TPC-71W Ubuntu 16.04

User Manual

V1.0

Contents

1. Overview of ARM Ubuntu 16.04

1.1. Cross-platform, consistent use experience on X86 and ARM platforms

Ubuntu is one of the most popular Linux distributions. At present, many industrial applications are based on the Ubuntu platform. Running Ubuntu on ARM is easier, more convenient, faster, and more friendly for industrial application development and maintenance. For existing X86 platform applications, it can be ported to the ARM Ubuntu platform more quickly and seamlessly, while reducing development difficulty, cost and risk.

1.2. Rich Software Support

Ubuntu officially provides a very rich ARM package. For the ssh, telnet, ftp, qt library, video playback software, mysql database software, Java, VNC, etc., which are commonly used in industrial control, off-the-shelf software packages are provided. When the customer needs these components, there is no need to cross-compile or port. Just like in x86, you can directly install it online through the apt command.

1.3. Rapid Customization System

The Ubuntu ARM version provides a very rich software package, so customers can quickly install the required software packages according to their own project requirements, quickly configure the relevant systems, and customize the required system. Moreover, developers who already have X86 Ubuntu experience don't have to spend time on ARM Linux development because the development techniques and methods are exactly the same.

1.4. Rapid Development and Maintenance

In addition to providing rich software support, Ubuntu is also very mature in development and has many resources. It supports multiple development environments such as gcc, qt, java, python, mono, php, etc., and provides related software development tools. Customers can choose familiar and appropriate development languages, environments and solutions as needed to accelerate software development.

1.5. Graphical System Settings

The Ubuntu ARM version supports the XFCE lightweight desktop environment and runs smoothly on ARM. Many system configurations, such as network configuration and 3G/4G configuration, can be configured graphically, which is very convenient, fast and simple.

1.6. Long-term Support Version

Ubuntu is developed and maintained by the commercial enterprise Canonical. Its stability and reliability are trustworthy. At the same time, Ubuntu provides a long-term support version - Ubuntu 16.04 LTS will provide 3 to 5 years of support and updates to meet the long-term support needs of industrial customers.

1.7. Advantech Provides Customized Industrial ARM Ubuntu Version

The ARM Ubuntu version provided by Advantech is based on ARM Ubuntu 16.04 and is adapted for TPC-71W. To meet the common needs of industrial customers, the following aspects are customized:

1) Provide a variety of hardware test procedures and test instructions to facilitate customer testing and verification of hardware, as well as learning how to use;
2) Support graphics hardware acceleration of TPC-71W platform to ensure smooth video playback;
3) Provide sample programs and source code, such as serial communication, video playback, for customer reference during development;
4) Support wireless modules such as peripheral WIFI/4G, and provide built-in drivers to facilitate customers to establish wireless solutions;
5) Provide common system configuration instructions for industrial customers, such as support for Chinese, dual display settings, applications that start automatically at boot time.

Note: You need to pay to use the Ubuntu system for commercial purposes. You first need to get the official Ubuntu license. Please contact Ubuntu official for further information. You can also contact Advantech PM, since Advantech has established communication and cooperation channels with Ubuntu.

2. Advantech Ubuntu Image Software Support

2.1. Built-in software

By default, image supports various software commonly used by industrial users.

- Chromium browser
- Video playback software
- Audio player software
- Qtcreator development environment

- ......

2.2. Install the package online with the apt command

If the required software is not included in the Ubuntu Image provided by Advantech, the ARM Ubuntu system also provides the APT (Advanced Package Tool) package management mechanism. In the case where the device is already connected to the network, the software can be queried and installed online through APT related commands. APT automatically handles dependencies and installs the required packages on the system.

Ubuntu offers a very rich ARM package. Most of the software required by customers can be installed directly through the apt command without cross-compilation from the source code, which is really convenient.

The apt command can automatically find the Ubuntu software server through the source configuration file, and download the software from the service. Image has added Ubuntu's official image source by default, so you don't need to re-edit the settings.

However, if some software does not exist in the official Ubuntu source, but a third-party Ubuntu software source can be provided. The user can first modify the configuration of the software source, add third-party software source, and then install the software online.

The image source for the Ubuntu 16.04 system is in /etc/apt/source.list.

Step1: Edit the source.list file to add a new image source.

Step2: Run the apt-get update command to update the image source.

2.3. Online Installation of Commonly Used Software

2.3.1. The apt command is used as follows:

- Install software package:
  # sudo apt-get install packagename
- Remove software package:
  # sudo apt-get remove packagename
- Get a list of new packages:
  # sudo apt-get update
- Upgrade the system with available updates:
  # sudo apt-get upgrade
- Query the required packages:
  #apt-cache search packagename
- List more commands and options:
  # apt-get help

For more information on the use of apt, you can check the relevant information online for a deeper understanding.

The installation of some commonly used software packages for industrial users is listed below:

2.3.2. Install Chromium Browser (Built-in)

```
# apt-get install chromium-browser
```

2.3.3. Install database software mysql

```
# apt-get install mysql-server
```

2.3.4. Install Apache web server

```
# apt-get install apache2 apache2-dev
```

2.3.5. Install PHP

```
# apt-get install php
```

2.3.6. Install Python

```
# apt-get install python
```

2.3.7. Install QtCreator

```
# apt-get install qtcreator
```

2.3.8. Install SSH (built-in)

```
# apt-get install openssh-server
```

2.3.9. Install VNC (built-in)

```
# apt-get install x11vnc
```

2.3.10. Install Mono

If installed in the default way:
sudo apt-get install mono-complete

The default version currently provided by Ubuntu is mono 4.0.
If the customer wants to use a higher version of mono 5.2, you can modify the software source configuration file by the method provided before, add Mono official software source, or you can modify the software source by the following command:

```
#sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
#echo "deb http://download.mono-project.com/repo/ubuntu xenial main" | sudo tee /etc/apt/sources.list.d/mono-official.list
#sudo apt-get update
#sudo apt-get install mono-complete
```

Up till now, mono 5.2 has been installed.

3. Ubuntu16.04 System Burning and Boot

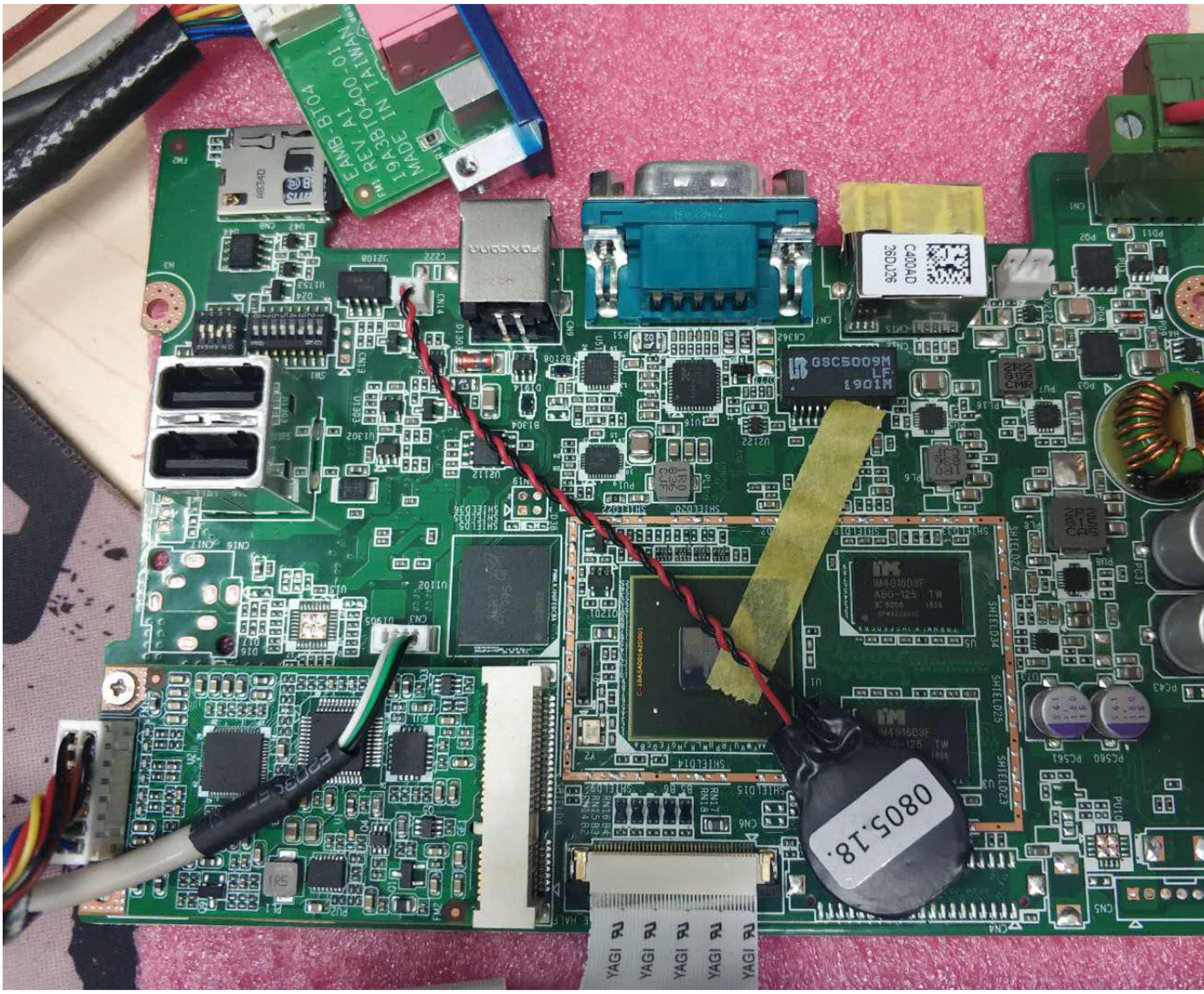
3.1. Required hardware and software environment

- TPC-71W
- Debug serial cable
- X86 development host with Ubuntu16.04 system
- TPC-71W Ubuntu Image
- SD card

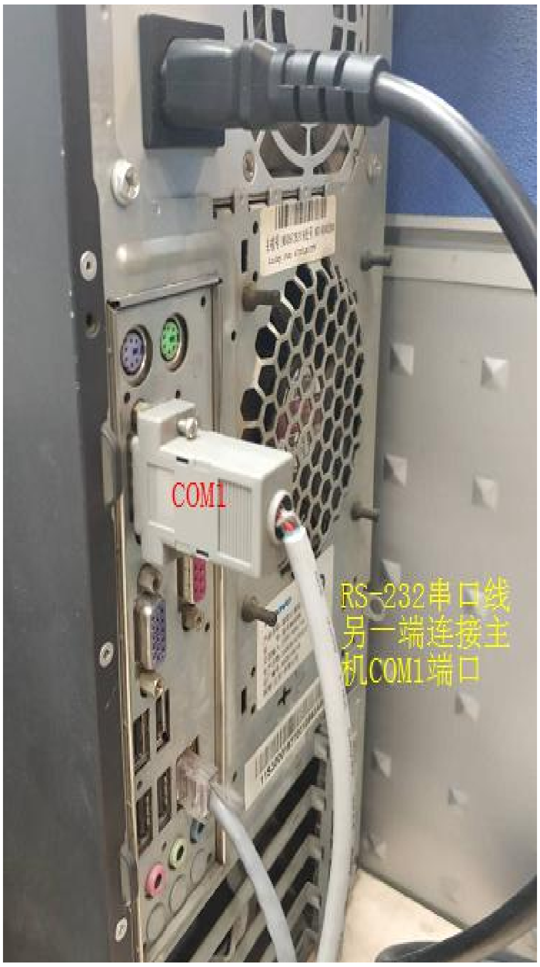
3.2. Debug port setting

(1) Connect debug serial cable.

Connect the RS-232 serial cable to the TPC-71W Debug port and the other end of the serial cable to your host.



TPC-71W



Development host

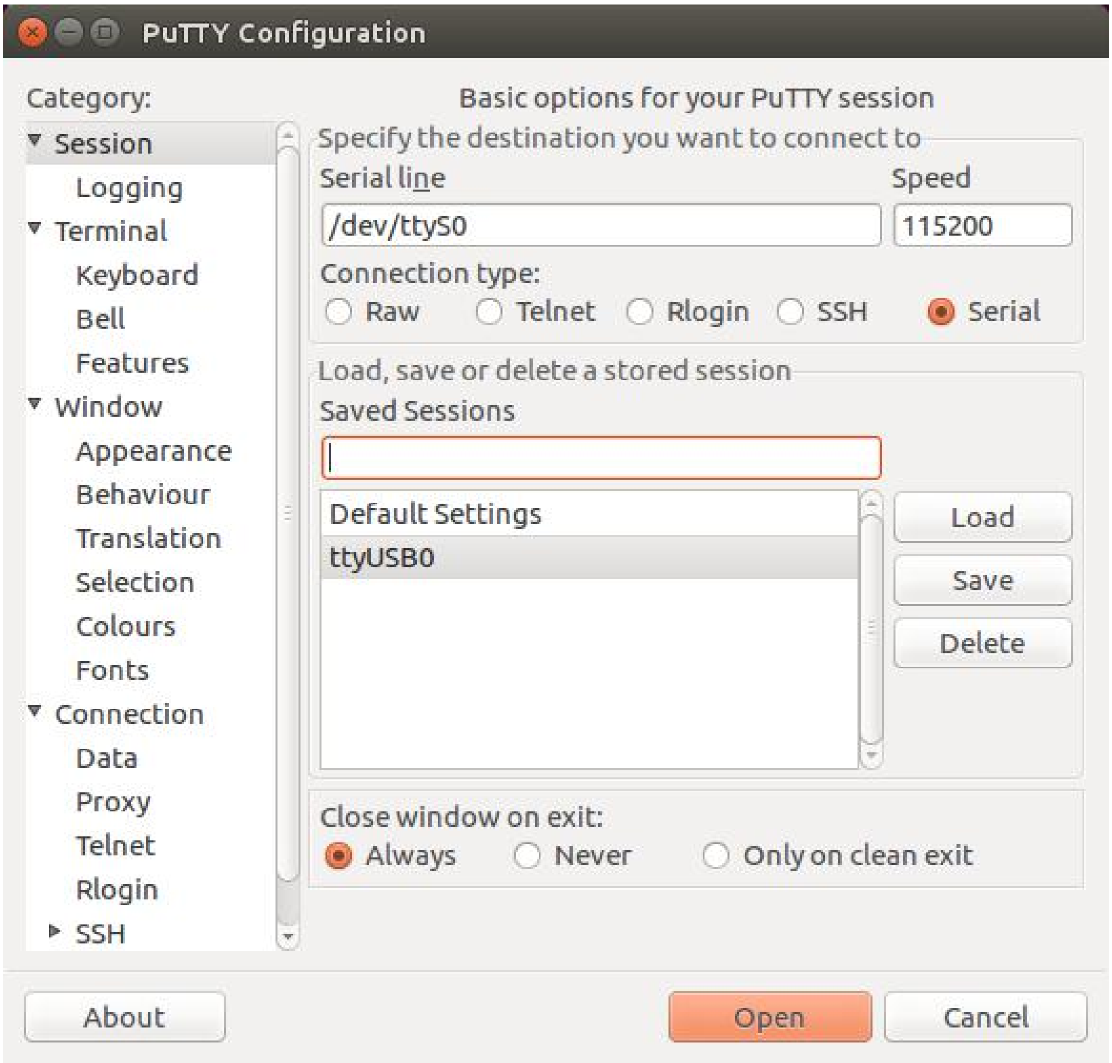
(2) Install the serial debugging tool putty under Linux.

```
# apt-get install putty
```

If it is a Windows environment, you can also download the Windows version of putty from the network to install.

(3) Use the putty tool to set the serial port number, the baud rate is 115200.

Please use the root user to open putty. The serial port number is the number of the serial port that debug first connects. For example, under Linux system, if the first serial port is connected (as shown above), it may be /dev/ttyS0; if you are using a USB to serial cable, it may be /dev/ttyUSB0.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyS0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

ttyUSB0

Close window on exit: ● Always ○ Never ○ Only on clean exit

putty configuration

(4) Connect the power supply and start the TPC-71W from the SD card or onboard storage. From the bootloader stage, debugging information is output to the putty window.

Note: If you are using the TPC-71W without a system installed, please refer to the following section to create a system SD card that can be booted, and you can burn the system to the onboard emmc via SD.

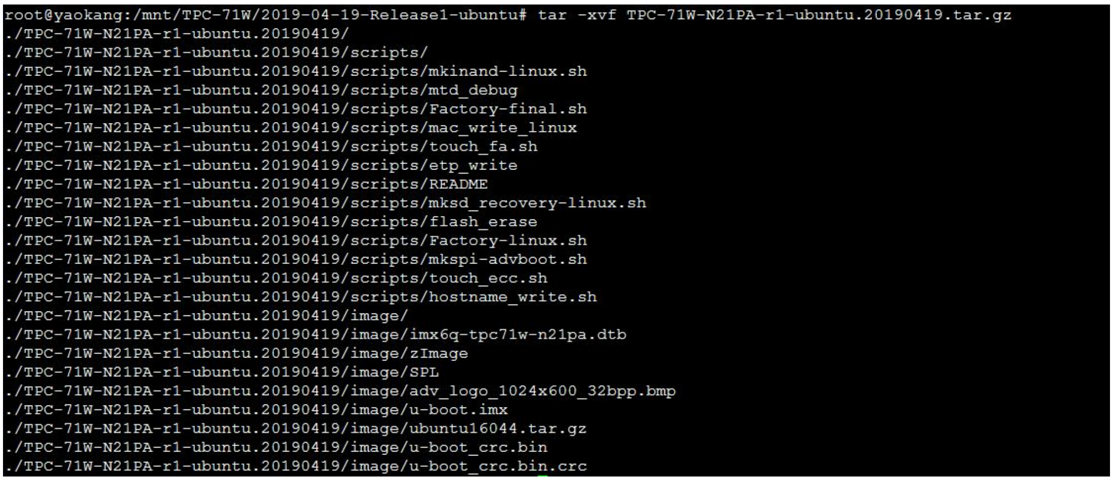
3.3. Preparation for Ubuntu Image Burning

3.4. Start TPC-71W with SD card

Use the SD card to start the TPC-71W and enter the system.

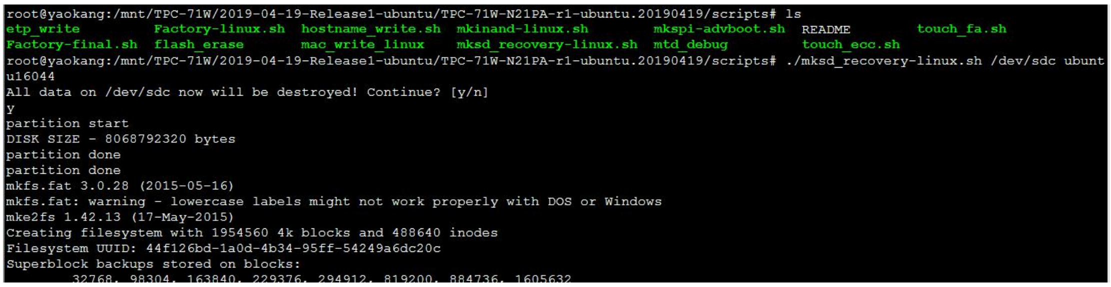
Under Linux:

1. Unzip the Ubuntu Image archive under Linux.



```
root@yaokang:/mnt/TPC-71W/2019-04-19-Release1-ubuntu# tar -xvf TPC-71W-N21PA-r1-ubuntu.20190419.tar.gz
./TPC-71W-N21PA-r1-ubuntu.20190419/
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/mkinand-linux.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/mtd_debug
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/Factory-final.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/mac_write_linux
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/touch_fa.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/etp_write
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/README
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/mksd_recovery-linux.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/flash_erase
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/Factory-linux.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/mkspi-advboot.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/touch_ecc.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/scripts/hostname_write.sh
./TPC-71W-N21PA-r1-ubuntu.20190419/image/
./TPC-71W-N21PA-r1-ubuntu.20190419/image/imx6q-tpc71w-n21pa.dtb
./TPC-71W-N21PA-r1-ubuntu.20190419/image/zImage
./TPC-71W-N21PA-r1-ubuntu.20190419/image/SPL
./TPC-71W-N21PA-r1-ubuntu.20190419/image/adv_logo_1024x600_32bpp.bmp
./TPC-71W-N21PA-r1-ubuntu.20190419/image/u-boot.imx
./TPC-71W-N21PA-r1-ubuntu.20190419/image/ubuntu16044.tar.gz
./TPC-71W-N21PA-r1-ubuntu.20190419/image/u-boot_crc.bin
./TPC-71W-N21PA-r1-ubuntu.20190419/image/u-boot_crc.bin.crc
```

2. Get the ubuntu image file, insert the SD into the host and perform the dd burning operation.

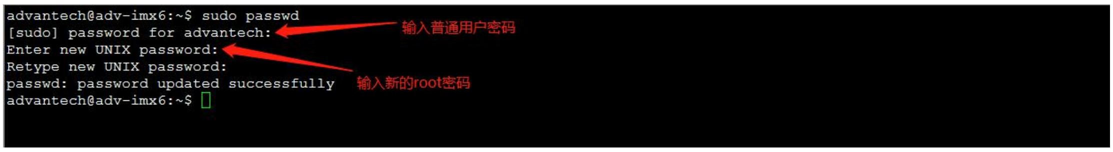


```
root@yaokang:/mnt/TPC-71W/2019-04-19-Release1-ubuntu/TPC-71W-N21PA-r1-ubuntu.20190419/scripts# ls
etp_write         Factory-linux.sh  hostname_write.sh  mkinand-linux.sh         mkspi-advboot.sh  README         touch_fa.sh
Factory-final.sh  flash_erase       mac_write_linux    mksd_recovery-linux.sh  mtd_debug         touch_ecc.sh
root@yaokang:/mnt/TPC-71W/2019-04-19-Release1-ubuntu/TPC-71W-N21PA-r1-ubuntu.20190419/scripts# ./mksd_recovery-linux.sh /dev/sdc ubunt
u16044
All data on /dev/sdc now will be destroyed! Continue? [y/n]
y
partition start
DISK SIZE - 8068792320 bytes
partition done
partition done
mkfs.fat 3.0.28 (2015-05-16)
mkfs.fat: warning - lowercase labels might not work properly with DOS or Windows
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 1954560 4k blocks and 488640 inodes
Filesystem UUID: 44f126bd-1a0d-4b34-95ff-54249a6dc20c
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

3. After the burning is completed, please remove the SD card and insert into the device to start (Jumper SW3: 1-off, 2-on).

3.5. Start the Ubuntu system

- After the system is launched:
  Common user: advantech   passwd：123
  Super user: root   passwd: 123456
- If you need to change the root password, you can change the root password by common users.

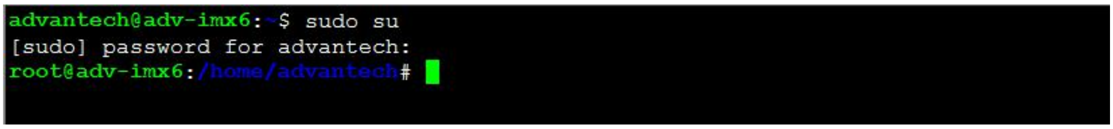


```
advantech@adv-imx6:~$ sudo passwd
[sudo] password for advantech:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
advantech@adv-imx6:~$ 
```

输入普通用户密码

输入新的root密码

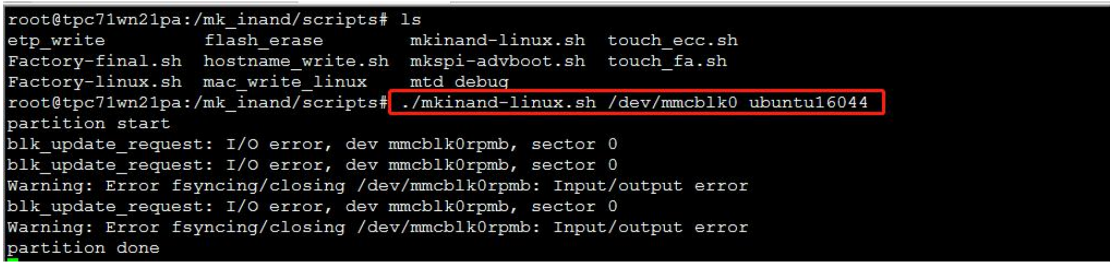
- Use the system:
  # sudo su (Enter your password to confirm)



```
advantech@adv-imx6:~$ sudo su
[sudo] password for advantech:
root@adv-imx6:/home/advantech# 
```

3.6. Start to Burn eMMC

- The system is started by SD card (/mk_inand/scripts directory)



```
root@tpc71wn21pa:/mk_inand/scripts# ls
etp_write          flash_erase        mkinand-linux.sh  touch_ecc.sh
Factory-final.sh   hostname_write.sh  mkspi-advboot.sh  touch_fa.sh
Factory-linux.sh   mac_write_linux    mtd_debug
root@tpc71wn21pa:/mk_inand/scripts# ./mkinand-linux.sh /dev/mmcblk0 ubuntu16044
partition start
blk_update_request: I/O error, dev mmcblk0rpmb, sector 0
blk_update_request: I/O error, dev mmcblk0rpmb, sector 0
Warning: Error fsyncing/closing /dev/mmcblk0rpmb: Input/output error
blk_update_request: I/O error, dev mmcblk0rpmb, sector 0
Warning: Error fsyncing/closing /dev/mmcblk0rpmb: Input/output error
partition done
```

Note: If the mkfs.vfat command is not found, you can use **[*# apt-get install dosfstools*]**

- After the burning is complete, sync with sync, then shut down and remove the SD card.
  Note: SD: /dev/mmcblk1p2   eMMc: /dev/mmcblk0p2

- After booting, you can choose to boot from SPI (jumper SW3 : 1-on 2-off).

4. TPC-71W Peripheral Test

4.1. eMMC Flash Read & Write Test

Step1: After the device boots from the SD card, run the following command to erase and check eMMC Flash.

(When booting from the SD card, the eMMC Flash node identified in the system is mmcblk1)

```
root@tpc71wn10pa:~# dd if=/dev/zero of=/dev/mmcblk0 bs=1024 count=1 seek=1
1+0 records in
1+0 records out
root@tpc71wn10pa:~# hexdump -C /dev/mmcblk0 -s 1024 -n 16
00000400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
```

Step 2: Run the following command to write and check the eMMC Flash.

```
root@tpc71wn10pa:~# echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk0 bs=1024 count=1 seek=1
0+1 records in
0+1 records out
root@tpc71wn10pa:~# hexdump -C /dev/mmcblk0 -s 1024 -n 16
00000400 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 |0123456789ABCDEF|
```

4.2. USB Read & Write Test

Step 1: Insert the USB storage device and view the TPC-71W device list to get the device node.

Step 2: Run the following command to erase and check the USB storage device.

```
root@tpc71wn10pa:~# dd if=/dev/zero of=/dev/sda bs=1024 count=1 seek=1
1+0 records in
1+0 records out
root@tpc71wn10pa:~# hexdump -C /dev/sda -s 1024 -n 16
01887800 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
```

Step 3: Run the following command to write and check the USB storage device.

```
root@tpc71wn10pa:~# echo -n "0123456789ABCDEF" | dd of=/dev/sda bs=1024
count=1 seek=1
0+1 records in
0+1 records out
root@tpc71wn10pa:~# hexdump -C /dev/sda -s 1024 -n 16
00000400 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 |0123456789ABCDEF|
```

Note! *1. NXP i.MX6D/Q has the limitation on USB device collection, we rec-ommend the follow brands: Logitech K120 / Lenovo K5819 LXH - EKB-10YA / RAPOO 1800.Pro / Dell MS111-P / Microsoft Wired Keyboard 200 (Model:1406) and so on.*

2. *This operation **may damage the data stored** in USB flash disk.Please make sure there is no critical data in the USB flash disk being used for this test. If your U Disk size is small, the **seek value need to be small**.*

4.3. SD Card Read & Write Test

Step 1: When the device boots from the internal eMMC Flash (the SD card is not inserted at startup), the following information can be viewed from the system.

```
root@imx6qitb200a1:~# ls /dev/mmcblk* -l
brw-rw---- 1 root disk 179,  8 Feb  5 17:01 /dev/mmcblk0
brw-rw---- 1 root disk 179, 16 Feb  5 17:01 /dev/mmcblk0boot0
brw-rw---- 1 root disk 179, 24 Feb  5 17:01 /dev/mmcblk0boot1
brw-rw---- 1 root disk 179,  9 Feb  5 17:01 /dev/mmcblk0p1
brw-rw---- 1 root disk 179, 10 Feb  5 17:01 /dev/mmcblk0p2
brw-rw---- 1 root disk 179, 32 Feb  5 17:01 /dev/mmcblk0rpmb
```

Step 2: Insert the SD card into the TPC-71W and re-check the device information. /dev/mmcblk1 represents the current SD card device (in this example, the SD card has two partitions).

```
root@imx6qitb200a1:~# ls -l /dev/mmcblk*
```

```
brw-rw---- 1 root disk 179,   8 Feb   5 17:08 /dev/mmcblk0
brw-rw---- 1 root disk 179, 16 Feb   5 17:08 /dev/mmcblk0boot0
brw-rw---- 1 root disk 179, 24 Feb   5 17:08 /dev/mmcblk0boot1
brw-rw---- 1 root disk 179,   9 Feb   5 17:08 /dev/mmcblk0p1
brw-rw---- 1 root disk 179, 10 Feb   5 17:08 /dev/mmcblk0p2
brw-rw---- 1 root disk 179, 32 Feb   5 17:08 /dev/mmcblk0rpmb
brw-rw---- 1 root disk 179,   0 Feb   5 17:08 /dev/mmcblk1
brw-rw---- 1 root disk 179,   1 Feb   5 17:08 /dev/mmcblk1p1
brw-rw---- 1 root disk 179,   2 Feb   5 17:08 /dev/mmcblk1p2
```

Step 3:   Run the following command to erase and check the SD card.

```
root@tpc71wn10pa:~# dd if=/dev/zero of=/dev/mmcblk1 bs=1024 count=1 seek=1
1+0 records in
1+0 records out
root@tpc71wn10pa:~# hexdump -C /dev/mmcblk1 -s 1024 -n 16
01887800 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
```

Step 4:   Write and check SD card.

```
root@tpc71wn10pa:~# echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk1
bs=1024 count=1 seek=25118
0+1 records in
0+1 records out
root@tpc71wn10pa:~# hexdump -C /dev/mmcblk1 -s 1024 -n 16
01887800 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46 |0123456789ABCDEF|
```

4.4. Serial Port Test

As shown in the table below, the TPC-71W has two serial ports. By default, COM1 is configured to TPC-71W debugging serial port, and COM2 is configured to RS232 mode.

HW	SW	DEVICE
COM2	232 / 485	/dev/ttyUSB0
COM1	Debug	/dev/ttymxc0

Test COM1 rs-232 loopback(baudrate 9600):

```
root@tpc71wn10pa:~# cd /usr/Advantech/Serial_test
root@tpc71wn10pa:/usr/Advantech/Serial_test#  ./st  -rsavo  -m  232  -b  9600
/dev/ttyUSB0
```

Test COM1 rs-232 read(baudrate 9600):

```
root@tpc71wn10pa:~# cd /usr/Advantech/Serial_test
root@tpc71wn10pa:/usr/Advantech/Serial_test#   ./st   -ravo   -m   232   -b   9600
/dev/ttyUSB0
```

Test COM1 rs-232 write(baudrate 9600):

```
root@tpc71wn10pa:~# cd /usr/Advantech/Serial_test
root@tpc71wn10pa:/usr/Advantech/Serial_test#   ./st   -savo   -m   232   -b   9600
/dev/ttyUSB0
```

Test COM1 rs-485 read(baudrate 115200):

```
root@tpc71wn10pa:~# cd /usr/Advantech/Serial_test
root@tpc71wn10pa:/usr/Advantech/Serial_test#   ./st   -ravo   -m   485   -b   115200
/dev/ttyUSB0
```

Test COM1 rs-485 write(baudrate 115200):

```
root@tpc71wn10pa:~# cd /usr/Advantech/Serial_test
root@tpc71wn10pa:/usr/Advantech/Serial_test#   ./st   -savo   -m   485   -b   115200
/dev/ttyUSB0
```

4.5. LAN Port test

4.5.1 TPC-71W sets DHCP as default network protocol

```
root@tpc71wn10pa:~# ifconfig eth0
eth0        Link encap:Ethernet    HWaddr 00:0b:ab:39:47:f7
             inet addr:169.254.101.178    Bcast:169.254.255.255    Mask:255.255.0.0
             UP BROADCAST RUNNING MULTICAST    MTU:1500    Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:155 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:0 (0.0 B)    TX bytes:58655 (57.2 KiB)
```

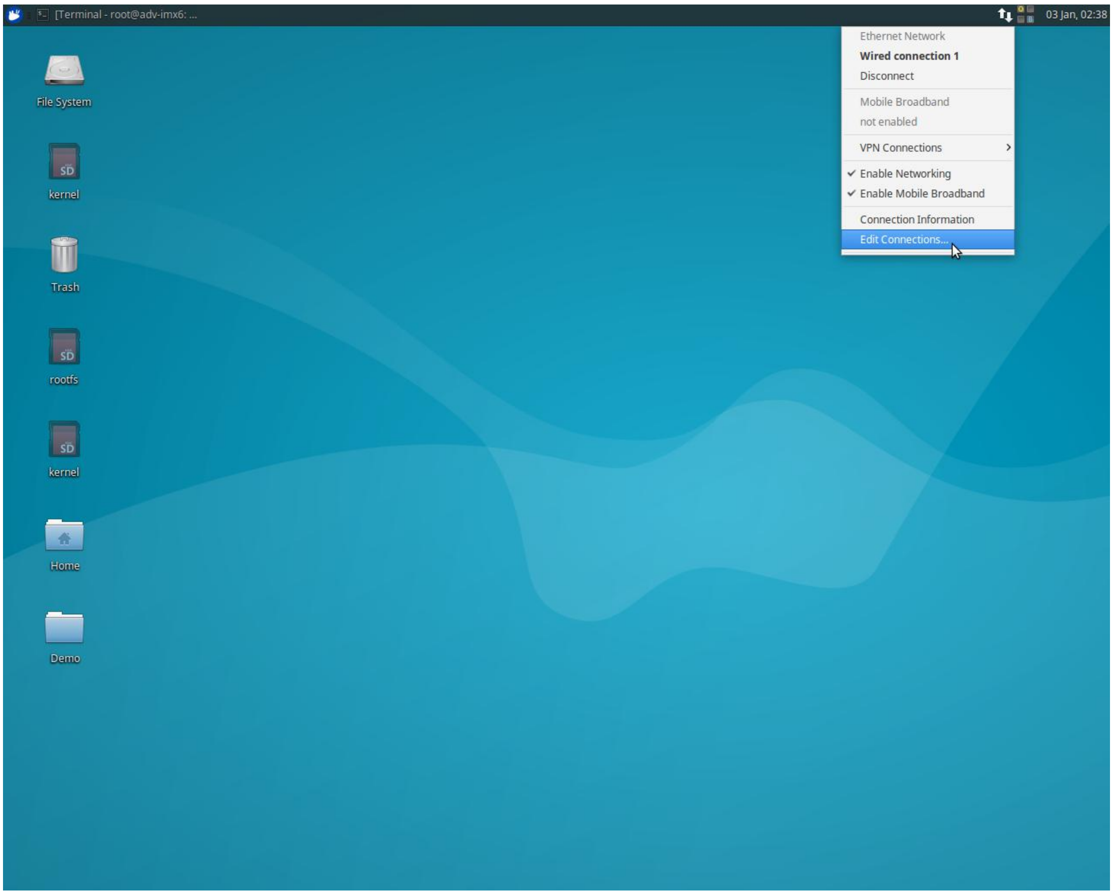
```
root@adv-imx6:~# ifconfig eth0 down
root@adv-imx6:~# ifconfig eth0 172.21.73.151 up
root@adv-imx6:~# ifconfig eth0
eth0        Link encap:Ethernet    HWaddr 00:0b:ab:e2:88:75
             inet addr:172.21.73.151    Bcast:172.21.255.255    Mask:255.255.0.0
             inet6 addr:    fe80::20b:abff:fee2:8875/64 Scope:Link
```

```
          UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
          RX packets:10448 errors:0 dropped:0 overruns:0 frame:0
          TX packets:681 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1120906 (1.1 MB)   TX bytes:73732 (73.7 KB)
```

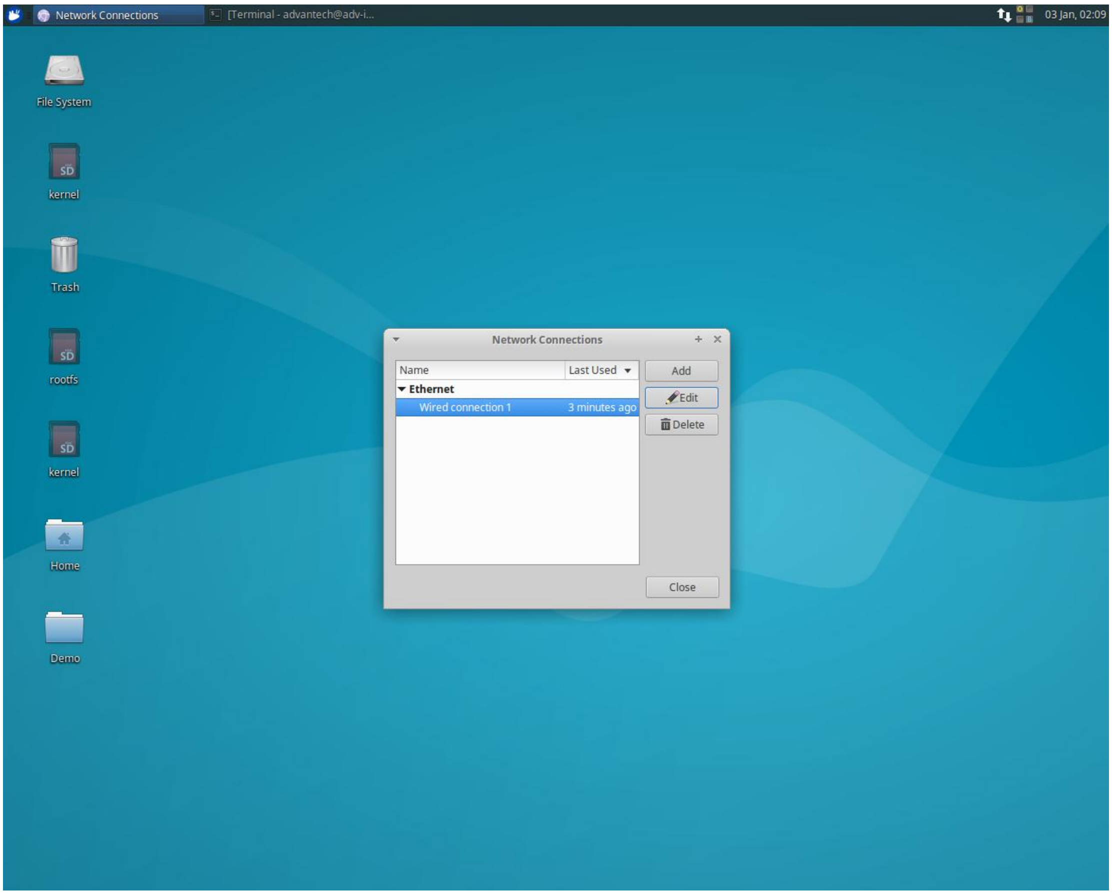
```
root@yaokang:~# ping 172.21.73.151
PING 172.21.73.151 (172.21.73.151) 56(84) bytes of data.
64 bytes from 172.21.73.151:   icmp_seq=1 ttl=64 time=0.602 ms
64 bytes from 172.21.73.151:   icmp_seq=2 ttl=64 time=0.415 ms
64 bytes from 172.21.73.151:   icmp_seq=3 ttl=64 time=0.402 ms
64 bytes from 172.21.73.151:   icmp_seq=4 ttl=64 time=0.458 ms
```

4.5.2 Configuring Static IP in the Desktop Graphics System

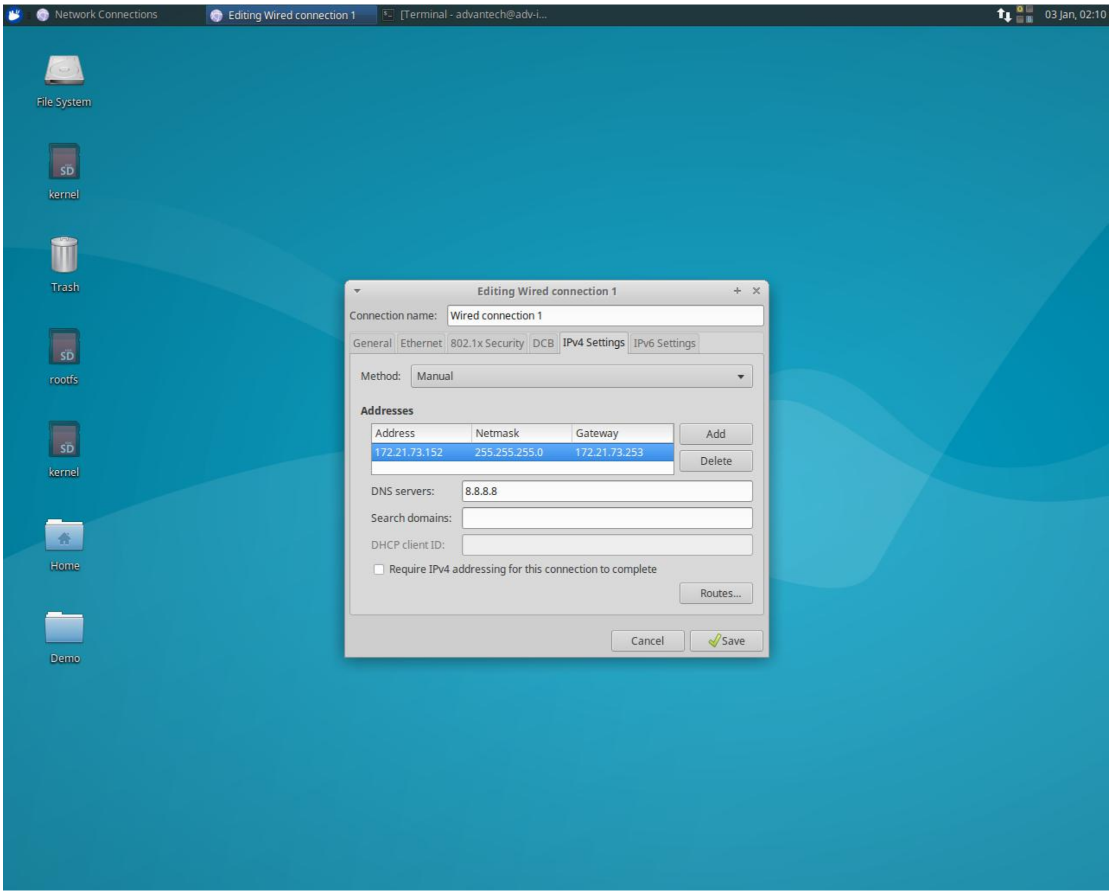
Step1: Click "Edit Connections" in the upper right corner.



Step2: Click "Edit".



Step3: Set IP, subnet mask, gateway, DNS and then click "Save".



Connection name: Wired connection 1

Method: Manual

Addresses

Address	Netmask	Gateway
172.21.73.152	255.255.255.0	172.21.73.253

DNS servers: 8.8.8.8

Search domains:

DHCP client ID:

□ Require IPv4 addressing for this connection to complete

Step4: Plug and unplug the network cable and view the IP through ifconfig

4.6. Connecting WIFI Modules

4.6.1Supported Modules

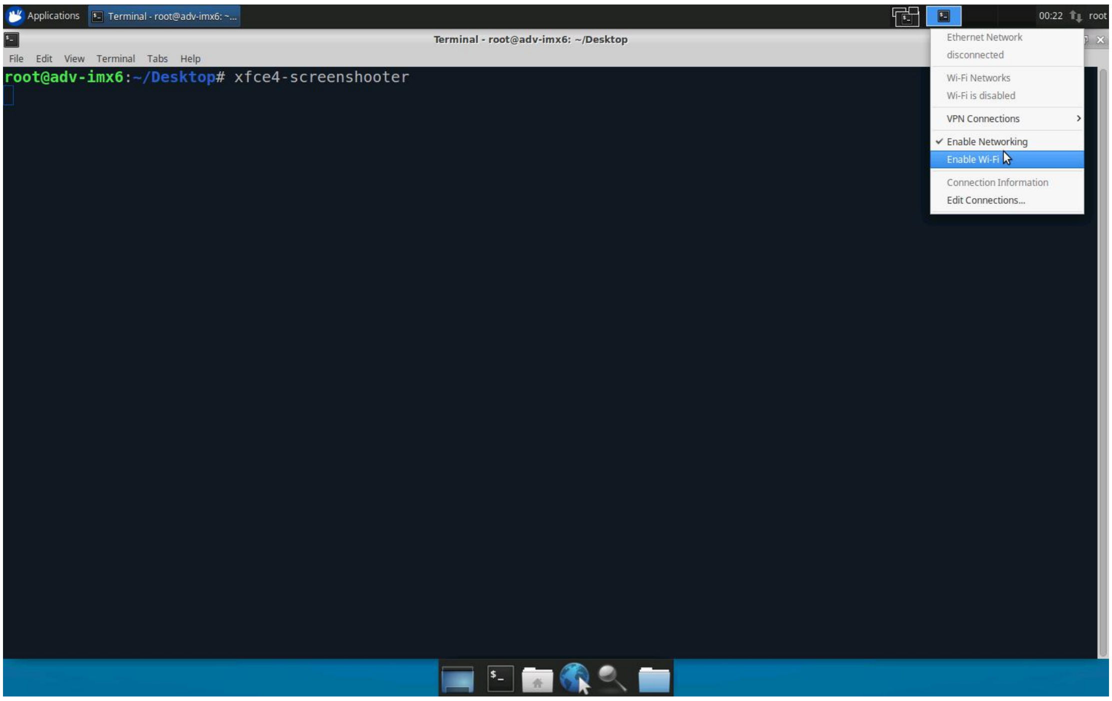
WiFi model: EWM-W151H01E, RTL8188EE

4.6.2 WIFI Configuration and Connection (EWM-W151H01E)

Configuring via graphics

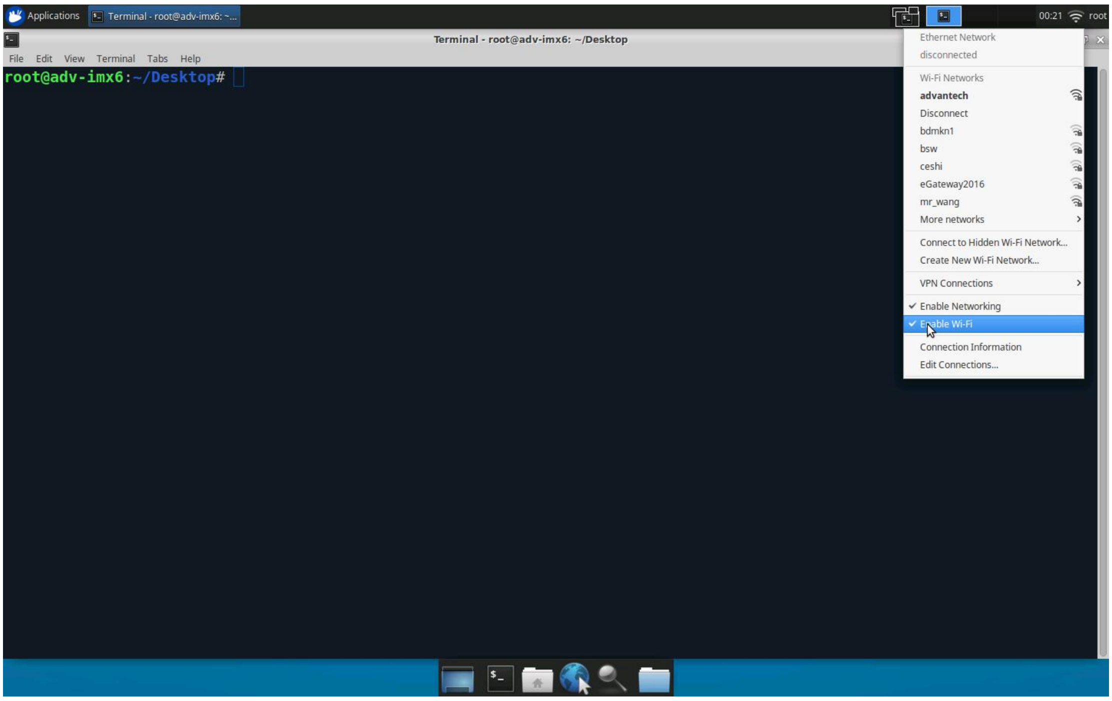
Step1: There is a network connection icon in the upper right corner of the main interface of the system. ⇅

Step2: Click the icon to enable WiFi.



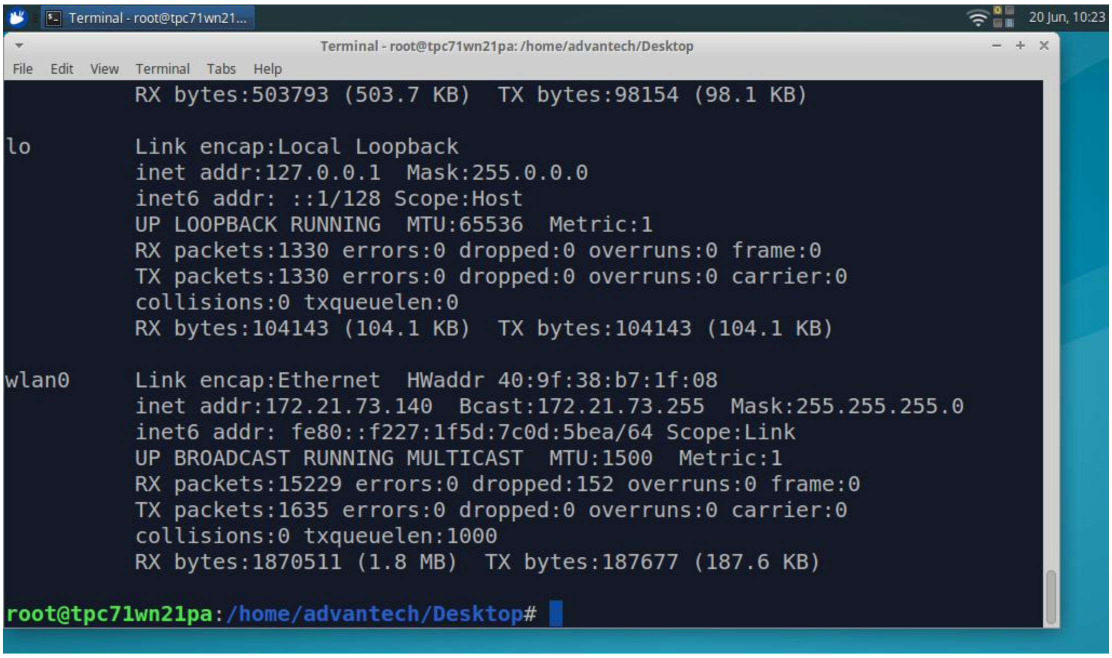
```
root@adv-imx6:~/Desktop# xfce4-screenshooter
```

Setp3: After enabling WiFi, you can view the surrounding wireless network and select one to connect.



```
root@adv-imx6:~/Desktop#
```

Step4: After connecting successfully, you can surf the Internet.



Terminal - root@tpc71wn21...
20 Jun, 10:23
Terminal - root@tpc71wn21pa: /home/advantech/Desktop
File
Edit
View
Terminal
Tabs
Help

```
          RX bytes:503793 (503.7 KB)  TX bytes:98154 (98.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1330 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1330 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:104143 (104.1 KB)  TX bytes:104143 (104.1 KB)

wlan0     Link encap:Ethernet  HWaddr 40:9f:38:b7:1f:08
          inet addr:172.21.73.140  Bcast:172.21.73.255  Mask:255.255.255.0
          inet6 addr: fe80::f227:1f5d:7c0d:5bea/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15229 errors:0 dropped:152 overruns:0 frame:0
          TX packets:1635 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1870511 (1.8 MB)  TX bytes:187677 (187.6 KB)

root@tpc71wn21pa:/home/advantech/Desktop#
```

4.7. Connect & Test 4G/LTE Modules

4.7.1 Supported Modules

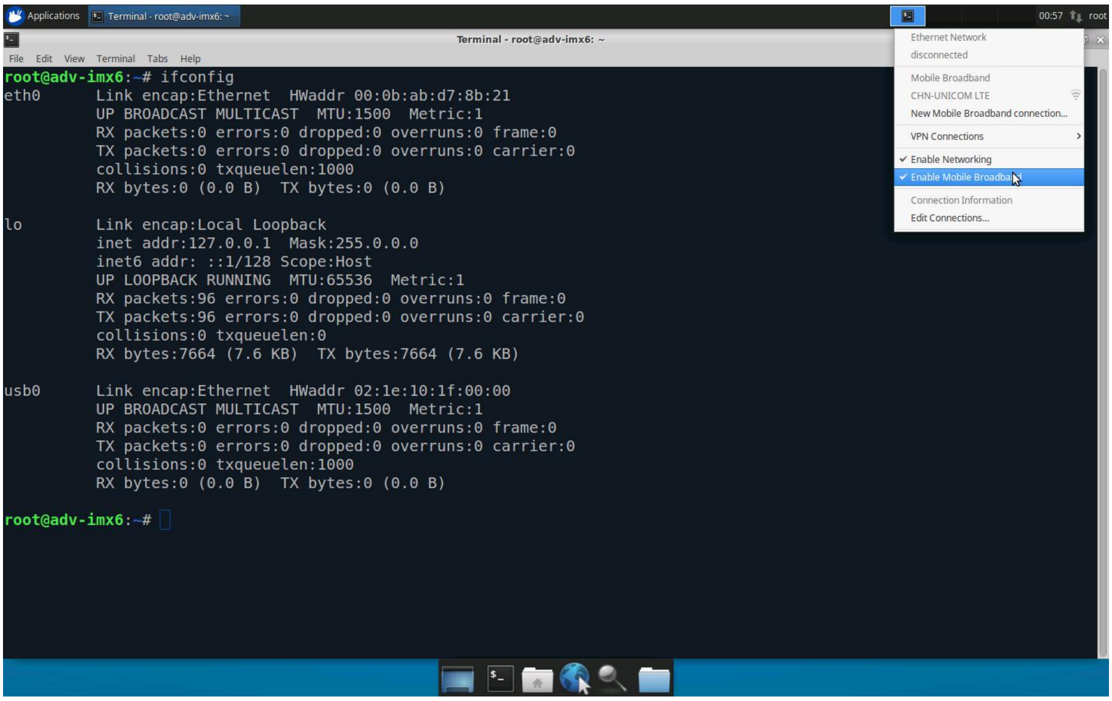
4G/LTE module model: EWM-C117FL01E

4.7.2 4G/LTE Module Configuration & Connection (Model EWM-C117FL01E)

Configuring via graphics

Step1: Click the connection icon in the upper right corner of the main interface of the system.⇅

Step2: Click the icon to enable 3G and Mobile Broadband.



```
root@adv-imx6:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0b:ab:d7:8b:21
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7664 (7.6 KB)  TX bytes:7664 (7.6 KB)

usb0      Link encap:Ethernet  HWaddr 02:1e:10:1f:00:00
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@adv-imx6:~#
```

Step3: After clicking "New Mobile Broadband" connection, enter the password in the pop-up the dialog box and click "Next" to verify.

4.8. Time & Date Setting

Set system time (2019/01/01 13:25:00):

```
root@tpc71wn10pa:~# date -s “2019/01/01 13:25:00”
```

Synchronize time from the NTP server:

```
root@tpc71wn10pa:~# ntpdate <NTPSERVERIP>
```

Reset RTC hardware clock time (use current system time):

```
root@tpc71wn10pa:~# hwclock -w
```

Reset system time (use RTC hardware clock time):

```
root@tpc71wn10pa:~# hwclock -s
```

Set system time zone (use Shanghai time):

```
root@tpc71wn10pa:~# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
root@tpc71wn10pa:~# sync
```

4.9. CAN Test

As you can see below, there are 1 flexCAN supported by TPC-71W internal.

HW	DEVICE	MODE
flexCAN0	can0	socket can

Setting: Open flexCAN device (125000 biterate, loopback off)

```
root@tpc71wn10pa:~# ip link set can0 down
root@tpc71wn10pa:~# ip link set can0 up type can bitrate 125000 loopback off
root@tpc71wn10pa:~# ip link set can0 up
root@tpc71wn10pa:~# ifconfig can0
can0       Link encap:UNSPEC    HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
           UP RUNNING NOARP    MTU:16    Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:10
           RX bytes:0 (0.0 B)    TX bytes:0 (0.0 B)
           Interrupt:31
```

Check can0 status:

```
root@tpc71wn10pa:~# ip -details link show can0
3:    can0:    <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UNKNOWN mode DEFAULT
group default qlen 10
    link/can    promiscuity 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 125000 sample-point 0.875
          tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
          flexcan:    tseg1 4..16 tseg2 2..8 sjw 1..4 brp 1..256 brp-inc 1
          clock 30000000
```

Send message (“123#11") to socket can0:

```
root@tpc71wn10pa:~# cansend can0 123#11
```

Recv message from socket can0:

```
root@tpc71wn10pa:~# candump can0
```

4.10. Watchdog Test

Please refer “EAPI Development Guide” to get more details.
Enable wdt with timeout value:

```
root@tpc71wn10pa:~# cd /usr/Advantech/EAPI_test
root@tpc71wn10pa:/usr/Advantech/EAPI_test# ./testdl_wdt -s <timeout>

# For example(enable wdt & set timeout=10s):
root@tpc71wn10pa:/usr/Advantech/EAPI_test# ./testdl_wdt -s 10
MaxDelay:0 MaxEventTimeout:0 MaxResetTimeout:6553
WDT start.
WDT timeout has been set to 10 seconds.
After that, WDT will reset CPU.
```

Disable wdt:

```
root@tpc71wn10pa:~# cd /usr/Advantech/EAPI_test
root@tpc71wn10pa:/usr/Advantech/EAPI_test# ./testdl_wdt -d
MaxDelay:0 MaxEventTimeout:0 MaxResetTimeout:6553
WDT stop.
```

4.11. Brightness Test

Please refer “EAPI Development Guide” to get more details.
increased the brightness step by step every second:

```
root@tpc71wn10pa:~# cd /usr/Advantech/EAPI_test
root@tpc71wn10pa:/usr/Advantech/EAPI_test# ./testdl_brightness
Value:    7
Current bright:    0
Current bright:    0
Current bright:    1
Current bright:    1
Current bright:    2
Current bright:    2
Current bright:    3
Current bright:    3
Current bright:    4
Current bright:    4
Current bright:    5
Current bright:    5
Current bright:    6
Current bright:    6
Current bright:    7
Current bright:    7
```

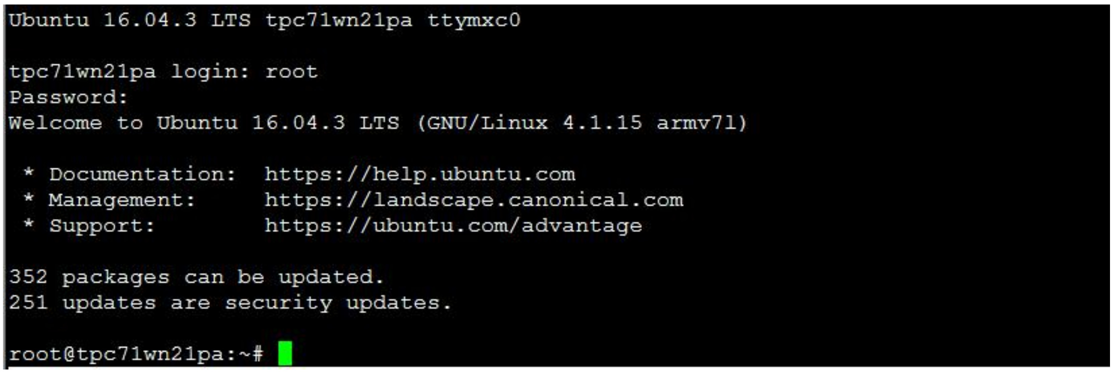
4.12. HWmon Test

Please refer “EAPI Development Guide” to get more details.
Gets the current CPU temperature (accurate in 3 decimal):

```
root@tpc71wn10pa:~# cd /usr/Advantech/EAPI_test
root@tpc71wn10pa:/usr/Advantech/EAPI_test# ./testdl_hwmon
BoardGetValue Id:    0x50000
Value:    60557
```

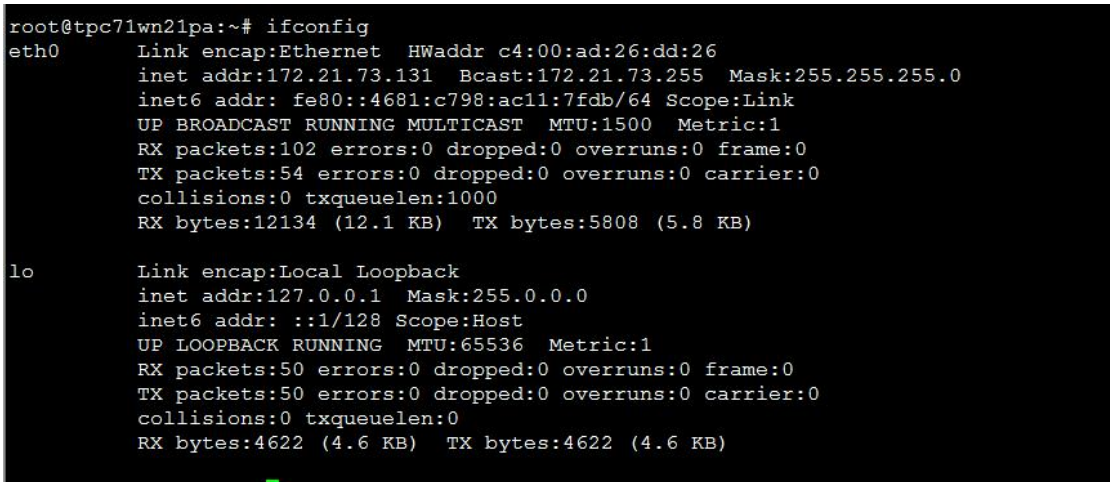
4.13. X11vnc Test

Step1: login with debug console



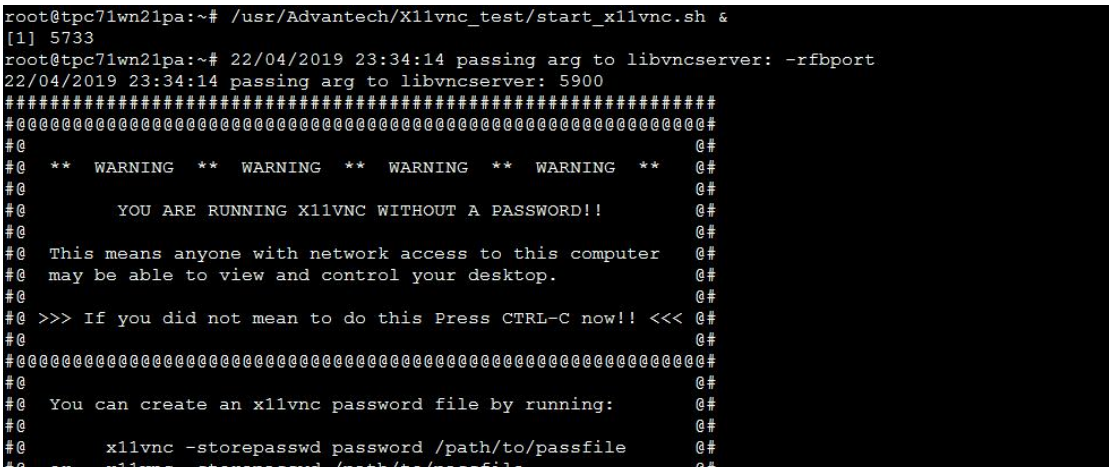
```
Ubuntu 16.04.3 LTS tpc71wn21pa ttymxc0

tpc71wn21pa login: root
Password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.1.15 armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

352 packages can be updated.
251 updates are security updates.

root@tpc71wn21pa:~#
```

Step2: get current ethernet IP



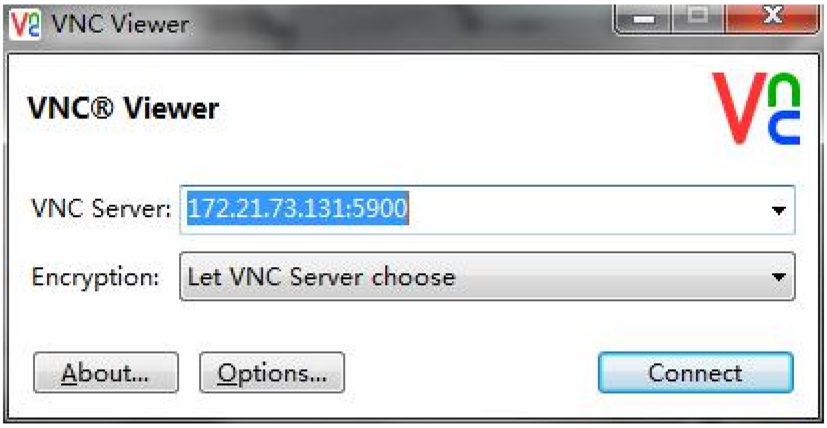
```
root@tpc71wn21pa:~# ifconfig
eth0      Link encap:Ethernet  HWaddr c4:00:ad:26:dd:26
          inet addr:172.21.73.131  Bcast:172.21.73.255  Mask:255.255.255.0
          inet6 addr: fe80::4681:c798:ac11:7fdb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12134 (12.1 KB)  TX bytes:5808 (5.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4622 (4.6 KB)  TX bytes:4622 (4.6 KB)
```

Step3:   start x11vnc server



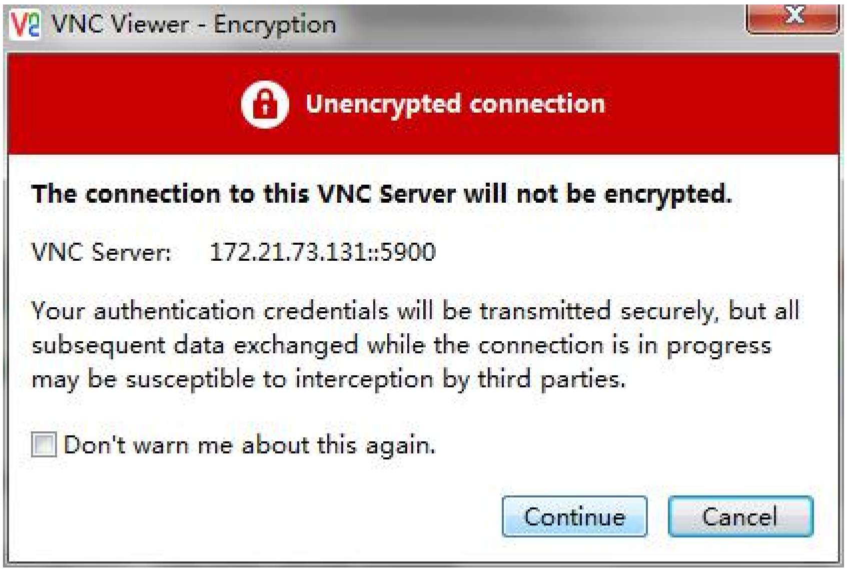
```
root@tpc71wn21pa:~# /usr/Advantech/X11vnc_test/start_x11vnc.sh &
[1] 5733
root@tpc71wn21pa:~# 22/04/2019 23:34:14 passing arg to libvncserver: -rfbport
22/04/2019 23:34:14 passing arg to libvncserver: 5900
###############################################################
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#
#@                                                           @#
#@  **  WARNING  **  WARNING  **  WARNING  **  WARNING  **   @#
#@                                                           @#
#@        YOU ARE RUNNING X11VNC WITHOUT A PASSWORD!!        @#
#@                                                           @#
#@  This means anyone with network access to this computer   @#
#@  may be able to view and control your desktop.            @#
#@                                                           @#
#@ >>> If you did not mean to do this Press CTRL-C now!! <<< @#
#@                                                           @#
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#
#@                                                           @#
#@  You can create an x11vnc password file by running:       @#
#@                                                           @#
#@       x11vnc -storepasswd password /path/to/passfile      @#
#@  or   x11vnc -storepasswd /path/to/passfile
```

Step4:   Remote desktop (use VNC Viewer 6.18.625)



VNC Server: 172.21.73.131:5900

Encryption: Let VNC Server choose



VNC Viewer - Encryption

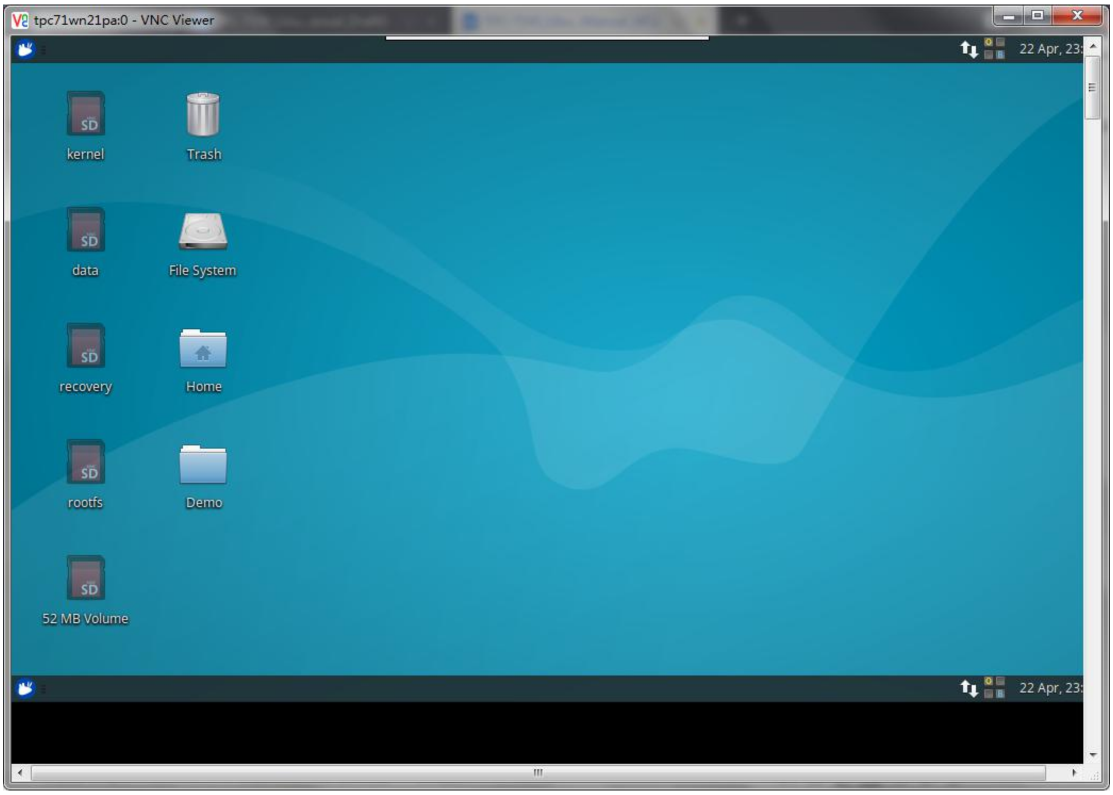
Unencrypted connection

The connection to this VNC Server will not be encrypted.

VNC Server: 172.21.73.131::5900

Your authentication credentials will be transmitted securely, but all subsequent data exchanged while the connection is in progress may be susceptible to interception by third parties.

□ Don't warn me about this again.



5. System Configuration

5.1. Terminal Command Line

Many system operations and configurations under Linux are performed under the command line. There are two ways to start the command line:

Method 1: Start a terminal and select Applications -> Accessories -> Terminal

Method 2: Shortcut: Ctrl + Alt + T

On the command line, you can run a variety of shell commands and scripts, common commands

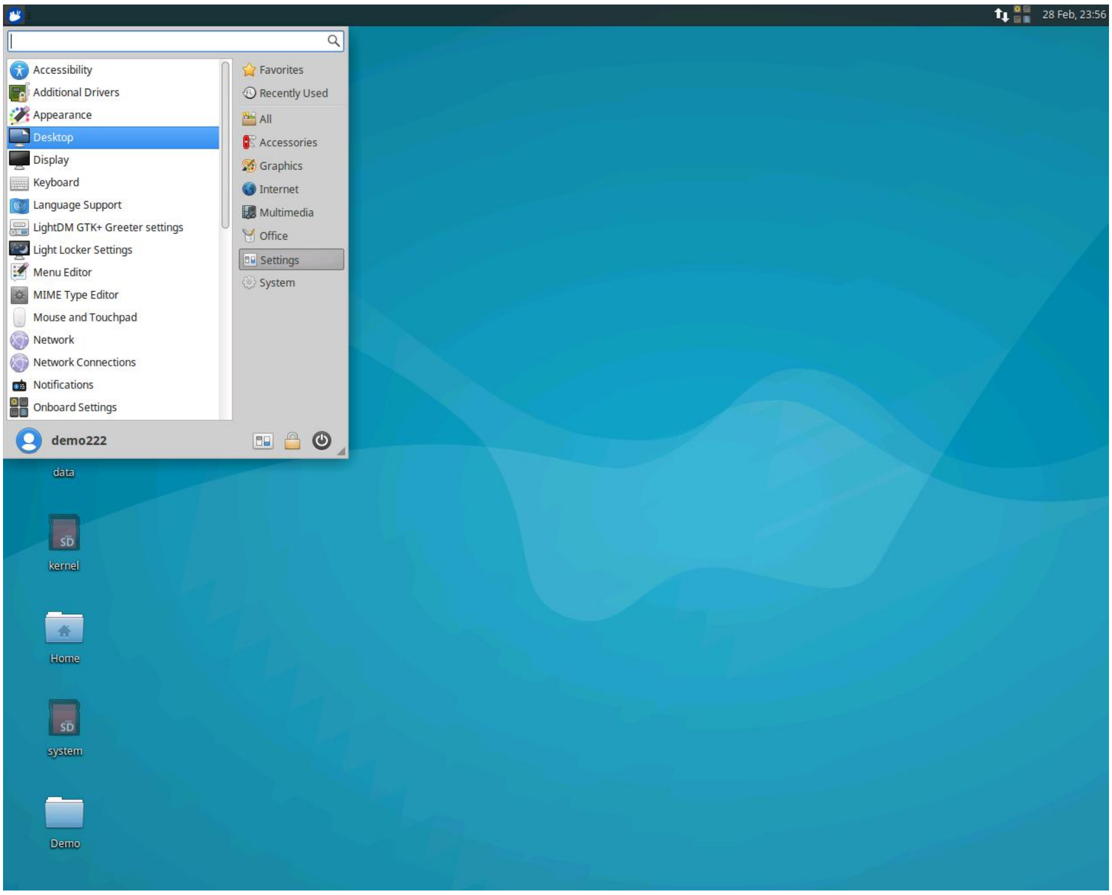
- View directory: ls
- Create a directory: mkdir (Directory name)
- Switch directory: cd (directory/location)
- Copy files/directories: cp (Source file or directory name) (Destination directory or file name)
- Delete files/directories: rm (File/Directory name)
- Rename the file/directory: mv (File/Directory name)
- Query file/directory: locate (File/Directory name)
- pwd Show current directory
- ifconfig Display system network

The command line is the most basic operation under Linux. It is an essential skill for every Linux developer and user. Thus it is not explained here.

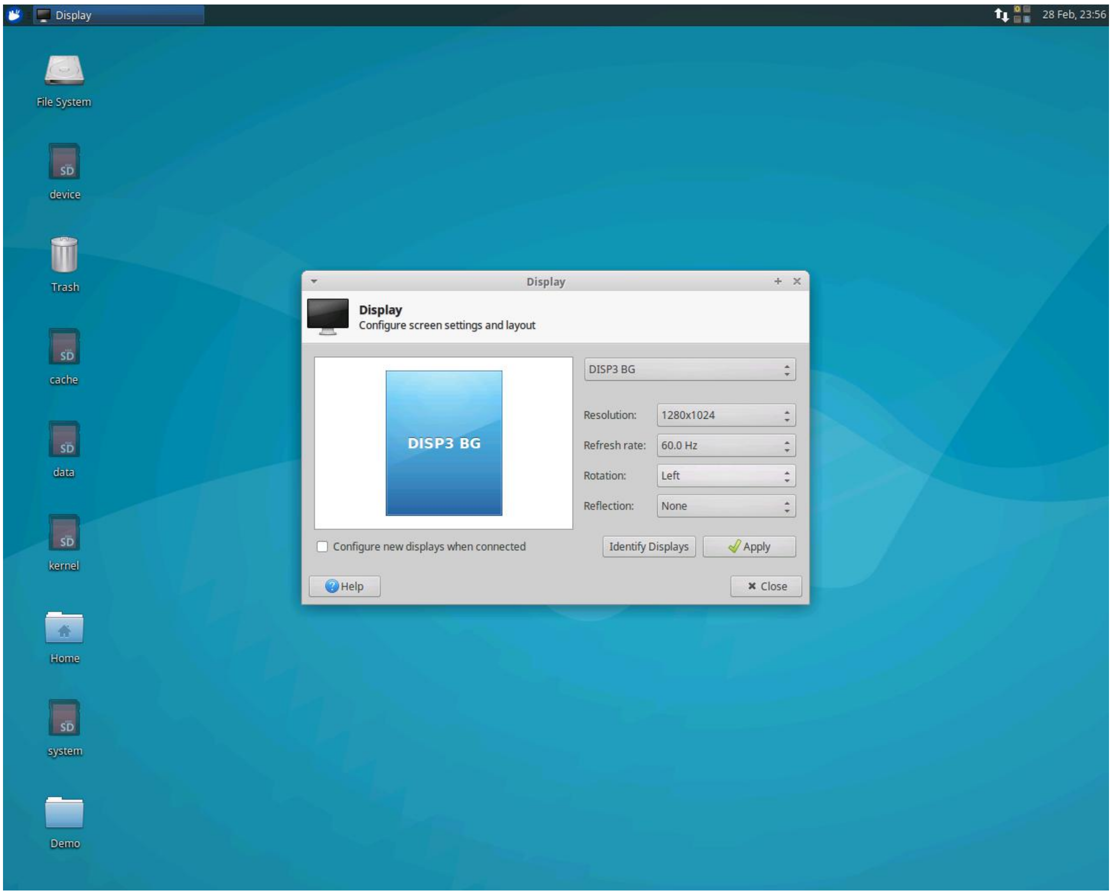
5.2. Display Output Configuration

5.2.1. Screen Flip Setting

Step1: Click Setting --> Display to enter the graphical interface.



Setp2: Set "Rotation" to "Left".



Display

Configure screen settings and layout

DISP3 BG

Resolution: 1280x1024

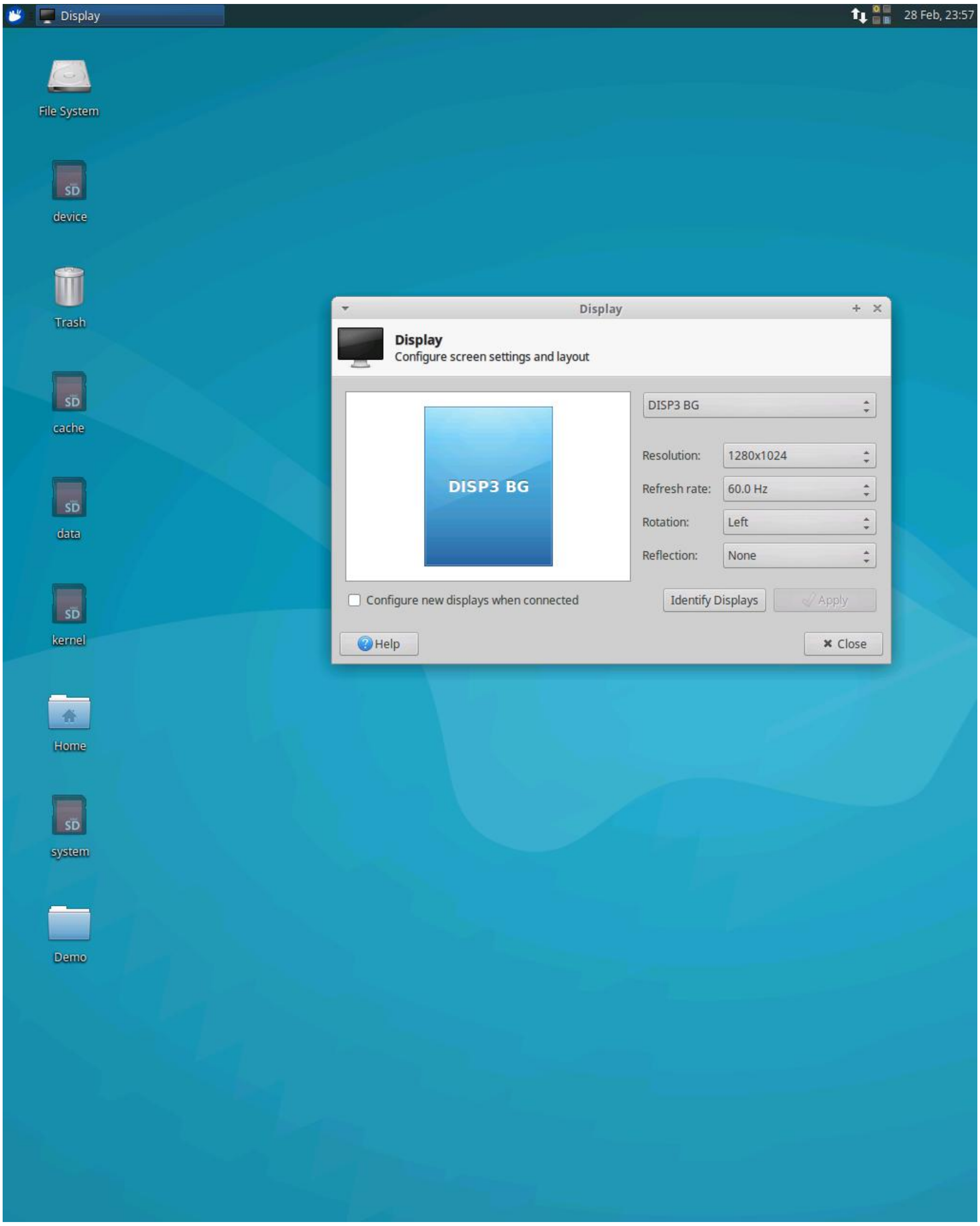
Refresh rate: 60.0 Hz

Rotation: Left

Reflection: None

□ Configure new displays when connected

Setp3: After the setting is completed, it is displayed as flipping to the left



Display
Configure screen settings and layout

DISP3 BG

Resolution: 1280x1024

Refresh rate: 60.0 Hz

Rotation: Left

Reflection: None

□ Configure new displays when connected

5.3. Language Setting

Step1: Applications --> Settings --> Language Support

Step2: Find [chinese(simplified)] in the pop-up window, and finally click [apply change].

Step3：After setting the Chinese language in Ubuntu, you need to close ubuntu and it will not take effect until it is restarted.

5.4. Add User & Password

Step1: Create a new user.

# useradd -d /home/test -g root -m test //Create a new test user and specify the user group as the root user group, and automatically create a login directory

# passwd test //Set a password for the test user

Step2: Add a working group to an existing user

# usermod -G root test //Set the root subgroup to the test user.

gpasswd -a test root //Set the root user group to the test user.

5.5. Start a Specific Program Automatically at Boot

The system default boot self-starting file is in /etc/rc.local, and the specific program that the client will run is written in the form of a script. 1. Run the script in the rc.local file.

For example:

(1) If you need to start the demo program at boot, please first create a sh script file, and write the demo to the script.

# vi demo.sh

cd demo/ && ./demo

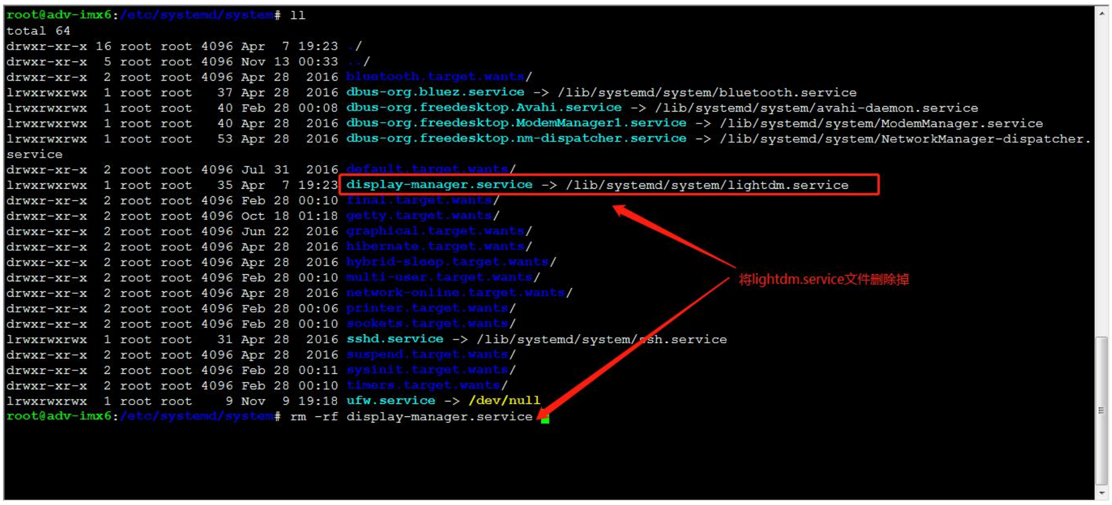
(2) After the execution of the script is completed, the command to run the script is written in /etc/rc.local.

# vi /etc/rc.local

/root/demo.sh

2. Start automatically at boot in graphical mode

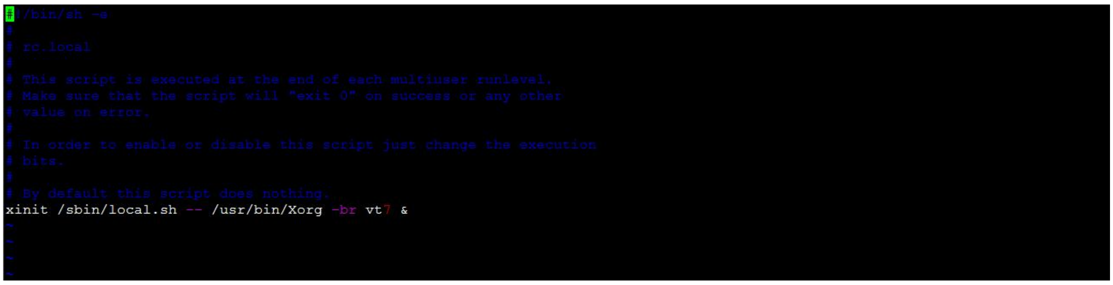
(1) First delete the service file of the login manager



```
root@adv-imx6:/etc/systemd/system# ll
total 64
drwxr-xr-x 16 root root 4096 Apr  7 19:23 ./
drwxr-xr-x  5 root root 4096 Nov 13 00:33 ../
drwxr-xr-x  2 root root 4096 Apr 28  2016 bluetooth.target.wants/
lrwxrwxrwx  1 root root   37 Apr 28  2016 dbus-org.bluez.service -> /lib/systemd/system/bluetooth.service
lrwxrwxrwx  1 root root   40 Feb 28 00:08 dbus-org.freedesktop.Avahi.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx  1 root root   40 Apr 28  2016 dbus-org.freedesktop.ModemManager1.service -> /lib/systemd/system/ModemManager.service
lrwxrwxrwx  1 root root   53 Apr 28  2016 dbus-org.freedesktop.nm-dispatcher.service -> /lib/systemd/system/NetworkManager-dispatcher.
service
drwxr-xr-x  2 root root 4096 Jul 31  2016 default.target.wants/
lrwxrwxrwx  1 root root   35 Apr  7 19:23 display-manager.service -> /lib/systemd/system/lightdm.service
drwxr-xr-x  2 root root 4096 Feb 28 00:10 final.target.wants/
drwxr-xr-x  2 root root 4096 Oct 18 01:18 getty.target.wants/
drwxr-xr-x  2 root root 4096 Jun 22  2016 graphical.target.wants/
drwxr-xr-x  2 root root 4096 Apr 28  2016 hibernate.target.wants/
drwxr-xr-x  2 root root 4096 Apr 28  2016 hybrid-sleep.target.wants/
drwxr-xr-x  2 root root 4096 Feb 28 00:10 multi-user.target.wants/
drwxr-xr-x  2 root root 4096 Apr 28  2016 network-online.target.wants/
drwxr-xr-x  2 root root 4096 Feb 28 00:06 printer.target.wants/
drwxr-xr-x  2 root root 4096 Feb 28 00:10 sockets.target.wants/
lrwxrwxrwx  1 root root   31 Apr 28  2016 sshd.service -> /lib/systemd/system/ssh.service
drwxr-xr-x  2 root root 4096 Apr 28  2016 suspend.target.wants/
drwxr-xr-x  2 root root 4096 Feb 28 00:11 sysinit.target.wants/
drwxr-xr-x  2 root root 4096 Feb 28 00:10 timers.target.wants/
lrwxrwxrwx  1 root root    9 Nov  9 19:18 ufw.service -> /dev/null
root@adv-imx6:/etc/systemd/system# rm -rf display-manager.service
```

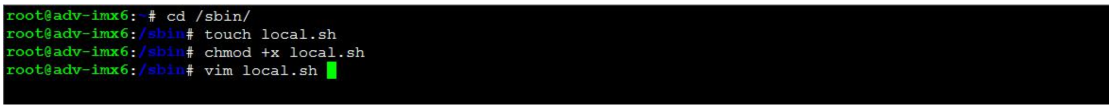
将lightdm.service文件删除掉

（2）Add the start Xorg command in /etc/rc.local

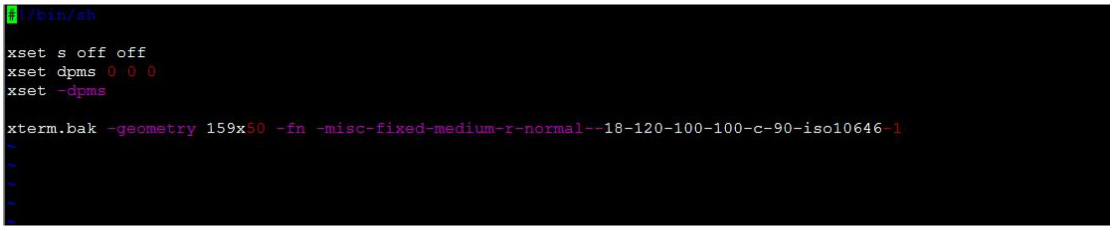


```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
xinit /sbin/local.sh -- /usr/bin/Xorg -br vt7 &
~
~
~
~
```

（3）Create a script in /sbin/local.sh to add a program that needs to be started



```
root@adv-imx6:~# cd /sbin/
root@adv-imx6:/sbin# touch local.sh
root@adv-imx6:/sbin# chmod +x local.sh
root@adv-imx6:/sbin# vim local.sh
```



```
#!/bin/sh

xset s off off
xset dpms 0 0 0
xset -dpms

xterm.bak -geometry 159x50 -fn -misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
~
~
~
~
~
```

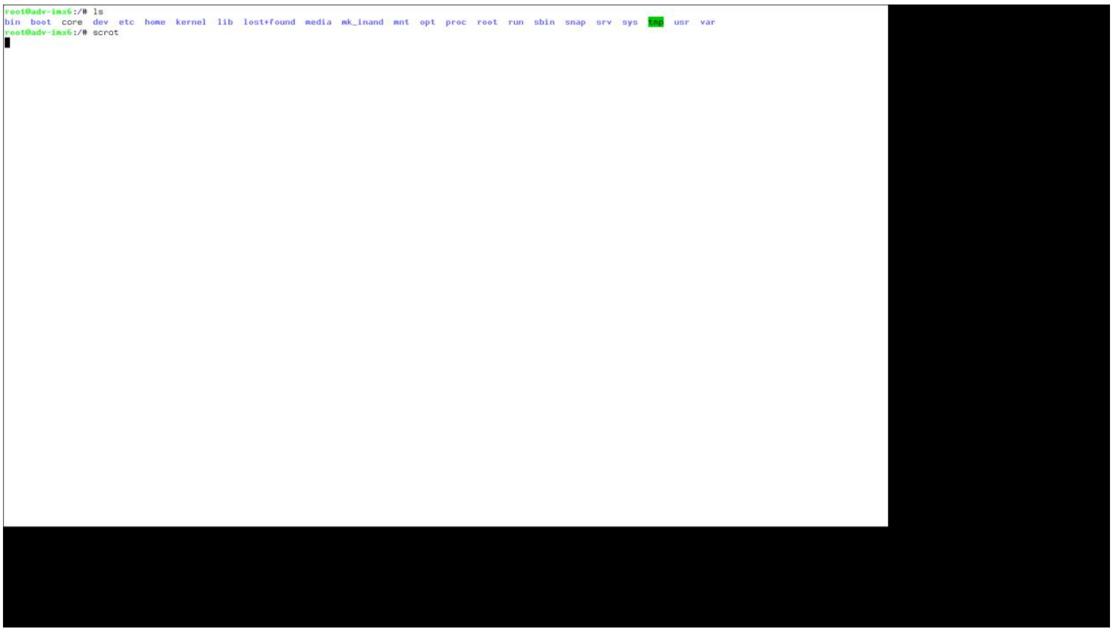
（4）Start the rc.local service file through Systemd

# systemctl enable rc-local.service

# systemctl restart rc-local.service

# reboot

（5）Effect of the program after booting



```
root@adv-imx6:/# ls
bin  boot  core  dev  etc  home  kernel  lib  lost+found  media  mk_inand  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@adv-imx6:/# scrot
```

5.6. Chinese Input Method Support

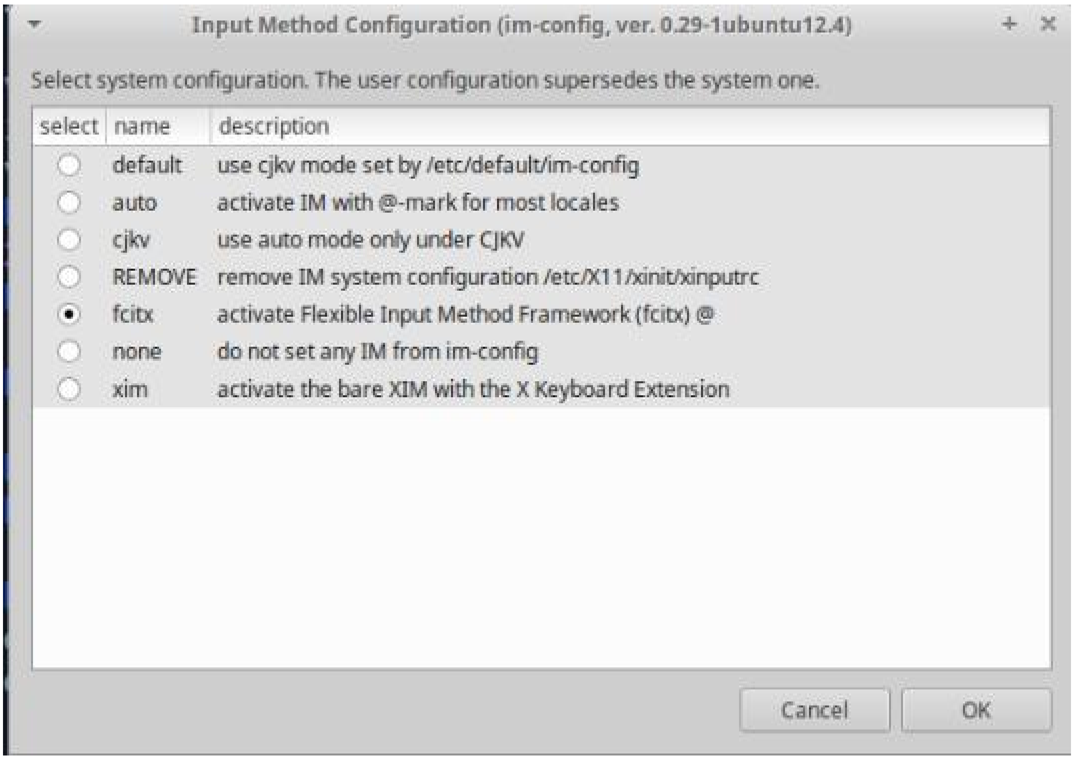
(1) Chinese input method support

Step1: Install Chinese installation package.

#apt-get install fcitx fcitx-pinyin im-config

Step2: Enter the following command in the terminal to open the input method configuration, and set fcitx as the default input method frame.

# im-config

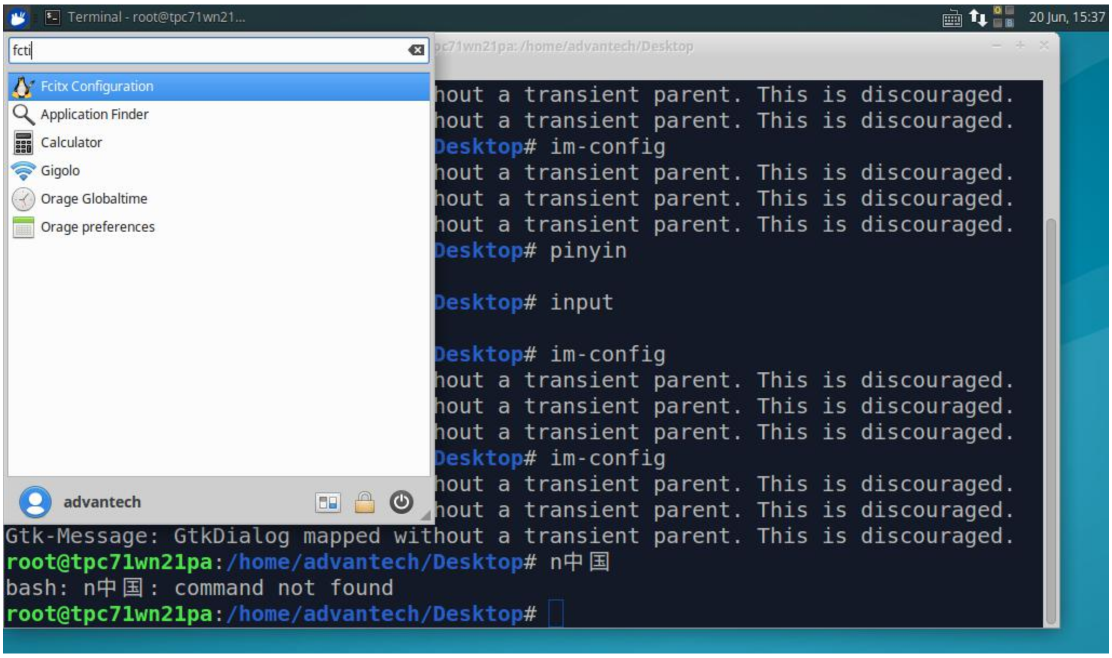


Input Method Configuration (im-config, ver. 0.29-1ubuntu12.4)

Select system configuration. The user configuration supersedes the system one.

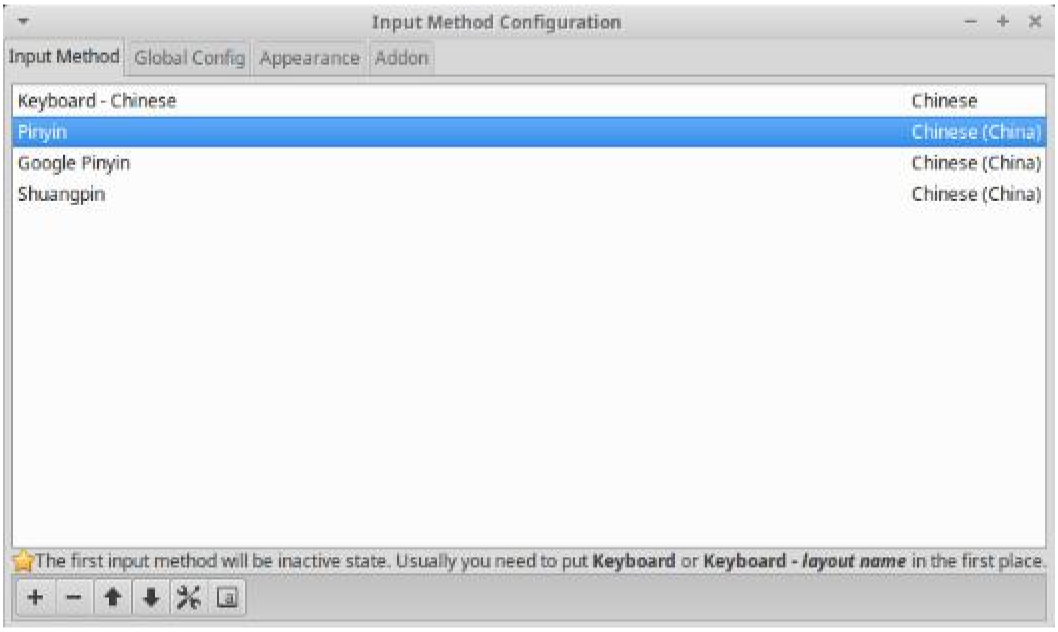
select	name	description
○	default	use cjkv mode set by /etc/default/im-config
○	auto	activate IM with @-mark for most locales
○	cjkv	use auto mode only under CJKV
○	REMOVE	remove IM system configuration /etc/X11/xinit/xinputrc
●	fcitx	activate Flexible Input Method Framework (fcitx) @
○	none	do not set any IM from im-config
○	xim	activate the bare XIM with the X Keyboard Extension

Step3: Click on the system menu and select Fctix Configuration.



```
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
Desktop# im-config
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
Desktop# pinyin

Desktop# input

Desktop# im-config
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
Desktop# im-config
hout a transient parent. This is discouraged.
hout a transient parent. This is discouraged.
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.
root@tpc71wn21pa:/home/advantech/Desktop# n中国
bash: n中国: command not found
root@tpc71wn21pa:/home/advantech/Desktop# 
```

Step4: Enter pinyin to find the pinyin input method and add it to the input method list.



Keyboard - Chinese	Chinese
Pinyin	Chinese (China)
Google Pinyin	Chinese (China)
Shuangpin	Chinese (China)

The first input method will be inactive state. Usually you need to put **Keyboard** or **Keyboard - *layout name*** in the first place.

(2) Set Chinese display in Qtcreater

Step1: Click Tools --> Option --> User Interface

Step2：Language --> Chinese（China）

Step3: Click "OK" to restart the software.

(3) Qt supports Chinese

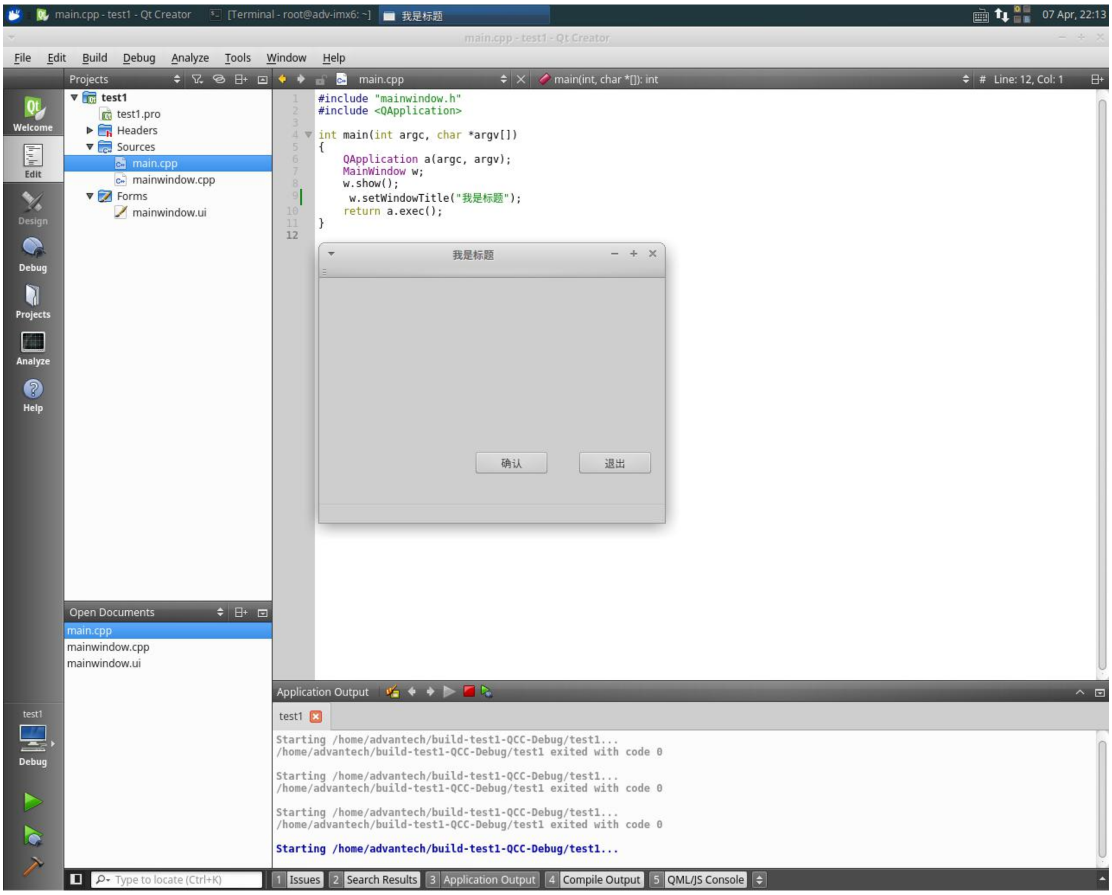
Step1: Install the Chinese input package supported by qt.

#apt-get install fcitx-libs-qt fcitx-libs-qt5

# cp -r

/usr/lib/arm-linux-gnueabihf/qt5/plugins/platforminputcontexts/libfcitxplatforminputcontextplugin.so /home/advantech/.config/QtProject/qtcreator/bin/plugins/platforminputcontexts/

Step2: Create a Qt project program



```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
     w.setWindowTitle("我是标题");
    return a.exec();
}
```

```
Starting /home/advantech/build-test1-QCC-Debug/test1...
/home/advantech/build-test1-QCC-Debug/test1 exited with code 0

Starting /home/advantech/build-test1-QCC-Debug/test1...
/home/advantech/build-test1-QCC-Debug/test1 exited with code 0

Starting /home/advantech/build-test1-QCC-Debug/test1...
/home/advantech/build-test1-QCC-Debug/test1 exited with code 0

Starting /home/advantech/build-test1-QCC-Debug/test1...
```

6. Ubuntu16.04 Development & Debugging

To develop ARM programs, the source code needs to be cross-compiled to run on ARM devices. We have a cross-compilation toolkit for TPC-71W, or developers can download it from the Internet.

In general, ARM application development steps are as follows

1) Install the gcc cross-compilation toolchain
2) Configuring environment variables
3) Write the program
4) Cross compilation and debugging
5) Copy or upload to ARM board to run.

6.1. Use the gcc Development Environment

We offer two ways to develop with arm gcc:

1) Develop on X86 machines
2) Develop directly on the TPC-71W ARM board

6.1.1. Gcc development and online debugging on X86 Linux machines

To use gcc for development on X86, you need to use the cross-compiler tool, then configure the environment variables to compile and debug.
This cross-compilation chain is for ITB-200 series devices. This example test was performed on the xubuntu16.04 system. When you get the compiler toolchain package, see the link to extract your development environment.

Online installation

Download the cross-compilation toolchain:

```
# apt-get  install  gcc-5-arm-linux-gnueabihf
```

Compile the binary file

```
# arm-linux-gnueabihf-gcc-5  demo.c
```

Upload or copy the generated binary file to the Arm development board and run.

Offline installation

Download the cross-compilation toolchain: https://pan.baidu.com/s/160B_JNyuNceX_RE2fBFoYg
Unzip:

```
# xz  -d  gcc-linaro-arm-linux-gnueabihf-4.9-2014.07_linux.tar.xz
# tar  -xvf  gcc-linaro-arm-linux-gnueabihf-4.9-2014.07_linux.tar
```

Configure environment variables:

```
# export  PATH=/gcc-linaro-arm-linux-gnueabihf-4.9-2014.07_linux/bin:$PATH
```

6.1.2. Developing with gcc on TPC-71W

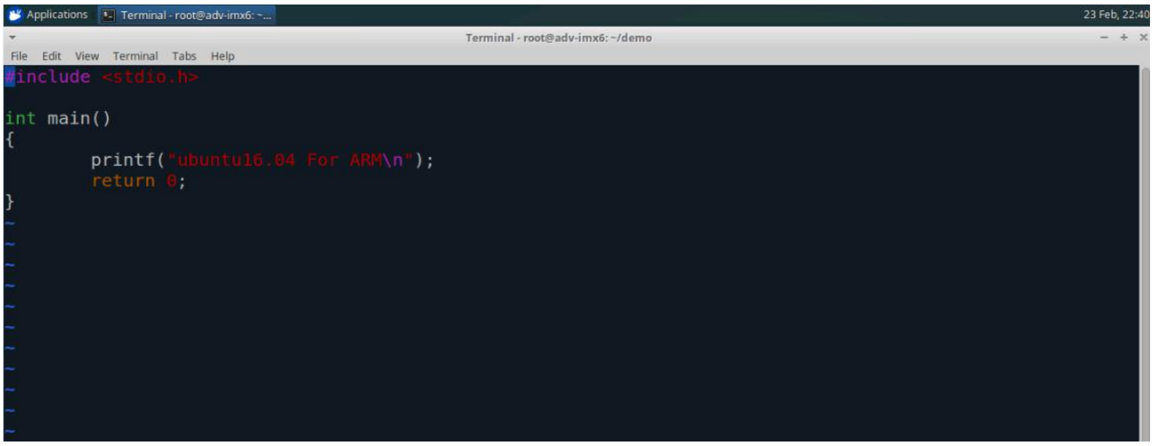
For the above mentioned developing with gcc on X86, you need to install the arm gcc cross-compiler and configure it. Online debugging is required during the development process. Compared with the previous x86 project development, it is slightly more complicated, and the debugging is slightly more troublesome. When getting familiar with this process, it should be very convenient.

The method introduced next is to use gcc for development and debugging directly in TPC-71W, which is completely consistent with the previous X86 development experience and use. You simply need to install the gcc compiler tool online and directly develop and debug it on the development board.

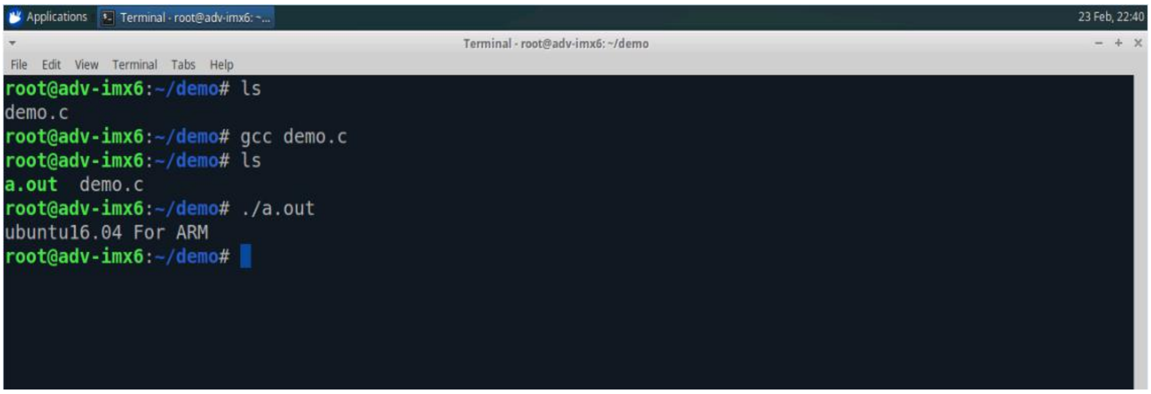
1. Install gcc compiler tool

```
# apt-get install gcc
```

2. Write a demo program


Applications
Terminal - root@adv-imx6: ~...
23 Feb, 22:40
Terminal - root@adv-imx6: ~/demo
File Edit View Terminal Tabs Help

```
#include <stdio.h>

int main()
{
        printf("ubuntu16.04 For ARM\n");
        return 0;
}
~
~
~
~
~
~
~
~
~
~
~
```


Applications
Terminal - root@adv-imx6: ~...
23 Feb, 22:40
Terminal - root@adv-imx6: ~/demo
File Edit View Terminal Tabs Help

```
root@adv-imx6:~/demo# ls
demo.c
root@adv-imx6:~/demo# gcc demo.c
root@adv-imx6:~/demo# ls
a.out  demo.c
root@adv-imx6:~/demo# ./a.out
ubuntu16.04 For ARM
root@adv-imx6:~/demo# 
```

Although developing and debugging on ARM is very convenient, the performance is relatively poor. If it is a large program, the debugging will be slower. Thus, the first method is recommended, which is to develop on X86 machines.

6.2. QtCreator Integrated Development Environment

If customers need to develop HMI programs with graphical interfaces, it is recommended to use Qt.

Similarly, as with GCC development, there are two ways to use qt-creator, either on X86 machine development or directly on TPC-71W.

6.2.1. Developing and Debugging with Qt Creator on X86 Linux Device

Qt Creator is a cross-platform Qt IDE that facilitates the development and debugging of QT applications. It also supports remote debugging of applications, which is very convenient for porting QT applications in ARM. This section describes how to configure Qt Creator and debug QT demo online.

(1) Download and install QT Creator.

Can be downloaded from Qt official website

If the development host uses Ubuntu, you can also use the apt-get command to install it online.

#apt-get install qtcreator

(2) To develop ARM programs, QT Creator needs to configure cross-compilation tools. The specific QtCreator reference links are as follows:

http://ess-wiki.advantech.com.tw/view/IoTGateway/BSP/Linux/iMX6/QtCreator

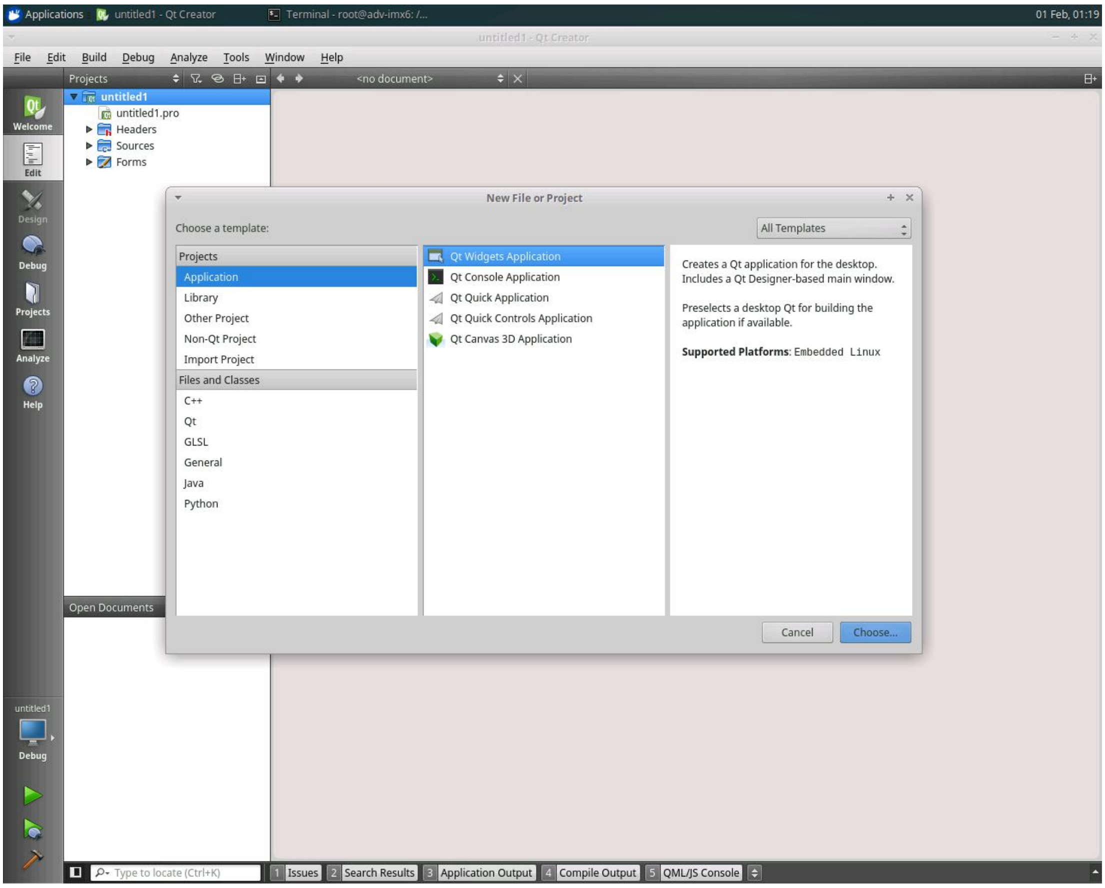
6.2.2. Developing and Debugging with Qt Creator on TPC-71W

We can also develop graphics programs based on Qt-creator directly on the TPC-71W.

- First, you need to install the qtcreator development environment and related development kits online:

```
#apt-get install qtcreator
```

- Run Qt-creator development environment and write Qt program case.
  a) Follow the wizard to build a project, File - -> New File or Project - -> Choose

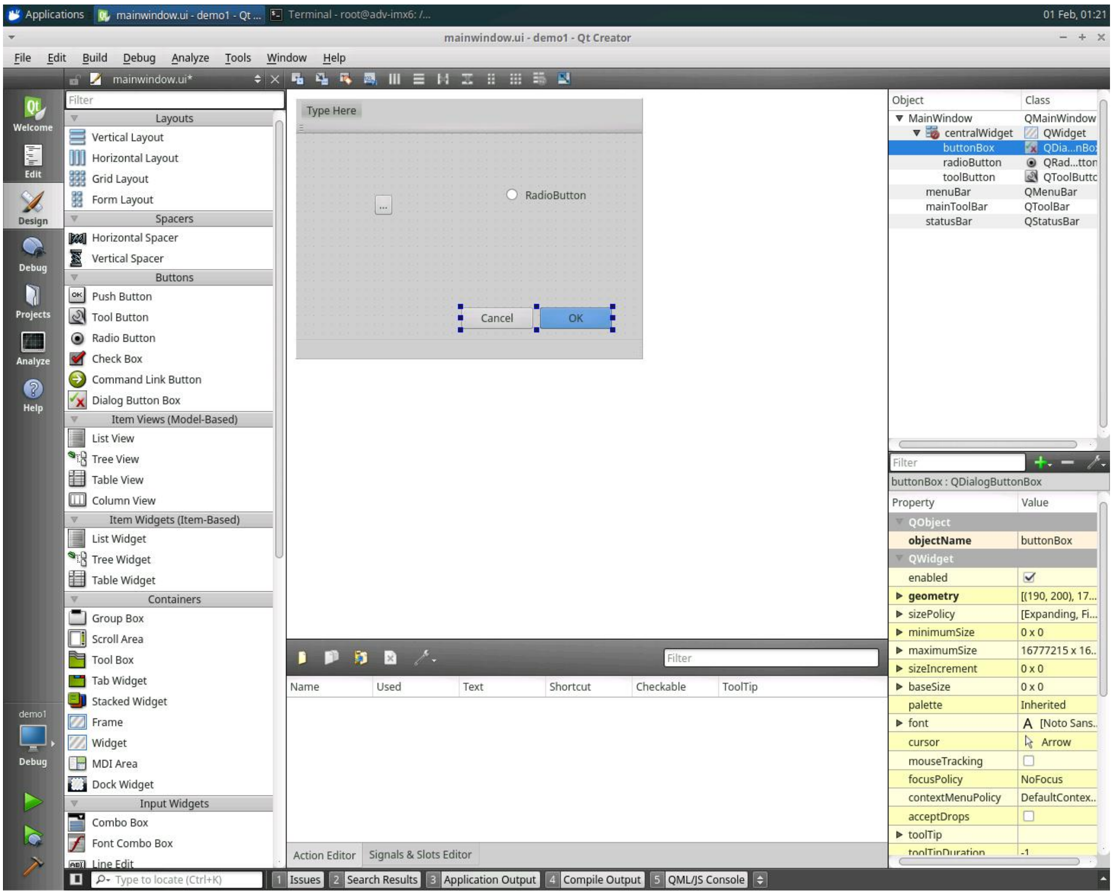


Creates a Qt application for the desktop. Includes a Qt Designer-based main window.

Preselects a desktop Qt for building the application if available.

Supported Platforms: Embedded Linux

  b) Design a graphical interface

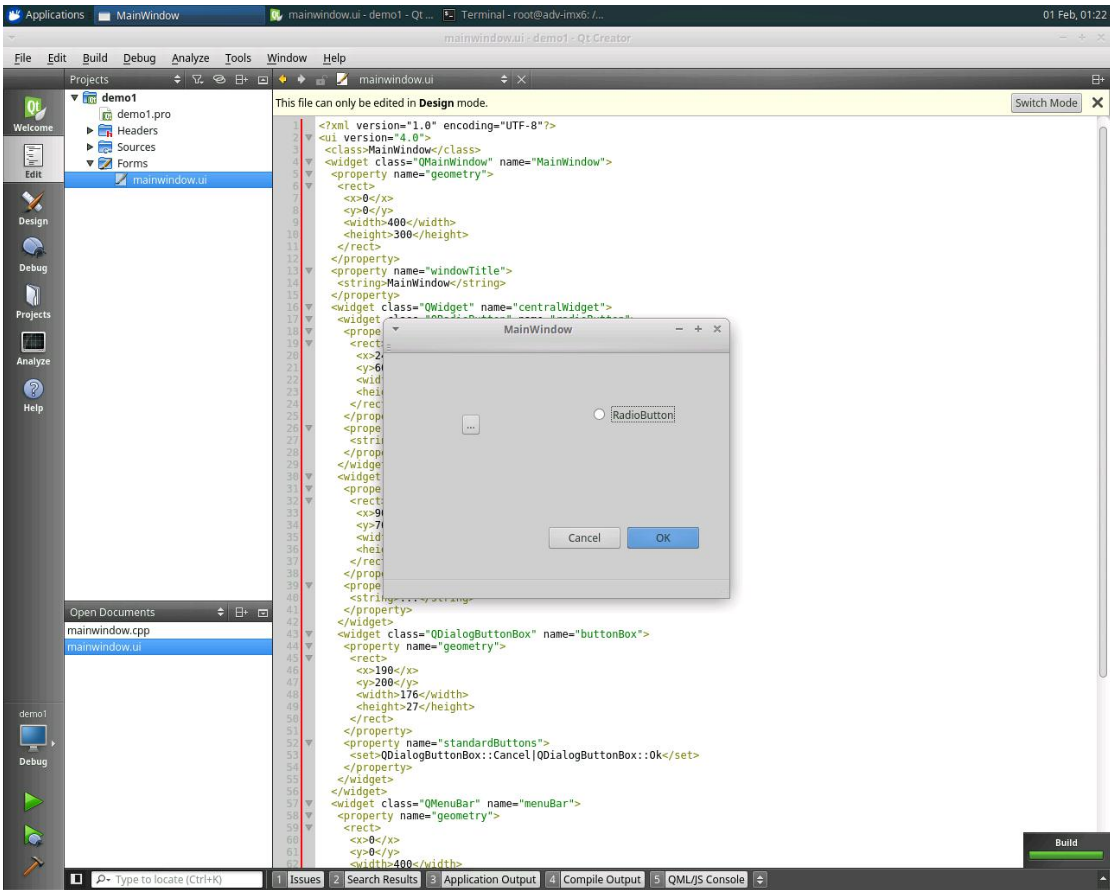


Object	Class
MainWindow	QMainWindow
centralWidget	QWidget
buttonBox	QDia...nBox
radioButton	QRad...tton
toolButton	QToolButto
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

buttonBox : QDialogButtonBox

Property	Value
QObject	
objectName	buttonBox
QWidget	
enabled	☑
geometry	[(190, 200), 17...
sizePolicy	[Expanding, Fi...
minimumSize	0 x 0
maximumSize	16777215 x 16..
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Inherited
font	A [Noto Sans..
cursor	Arrow
mouseTracking	□
focusPolicy	NoFocus
contextMenuPolicy	DefaultContex..
acceptDrops	□
toolTip	
toolTipDuration	-1

c) Run results directly on the TPC-71W



This file can only be edited in **Design** mode.

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>400</width>
    <height>300</height>
   </rect>
  </property>
  <property name="windowTitle">
   <string>MainWindow</string>
  </property>
  <widget class="QWidget" name="centralWidget">
   <widget
    <prope
     <rect
      <x>2
      <y>6
      <wid
      <hei
     </rec
    </prop
    <prope
     <stri
    </prop
   </widge
   <widget
    <prope
     <rect
      <x>9
      <y>7
      <wid
      <hei
     </rec
    </prop
    <prope
     <string>...</string>
    </property>
   </widget>
   <widget class="QDialogButtonBox" name="buttonBox">
    <property name="geometry">
     <rect>
      <x>190</x>
      <y>200</y>
      <width>176</width>
      <height>27</height>
     </rect>
    </property>
    <property name="standardButtons">
     <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
    </property>
   </widget>
  </widget>
  <widget class="QMenuBar" name="menuBar">
   <property name="geometry">
    <rect>
     <x>0</x>
     <y>0</y>
     <width>400</width>
```

6.2.3. Qt Sample Program and Source Code Description

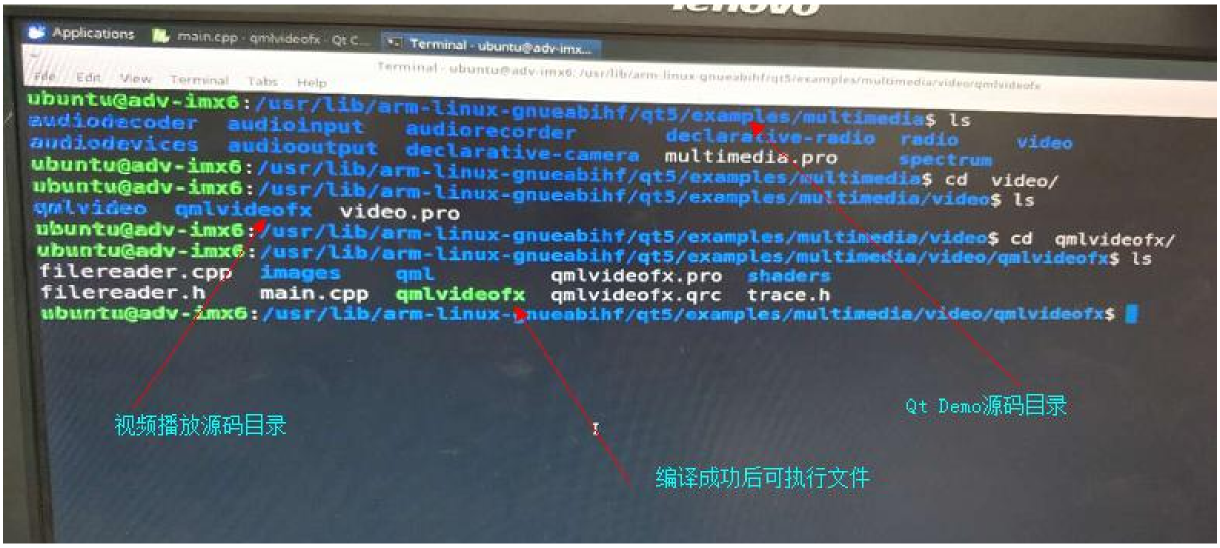
By default, the ARM Ubuntu system also provides some Qt sample programs for users to test the serial port, play audio and video, and so on. The Demo program is located in the system desktop Demo directory (/home/ubuntu/Desktop/Demo) and the source code is located under /opt.

(1) Video playback sample program and source code

Step1: Install qt video playback runtime

# apt-get install qtmultimedia5-dev qtdeclarative5-dev qtmultimedia5-examples

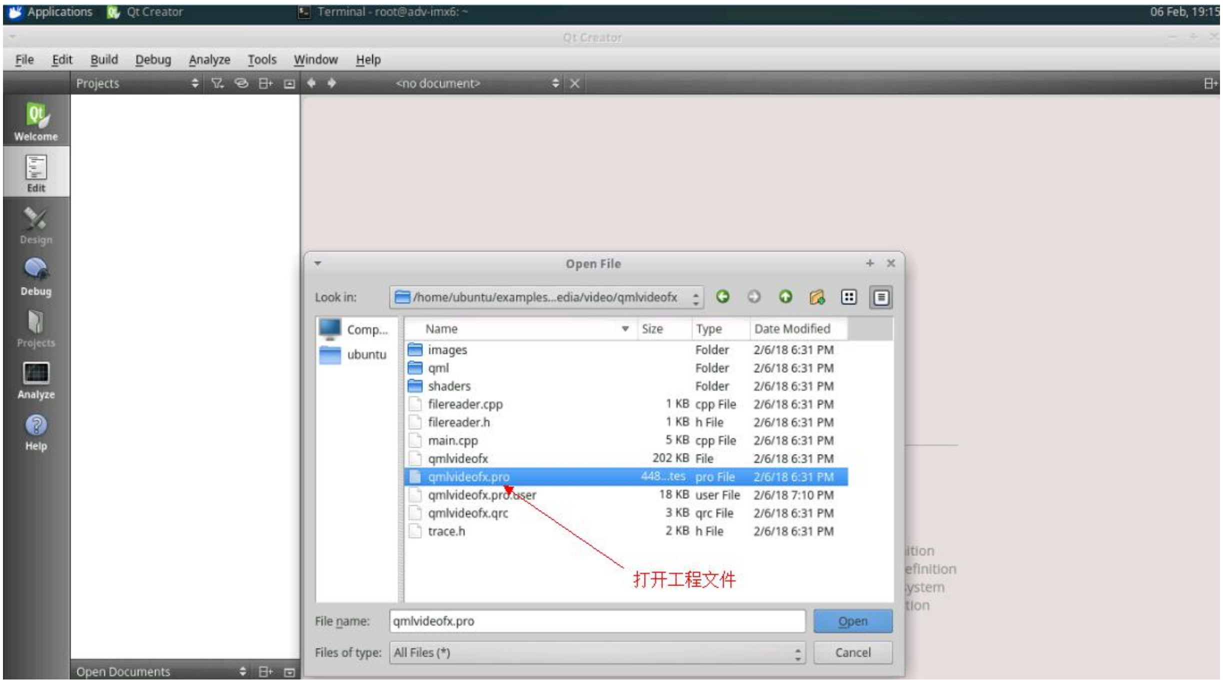
Step2: The source directory is placed in the /usr/lib/arm-linux-gnueabihf/qt5/examples/ directory.



Applications
main.cpp - qmlvideofx - Qt C...
Terminal - ubuntu@adv-imx...
Terminal - ubuntu@adv-imx6: /usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia/video/qmlvideofx
File Edit View Terminal Tabs Help

```
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia$ ls
audiodecoder   audioinput   audiorecorder            declarative-radio  radio     video
audiodevices   audiooutput  declarative-camera  multimedia.pro     spectrum
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia$ cd  video/
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia/video$ ls
qmlvideo  qmlvideofx  video.pro
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia/video$ cd  qmlvideofx/
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia/video/qmlvideofx$ ls
filereader.cpp  images    qml         qmlvideofx.pro  shaders
filereader.h    main.cpp  qmlvideofx  qmlvideofx.qrc  trace.h
ubuntu@adv-imx6:/usr/lib/arm-linux-gnueabihf/qt5/examples/multimedia/video/qmlvideofx$
```

Step3: Qtcreator import video playback Demo project



Look in: /home/ubuntu/examples...edia/video/qmlvideofx

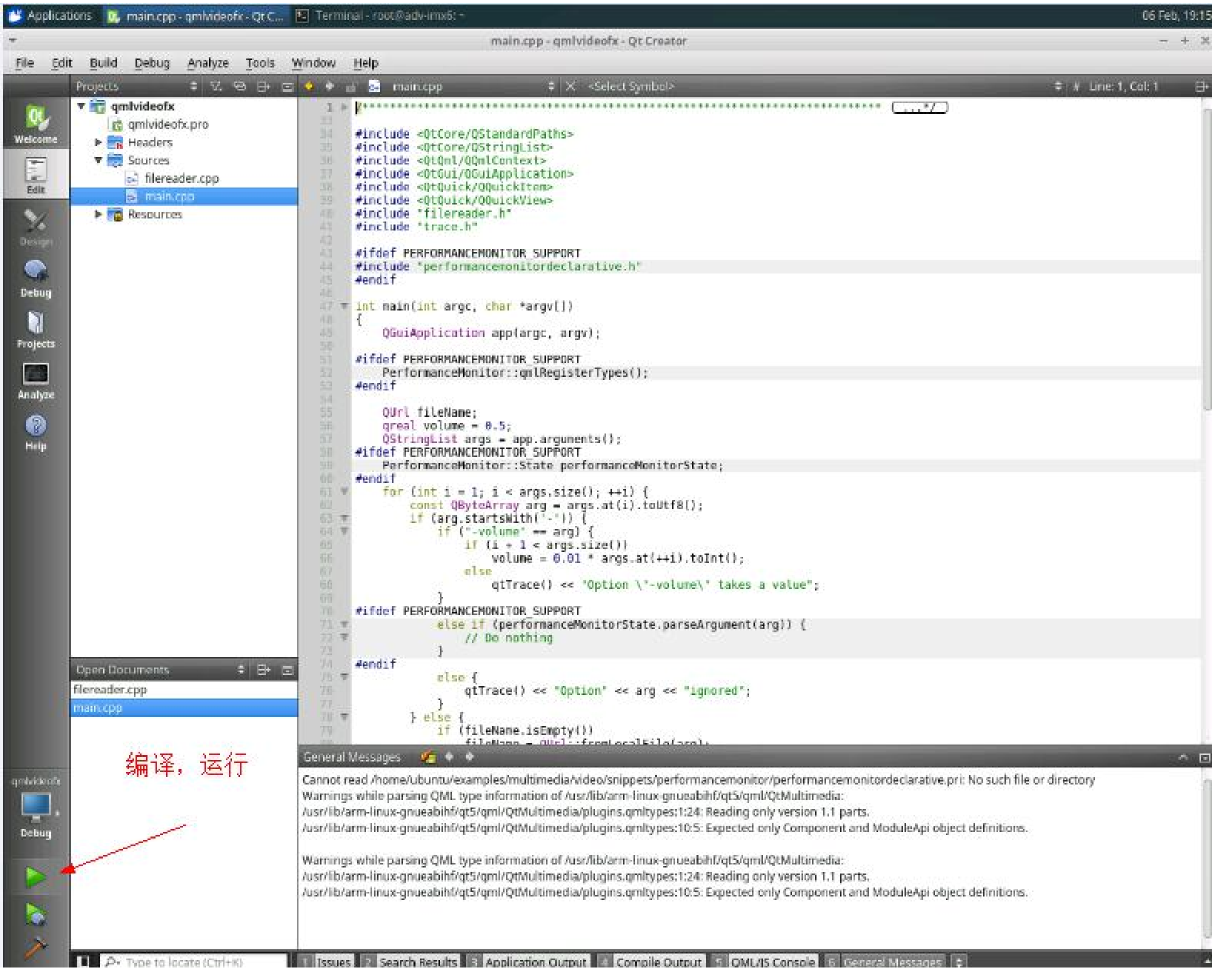
Name	Size	Type	Date Modified
images		Folder	2/6/18 6:31 PM
qml		Folder	2/6/18 6:31 PM
shaders		Folder	2/6/18 6:31 PM
filereader.cpp	1 KB	cpp File	2/6/18 6:31 PM
filereader.h	1 KB	h File	2/6/18 6:31 PM
main.cpp	5 KB	cpp File	2/6/18 6:31 PM
qmlvideofx	202 KB	File	2/6/18 6:31 PM
qmlvideofx.pro	448...tes	pro File	2/6/18 6:31 PM
qmlvideofx.pro.user	18 KB	user File	2/6/18 7:10 PM
qmlvideofx.qrc	3 KB	qrc File	2/6/18 6:31 PM
trace.h	2 KB	h File	2/6/18 6:31 PM

File name: qmlvideofx.pro

Files of type: All Files (*)

打开工程文件

Step4: Compile and run



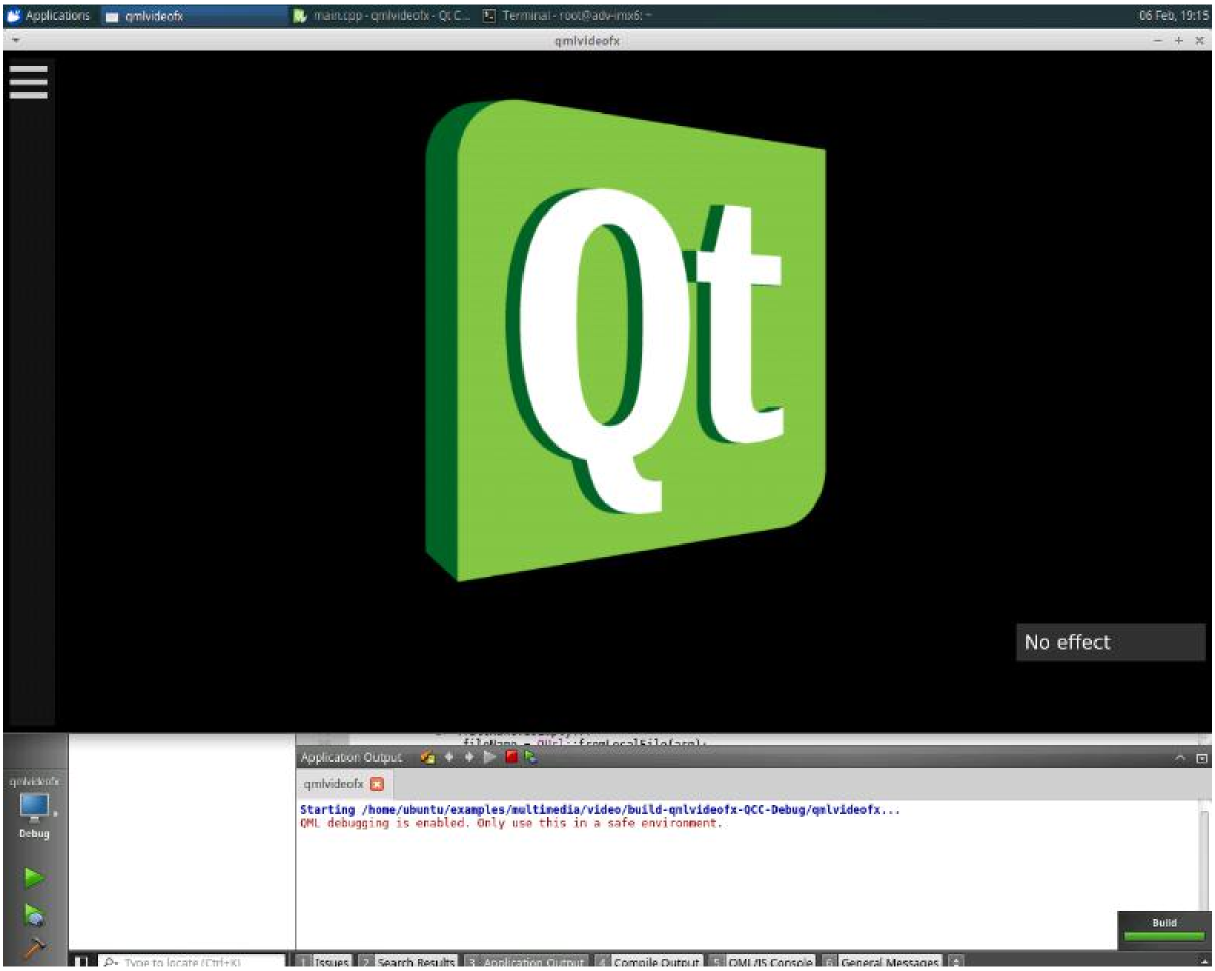
```
/****************************************************************************
  
#include <QtCore/QStandardPaths>
#include <QtCore/QStringList>
#include <QtQml/QQmlContext>
#include <QtGui/QGuiApplication>
#include <QtQuick/QQuickItem>
#include <QtQuick/QQuickView>
#include "filereader.h"
#include "trace.h"

#ifdef PERFORMANCEMONITOR_SUPPORT
#include "performancemonitordeclarative.h"
#endif

int main(int argc, char *argv[])
{
    QGuiApplication app(argc, argv);

#ifdef PERFORMANCEMONITOR_SUPPORT
    PerformanceMonitor::qmlRegisterTypes();
#endif

    QUrl fileName;
    qreal volume = 0.5;
    QStringList args = app.arguments();
#ifdef PERFORMANCEMONITOR_SUPPORT
    PerformanceMonitor::State performanceMonitorState;
#endif
    for (int i = 1; i < args.size(); ++i) {
        const QByteArray arg = args.at(i).toUtf8();
        if (arg.startsWith('-')) {
            if ("-volume" == arg) {
                if (i + 1 < args.size())
                    volume = 0.01 * args.at(++i).toInt();
                else
                    qtTrace() << "Option \"-volume\" takes a value";
            }
#ifdef PERFORMANCEMONITOR_SUPPORT
            else if (performanceMonitorState.parseArgument(arg)) {
                // Do nothing
            }
#endif
            else {
                qtTrace() << "Option" << arg << "ignored";
            }
        } else {
            if (fileName.isEmpty())
```

General Messages

```
Cannot read /home/ubuntu/examples/multimedia/video/snippets/performancemonitor/performancemonitordeclarative.pri: No such file or directory
Warnings while parsing QML type information of /usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia:
/usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia/plugins.qmltypes:1:24: Reading only version 1.1 parts.
/usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia/plugins.qmltypes:10:5: Expected only Component and ModuleApi object definitions.

Warnings while parsing QML type information of /usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia:
/usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia/plugins.qmltypes:1:24: Reading only version 1.1 parts.
/usr/lib/arm-linux-gnueabihf/qt5/qml/QtMultimedia/plugins.qmltypes:10:5: Expected only Component and ModuleApi object definitions.
```

编译，运行

Step5: The result



```
Starting /home/ubuntu/examples/multimedia/video/build-qmlvideofx-GCC-Debug/qmlvideofx...
QML debugging is enabled. Only use this in a safe environment.
```

(2) 5.4.2 Audio playback sample program and source code

Step1: The source code is located in /usr/lib/arm-linux-gnueabihf/qt5/examples/

Step2: For compilation steps, please refer to Section 5.4.1 sample program and source code

(3) Operating serial port

Please refer to: http://doc.qt.io/qt-5/qtserialport-examples.html

6.3. Introduction to the Java Development Environment

Java installation and configuration

Step1: Update the system installation package cache and install OpenJDK8

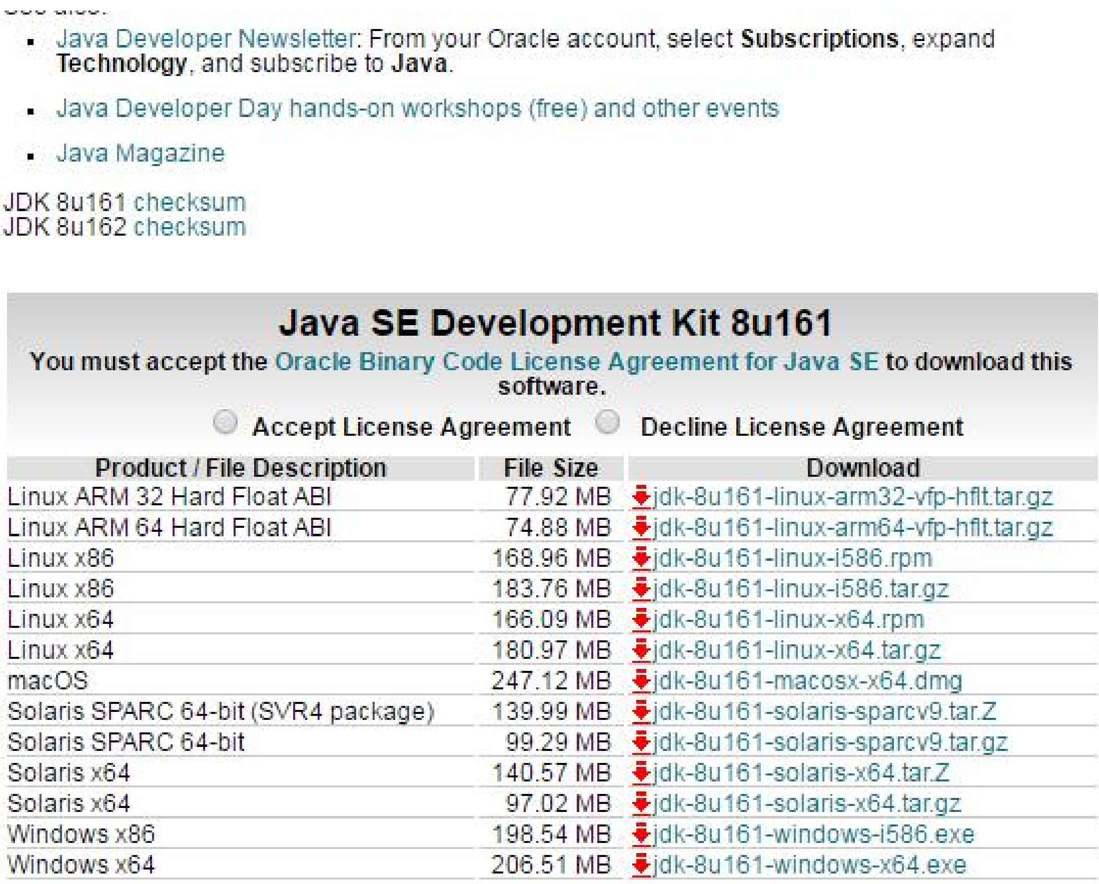
# apt-get update

# apt-get install openjdk-8-jdk

Step2: Download the java installation package manually.

Download link:

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html



- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u161 checksum
JDK 8u162 checksum

Java SE Development Kit 8u161

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ○ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	jdk-8u161-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.88 MB	jdk-8u161-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.96 MB	jdk-8u161-linux-i586.rpm
Linux x86	183.76 MB	jdk-8u161-linux-i586.tar.gz
Linux x64	166.09 MB	jdk-8u161-linux-x64.rpm
Linux x64	180.97 MB	jdk-8u161-linux-x64.tar.gz
macOS	247.12 MB	jdk-8u161-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	jdk-8u161-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.29 MB	jdk-8u161-solaris-sparcv9.tar.gz
Solaris x64	140.57 MB	jdk-8u161-solaris-x64.tar.Z
Solaris x64	97.02 MB	jdk-8u161-solaris-x64.tar.gz
Windows x86	198.54 MB	jdk-8u161-windows-i586.exe
Windows x64	206.51 MB	jdk-8u161-windows-x64.exe

Step3: Extract the downloaded jdk-8u161-linux-arm32-vfp-hflt.tar.gz.

```
# tar -zxvf jdk-8u161-linux-arm32-vfp-hflt.tar.gz
# mv jdk-8u161-linux-arm32-vfp-hflt/     /opt/
```

Step4: Configure environment variables

Edit .bashrc file

```
# export JAVA_HOME=/opt/jdk1.8.0_161
# export JRE_HOME=$JAVA_HOME/jre
# export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib
# export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
```

Step2: check Java installation

# java   -version (The following output information indicates that the installation was successful.)

```
java version “1.8.0_161”
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) Client VM (build 25.161-b12， mixed mode)
```

6.4. Mono Configuration

Please refer to: http://www.mono-project.com/docs/

7. Backup and Deployment of Secondary Developed Custom Systems

For specific applications, developers will need to re-develop and customize the systems we provide to meet the needs of specific application scenarios. However, how to back up the system and deploy other machines in batches after the customer completes the secondary development customization on the TPC-71W? This section gives the corresponding method:

When you have installed the system on emmc, integrated the application, finished the debugging, and need to back up the system for bulk deployment, you can use the backup script program provided by us to perform simple backup and deployment.

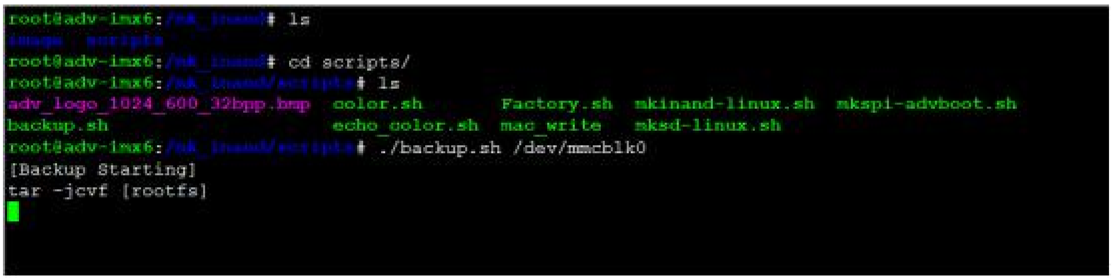
Note: This is to back up the system on emmc to SD, and then burn the emmc to other devices through SD, so the secondary development system is on emmc.

7.1. Backup Target System Image

1) Prepare an SD in advance and burn the original ARM Ubuntu system we provided, and boot it from the SD card.

   Note: This system uses the system image we have burned to SD.

2) After the system starts, enter the (/mk_inand/scripts/) directory



```
root@adv-imx6:/mk_inand# ls
image  scripts
root@adv-imx6:/mk_inand# cd scripts/
root@adv-imx6:/mk_inand/scripts# ls
adv_logo_1024_600_32bpp.bmp  color.sh       Factory.sh  mkinand-linux.sh  mkspi-advboot.sh
backup.sh                    echo_color.sh  mac_write   mksd-linux.sh
root@adv-imx6:/mk_inand/scripts# ./backup.sh /dev/mmcblk0
[Backup Starting]
tar -jcvf [rootfs]
█
```

3) Wait for the backup to complete before performing the synchronization command
4) Run the poweroff command to shut down, and then pull out the SD card.

7.2. Deploy to Other Devices

At this point, the latest system has been backed up in the SD card, and can be burned to other TPC-71W devices. The specific burning steps are the same as the previous Ubuntu burning to emmc, please refer to section 3.5.

8. Value-added Customized Solution

Advantech has a corresponding solution for the following solutions. However, these programs require special customization, and some of them require payment. Please contact the product PM for further information. .

8.1. Remote OTA Update

WISE-PaaS/OTA can update Ubuntu systems or applications remotely.

8.2. Cross-network Remote Desktop

By default, VNC can only use remote desktops in the LAN. By customization, it can support remote desktops across network segments.

8.3. System Backup

Back up your system to restore to the original version if the system is damaged.

8.4. Read-only File System

To ensure power-off reliability, a read-only file system scheme is provided to ensure that the system partition is read-only and user data is recorded to another readable and writable partition.