



DWIN Technology Co., Ltd.

Professional, Creditable, Successful

DWIN Linux screens development guide (35 series)



DWIN Linux screens development guide (35 series)	1
1. Environment Configuration	3
1.1 Ubuntu16.04 configuration	3
1.1.1 Introduction	3
1.1.2 Environment Requirements	3
1.1.3 VMware Workstation Installation Steps	3
1.1.4 Download Ubuntu	6
1.1.5 Ubuntu Installation	6
1.1.6 Setting up the shared folders	14
1.2 Installing the a40i Toolchain	17
1.3 Screen Configuration	18
1.3.1 Hardware Introduction	18
1.3.2 Terminal Software	18
1.3.3 Serial communication	18
1.3.4 Telnet Connection via Ethernet	20
1.3.5 Screen IP Configuration	22
1.3.6 Application Upgrade Guide	22
2. Cross-Compilation of QT Project Files	25
2.1 qmake	25
2.2 USB Drive Download	26
2.3 Running dwinqtdemo	27
2.4 Networking Function	29
2.4.1 Network Configuration	29
2.4.2 Browsing Webpages	30
2.5 Time Setting	32



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.

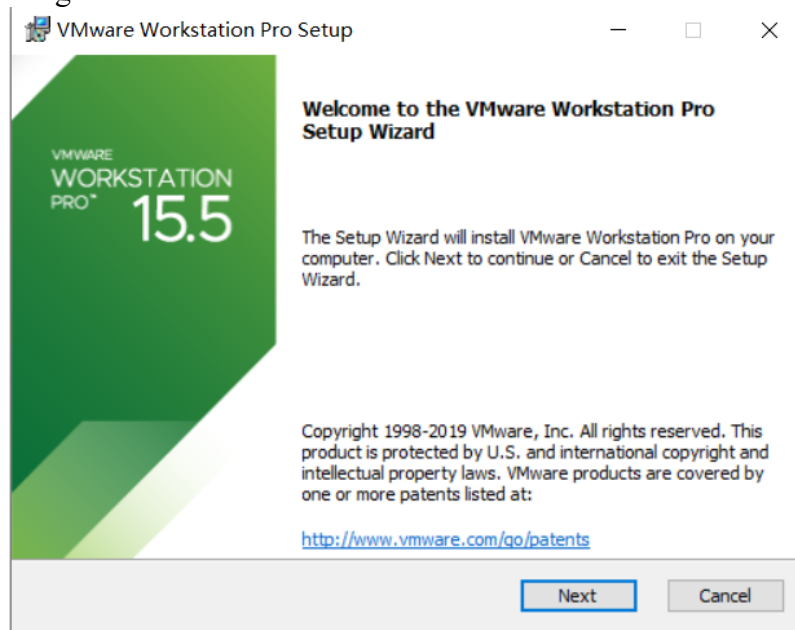
1.1.2 Environment Requirements

- 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.

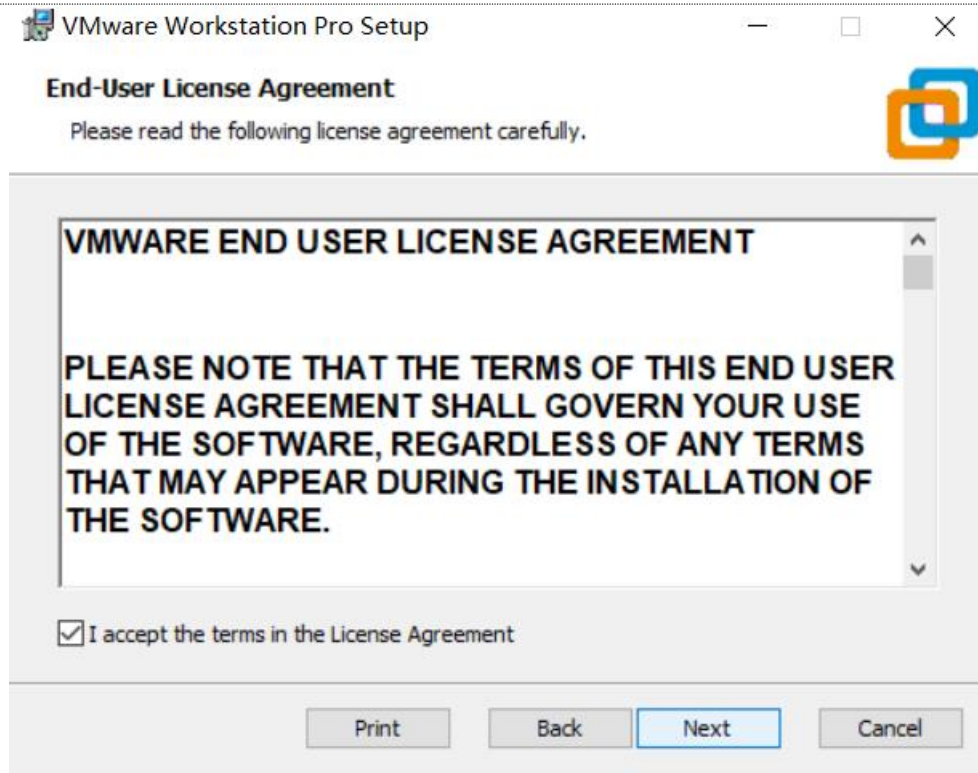
Note: This example demonstrates the installation using VMware Workstation 15 Pro. If you have already installed the virtual machine and Ubuntu, you can proceed directly to Section 1.2 for toolchain installation.

1.1.3 VMware Workstation Installation Steps

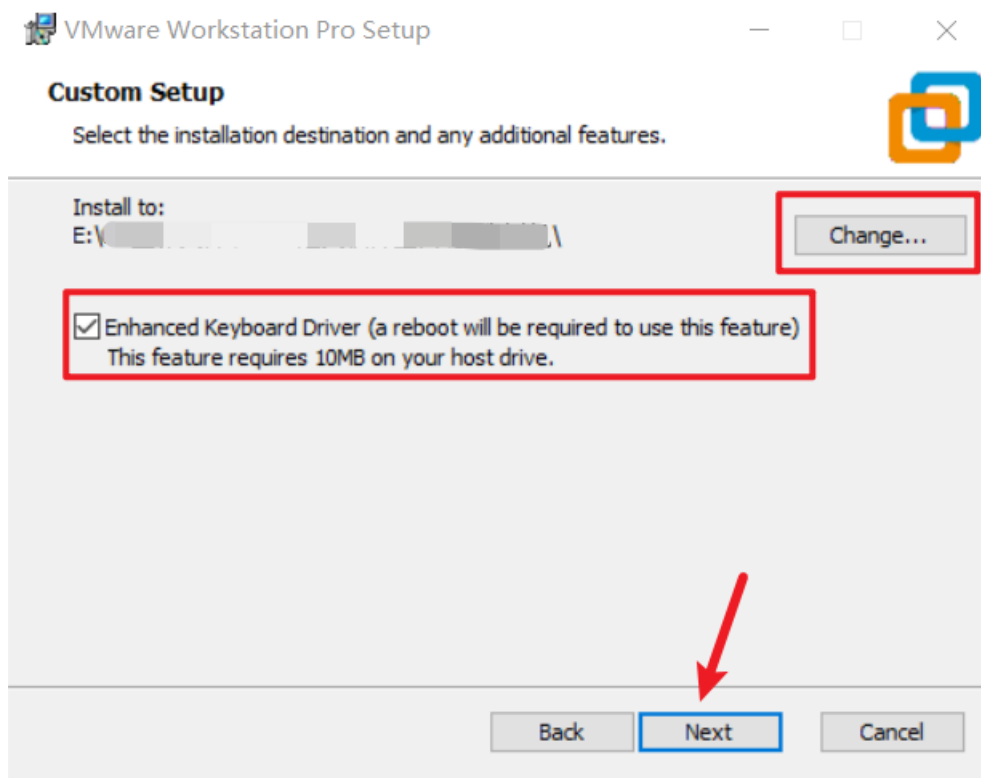
Running installation package



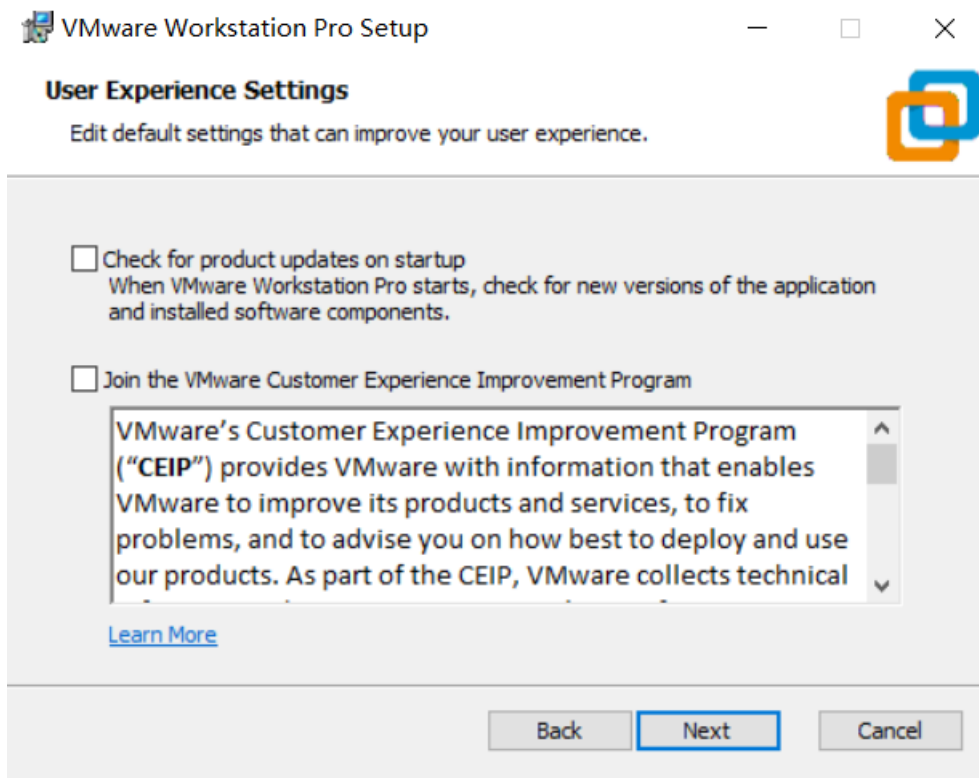
Select 'I accept the terms in the License Agreement' and click 'next'.



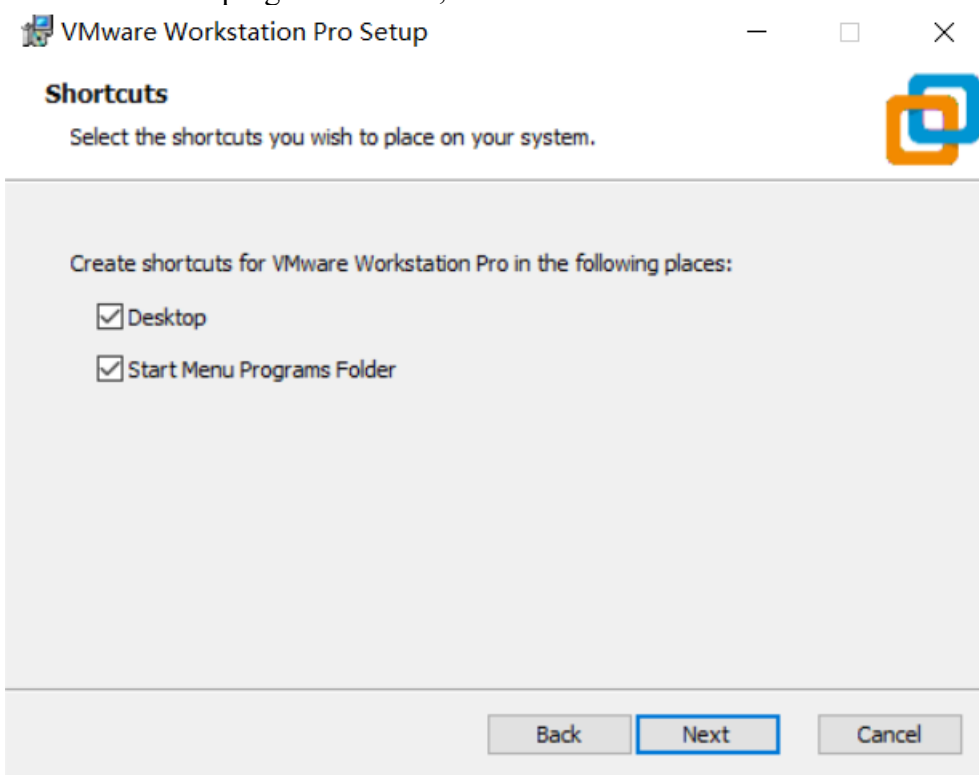
Select the installation path (or choose default path), check the 'Enhanced keyboard driver' option and click 'Next'.



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



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



Click "install". Wait until installation is complete. Click "Finish".

1.1.4 Download Ubuntu

- Get Ubuntu16.04 from official website: <https://releases.ubuntu.com/16.04/>
- Choose 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

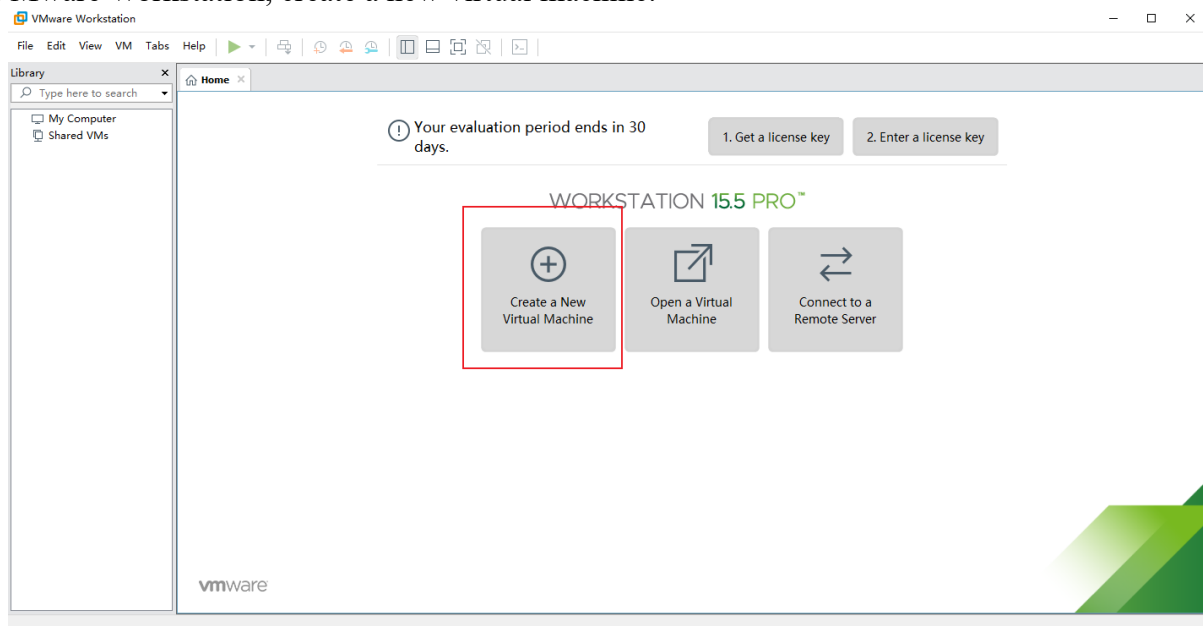
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

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.

1.1.5 Ubuntu Installation

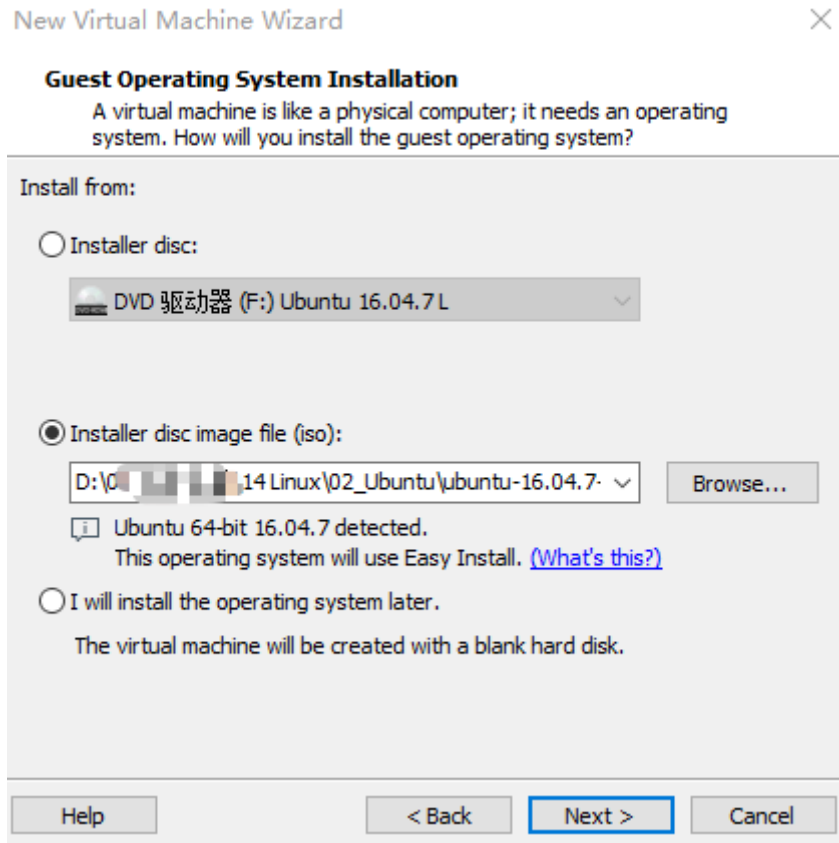
Open VMware Workstation, create a new virtual machine.



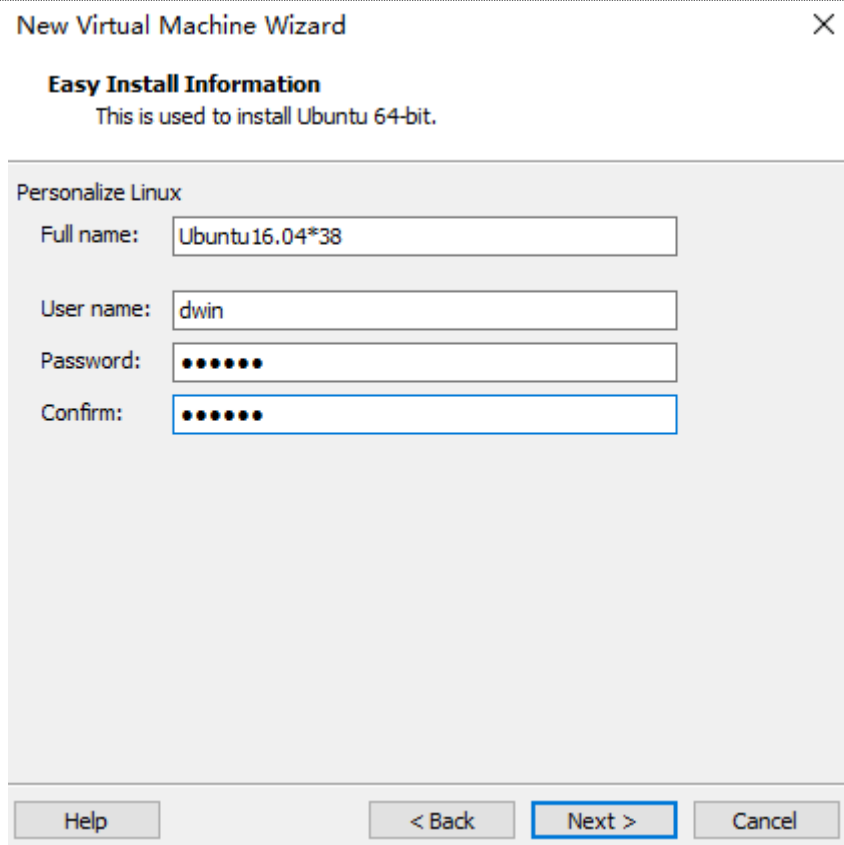
Select 'custom (advanced)' and click 'Next'



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'.



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 [X]

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

Personalize Linux

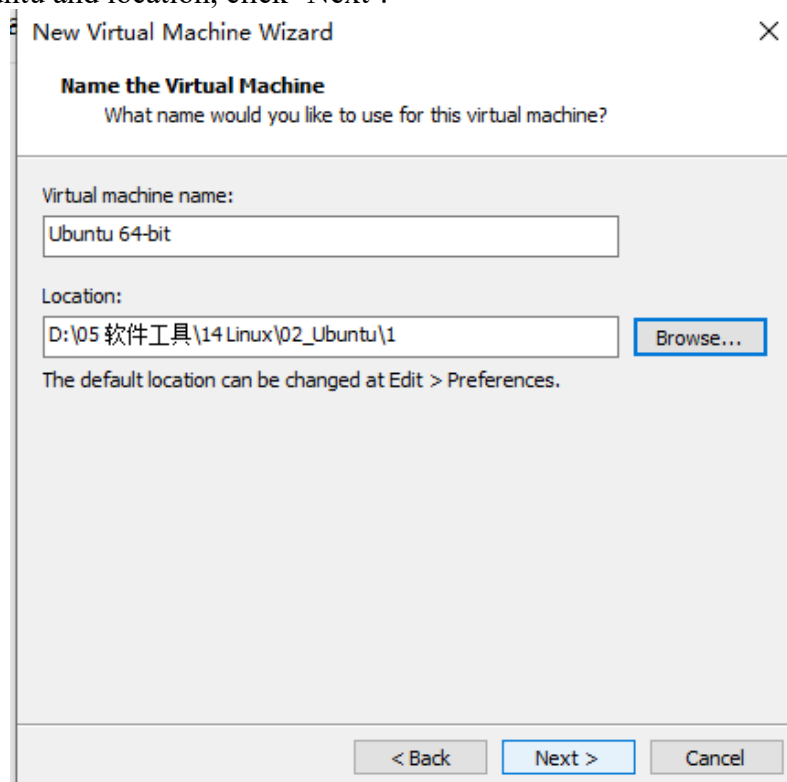
Full name:

User name:

Password:

Confirm:

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



New Virtual Machine Wizard [X]

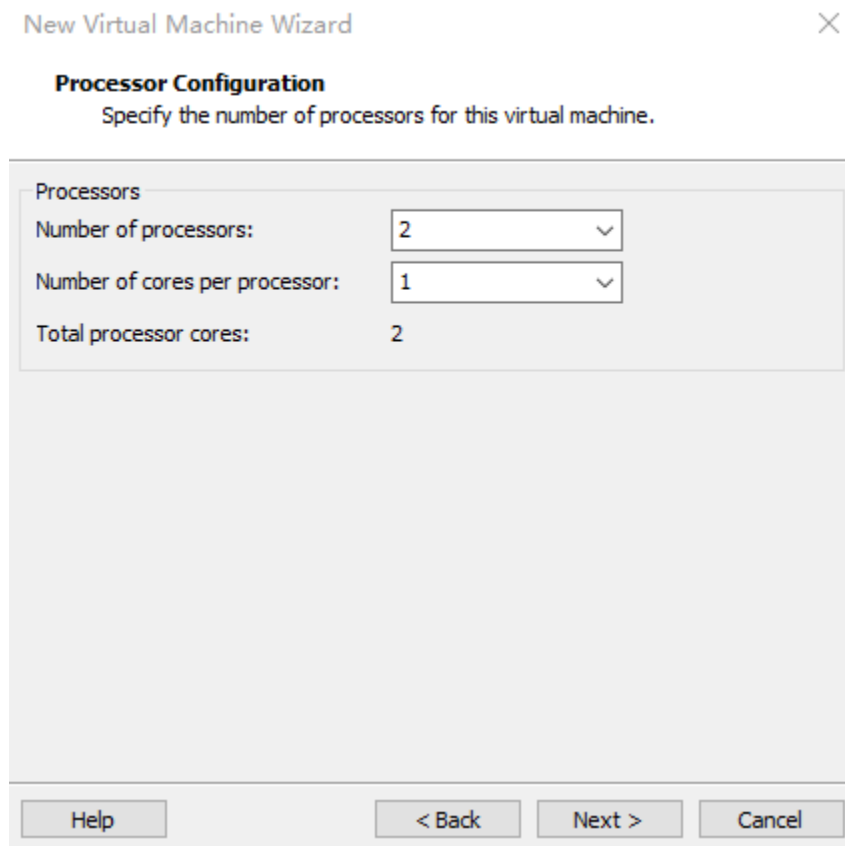
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:

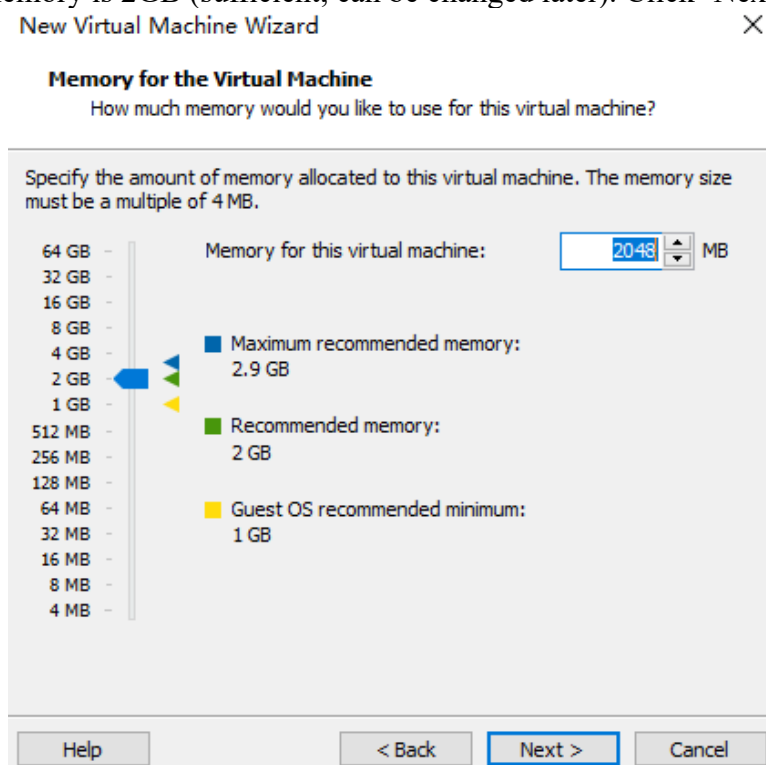
Location:

The default location can be changed at Edit > Preferences.

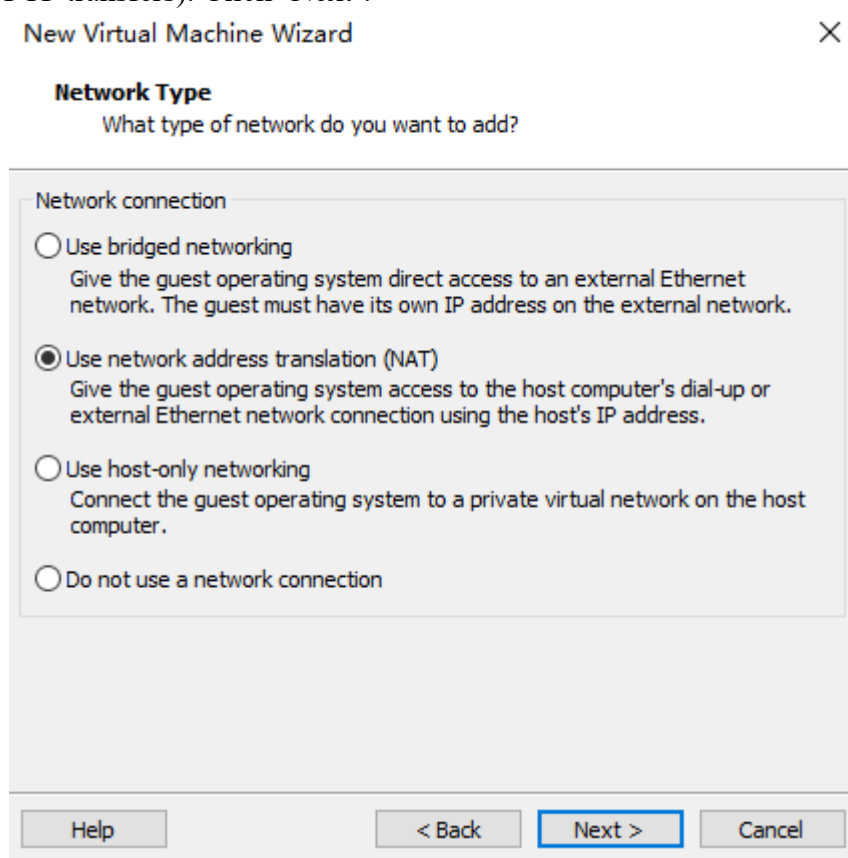
Based on your requirements and computer configuration, allocate the number of processors and cores. Here, the author sets the total number of processor cores to 2. Click 'Next'.



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



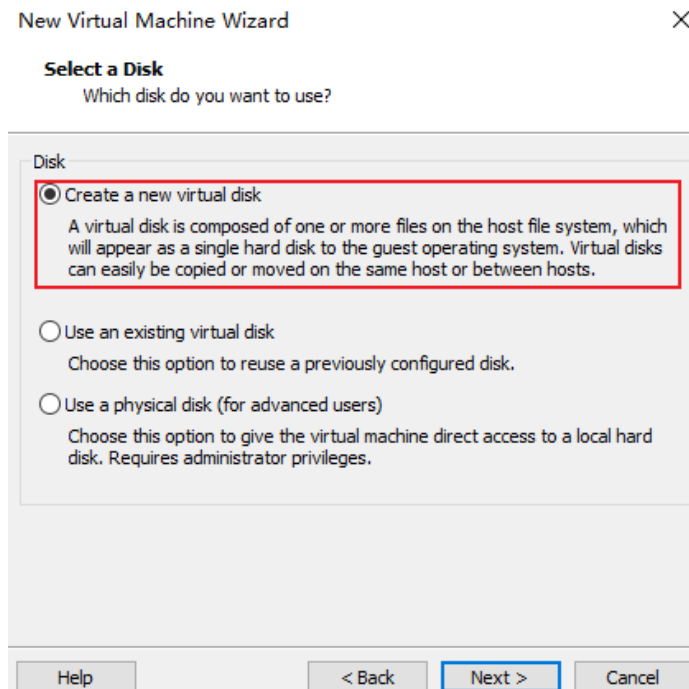
The default configuration is fine (although, in the Network Type, you can choose Bridged Networking, which is useful for TFTP transfers). Click 'Next'.



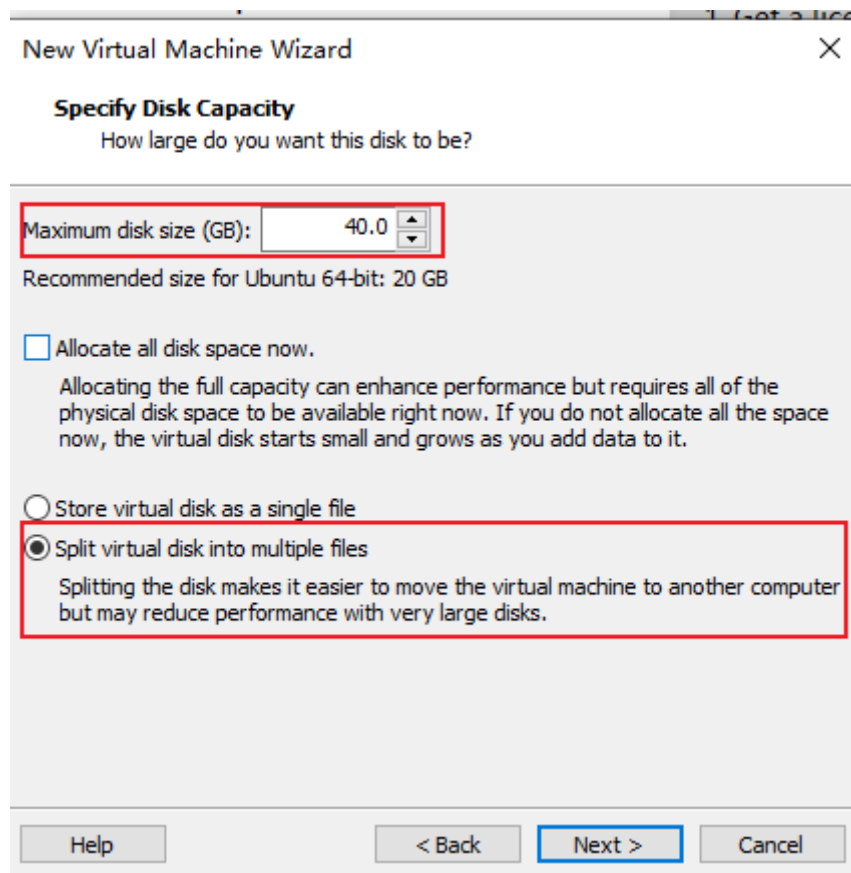
Choose 'Recommend' and 'Next'

Choose 'Recommend' and 'Next'

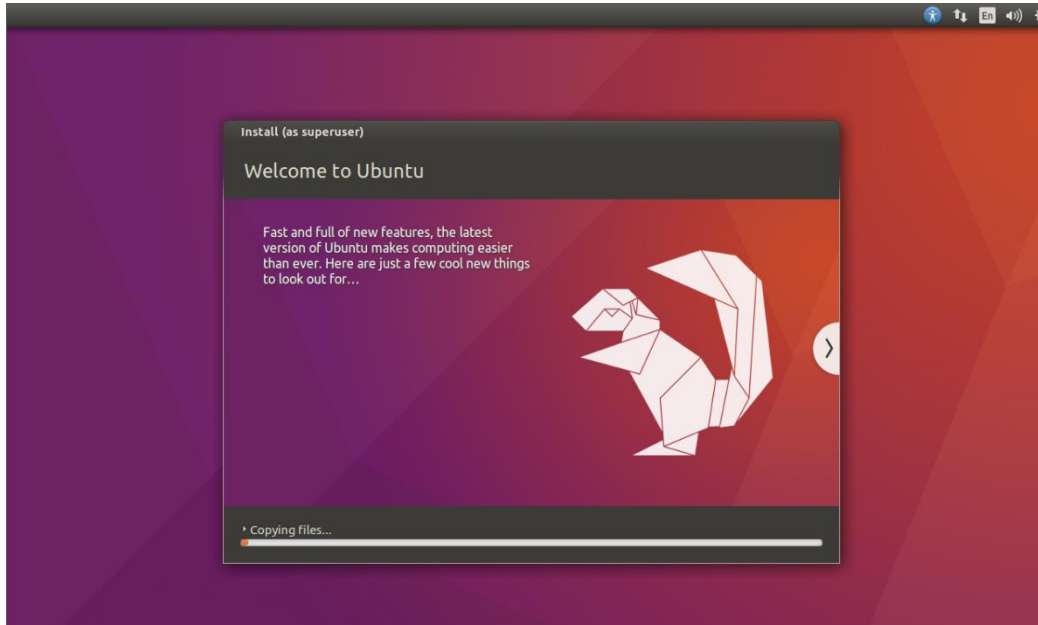
Select 'Create a new virtual disk', and click 'Next'.



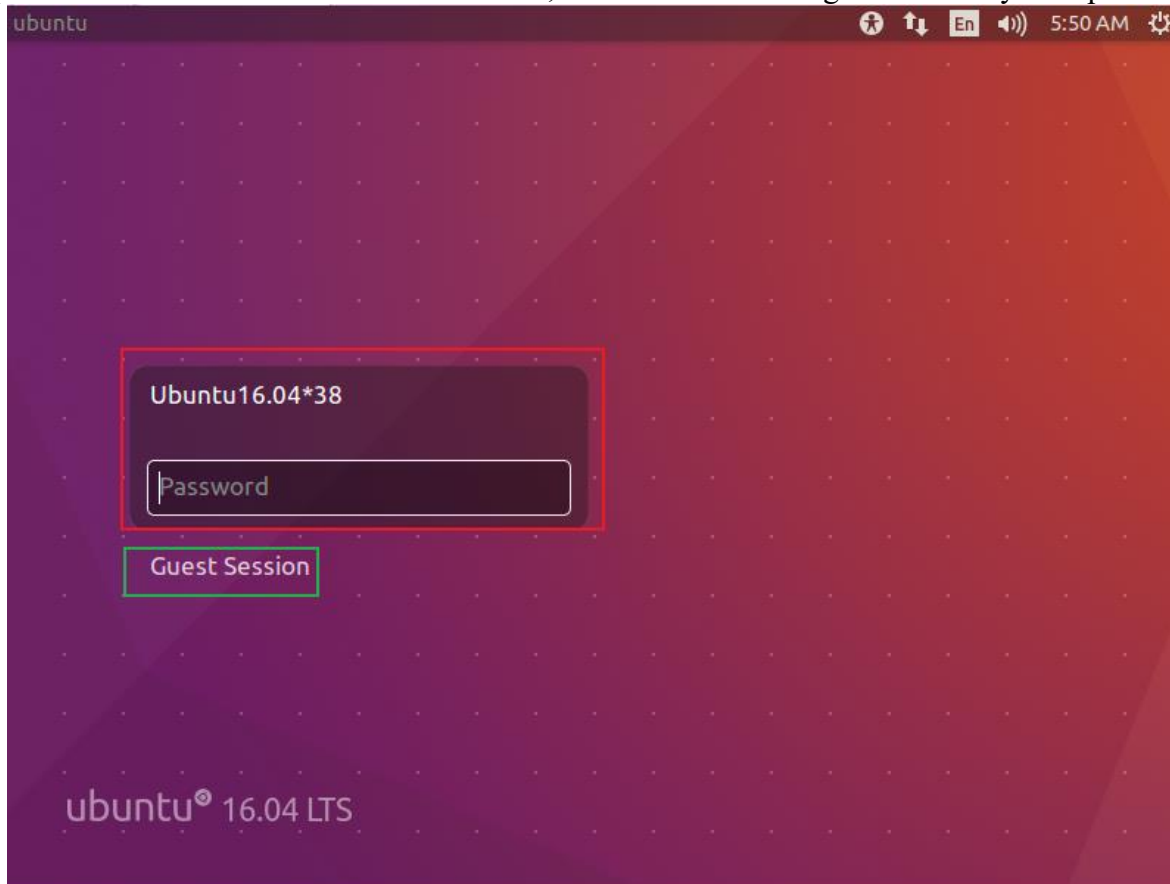
Specify the disk capacity. If your computer has sufficient memory, it's advisable to set it larger, ideally 30GB or more. Click 'Next'. If the disk capacity is insufficient, you can expand it later (as explained in subsequent chapters).



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.



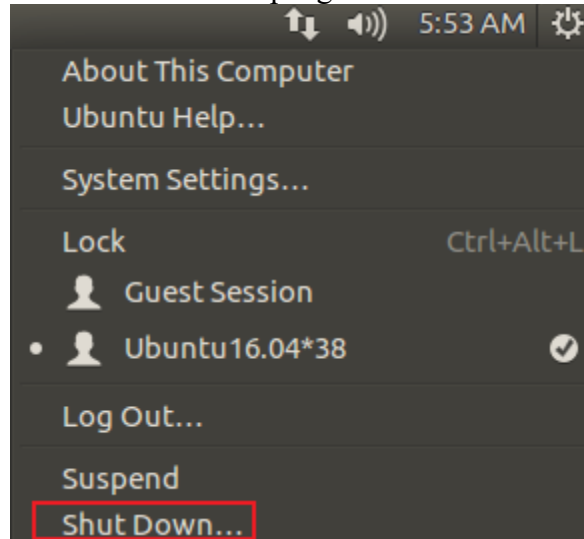
After this interface appears, the installation of Ubuntu is complete. (Note: The login screen has two user login inputs. The one in the red box is user-defined, while the one in the green box is system-provided.)



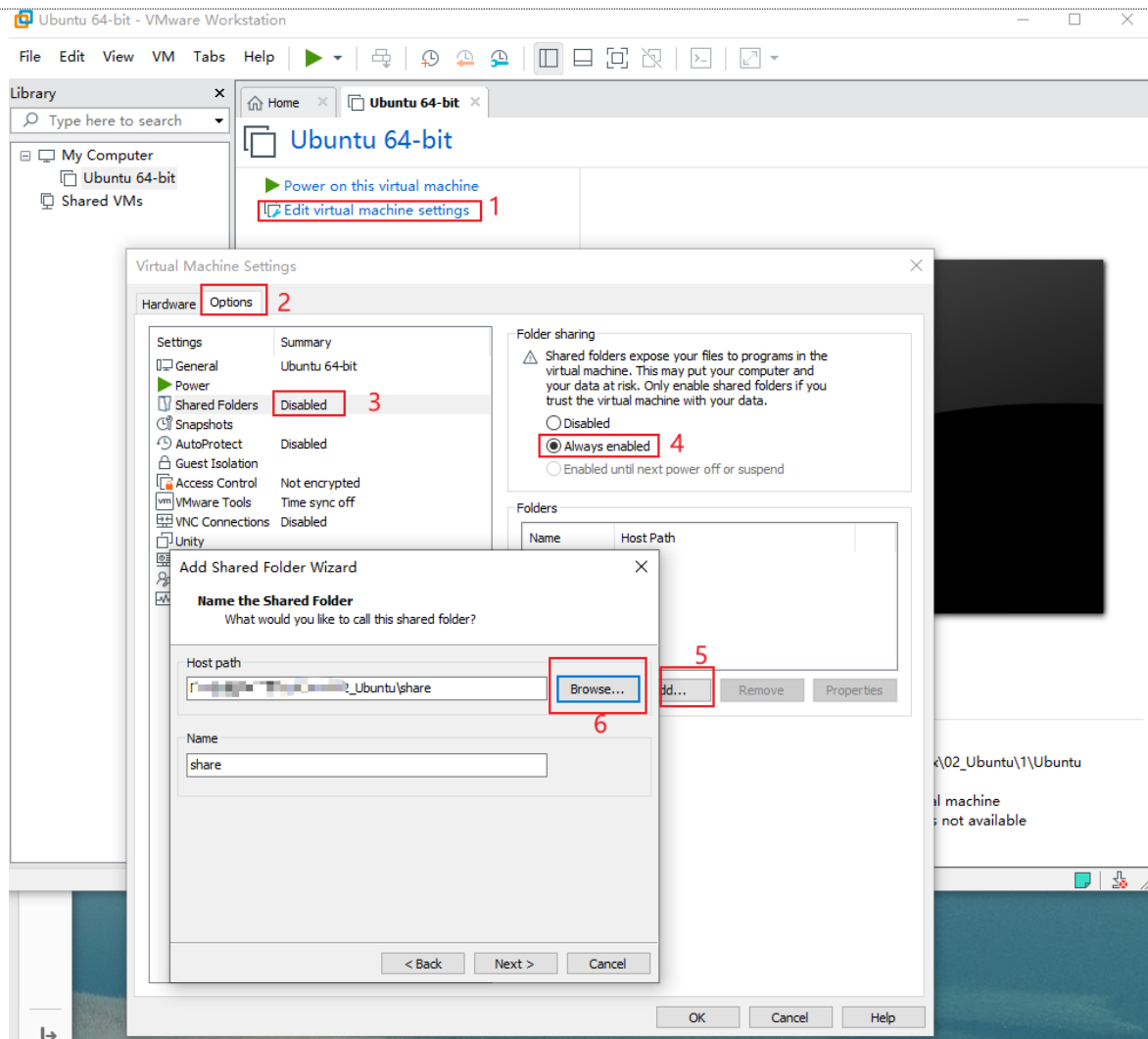
Next start to configure some necessary environments for Ubuntu.

1.1.6 Setting up the shared folders

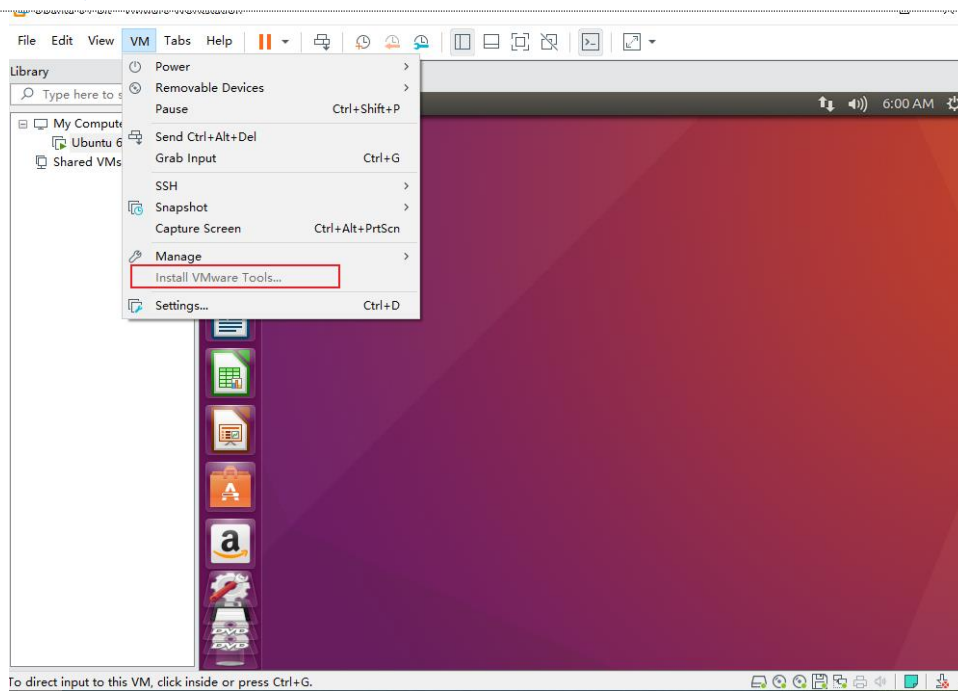
Shut down Ubuntu by clicking 'shut down' in the top-right corner of the desktop.



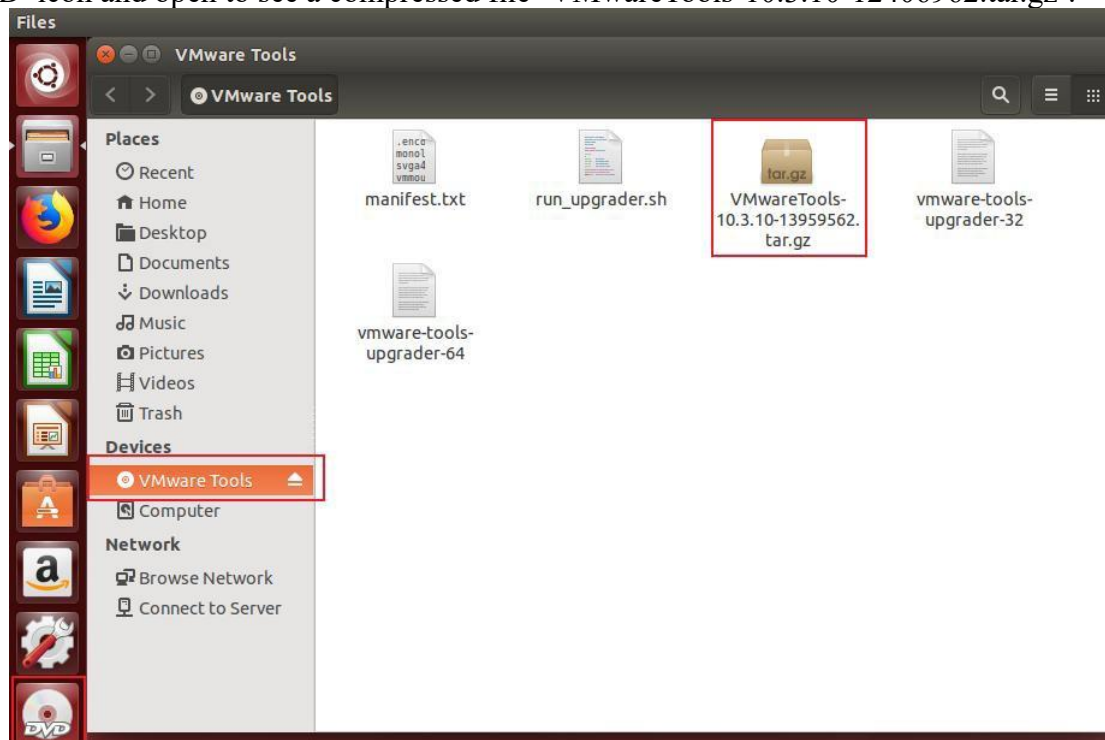
After shutting down, on the corresponding virtual machine page, click 'Edit Virtual Machine Settings' -> 'Options' -> 'Shared Folders' -> 'Always Enabled' -> 'Add'. Add a folder to serve as a medium for transferring files between the host and the virtual machine. Finally, click 'OK'.



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').



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,

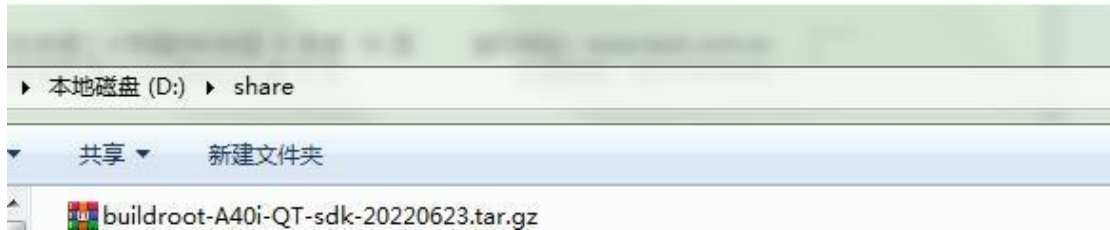
--the VMWare team
```

Now, enter the command: **cd /mnt** (use Tab to navigate to the shared folder you set), the path is /mnt/hgfs/** (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$
```

1.2 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.



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/local/Qt-5.12.6/lib
```

1.3 Screen Configuration

1.3.1 Hardware Introduction

Please refer to the related datasheet

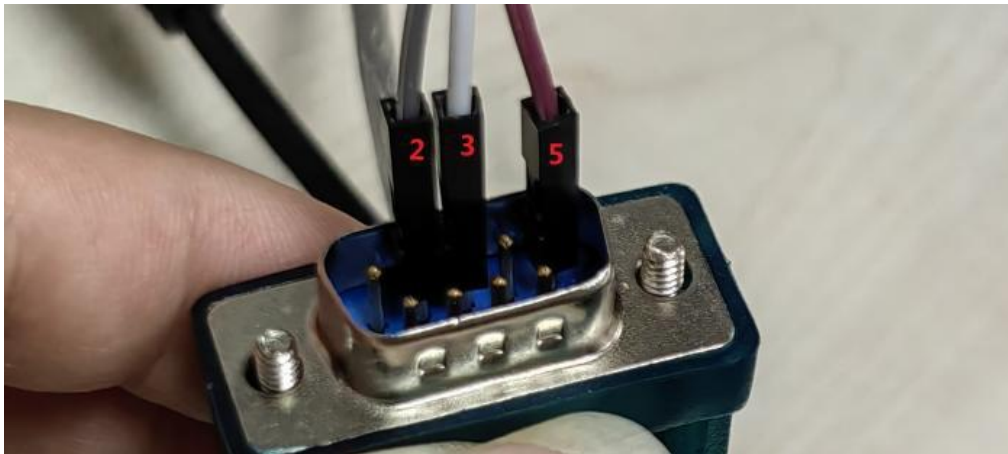
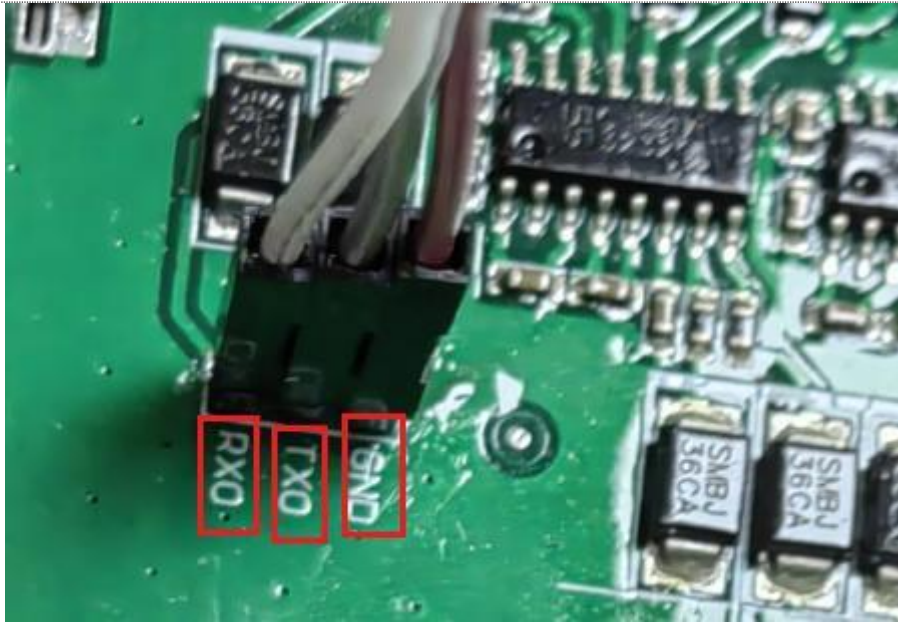
1.3.2 Terminal Software

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

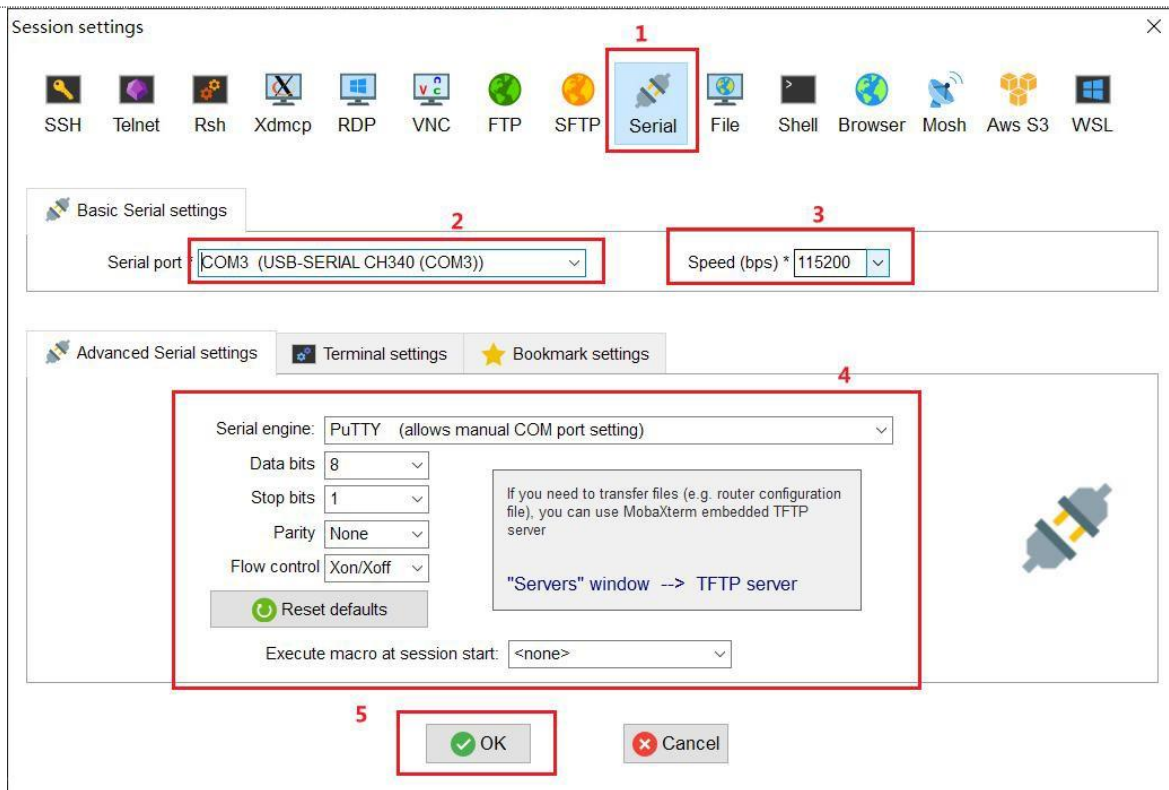
Two types of connection: serial or Telnet.

1.3.3 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.



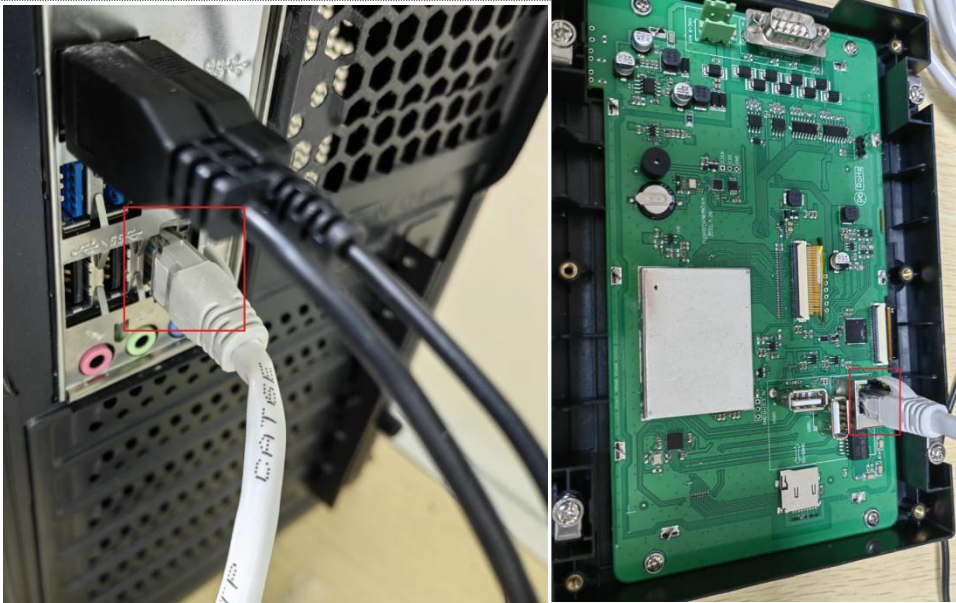
- Powering up the development board and input 'root' to begin operations. Note: if there's a delay between power-up and connection, and the screen shows no text, only a black screen, input 'root'.

1.3.4 Telnet Connection via Ethernet

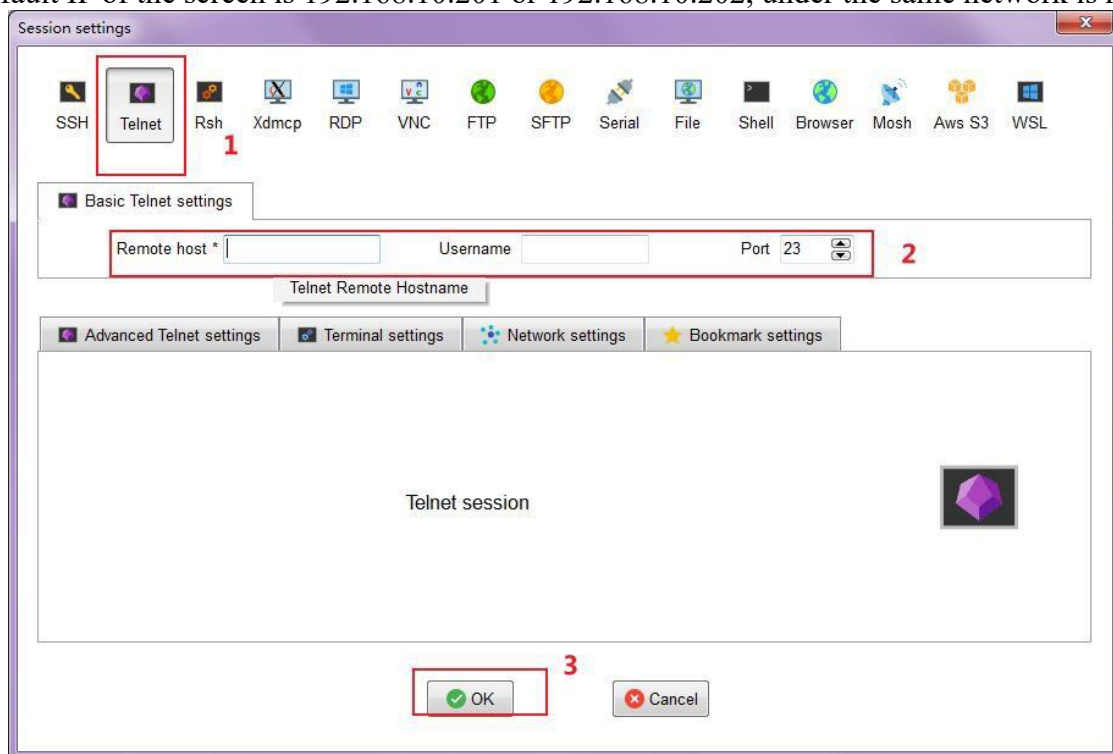
Ensure the computer and the device are in the same network segment (default device IP: 192.168.10.202). Use a cable to connect them if not. Set the computer to a static IP (192.168.10.xxx, xxx not 202). Please refer to 1.3.5 to modify IP, then use 2 network cables to connect the device and computer. 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
#
```

1.3.5 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 &
```

修改此处
modify here

1.3.6 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.

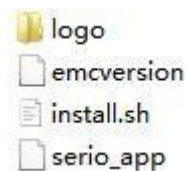
```
#!/bin/sh copy_dir()
{
  if [ -d $1 ]; then
    for libfile in $1/*; do
      if [ -f $libfile ]; then cp
        $libfile $2/
        chmod $3 $2/${libfile##*/} #ech
        o $2/${libfile##*/}
      fi done
    fi
  }
  instdir=$(cd `dirname $0`; pwd) # up
  date the emcversion
```



```
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:



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,

« DWIN_V1-0-1 » logoupdate » logo



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 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-A40t-QT5_12_6-sdk$ source env-setup.sh  
dwin@ubuntu:~/buildroot-A40t-QT5_12_6-sdk$ qmake -v  
QMake version 3.1  
Using Qt version 5.12.6 in /home/dwin/buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/opt/5.12.6/lib  
dwin@ubuntu:~/buildroot-A40t-QT5_12_6-sdk$ cd ..  
dwin@ubuntu:~$ cd DWIN_QT_DEMO/  
dwin@ubuntu:~/DWIN_QT_DEMO$ ls  
DWIN_QT_DEMO.pro  DWIN_QT_DEMO.pro.user  Makefile  README  build  
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 work!  
dwin@ubuntu:~/DWIN_QT_DEMO$ ls  
DWIN_QT_DEMO.pro  DWIN_QT_DEMO.pro  Makefile  README  build  
dwin@ubuntu:~/DWIN_QT_DEMO$ make  
arm-linux-gnueabihf-g++ -c -pipe -sysroot=/home/dwin/buildroot-A40t-QT5_12_6-sdk/sysroot -O2 -std=gnu+hi -Wall -W -D REENTRANT -ffPIC -DQT_NO_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_SERIALPORT_LIB -DQT_SQL_LIB -DQT_CORE_LIB -I. -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtWidgets -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtSerialPort -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtCore -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtGui -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtSql -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtTest -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtNetwork -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtWebSockets -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtXml -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtXmlPatterns -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtSvg -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtPrintSupport -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtDBus -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtOpenGL -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuick -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickControls2 -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickTemplates2 -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickControls2Private -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickTemplates2Private -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickControls2Private -I../buildroot-A40t-QT5_12_6-sdk/sysroot/usr/local/qt-5.12.6/include/QtQuickTemplates2Private
```

```

dwin@ubuntu:~/DWIN_QT_DEMO$ ls
dwinatdemo          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:mmcsize=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
#qtesttool &
/usr/bin/dwingtdemo
```

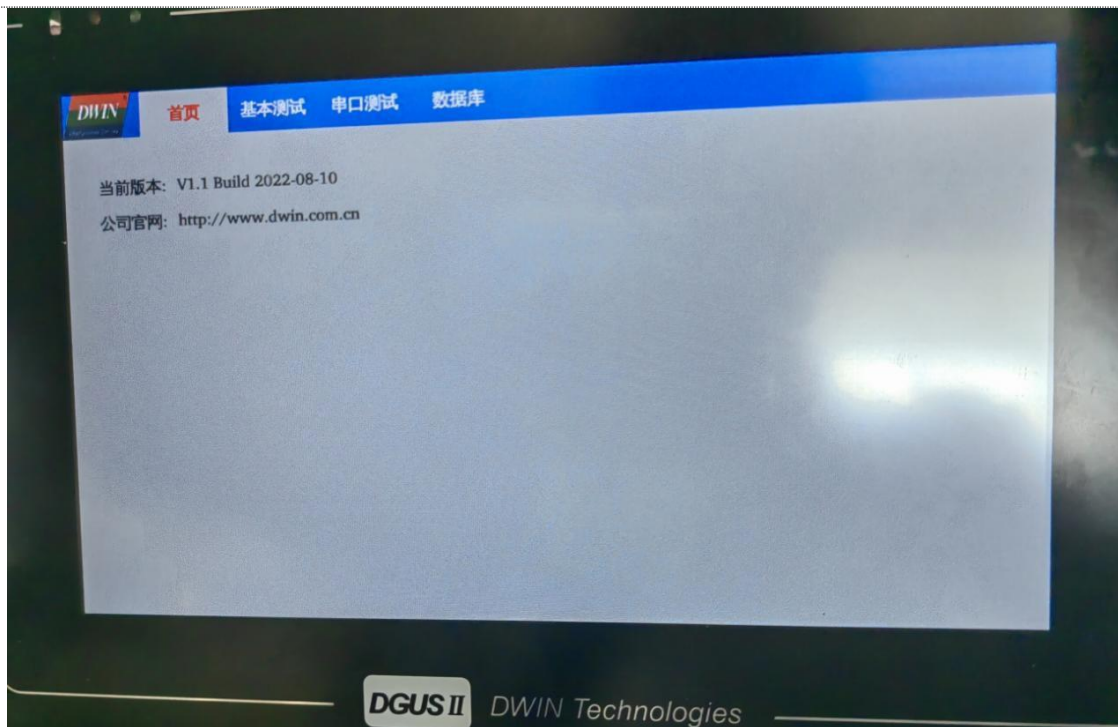
自定义文件路径
custom file path

"runqt" 13L, 555C 13,21 All

Then the program can be ran by runqt

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





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.

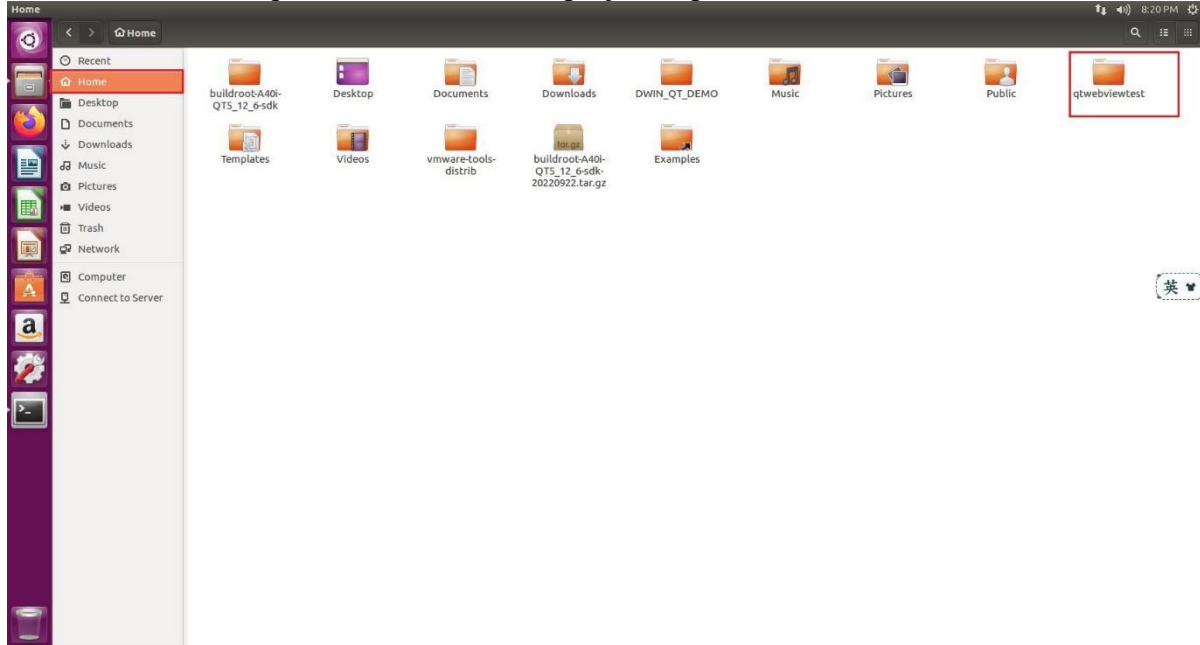
```
# ping www.baidu.com
PING www.baidu.com (112.80.248.75): 56 data bytes
64 bytes from 112.80.248.75: seq=0 ttl=55 time=88.332 ms
64 bytes from 112.80.248.75: seq=1 ttl=55 time=109.084 ms
64 bytes from 112.80.248.75: seq=2 ttl=55 time=68.276 ms
64 bytes from 112.80.248.75: seq=3 ttl=55 time=73.401 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/local/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 mainwindow.ui -o ui_mainwindow.h
arm-linux-gnueabi-g++ -c -pipe --sysroot=/home/dwin/buildroot-A40i-QT5_12_6-sdk/sysroot -O2 -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_NO_DEBUG -DQT_WEBENGINEWIDGETS_LIB -DQT_WEBENGINECORE_LIB -DQT_QUICK_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_WEBCHANNEL_LIB -DQT_QML_LIB -DQT_NETWORK_LIB -DQT_POSITIONING_LIB -DQT_CORE_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.

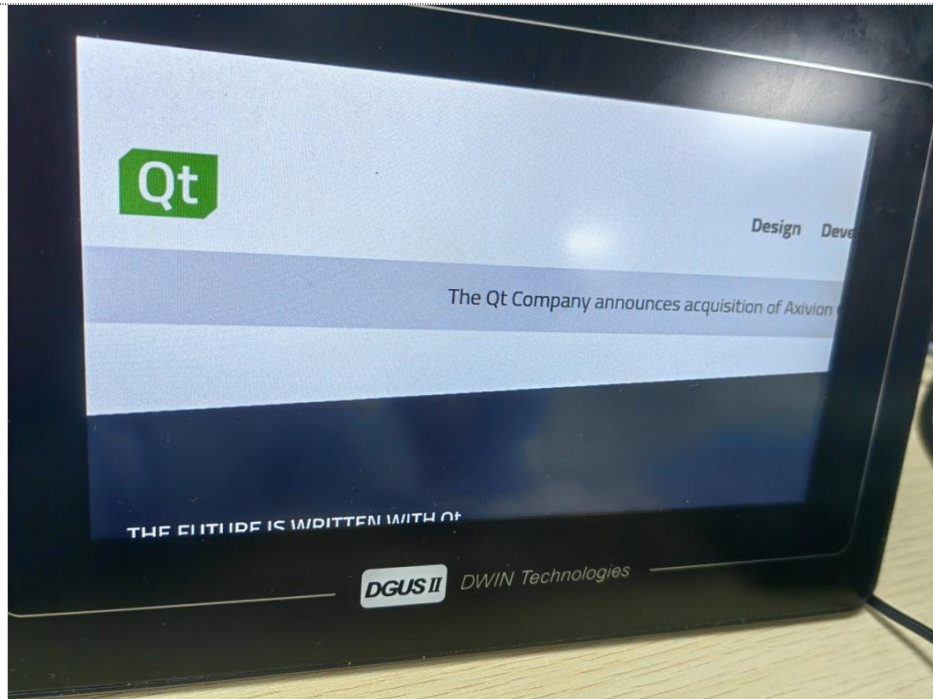
```
# vi runqt
#!/bin/sh
export LD_LIBRARY_PATH=/usr/local/Qt-5.12.6/lib:$LD_LIBRARY_PATH
export QT_PLUGIN_PATH=/usr/local/Qt-5.12.6/plugins
export QT_QPA_FONTDIR=/usr/local/Qt-5.12.6/fonts
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_PLUGIN_PATH/platforms
export QT_QPA_PLATFORM="linuxfb:fb=/dev/fb0:size=1024x600:mmc size=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
#qtesttool &
/usr/bin/qtwebviewtest --no-sandbox &
```

Move the cursor to the beginning of the qtesttool 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 screens, time and date can only be modified using this method; other methods may become ineffective after 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;
```




DWIN Technology Co., Ltd.

Professional, Creditable, Successful

```
unsigned char cmd[11]= {0x5A, 0xA5, 0x08, 0x02, 0x16, 0x04, 0x19, 0x0D, 0x18, 0x1C,  
0x7E};  
int fd;  
int i;
```

