DWIN Linux screens development guide (35 series)

1 Environment Configuration

1.1 Ubuntu16.04 configuration

1.1.1 Introduction

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

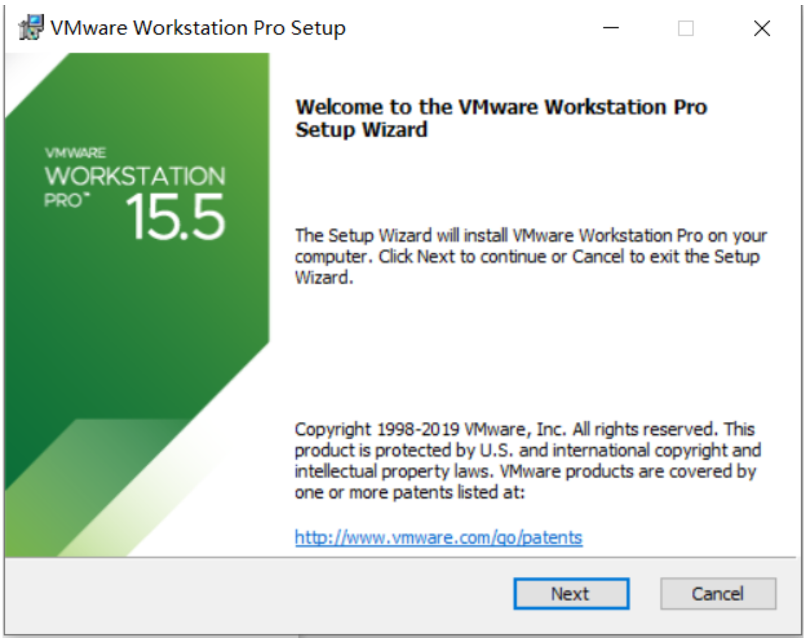
1.1.2 Environment Requirements

- CPU: No specific requirements.
- Memory: Generally, 2GB or more.
- Host Operating System: Windows XP, Windows 7, and above.
- Version Selection: Depending on your needs (Windows version), choose VMware Workstation 10 or a higher version. Versions below 10 are not recommended.

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

1.2 VMware Workstation Installation Steps

Running installation package



VMware Workstation Pro Setup

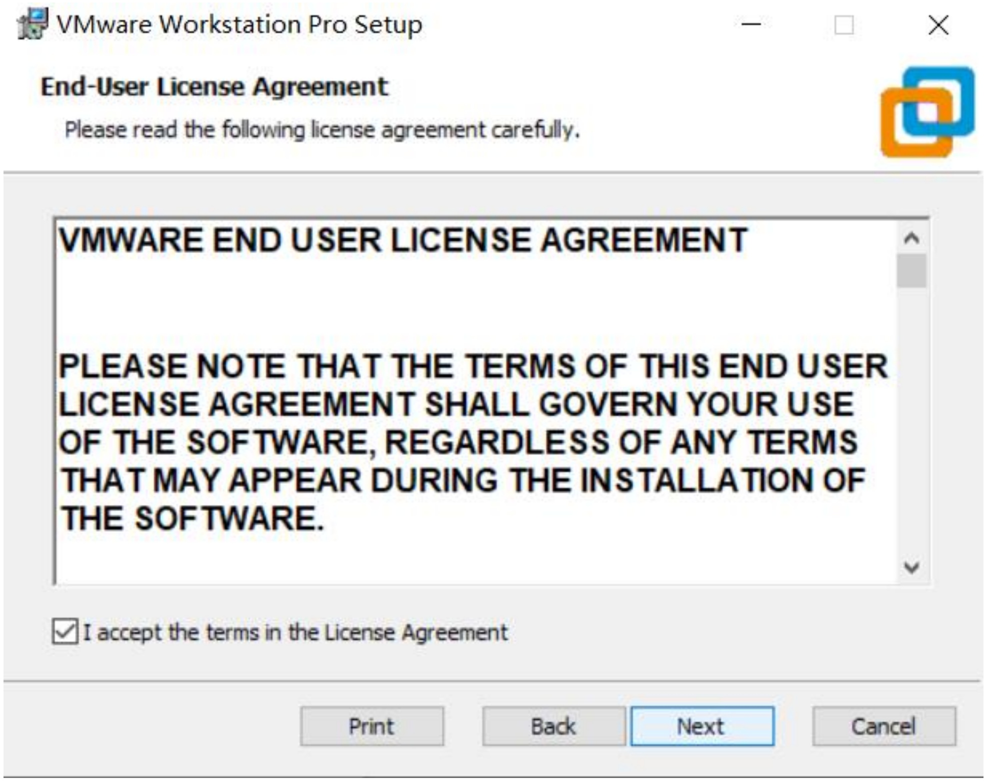
Welcome to the VMware Workstation Pro Setup Wizard

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

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

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

Select 'I accept the terms in the License Agreement' and click 'next'.



VMware Workstation Pro Setup

End-User License Agreement

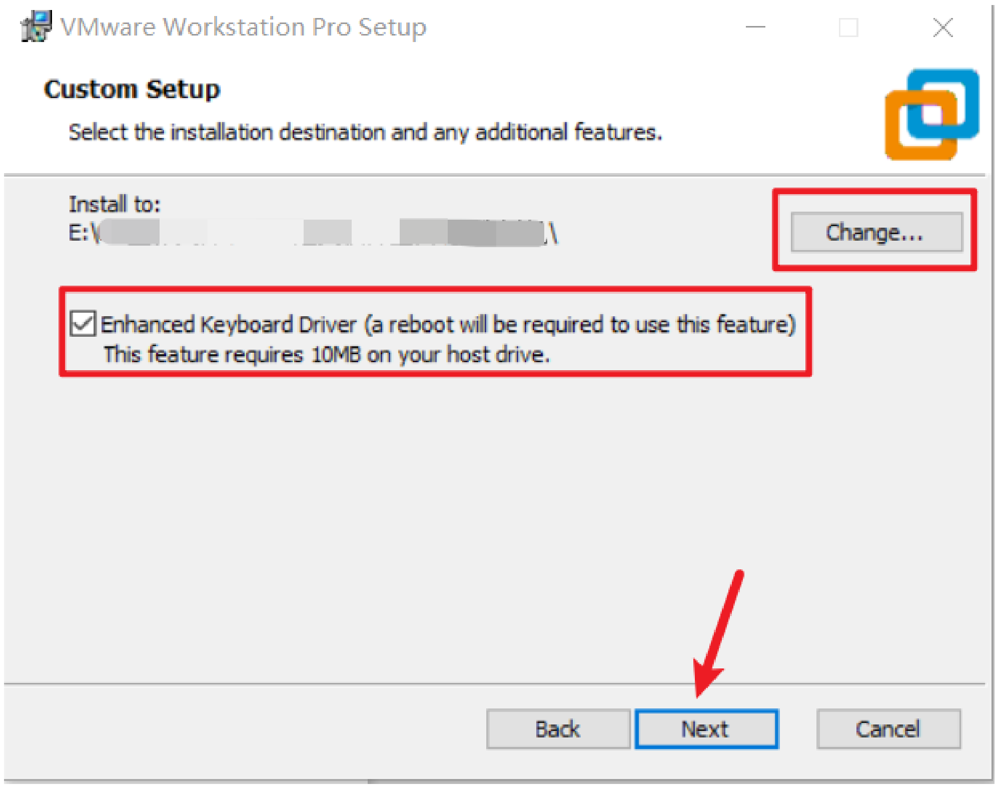
Please read the following license agreement carefully.

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

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


VMware Workstation Pro Setup

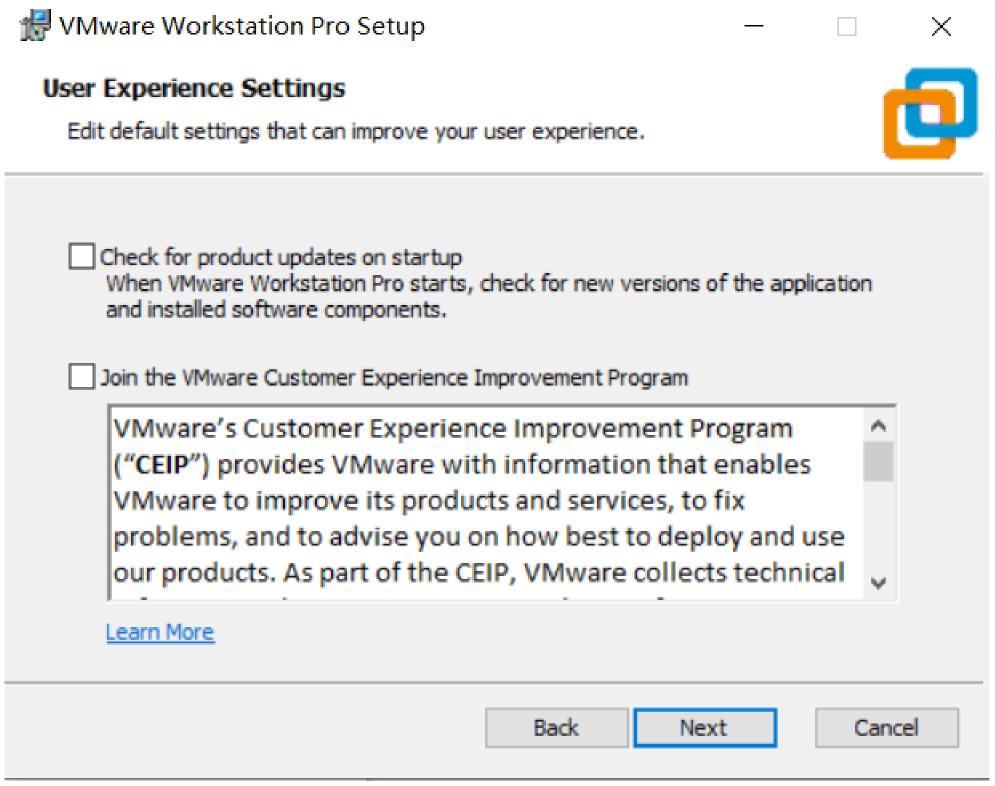
Custom Setup

Select the installation destination and any additional features.

Install to:
E:\[redacted]\

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

According on your preference, selectively choose ‘Check for product updates on startup’ and ‘Join the VMware customer experience improvement program’ checkboxes. Then, click ‘Next.’


VMware Workstation Pro Setup

User Experience Settings

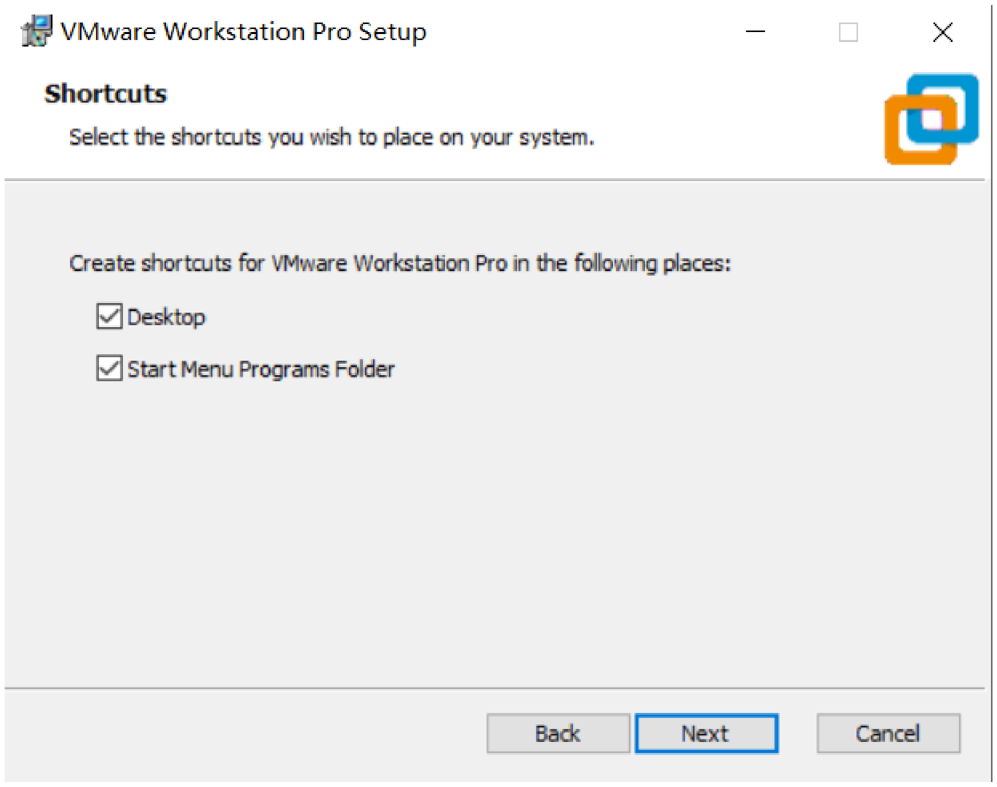
Edit default settings that can improve your user experience.

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

☐ Join the VMware Customer Experience Improvement Program

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

Select ‘desktop’ and ‘start menu programs folder’, then click ‘Next’.


VMware Workstation Pro Setup

Shortcuts

Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Workstation Pro in the following places:

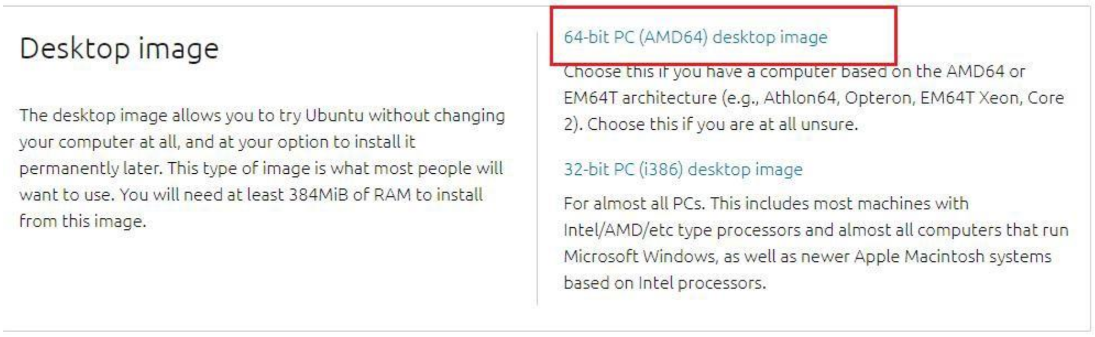
☑ Desktop

☑ Start Menu Programs Folder

Click “install”. Wait until installation is complete. Click “Finish”.

1.2.1 Download Ubuntu

- Get Ubuntu16.04 from official website: https://releases.ubuntu.com/16.04/
- Choose and and download 64-bit PC desktop image ‘ubuntu-16.04.7-desktop-amd64.iso’ 64bit PC


Desktop image

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

64-bit PC (AMD64) desktop image

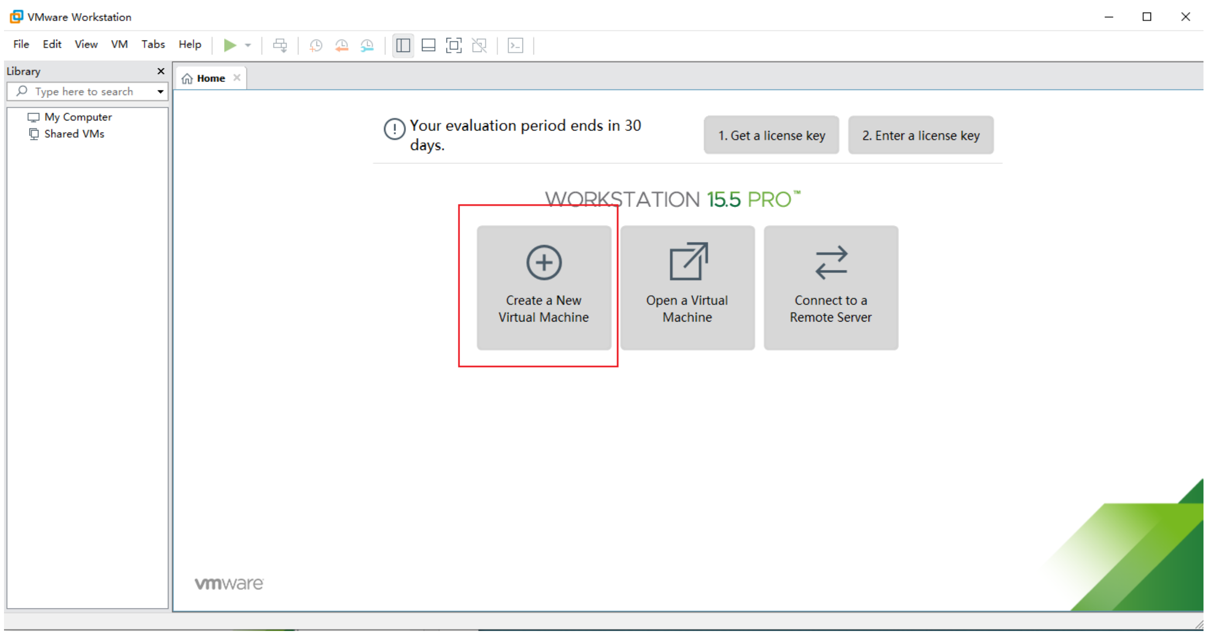
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

32-bit PC (i386) desktop image

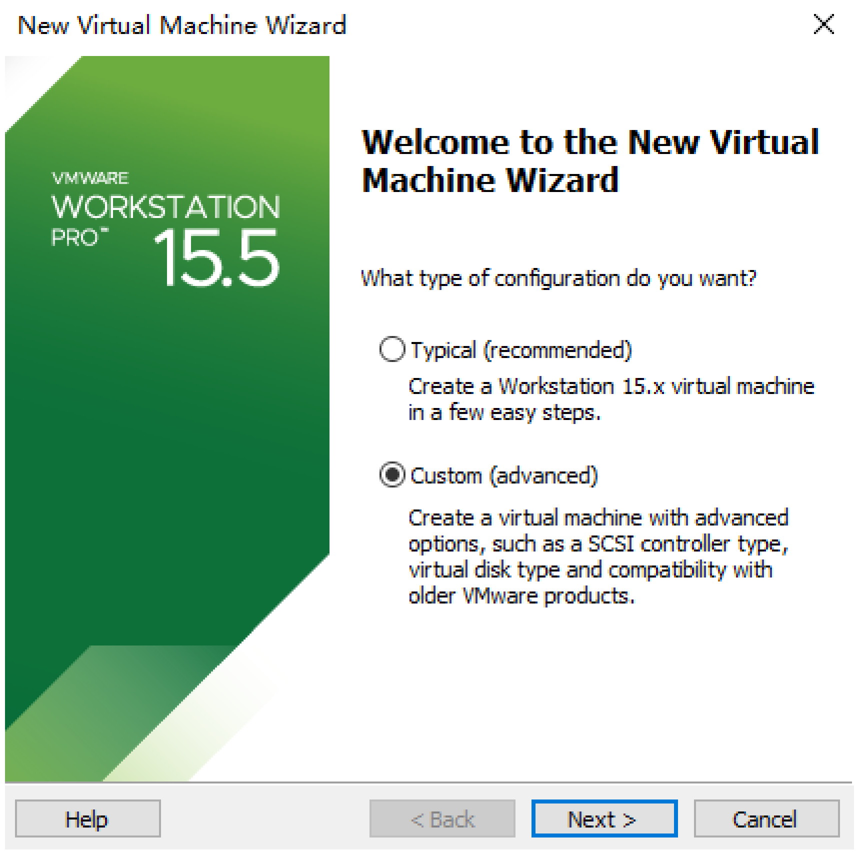
For almost all PCs. This includes most machines with Intel/AMD/etc type processors and almost all computers that run Microsoft Windows, as well as newer Apple Macintosh systems based on Intel processors.

Ubuntu Installation

Open VMware Workstation, create a new virtual machine.



Select 'custom (advanced)' and click 'Next'


New Virtual Machine Wizard

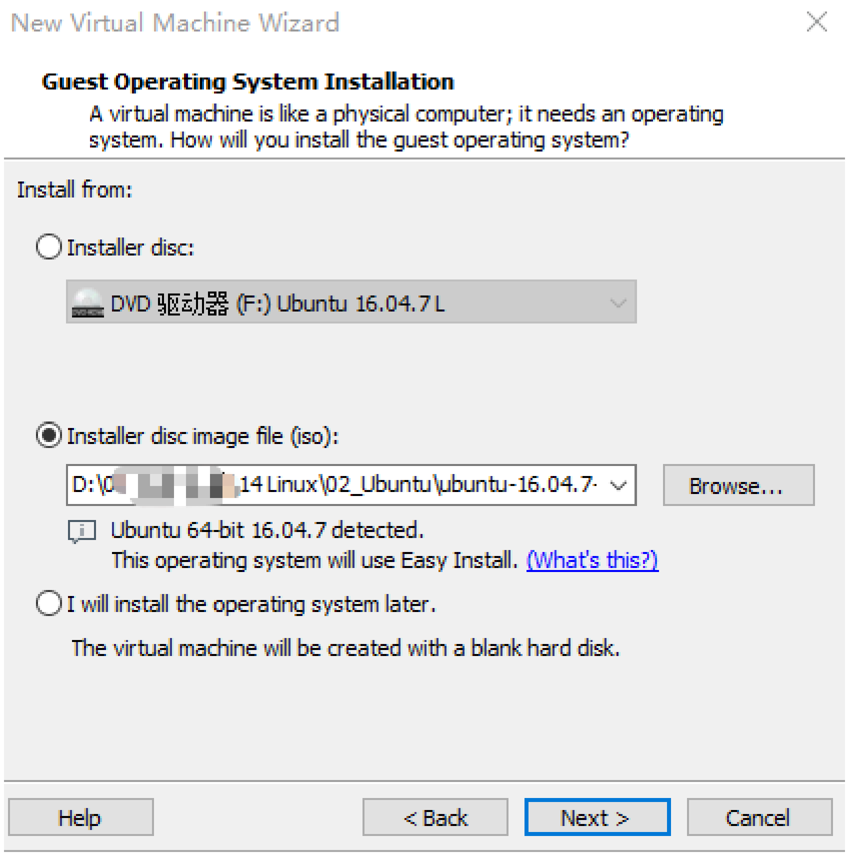
Welcome to the New Virtual Machine Wizard

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.

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


New Virtual Machine Wizard

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

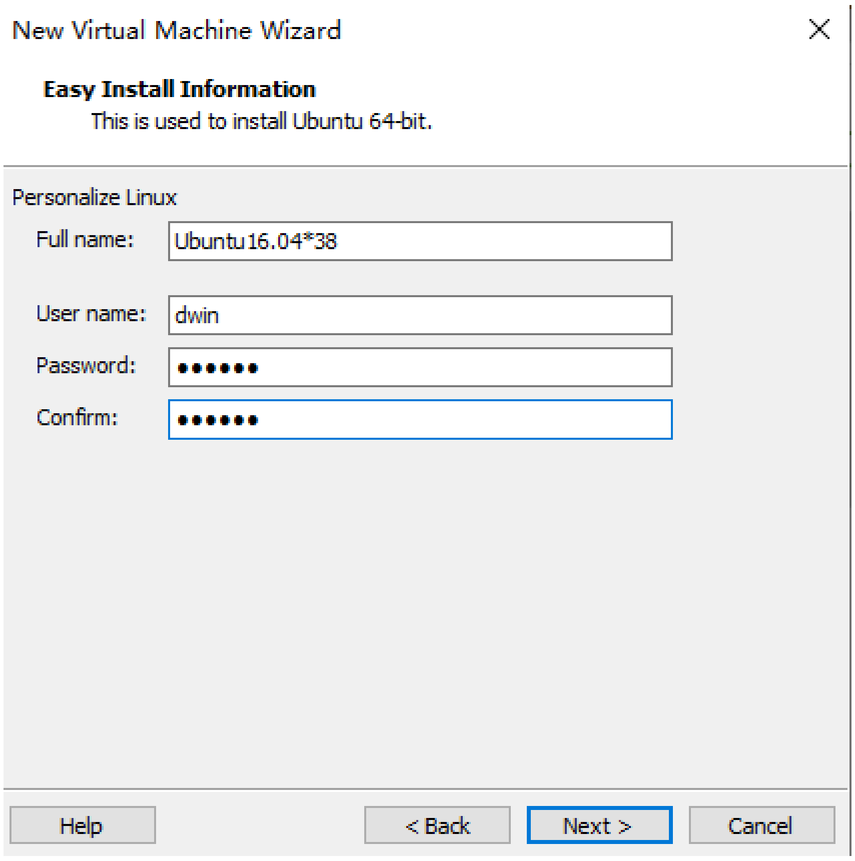
Install from:

( ) Installer disc:
DVD 驱动器 (F:) Ubuntu 16.04.7 L

(●) Installer disc image file (iso):
D:\0 14 Linux\02_Ubuntu\ubuntu-16.04.7-
Ubuntu 64-bit 16.04.7 detected.
This operating system will use Easy Install. (What's this?)

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

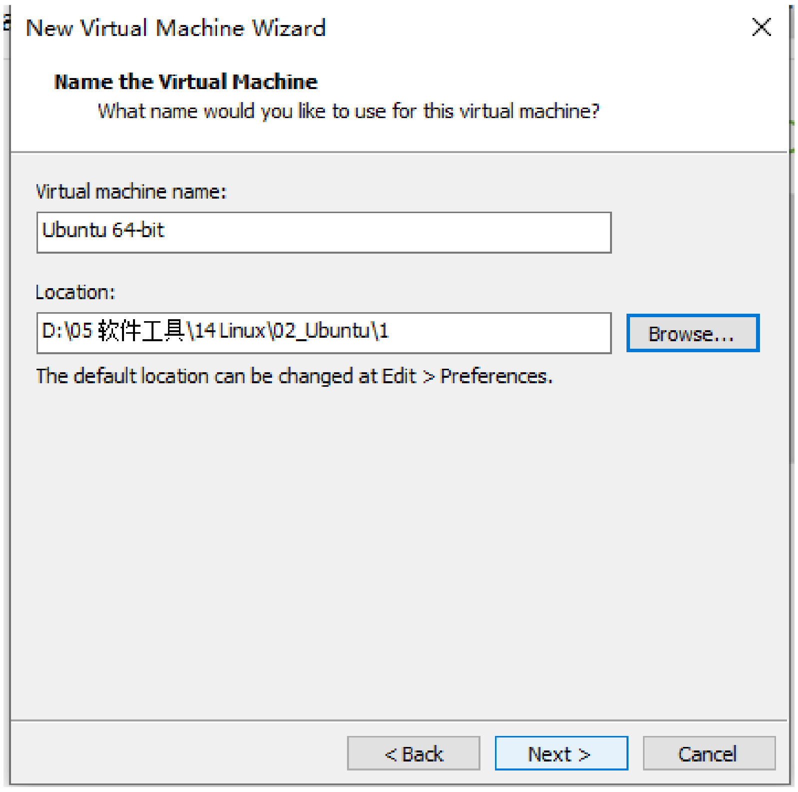
Input custom name and password, the password will serve as the login password for Ubuntu and the sudo authorization password, then click ‘Next’.


New Virtual Machine Wizard

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

Personalize Linux
Full name: Ubuntu16.04*38
User name: dwin
Password: ●●●●●●
Confirm: ●●●●●●

Setup the name of Ubuntu and location, click 'Next'.


New Virtual Machine Wizard

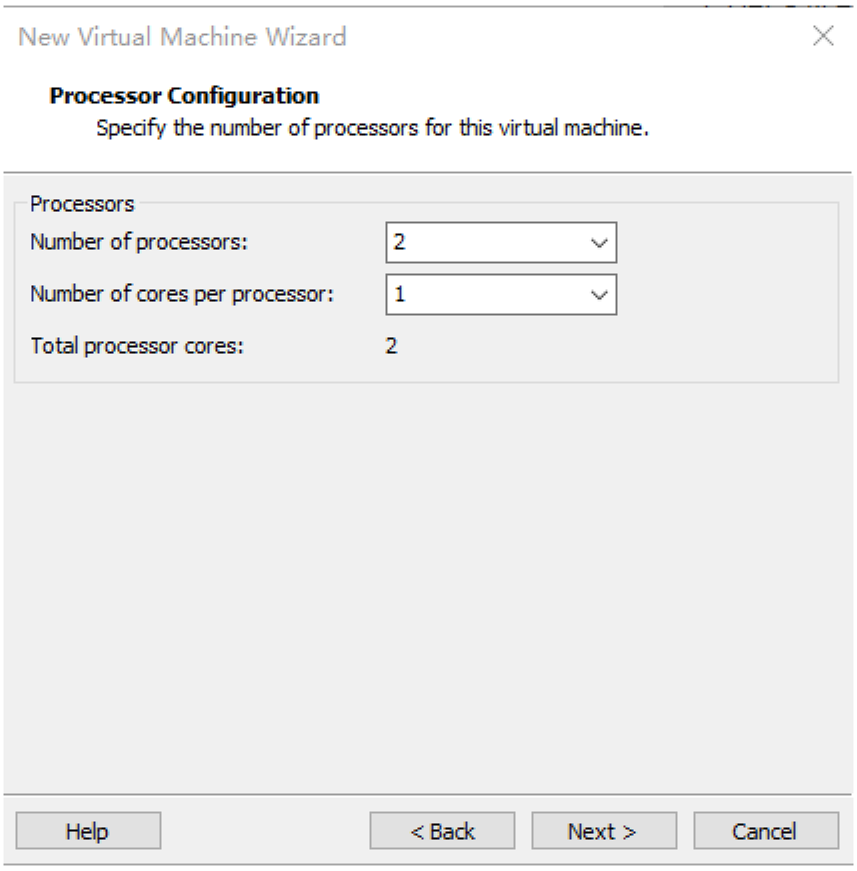
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name: Ubuntu 64-bit

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

The default location can be changed at Edit > Preferences.

Based on your requirements and computer configuration, allocate the number of processors and cores. Here, the author sets the total number of processor cores to 2. Click ‘Next’.



New Virtual Machine Wizard

Processor Configuration
Specify the number of processors for this virtual machine.

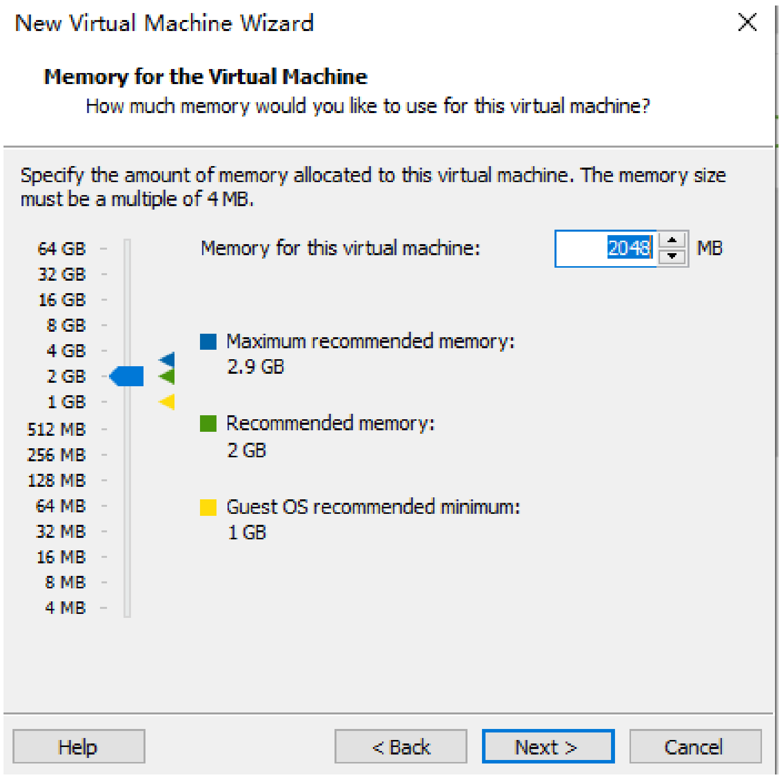
Processors

Number of processors: 2

Number of cores per processor: 1

Total processor cores: 2

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



New Virtual Machine Wizard

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

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

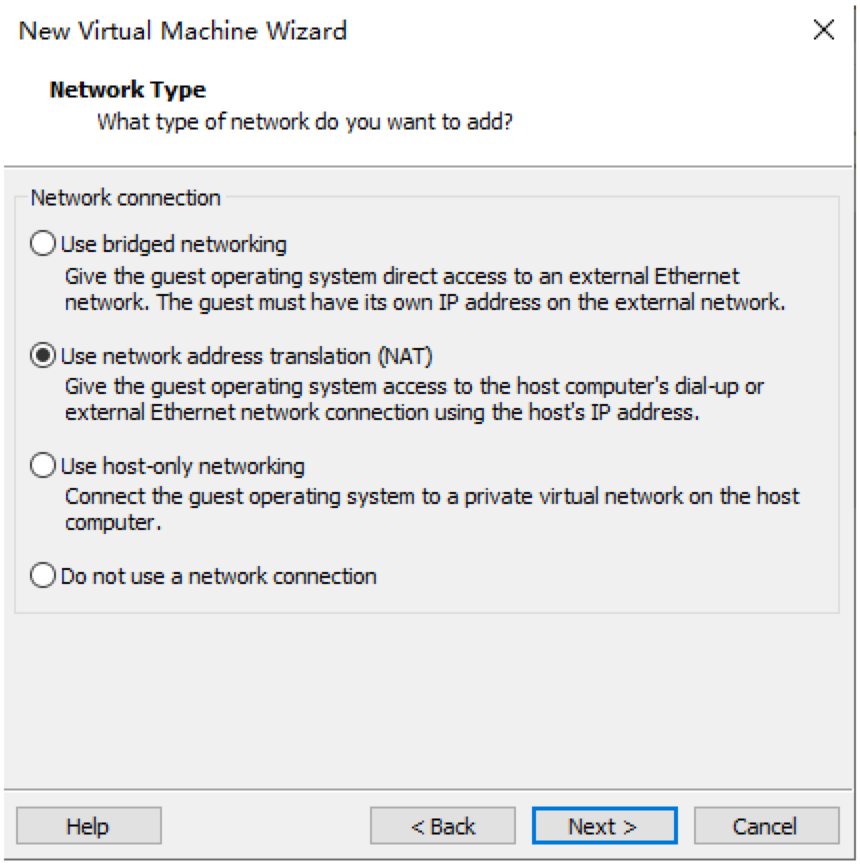
Memory for this virtual machine: 2048 MB

🟦 Maximum recommended memory: 2.9 GB

🟩 Recommended memory: 2 GB

🟨 Guest OS recommended minimum: 1 GB

The default configuration is fine (although, in the Network Type, you can choose Bridged Networking, which is useful for TFTP transfers). Click 'Next'.



Network Type

What type of network do you want to add?

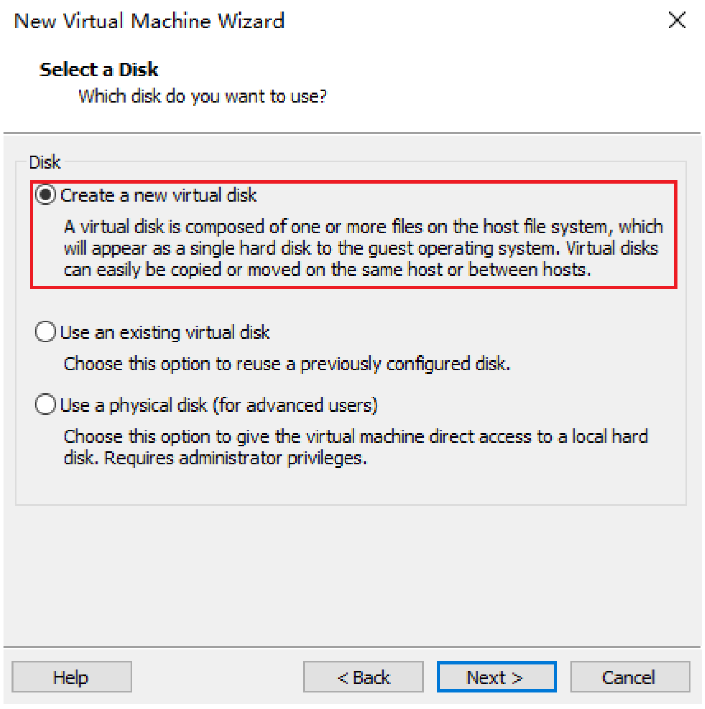
Network connection

- ( ) 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

Choose 'Recommend' and 'Next'

Choose 'Recommend' and 'Next'

Select 'Create a new virtual disk', and click 'Next'.


New Virtual Machine Wizard

Select a Disk
Which disk do you want to use?

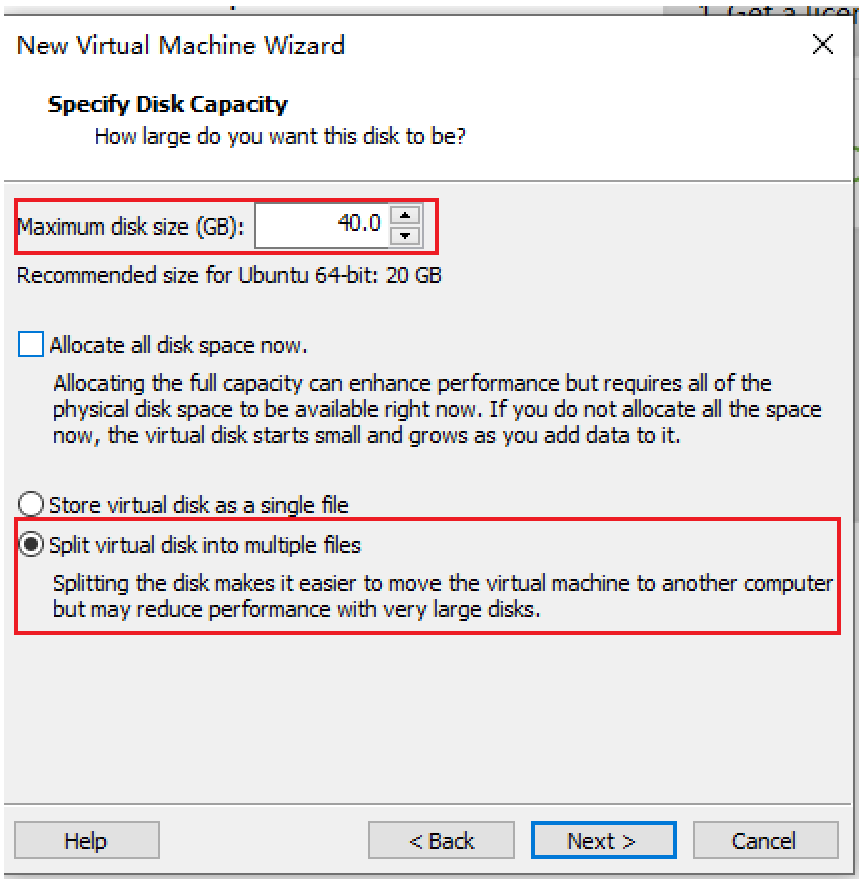
Disk

(●) 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.

Specify the disk capacity. If your computer has sufficient memory, it's advisable to set it larger, ideally 30GB or more. Click ‘Next’. If the disk capacity is insufficient, you can expand it later (as explained in subsequent chapters).


New Virtual Machine Wizard

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

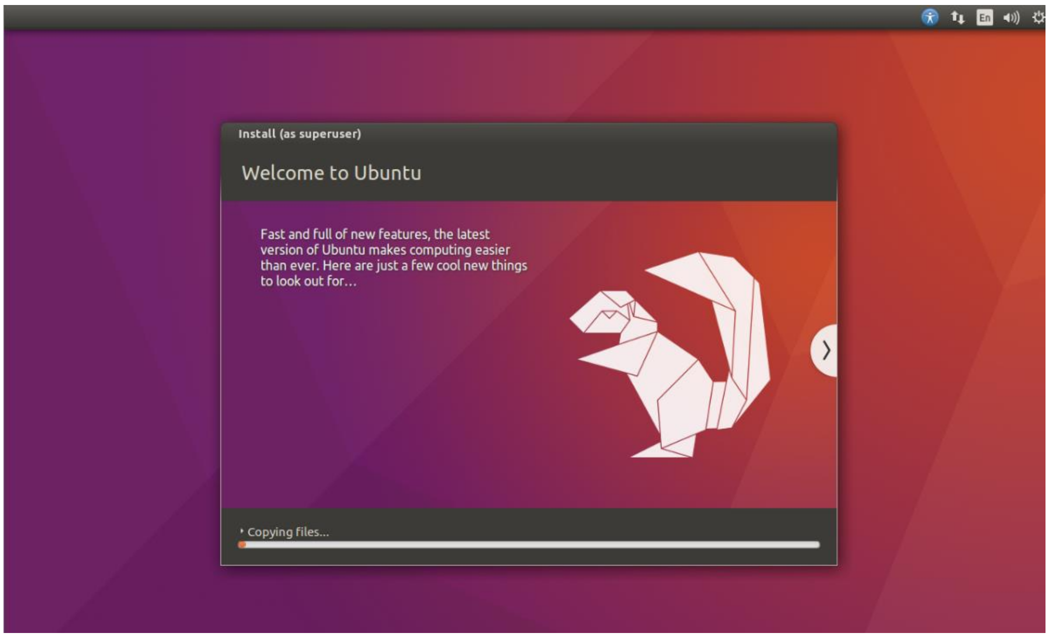
Maximum disk size (GB): 40.0

Recommended size for Ubuntu 64-bit: 20 GB

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

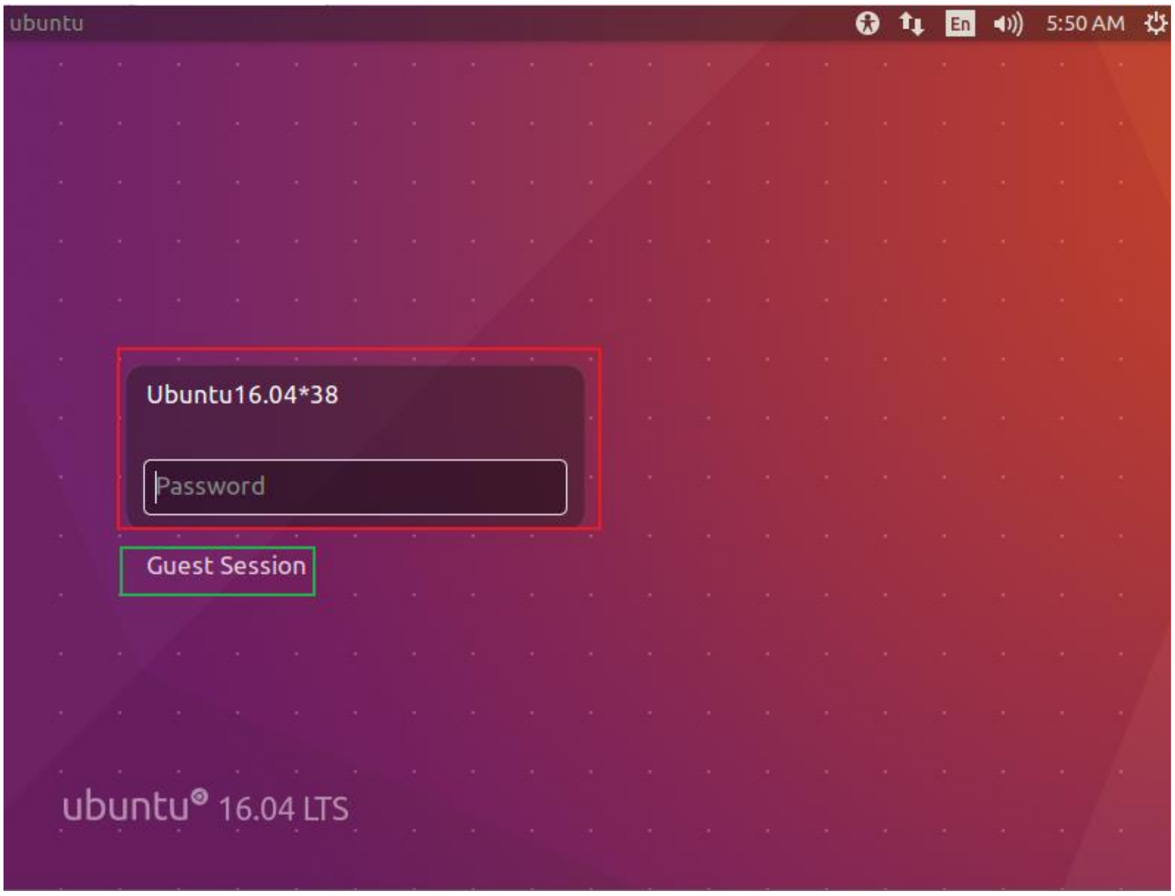
( ) Store virtual disk as a single file
(●) Split virtual disk into multiple files
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

The disk will be automatically named; keep the default and click ‘Next’.
After clicking ‘Finish’, the virtual machine will start and the installation will begin.
Be patient and wait awhile.



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

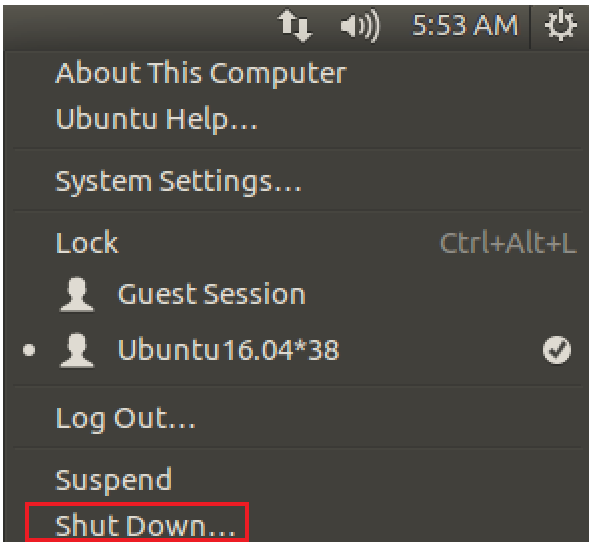
After this interface appears, the installation of Ubuntu is complete. (Note: The login screen has two user login inputs. The one in the red box is user-defined, while the one in the green box is system-provided.)



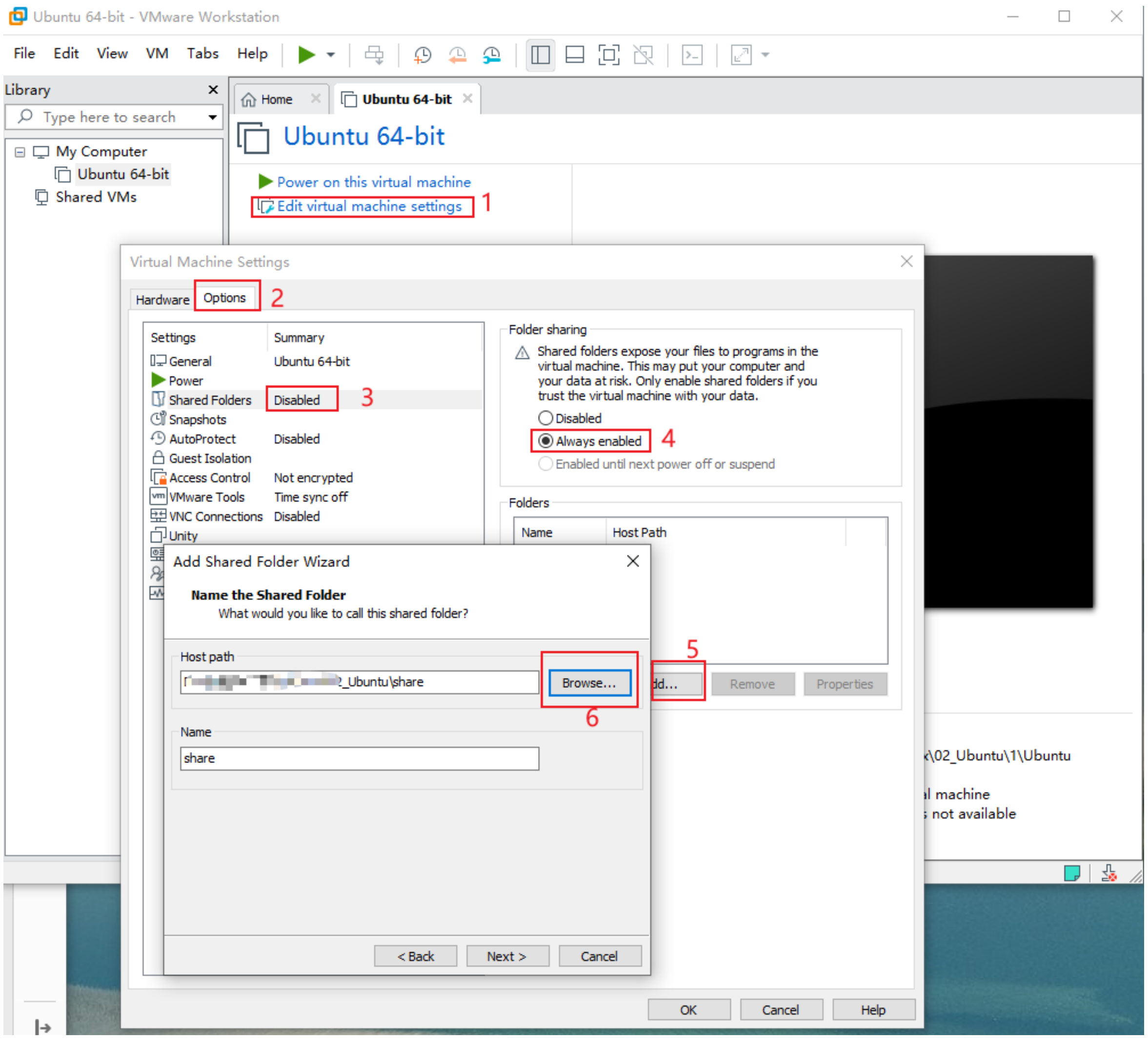
Next start to configure some necessary environments for Ubuntu.

1.2.2 Setting up the shared folders

Shut down Ubuntu by clicking ‘shut down’ in the top-right corner of the desktop.



After shutting down, on the corresponding virtual machine page, click ‘Edit Virtual Machine Settings’ -> ‘Options’ -> ‘Shared Folders’ -> ‘Always Enabled’ -> ‘Add’. Add a folder to serve as a medium for transferring files between the host and the virtual machine. Finally, click ‘OK’.



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

Folder sharing

Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

( ) Disabled [✓ Always enabled] ( ) Enabled until next power off or suspend

Folders

Name	Host Path

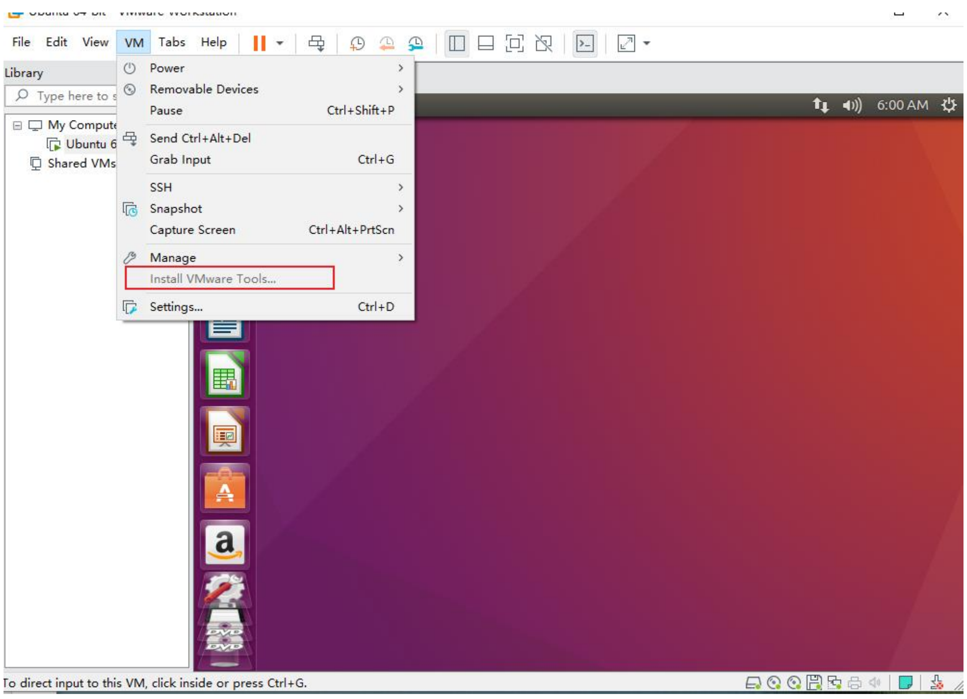
Name the Shared Folder

What would you like to call this shared folder?

Host path: _Ubuntu\share

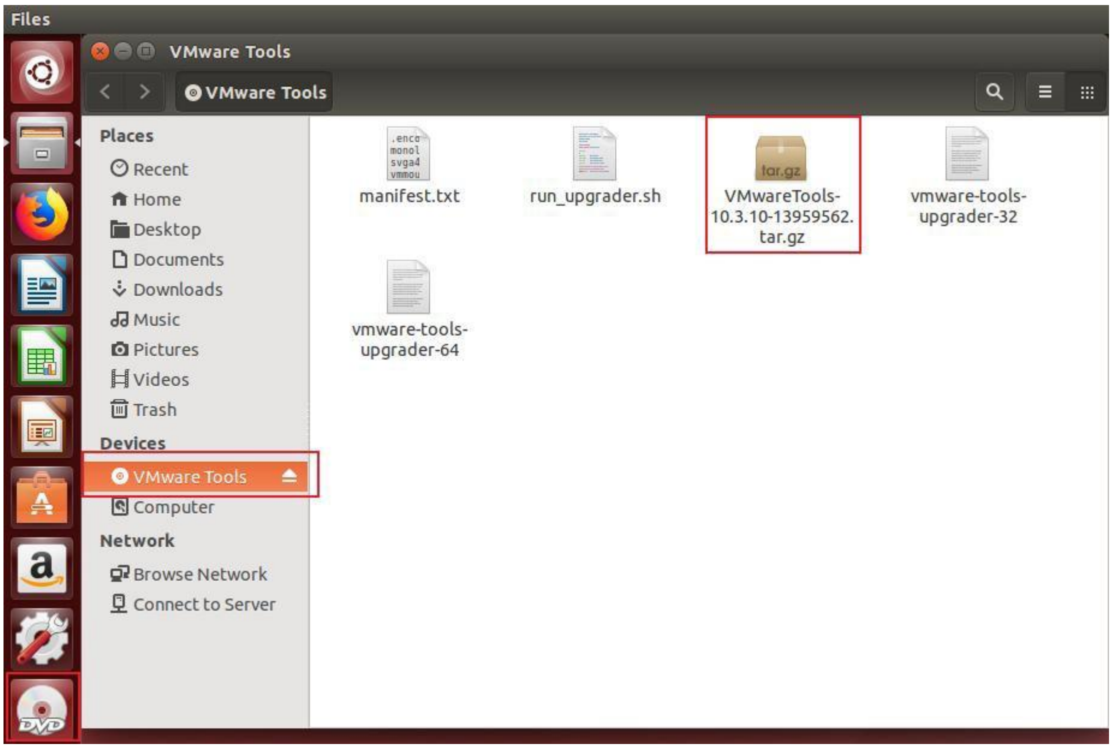
Name: share

When starting up, click ‘Virtual Machine’ -> ‘Install VMware Tools’ (Note: It must be selected during startup; otherwise, it will be a grayed-out and unselectable option. If have installed VMware Tools, so it will display 'Reinstall VMware Tools').



To direct input to this VM, click inside or press Ctrl+G.

Click ‘DVD’ icon and open to see a compressed file ‘VMwareTools-10.3.10-12406962.tar.gz’.

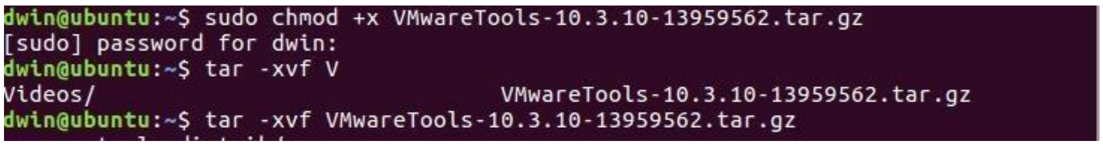


Right-click and choose "Copy to" to a path with permissions, you can directly copy it to "home."

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

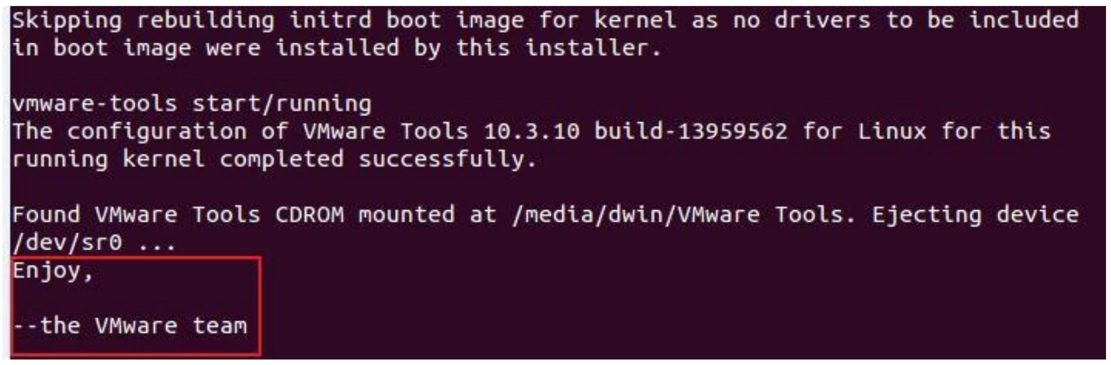
Enter the command to add executable permissions: **sudo chmod +x VM** (use the Tab key to display the full name) (Enter) (Note: When using sudo privileges for the first time, you need to enter the password, and it won't be visible when entering the password).

Enter the decompression command: **tar -xvf VM** (Tab key) (Enter), and it will automatically decompress to generate 'vmware-tools-distrib' in the current directory. Enter the command: **cd vm** (Tab key) (Enter, subsequent steps will be omitted).



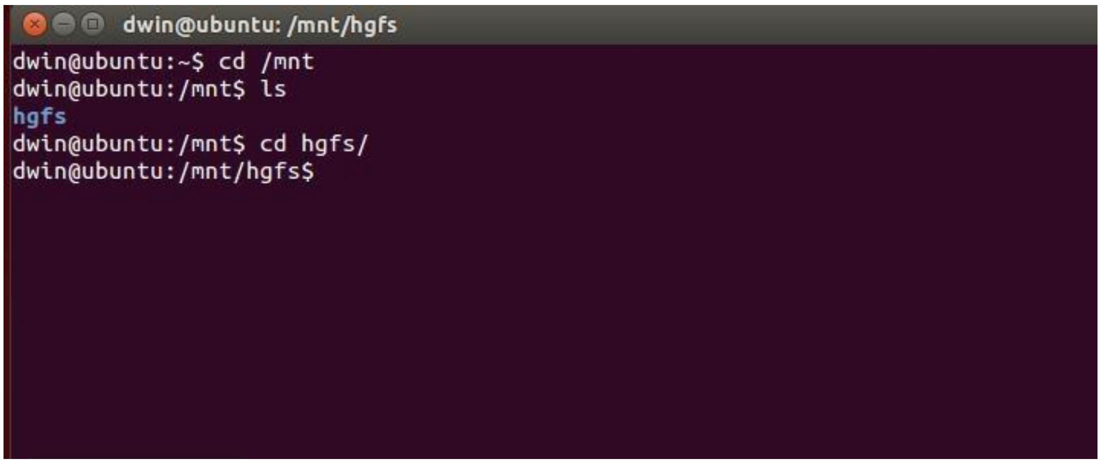
```
dwin@ubuntu:~$ sudo chmod +x VMwareTools-10.3.10-13959562.tar.gz
[sudo] password for dwin:
dwin@ubuntu:~$ tar -xvf V
Videos/                            VMwareTools-10.3.10-13959562.tar.gz
dwin@ubuntu:~$ tar -xvf VMwareTools-10.3.10-13959562.tar.gz
```

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



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

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

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

--the VMware team
```

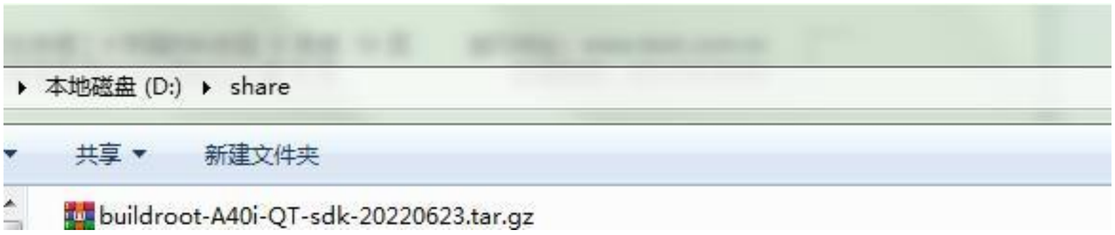
Now, enter the command: **cd /mnt** (use Tab to navigate to the shared folder you set), the path is /mnt/hfgs/*** (replace *** with your specific path). The shared folder is now set up, and you can proceed to install the Ti toolchain on Ubuntu.



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

1.3 **Installing the a40i Toolchain**

Move the a40i compressed package (buildroot-A40i-QT-sdk-20220623.tar.gz) to Ubuntu, you can use a shared folder or transfer via SFTP, etc.



buildroot-A40i-QT-sdk-20220623.tar.gz

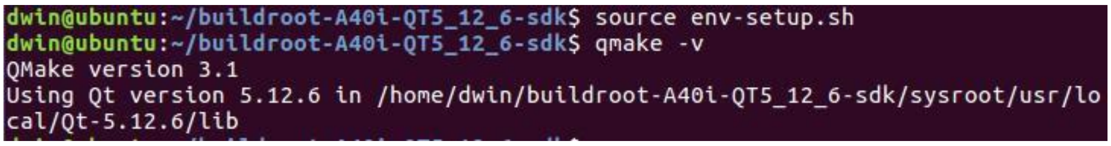
Move the file to the root directory (/home/dwin). When using a shared folder, enter the command: **sudo mv buil (Tab)**, wait a moment, and it will be moved to the root directory.

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

Enter the following commands one by one:

```

**cd bui(TAB)**

**source env-setup**

```

Input the **qmake-V** to check the version information to see if it is set up successfully.

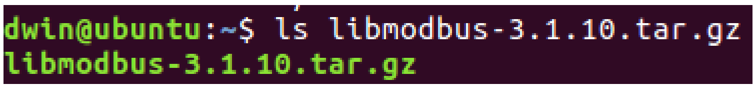


```
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ source env-setup.sh
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ qmake -v
QMake version 3.1
Using Qt version 5.12.6 in /home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot/usr/lo
cal/Qt-5.12.6/lib
```

1.4 Cross-compilation to generate dynamic link library example.

Here is an example using libmodbus.

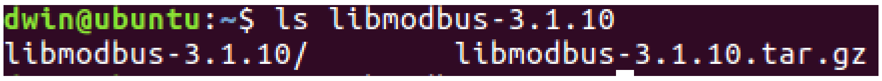
1.4.1 Copy the files to the specified directory and unzip them.



```
dwin@ubuntu:~$ ls libmodbus-3.1.10.tar.gz
libmodbus-3.1.10.tar.gz
```

Unzip:

```
tar -zxvf libmodbus-3.1.10.tar.gz
```



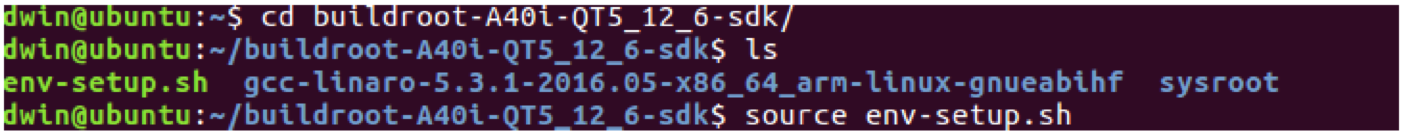
```
dwin@ubuntu:~$ ls libmodbus-3.1.10
libmodbus-3.1.10/         libmodbus-3.1.10.tar.gz
```

1.4.2 Create the installation directory.

```
mkdir install
chmod 777 install
```

1.4.3 Configure the compilation environment.

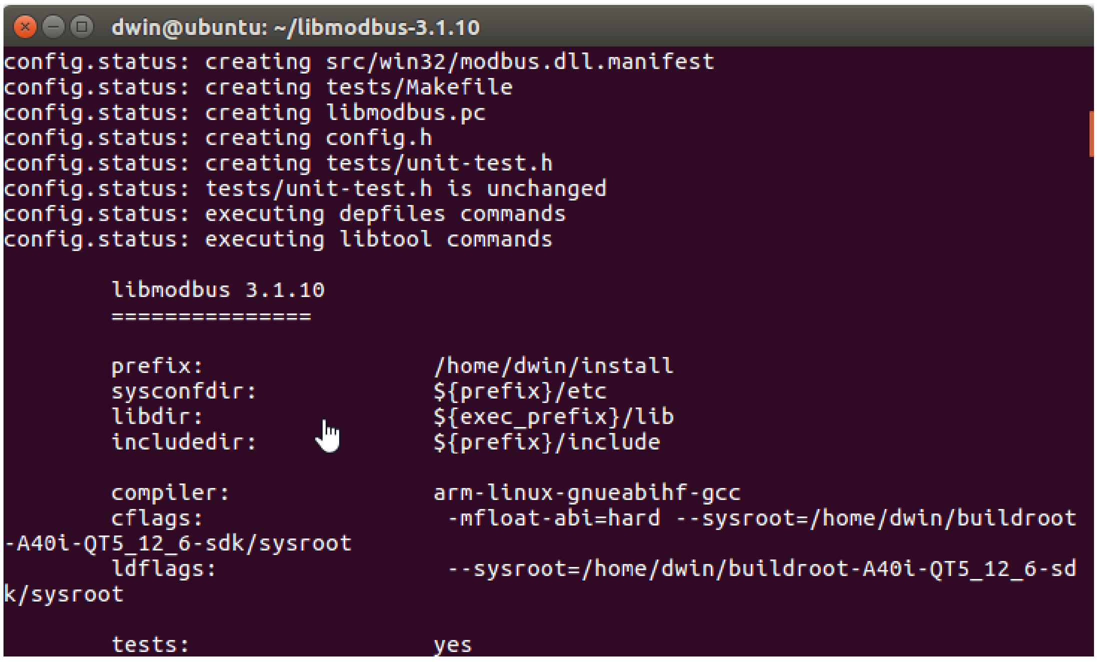
Navigate to the unpacked toolchain folder and run the env-setup.sh script.



```
dwin@ubuntu:~$ cd buildroot-A40i-QT5_12_6-sdk/
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ ls
env-setup.sh  gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabihf  sysroot
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ source env-setup.sh
```

1.4.4 Configure the compilation options.

```
// enter the libmodbus extraction directory
cd ../libmodbus-3.1.10
// ./configure --host=[ Cross-Compilation Toolchain Prefix] --enable-static --prefix=[install
path]/install/
./configure --host=arm-linux-gnueabihf --enable-static --prefix=/home/dwin/install/
```

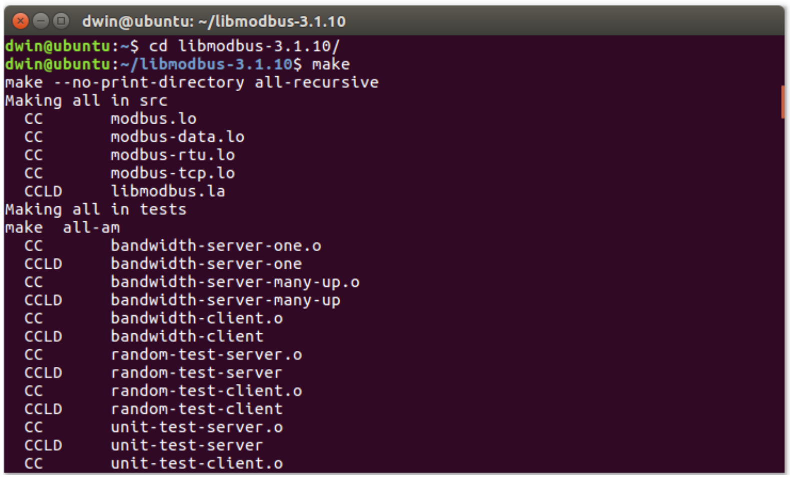


```
config.status: creating src/win32/modbus.dll.manifest
config.status: creating tests/Makefile
config.status: creating libmodbus.pc
config.status: creating config.h
config.status: creating tests/unit-test.h
config.status: tests/unit-test.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands

        libmodbus 3.1.10
        ===============

        prefix:                 /home/dwin/install
        sysconfdir:             ${prefix}/etc
        libdir:                 ${exec_prefix}/lib
        includedir:             ${prefix}/include

        compiler:               arm-linux-gnueabihf-gcc
        cflags:                  -mfloat-abi=hard --sysroot=/home/dwin/buildroot
-A40i-QT5_12_6-sdk/sysroot
        ldflags:                 --sysroot=/home/dwin/buildroot-A40i-QT5_12_6-sd
k/sysroot

        tests:                  yes
```

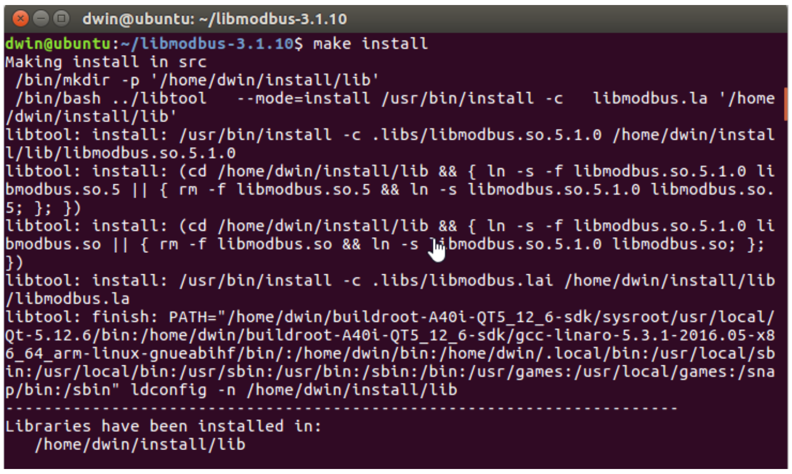
1.4.5 Compile and Install

```
make
```

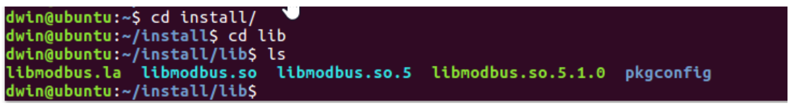
```
2.  make install
```



```
dwin@ubuntu: ~/libmodbus-3.1.10
dwin@ubuntu:~$ cd libmodbus-3.1.10/
dwin@ubuntu:~/libmodbus-3.1.10$ make
make --no-print-directory all-recursive
Making all in src
  CC       modbus.lo
  CC       modbus-data.lo
  CC       modbus-rtu.lo
  CC       modbus-tcp.lo
  CCLD     libmodbus.la
Making all in tests
make  all-am
  CC       bandwidth-server-one.o
  CCLD     bandwidth-server-one
  CC       bandwidth-server-many-up.o
  CCLD     bandwidth-server-many-up
  CC       bandwidth-client.o
  CCLD     bandwidth-client
  CC       random-test-server.o
  CCLD     random-test-server
  CC       random-test-client.o
  CCLD     random-test-client
  CC       unit-test-server.o
  CCLD     unit-test-server
  CC       unit-test-client.o
```



```
dwin@ubuntu: ~/libmodbus-3.1.10
dwin@ubuntu:~/libmodbus-3.1.10$ make install
Making install in src
 /bin/mkdir -p '/home/dwin/install/lib'
 /bin/bash ../libtool   --mode=install /usr/bin/install -c   libmodbus.la '/home
/dwin/install/lib'
libtool: install: /usr/bin/install -c .libs/libmodbus.so.5.1.0 /home/dwin/instal
l/lib/libmodbus.so.5.1.0
libtool: install: (cd /home/dwin/install/lib && { ln -s -f libmodbus.so.5.1.0 li
bmodbus.so.5 || { rm -f libmodbus.so.5 && ln -s libmodbus.so.5.1.0 libmodbus.so.
5; }; })
libtool: install: (cd /home/dwin/install/lib && { ln -s -f libmodbus.so.5.1.0 li
bmodbus.so || { rm -f libmodbus.so && ln -s libmodbus.so.5.1.0 libmodbus.so; };
})
libtool: install: /usr/bin/install -c .libs/libmodbus.lai /home/dwin/install/lib
/libmodbus.la
libtool: finish: PATH="/home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot/usr/local/
Qt-5.12.6/bin:/home/dwin/buildroot-A40i-QT5_12_6-sdk/gcc-linaro-5.3.1-2016.05-x8
6_64_arm-linux-gnueabihf/bin/:/home/dwin/bin:/home/dwin/.local/bin:/usr/local/sb
in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/sna
p/bin:/sbin" ldconfig -n /home/dwin/install/lib
----------------------------------------------------------------------
Libraries have been installed in:
   /home/dwin/install/lib
```



```
dwin@ubuntu:~$ cd install/
dwin@ubuntu:~/install$ cd lib
dwin@ubuntu:~/install/lib$ ls
libmodbus.la  libmodbus.so  libmodbus.so.5  libmodbus.so.5.1.0  pkgconfig
dwin@ubuntu:~/install/lib$
```

1.5 Screen Configuration

1.5.1 Hardware Introduction

Please refer to the related datasheet

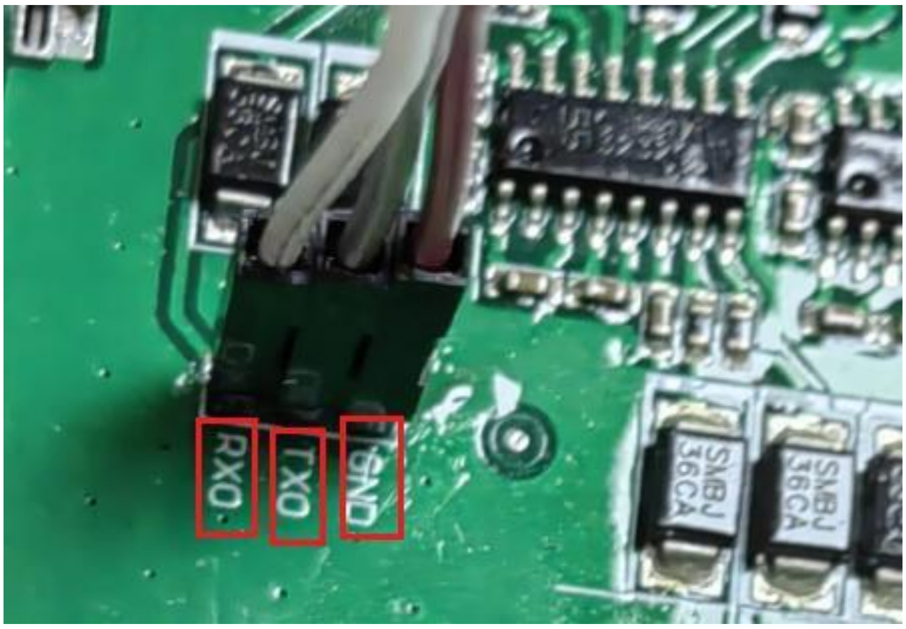
1.5.2 Terminal Software

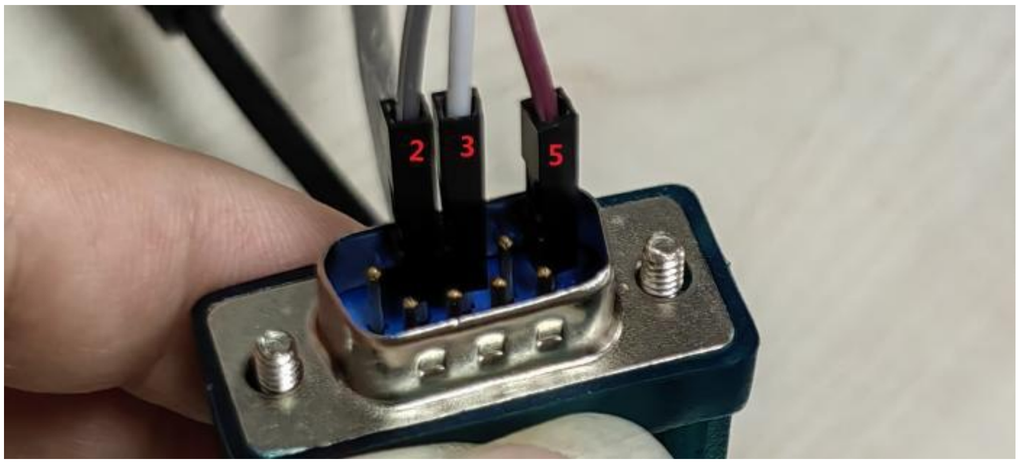
SecureCRT or MobaXterm are optional, we will introduce MobaXterm in this section.

Two types of connection: serial or Telnet.

1.5.3 Serial communication

Connect to Serial (Serial 0), RS232 connection in this example.

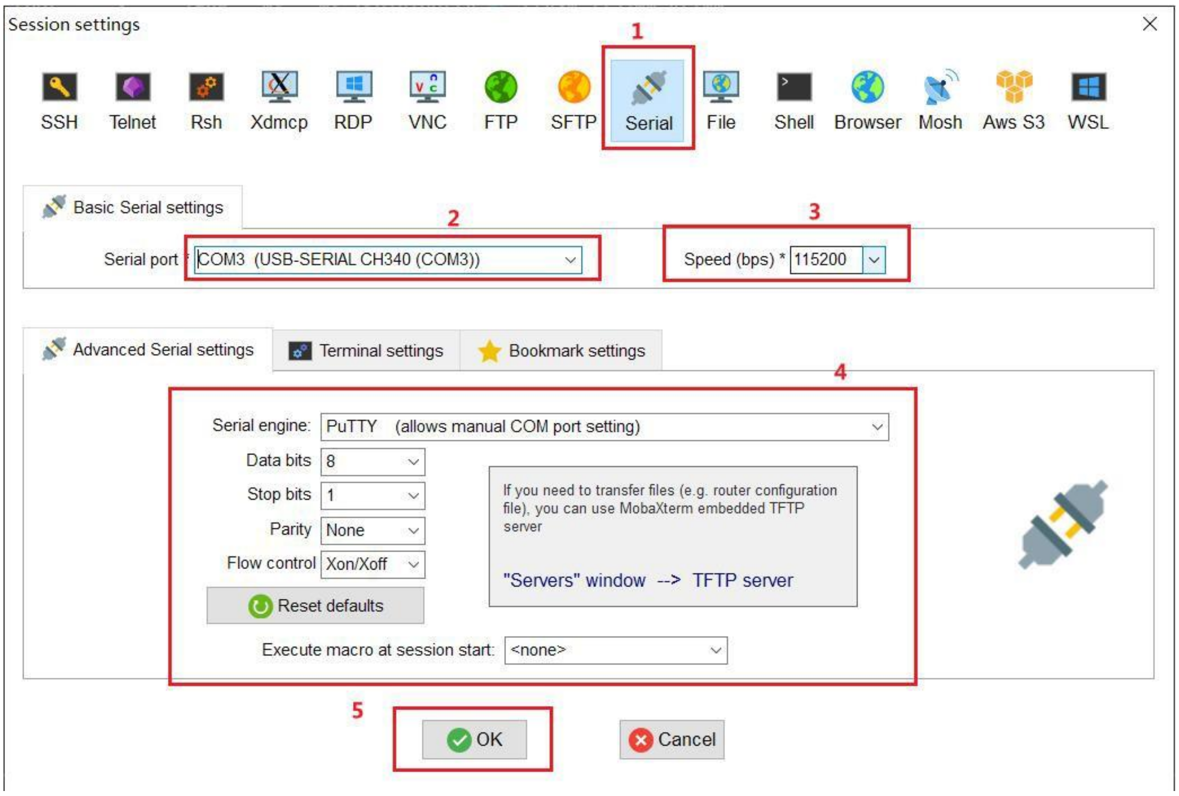




- Connect 2(RX) to TX, 3(TX) to RX, and 5(GND) to GND.

  Please use an RS232 or TTL interface, and refer to the datasheet for the corresponding port type (serial port 0).

- MobaXterm Configuration: session ->New session -> choose 'serial' ->select the serial port, set the baud rate in the third step, and cross-check the information ->click 'OK' to complete.



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

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

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

"Servers" window --> TFTP server

Execute macro at session start: <none>

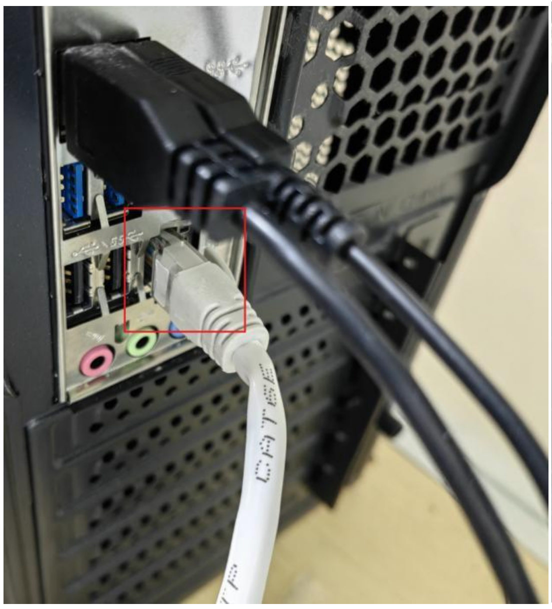
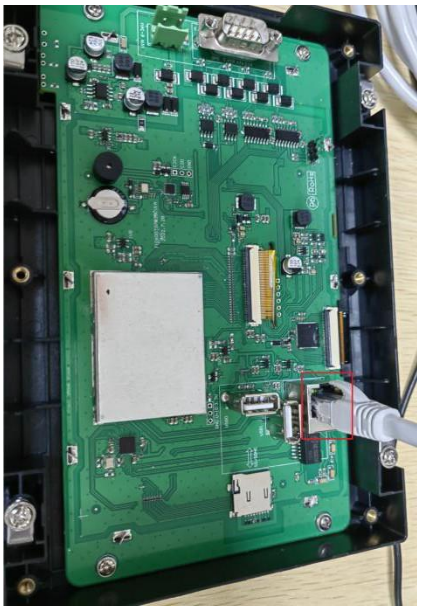
- Powering up the development board and input 'root' to begin operations. Note: if there's a delay between power-up and connection, and the screen shows no text, only a black screen, input 'root'.

1.5.4 Telnet Connection via Ethernet

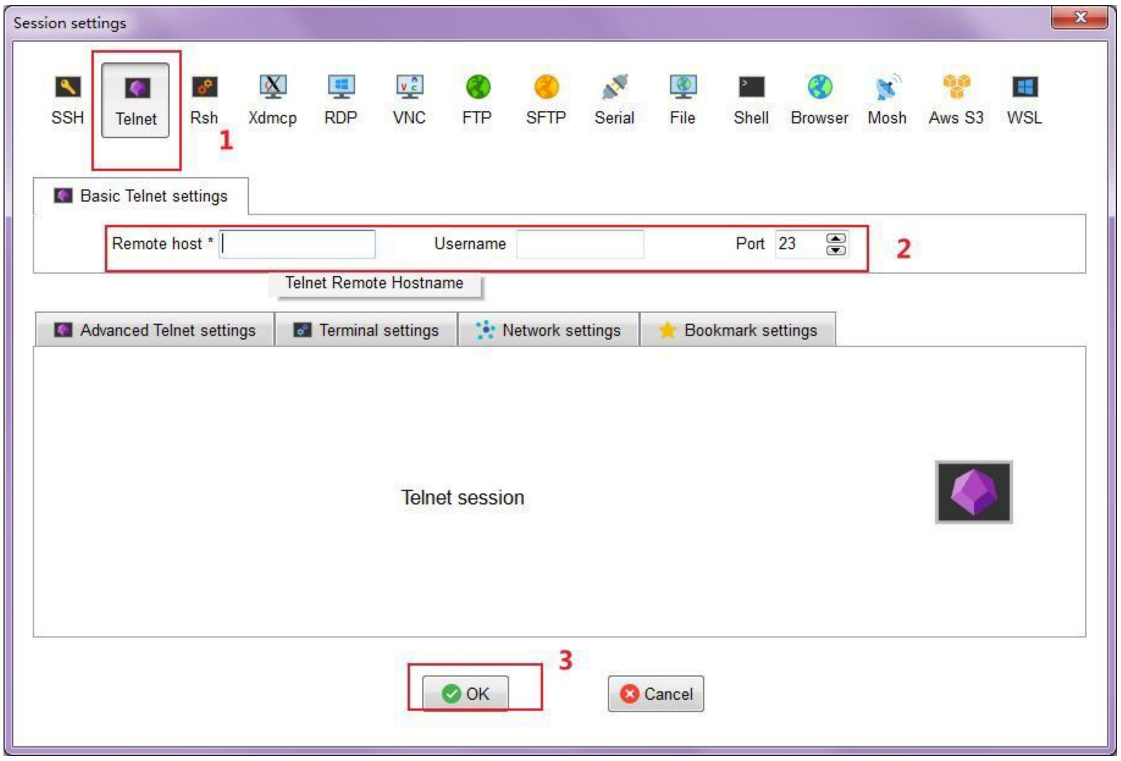
Ensure the computer and the device are in the same network segment (default device IP: 192.168.10.202). Use a cable to connect them if not. Set the computer to a static IP (192.168.10.xxx, xxx not 202). Please refer to 1.3.5 to modify IP, then use 2 network cables to connect the device and computer. The following operations assume that the device and computer are in the same network segment. In this example, the computer IP is 192.168.10.14 and the device IP is 192.168.10.202.

Follow these steps:

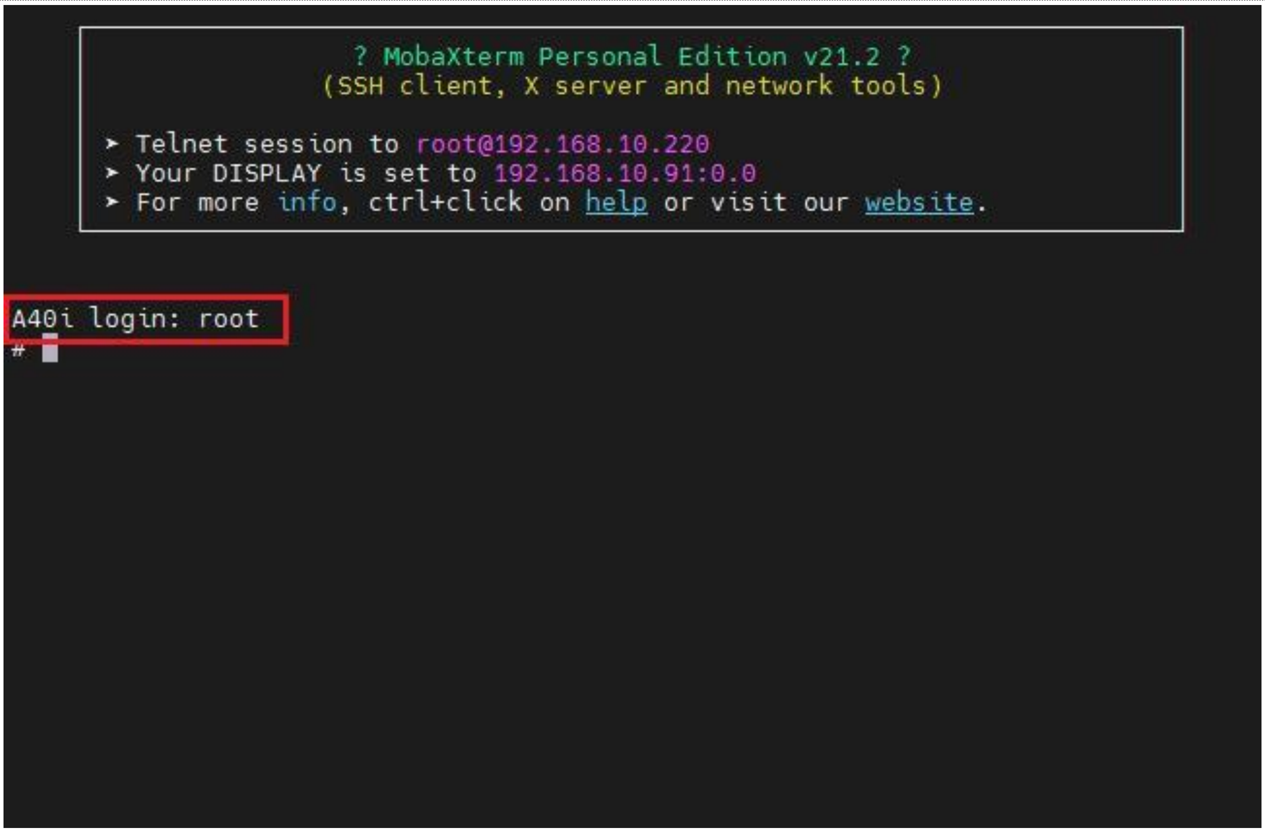
- Plug the cable into the device's network port.

- In sessions, click 'New Session', select 'Telnet', input the screen's IP, and click 'OK'. (Please note that the default IP of the screen is 192.168.10.201 or 192.168.10.202, under the same network is required).



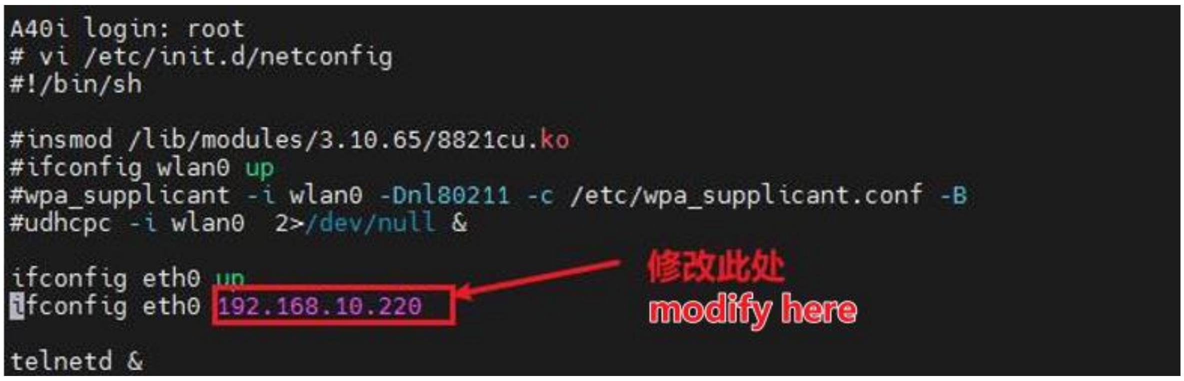
- Power up the screen, the interface shown below, and input 'root' to start operations.



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

➤ Telnet session to root@192.168.10.220
➤ Your DISPLAY is set to 192.168.10.91:0.0
➤ For more info, ctrl+click on help or visit our website.

A40i login: root
#
```

1.5.5 Screen IP Configuration

To modify the IP, type: **vi /etc/init.d/netconfig** (if available) or **vi /etc/init.d/rcS**. Move the cursor to the "ifconfig" line, press "i" to edit, after modified the IP, press "Esc," input :wq (Enter), and save.



```
A40i login: root
# vi /etc/init.d/netconfig
#!/bin/sh

#insmod /lib/modules/3.10.65/8821cu.ko
#ifconfig wlan0 up
#wpa_supplicant -i wlan0 -Dnl80211 -c /etc/wpa_supplicant.conf -B
#udhcpc -i wlan0  2>/dev/null &

ifconfig eth0 up
ifconfig eth0 192.168.10.220

telnetd &
```

1.5.6 Application Upgrade Guide

● Upgrade Package Principles:

In the standard screen's environment, there's an /etc/emcversion file storing the current version number.

Upgrades require the filename to match the version number. To avoid re-upgrade, usually modify the version name of the upgrade package.

Naming convention: version number.tar.

During startup, the standard screen detects a USB drive, searches for an upgrade in the **update** directory, and automatically executes the **install.sh** script.

The **install.sh** script gains control to copy files, modify file attributes, and complete the upgrade.

- Creating an Upgrade Package

In the Ubuntu environment, keep upgrade files in the same directory.

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

Package the directory using the command: tar -cvf DWIN_V1.X.X.tar <INSTALL>

Copy the file (e.g., DWIN_V1.X.X.tar) to the USB drive's /update directory.

- Using an Upgrade Package

The standard screen has a standard program and can be powered on.

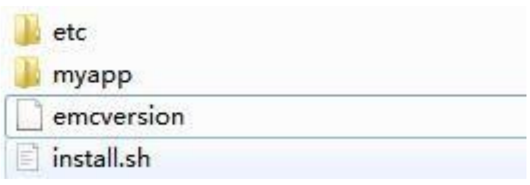
Copy the desired upgrade package to the **update directory** on the USB drive.

Before powering up, insert the USB drive into the standard screen.

After powering up, wait for the standard screen to shut down automatically, indicating a successful upgrade.

- Upgrade Package Example (Modifying Boot Programs)

Folder structure before compression:



emcversion file stores the updated version number for device updates.

myapp folder contains files to be upgraded.

etc folder stores scripts in /etc/init.d/ that may need modification, you can also exclude this folder if there are no files that need to be modified..

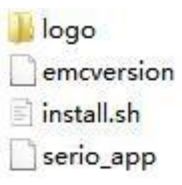
The example of **Install.sh** as below, can be self-modified.

```
#!/bin/sh copy_dir()
```

```
{
  if [ -d $1 ]; then
     for libfile in $1/*; do
         if [ -f $libfile ]; then cp
            $libfile $2/

            chmod $3 $2/${libfile##*/} #ech
           o $2/${libfile##*/}

         fi done
  fi
}
instdir=$(cd `dirname $0`; pwd) # up
date the emcversion

cp $instdir/emcversion /etc/ # co
py application file

cp $instdir/myapp/myapp /usr/local/bin/myapp # mo
dify permission

chmod 755 /usr/local/bin/myapp
# modify runqt script file if needed
cp -a $instdir/etc/init.d/* /etc/init.d/
```

- Upgrade Package Example (Modifying Boot Logo)

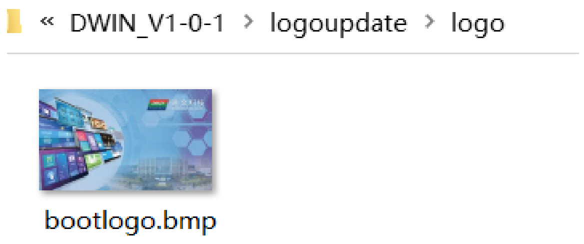
➢ Before compression, the folder structure DWIN_V1-0-1.tar is as follows:



logo
emcversion
install.sh
serio_app

The **emcversion** file stores the updated version number for device updates.

The logo folder contains the logo image for replacement, named bootlogo.bmp. The logo file must be a 24-bit BMP format image,



Example install.sh file:

```
#!/bin/sh


copy_dir()

{

  if [ -d $1 ]; then

     for libfile in $1/*; do

         if [ -f $libfile ]; then cp
            $libfile $2/

            chmod $3 $2/${libfile##*/} #ech
           o $2/${libfile##*/}

         fi done

  fi

}

instdir=$(cd `dirname $0`; pwd) # up
date the emcversion

cp $instdir/emcversion /etc/ # co
py logo file

if [ -f $instdir/logo/bootlogo.bmp ]; then mkdi
   r -p /extp/temp0p2

   mount /dev/mmcblk0p2 /extp/temp0p2

   cp -a $instdir/logo/bootlogo.bmp /extp/temp0p2/ umou
   nt /extp/temp0p2
```

```
    rm -r /extp/temp0p2 sync

  fi sync



  $instdir/serio_app
```

- Compress the "logoupdate" directory into a tar file named " DWIN_V1-0-0.tar " copy it to the USB flash drive's "update" directory. Power off the device, insert the USB flash drive, power on the device again. Upon successful upgrade, you will hear a "beep" sound, the screen will turn off, then remove the USB flash drive, power off, and restart. Verify if the startup logo is correct.
- Important Notes:

- After testing on Windows, updating is successful by simply replacing "bootlogo.bmp" and repackaging. If the upgrade fails, it might be due to lack of execution permissions. Please confirm whether "install.sh" has execution permissions in a Linux environment.

- Command for compressing the tar file in Linux: `tar -cvf DWIN_V1-0-0.tar logoupdate`

- Avoid storing too many files in the USB flash drive (it is recommended to use a dedicated USB flash drive for updating) as it may lead to update failures.

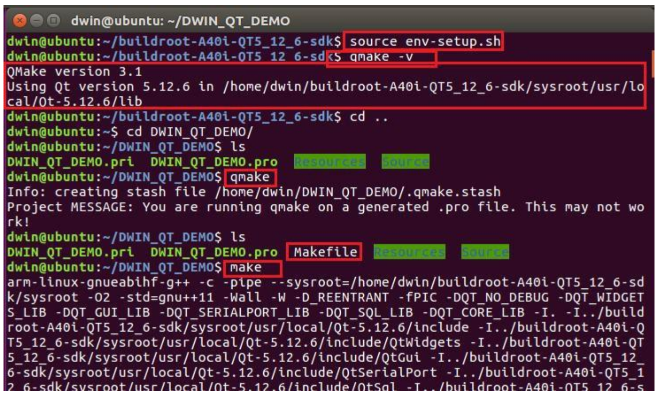
- This upgrade package is available upon request from the sales.

2 Cross-Compilation of QT Project Files

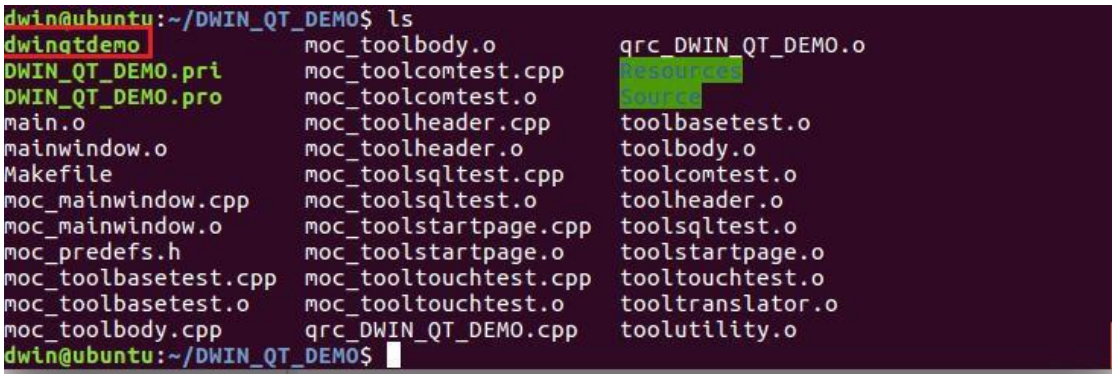
2.1 **qmake**

- After entering the environment configured in the previous section, 1.2 (i.e., after running the command source env-setup.sh), you can verify if the environment is correct by using the command qmake -v. Next, open the project folder that needs cross-compilation (for example, we'll use the provided DWIN_QT_DEMO, place the folder in the Ubuntu/home/dwin), and enter the following command: qmake. If the .pro file has not been generated yet, you may need to run qmake -project first to generate the Makefile.

- Enter the command: **make**. Subsequently, a binary file named as a project will be generated. However, this file cannot be executed in Ubuntu and needs to be downloaded to the screen. For the download process, refer to Section 2.2.



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



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

2.2 USB Drive Download

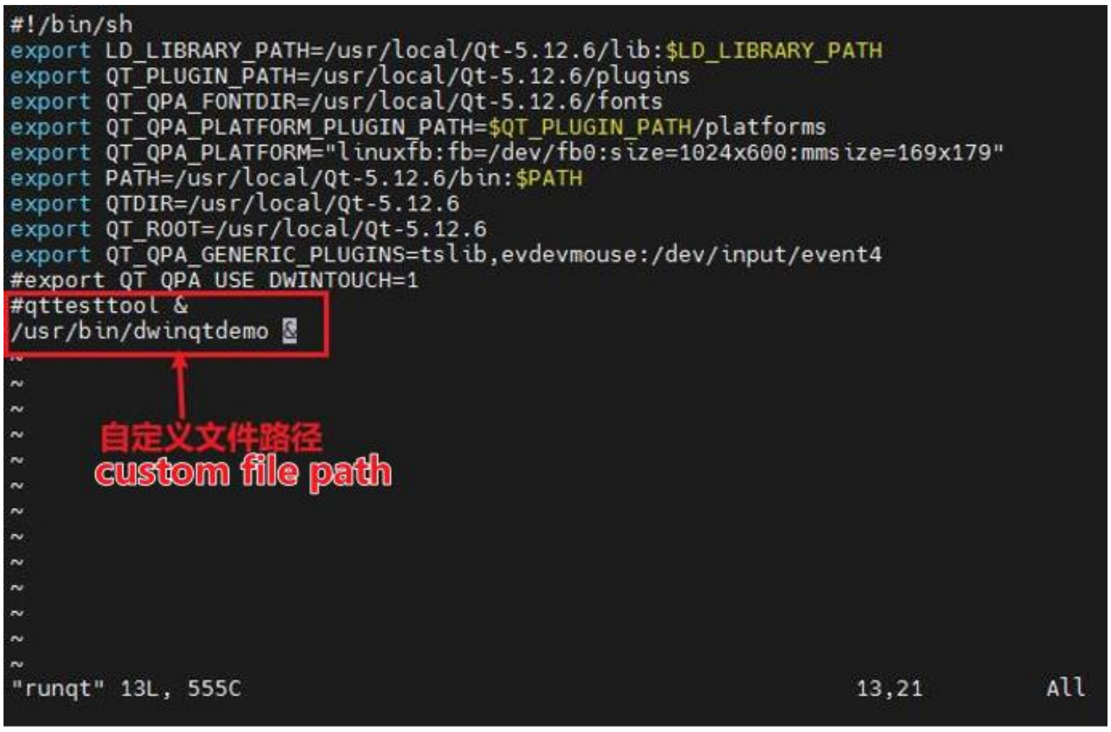
- Place the compiled files in the shared folder. You can copy the files using the command: cp (filename) (shared folder path), for example: **cp dwinqtdemo /mnt/hgfs/share/**.
- Move the target files from the computer's files to the USB drive. You can use various methods to copy the files to the USB drive.
- Insert the USB drive into the screen.
- Open MobaXterm and connect. Enter the command: **cd /mnt/usb** to enter the "usb" folder, select the "sdax" folder, and copy or move the target files to the destination directory (you can choose any custom folder to avoid clutter due to too many files): cp (target file) (folder), for example: **cp dwinqtdemo /usr/bin/**.

2.3 Running dwinqtdemo

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

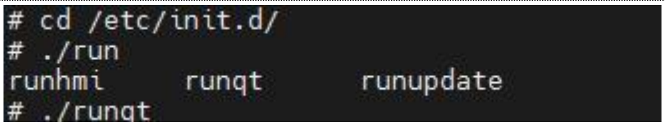
Enter: **vi /etc/init.d/runqt**.

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

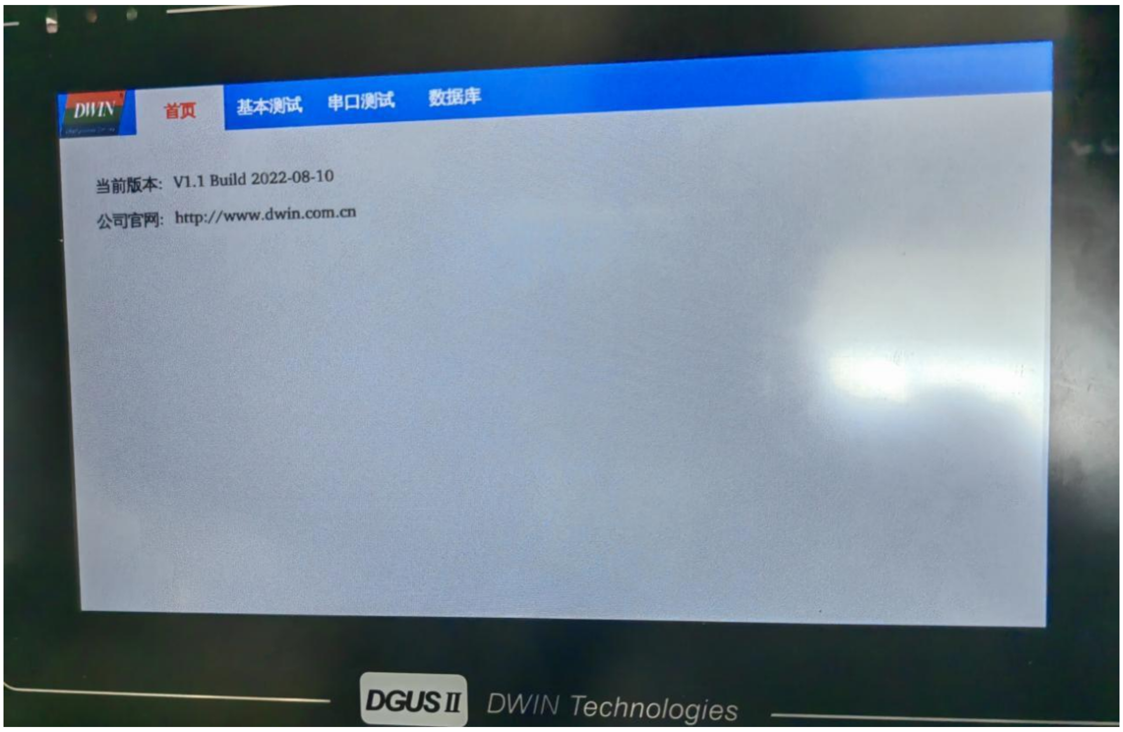


```
#!/bin/sh
export LD_LIBRARY_PATH=/usr/local/Qt-5.12.6/lib:$LD_LIBRARY_PATH
export QT_PLUGIN_PATH=/usr/local/Qt-5.12.6/plugins
export QT_QPA_FONTDIR=/usr/local/Qt-5.12.6/fonts
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_PLUGIN_PATH/platforms
export QT_QPA_PLATFORM="linuxfb:fb=/dev/fb0:size=1024x600:mmsize=169x179"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

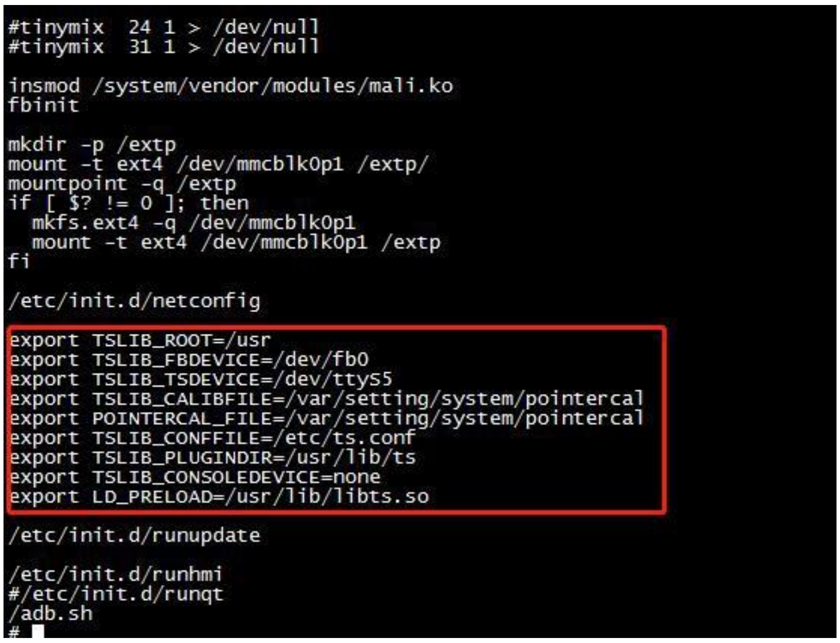
Then the program can be ran by runqt



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

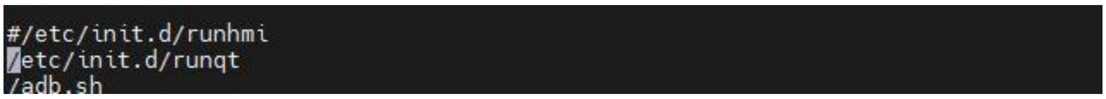

DWIN
首页
基本测试
串口测试
数据库
当前版本: V1.1 Build 2022-08-10
公司官网: http://www.dwin.com.cn
DGUS II DWIN Technologies

Note: At this point, touch functionality may not work due to missing environment variable configurations. It can be restored after a reboot or by referring to the /etc/init.d/rcS file to set the environment variables.



```
#tinymix  24 1 > /dev/null
#tinymix  31 1 > /dev/null

insmod /system/vendor/modules/mali.ko
fbinit

mkdir -p /extp
mount -t ext4 /dev/mmcblk0p1 /extp/
mountpoint -q /extp
if [ $? != 0 ]; then
  mkfs.ext4 -q /dev/mmcblk0p1
  mount -t ext4 /dev/mmcblk0p1 /extp
fi

/etc/init.d/netconfig

export TSLIB_ROOT=/usr
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_TSDEVICE=/dev/ttyS5
export TSLIB_CALIBFILE=/var/setting/system/pointercal
export POINTERCAL_FILE=/var/setting/system/pointercal
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_PLUGINDIR=/usr/lib/ts
export TSLIB_CONSOLEDEVICE=none
export LD_PRELOAD=/usr/lib/libts.so

/etc/init.d/runupdate

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

If the initial configuration does not run the `runqt` program by default, you can modify the `/etc/init.d/rcS` file. If you want to set `runqt` for automatic startup, adjust the last three lines of the `/etc/init.d/rcS` file as shown in the image below:



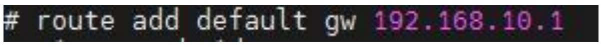
```
#/etc/init.d/runhmi
/etc/init.d/runqt
/adb.sh
```

Save it and re-power by input command reboot.

2.4 Networking Function

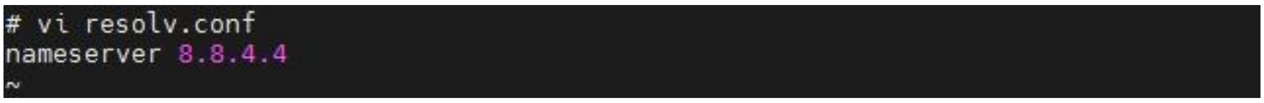
2.4.1 Network Configuration

Here, it is recommended to use a serial connection for device configuration or connect the device and PC using two network cables in the same LAN.

- After connecting the network cable, configure the gateway: **route add default gw** Gateway_IP (in this case, the author uses 192.168.10.1 as the gateway IP). 

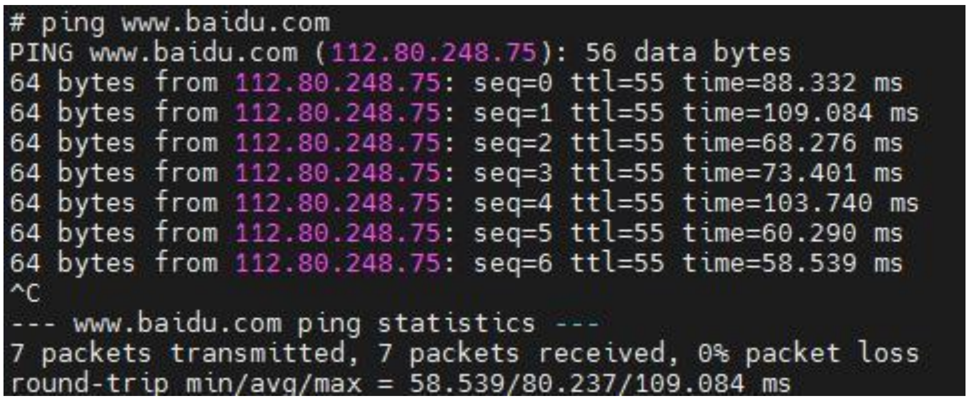
```
# route add default gw 192.168.10.1
```

  After configuration, you can use the **route -n** command to check if the gateway is configured successfully.
- Configure DNS: **vi /etc/resolv.conf**, press **i** to enter the input mode, type nameserver 8.8.4.4, press ESC, then type :**wq** to save and exit.



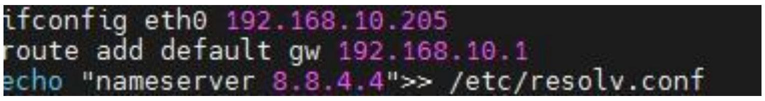
```
# vi resolv.conf
nameserver 8.8.4.4
~
```

- Try pinging an external IP address and check the result.



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

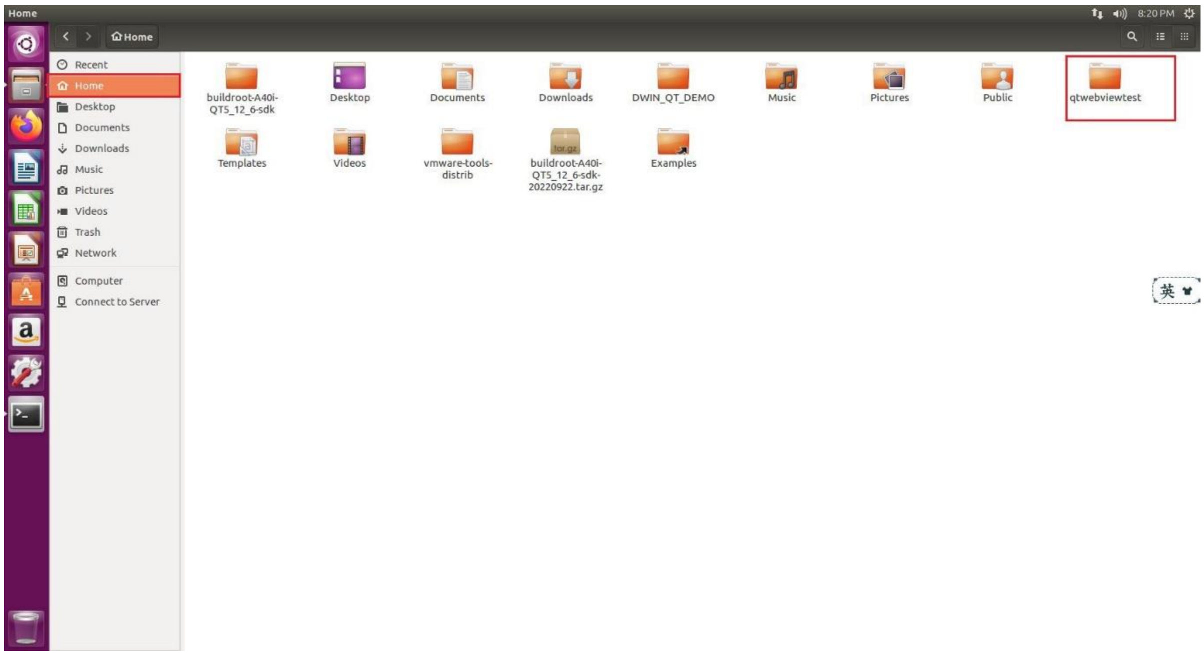
- If need to permanently modify the gateway and DNS, in the /etc/init.d/rcS file or the /etc/init.d/netconfig file (if available), after the ifconfig eth0 IP address, add the statement: route add default gw Gateway_address; echo ‘nameserver 8.8.4.4’ >> /etc/resolv.conf. The modified file looks like the image below.



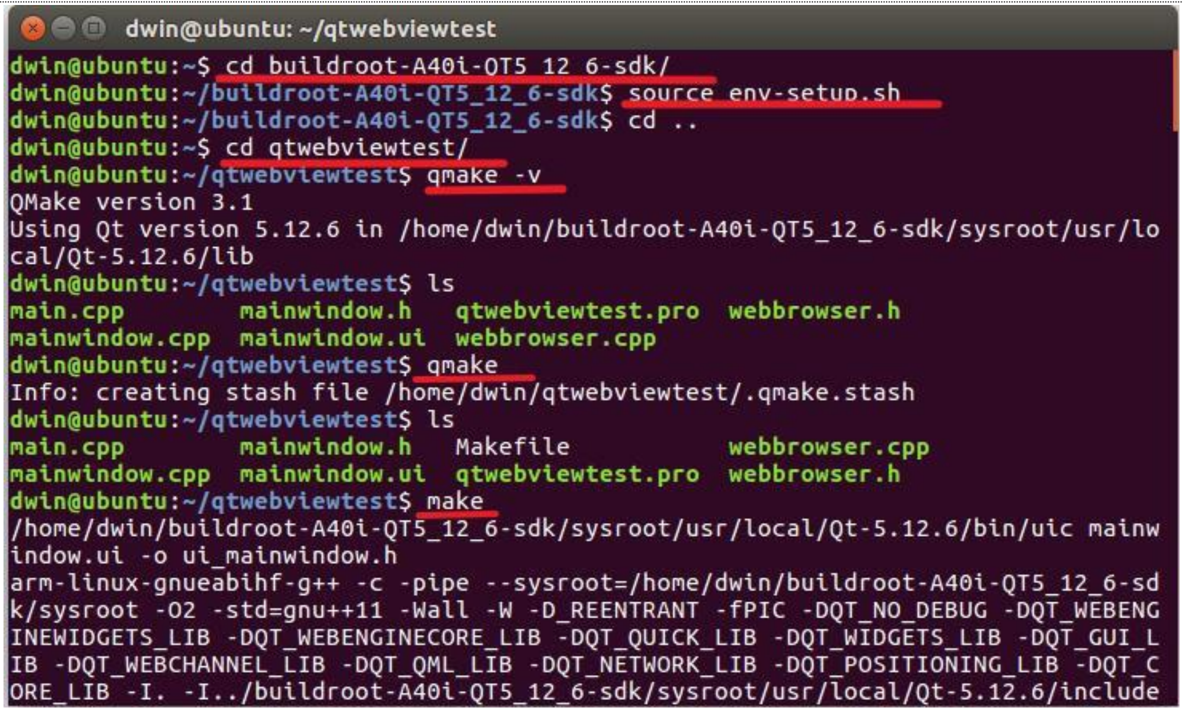
```
ifconfig eth0 192.168.10.205
route add default gw 192.168.10.1
echo "nameserver 8.8.4.4">> /etc/resolv.conf
```

2.4.2 Browsing Webpages

- The 35 series supports QtWebEngine, which can be used to implement a browser. The test project for the browser here is named qtwebviewtest, and the project is placed in the /home/dwin/ folder.

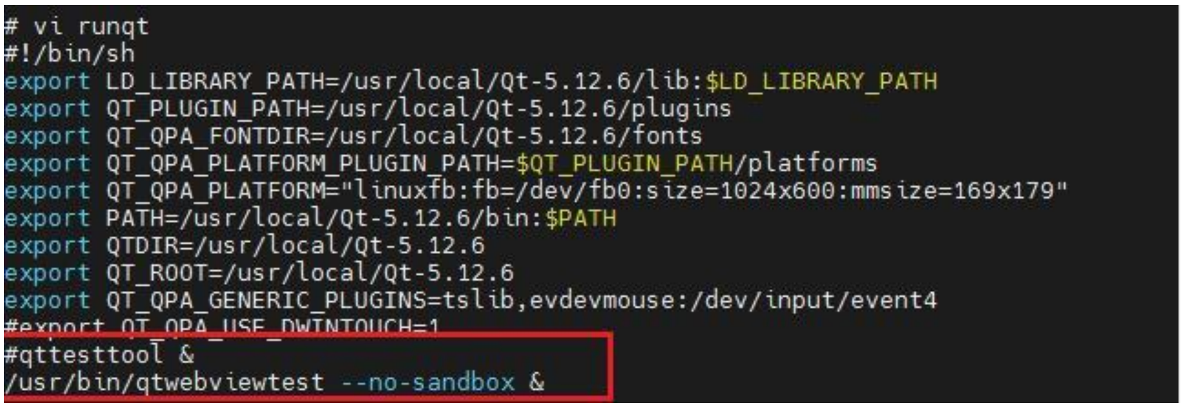


- Follow the steps described in 1.2 to configure the environment. Verify the configuration with the command qmake -v. Then enter the qtwebviewtest folder and compile the project as described in 2.1.



```
dwin@ubuntu: ~/qtwebviewtest
dwin@ubuntu:~$ cd buildroot-A40i-QT5_12_6-sdk/
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ source env-setup.sh
dwin@ubuntu:~/buildroot-A40i-QT5_12_6-sdk$ cd ..
dwin@ubuntu:~$ cd qtwebviewtest/
dwin@ubuntu:~/qtwebviewtest$ qmake -v
QMake version 3.1
Using Qt version 5.12.6 in /home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot/usr/lo
cal/Qt-5.12.6/lib
dwin@ubuntu:~/qtwebviewtest$ ls
main.cpp        mainwindow.h   qtwebviewtest.pro  webbrowser.h
mainwindow.cpp  mainwindow.ui  webbrowser.cpp
dwin@ubuntu:~/qtwebviewtest$ qmake
Info: creating stash file /home/dwin/qtwebviewtest/.qmake.stash
dwin@ubuntu:~/qtwebviewtest$ ls
main.cpp        mainwindow.h   Makefile           webbrowser.cpp
mainwindow.cpp  mainwindow.ui  qtwebviewtest.pro  webbrowser.h
dwin@ubuntu:~/qtwebviewtest$ make
/home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot/usr/local/Qt-5.12.6/bin/uic mainw
indow.ui -o ui_mainwindow.h
arm-linux-gnueabihf-g++ -c -pipe --sysroot=/home/dwin/buildroot-A40i-QT5_12_6-sd
k/sysroot -O2 -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_NO_DEBUG -DQT_WEBENG
INEWIDGETS_LIB -DQT_WEBENGINECORE_LIB -DQT_QUICK_LIB -DQT_WIDGETS_LIB -DQT_GUI_L
IB -DQT_WEBCHANNEL_LIB -DQT_QML_LIB -DQT_NETWORK_LIB -DQT_POSITIONING_LIB -DQT_C
ORE_LIB -I. -I../buildroot-A40i-QT5_12_6-sdk/sysroot/usr/local/Qt-5.12.6/include
```

- After downloading as described in 2.2, enter vi /etc/init.d/runqt to modify the /etc/init.d/runqt file.

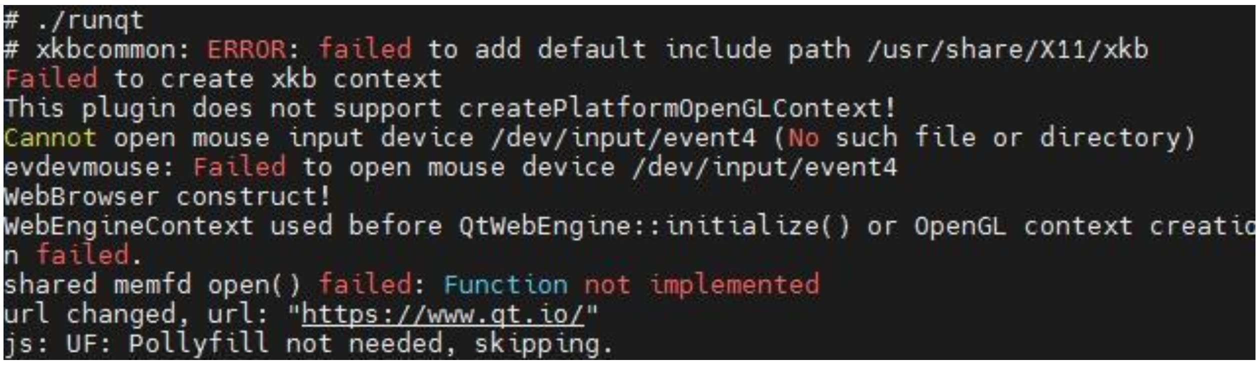


```
# vi runqt
#!/bin/sh
export LD_LIBRARY_PATH=/usr/local/Qt-5.12.6/lib:$LD_LIBRARY_PATH
export QT_PLUGIN_PATH=/usr/local/Qt-5.12.6/plugins
export QT_QPA_FONTDIR=/usr/local/Qt-5.12.6/fonts
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_PLUGIN_PATH/platforms
export QT_QPA_PLATFORM="linuxfb:fb=/dev/fb0:size=1024x600:mmsize=169x179"
export PATH=/usr/local/Qt-5.12.6/bin:$PATH
export QTDIR=/usr/local/Qt-5.12.6
export QT_ROOT=/usr/local/Qt-5.12.6
export QT_QPA_GENERIC_PLUGINS=tslib,evdevmouse:/dev/input/event4
#export QT_QPA_USE_DWINTOUCH=1
#qttesttool &
/usr/bin/qtwebviewtest --no-sandbox &
```

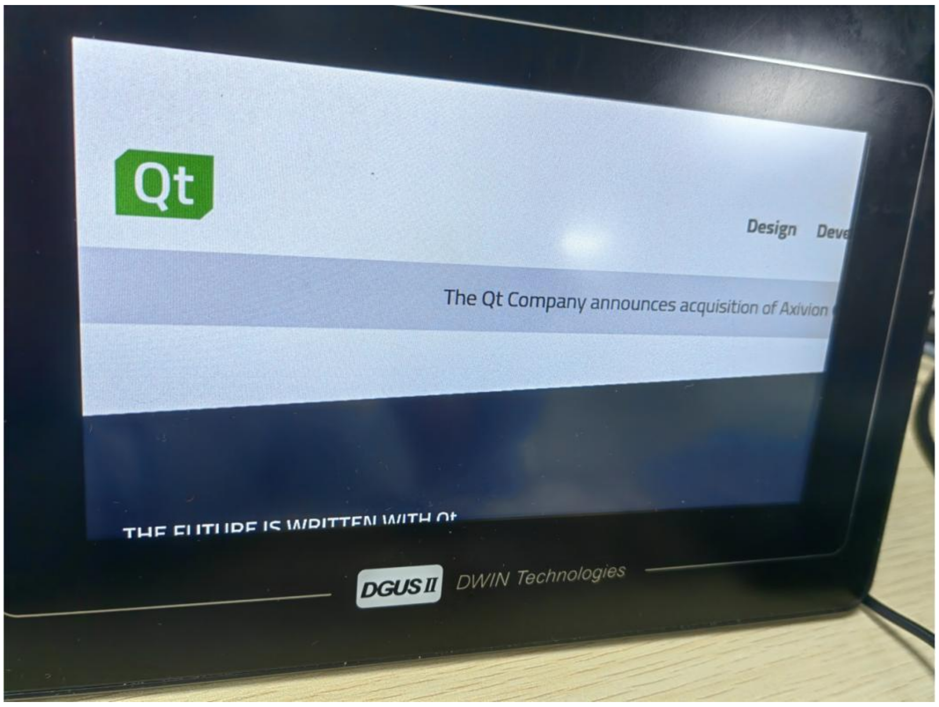
Move the cursor to the beginning of the qttesttool line, press i to enter input mode, type #, then move the cursor to the end of the line, press Enter, move to the next line, type the absolute path of the qtwebviewtest program +space+--no-sandbox+space+&

Press ESC to exit input mode, type :wq to save the file modification.

- Before running the program, configure the network as described in 2.4.1 (gateway settings in 2.4.1 become ineffective after a reboot), and then run the program in the /etc/init.d/ directory.



```
# ./runqt
# xkbcommon: ERROR: failed to add default include path /usr/share/X11/xkb
Failed to create xkb context
This plugin does not support createPlatformOpenGLContext!
Cannot open mouse input device /dev/input/event4 (No such file or directory)
evdevmouse: Failed to open mouse device /dev/input/event4
WebBrowser construct!
WebEngineContext used before QtWebEngine::initialize() or OpenGL context creatio
n failed.
shared memfd open() failed: Function not implemented
url changed, url: "https://www.qt.io/"
js: UF: Pollyfill not needed, skipping.
```



The device will display the default webpage (www.qt.io).

2.5 Time Setting

The time setting can be referenced using the following code. Please note that on DWIN screen s, time and date can only be modified using this method; other methods may become ineffective af ter power off. (Complete code can be obtained through the sales.)

```
void usage()
{
    printf("Usage: settime -s YYYY.MM.DD-HH:MM:SS\n");
```

```
}

int main(int argc, char *argv[])
{
    int year = 0;
    int month = 0;
    int day = 0;
    int hour = 0;
    int minute = 0;
    int second = 0;
    unsigned char crc;
     struct tm t;
     struct timeval vt;
     unsigned char cmd[11]= {0x5A, 0xA5, 0x08, 0x02, 0x16, 0x04, 0x19, 0x0D, 0x18, 0x1C,
0x7E};
    int fd;
    int i;
```