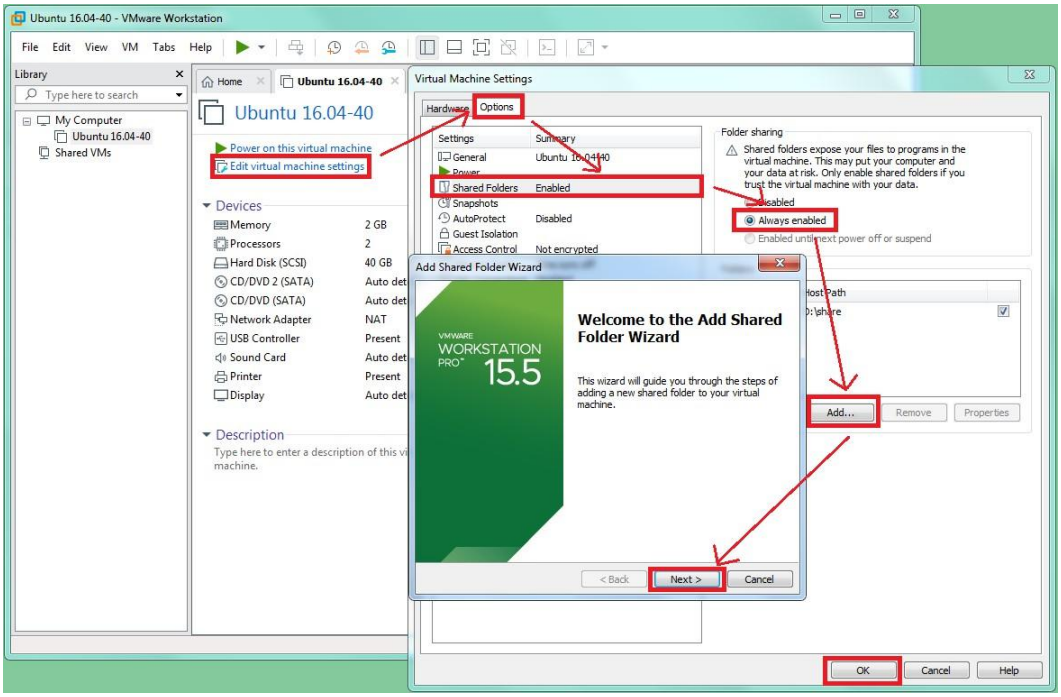

（19）Next, we'll start configuring some of the required settings for Ubuntu.

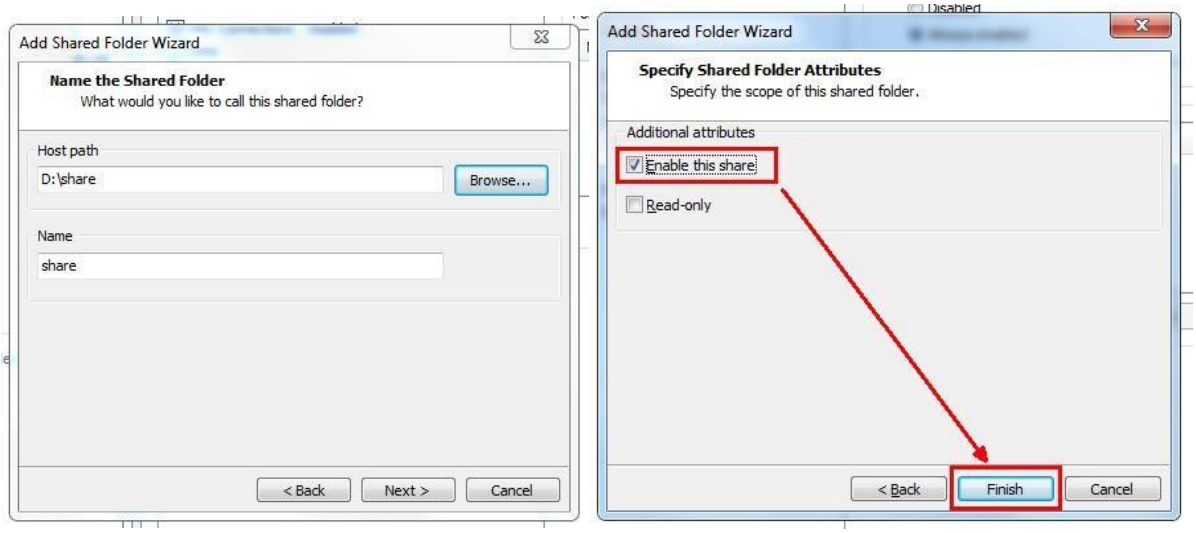
## 2.1.6 Shared Folder Setting

(1) Shut down Ubuntu.

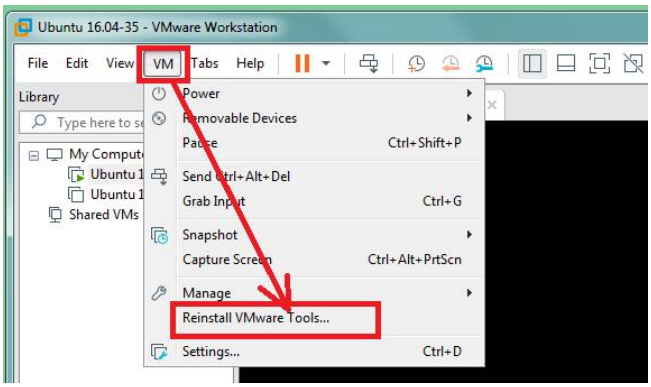



(2) After shutdown, click “Edit Virtual Machine Settings” -> “Options” -> “Shared Folder” -> “Always Enable” -> “Add”, to add a folder as a medium for file transfer between the host and the virtual machine. click “Next” and follow the Add Shared Folder Wizard. Finally click “OK”.

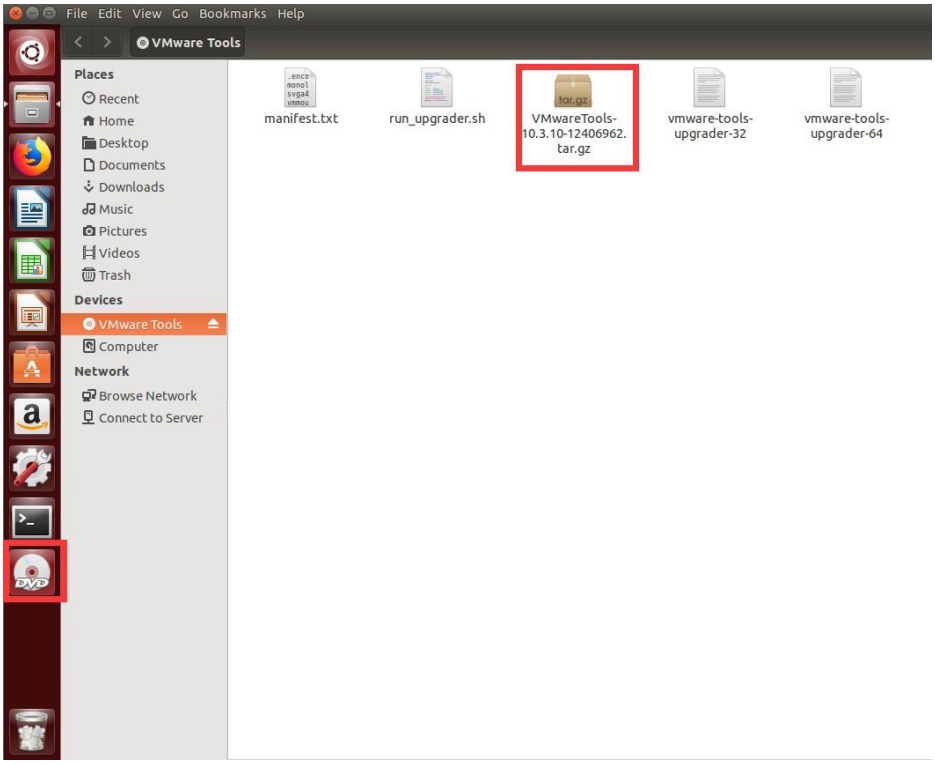


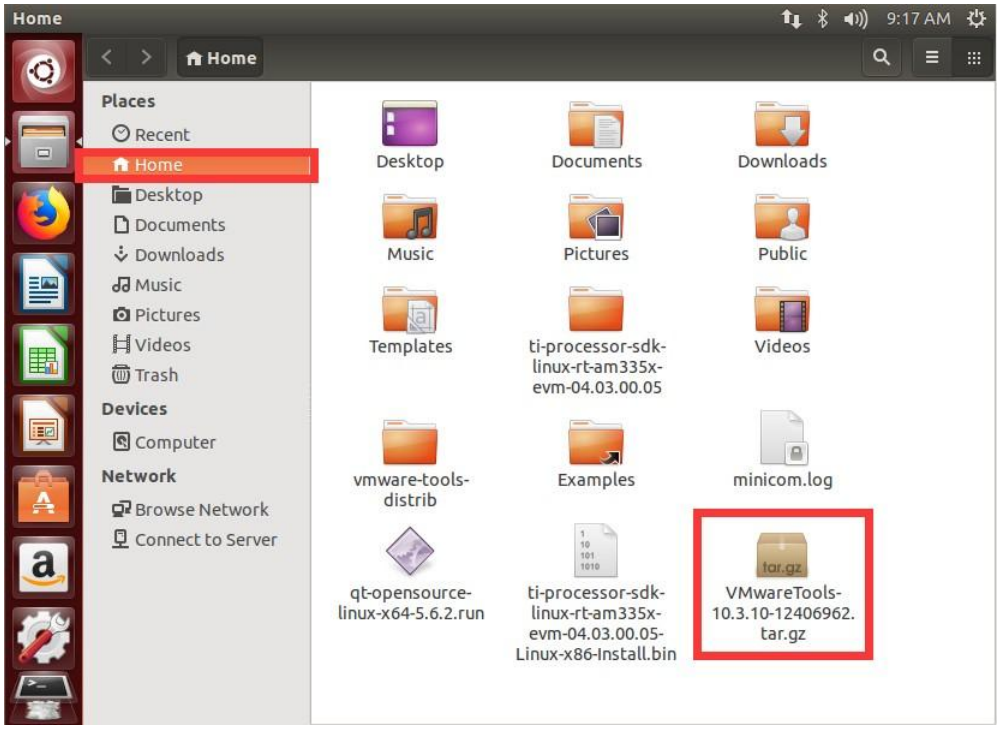
(3) Click "**Power on this virtual machine**" to start the virtual machine. Click "**VM**" -> "**Install VMware Tools**" (Note: The 'Install VMware Tools' is only selectable after powering on. In the example, VMware Tools have already been installed, so it shows 'Reinstall VMware Tools').




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




(5) Right click the tar file and click "**copy to**" a path with permission, e.g., to "**Home**".

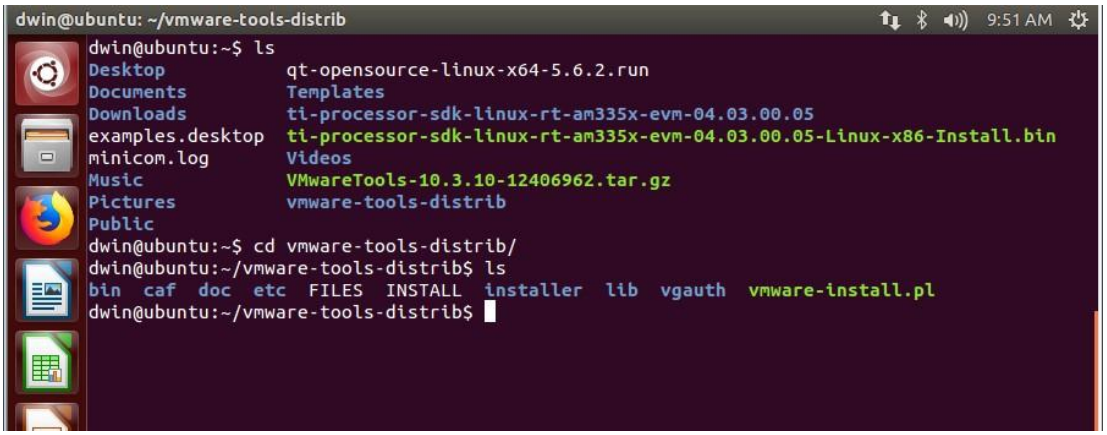



(6) At this point, we need to open the terminal as shown. Click the upper left icon and enter "**Terminal**" and click the "**Terminal**" icon (the terminal can be locked in the taskbar by right-clicking the icon and select "**Lock to Launcher**"). You can also press [Ctrl]+[Alt]+[T] under the root directory to open the terminal.

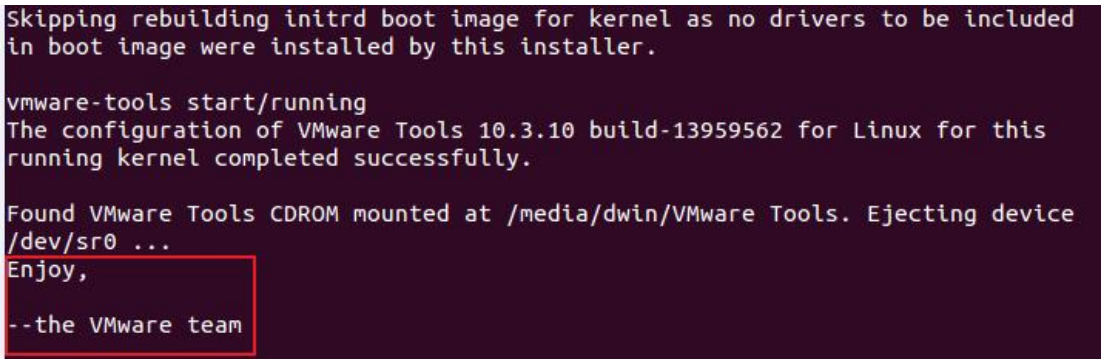

(7) Enter the command to enable the operable privilege: **sudo chmod +x VM** (to display the full name by **Tab** key) (**enter**). (Note: for the first time to use the administrator sudo privilege, youneed to enter the password, i.e., the login password, which is not visible when entering.)

```
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
```

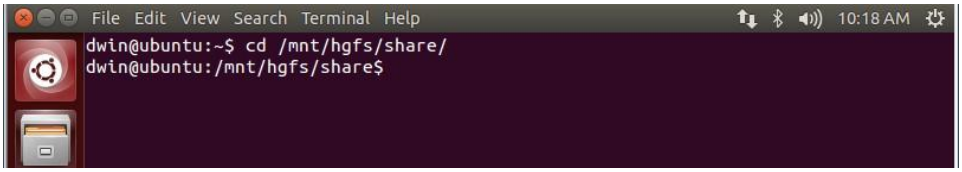
(8) Enter the decompression command: **tar -xvf VM (Tab key) (Enter)**, then it will automatically extract the tar file to the current directory. You can see the decompressed file named "vmware-tools- distrib" in the current directory. Enter the command: **cd vm (Tab key)** (The rest part is omitted).




(9) Enter the operation command: **sudo . /vm (Tab),** and then the installation will start. When [yes] or [no] appears, just type **yes** and enter for all the following options until the installation is complete as shown.

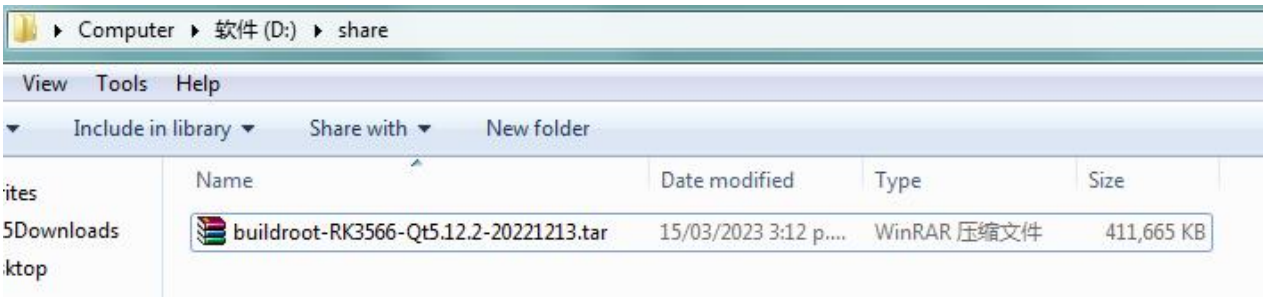



(10) At this time, we can enter the command: **cd /mn** (Tab all the way to the shared folder you set), the path is /mnt/hfgs/***, and the shared folder is set up here.




## 2.2 Install RK3566 Toolchain

(1)Use the shared folder or SFTP to move the RK3566 tar file to Ubuntu.




(2) Move the tar file to the root directory (/home/dwin) by shared folder. Enter the command: **sudo mv buil (Tab)**~. Wait a while and it will be moved to the root directory.

(3) Enter the command **tar -xvf bu(TAB)(enter)** to extract the tar file.

(4) Enter the following command in substance:

#### 2.2.1.1 cd bui(TAB)(enter)

#### 2.2.1.2 source env-setup(enter)

Enter the command **qmake -v** to check the version of qmake and see if the environment is successfully built.

```
dwin@ubuntu:~/buildroot-RK3566-Qt5.12.2-20221213$ source env-setup.sh
dwin@ubuntu:~/buildroot-RK3566-Qt5.12.2-20221213$ qmake -v
QMake version 3.1
Using Qt version 5.12.2 in /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/usr/loc
al/Qt-5.12.2/lib
```

## 2.3 Screen Configuration for QT Development

### 2.3.1 Hardware Introduction

Please refer to the relevant model specification for specific details.

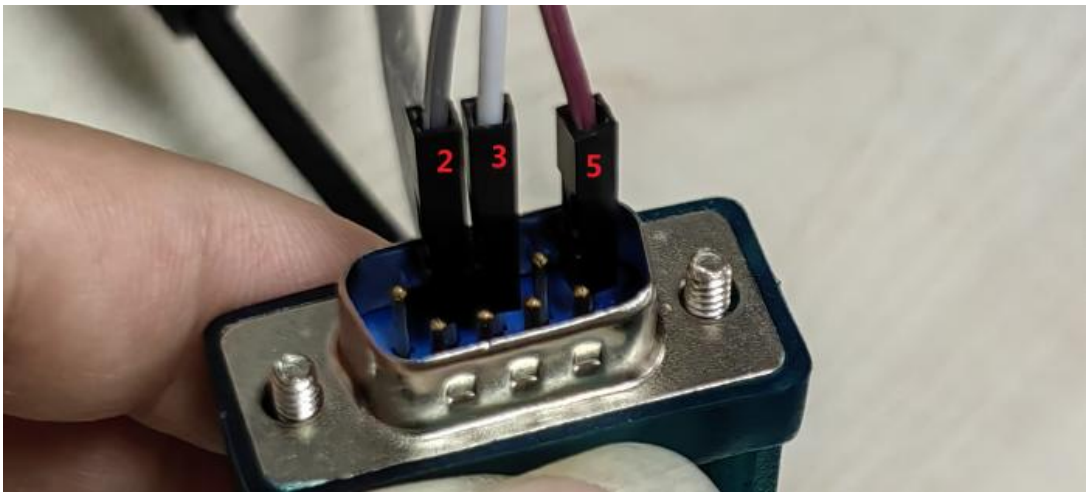
### 2.3.2 Terminal Software

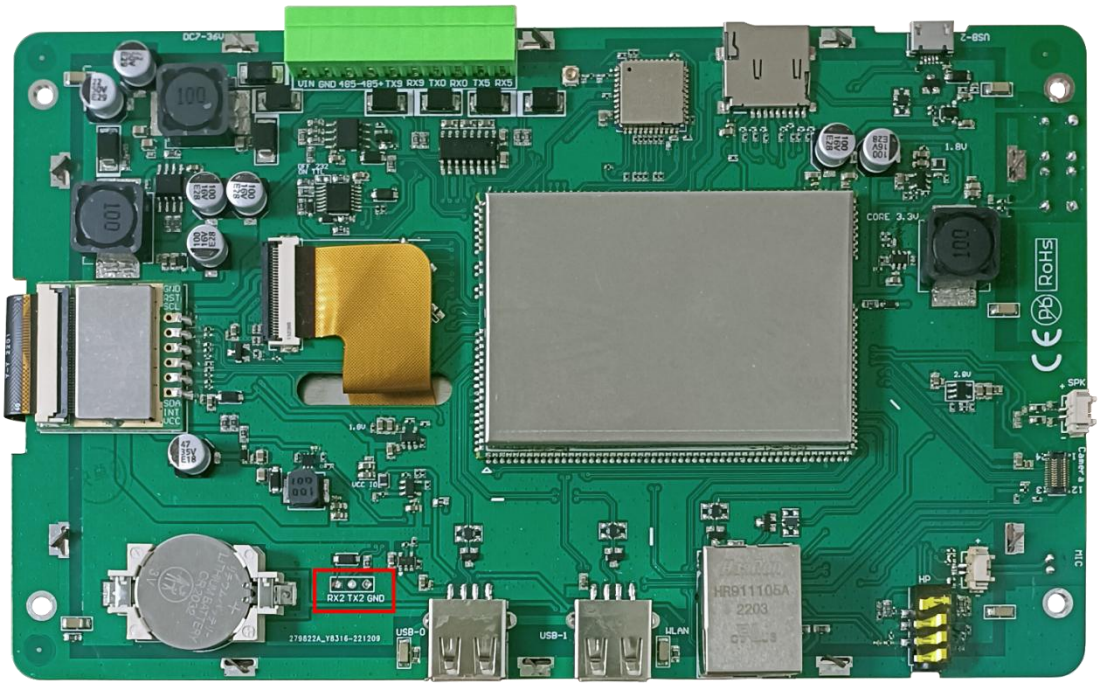
(1) You can download and use either SecureCRT or MobaXterm, and this section will introduce the use of MobaXterm.

(2) There are two connection options: Serial connection and Telnet connection by a network cable.

### 2.3.3 Serial Connection

(1) Serial connection. As illustrated in the following pictures(RS232 as an example here.)




2(RX)⇔TX, 3(TX)⇔RX, 5(GND)⇔GND. (Please prepare RS232 or TTL interface before connection. For the specific type of the debugging serial port, please refer to the datasheet of the corresponding model. At present, the debugging serial port of this series is RX2 and TX2.)

(2) Select [Sessions]-> [New Session]. First, select “Serial”. Next, select serial port and select speed. Last, check the information and click “OK” to finish.

(3) Power up the development board, and enter “root” to start. (Note: If you operate after a while after powering up, there may be no text on the displayed interface, and only a black screen with no boot information. In this case, you only need to enter “root”).

### 2.3.4 Static IP

Modify the file /etc/init.d/S40netconfig. Adjust the static IP as required and remove the comment symbol #.

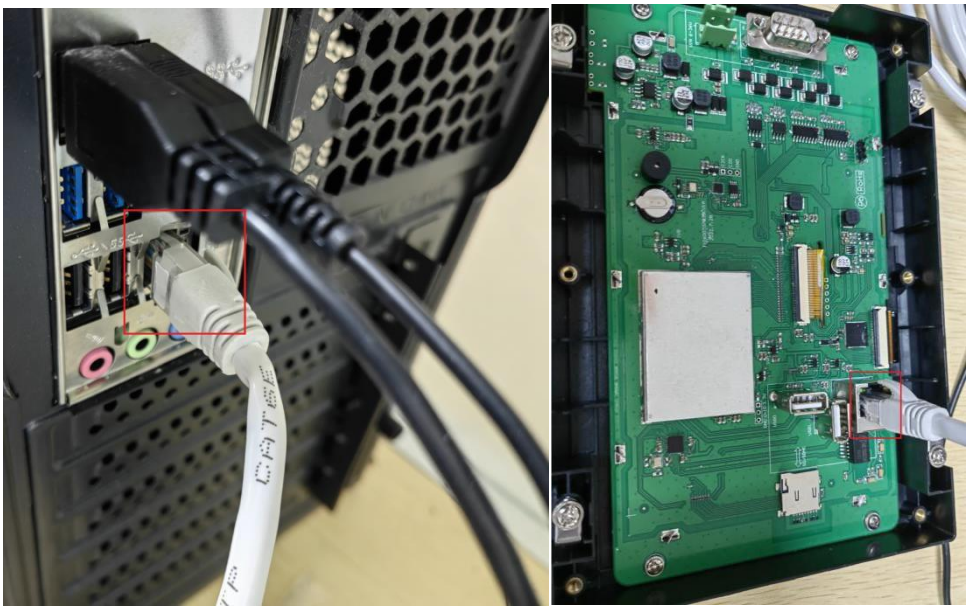
```
root@RK356X:/etc/init.d# cat S40netconfig
#!/bin/sh

case "$1" in
start)
    ifconfig eth0 up
    # ifconfig eth0 192.168.10.202

    /sbin/ifup -a
    [ $? = 0 ] && echo "OK" || echo "FAIL"
    ;;
stop)
    /sbin/ifdown -a
    [ $? = 0 ] && echo "OK" || echo "FAIL"

    ifconfig eth0 down
    printf "stop finished"
    ;;
```

Comment out two lines in the /etc/init.d/S41dhcpcd file, as shown in the figure below:

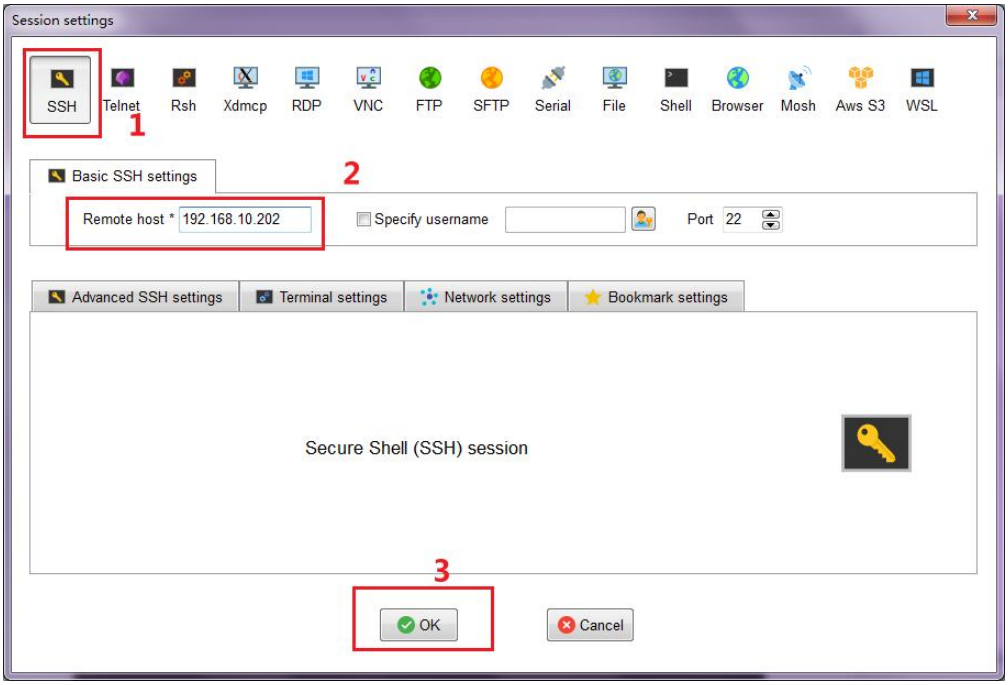
```
#!/bin/sh
#
# Start/stop dhcpcd
#

DAEMON=/sbin/dhcpcd
CONFIG=/etc/dhcpcd.conf
PIDFILE=/var/run/dhcpcd.pid

[ -x $DAEMON ] || exit 0
[ -f $CONFIG ] || exit 0

case "$1" in
  start)
        # echo "Starting dhcpcd ... "
        # start-stop-daemon -S -x "$DAEMON" -p "$PIDFILE" -- -f "$CONFIG"
        ;;
  stop)
        echo "Stopping dhcpcd ... "
        start-stop-daemon -K -x "$DAEMON" -p "$PIDFILE" -o
```

## 2.3.5 Ethernet SSH Connection

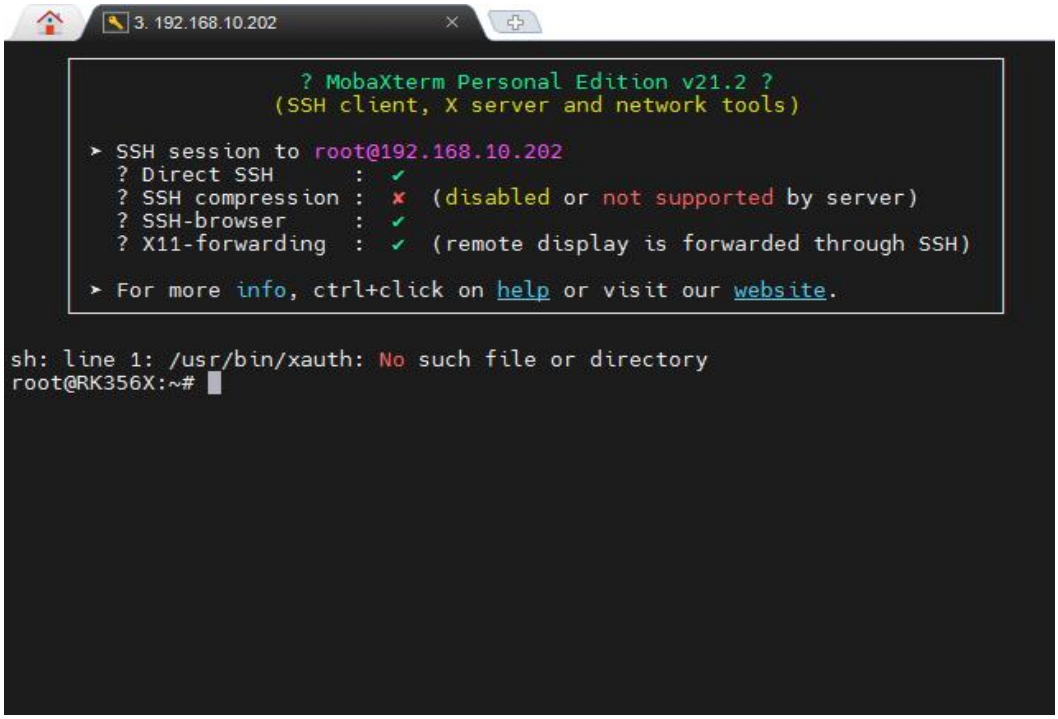
(1) Plug the network cable into the development board network port, an refer the specific notes on the Internet.



(2) First, click “Sessions” and select “New sessions” then select “SSH”. Next, enter the IP of the development board and click “OK” (Note: For 40 series devices, the IP is automatically assigned via DHCP. There is no default IP. You need to refer to the previous step. After connecting to the device via the serial port, check the current IP value of the device. The IP here is 192.168.10.202.)



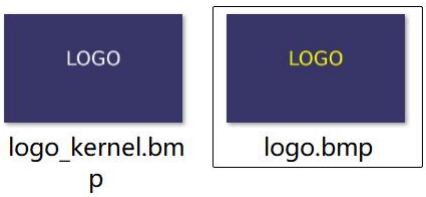
(3) Power on the development board and the following interface is displayed. Enter "root" for the username and "rockchip" for the password to start the operation.




## 2.3.6 Modify the startup logo

Create a new folder named "update" on the USB drive. Then, create a sub-folder named "DWIN_V1-0-0". Inside the "DWIN_V1-0-0" sub - folder, create a folder named "logo" and a file named "emcversion".


Prepare two BMP-format pictures and name them "logo" and "logo_kernel" respectively. Then put them into the "logo" folder.



Put the install script file into the DWIN_V1-0-0 folder, and add the following code to the install script 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)

#cp $instdir/emcversion /etc/

#cp -f $instdir/lib/libqlinuxfb.so /usr/local/Qt_5.12.5/plugins/platforms/
#chmod 755 /usr/local/Qt_5.12.5/plugins/platforms/libqlinuxfb.so
#
#cp -f $instdir/etc/runqt /etc/
#chmod 777 /etc/runqt

mkdir -p /userdata/custom_logo
cp -r $instdir/logo/* /userdata/custom_logo

sync
#$instdir/serio_app
```

Compress the DWIN_V1-0-0 folder into a tar-format file. Then put it into the update folder on the USB drive. The upgrade package is ready.

Insert the USB flash drive into the Linux screen and power it on again. Wait for the progress bar during the upgrade process. If a green progress bar appears, it means the logo replacement is successful. After the replacement is successful, the device will automatically shut down. Then remove the USB drive and power it on again.

### 2.3.7 Wi-Fi Connection

In the update folder within the "Modify LOGO" section, there is a code file named wpa_supplicant.conf. The code is shown in the figure below. The content marked by the red frame is the Wi-Fi name, and the content marked by the blue frame is the Wi-Fi password. Users can modify the content within the double quotes according to the actual situation.

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1
update_config=1

network={
    ssid="SSID"
    psk="PASSWORD"
    key_mgmt=WPA-PSK
}
```

After the modification, place the code file wpa_supplicant.conf in the same directory as install.sh.


Add the code within the following green box to the install.sh folder.

```
mkdir -p /userdata/custom_logo
cp -r $instdir/logo/* /userdata/custom_logo
cp  $instdir/wpa_supplicant.conf   /etc
sync
```

Insert the USB flash drive into the Linux screen and power it on again.

## 2.3.8 Time Zone Replacement

As mentioned in the previous section, add the following code to the install.sh file.

```
ln -sf  /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
```

Note: Please change the specific time zone according to your needs. `/Asia/Shanghai`

## 2.3.9 System Time Settings

date: View the system time.

date-s: Set the system time.

hwclock: View the hardware clock.

hwclock-w: Write the system time to the hardware clock.

```
root@RK356X:/etc/network# date
Fri Sep 27 07:25:10 UTC 2024
root@RK356X:/etc/network# date -s "2023-10-10 10:10:10"
Tue Oct 10 10:10:10 UTC 2023
root@RK356X:/etc/network# hwclock
Fri Sep 27 07:25:47 2024  0.000000 seconds
root@RK356X:/etc/network# hwclock -w
root@RK356X:/etc/network# hwclock
Tue Oct 10 10:10:35 2023  0.000000 seconds
root@RK356X:/etc/network#
```

## 2.3.10 Screen Volume and Brightness

### 2.3.10.1 Volume

Command: amixer set Master 5% (Range: 0%~100%)

Pressing the numbers from 1 to 9 sets the volume to 10, 20, 30...

Press Shift +"+" or"-" to increase or decrease the volume by single-digit percentages.

```
root@RK356X:/# amixer set Master 5%
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch pswitch-joined
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 65536
  Mono:
  Front Left: Playback 3277 [5%] [on]
  Front Right: Playback 3277 [5%] [on]
root@RK356X:/# amixer set Master 50%
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch pswitch-joined
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 65536
  Mono:
  Front Left: Playback 32768 [50%] [on]
  Front Right: Playback 32768 [50%] [on]
```

### 2.3.10.2 Brightness

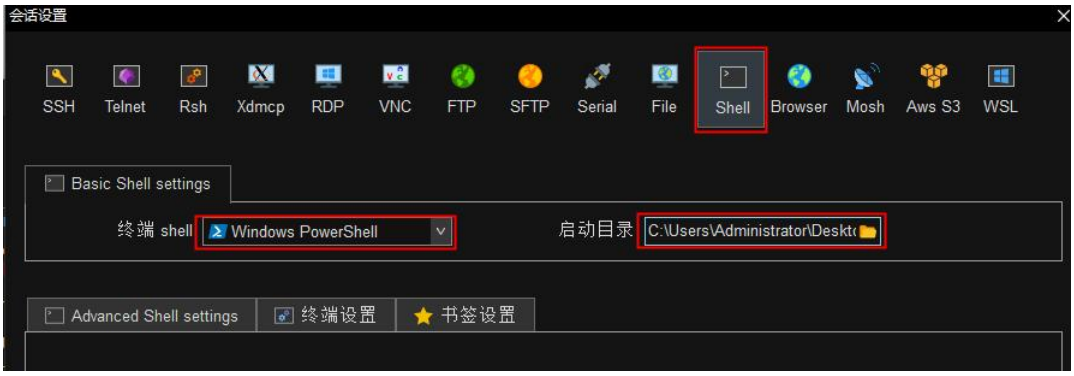
Modify the value in the file /sys/class/backlight/backlight/brightness.

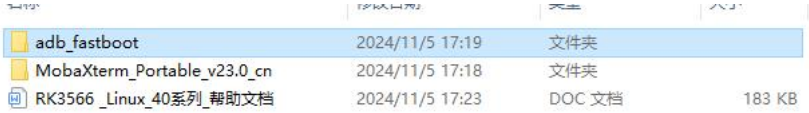
Range: 0~255.

```
root@RK356X:/# cat /sys/class/backlight/backlight/brightness
200
root@RK356X:/# echo 100 > /sys/class/backlight/backlight/brightness
root@RK356X:/# echo 5 > /sys/class/backlight/backlight/brightness
```

## 2.3.11 ADB

Open the serial communication tool→Shell→Windows PowerShell(Select the directory where the adb_fastboot folder is located as the startup director)→OK→Enter .\adb.exe shell.





```
PS C:\Users\Administrator\Desktop\adb_fastboot> .\adb.exe  shell
root@RK356X:/#
root@RK356X:/#
```

# 2.4 Screen Confguration for Debian Development

## 2.4.1 Wi-Fi

Locate Wi-Fi

```
nmcli dev wifi list
```

```
root@linaro-alip:/# nmcli dev wifi list
IN-USE  BSSID              SSID                    MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
        88:25:93:5D:7E:4A  DWIN-813                Infra  6     405 Mbit/s  57      ▂▄▆_  WPA1 WPA2
        D0:76:E7:12:13:B0  TP-LINK_13B0            Infra  1     405 Mbit/s  47      ▂▄__  --
        4C:10:D5:3B:A2:88  2.4G-AP                 Infra  6     405 Mbit/s  39      ▂▄__  WPA1 WPA2
        0A:71:90:37:CE:22  DIRECT-OAKWIOKmsPK      Infra  7     130 Mbit/s  34      ▂▄__  WPA2
        32:24:A9:07:D9:FA  DIRECT-fa-HP 8133       Infra  6     65 Mbit/s   32      ▂▄__  WPA2
```

Show Wi-Fi connection

```
nmcli connection show
```

```
root@linaro-alip:/# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
DWIN-813            395e7634-bfbc-4753-8b25-728ed626b979  wifi      p2p0
4G-UFI-6C37         d46bd0d3-4edb-403b-9473-e36f7cd089be  wifi      --
A8CC                eb4046a1-850f-483e-abe0-ec8633c2b37b  wifi      --
DIRECT-fa-HP 8133   8bf27efd-170d-41f4-a442-1414f94937ee  wifi      --
Wired connection 1  861bbc46-b69b-3fb9-a164-62e11d3f07fa  ethernet  --
```

Remove the Wi-Fi connection

```
nmcli con delete id "DWIN-813"
```

```
root@linaro-alip:/# nmcli con show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  861bbc46-b69b-3fb9-a164-62e11d3f07fa  ethernet  --
```

Add a new Wi-Fi connection

```
nmcli dev wifi connect "DWIN-813" password "123456"
```

```
root@linaro-alip:/# nmcli con show
NAME                UUID                                  TYPE      DEVICE
DWIN-813            f6caf5d7-e52d-4352-8b50-71d059639f45  wifi      p2p0
Wired connection 1  861bbc46-b69b-3fb9-a164-62e11d3f07fa  ethernet  --
```

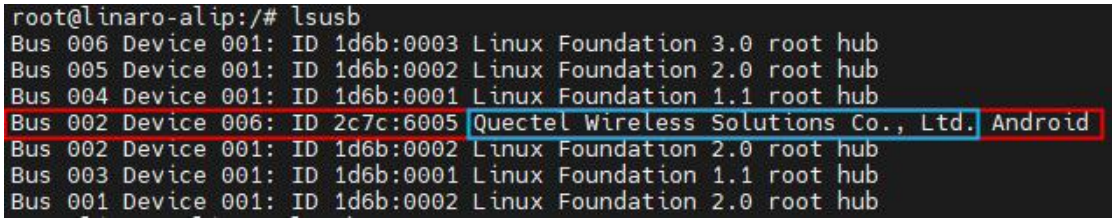
```
root@linaro-alip:/# nmcli dev wifi connect "Redmi K50 Ultra" password "[redacted]"
Device 'p2p0' successfully activated with '7f8d8286-c80c-418e-9713-2182f94ba6d8'.
root@linaro-alip:/# nmcli con show
NAME                UUID                                  TYPE      DEVICE
Redmi K50 Ultra     7f8d8286-c80c-418e-9713-2182f94ba6d8  wifi      p2p0
DWIN-813            5a5575b3-b114-4d02-ba83-44700aa4286e  wifi      --
Wired connection 1  861bbc46-b69b-3fb9-a164-62e11d3f07fa  ethernet  --
```

It is necessary to remove other invalid connections before creating a new Wi-Fi connection.

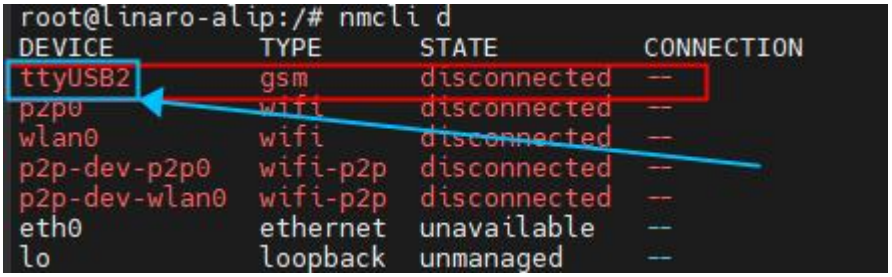
## 2.4.2 4G Network

```
nmcli con add type gsm ifname "xxx" con-name "xxx"
```

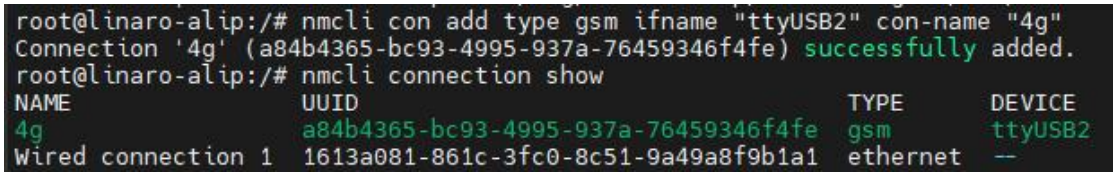
lsusb: Check if the 4G module is successfully loaded



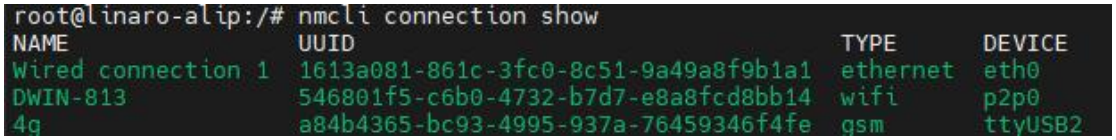

```
ifname "": nmcli d
```




con-name"": Customize the connection name for quick reference in subsequent operations.



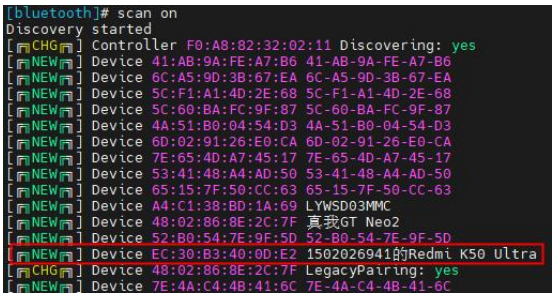

Priority： Wired connection > wifi > 4g



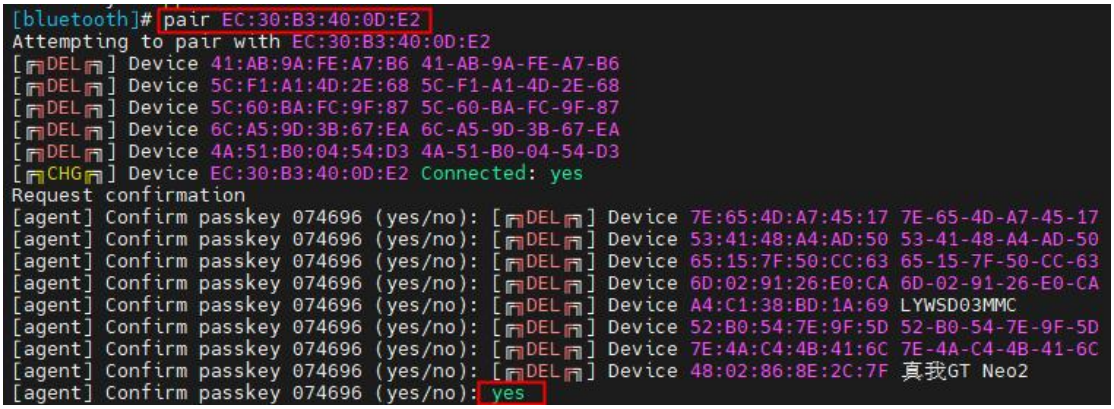

## 2.4.3 Bluetooth

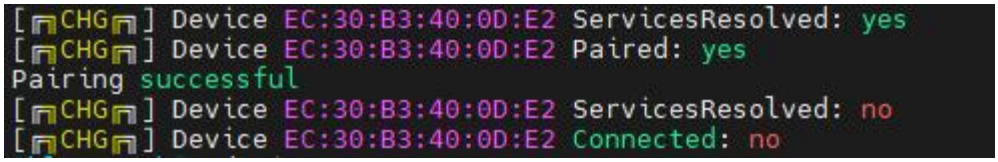
Run the "bluetoothctl" command to enter the bluetooth settings interface.

```
scan on
```

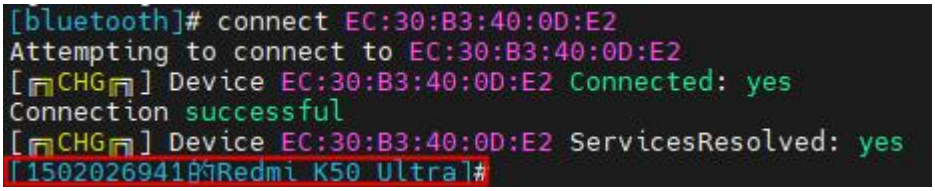


```
scan off

pair [dev]
```

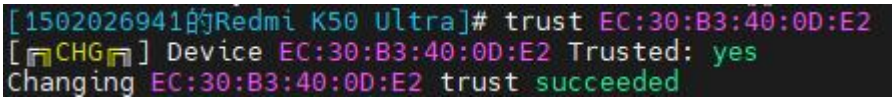



```
connect [dev]
```




```
trust [dev]
```




Use the "Help" option in the menu to get a list of available commands.

# 3 QT Project Cross-compilation

## 3.1 Install Qt Creator

### 3.1.1 System Requirements

This document is based on Ubuntu 14.04 system for verification. Other versions of Ubuntu systems should work but are not verified.

### 3.1.2 Download Qt Creator

The version of Qt Creator used in this document is 2.7.2. Please download the version that matches the operating system.

| | | | |
|---|---|---|---|
| source/ | 02-Jul-2013 19:43 | - | |
| qt-creator-windows-opensource-2.7.2.exe | 02-Jul-2013 19:43 | 53M | Details |
| qt-creator-mac-opensource-2.7.2.dmg | 02-Jul-2013 19:43 | 53M | Details |
| qt-creator-linux-x86_64-opensource-2.7.2.bin | 02-Jul-2013 19:43 | 62M | Details |
| qt-creator-linux-x86-opensource-2.7.2.bin | 02-Jul-2013 19:43 | 63M | Details |
| qt-creator-2.7.2-src.zip | 02-Jul-2013 19:43 | 27M | Details |
| qt-creator-2.7.2-src.tar.gz | 02-Jul-2013 19:43 | 22M | Details |

### 3.1.3 Install Qt Creator

Copy the installer to your Ubuntu system and add execute permissions to the file:

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

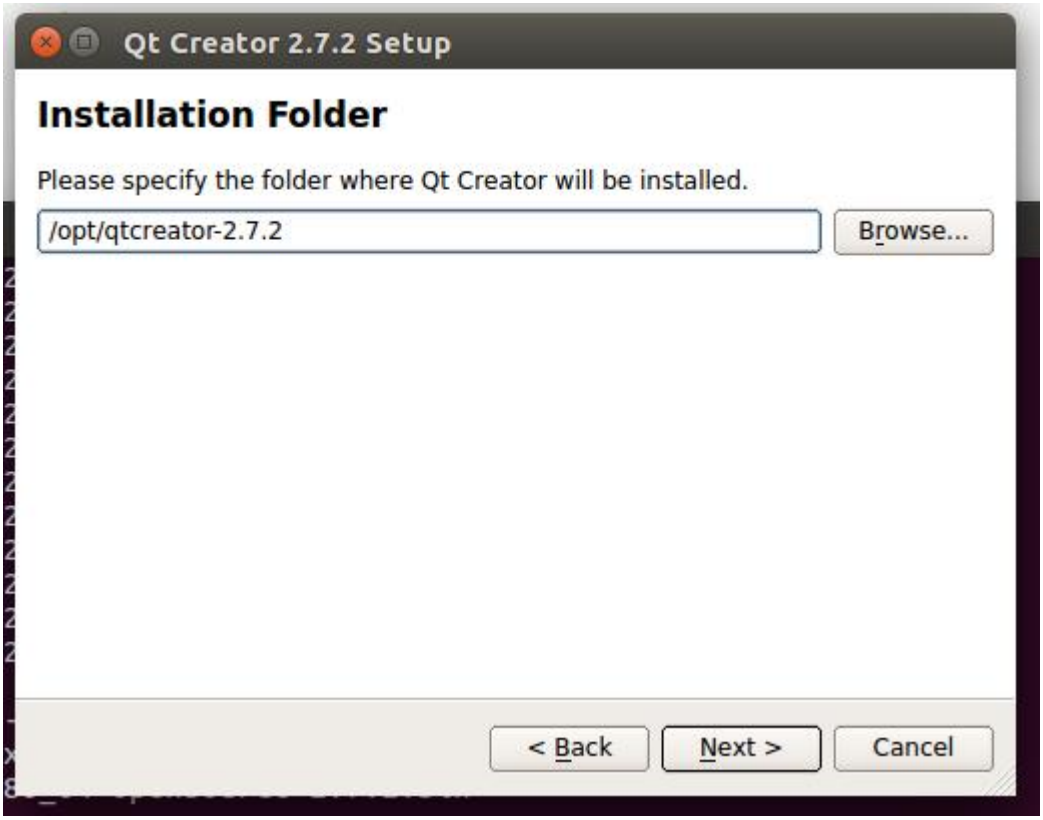
```
dwin@ubuntu:~$ cd /home/dwin
dwin@ubuntu:~$ chmod +x qt-creator-linux-x86_64-opensource-2.7.2.bin
```

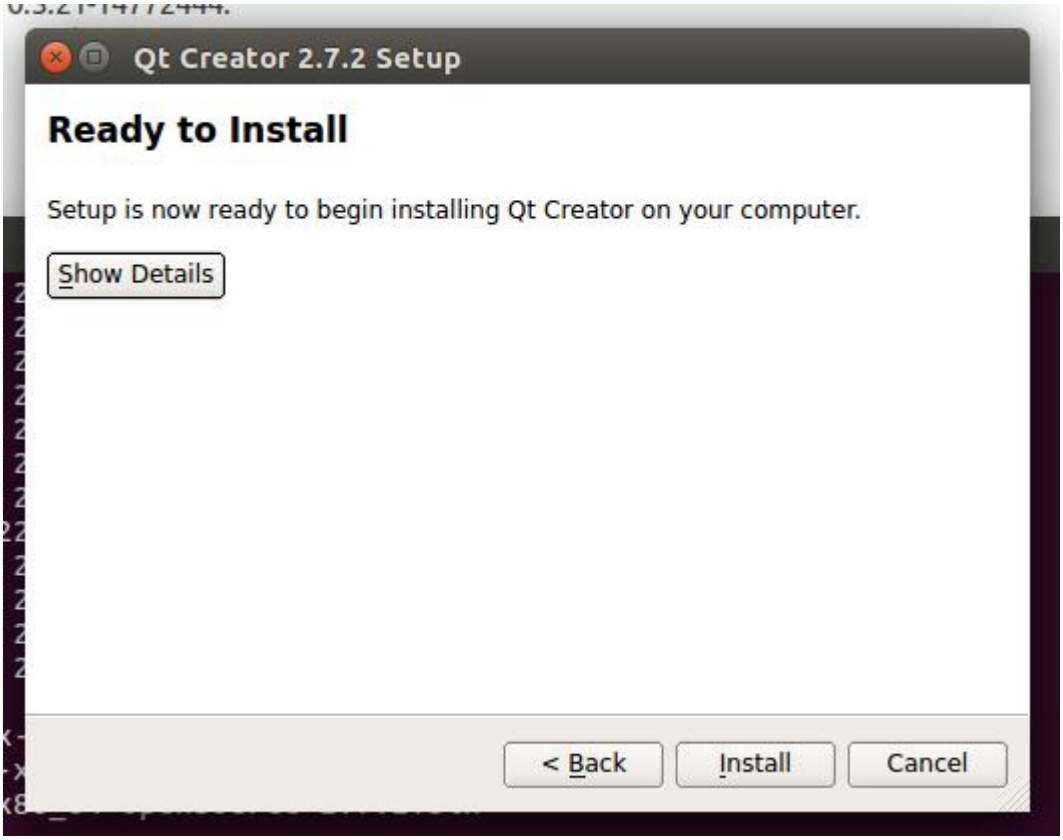
Run the installer

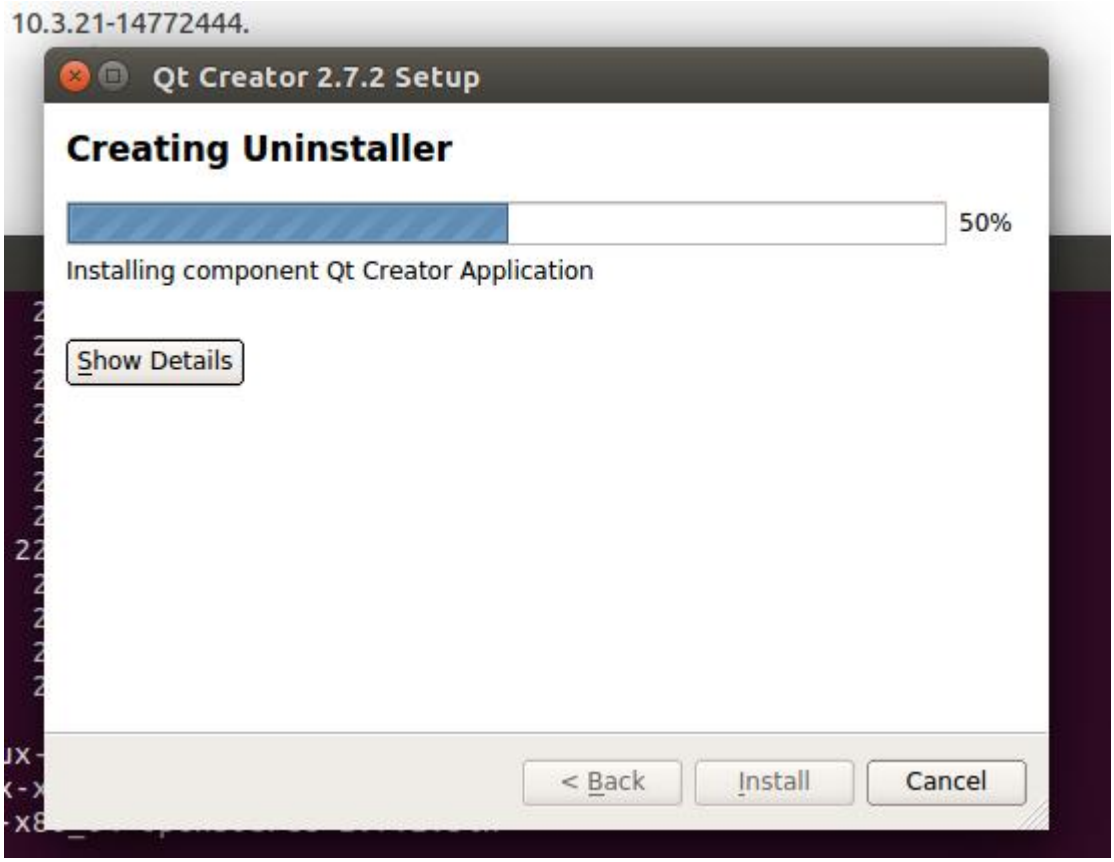
# sudo ./qt-creator-linux-x86_64-opensource-2.7.2.bi

Click the “Next”:


Qt Creator 2.7.2 Setup
Setup - Qt Creator
Welcome to the Qt Creator Setup Wizard.
Code less.
Create more.
Deploy everywhere.
Next >
Quit


Qt Creator 2.7.2 Setup
Installation Folder
Please specify the folder where Qt Creator will be installed.
/opt/qtcreator-2.7.2
Browse...
< Back
Next >
Cancel


Qt Creator 2.7.2 Setup
Ready to Install
Setup is now ready to begin installing Qt Creator on your computer.
Show Details
< Back
Install
Cancel

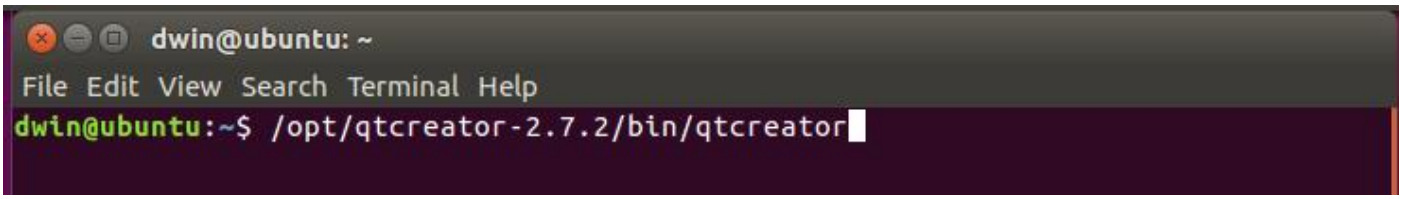

10.3.21-14772444.
Qt Creator 2.7.2 Setup
Creating Uninstaller
50%
Installing component Qt Creator Application
Show Details
< Back
Install
Cancel


## 3.2 Set up the cross-compilation environment

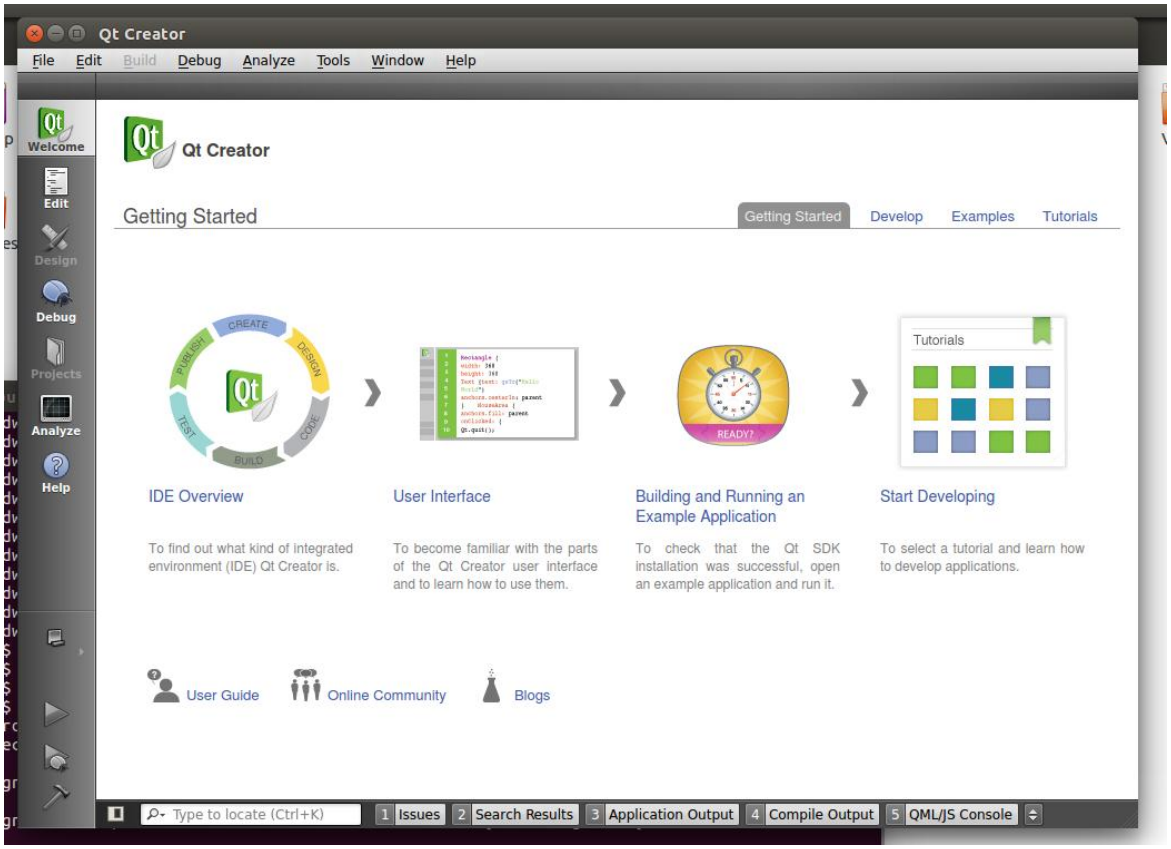
### 3.2.1 Run the Qt Creator

The Qt Creator executable file is in the bin directory of the installation directory.

# /opt/qtcreator-2.7.2/bin/qtcreator

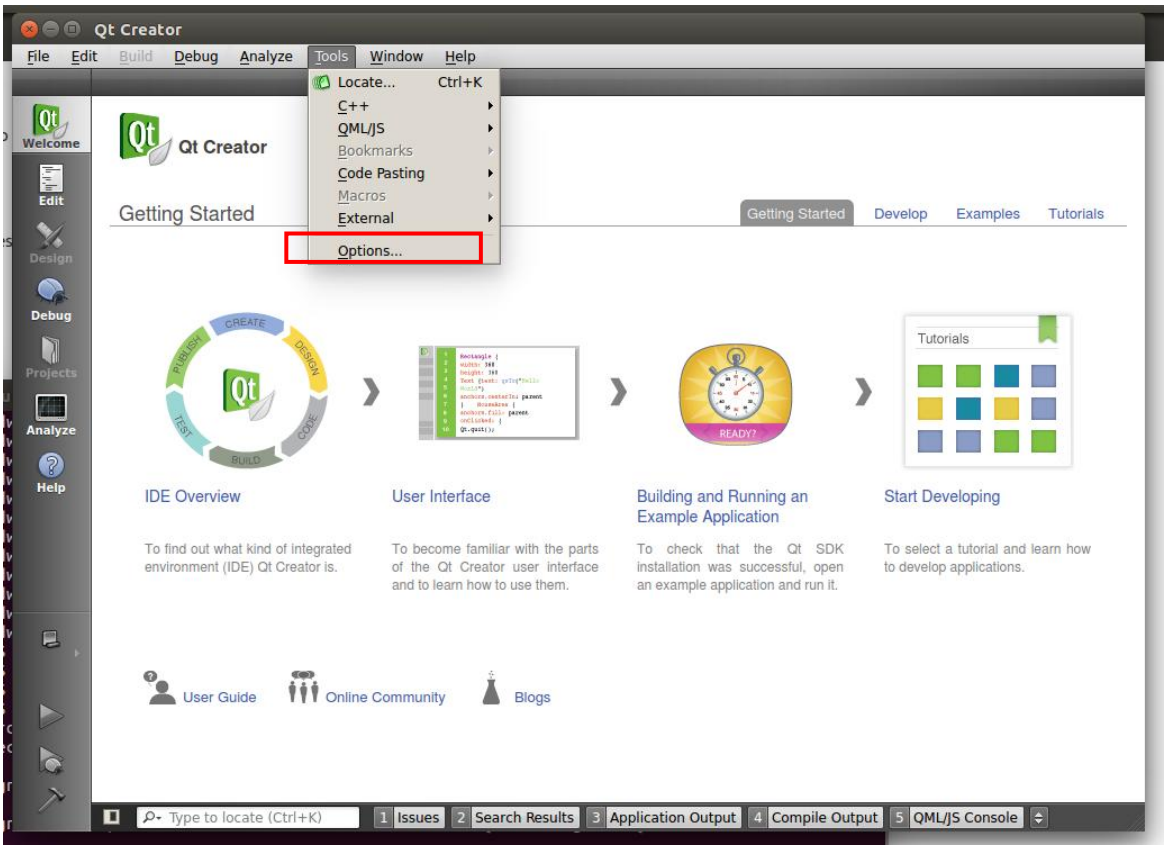


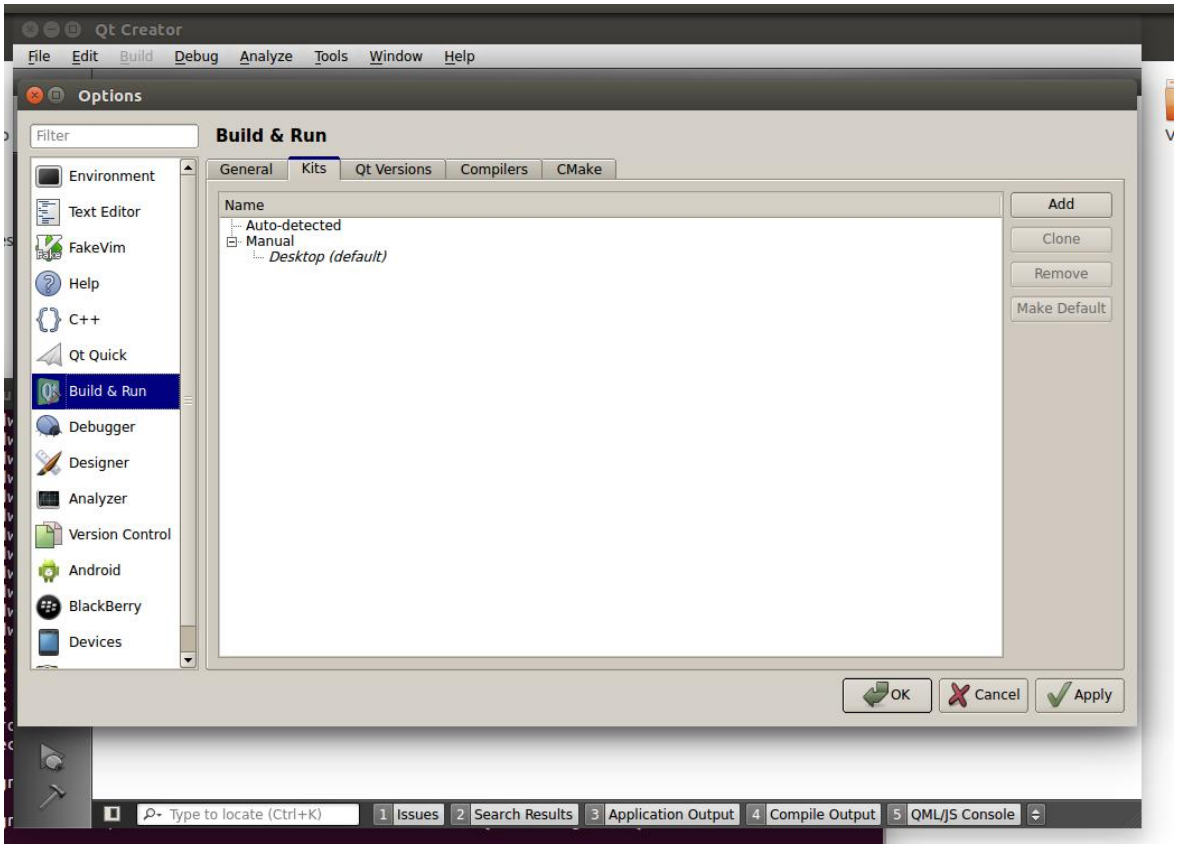
The interface of software is as below.



## 3.2.2 Set up the cross-compilation environment

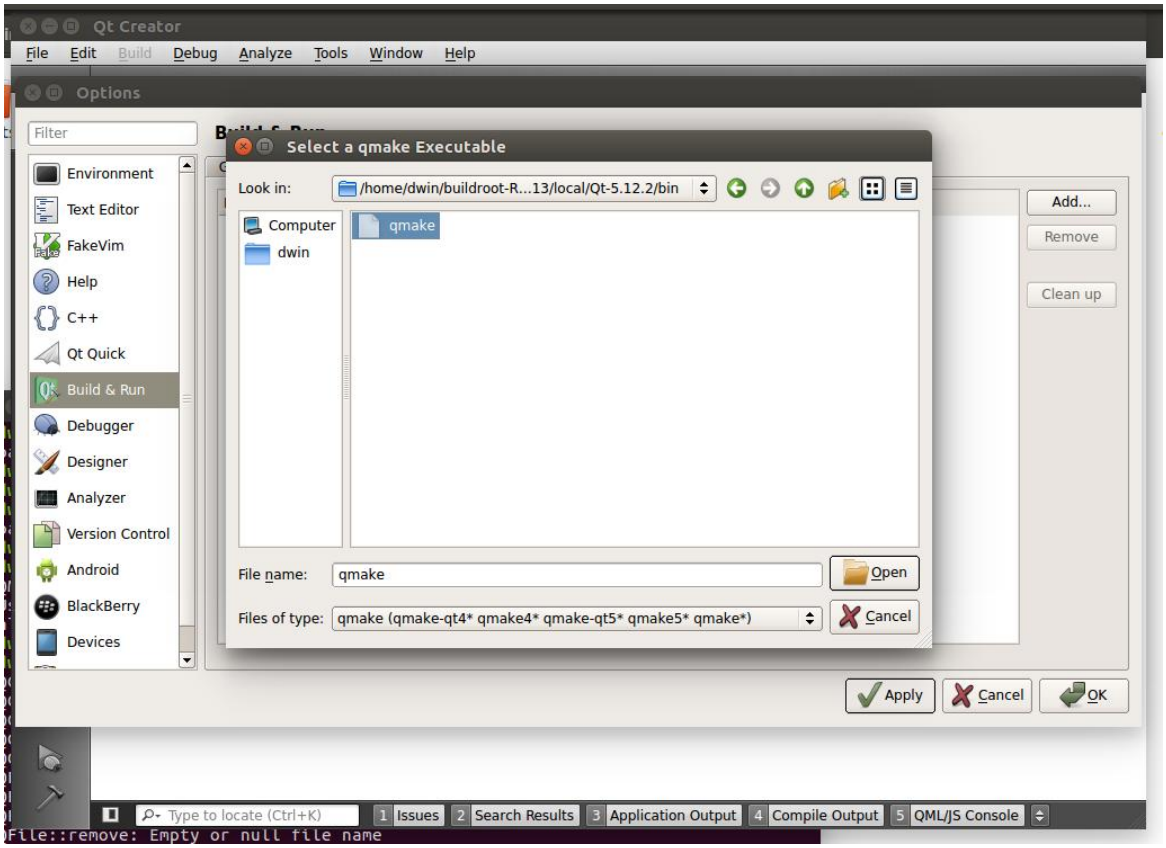
Choose [tool] – [options] as below.

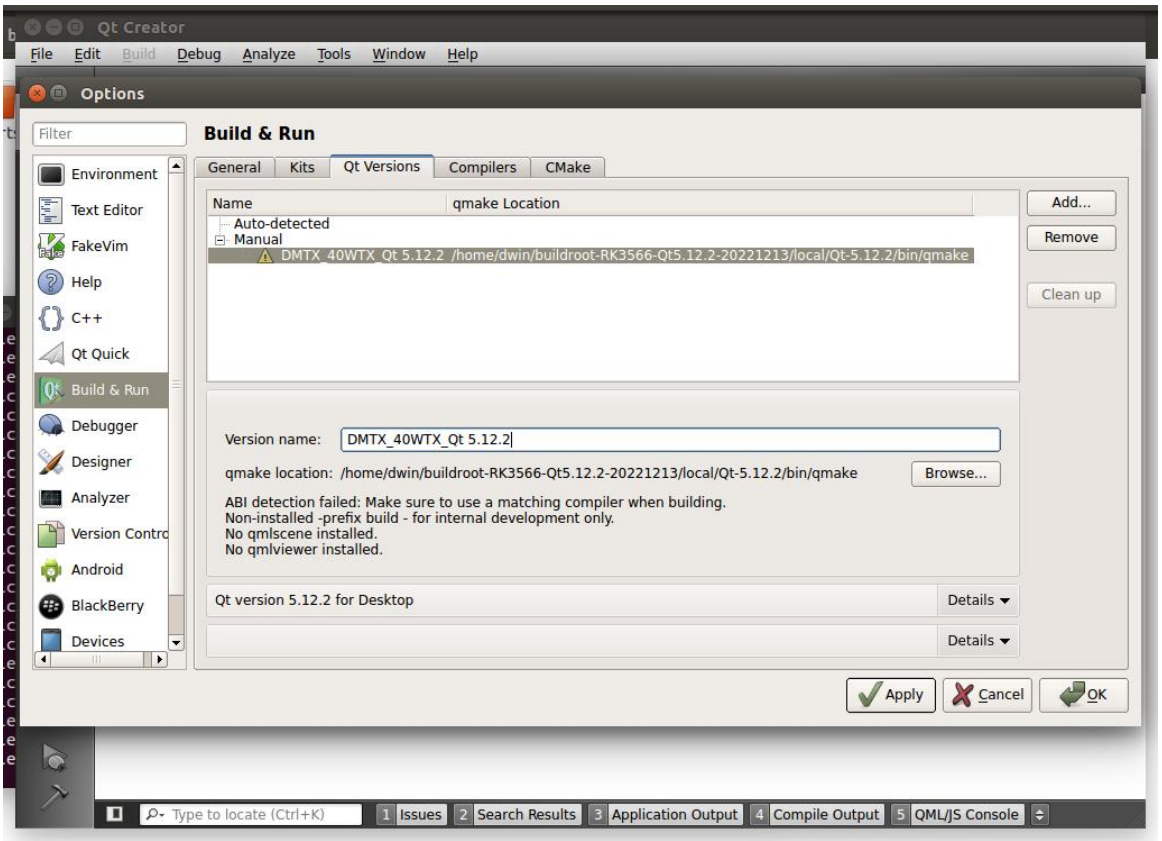


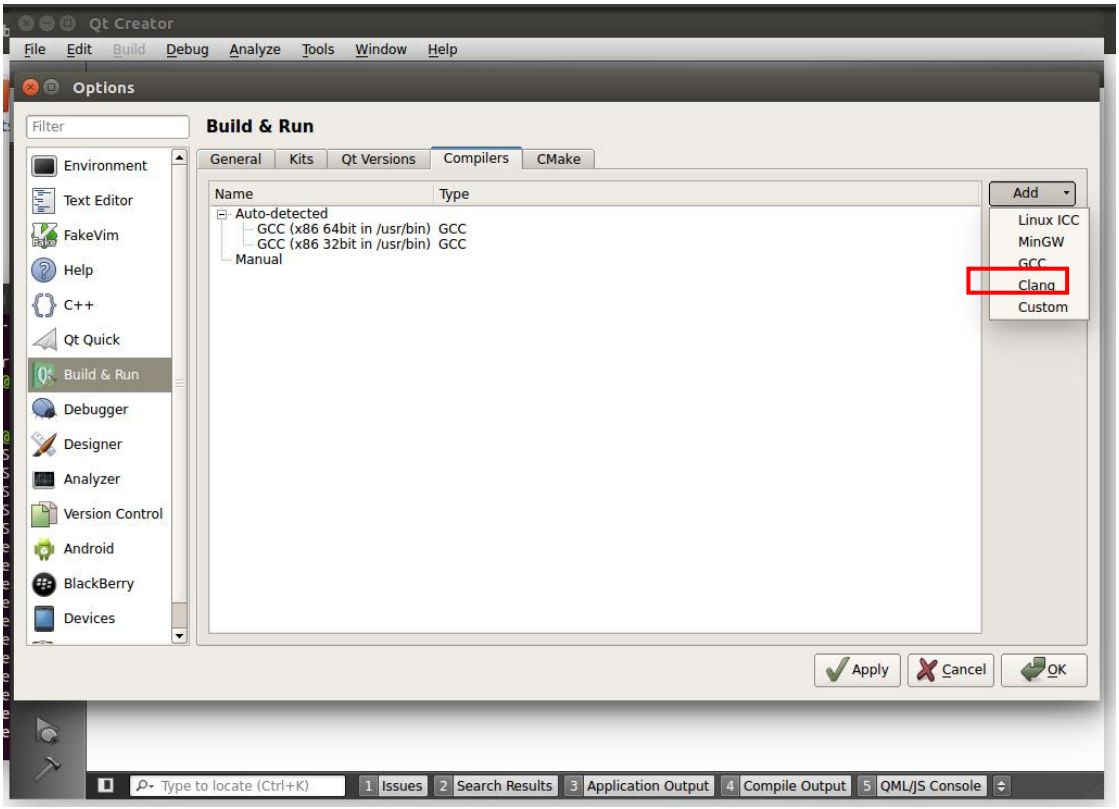
Set qmake: choose [Build & Run] – [Qt Version] – [Add],

“qmake” is in the ‘local/Qt-5.12.2/bin/’ directory of buildroot-RK3566-Qt5.12.2-20221213.tar.gz

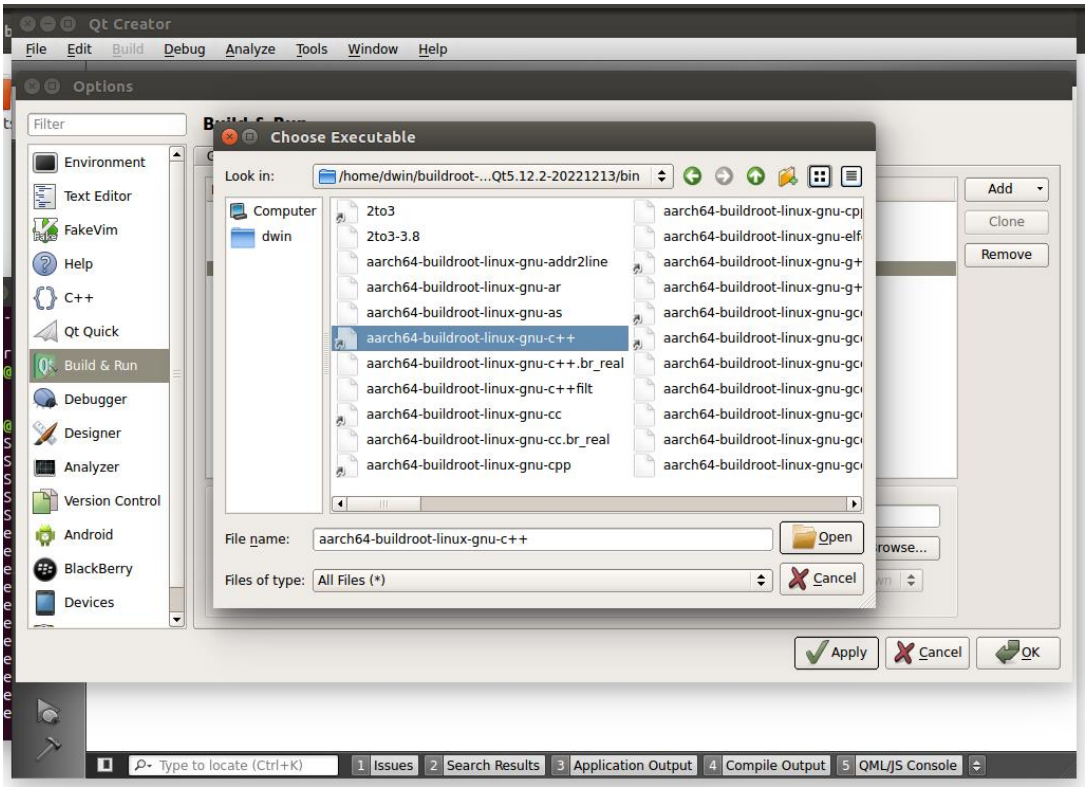


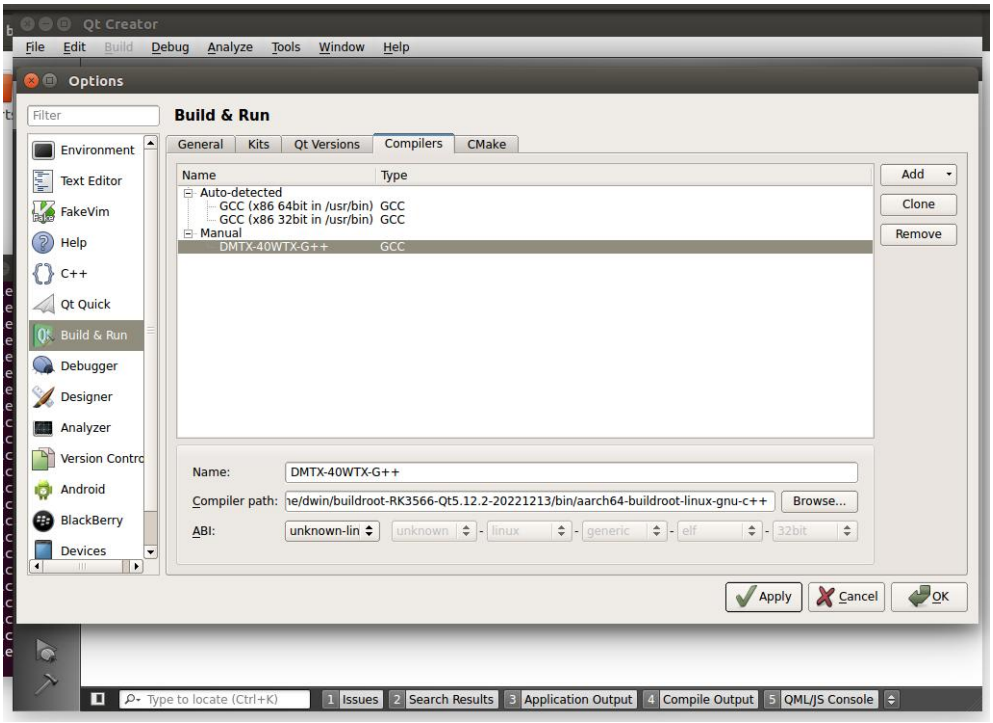



Set compilation toolchain: choose [Build & Run] – [Compilers] – [Add] – [GCC]:

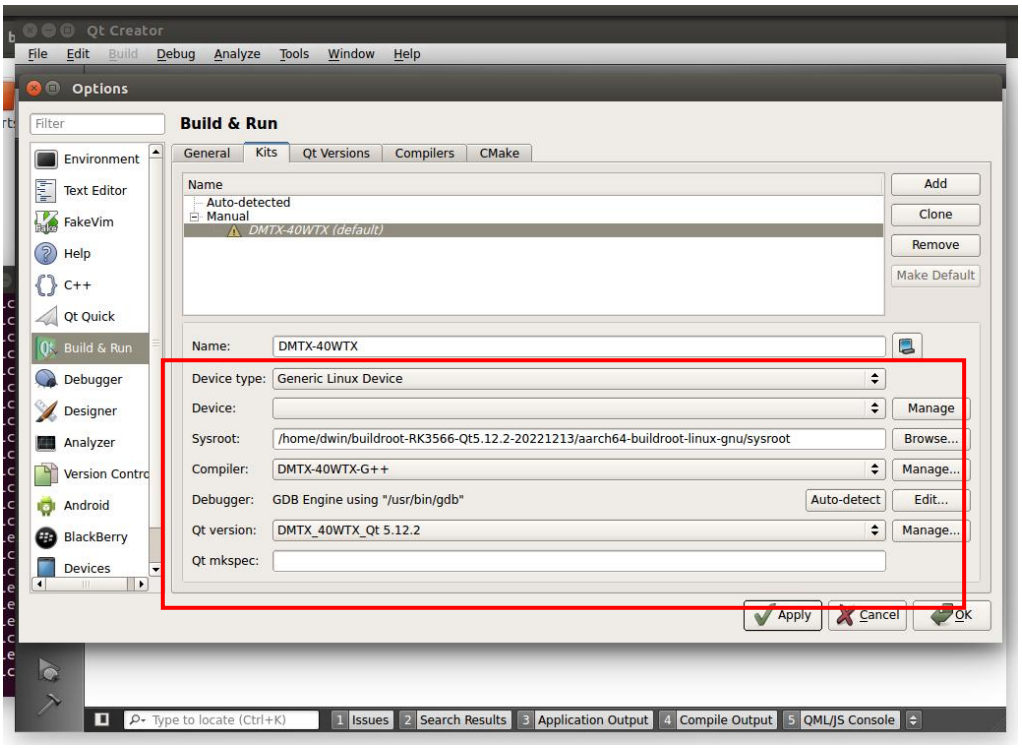



The compiler is located in the 'bin' directory of the 'buildroot-RK3566-Qt5.12.2-20221213.tar.gz' package.





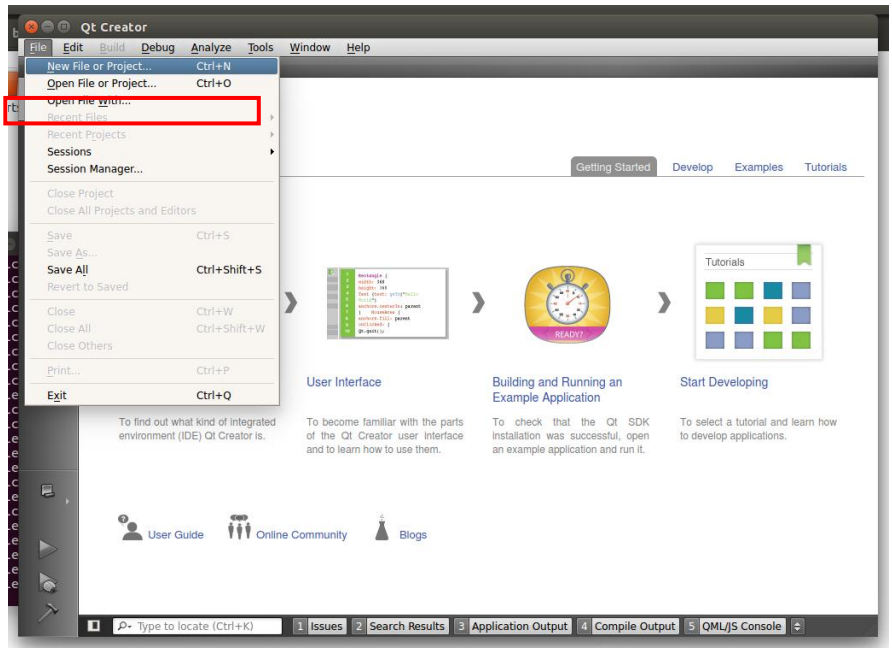

Set up the build kit: choose [Build & Run] – [Kits]:



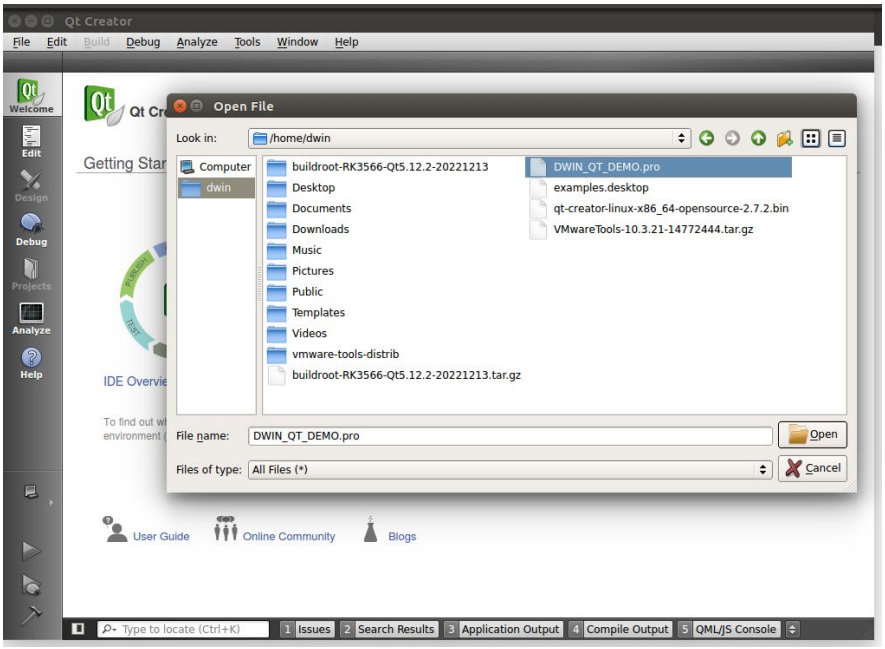

## 3.3 Compile Qt project

### 3.3.1 Open the project

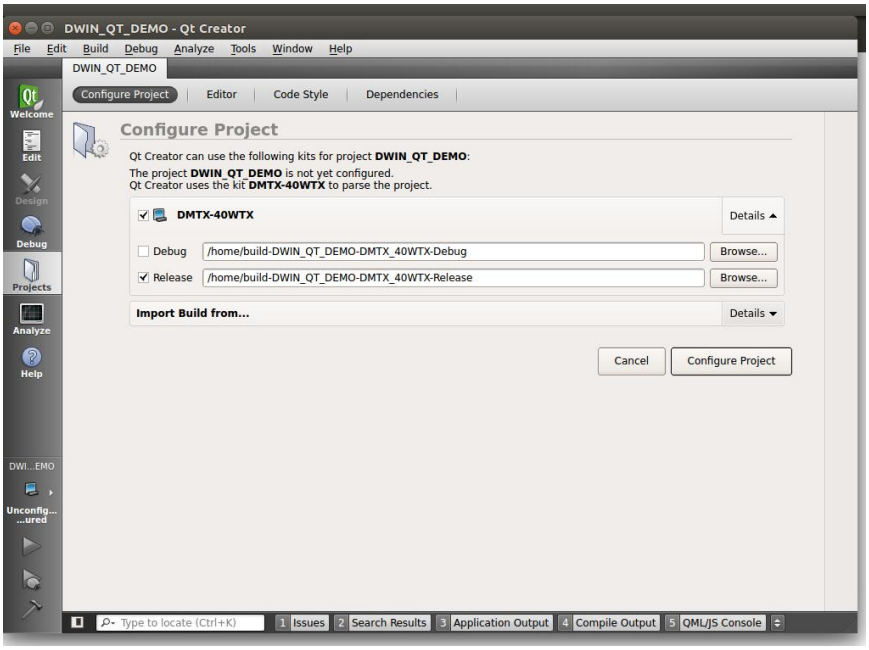
[File] – [open file or project]:




Choose the Qt project:




Configure the project:




## 3.3.2 Add environment variables

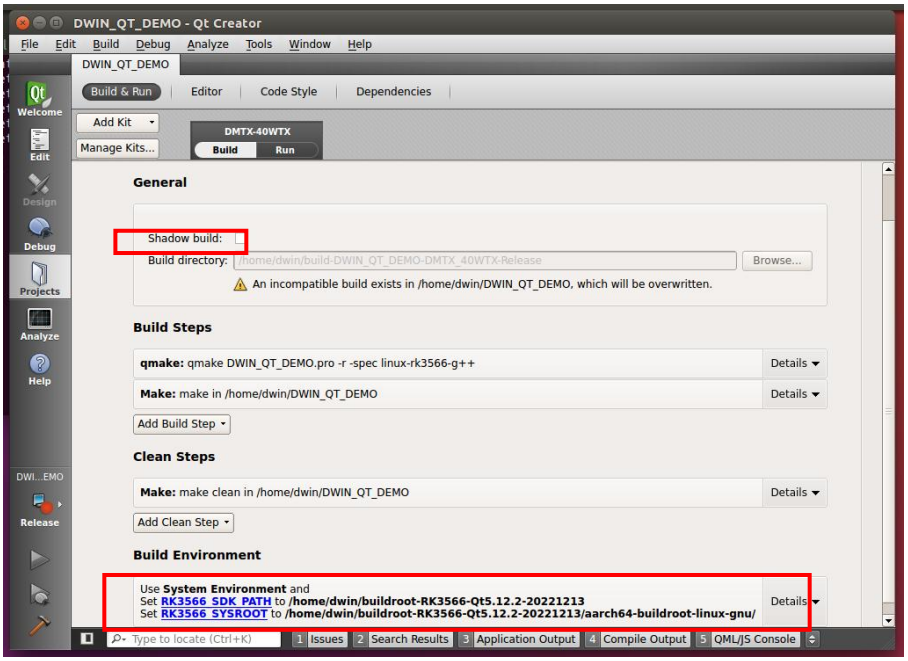
Go to [Projects] – [Build & Run[ - [Build Environment], and add a variable:

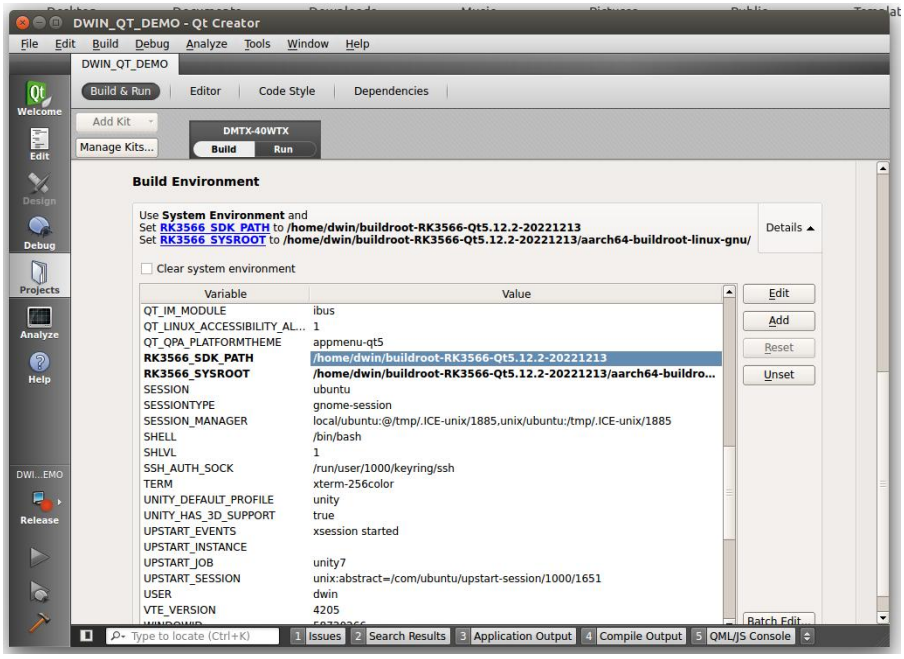
Variable Name 1: RK3566_SDK_PATH

Value 1: Root directory of the 'buildroot-RK3566-Qt5.12.2-20221213.tar.gz' package

Variable Name 2: RK3566_SYSROOT

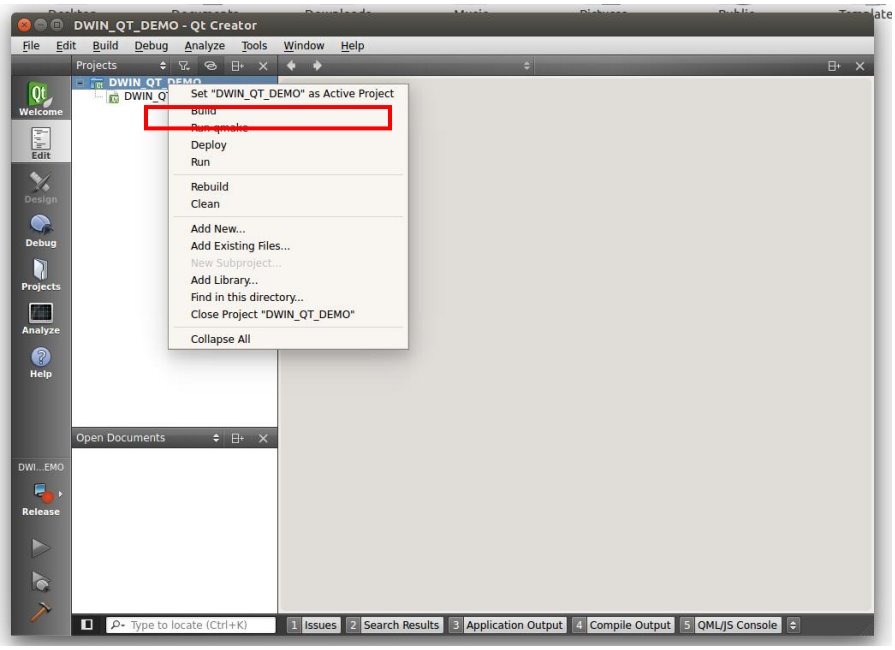
Value 2: 'sysroot' directory in the 'aarch64-buildroot-linux-gnu' directory of the 'buildroot-RK3566-Qt5.12.2-20221213.tar.gz' package.



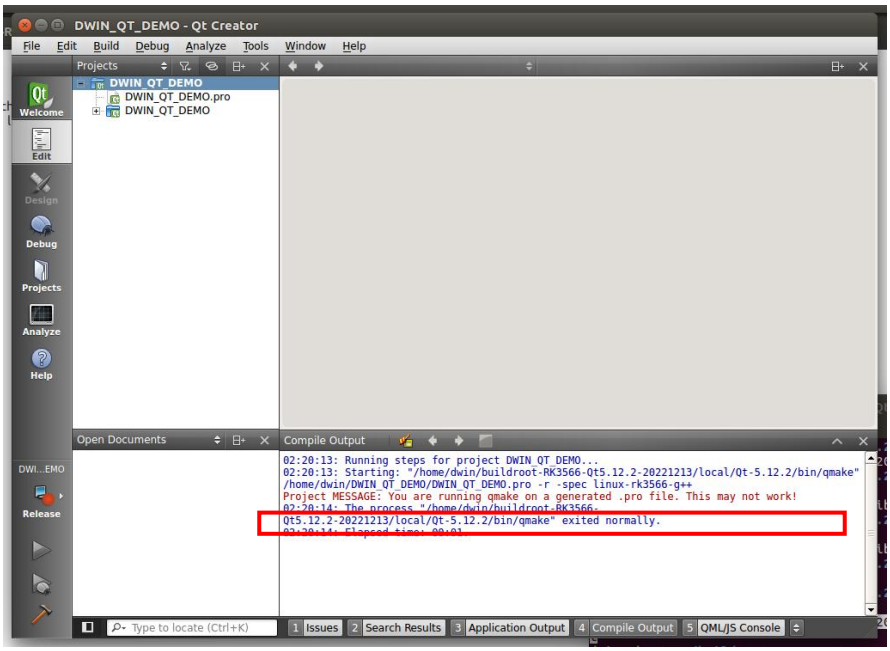



## 3.3.3 Run qmake

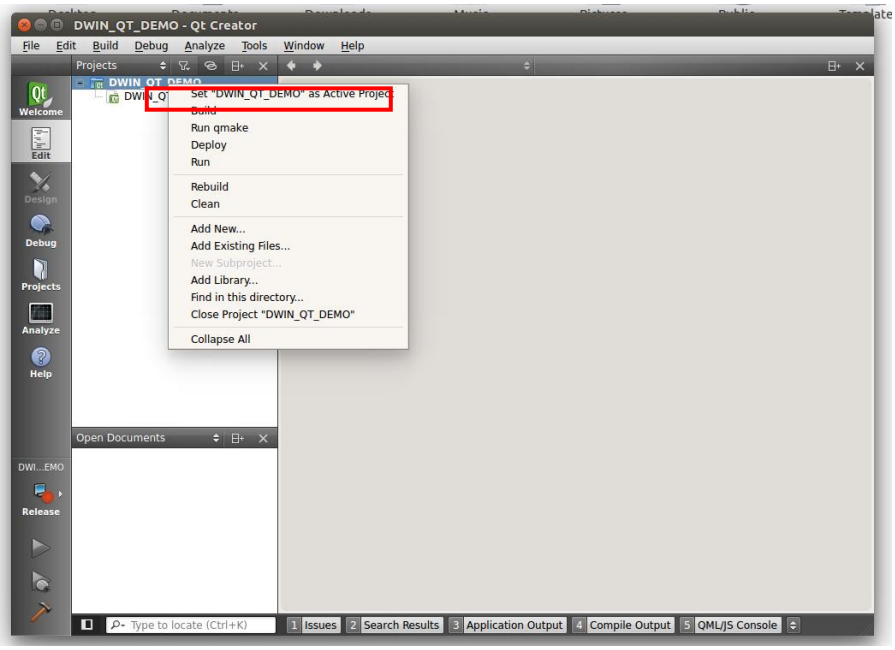
Choose project, ‘right key’ – run qmake

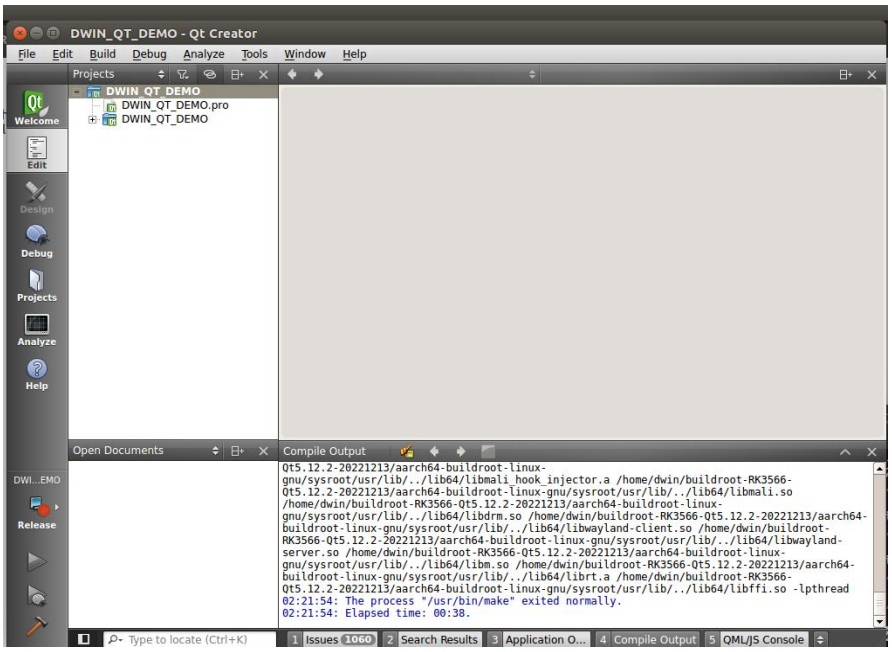



When qmake is successful, it looks like the image below (the red part is the printout of DWIN_QT_DEMO.pri, which does not affect).

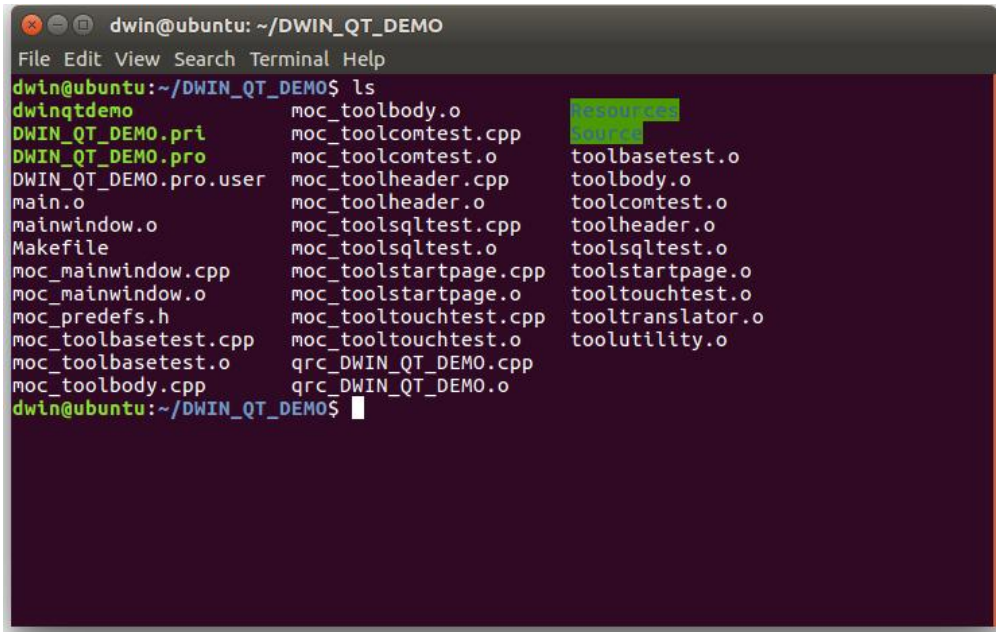

DWIN_QT_DEMO - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
DWIN_QT_DEMO
DWIN_QT_DEMO.pro
DWIN_QT_DEMO
Welcome
Edit
Design
Debug
Projects
Analyze
Help
Release
Open Documents
Compile Output
02:20:13: Running steps for project DWIN_QT_DEMO...
02:20:13: Starting: "/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/local/Qt-5.12.2/bin/qmake" /home/dwin/DWIN_QT_DEMO/DWIN_QT_DEMO.pro -r -spec linux-rk3566-g++
Project MESSAGE: You are running qmake on a generated .pro file. This may not work!
02:20:14: The process "/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/local/Qt-5.12.2/bin/qmake" exited normally.
Type to locate (Ctrl+K)
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML/JS Console

## 3.3.4 Build


DWIN_QT_DEMO - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Set "DWIN_QT_DEMO" as Active Project
Build
Run qmake
Deploy
Run
Rebuild
Clean
Add New...
Add Existing Files...
New Subproject...
Add Library...
Find in this directory...
Close Project "DWIN_QT_DEMO"
Collapse All
Open Documents
Welcome
Edit
Design
Debug
Projects
Analyze
Help
Release
Type to locate (Ctrl+K)
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML/JS Console




At this point, the target files have been generated in the project directory and can be copied to thescreen for execution.




## 3.4 qmake

(1) After enter the environment (running the "**source env-setup.sh**" command), enter the command **"qmake-v"** to check if the environment is correct. Open the project folder you need to cross-compile (here using the provided folder named "DWIN_QT_DEMO" and adding it to "Ubuntu /home/dwin"). Enter the command: **qmake** (if the .pro file hasn't been generated, enter "**qmake -project**".) to generate the Makefile.


(2) Enter the command: **make**, and then a binary file named after the project will be generated. But the file cannot be run in Ubuntu, so you need to download it to the development board. You can refer to 2.2.

```
dwin@ubuntu: ~/DWIN_QT_DEMO
dwin@ubuntu:~/DWIN_QT_DEMO$ qmake  -v
QMake version 3.1
Using Qt version 5.12.2 in /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/usr/loc
al/Qt-5.12.2/lib
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
aarch64-buildroot-linux-gnu-g++ -c -pipe --sysroot=/home/dwin/buildroot-RK3566-Q
t5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot -O2 -Wall -W -D_REENTRANT -
fPIC -DQT_NO_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_SERIALPORT_LIB -DQT_SQL_LI
B -DQT_CORE_LIB -I. -I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/
include -I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/include/QtWi
dgets -I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/include/QtGui
-I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/include/QtSerialPort
 -I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/include/QtSql -I../
buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/include/QtCore -I. -I../b
uildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot/usr/includ
e/libdrm -I../buildroot-RK3566-Qt5.12.2-20221213/usr/local/Qt-5.12.2/mkspecs/lin
```

```
/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot/usr/lib/
../lib64/librt.a /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot
-linux-gnu/sysroot/usr/lib/../lib64/libffi.so -lpthread
dwin@ubuntu:~/DWIN_QT_DEMO$ ls
dwingtdemo             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$
```

## 3.5 USB Download

(1) Put the cross-compiled files in the shared folder, you can copy the files using the command: **cp (file name) (the path of shared folder)**, i.e., **cp dwinqtdemo /mnt/hgfs/share/**

(2) Move the target file in the shared folder from the computer to the USB/SD card.

(3) Insert the USB into the development board.

(4) Open MobaXterm and connect. Enter the command: **cd /mnt/usb** to open the "usb" folder and select "sdax" folder. Copy or move the target file to the target directory (you can customize the folder to avoid clutter) using the command: **cp (target file)(folder)**,i.e. **cp dwinqtdemo /usr/bin/**.

## 3.6 Run the Dwinqtdemo Program

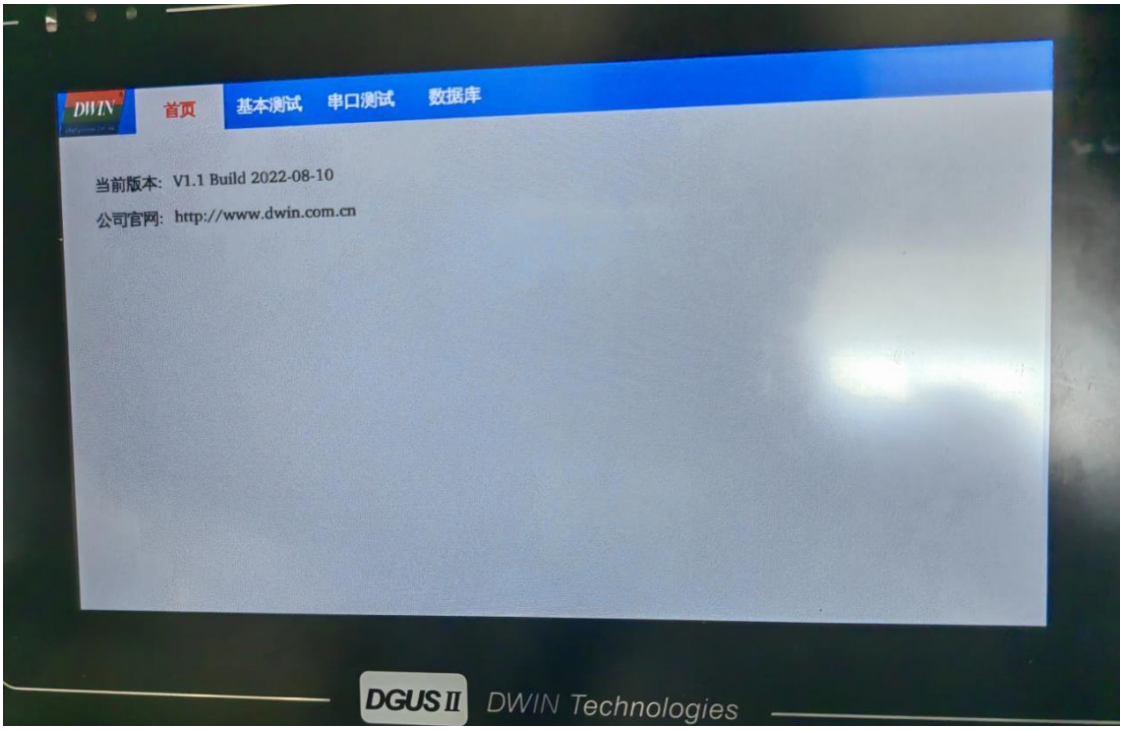
Configuration file **/etc/init.d/S99qtlinuxfb** should be modified to run the demo.

Enter the command: **vi/etc/init.d/S99qtlinuxfb** and move the cursor to the beginning of the "qttesttool" line. Press **i** to enter insert mode. Input "#" to comment out this line. Move the cursor to the end of this line and press enter. Input the absolute path of dwinqtdemo+ a blank space +&. Then press Esc to exit insert mode. Enter ": wq" to save the modification.

```
#!/bin/bash
vnclinuxfb_conf=/etc/vnclinuxfb.conf

vnclinuxfb_enable=`awk -F '=' '{a=1}a==1&&$1~/enable/{gsub(/[[:blank:]]*/,"",$2);

if [ $vnclinuxfb_enable == "1" ]; then
    export QT_QPA_PLATFORM="vnclinuxfb:fb=/dev/fb0:rotation=0"
else
    export QT_QPA_PLATFORM="linuxfb:fb=/dev/fb0:rotation=0"
fi

#qttesttool &
/usr/bin/dwinqtdemo &

if [ $vnclinuxfb_enable == "1" ]; then
    /etc/init.d/runfrpc &
fi
~
~
~
~
~
~
~
~
~
~
I runqt [Modified] 13/15 86%
```

You can run the demo using the command "./runqt".

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




## 3.7 Network Connection

Brightness adjustment, buzzer, and system time setting for the 40 series are Linux universal interfaces.

To check the current brightness:

cat /sys/class/backlight/backlight/brightness

Config the backlight to 0 (display off):

echo 0 > /sys/class/backlight/backlight/brightness

Config the backlight to 200:

echo 200 > /sys/class/backlight/backlight/brightness

## 3.8 System time setting

date -s "2023-03-01 11:07:09"

hwclock -w

# 4 Set up the build environment

## 4.1 The build environment of Linux SDK

Note:

(1) It is recommended to develop in the Ubuntu 18.04 system environment. If other system versions are used, the build environment may need to be adjusted accordingly.

(2) Compile with normal user, do not compile with root user authority.

### 4.1.1 Directory

```
.
├── app
├── buildroot                                    # Buildroot root filesystem build directory
├── build.sh -> device/rockchip/common/build.sh                    # compile script
├── debian                                       # Debian root filesystem compilation directory
├── device                                       # Compile related configuration files
├── docs                                                                    # Documentation
├── envsetup.sh -> buildroot/build/envsetup.sh
├── external
├── kernel                                                                  # Kernel
├── Makefile -> buildroot/build/Makefile
├── mkfirmware.sh -> device/rockchip/common/mkfirmware.sh        # Link script
├── prebuilts                                          # Cross compilation toolchain
├── rkbin
├── rkflash.sh -> device/rockchip/common/rkflash.sh                          # Flash script
├── tools                                                                   # Tools directory
└── u-boot                                  #U-Boot
```

### 4.1.2 Install Dependencies

Install directly on PC:

```
sudo apt-get install repo git ssh make gcc libssl-dev liblz4-tool \
expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support \
qemu-user-static live-build bison flex fakeroot cmake \
unzip device-tree-compiler python-pip ncurses-dev python-pyelftools
```

## 4.2 Compile Debian Firmware

This chapter introduces the compilation process of Debian firmware. It is recommended to develop under Ubuntu 18.04 system environment. If you use other system versions, you may need to adjust the compilation environment accordingly.

### 4.2.1 Compile SDK

#### 4.2.1.1 Configuration before compilation

In the **device/rockchip/rk356x/** directory, there are configuration files of different board types.

Return to SDK root directory and execute build.sh to select the configuration file:

```
./build.sh BoardConfig-rk3566-dwin.mk
```

The configuration file will be linked to **device/rockchip/.BoardConfig.mk**，check the file to verify whether the configuration is successful.

#### 4.2.1.2 Debian root filesystem

Change to the root filesystem directory:

```
cd debian
./build.sh
```

Create a link and link the filesystem to **linaro-rootfs.img**：

```
cd ..
ln -rsf debian/linaro-rootfs.img  rockdev/rootfs.img
```

#### 4.2.1.3 Automatic compilation

Fully automatic compilation will perform all compilation and packaging operations to generate complete RK firmware.

```
./build.sh
```

#### 4.2.1.4 Partial compilation

- Compile u-boot

```
./build.sh uboot
```

- Compile kernel

```
./build.sh kernel
```

- Compile recovery

```
./build.sh recovery
```

#### 4.2.1.5 Update link

Update each part of the mirror link to **rockdev/** directory:

```
./build.sh firmware
```

#### 4.2.1.6 Package the firmware

Pack the firmware, the generated complete firmware will be saved to the **rockdev/pack/** directory.

RK firmware is the firmware packaged in Rockchip's proprietary format, and can be flashed to eMMC or SD card with the tools provided by Rockchip.

```
./build.sh updateimg
```

## 4.3 Compile Buildroot firmware

This chapter introduces the compilation process of Buildroot firmware. It is recommended to develop in the Ubuntu 18.04 system environment. If you use other system versions, you may need to adjust the compilation environment accordingly.

### 4.3.1 Compile SDK

In the **device/rockchip/rk356x/** directory, there are configuration files of different board types.

Return to SDK root directory to select the configuration file:

```
./build.sh BoardConfig-rk3566-dwin.mk
```

The configuration file will be linked to **device/rockchip/.BoardConfig.mk**，check the file to verify whether the configuration is successful.

#### 4.3.1.1 Partial compilation

- Compile u-boot

```
./build.sh uboot
```

- Compile Kernel

```
./build.sh kernel
```

- Compile recovery

```
./build.sh recovery
```

- Compile Buildroot root filesystem

Compiling the Buildroot root filesystem will generate a compilation output directory in **buildroot/output** :

```
./build.sh rootfs
# Note: Make sure to compile the Buildroot root filesystem as a normal user to avoid unnecessary errors.
```

#### 4.3.1.2 Package the firmware

Update each part of the mirror link to the **rockdev**/ diectory：

```
./build.sh firmware
```

Pack the firmware, the generated complete firmware will be saved to the rockdev/pack/ directory.

```
./build.sh updateimg
```

## 4.4 Upgrade the firmware(40 series & 40ZOS-1 series)

### 4.4.1 Upgrade the firmware via SD card

To upgrade the firmware using an SD card, you need to use a tool on a computer to write the unified firmware onto the SD card. Currently, this operation is only supported on the Windows operating system.

Operation steps:

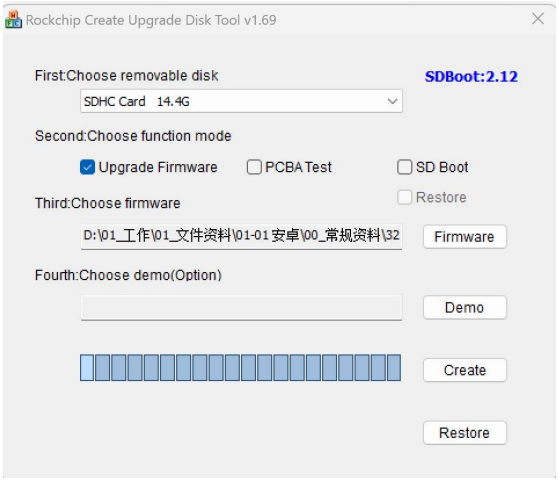
Run SDDiskTool_v1.69，check the “Upgrade Firmware” box and select the correct removable disk device.

Insert SD card into USB card reader and then into USB port of host computer.

Click button “Create” to make it and wait until it is finished.

Remove the SD card, insert it into the SD card slot of the motherboard, power on the board, it will start upgrading automatically.

After the upgrade, remove the SD card and restart the motherboard automatically to complete the whole process of firmware update.




### 4.4.2 Upgrade the firmware via Micro USB

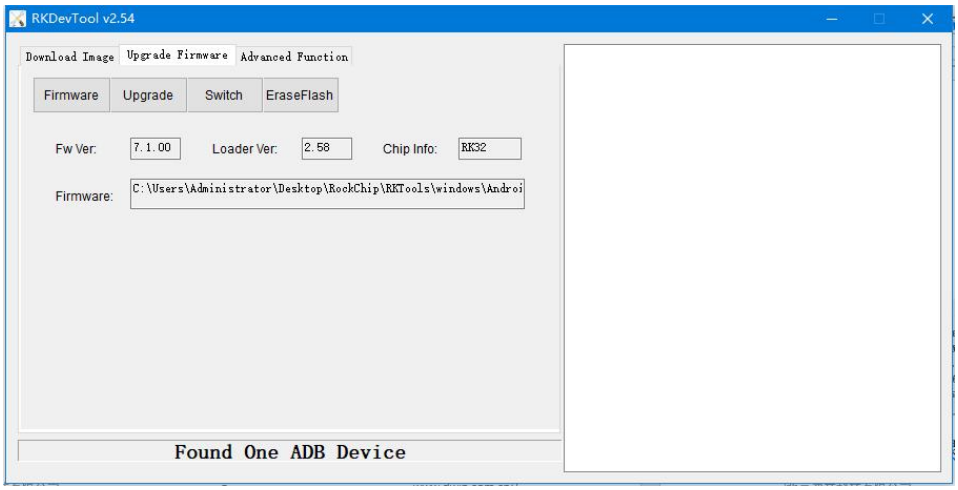
If the computer is being used for the first time to perform the burn-in process, you need to install the driver.

Please refer to the "USB Driver Installation Instructions" in the USB driver directory.

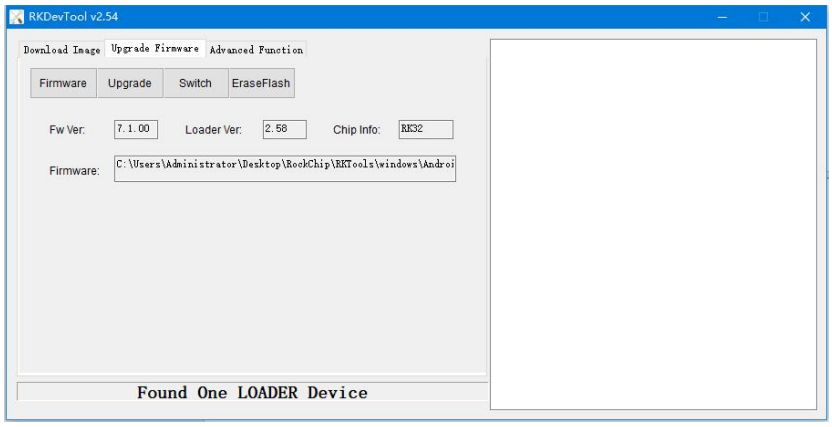
Operation steps:

Download the firmware you need to upgrade to the screen.

Open RKDevTool, select "Upgrade Firmware," and click on "Firmware" to choose the **.img** file to be burned.




While the device is powered off, press and hold the Recover button on the Android screen. First, connect the PC using a USB cable, and then connect the power supply (DC-12V). The following interface will appear. Click "Upgrade" to start the burn-in process. Once the burn-in is complete, the device will automatically restart.



# 5 Revision Records

| Rev | Revise Date | Content | Editor |
|---|---|---|---|
| 00 | 2023-2-20 | First Edition | Yu Yihe |
| 01 | 2023-3-17 | English version | Chen Lvzhi |
| 02 | 2024-3-20 | Added examples about brightness adjustment and system time settings | Chen Yan |
| 03 | 2024-7-25 | Add QT creator compile configurate, compile Linux 4.19 firmware, and chapter 1. | Chen Yan |
| 04 | 2024-11-07 | Add chapter 4.4 (40 series & 40ZOS-1 series) | Chen Xian |
| 05 | 2025-02-08 | Add Boot logo update method and other QT screen configuration options.<br>Add examples of Debian screen configuration. | Chen Xian |
| 06 | 2025-02-13 | Update the location of "Run dwinqtdemo" | Chen Xian |
| 07 | 2025-02-17 | Delete the part of "obtaining the SDK by downloading via repo" , Sync code and Download SDK | Chen Xian |
| 08 | 2025-03-10 | Update “Modify the Startup LOGO” | Chen Xian |

**Disclaimer: The product design is subject to alternation and improvement without prior notice.**

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!