DWIN Linux Screen Development Guide

(40 Series & 40ZOS-1 Series)

Contents

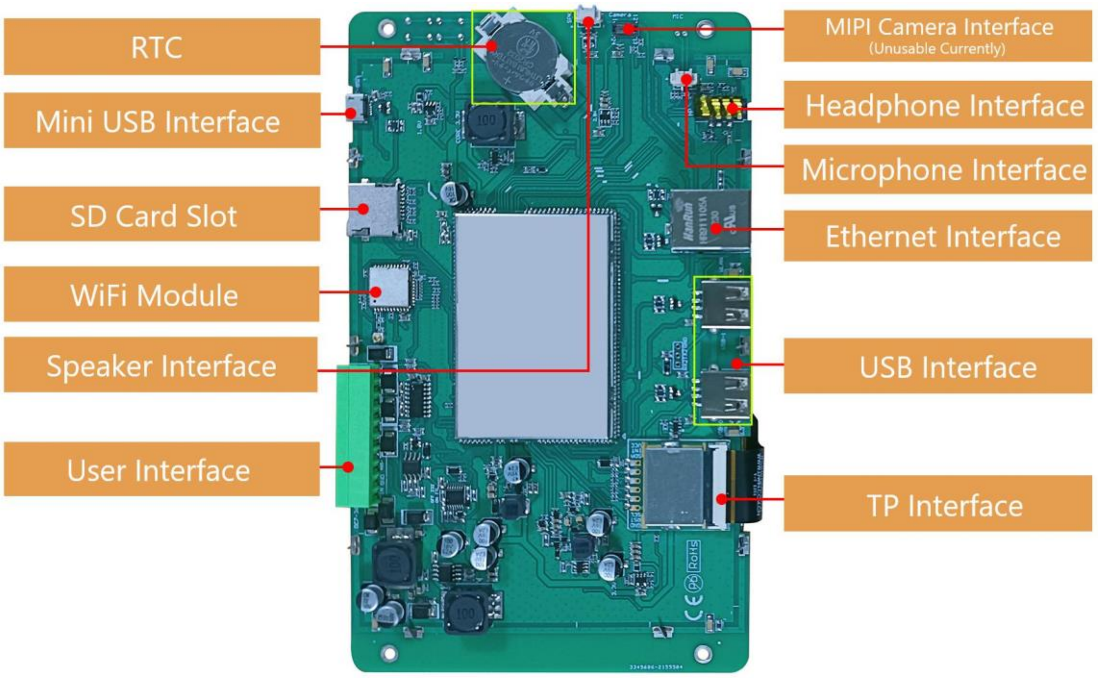
1 Product Introduction

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**

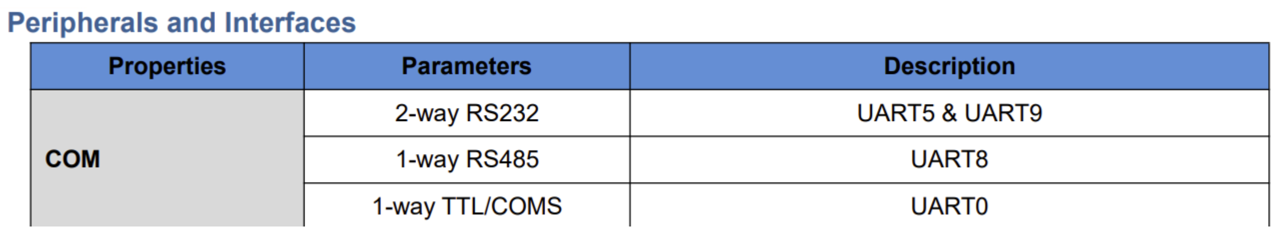
  China and India: LUAT Air780EI
  Europe: QUECTEL EC200A-EU
  Australia: QUECTEL EC200A-AU

- **Camera**

  Support camera with USB interface

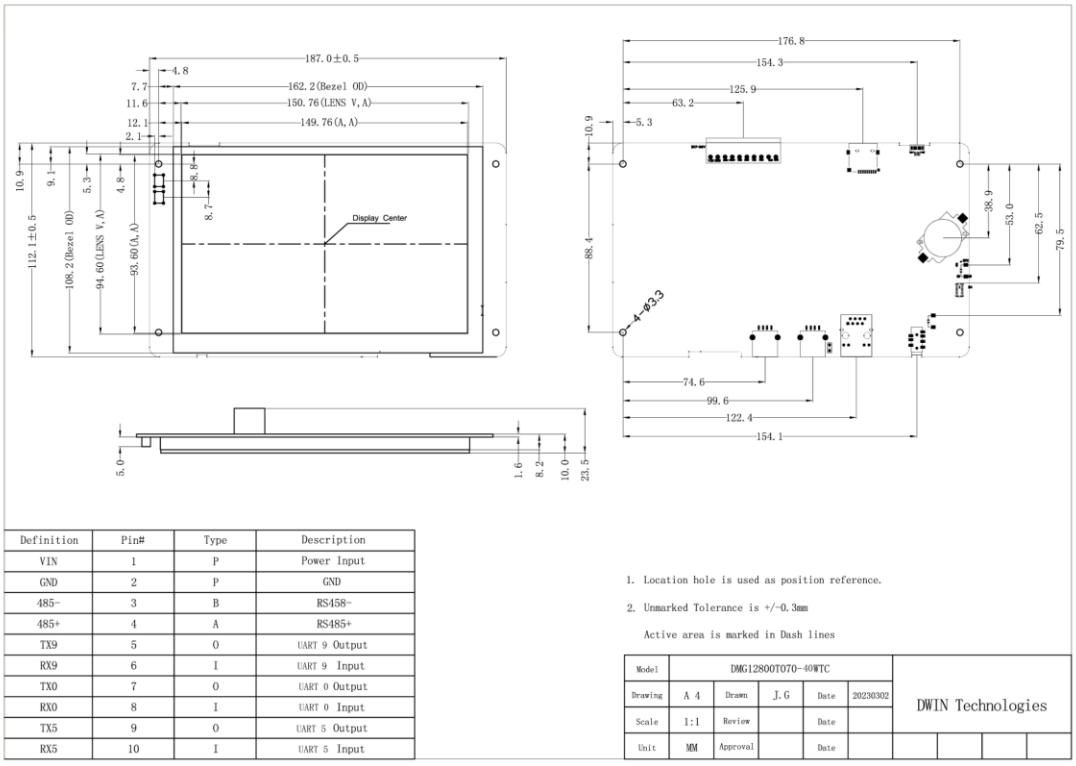
1.2 Wiring

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

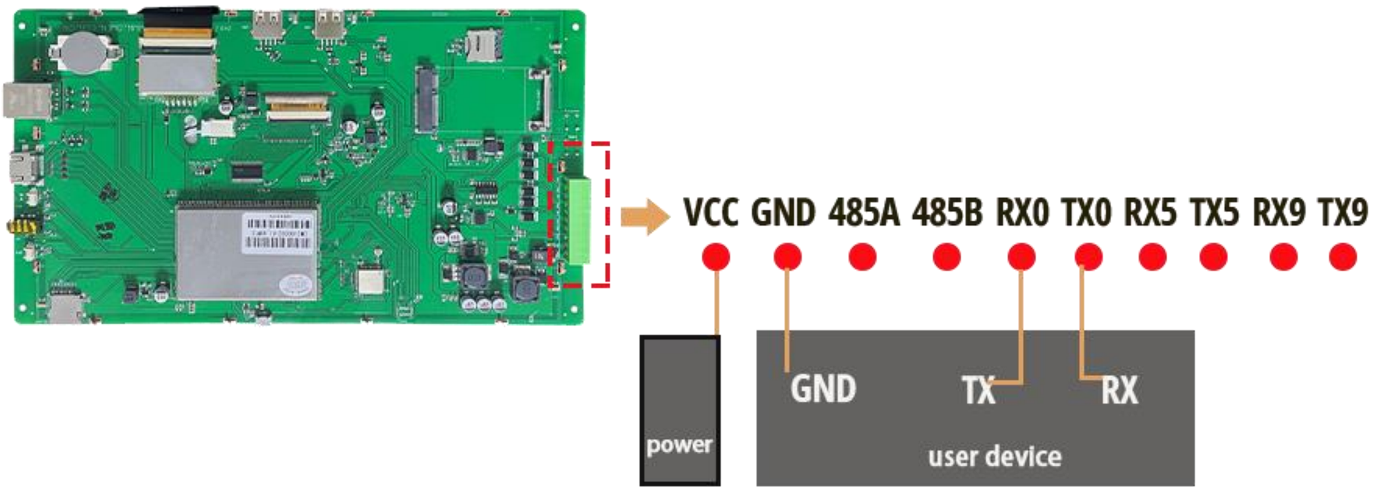
Model	DMG12800T070-40WTC					DWIN Technologies
Drawing	A 4	Drawn	J.G	Date	20230302	
Scale	1:1	Review		Date		
Unit	MM	Approval		Date		

1.2.1 Hardware Connection

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

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

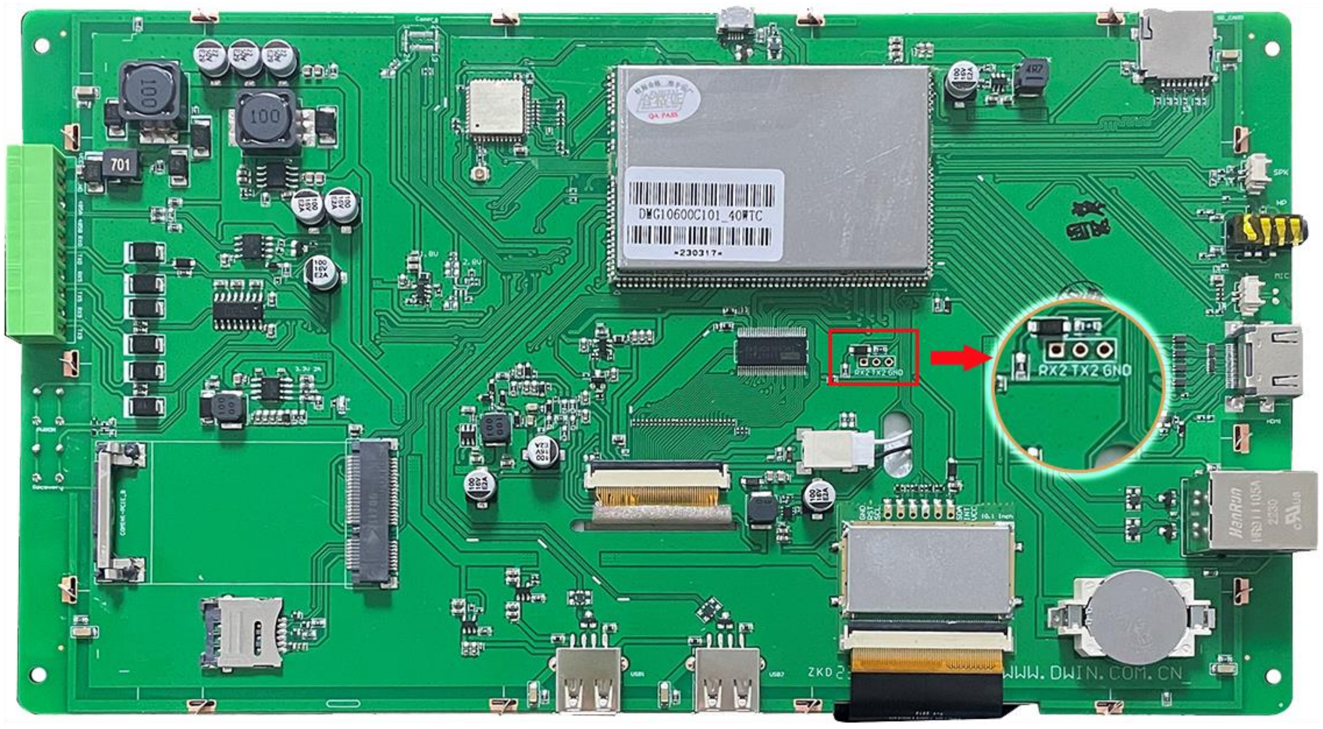
RXD，Receive, connect to RX pin of the user device.



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.

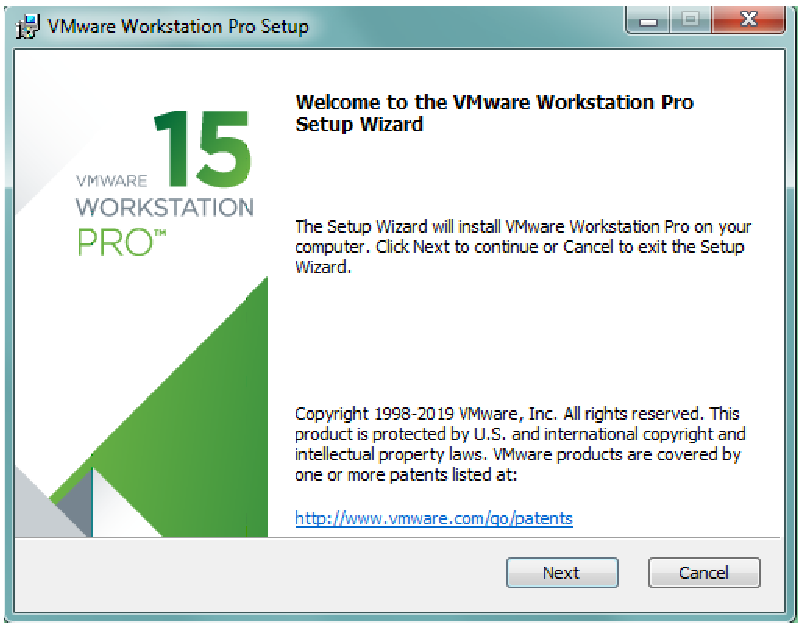
Note:

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



Welcome to the VMware Workstation Pro Setup Wizard

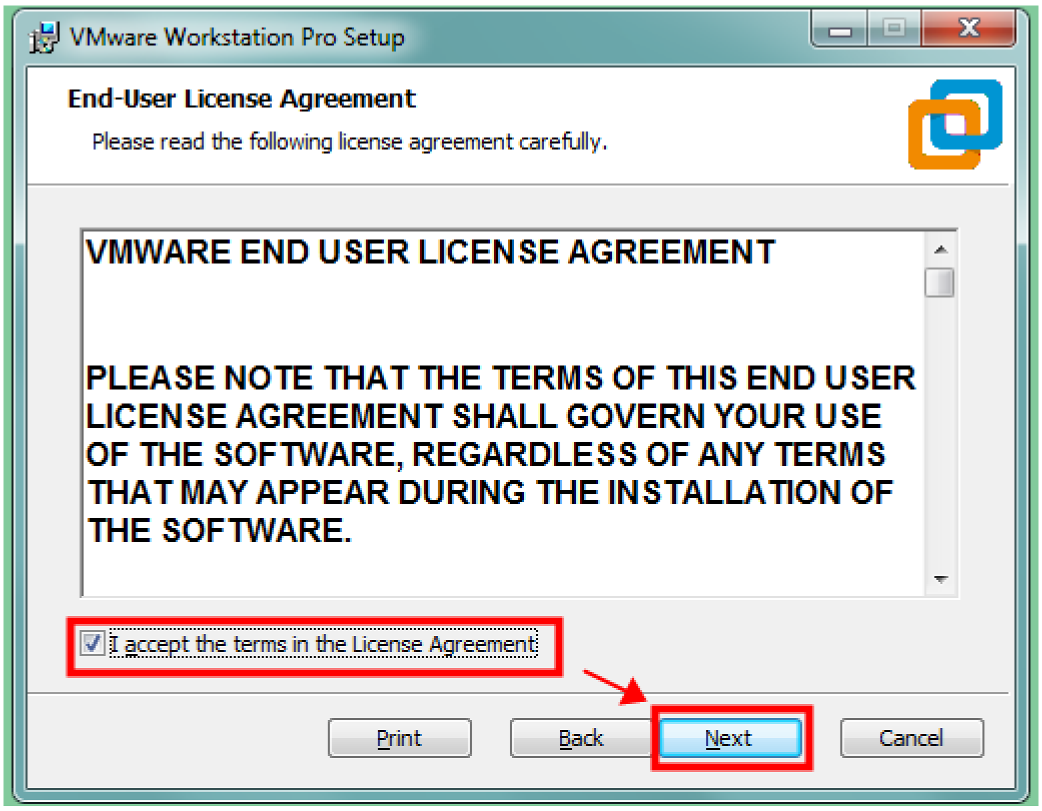
The Setup Wizard will install VMware Workstation Pro on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

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）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”.


VMware Workstation Pro Setup

End-User License Agreement

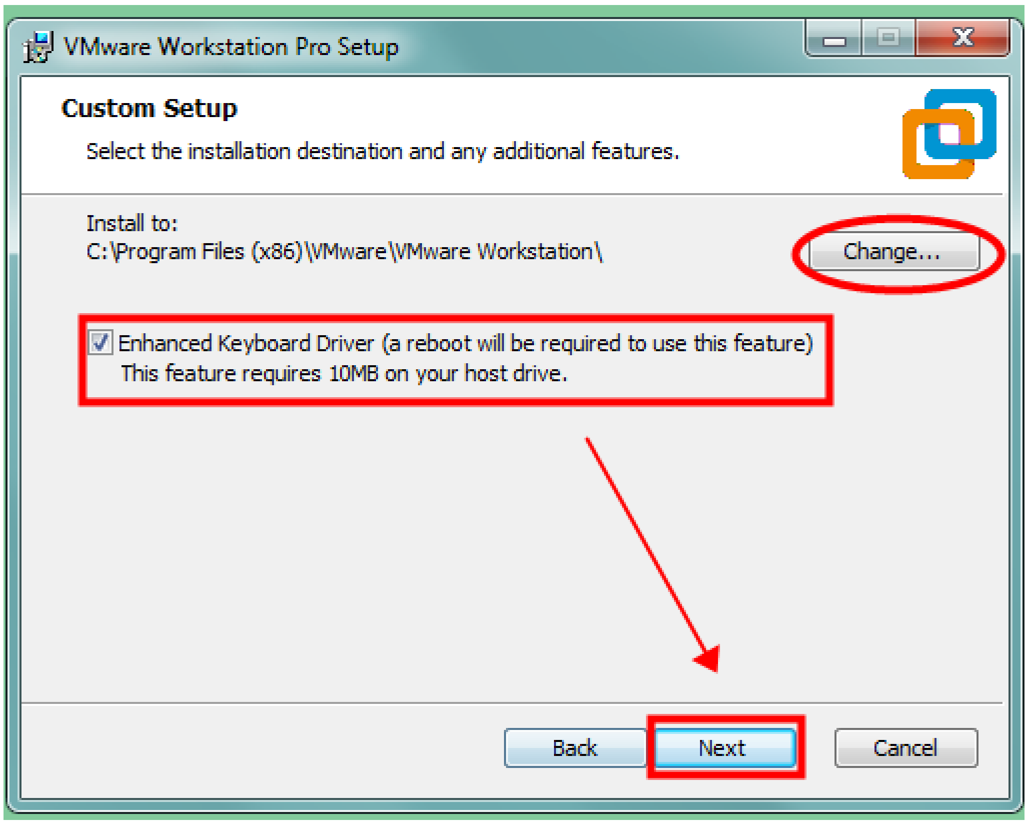
Please read the following license agreement carefully.

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

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


VMware Workstation Pro Setup

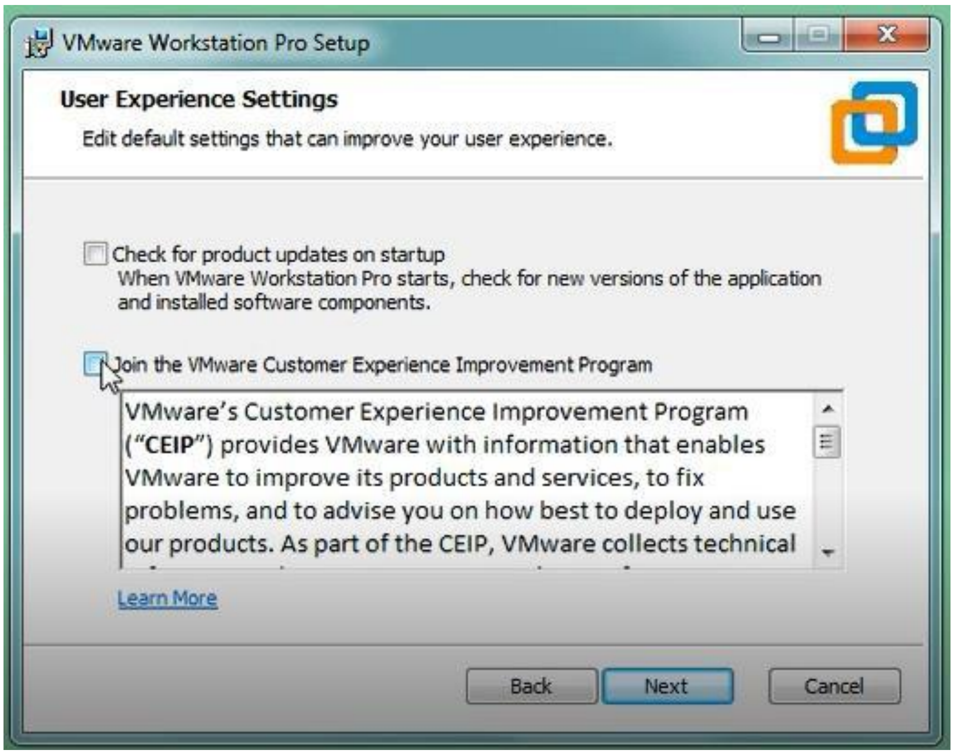
Custom Setup

Select the installation destination and any additional features.

Install to:
C:\Program Files (x86)\VMware\VMware Workstation\

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

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



User Experience Settings

Edit default settings that can improve your user experience.

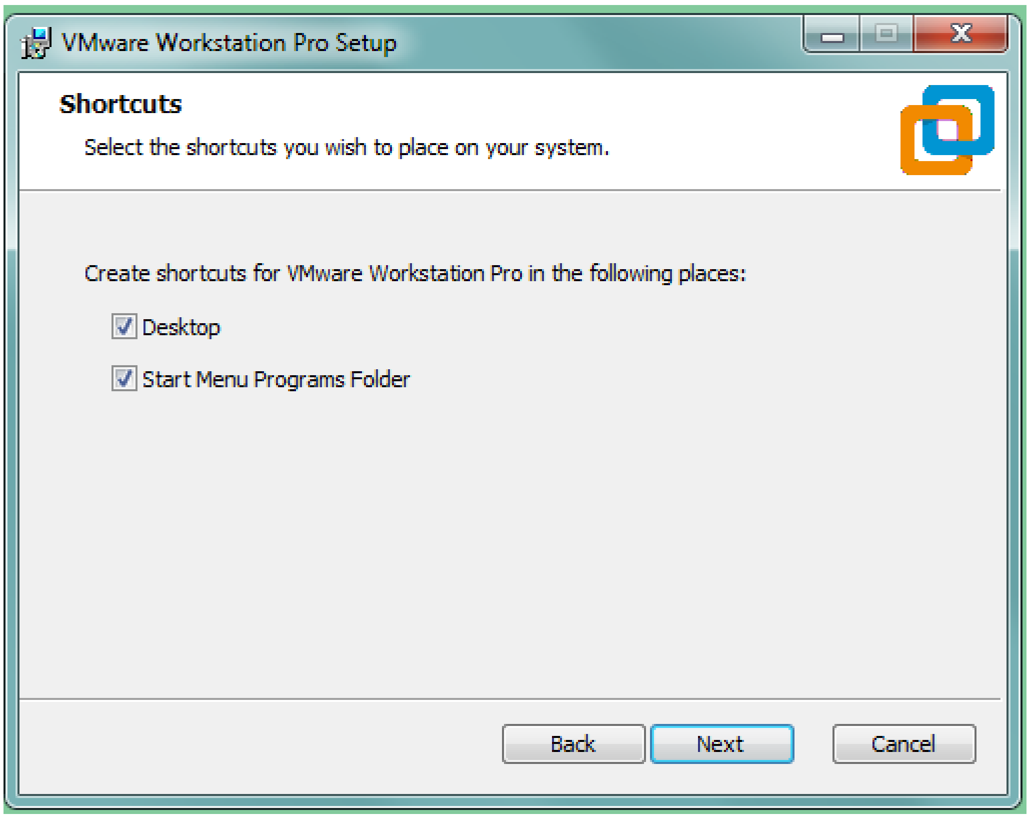
☐ Check for product updates on startup
When VMware Workstation Pro starts, check for new versions of the application and installed software components.

☐ 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

Learn More

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



Shortcuts

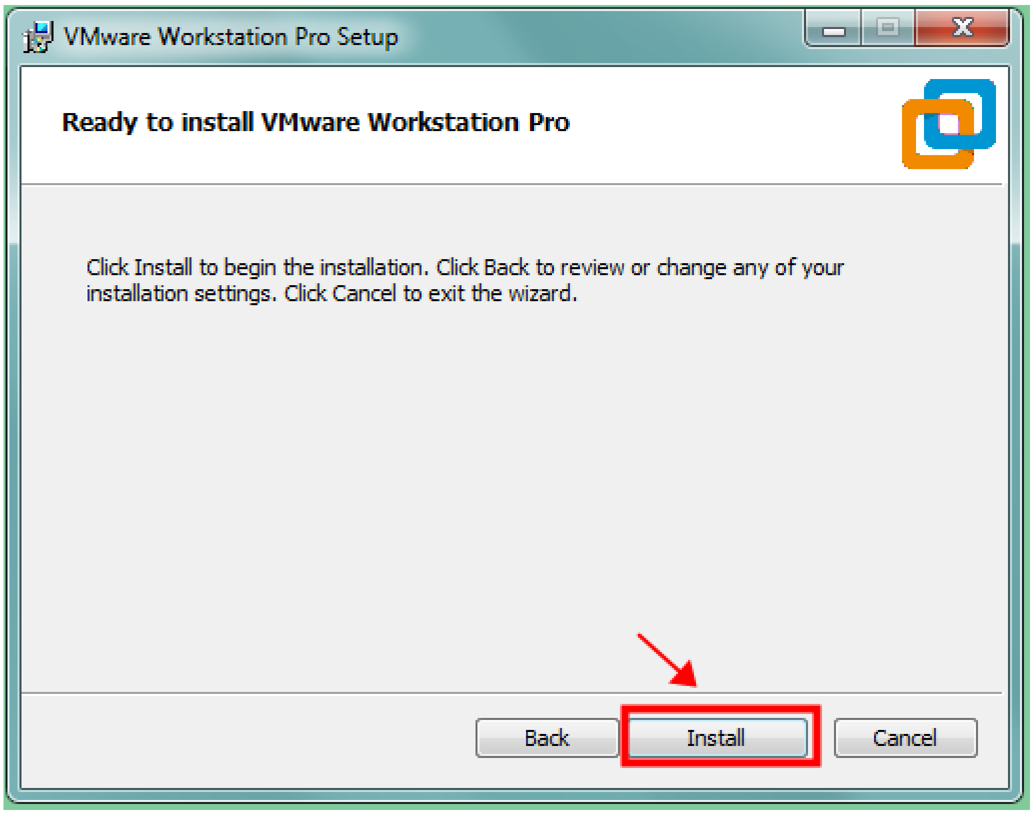
Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Workstation Pro in the following places:

☑ Desktop

☑ Start Menu Programs Folder

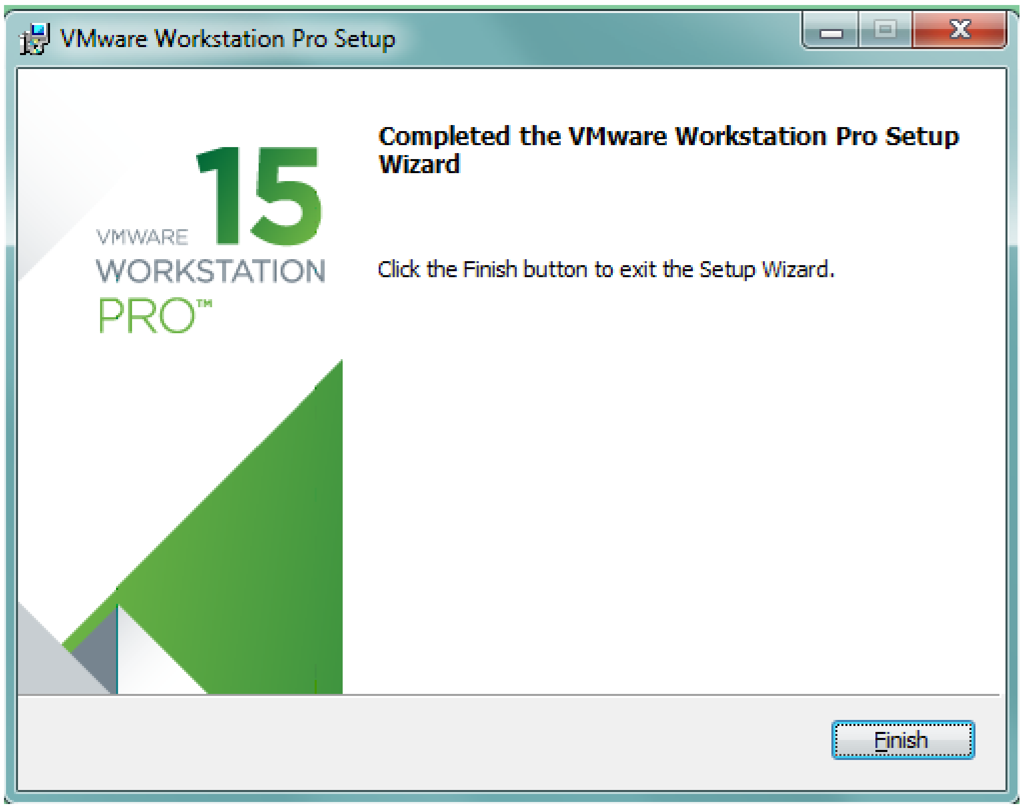
（7）Click "**Install**" to start the installation.


VMware Workstation Pro Setup

Ready to install VMware Workstation Pro

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

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


VMware Workstation Pro Setup

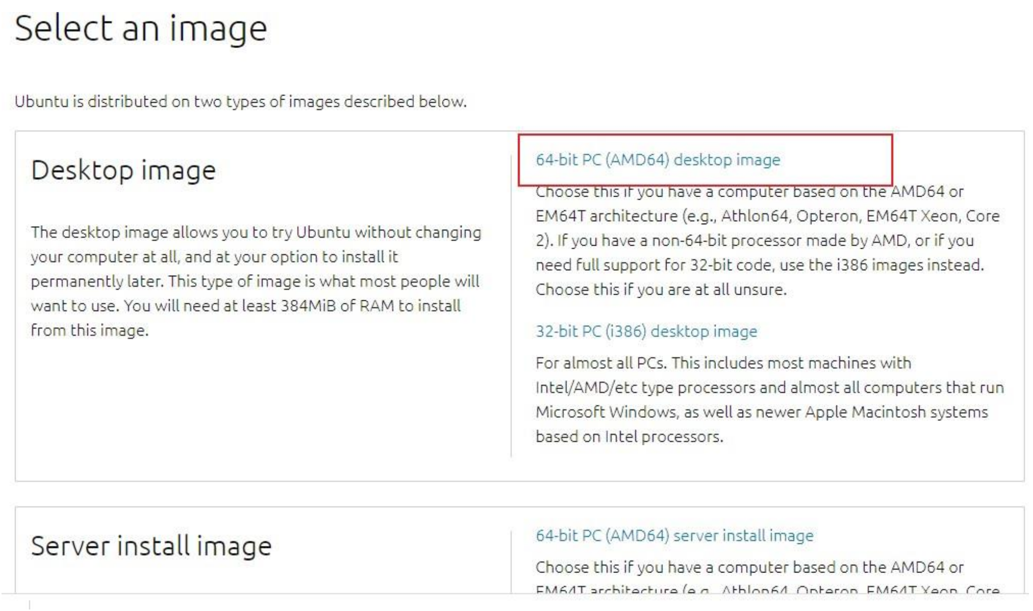
Completed the VMware Workstation Pro Setup Wizard

Click the Finish button 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”.



Select an image

Ubuntu is distributed on two types of images described below.

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). If you have a non-64-bit processor made by AMD, or if you need full support for 32-bit code, use the i386 images instead. 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.

Server install image

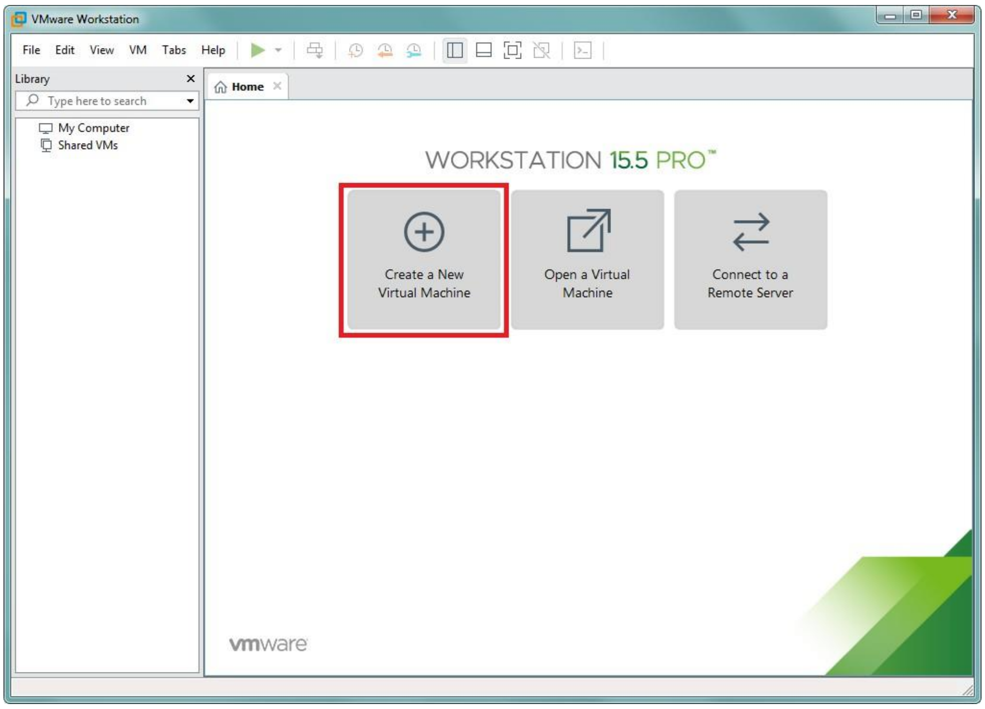
64-bit PC (AMD64) server install image

Choose this if you have a computer based on the AMD64 or

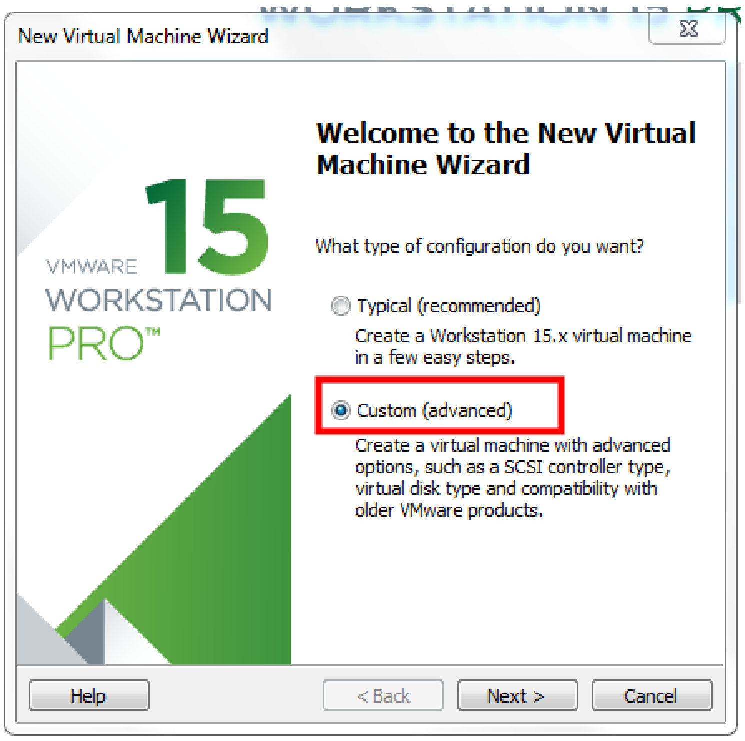
2.1.5 Install Ubuntu

（1）Open VMware Workstation Pro.

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



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

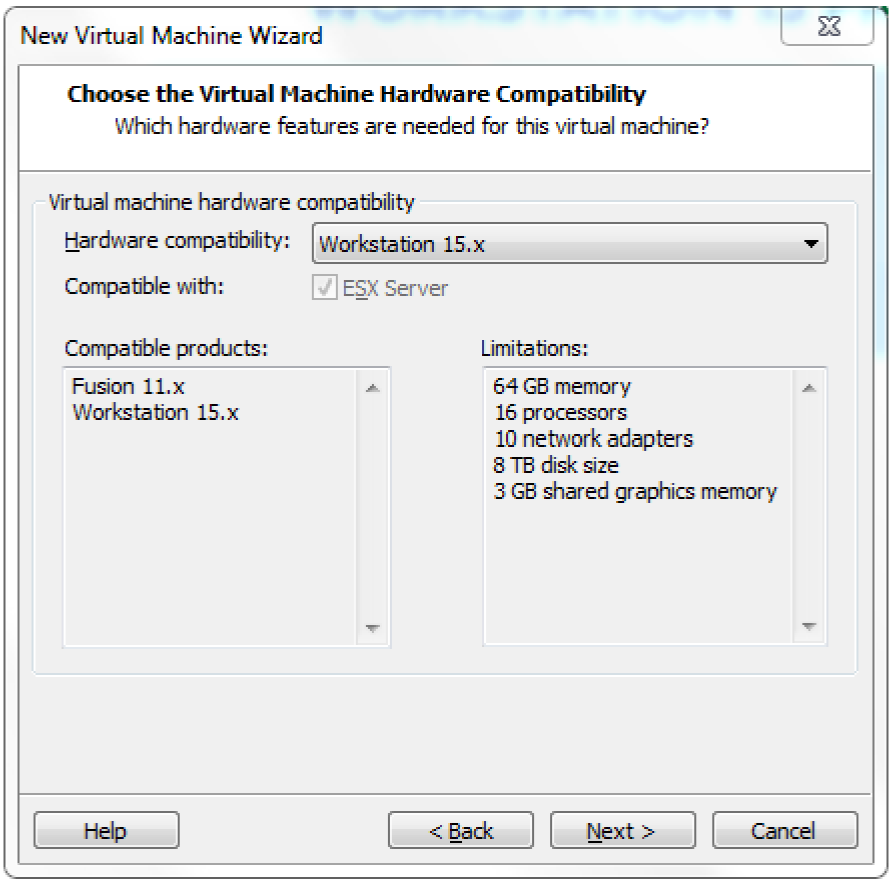

New Virtual Machine Wizard

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.


New Virtual Machine Wizard

Choose the Virtual Machine Hardware Compatibility
Which hardware features are needed for this virtual machine?

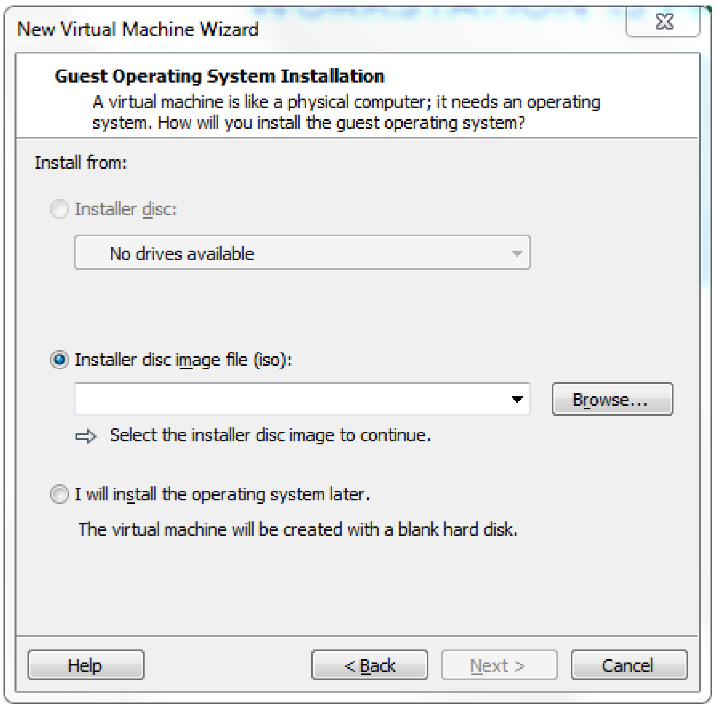
Virtual machine hardware compatibility
Hardware compatibility: Workstation 15.x
Compatible with: ESX Server

Compatible products:
Fusion 11.x
Workstation 15.x

Limitations:
64 GB memory
16 processors
10 network adapters
8 TB disk size
3 GB shared graphics memory

（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**”.


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:

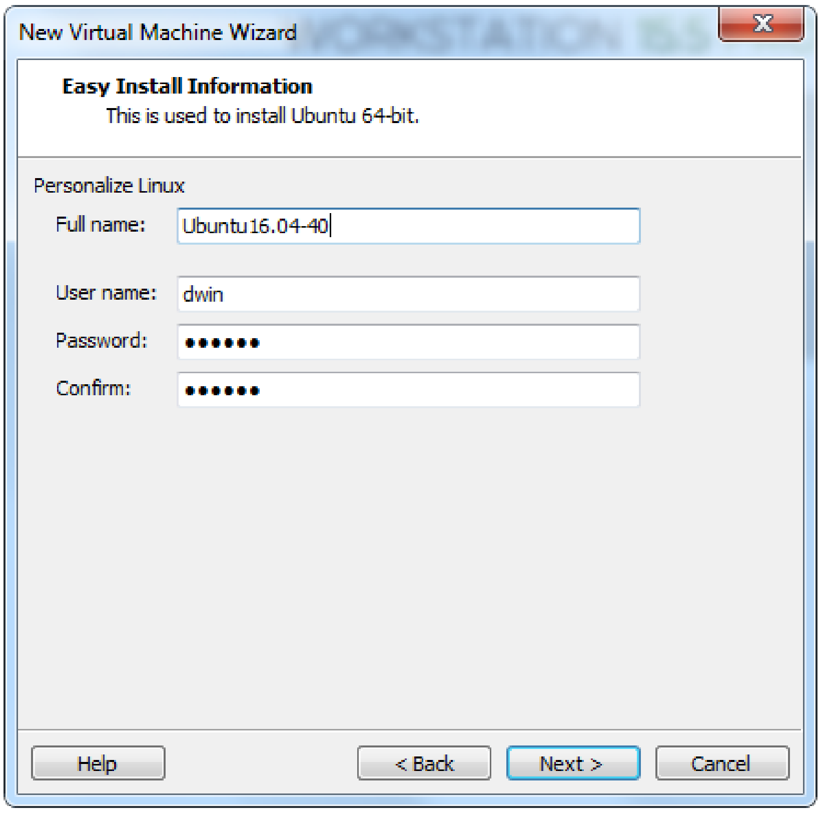
Installer disc: No drives available

Installer disc image file (iso): [✓]

Select the installer disc image to continue.

I will install the operating system later.
The virtual machine will be created with a blank hard disk.

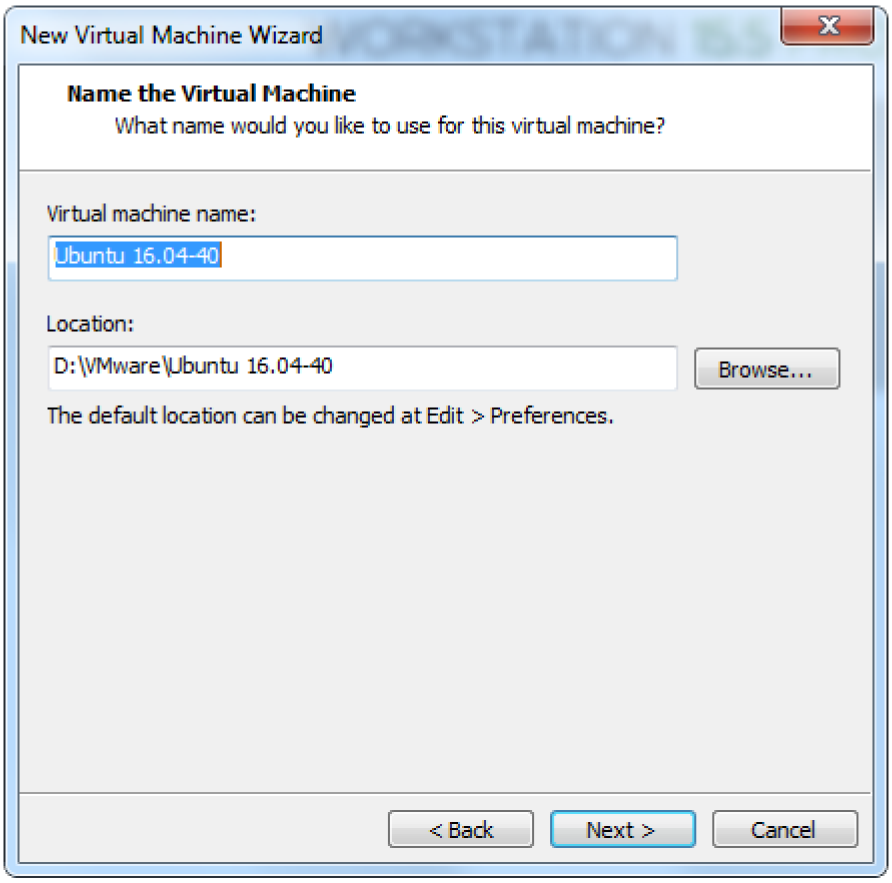
（7）Enter the custom name and password. The password is the Ubuntu login password and sudo privilege password. Click “**Next**”.


New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu 64-bit.

Personalize Linux
Full name: Ubuntu16.04-40
User name: dwin
Password: ●●●●●●
Confirm: ●●●●●●

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



New Virtual Machine Wizard

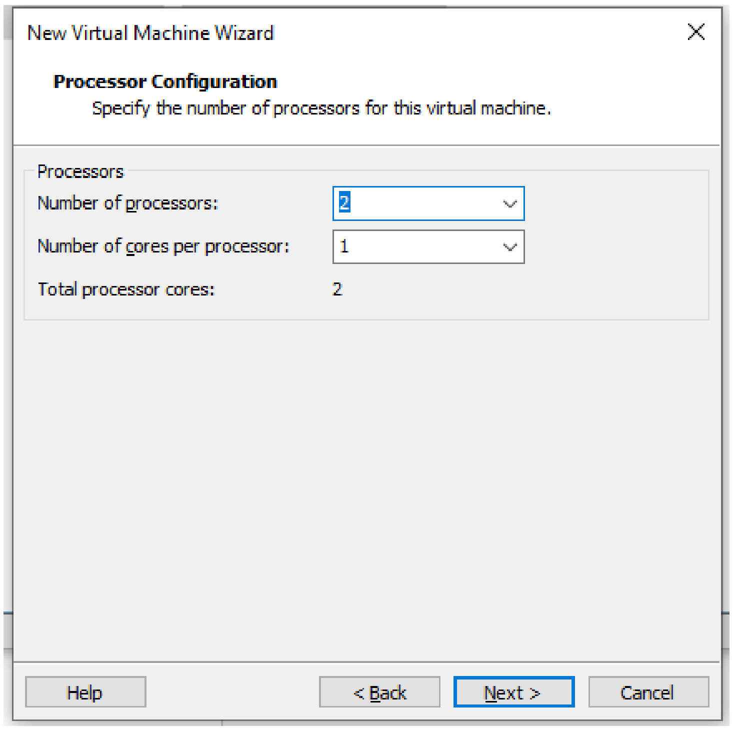
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
Ubuntu 16.04-40

Location:
D:\VMware\Ubuntu 16.04-40

The default location can be changed at Edit > Preferences.

（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**”.

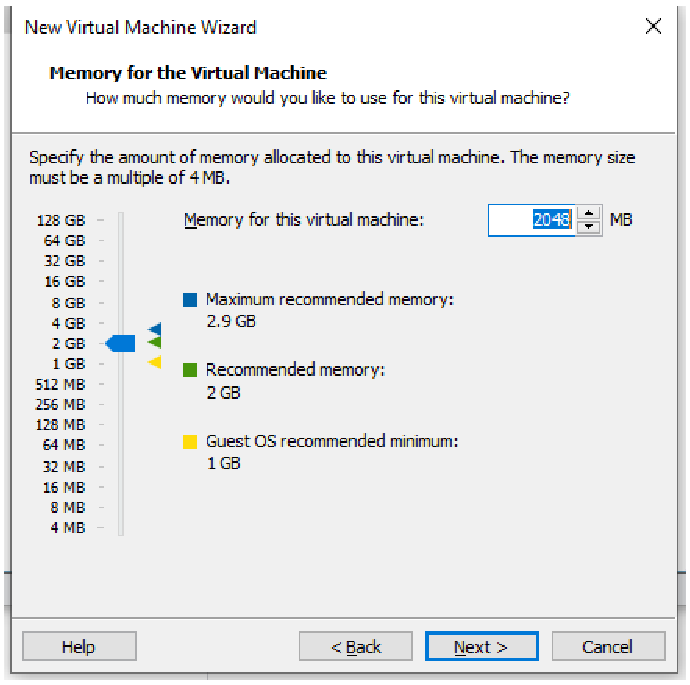


New Virtual Machine Wizard

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

（9）The default operating memory is 2G (enough and changeable), 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.

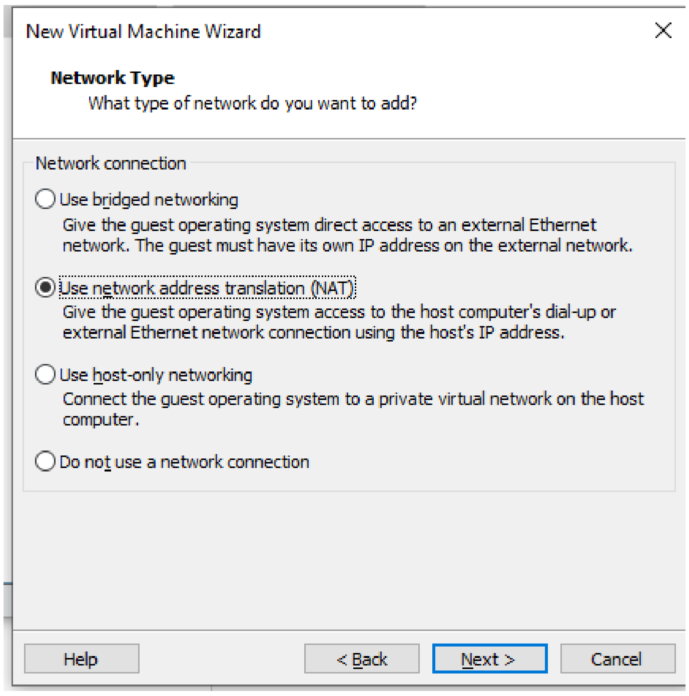
Memory for this virtual machine: 2048 MB

🟦 Maximum recommended memory: 2.9 GB

🟩 Recommended memory: 2 GB

🟨 Guest OS recommended minimum: 1 GB

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


New Virtual Machine Wizard

Network Type

What type of network do you want to add?

Network connection

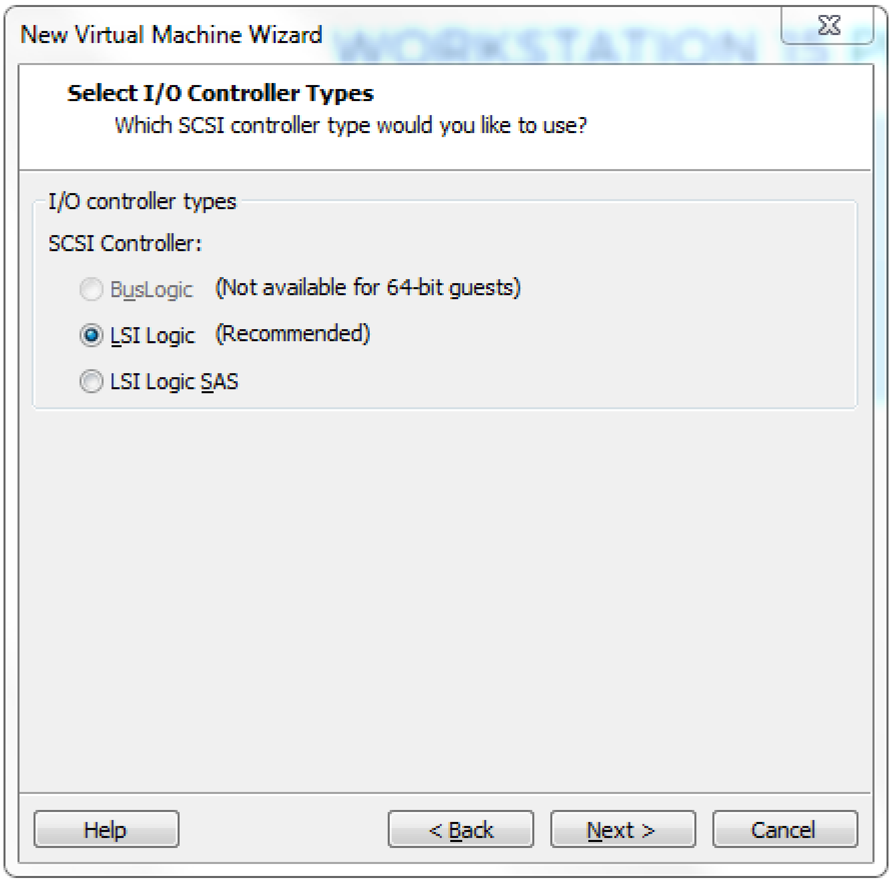
○ Use bridged networking
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.

● 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

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


New Virtual Machine Wizard

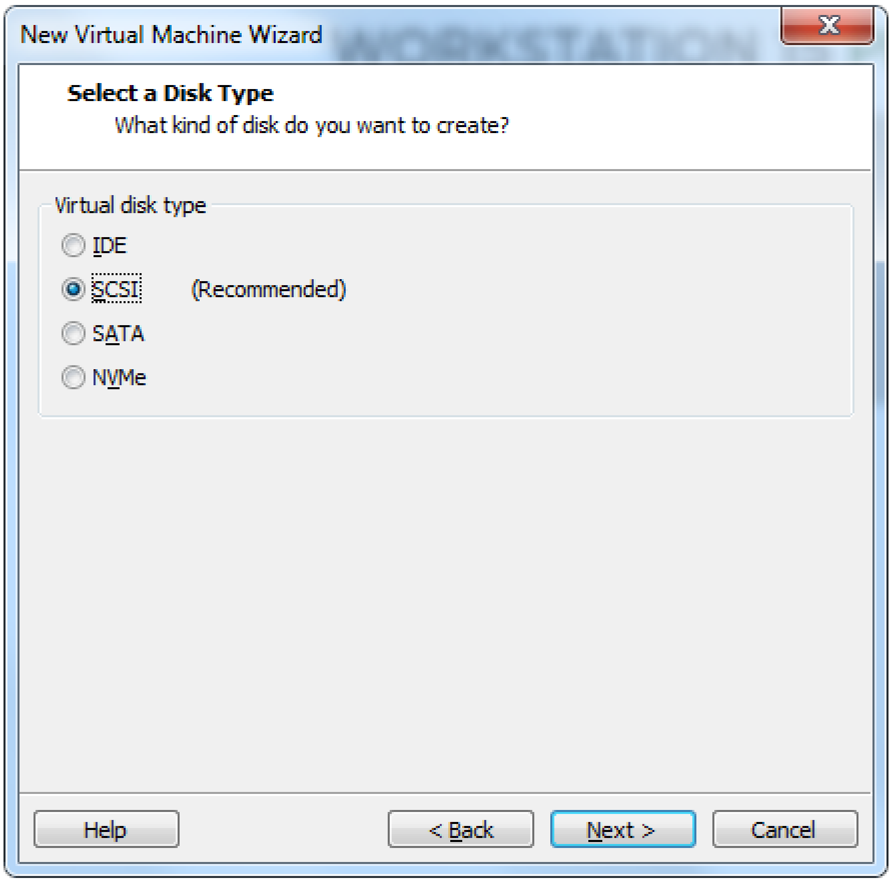
Select I/O Controller Types

Which SCSI controller type would you like to use?

I/O controller types

SCSI Controller:

- ○ BusLogic (Not available for 64-bit guests)
- ● LSI Logic (Recommended)
- ○ LSI Logic SAS


New Virtual Machine Wizard

Select a Disk Type

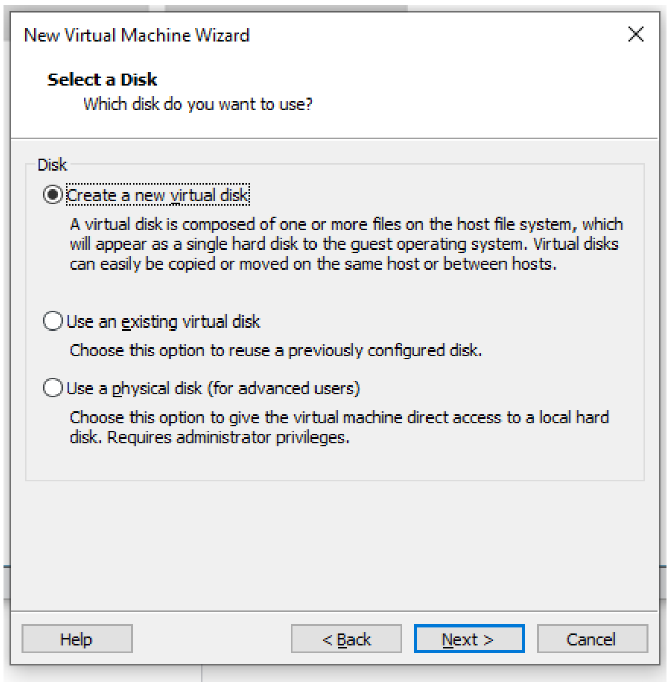
What kind of disk do you want to create?

Virtual disk type

- ○ IDE
- ● SCSI (Recommended)
- ○ SATA
- ○ NVMe

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

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



New Virtual Machine Wizard

Select a Disk
Which disk do you want to use?

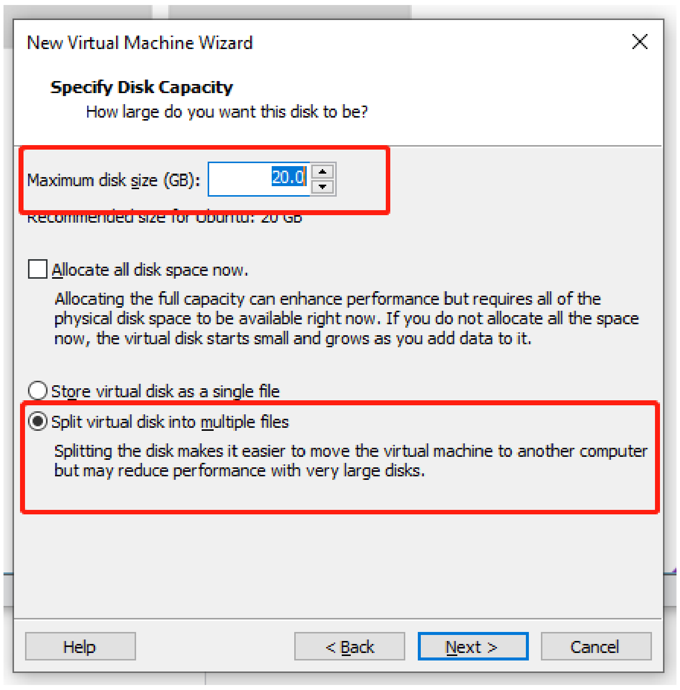
Disk

[● 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.

（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).



New Virtual Machine Wizard

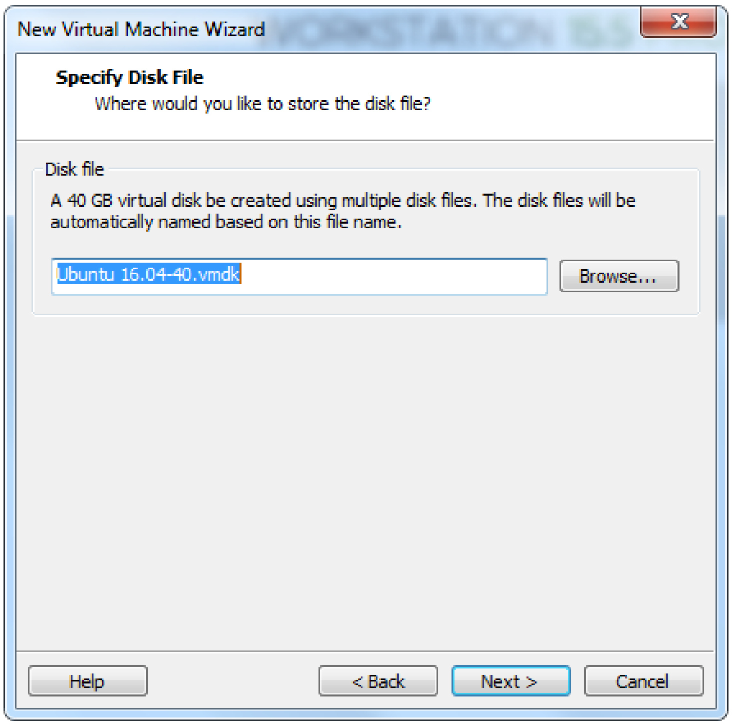
Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB): 20.0
Recommended size for Ubuntu: 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.

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

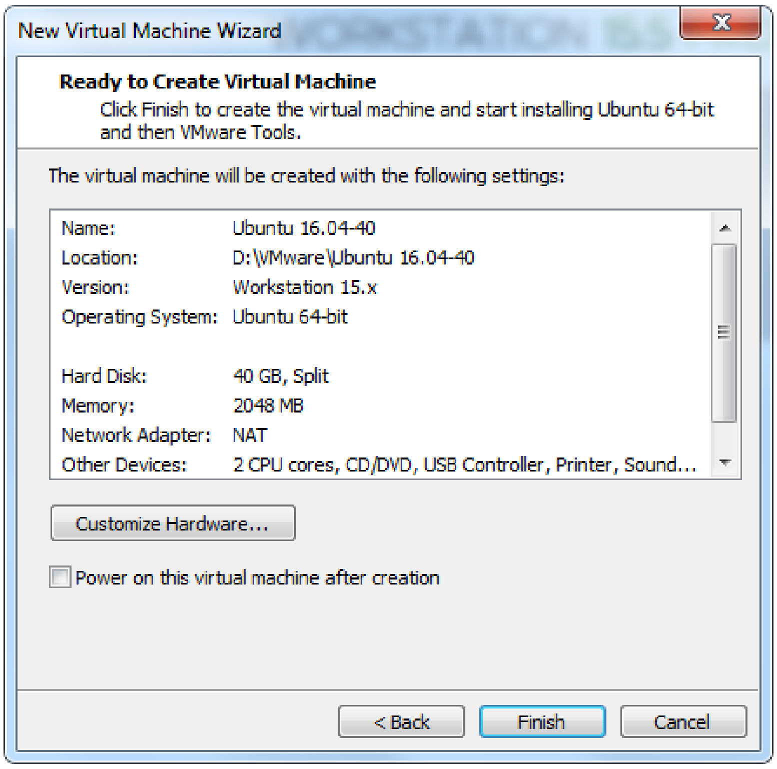

New Virtual Machine Wizard

Specify Disk File
Where would you like to store the disk file?

Disk file
A 40 GB virtual disk be created using multiple disk files. The disk files will be automatically named based on this file name.

Ubuntu 16.04-40.vmdk

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


New Virtual Machine Wizard

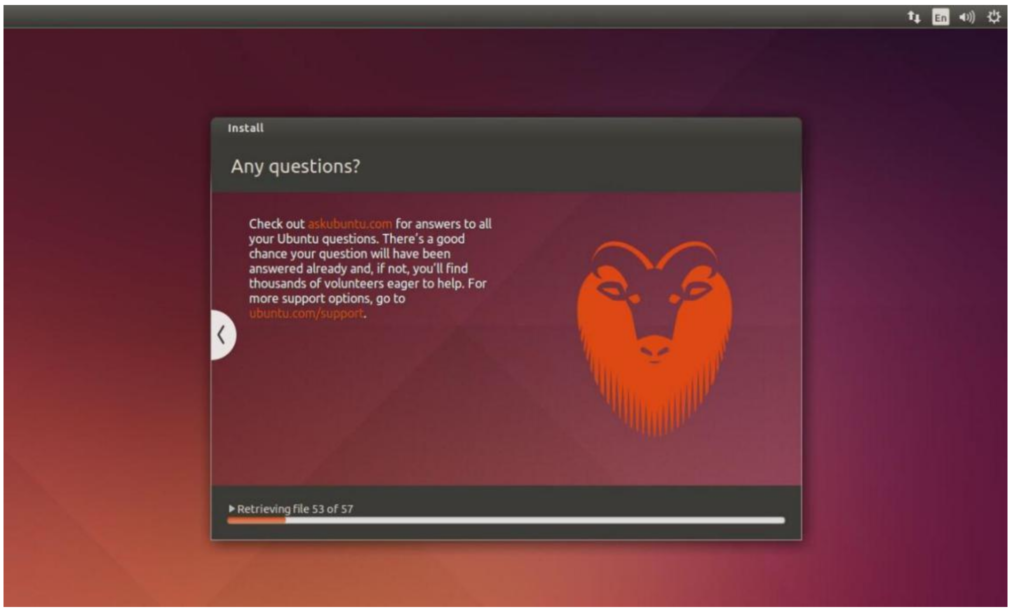
Ready to Create Virtual Machine
Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

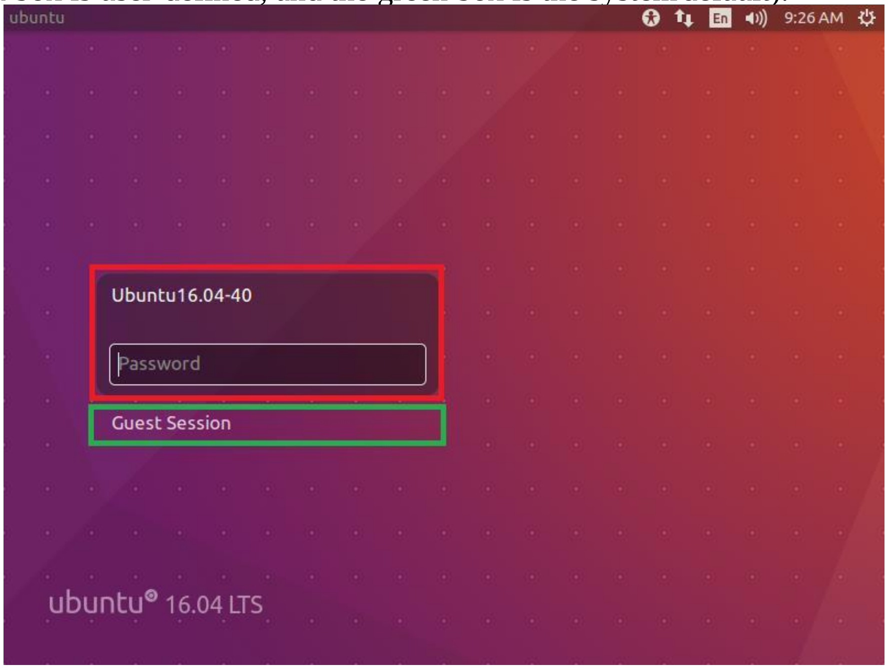
Name:	Ubuntu 16.04-40
Location:	D:\VMware\Ubuntu 16.04-40
Version:	Workstation 15.x
Operating System:	Ubuntu 64-bit
Hard Disk:	40 GB, Split
Memory:	2048 MB
Network Adapter:	NAT
Other Devices:	2 CPU cores, CD/DVD, USB Controller, Printer, Sound...

☐ Power on this virtual machine after creation

（17）Wait for a while.


Install
Any questions?
Check out askubuntu.com for answers to all your Ubuntu questions. There's a good chance your question will have been answered already and, if not, you'll find thousands of volunteers eager to help. For more support options, go to ubuntu.com/support.
Retrieving file 53 of 57

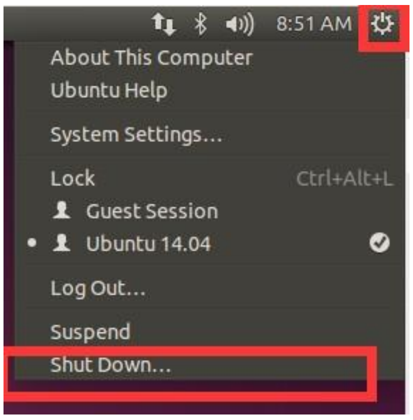
（18）When this page appears, the installation of Ubuntu is complete (Note: The login interface has two user login entries. The red box is user-defined, and the green box is the system default).



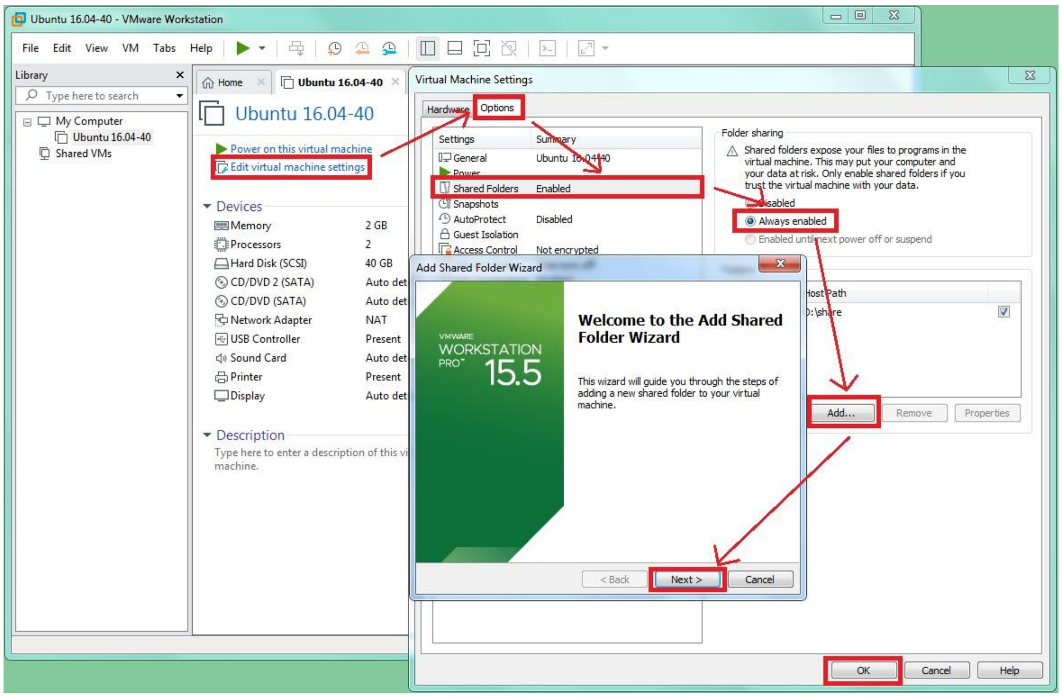
（19）Next, we’ll start configuring some of the required settings for Ubuntu.

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".


Ubuntu 16.04-40 - VMware Workstation

Ubuntu 16.04-40

Power on this virtual machine
Edit virtual machine settings

Devices

Memory	2 GB
Processors	2
Hard Disk (SCSI)	40 GB
CD/DVD 2 (SATA)	Auto det
CD/DVD (SATA)	Auto det
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto det
Printer	Present
Display	Auto det

Description
Type here to enter a description of this vi machine.

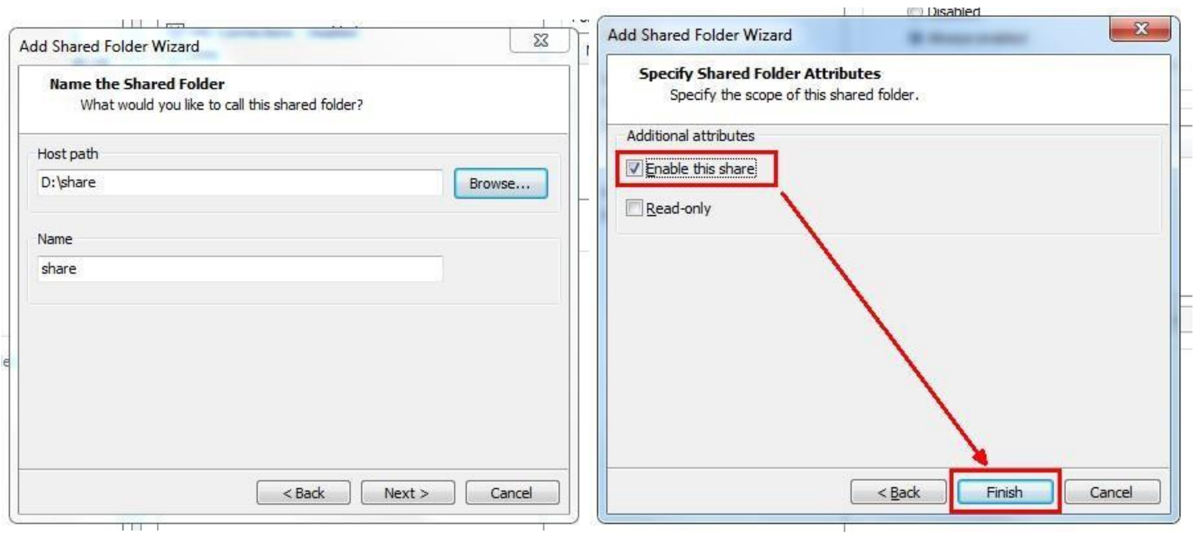
Virtual Machine Settings

Settings	Summary
General	Ubuntu 16.04-40
Power	
Shared Folders	Enabled
Snapshots	
AutoProtect	Disabled
Guest Isolation	
Access Control	Not encrypted

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

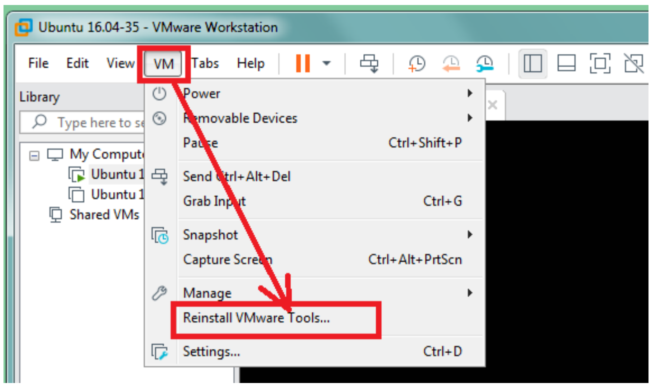
Host Path
D:\share

Add Shared Folder Wizard
Welcome to the Add Shared Folder Wizard
This wizard will guide you through the steps of adding a new shared folder to your virtual machine.

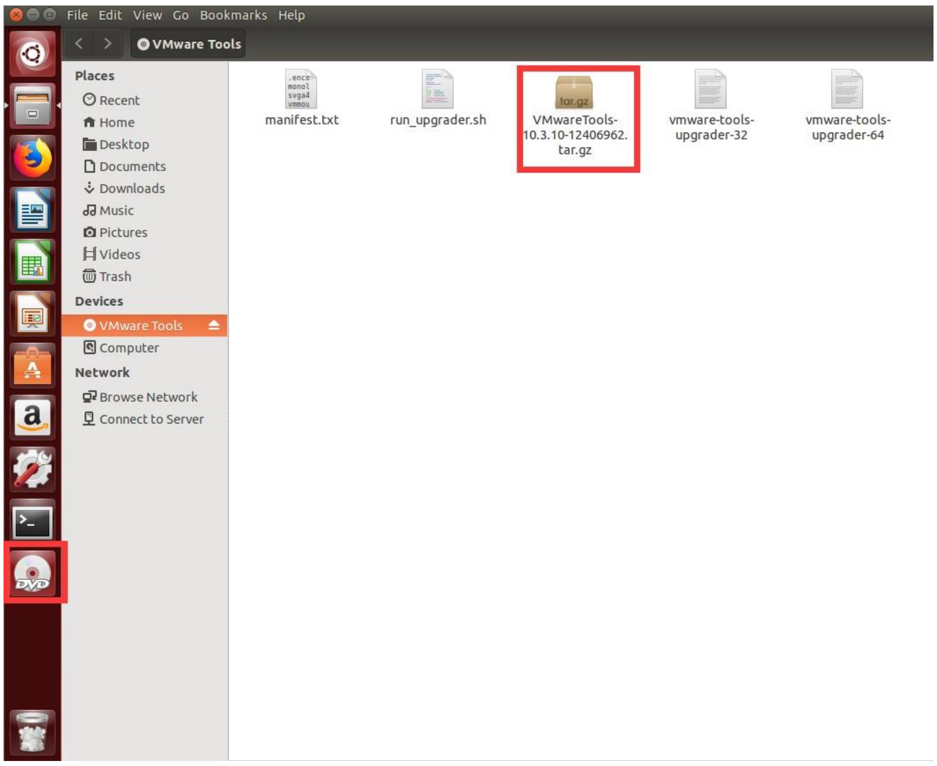

Add Shared Folder Wizard
Name the Shared Folder
What would you like to call this shared folder?
Host path: D:\share
Name: share

Add Shared Folder Wizard
Specify Shared Folder Attributes
Specify the scope of this shared folder.
Additional attributes
[✓] Enable this share
[ ] Read-only

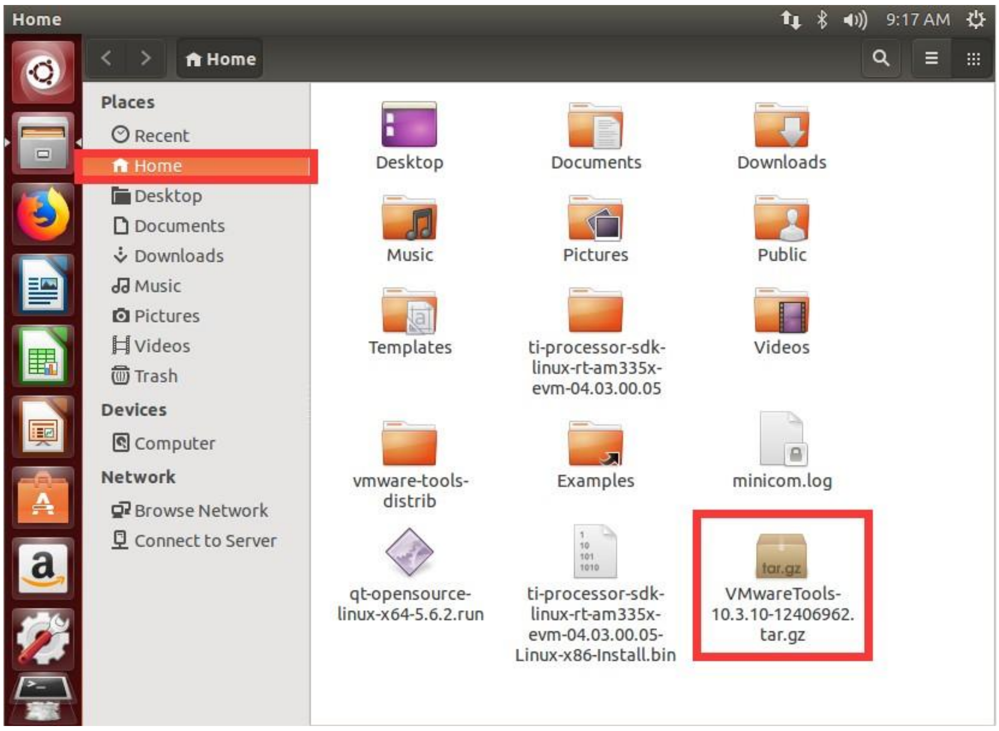
(3) Click "**Power on this virtual machine**" to start the virtual machine. Click "**VM**" -> "**Install VMware Tools**" (Note: The 'Install VMware Tools' is only selectable after powering on. In the example, VMware Tools have already been installed, so it shows 'Reinstall VMware Tools').


Ubuntu 16.04-35 - VMware Workstation
File Edit View VM Tabs Help
Library
Power
Removable Devices
Pause Ctrl+Shift+P
Send Ctrl+Alt+Del
Grab Input Ctrl+G
Snapshot
Capture Screen Ctrl+Alt+PrtScn
Manage
Reinstall VMware Tools...
Settings... Ctrl+D

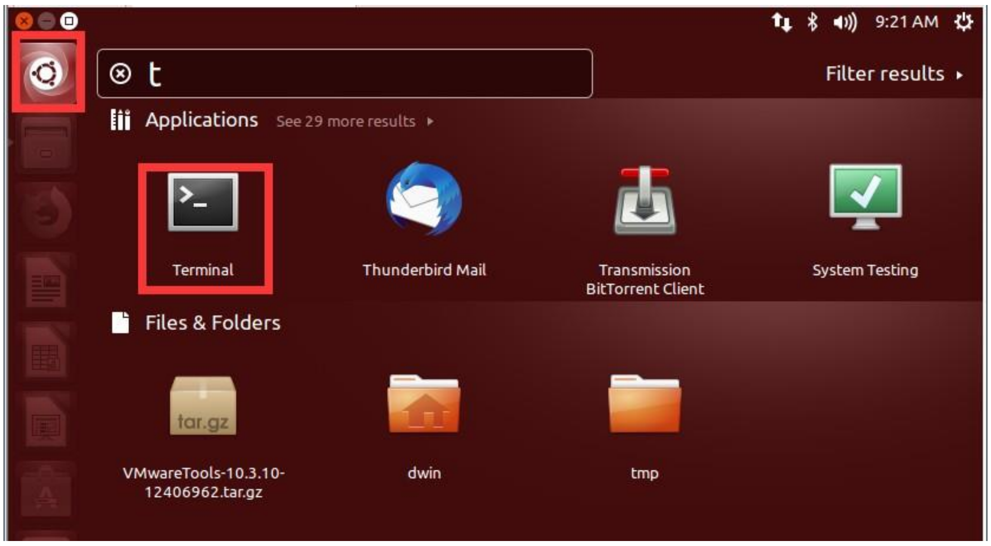
(4) Click the "**DVD**" icon and open it to see a tar file "**VMwareTools-10.3.10-12406962.tar.gz**".



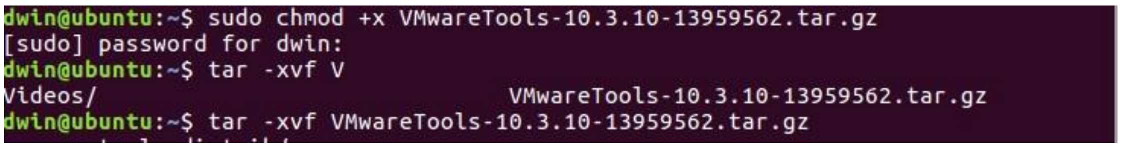
(5) Right click the tar file and click "**copy to**" a path with permission, e.g., to "**Home**".



(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.

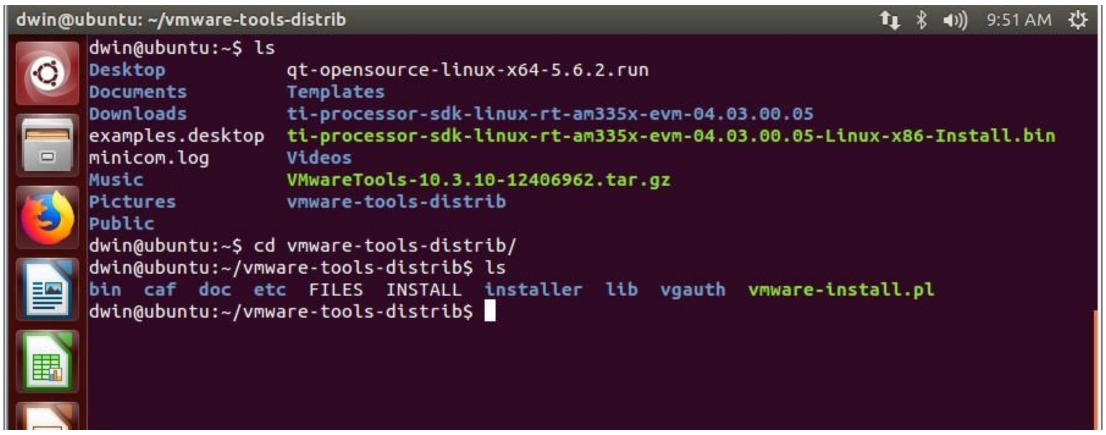


(7) Enter the command to enable the operable privilege: **sudo chmod +x VM** (to display the full name by **Tab** key) (**enter**). (Note: for the first time to use the administrator sudo privilege, you need to enter the password, i.e., the login password, which is not visible when entering.)



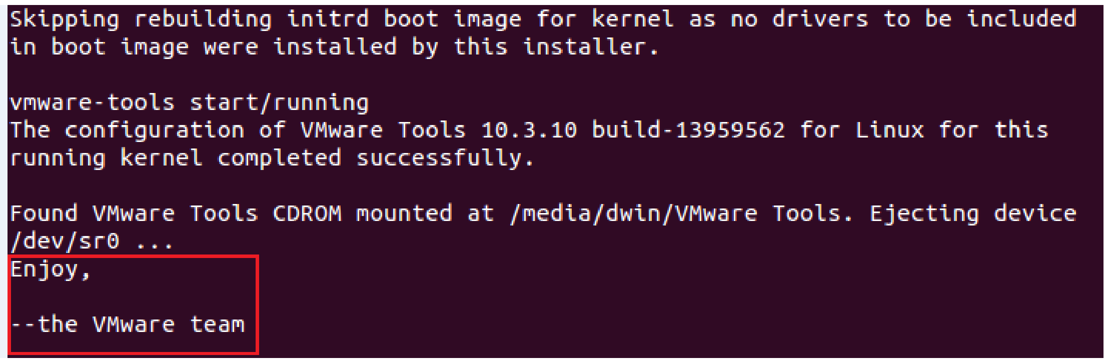
```
dwin@ubuntu:~$ sudo chmod +x VMwareTools-10.3.10-13959562.tar.gz
[sudo] password for dwin:
dwin@ubuntu:~$ tar -xvf V
Videos/                          VMwareTools-10.3.10-13959562.tar.gz
dwin@ubuntu:~$ tar -xvf VMwareTools-10.3.10-13959562.tar.gz
```

(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).



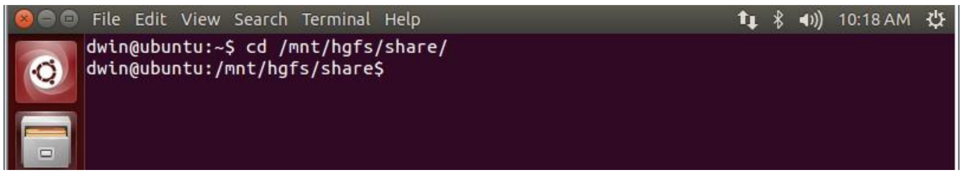
```
dwin@ubuntu:~$ ls
Desktop          qt-opensource-linux-x64-5.6.2.run
Documents        Templates
Downloads        ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05
examples.desktop ti-processor-sdk-linux-rt-am335x-evm-04.03.00.05-Linux-x86-Install.bin
minicom.log      Videos
Music            VMwareTools-10.3.10-12406962.tar.gz
Pictures         vmware-tools-distrib
Public
dwin@ubuntu:~$ cd vmware-tools-distrib/
dwin@ubuntu:~/vmware-tools-distrib$ ls
bin  caf  doc  etc  FILES  INSTALL  installer  lib  vgauth  vmware-install.pl
dwin@ubuntu:~/vmware-tools-distrib$
```

(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.



```
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
```

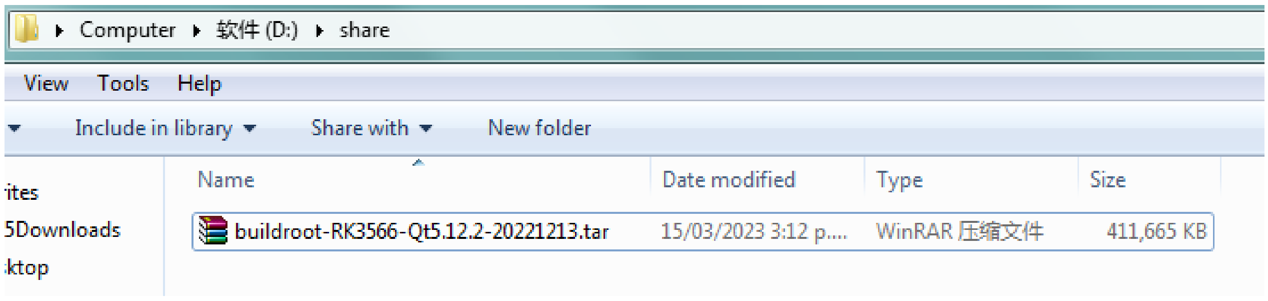
(10) At this time, we can enter the command: **cd /mn** (Tab all the way to the shared folder you set), the path is /mnt/hfgs/***, and the shared folder is set up here.



```
dwin@ubuntu:~$ cd /mnt/hfgs/share/
dwin@ubuntu:/mnt/hfgs/share$
```

2.2 Install RK3566 Toolchain

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


Computer ▸ 软件 (D:) ▸ share
View Tools Help
Include in library ▾ Share with ▾ New folder
ites
5Downloads
ktop

```
Name                                  Date modified        Type             Size
buildroot-RK3566-Qt5.12.2-20221213.tar 15/03/2023 3:12 p.... WinRAR 压缩文件   411,665 KB
```

(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.

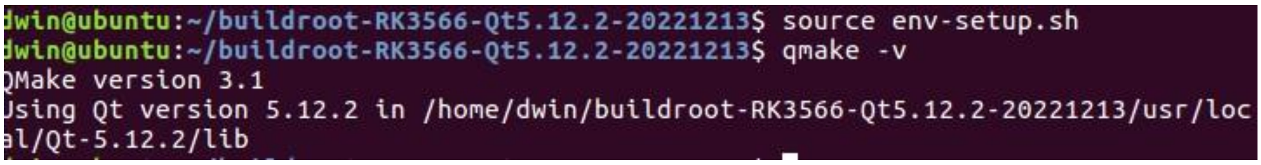
(3) Enter the command **tar -xvf bu(TAB)(enter)** to extract the tar file.

(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.

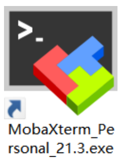


```
dwin@ubuntu:~/buildroot-RK3566-Qt5.12.2-20221213$ source env-setup.sh
dwin@ubuntu:~/buildroot-RK3566-Qt5.12.2-20221213$ qmake -v
QMake version 3.1
Using Qt version 5.12.2 in /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/usr/loc
al/Qt-5.12.2/lib
```

2.3 Development Board Configuration

2.3.1 Terminal Software

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

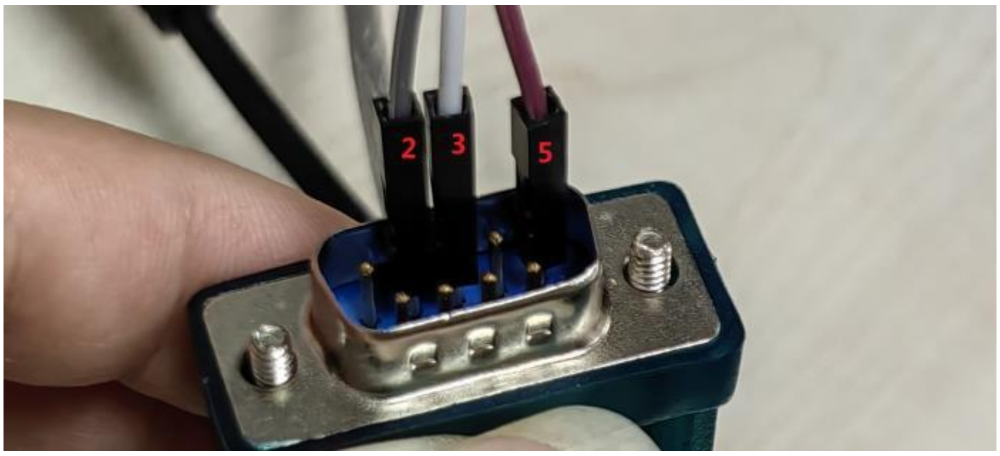


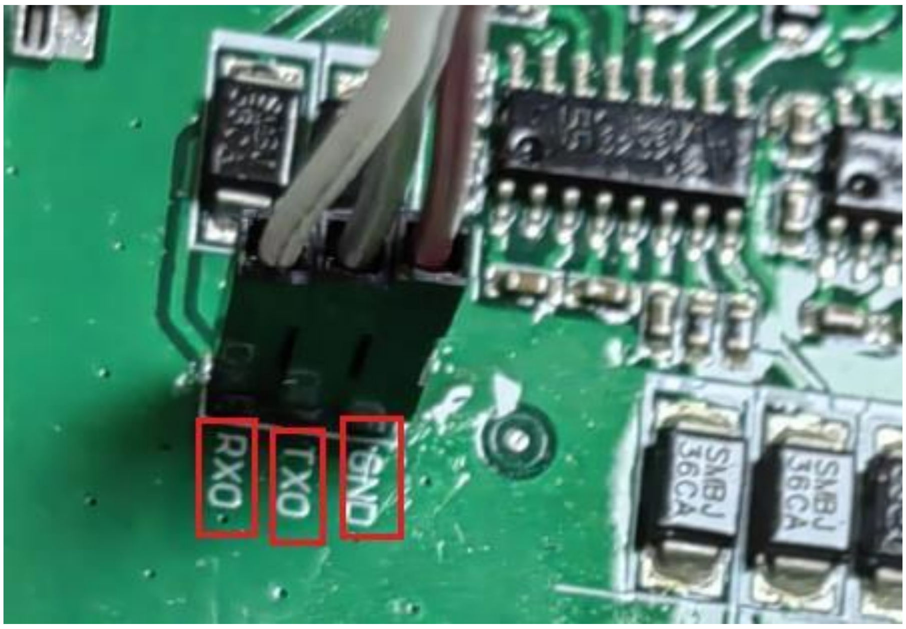
(2) There are two connection options: Serial (UART 0) connection and Telnet connection by a network cable.

2.3.2 Serial Connection

(1) Serial (UART 0) connection. As illustrated in the following pictures, connect No.2 to TX, No.3 to

RX and No.5 to GND. (RS232 as an example here.)



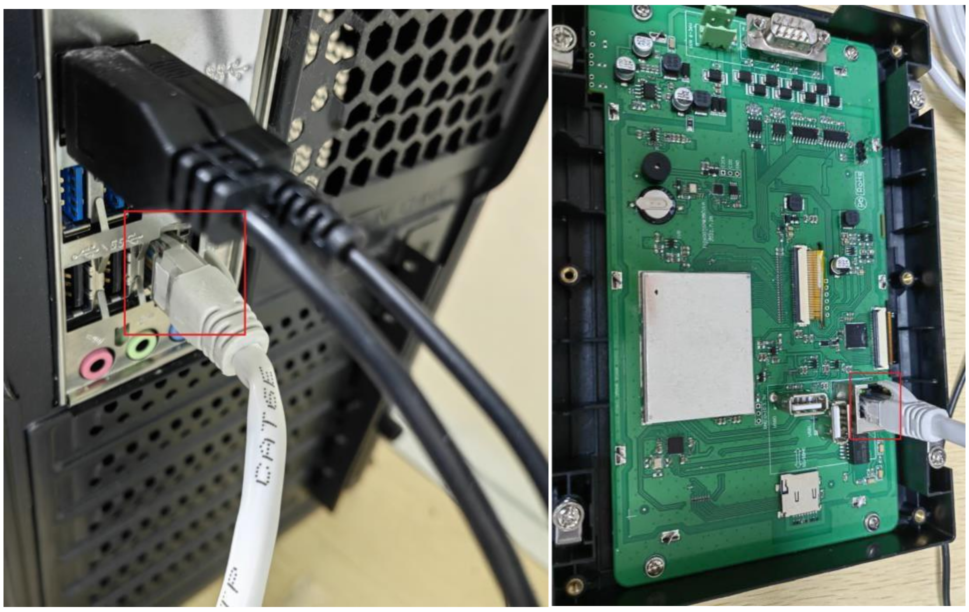


(2) Select **[Sessions]->** **[New Session]**. First, select “**serial**”. Next, select **serial port** and select **speed**. Last, check the information and click “**OK**” to finish.

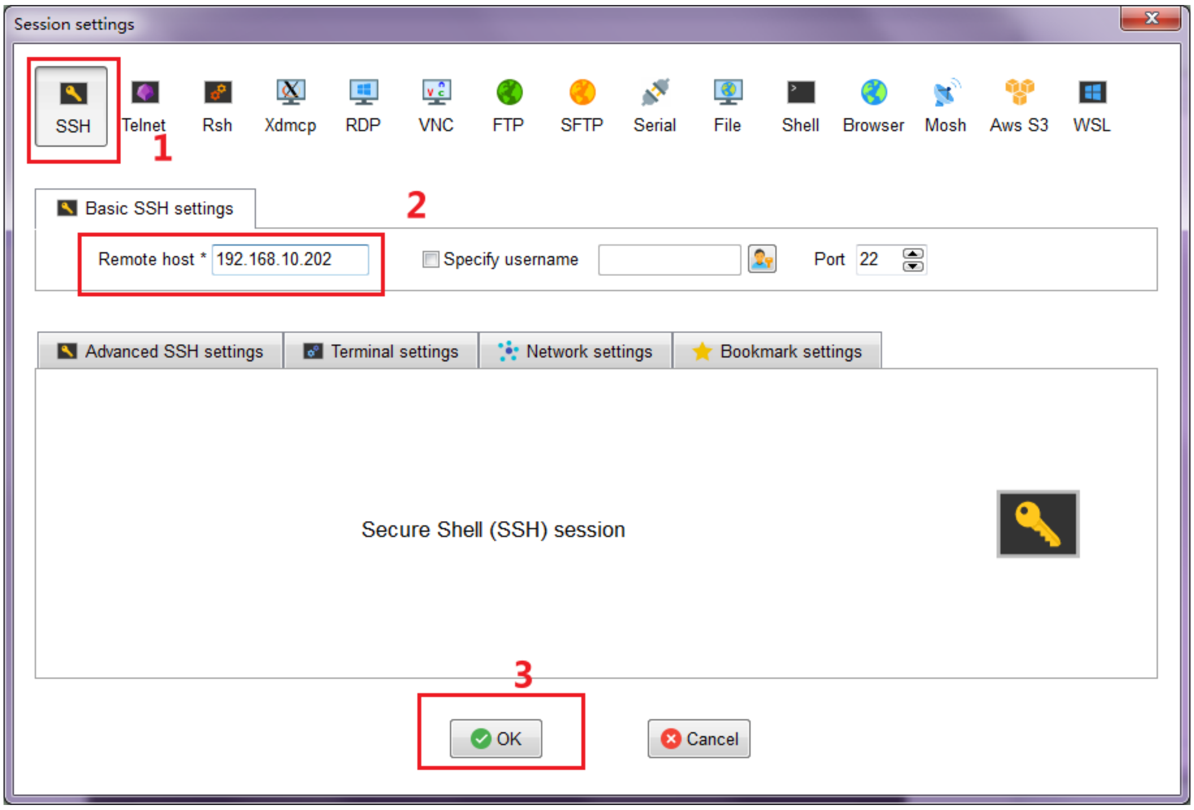
(3) Power up the development board, and enter “root” to start. (Note: If you operate after a while after powering up, there may be no text on the displayed interface, and only a black screen with no boot information. In this case, you only need to enter “root”).

2.3.3 Ethernet SSH Connection

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

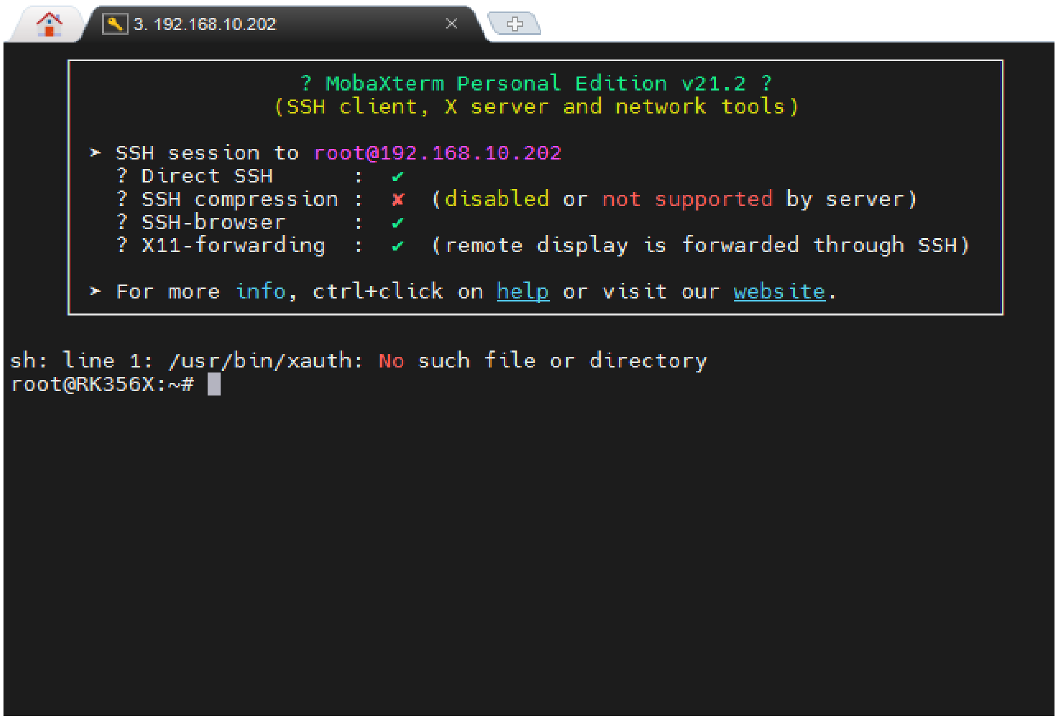


(2) First, click “**Session**” and select “**New session**” then select “**SSH**”. Next, enter the IP of the development board and click “OK” (Note: the default IP of the development board is: 192.168.10.201 or 192.168.10.202. To achieve communication, the development board should be connected to the same LAN as the computer.



Remote host * 192.168.10.202
Specify username
Port 22
Secure Shell (SSH) session

(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.



```
              ? MobaXterm Personal Edition v21.2 ?
            (SSH client, X server and network tools)

 ➤ SSH session to root@192.168.10.202
   ? Direct SSH       :  ✓
   ? SSH compression  :  ✘  (disabled or not supported by server)
   ? SSH-browser      :  ✓
   ? X11-forwarding   :  ✓  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.

sh: line 1: /usr/bin/xauth: No such file or directory
root@RK356X:~#
```

2.3.4 Boot the startup logo

The DWIN 40 and 40OS-1 series Linux screens currently do not support user to modify the startup interface LOGO themselves. However, user can provide BMP images for modification by the DWIN R&D team.

3 QT Project Cross-compilation

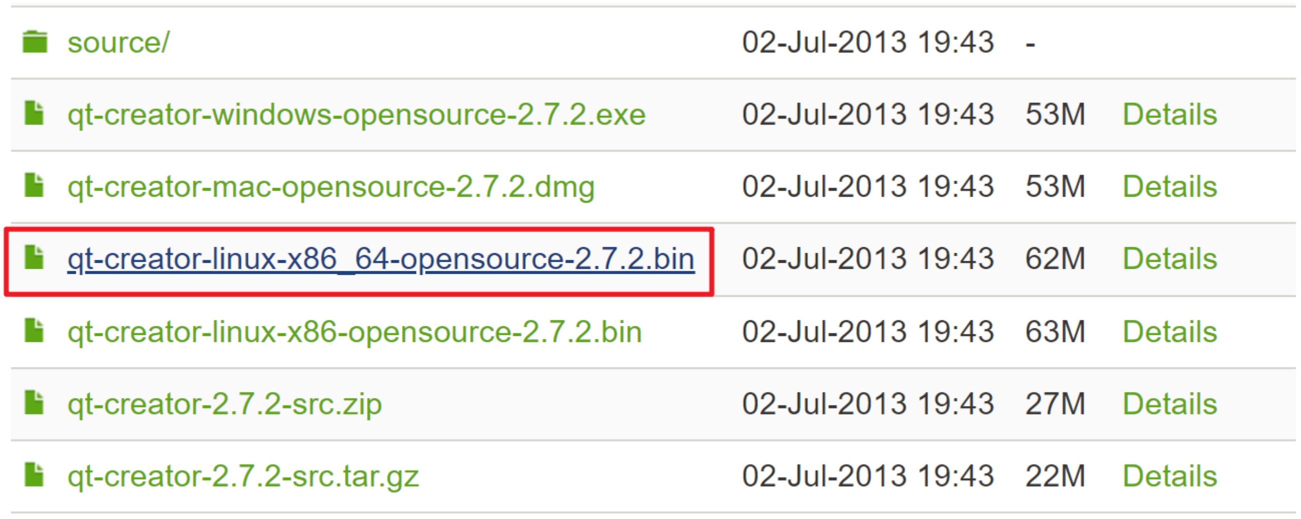
3.1 Install Qt Creator

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.

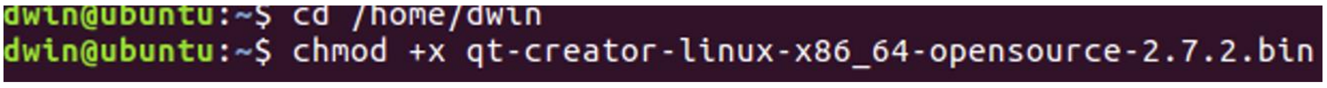


source/	02-Jul-2013 19:43	-	
qt-creator-windows-opensource-2.7.2.exe	02-Jul-2013 19:43	53M	Details
qt-creator-mac-opensource-2.7.2.dmg	02-Jul-2013 19:43	53M	Details
qt-creator-linux-x86_64-opensource-2.7.2.bin	02-Jul-2013 19:43	62M	Details
qt-creator-linux-x86-opensource-2.7.2.bin	02-Jul-2013 19:43	63M	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

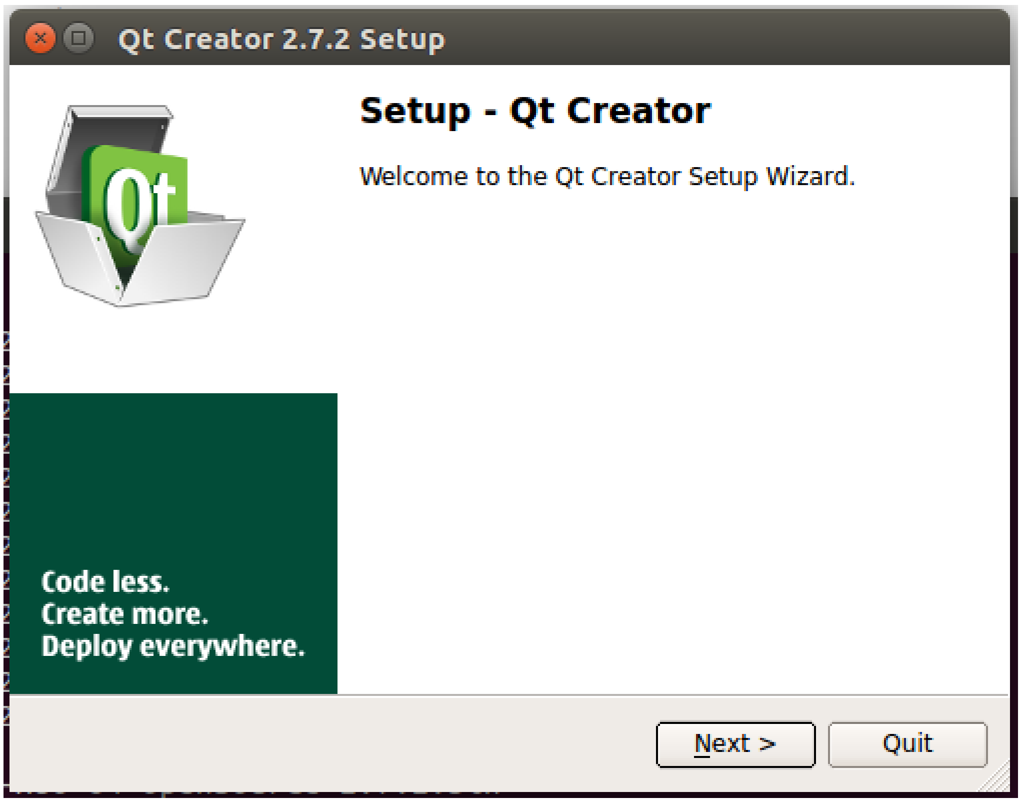


```
dwin@ubuntu:~$ cd /home/dwin
dwin@ubuntu:~$ chmod +x qt-creator-linux-x86_64-opensource-2.7.2.bin
```

Run the installer

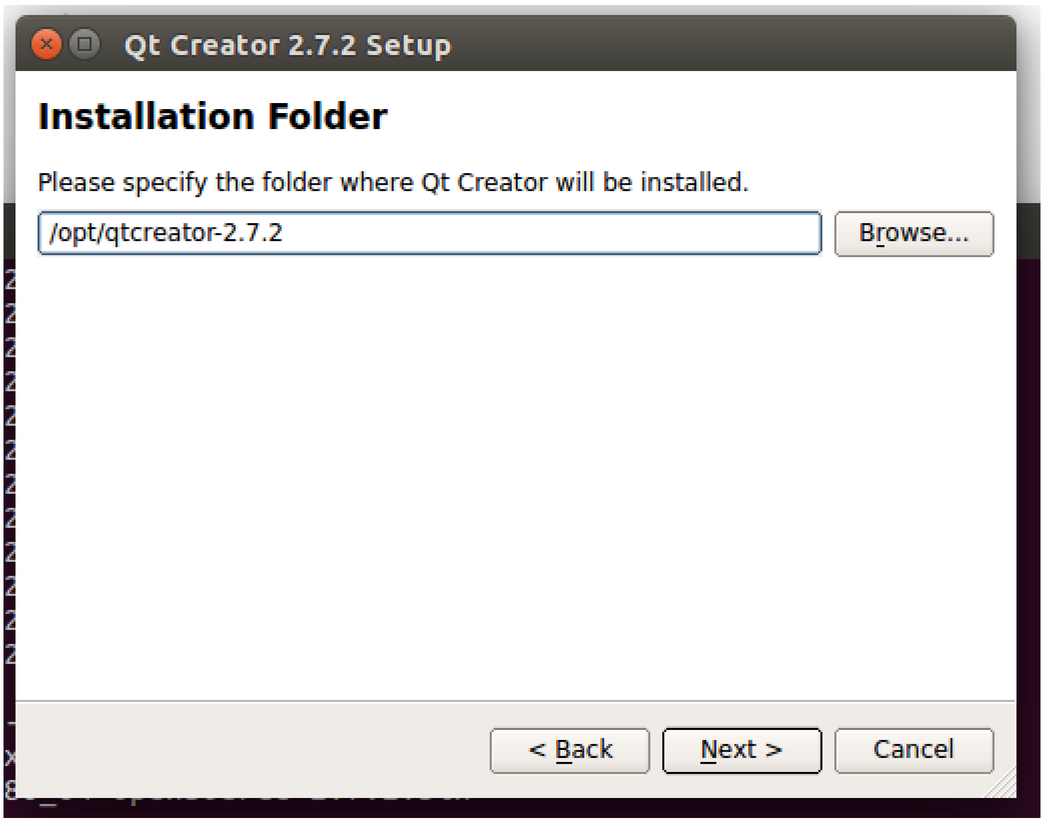
# sudo ./qt-creator-linux-x86_64-opensource-2.7.2.bi

Click the “Next”:


Qt Creator 2.7.2 Setup

Setup - Qt Creator

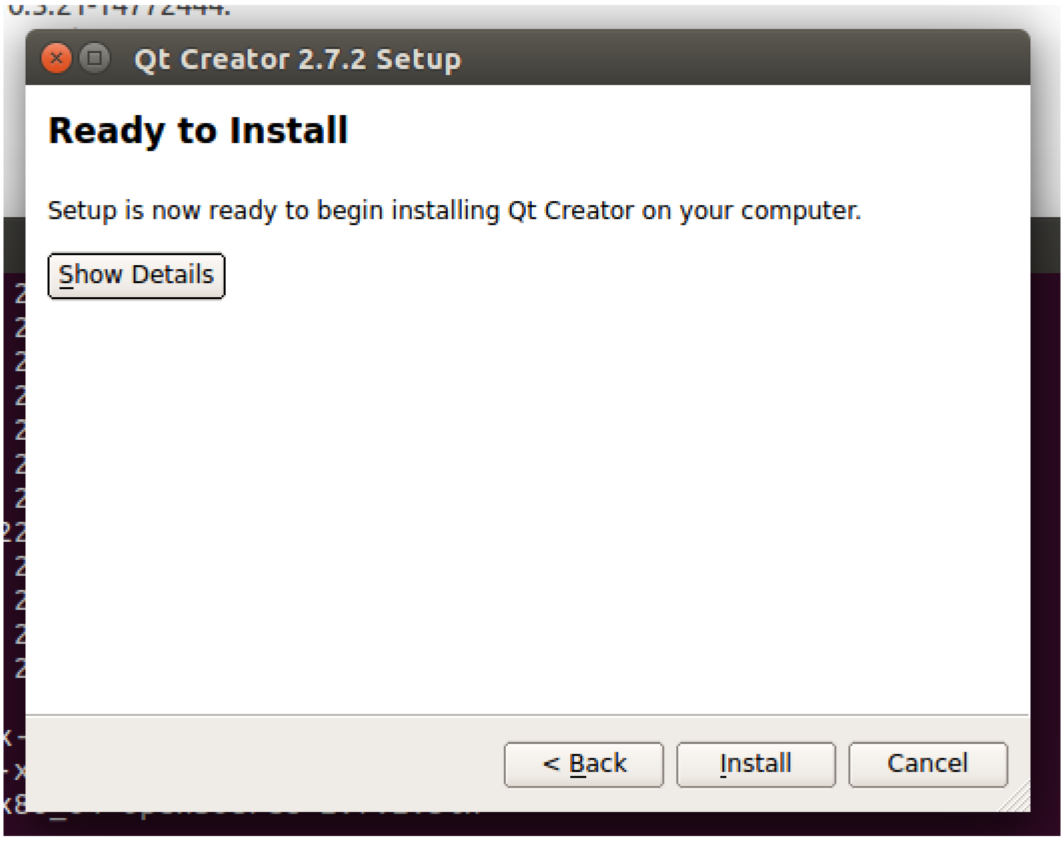
Welcome to the Qt Creator Setup Wizard.


Qt Creator 2.7.2 Setup

Installation Folder

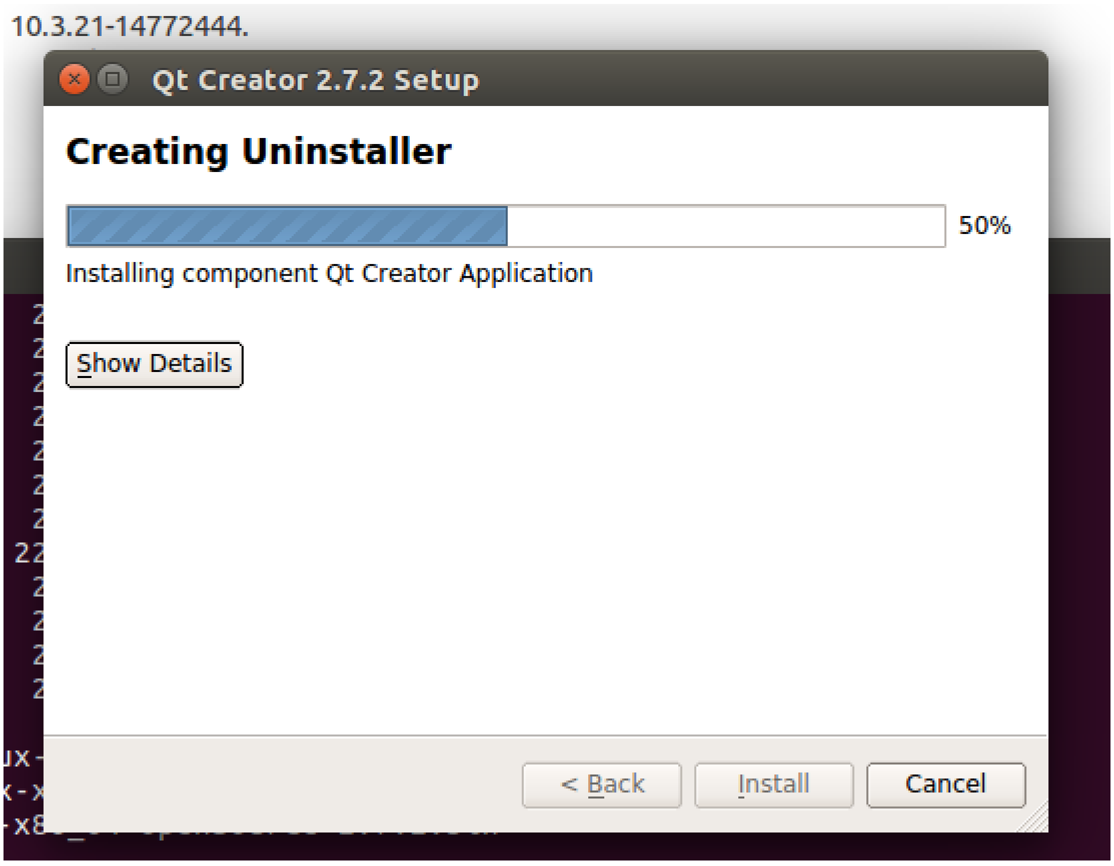
Please specify the folder where Qt Creator will be installed.

/opt/qtcreator-2.7.2


Qt Creator 2.7.2 Setup

Ready to Install

Setup is now ready to begin installing Qt Creator on your computer.

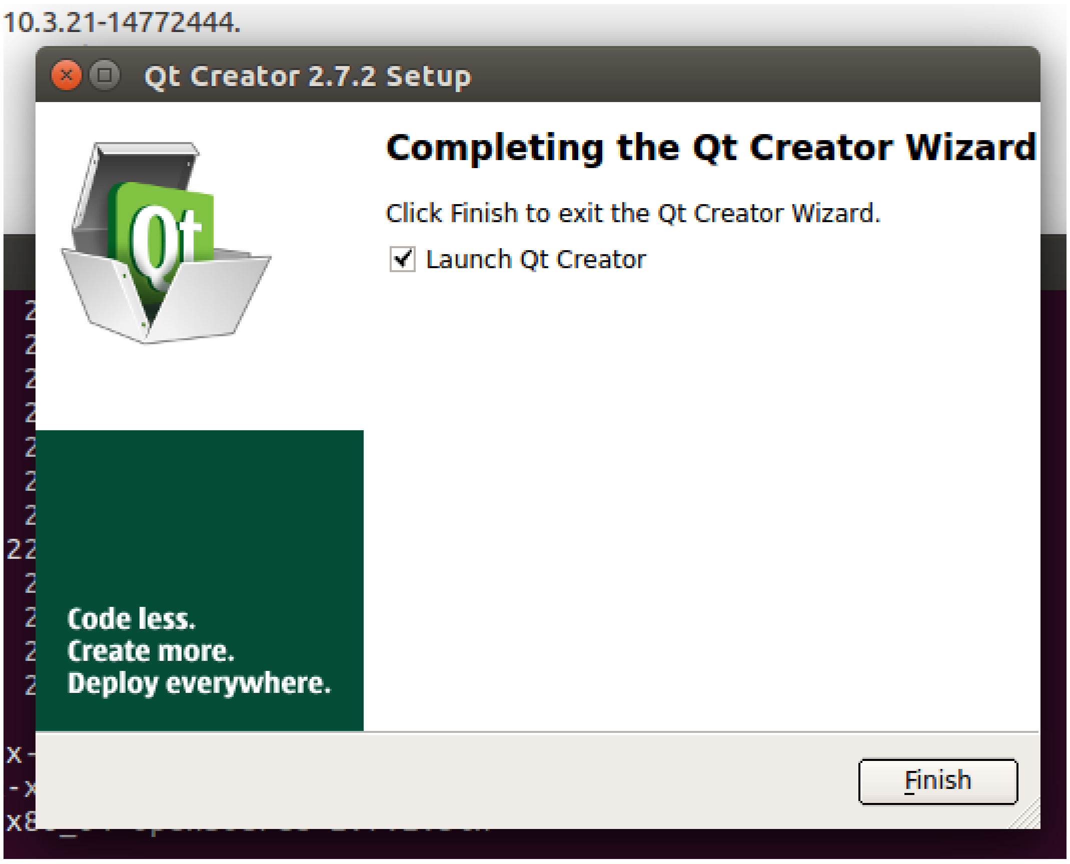

10.3.21-14772444.

Qt Creator 2.7.2 Setup

Creating Uninstaller

50%

Installing component Qt Creator Application


10.3.21-14772444.

Qt Creator 2.7.2 Setup

Completing the Qt Creator Wizard

Click Finish to exit the Qt Creator Wizard.

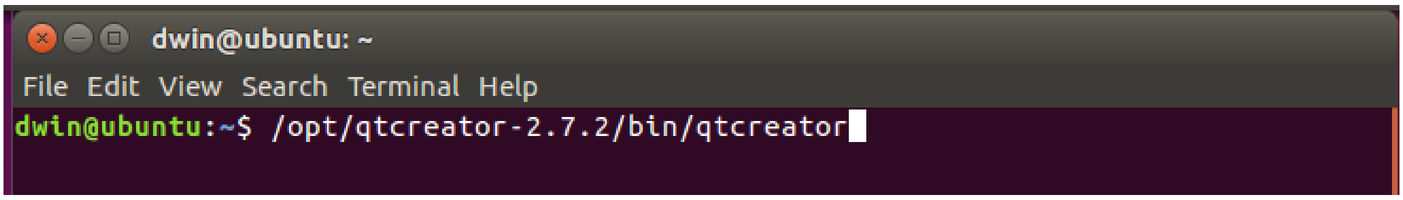
☑ Launch Qt Creator

3.2 Set up the cross-compilation environment

3.2.1 Run the Qt Creator

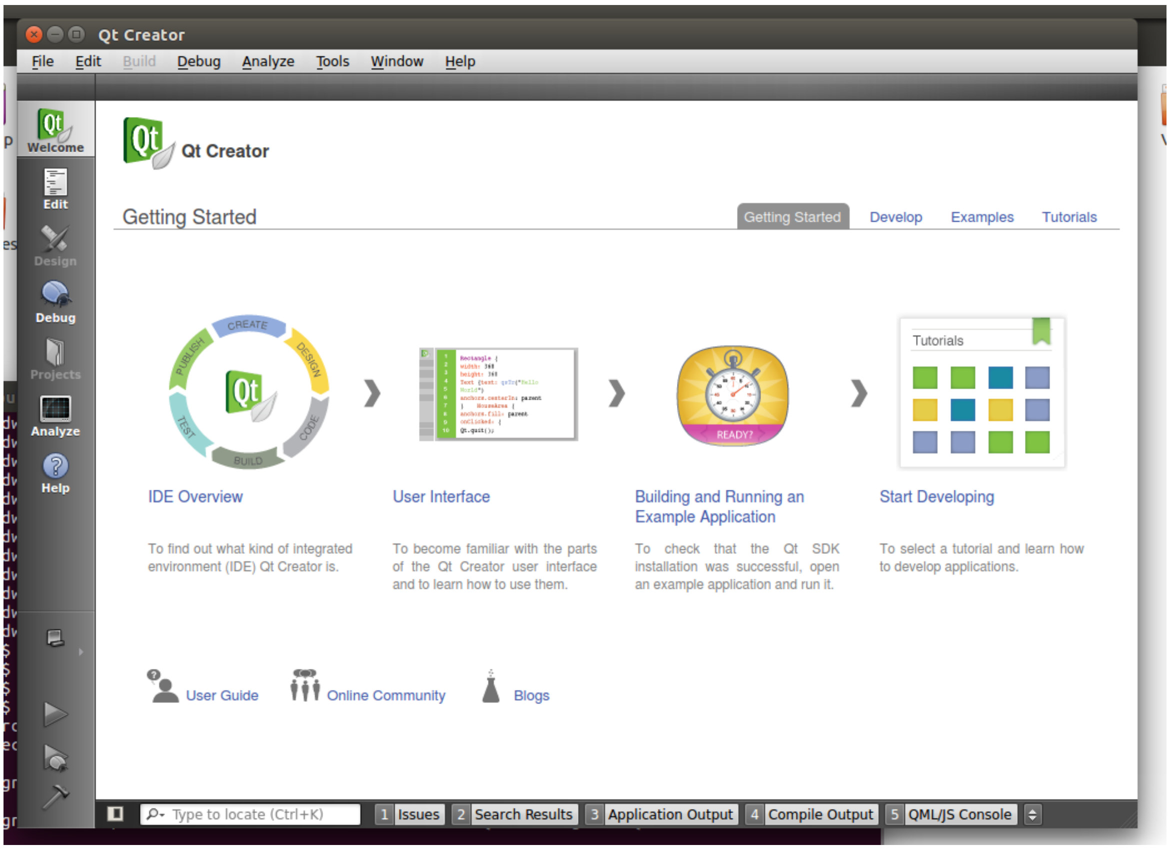
The Qt Creator executable file is in the bin directory of the installation directory.

# /opt/qtcreator-2.7.2/bin/qtcreator



```
dwin@ubuntu:~$ /opt/qtcreator-2.7.2/bin/qtcreator
```

The interface of software is as below.



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

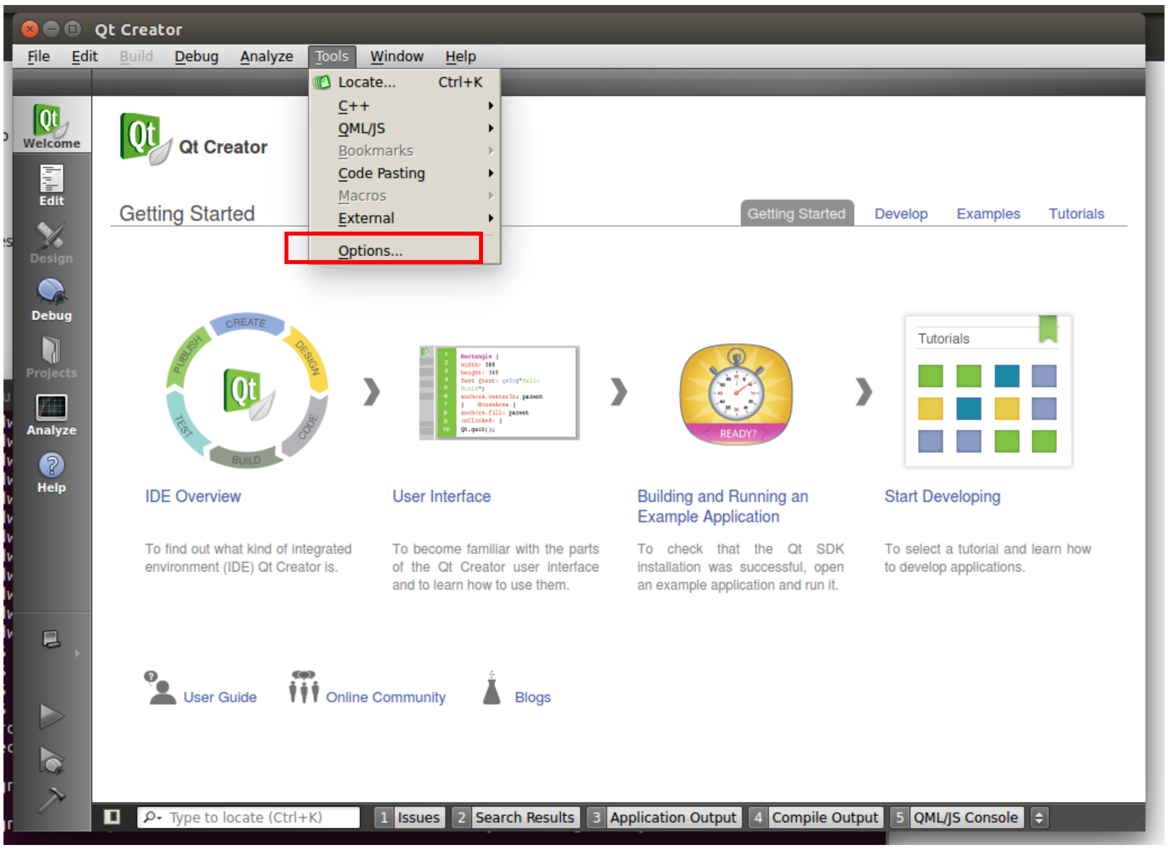
To check that the Qt SDK installation was successful, open an example application and run it.

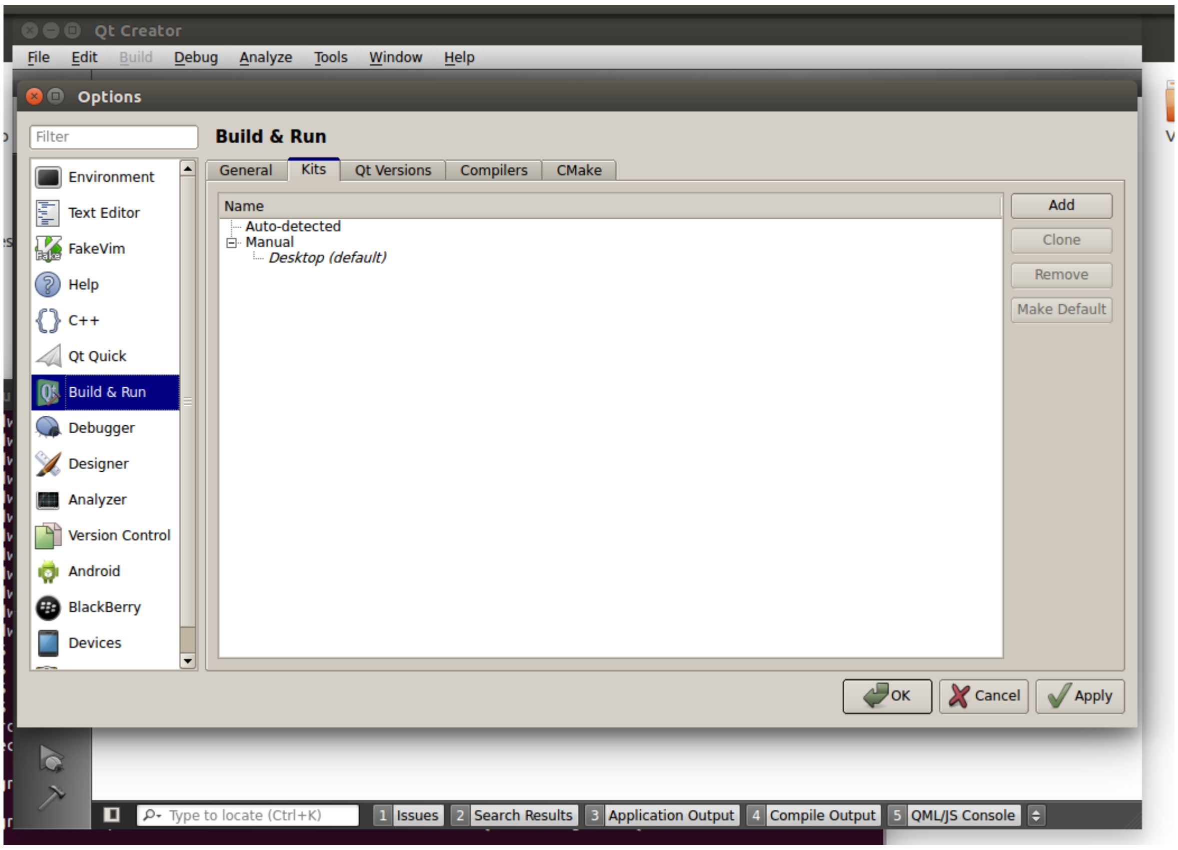
Start Developing

To select a tutorial and learn how to develop applications.

3.2.2 Set up the cross-compilation environment

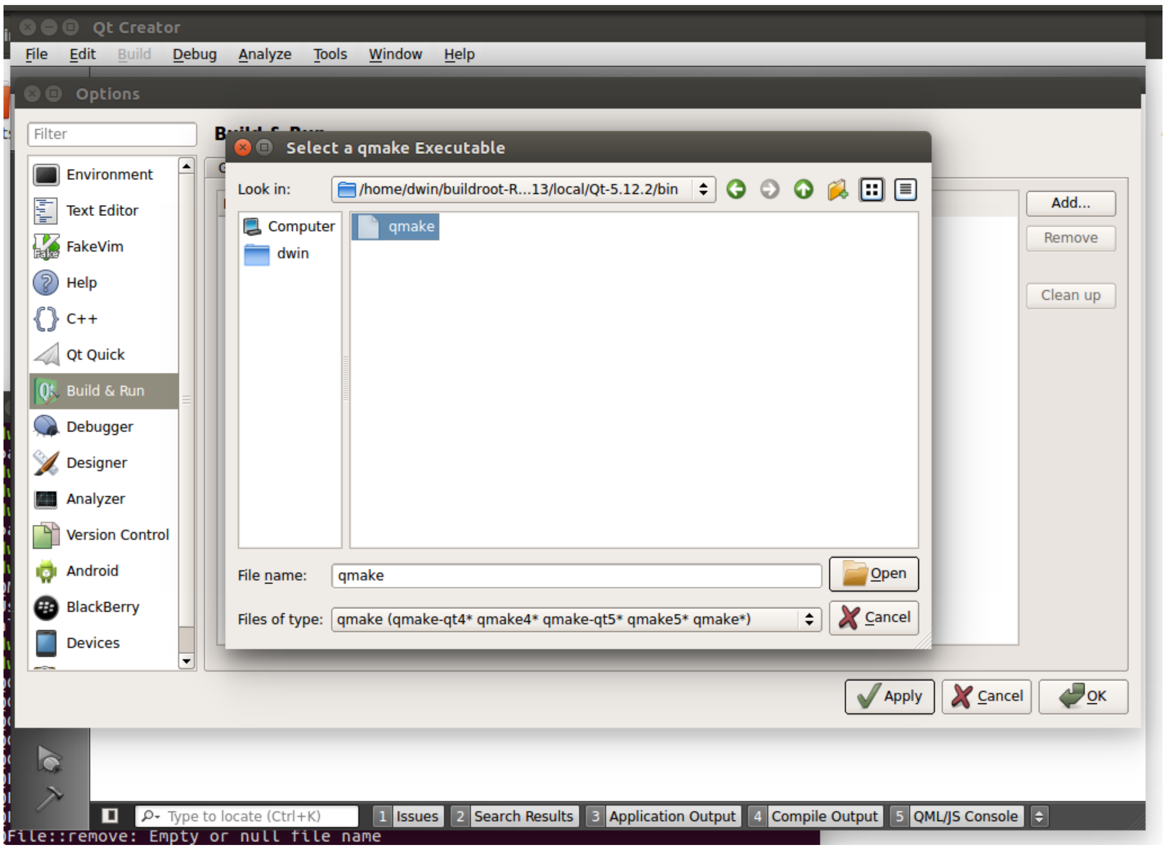
Choose [tool] – [options] as below.





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



Select a qmake Executable

Look in: /home/dwin/buildroot-R...13/local/Qt-5.12.2/bin

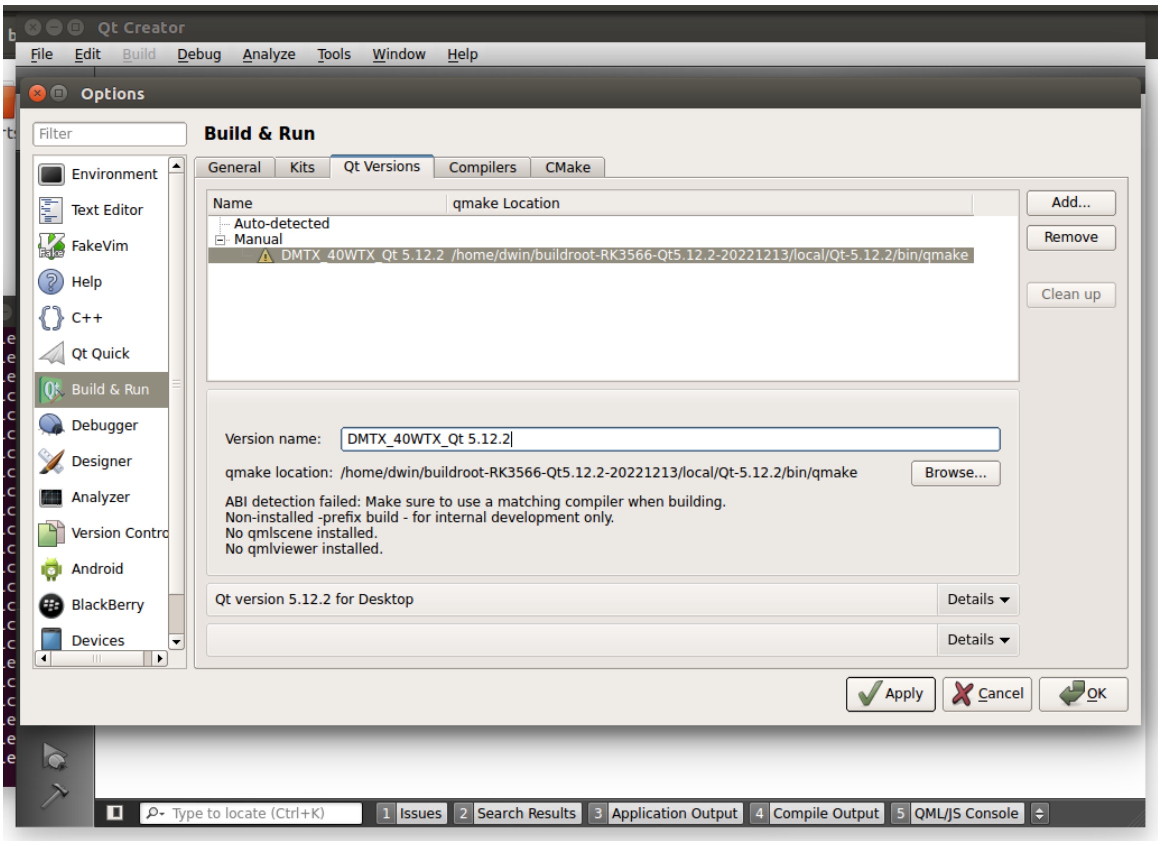
Computer
dwin
qmake

File name: qmake

Files of type: qmake (qmake-qt4* qmake4* qmake-qt5* qmake5* qmake*)

Open | Cancel

```
File::remove: Empty or null file name
```



Build & Run

General | Kits | Qt Versions | Compilers | CMake

Name	qmake Location
Auto-detected	
Manual	
DMTX_40WTX_Qt 5.12.2	/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/local/Qt-5.12.2/bin/qmake

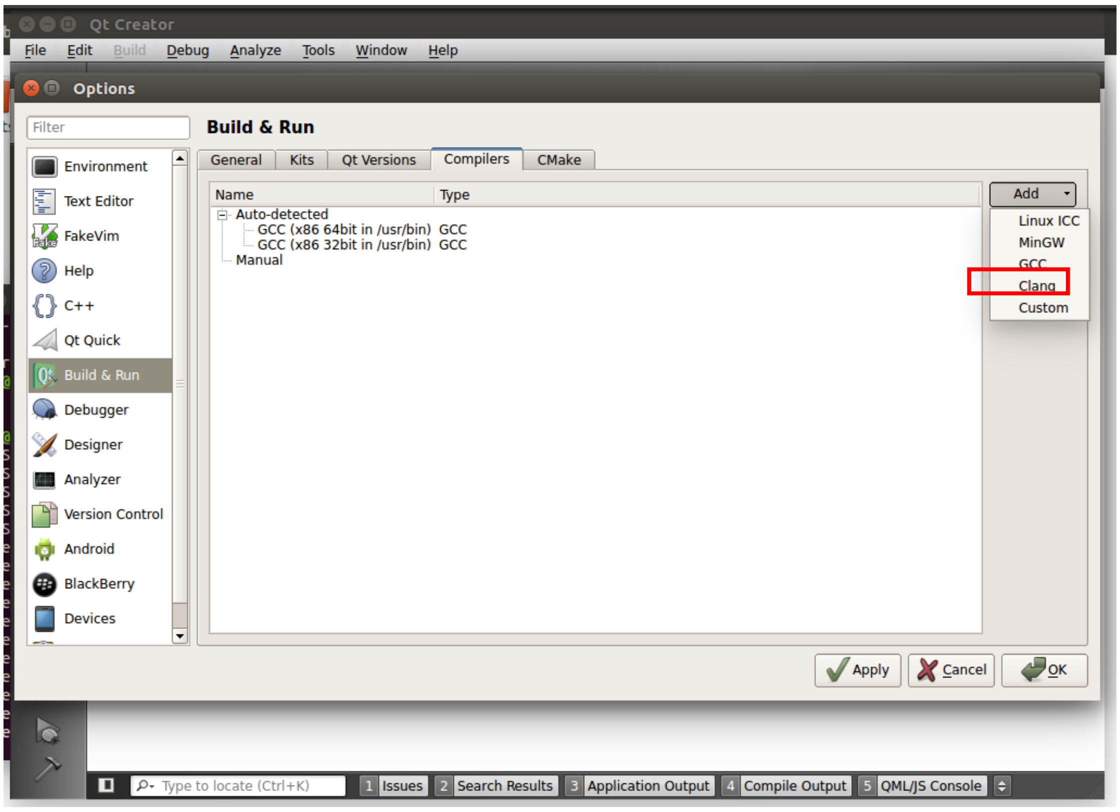
Version name: DMTX_40WTX_Qt 5.12.2

qmake location: /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/local/Qt-5.12.2/bin/qmake

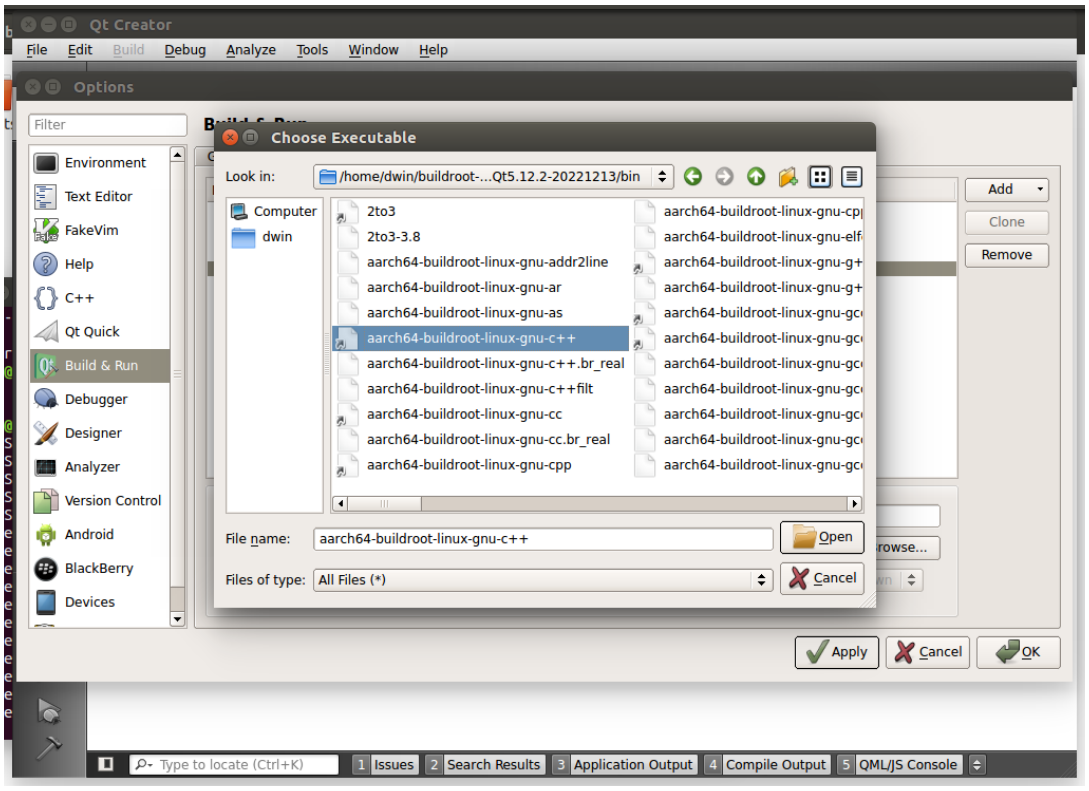
ABI detection failed: Make sure to use a matching compiler when building.
Non-installed -prefix build - for internal development only.
No qmlscene installed.
No qmlviewer installed.

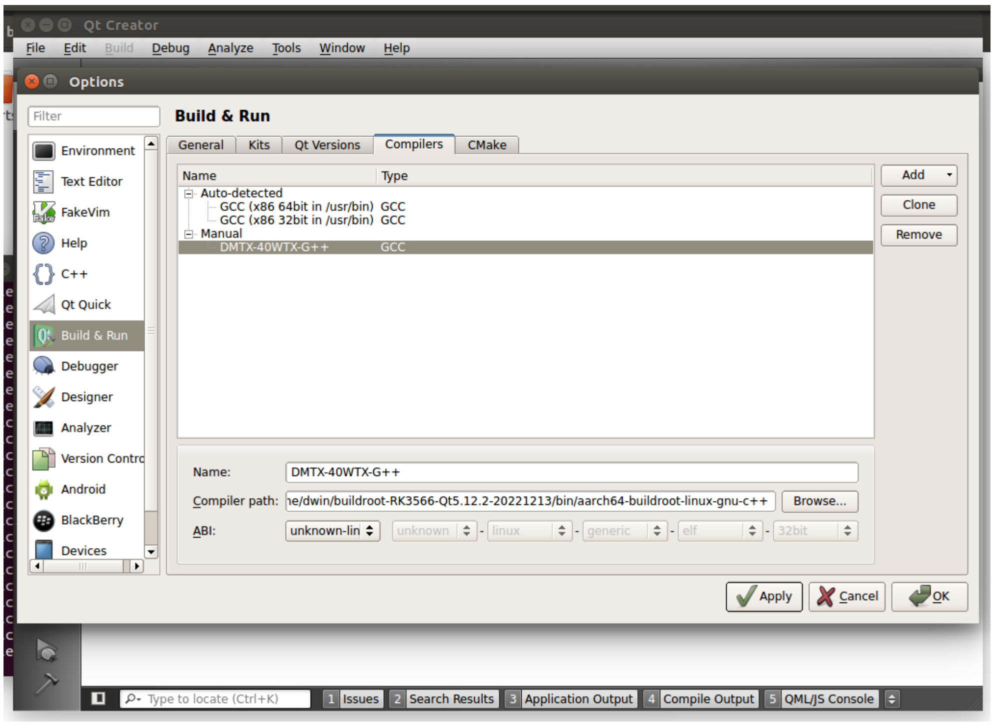
Qt version 5.12.2 for Desktop

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


Qt Creator
File Edit Build Debug Analyze Tools Window Help
Options
Build & Run
General Kits Qt Versions Compilers CMake
Name Type
Auto-detected
GCC (x86 64bit in /usr/bin) GCC
GCC (x86 32bit in /usr/bin) GCC
Manual
Add
Linux ICC
MinGW
GCC
Clang
Custom
Apply Cancel OK

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

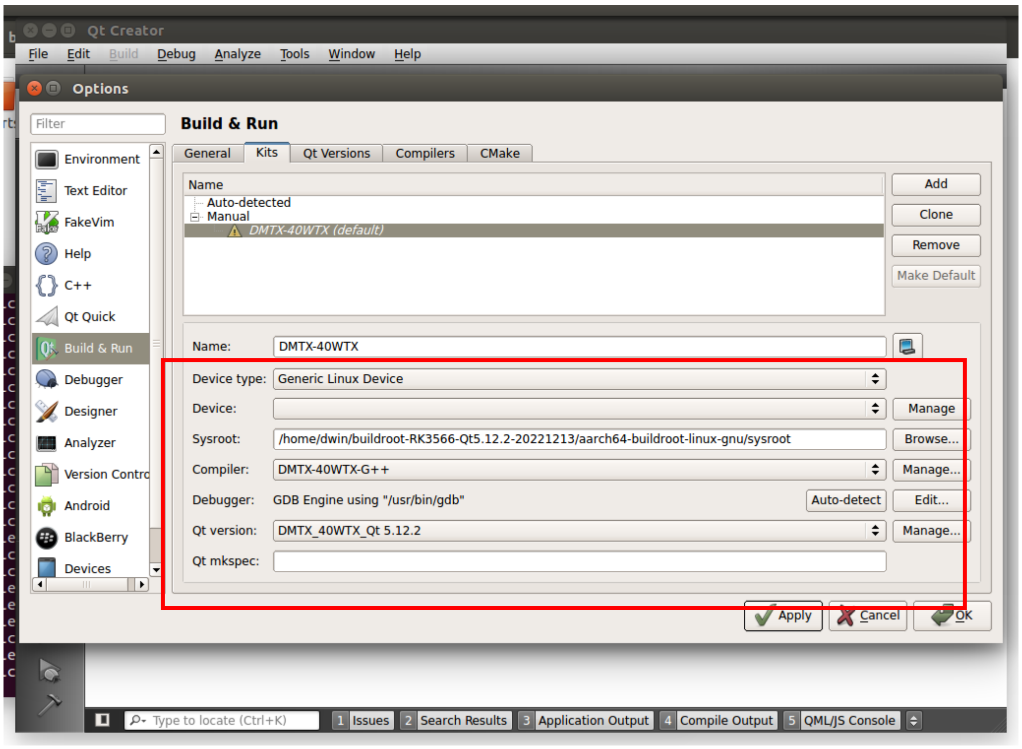

Choose Executable
Look in: /home/dwin/buildroot-...Qt5.12.2-20221213/bin
Computer
dwin
2to3
2to3-3.8
aarch64-buildroot-linux-gnu-addr2line
aarch64-buildroot-linux-gnu-ar
aarch64-buildroot-linux-gnu-as
aarch64-buildroot-linux-gnu-c++
aarch64-buildroot-linux-gnu-c++.br_real
aarch64-buildroot-linux-gnu-c++filt
aarch64-buildroot-linux-gnu-cc
aarch64-buildroot-linux-gnu-cc.br_real
aarch64-buildroot-linux-gnu-cpp
File name: aarch64-buildroot-linux-gnu-c++
Files of type: All Files (*)
Open Cancel



Name	Type
Auto-detected	
GCC (x86 64bit in /usr/bin)	GCC
GCC (x86 32bit in /usr/bin)	GCC
Manual	
DMTX-40WTX-G++	GCC

Name: DMTX-40WTX-G++
Compiler path: ne/dwin/buildroot-RK3566-Qt5.12.2-20221213/bin/aarch64-buildroot-linux-gnu-c++ Browse...
ABI: unknown-lin - unknown - linux - generic - elf - 32bit

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



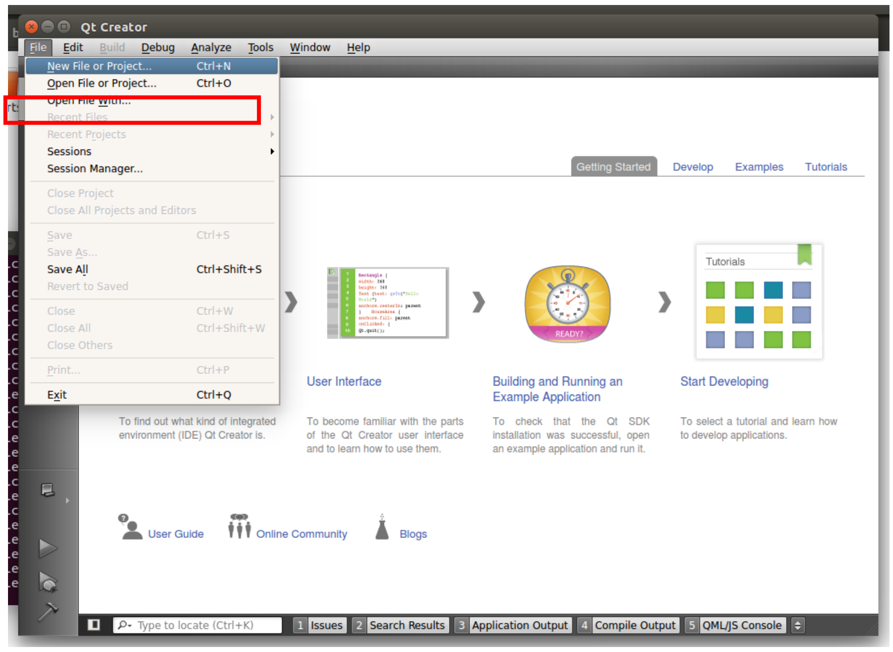
Name
Auto-detected
Manual
DMTX-40WTX (default)

Name: DMTX-40WTX
Device type: Generic Linux Device
Device:
Sysroot: /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot
Compiler: DMTX-40WTX-G++
Debugger: GDB Engine using "/usr/bin/gdb"
Qt version: DMTX_40WTX_Qt 5.12.2
Qt mkspec:

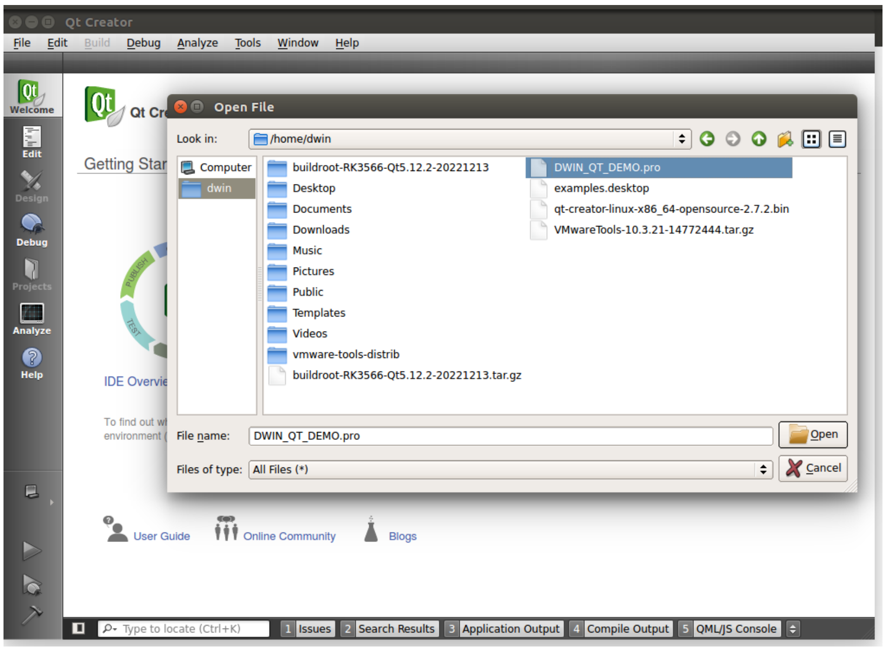
3.3 Compile Qt project

3.3.1 Open the project

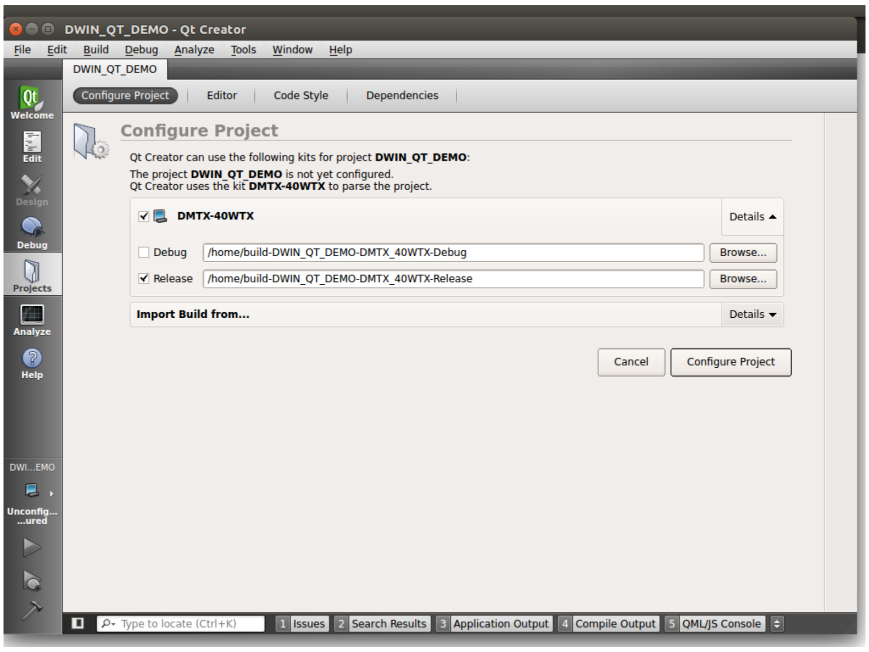
[File] – [open file or project]:



Choose the Qt project:


Qt Creator
File Edit Build Debug Analyze Tools Window Help
Welcome
Edit
Design
Debug
Projects
Analyze
Help
Open File
Look in: /home/dwin
Computer
dwin
buildroot-RK3566-Qt5.12.2-20221213
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
vmware-tools-distrib
buildroot-RK3566-Qt5.12.2-20221213.tar.gz
DWIN_QT_DEMO.pro
examples.desktop
qt-creator-linux-x86_64-opensource-2.7.2.bin
VMwareTools-10.3.21-14772444.tar.gz
File name: DWIN_QT_DEMO.pro
Files of type: All Files (*)
Open
Cancel
User Guide
Online Community
Blogs

Configure the project:


DWIN_QT_DEMO - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Configure Project | Editor | Code Style | Dependencies
Welcome
Edit
Design
Debug
Projects
Analyze
Help
DWI...EMO
Unconfig...ured
Details
Browse...
Cancel
Configure Project
Type to locate (Ctrl+K)
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML/JS Console

Configure Project

Qt Creator can use the following kits for project **DWIN_QT_DEMO**:

The project **DWIN_QT_DEMO** is not yet configured.
Qt Creator uses the kit **DMTX-40WTX** to parse the project.

[✓] **DMTX-40WTX**

[ ] Debug /home/build-DWIN_QT_DEMO-DMTX_40WTX-Debug

[✓] Release /home/build-DWIN_QT_DEMO-DMTX_40WTX-Release

Import Build from...

3.3.2 Add environment variables

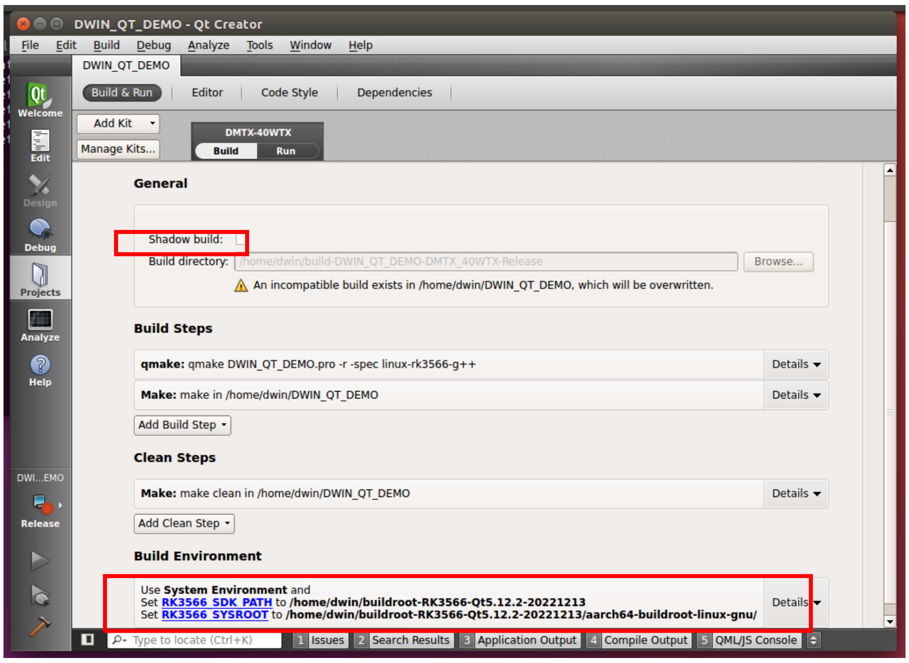
Go to [Projects] – [Build & Run[- [Build Environment], and add a variable:

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.


DWIN_QT_DEMO - Qt Creator
File Edit Build Debug Analyze Tools Window Help
DWIN_QT_DEMO
Build & Run | Editor | Code Style | Dependencies
Add Kit
Manage Kits...
DMTX-40WTX
Build Run
Welcome
Edit
Design
Debug
Projects
Analyze
Help
DWI...EMO
Release
Details
Browse...
Type to locate (Ctrl+K)
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML/JS Console

General

Shadow build: [ ]

Build directory: /home/dwin/build-DWIN_QT_DEMO-DMTX_40WTX-Release

An incompatible build exists in /home/dwin/DWIN_QT_DEMO, which will be overwritten.

Build Steps

qmake: qmake DWIN_QT_DEMO.pro -r -spec linux-rk3566-g++

Make: make in /home/dwin/DWIN_QT_DEMO

Add Build Step

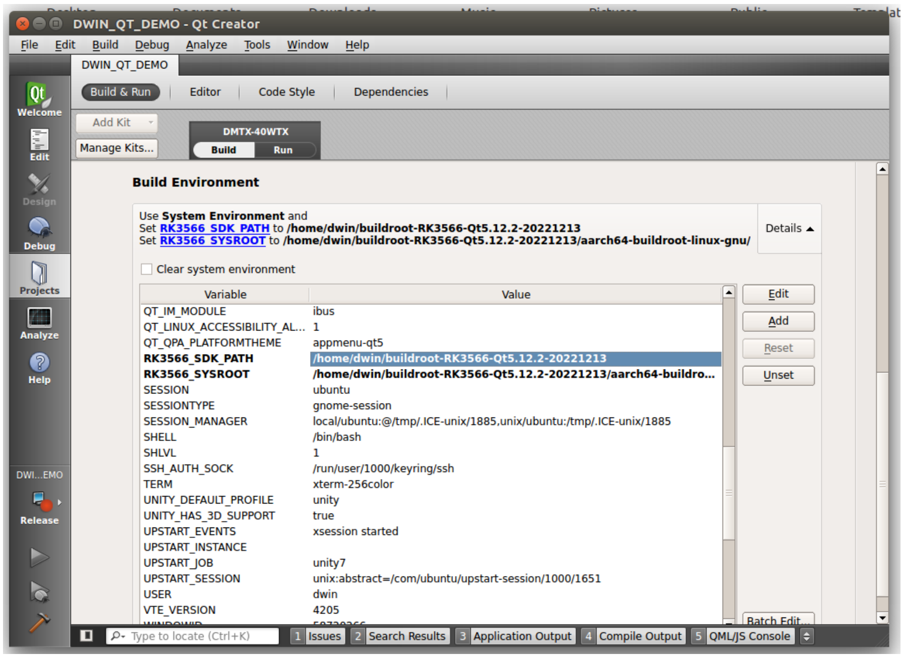
Clean Steps

Make: make clean in /home/dwin/DWIN_QT_DEMO

Add Clean Step

Build Environment

Use **System Environment** and
Set **RK3566_SDK_PATH** to **/home/dwin/buildroot-RK3566-Qt5.12.2-20221213**
Set **RK3566_SYSROOT** to **/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/**


Build Environment

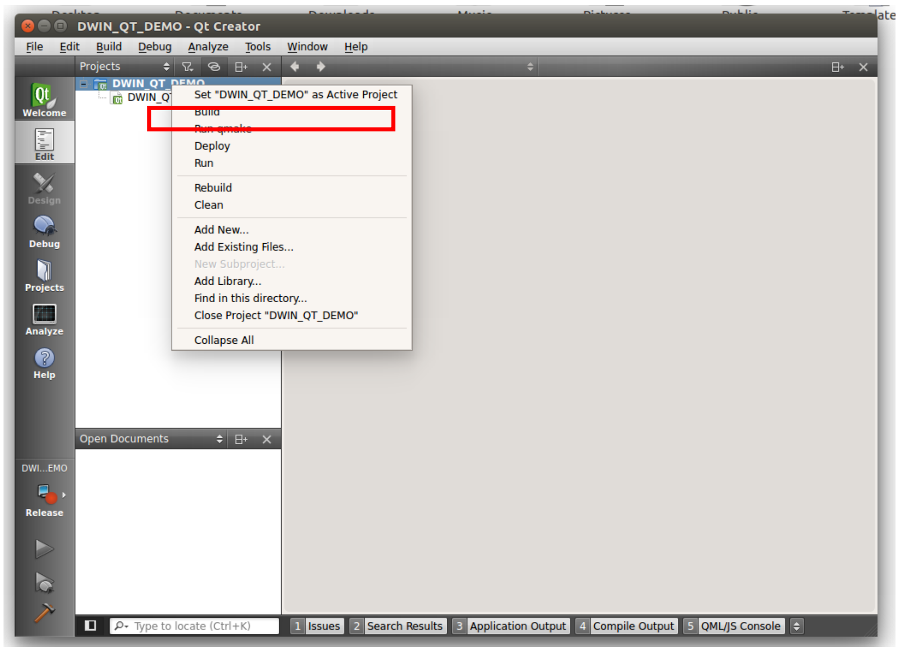
Use **System Environment** and
Set RK3566_SDK_PATH to **/home/dwin/buildroot-RK3566-Qt5.12.2-20221213**
Set RK3566_SYSROOT to **/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/**

Clear system environment

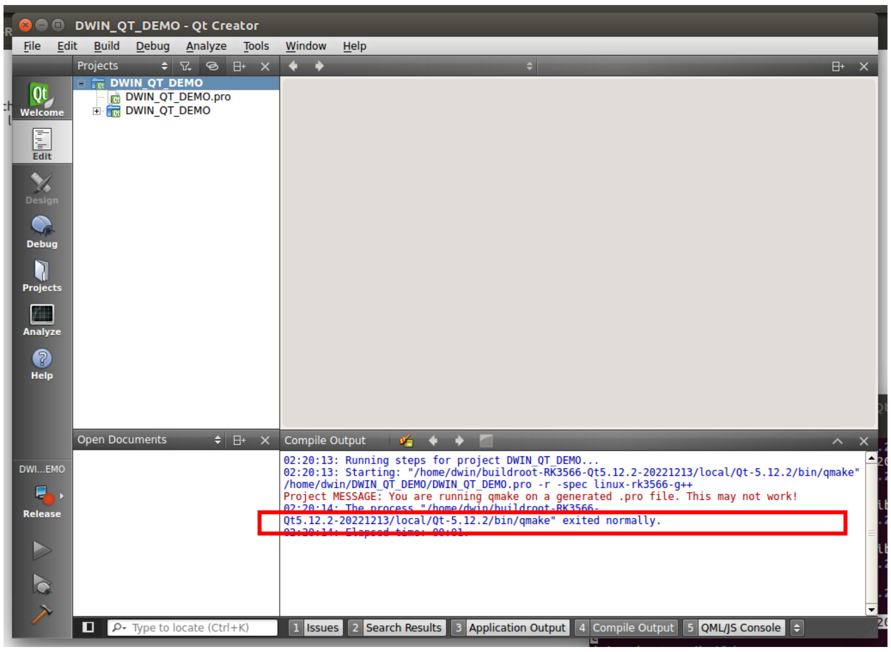
Variable	Value
QT_IM_MODULE	ibus
QT_LINUX_ACCESSIBILITY_AL...	1
QT_QPA_PLATFORMTHEME	appmenu-qt5
RK3566_SDK_PATH	**/home/dwin/buildroot-RK3566-Qt5.12.2-20221213**
RK3566_SYSROOT	**/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildro...**
SESSION	ubuntu
SESSIONTYPE	gnome-session
SESSION_MANAGER	local/ubuntu:@/tmp/.ICE-unix/1885,unix/ubuntu:/tmp/.ICE-unix/1885
SHELL	/bin/bash
SHLVL	1
SSH_AUTH_SOCK	/run/user/1000/keyring/ssh
TERM	xterm-256color
UNITY_DEFAULT_PROFILE	unity
UNITY_HAS_3D_SUPPORT	true
UPSTART_EVENTS	xsession started
UPSTART_INSTANCE	
UPSTART_JOB	unity7
UPSTART_SESSION	unix:abstract=/com/ubuntu/upstart-session/1000/1651
USER	dwin
VTE_VERSION	4205

3.3.3 Run qmake

Choose project, 'right key' – run qmake

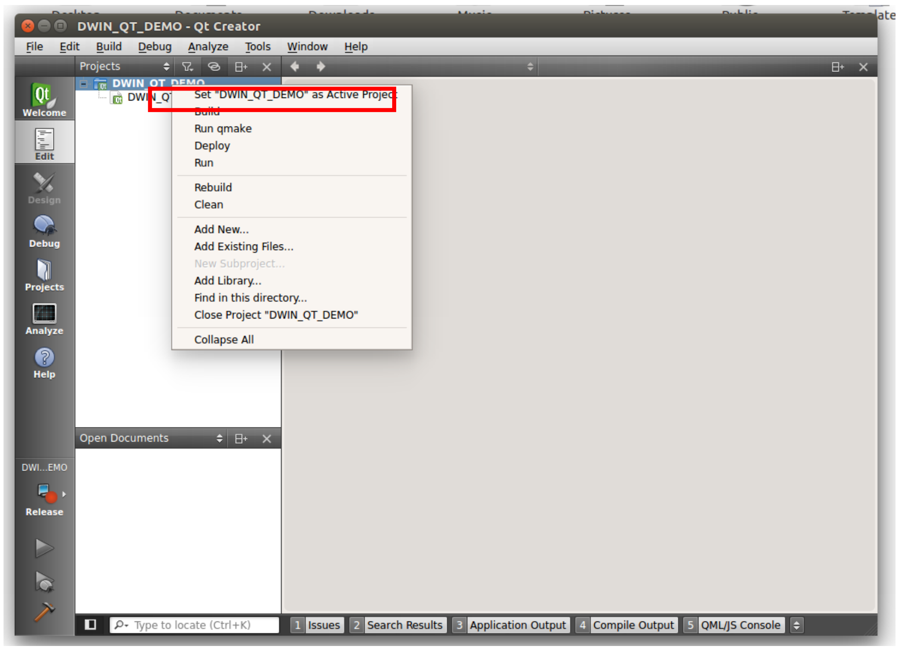


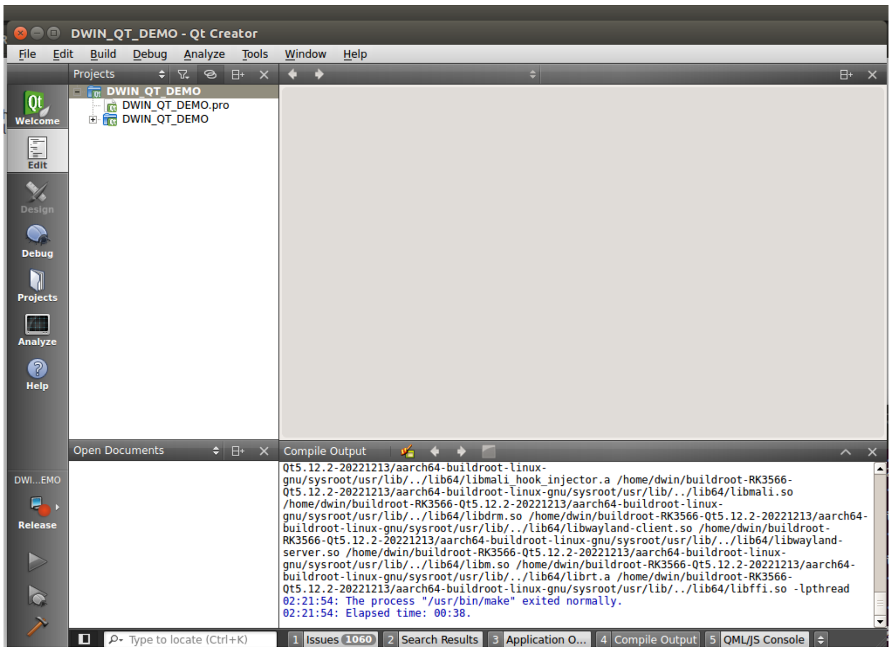
When qmake is successful, it looks like the image below (the red part is the printout of DWIN_QT_DEMO.pri, which does not affect).



```
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.
02:20:14: Elapsed time: 00:01.
```

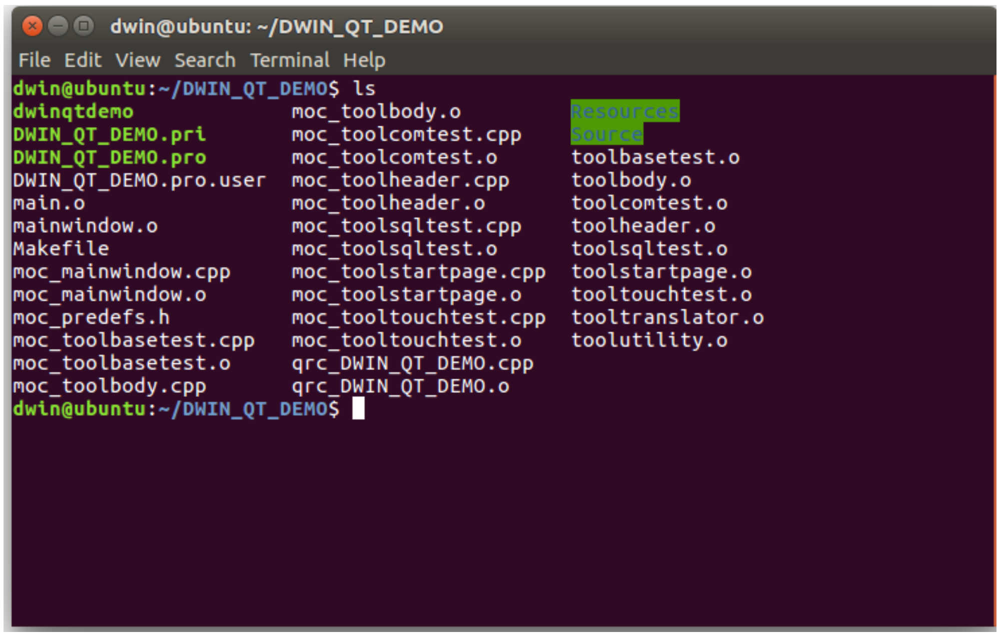
3.3.4 Build





```
Qt5.12.2-20221213/aarch64-buildroot-linux-
gnu/sysroot/usr/lib/../lib64/libmali_hook_injector.a /home/dwin/buildroot-RK3566-
Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot/usr/lib/../lib64/libmali.so
/home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-
gnu/sysroot/usr/lib/../lib64/libdrm.so /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-
buildroot-linux-gnu/sysroot/usr/lib/../lib64/libwayland-client.so /home/dwin/buildroot-
RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-gnu/sysroot/usr/lib/../lib64/libwayland-
server.so /home/dwin/buildroot-RK3566-Qt5.12.2-20221213/aarch64-buildroot-linux-
gnu/sysroot/usr/lib/../lib64/libm.so /home/dwin/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
02:21:54: The process "/usr/bin/make" exited normally.
02:21:54: Elapsed time: 00:38.
```

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



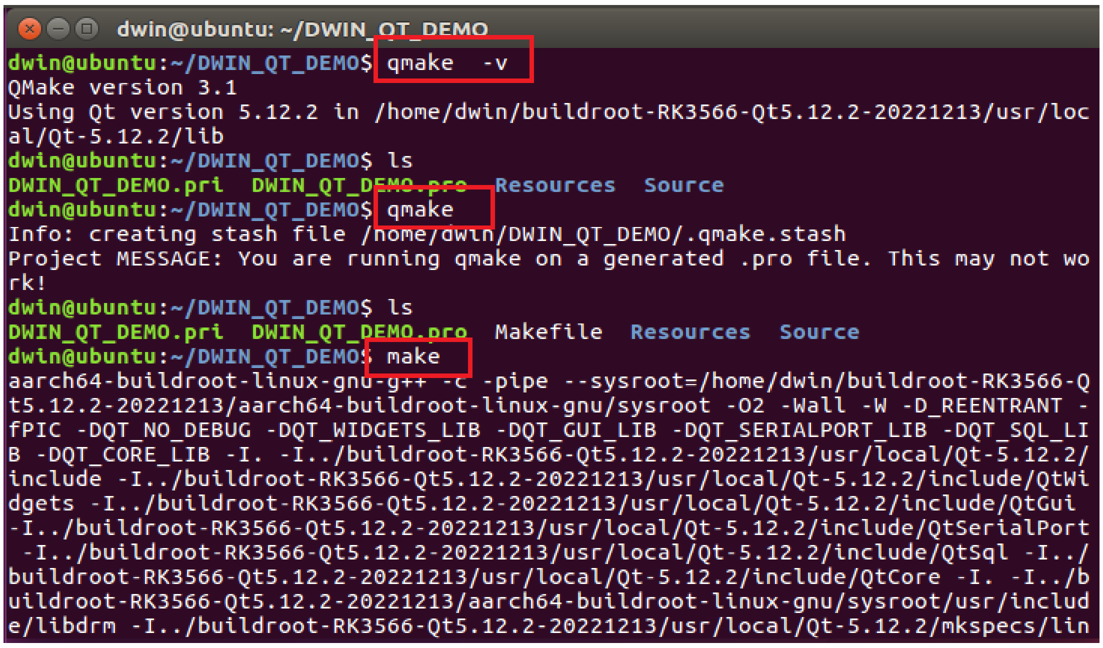
```
dwin@ubuntu:~/DWIN_QT_DEMO$ ls
dwinqtdemo              moc_toolbody.o          Resources
DWIN_QT_DEMO.pri        moc_toolcomtest.cpp     Source
DWIN_QT_DEMO.pro        moc_toolcomtest.o       toolbasetest.o
DWIN_QT_DEMO.pro.user   moc_toolheader.cpp      toolbody.o
main.o                  moc_toolheader.o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

3.4 qmake

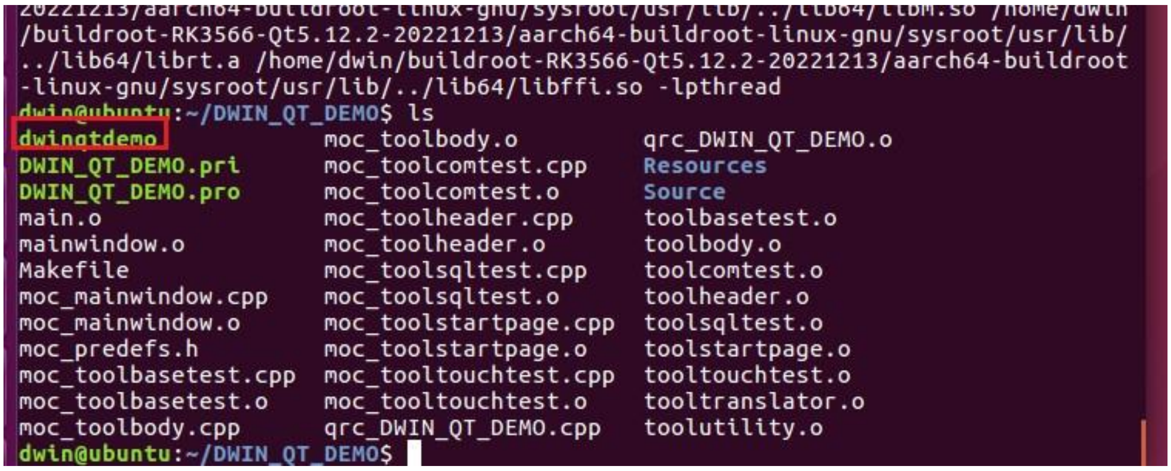
(1) After enter the environment (running the "**source env-setup.sh**" command), enter the command

"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
```



```
20221213/aarch64-buildroot-linux-gnu/sysroot/usr/lib/../lib64/libm.so /home/dwin
/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
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$
```

3.5 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.

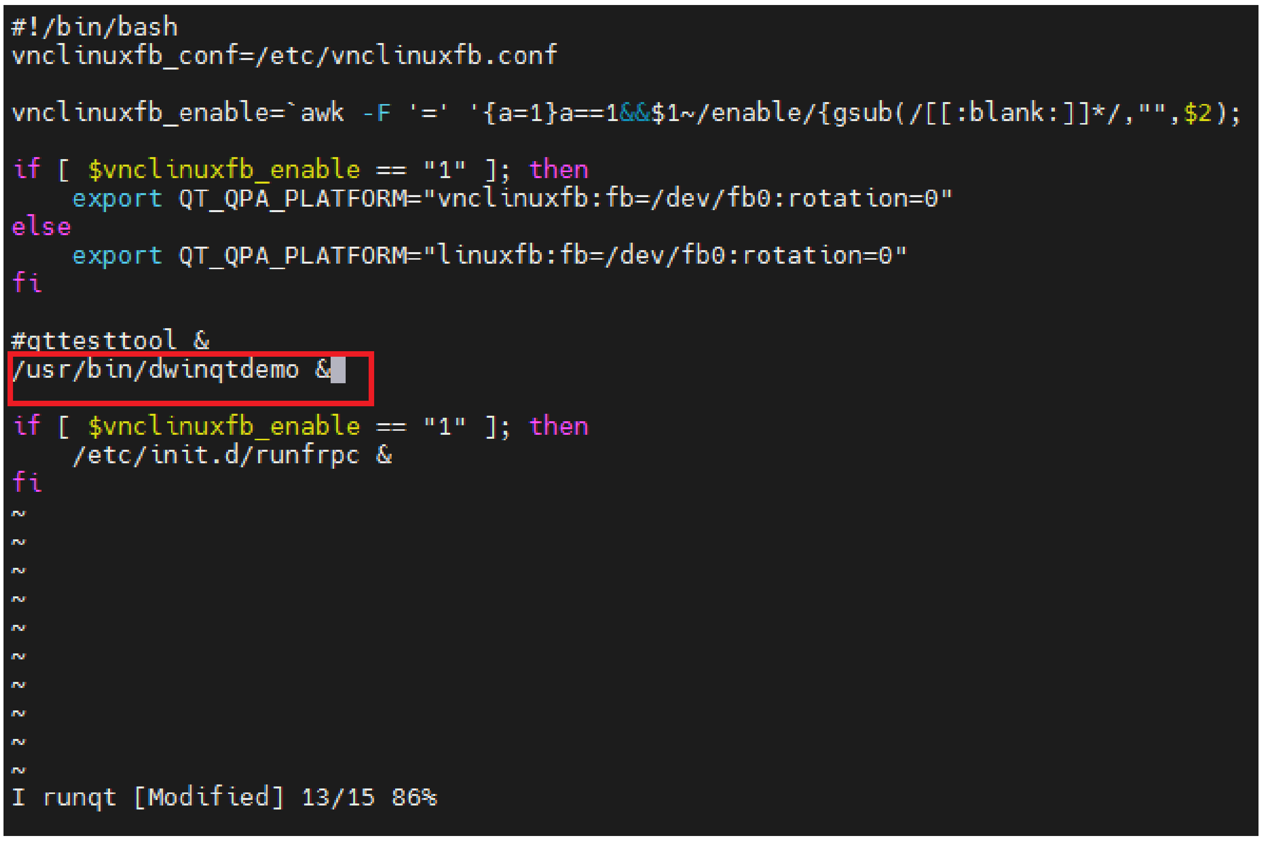
(3) Insert the USB into the development board.

(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

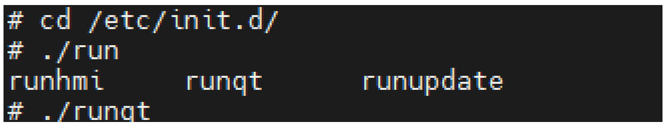
Configuration file should be modified to run the demo.

Enter the command: **vi /etc/init.d/runqt(enter)** 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.

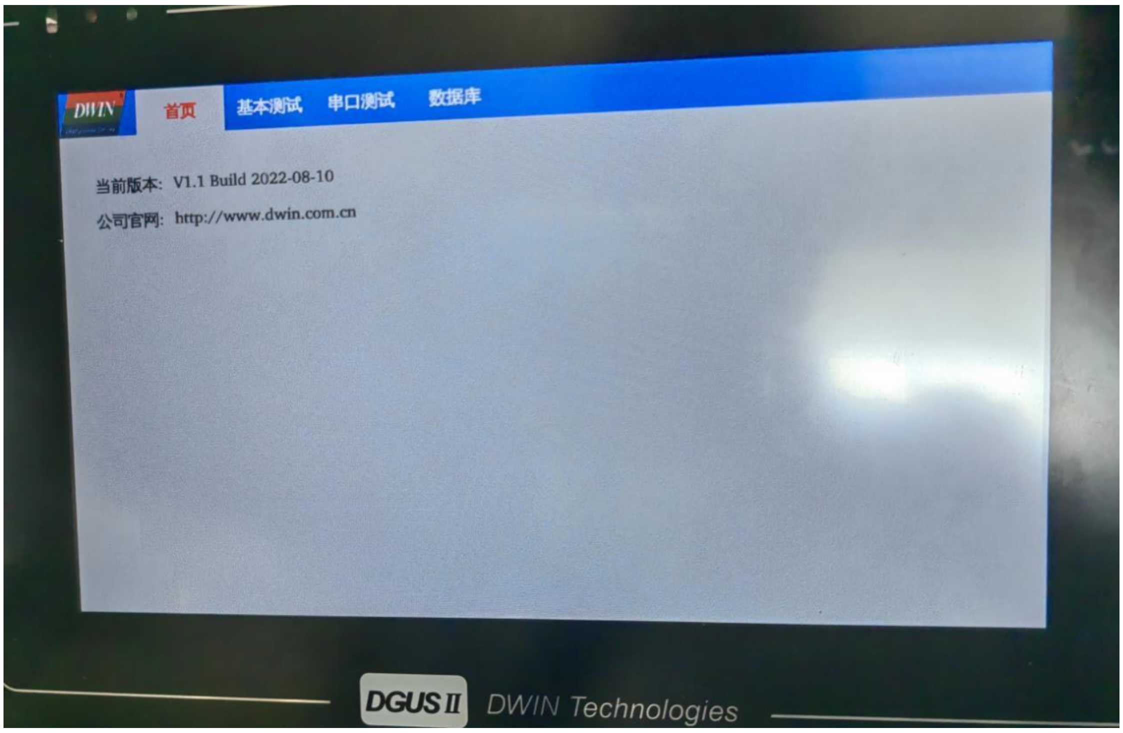


```
#!/bin/bash
vnclinuxfb_conf=/etc/vnclinuxfb.conf

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
```



当前版本: V1.1 Build 2022-08-10

公司官网: http://www.dwin.com.cn

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 Download SDK

First prepare an empty folder to place SDK, better under home, here we use **~/proj** as example.

Attention: To avoid unnecessary errors, please do not place/unzip the SDK in VM shared folders or non-english directories.

Install a few software packages before getting SDK:

```
sudo apt update

sudo apt install -y repo git python
```

Using the command **repo** requires a higher network standard. Choose whether to use it.

Get full SDK options：

Please kindly note: contact sales to get the account and password of DWIN server, then submit the key.

```
mkdir ~/proj/rk356x_linux_release_v1.3.0b_20221213/

cd ~/proj/rk356x_linux_release_v1.3.0b_20221213/



## full SDK
```

```
    repo init -u ssh://dwin@192.168.10.107:/work/gitwork/platform/rk3566_linux/manifest.git --repo-url
ssh://dwin@192.168.10.107:/work/gitwork/platform/rk3566_linux/repo.git
```

4.1.2 Sync Code

Execute the following command to synchronize the code:

```
# enter the SDK root directory
cd ~/proj/rk356x_linux_release_v1.3.0b_20221213/


# sync
.repo/repo/repo sync -c
.repo/repo/repo sync -c --no-tags
.repo/repo/repo start firefly --all
```

You can use the following command to update the SDK later:

```
.repo/repo/repo sync -c --no-tags
```

Probably due to the unstable network environment， **.repo/repo/repo sync -c --no-tags** may fail to update. You can execute it repeatedly.

4.1.3 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
```

4.1.4 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:

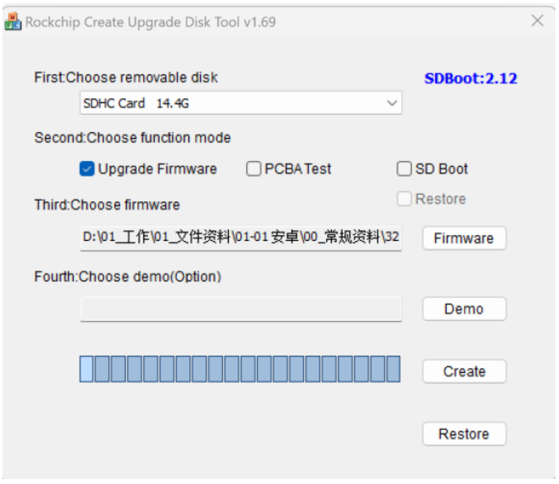
Run SDDiskTool_v1.69，check the "Upgrade Firmware" box and select the correct removable disk device.

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.


Rockchip Create Upgrade Disk Tool v1.69

First:Choose removable disk: SDHC Card 14.4G

SDBoot:2.12

Second:Choose function mode: [✓ Upgrade Firmware] PCBA Test SD Boot Restore

Third:Choose firmware: D:\01_工作\01_文件资料\01-01 安卓\00_常规资料\32

Fourth:Choose demo(Option)

4.4.2 Upgrade the firmware via Micro USB

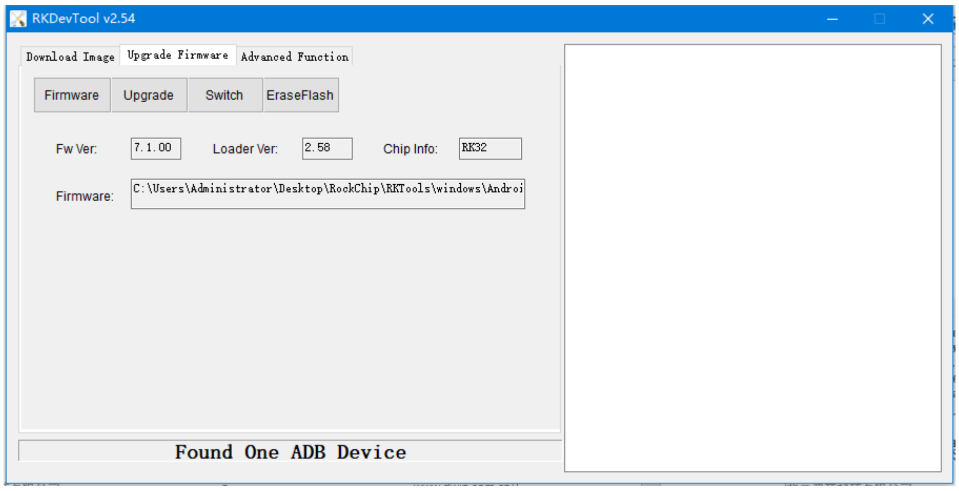
If the computer is being used for the first time to perform the burn-in process, you need to install the driver.

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.

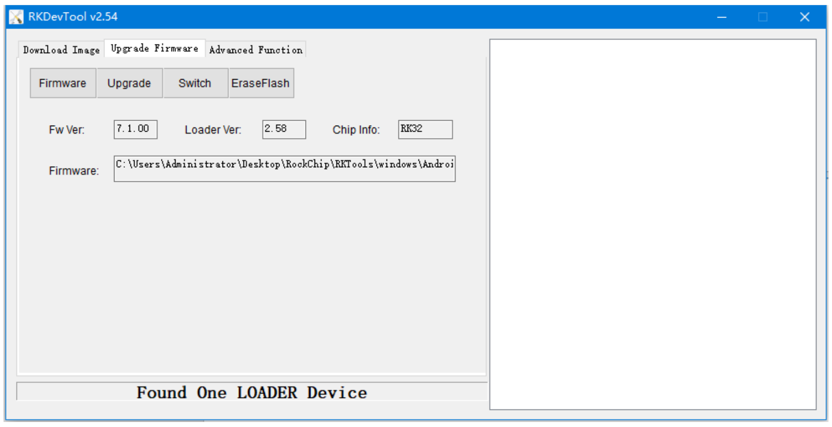


Fw Ver: 7.1.00 Loader Ver: 2.58 Chip Info: RK32

Firmware: C:\Users\Administrator\Desktop\RockChip\RKTools\windows\Androi

Found One ADB Device

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.



Fw Ver: 7.1.00 Loader Ver: 2.58 Chip Info: RK32

Firmware: C:\Users\Administrator\Desktop\RockChip\RKTools\windows\Androi

Found One LOADER Device

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

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