WISE-710 Ubuntu 16.04
用户手册
V1.0

目录

1. ARM Ubuntu16.04 概述

1.1. 跨平台性，X86 和 ARM 平台使用体验一致

Ubuntu 是最受欢迎的 Linux 发行版之一，目前很多工控应用方案也都是基于 Ubuntu 平台的，在 ARM 上运行 Ubuntu，对工业应用开发和维护来说，更容易，更方便，更快捷，

更友好，对于已有的 X86 平台的应用方案，也可以更加快速和无缝的移植到 ARM Ubuntu 平台，同时降低开发难度，成本和风险。

1.2. 软件支持丰富

Ubuntu 官方提供了非常丰富的 ARM 版本软件包，工控中常用的 ssh,telnet,ftp,qt 库，视屏播放软件，mysql 数据库软件，Java，VNC 等等，都有现成的软件包，客户需求时这些组件时，不用去交叉编译和移植，和 x86 一样，直接通过 apt 命令在线安装就可以。

1.3. 快速定制系统

正是 Ubuntu ARM 版本提供了非常丰富的软件包，客户可以根据自身项目需求，快速安装所需软件包，快速进行相关配置，客制出所需的系统，而且，对于 X86 Ubuntu 经验的开发人员来说，不用花费时间进行进行 ARM Linux 的开发学习，因为开发技术和方式完全相同。

1.4. 快速开发和维护

Ubuntu 除了提供丰富的软件支持，在开发方面也非常成熟，资源非常多。可以支持 gcc,qt,java,python,mono,php 等非常多的开发环境， 并提供了相关软件开发工具，客户可以根据需要，选择熟悉和合适的开发语言，环境和方案，加速软件开发。

1.5. 图形化的系统设定

Ubuntu ARM 版本支持 XFCE 轻量级桌面环境，可以在 ARM 上顺畅运行，很多系统配置，如网路配置，3G/4G 配置等，都可以通过图形方式配置，非常方便，快速和简单。

1.6. 有长期支持版本

Ubuntu 有商业公司 Canonical 进行开发和维护，稳定性和可靠性值得信赖，同时 Ubuntu 提供长期支持版本， Ubuntu 16.04 LTS 会提供 3~5 年以上的支持和更新，很好满足工控客户的长期支持需求。

1.7. 研华提供工控定制 ARM Ubuntu 版本

研华提供的 ARM Ubuntu 版本，基于 ARM Ubuntu 16.04，针对 WISE-710 进行了适配，针对工控客户常见的需求，提供的以下方面的客制优化：

1) 提供各种针对硬件的测试程序和测试说明，方便客户对硬件进行测试验证，了解使用方法；
2) 提供示例程序及源码，如串口通信，视频播放等，方便客户开发时参考；
3) 提供外围 WIFI/4G 等无线模块支持，内置相关驱动，方便客户建立无线方案；

注意： Ubuntu 系统进行商业使用，是需要付费的，需要得到 Ubuntu 官方的许可，请联系 Ubuntu 官方，进一步了解情况，也可以联系研华产品 PM，研华和跟 Ubuntu 官方有建立沟通合作渠道。

2. 研华 Ubuntu Image 软件支持

2.1. 系统内置软件

Image 默认支持的很多工业用户常用的软件，

- Chromium 浏览器
- 视屏播放软件
- 音频播放软件
- ……

2.2. 通过 apt 命令在线安装软件包

如果需要的软件没有包含在研华提供的 Ubuntu Image 中，ARM Ubuntu 系统同样提供了 APT (Advanced Package Tool,高级软件包工具)包管理机制，在设备联网的情况的，可以直接通过 APT 相关命令在线查询和安装软件，APT 会自动处理依赖关系并在系统安装所要的软件包。

Ubuntu 提供了非常丰富的 ARM 版软件包，可以说，绝大部分客户所需的软件，都可以通过 apt 命令直接安装，无需从源码进行交叉编译，非常方便。

apt 命令可以通过源配置文件自动找到 Ubuntu 的软件服务器，从服务请上下载软件，Image 已默认添加了 Ubuntu 官方的镜像源，所以一般用户不用重新去修改设置。

但如果有有些软件 Ubuntu 官方源中并不存在，但有第三方的 Ubuntu 软件源可以提供，用户可以先修改软件源的配置，增加第三方的软件源，在进行软件的在线安装即可。

Ubuntu16.04 系统的镜像源在/etc/apt/source.list。

Step1：编辑 source.list 文件添加新的镜像源。

Step2：添加完成后执行命令 apt-get update 更新镜像源。

2.3. 常用软件在线安装

2.3.1. apt 命令用法如下：

- 安装软件包：

  # sudo apt-get install packagename
- 删除软件包：

  # sudo apt-get remove packagename
- 获取新的软件包列表：

  # sudo apt-get update

- 升级有可用更新的系统：
  # sudo apt-get upgrade
- 查询所需软件包：
  #apt-cache search packagename
- 列出更多命令和选项：
  # apt-get help

更多关于 apt 的用法，可以上网查询相关资料，进行更加深入的了解。
下面列出了一些工控用户常用软件包的安装：

2.3.2. 安装 Chromium 浏览器 （已内置）

 # apt-get install chromium-browser

2.3.3. 安装数据库软件 mysql

 # apt-get install mysql-server

2.3.4. 安装 Apache web 服务器

 # apt-get install apache2 apache2-dev

2.3.5. 安装 PHP

 # apt-get install php

2.3.6. 安装 Python

 # apt-get install python

2.3.7. 安装 QtCreator

 # apt-get install qtcreator

2.3.8. 安装 SSH（已内置）

 # apt-get install openssh-server

2.3.9. 安装 VNC（已内置）

 # apt-get install x11vnc

2.3.10. 安装 Mono

如果采用默认方式安装：
sudo apt-get install mono-complete

Ubuntu 目前提供的默认版本是 mono 4.0
如果客户希望使用更新的 mono 5.2,就可以通过之前提供的方法，修改软件源配置文件，添加 Mono 官方的软件源，可以通过一下命令修改软件源：
#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
这样就安装了 mono 5.2 版本。

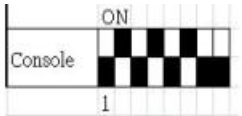
3. Ubuntu16.04 系统烧录和启动

3.1. 所需软硬件环境

- WISE-710
- RS232 串口线
- X86 开发主机安装 Ubuntu16.04 系统（WISE-710 Ubuntu16.04 设备也可以）
- WISE-710 Ubuntu Image
- SD 卡（大于等于 8G）

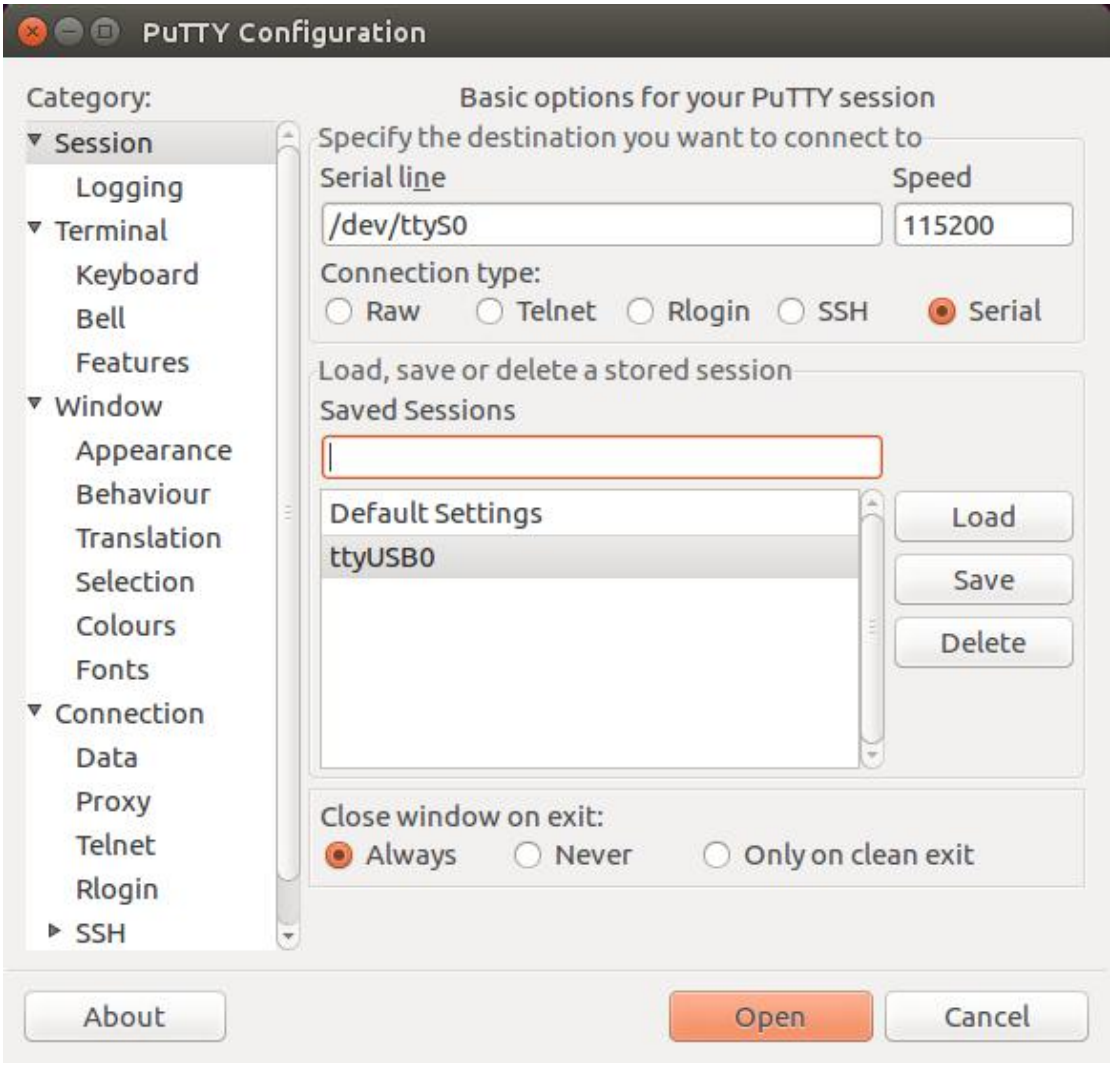
3.2. 调试接口设定

(1) 连接 RS232 串口线，
设置 WISE-710 的 SW9 拨码开关，把 COM1 切换到 debug 模式。

Console Mode	Bit 2,4,6 ON Bit 1,3,5,7,8 OFF	

RS-232 串行电缆到 WISE-710 COM 口，连接串行电缆的另一端到你的主机。
(2) 安装 Linux 下串口调试工具 putty。
 # apt-get install putty
如果是 Windows 环境，也可以从网络上下载 Windows 版本 putty 进行安装

(3) 主机打开 putty 工具设置串口号，波特率 115200。
请使用 root 用户打开 putty，串口号是 debug 先连接的串口。比如 Linux 下，如果连接的是第一个串口（如上图），可能是/dev/ttyS0 ，如果使用的是 USB 转串口线，可能是/dev/ttyUSB0.



putty 配置

(4) 连接电源，从 SD 卡或板载存储启动 WISE-710。从 bootloader 阶段开始,调试信息会被输出到 putty 窗口。

注意: 如果你使用的 WISE-710 还没有系统，请参考下面章节来制作一张可以启动的系统 SD 卡，并可以通过 SD 将，将系统烧录到板载 emmc.

3.3. Ubuntu Image 烧录准备

3.4. 使用 SD 卡启动 WISE-710

使用 SD 卡启动 WISE-710，进入系统。

Linux 端：

1. 在 linux 系统下解压 Ubuntu Image 压缩包

```
root@ubuntu:/home/work/wise-710/release/2019-06-05-Release3-ubuntu-A101-3# ls
WISE-710-r3-ubuntu.20190605.tar.gz  WISE-710-r3-ubuntu.20190605.tar.gz.md5  WISE-710-r3-ubuntu.20190605.tar.gz.sha256
root@ubuntu:/home/work/wise-710/release/2019-06-05-Release3-ubuntu-A101-3# tar -zxvf WISE-710-r3-ubuntu.20190605.tar.gz
WISE-710-r3-ubuntu.20190605/
WISE-710-r3-ubuntu.20190605/scripts/
WISE-710-r3-ubuntu.20190605/scripts/mkinand-linux.sh
WISE-710-r3-ubuntu.20190605/scripts/Factory-final.sh
WISE-710-r3-ubuntu.20190605/scripts/mac_write_linux
WISE-710-r3-ubuntu.20190605/scripts/touch_fa.sh
WISE-710-r3-ubuntu.20190605/scripts/etp_write
WISE-710-r3-ubuntu.20190605/scripts/mksd_recovery-linux.sh
WISE-710-r3-ubuntu.20190605/scripts/Factory-linux.sh
WISE-710-r3-ubuntu.20190605/scripts/mkspi-advboot.sh
WISE-710-r3-ubuntu.20190605/scripts/touch_ecc.sh
WISE-710-r3-ubuntu.20190605/scripts/hostname_write.sh
WISE-710-r3-ubuntu.20190605/image/
WISE-710-r3-ubuntu.20190605/image/zImage
WISE-710-r3-ubuntu.20190605/image/SPL
WISE-710-r3-ubuntu.20190605/image/adv_logo_1024x600_32bpp.bmp
WISE-710-r3-ubuntu.20190605/image/u-boot.imx
WISE-710-r3-ubuntu.20190605/image/imx6dl-wise710-a1.dtb
WISE-710-r3-ubuntu.20190605/image/ubuntu16044.tar.gz
WISE-710-r3-ubuntu.20190605/image/u-boot_crc.bin
WISE-710-r3-ubuntu.20190605/image/8111g-cfg/
```

2. 得到 ubuntu image 文件，将 SD 插入到主机之后进行 dd 烧录操作

```
root@ubuntu:/home/work/wise-710/release/2019-06-05-Release3-ubuntu-A101-3/WISE-710-r3-ubuntu.20190605/scripts# ls ../image/
8111g-cfg  adv_logo_1024x600_32bpp.bmp  imx6dl-wise710-a1.dtb  SPL  u-boot_crc.bin  u-boot_crc.bin.crc  u-boot.imx  ubuntu16044.tar.gz  zImage
root@ubuntu:/home/work/wise-710/release/2019-06-05-Release3-ubuntu-A101-3/WISE-710-r3-ubuntu.20190605/scripts# ls
etp_write         Factory-linux.sh   mac_write_linux   mksd_recovery-linux.sh  touch_ecc.sh
Factory-final.sh  hostname_write.sh  mkinand-linux.sh  mkspi-advboot.sh        touch_fa.sh
root@ubuntu:/home/work/wise-710/release/2019-06-05-Release3-ubuntu-A101-3/WISE-710-r3-ubuntu.20190605/scripts# ./mksd_recovery-linux.sh /dev/sdc ubuntu16044
All data on /dev/sdc now will be destroyed! Continue? [y/n]
y
partition start
DISK SIZE - 7742685184 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 1874944 4k blocks and 469568 inodes
Filesystem UUID: 519c8a4f-f5fa-4d2e-b52c-908fd91a5b48
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

dd [adv_boot & u-boot]
copy [zImage & dtb]
copy [rootfs]
[Copying iNAND upgrate tools...]
```

3. 等待烧录完成即可拔出 SD 卡，插到设备上启动（跳线 SW2 ： 1-off 2-on）。

注意：如果是在 WISE-710 设备的 EMMC 系统中制作 SD 启动卡则应该执行如下命令：./mksd_recovery-linux.sh /dev/mmcblk1 ubuntu16044

3.5. 启动 Ubuntu 系统

- 系统启动后：

  普通用户：advantech　　passwd：123

  超级用户：root　　passwd：123456

- 如需更改 root 密码，则可通过普通用户修改 root 密码

```
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

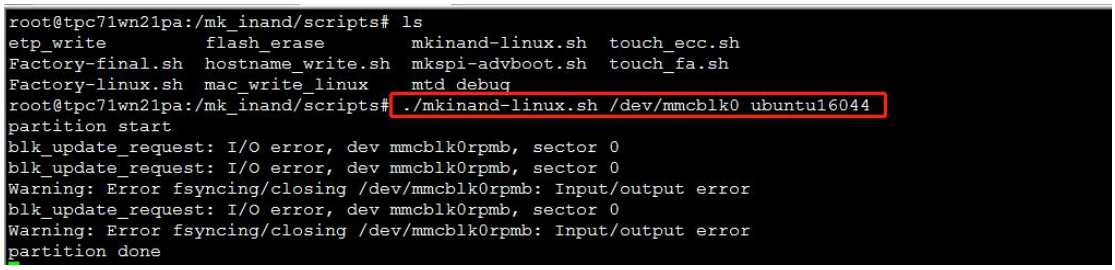
- 使用系统：

  # sudo su （输入密码确认）

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

3.6. 烧录 eMMC 启动

- 系统在 SD 卡启动后(/mk_inand/scripts 目录)



注：如提示 mkfs.vfat 命令没有找到，则可以使用【***# apt-get install dosfstools***】

- 等待烧录完成后，sync 同步，poweroff 关机，移除 SD 卡。
  注：SD：/dev/mmcblk1　eMMc：/dev/mmcblk0
- 启动之后就可以选择从 SPI 启动（跳线 SW2 ：1-on 2-off)。

4. WISE-710 外设测试

4.1. eMMC Flash 读写测试

Step1：当设备从 SD 卡启动后，运行如下命令擦除并检查 eMMC Flash

（注：从 SD 卡启动时，系统中识别到的 eMMC Flash 节点为 mmcblk0。）

root@wise710a1:~# ***dd if=/dev/zero of=/dev/mmcblk0 bs=1024 count=1 seek=1***
1+0 records in
1+0 records out
root@wise710a1:~# ***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 |................|

步骤 2: 运行如下命令写入并检查 eMMC Flash。

root@wise710a1:~# ***echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk0 bs=1024 count=1 seek=1***
0+1 records in
0+1 records out
root@wise710a1:~# ***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 读写测试

步骤 1: 插入 USB 存储设备并查看 WISE-710 设备列表获取设备节点。

步骤 2: 运行如下命令擦除并检查 USB 存储设备。

```
root@wise710a1:~#dd if=/dev/zero of=/dev/sda bs=1024 count=1 seek=1
1+0 records in
1+0 records out
root@wise710a1:~# 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 |................|
```

步骤 3: 运行如下命令写入并检查 USB 存储设备。

```
root@wise710a1:~# echo -n "0123456789ABCDEF" | dd of=/dev/sda bs=1024 count=1 seek=1
0+1 records in
0+1 records out
root@wise710a1:~# 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 卡读写测试

步骤 1: 当设备从内部 eMMC Flash 启动（启动时不插入 SD 卡）后，可从系统中查看以下信息。

```
root@wise710a1:~# 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
```

步骤 2: 将 SD 卡插入 WISE-710 并重新查看设备信息，/dev/mmcblk1 代表当前 SD 卡设备（本例中 SD 卡存在两个分区）。

```
root@wise710a1:~# 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
```

步骤 3: 运行如下命令擦除并检查 SD 卡。

```
root@wise710a1:~# dd if=/dev/zero of=/dev/mmcblk1 bs=1024 count=1 seek=1
1+0 records in
1+0 records out
root@wise710a1:~# 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 |................|
```

步骤 4: 写入并检查 SD 卡。

```
root@wise710a1:~# echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk1 bs=1024
count=1 seek=25118
0+1 records in
0+1 records out
root@wise710a1:~# 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. 串口测试

如下表所示，WISE-710 共有 3 个串口。其中 COM1 有三种模式分别是 debug 模式、RS232 模式、RS485 模式。 COM2 和 COM3 只有 RS485 模式。

HW	SW	DEVICE
COM1	Debug port	/dev/ttymxc0
COM1	232 / 485	/dev/ttyUSB0
COM2	485	/dev/ttyUSB2
COM3	485	/dev/ttyUSB3

Test COM1 rs-232 loopback(baudrate 9600):

```
root@wise710a1:~# cd /usr/Advantech/Serial_test
root@wise710a1:/usr/Advantech/Serial_test# ./st -rsavo -m 232 -b 9600
/dev/ttyUSB0
```

Test COM1 rs-232 read(baudrate 9600):

```
root@wise710a1:~# cd /usr/Advantech/Serial_test
```

```
root@wise710a1:/usr/Advantech/Serial_test# ./st -ravo -m 232 -b 9600 /dev/ttyUSB0
```

Test COM1 rs-232 write(baudrate 9600):

```
root@wise710a1:~# cd /usr/Advantech/Serial_test
root@wise710a1:/usr/Advantech/Serial_test# ./st -savo -m 232 -b 9600 /dev/ttyUSB0
```

Test COM2 rs-485 read(baudrate 115200):

```
root@wise710a1:~# cd /usr/Advantech/Serial_test
root@wise710a1:/usr/Advantech/Serial_test# ./st -ravo -m 485 -b 115200 /dev/ttyUSB2
```

Test COM3 rs-485 write(baudrate 115200):

```
root@wise710a1:~# cd /usr/Advantech/Serial_test
root@wise710a1:/usr/Advantech/Serial_test# ./st -savo -m 485 -b 115200 /dev/ttyUSB3
```

注意：COM1 的 debug、RS232、RS485 三种模式切换时通过 SW9 和软件一起实现的。SW9 的具体操作请参考 WISE-710 硬件操作手册。需要注意的是改变 SW9 以后需要重启两次 COM1 的模式才能真正切换过来。

4.5. LAN 口测试

4.5.1 命令行配置

查看当前 IP.

```
root@wise710a1:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr c4:00:ad:2b:72:00
          inet addr:172.21.73.179  Bcast:172.21.73.255  Mask:255.255.255.0
          inet6 addr: fe80::c600:adff:fe2b:7200/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:819 errors:0 dropped:0 overruns:0 frame:0
          TX packets:41 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:66038 (64.4 KiB)  TX bytes:8198 (8.0 KiB)
```

开启 Eth0 网络

```
root@wise710a1:~# nmcli connection up eth0
```

关闭 Eth0 网络

```
root@wise710a1:~# nmcli connection down eth0
```

查看当前网络的 **connection NAME & UUID**

```
root@wise710a1:~# nmcli c
NAME                UUID                                  TYPE            DEVICE
eth0    ba093436-fba3-46d9-991d-97ec56064bce  802-3-ethernet  --
eth1    b515183e-dc86-4486-81d0-9936fd1c0125  802-3-ethernet  --
```

删除当前 Eth0 链接

```
root@wise710a1:~# nmcli connection delete eth0
```

或

```
root@wise710a1:~# nmcli connection delete ba093436-fba3-46d9-991d-97ec56064bce
```

设置 Eth0 为动态 IP 模式

```
root@wise710a1:~# nmcli connection add con-name "eth0" type ethernet ifname eth0
```

设置 Eth0 静态 IP 模式

```
root@wise710a1:~# nmcli connection add con-name "eth0" ifname eth0 autoconnect yes type ethernet ip4 172.21.73.179/24 gw4 172.21.73.253
root@wise710a1:~# nmcli connection down "eth0"
root@wise710a1:~# nmcli connection mod "eth0" ipv4.dns 172.21.128.10
root@wise710a1:~# nmcli connection up "eth0"
```

查看当前 Eth0 网络状态

```
root@wise710a1:~# nmcli device show eth0
GENERAL.DEVICE:                         eth0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                           C4:00:AD:2B:72:00
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                       eth0
```

```
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/ActiveConnection/3
WIRED-PROPERTIES.CARRIER:                    on
IP4.ADDRESS[1]:                              172.21.73.179/24
IP4.GATEWAY:                                 172.21.73.253
IP4.DNS[1]:                                  172.21.128.10
IP6.ADDRESS[1]:
fe80::c600:adff:fe2b:7200/64
IP6.GATEWAY:
IP6.ROUTE[1]:                                dst = ff00::/8, nh = ::, mt =
256
```

Ping 测试 WISE-710 的 IP 是 172.21.73.179 目标机的 IP 是 172.21.73.29

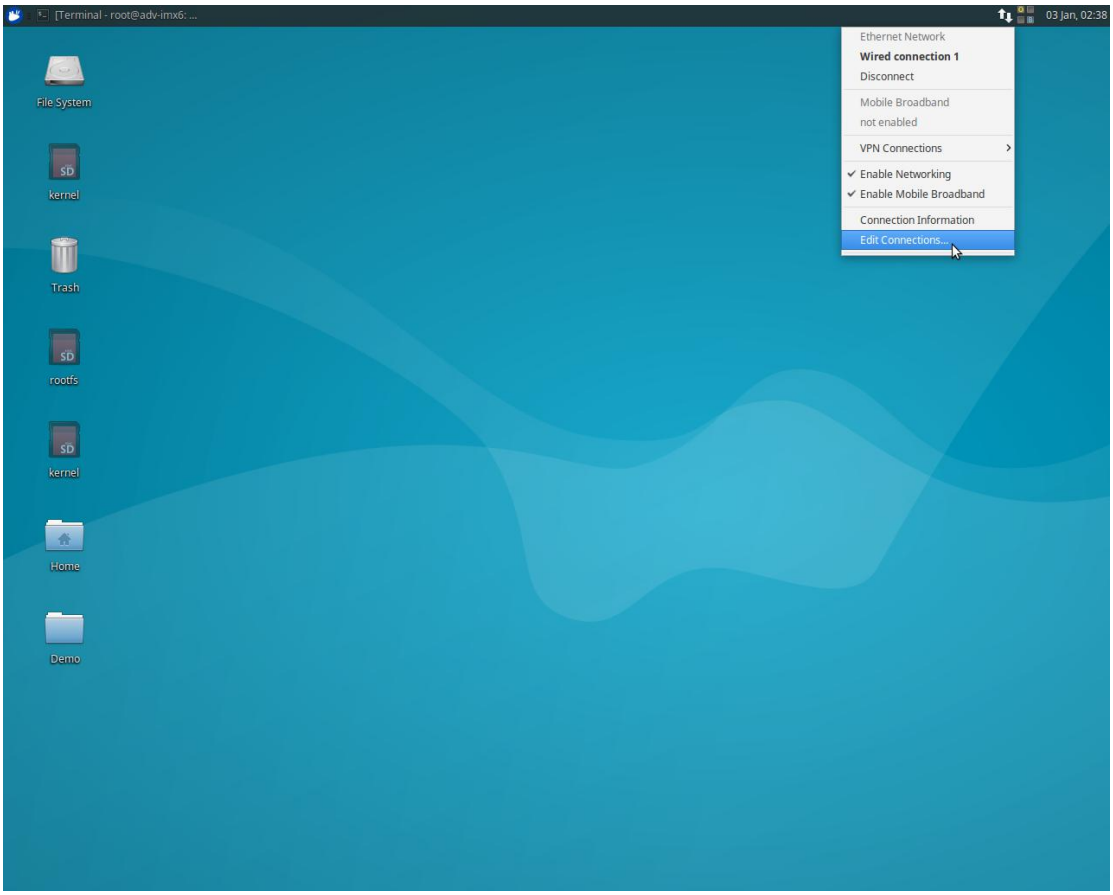
```
root@wise710a1:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr c4:00:ad:2b:72:00
          inet addr:172.21.73.179  Bcast:172.21.73.255
Mask:255.255.255.0
          inet6 addr: fe80::c600:adff:fe2b:7200/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21354 errors:0 dropped:0 overruns:0 frame:0
          TX packets:240 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1642835 (1.5 MiB)  TX bytes:40223 (39.2 KiB)
root@wise710a1:~# ping 172.21.73.29 -c 5
PING 172.21.73.29 (172.21.73.29) 56(84) bytes of data.
64 bytes from 172.21.73.29: icmp_seq=1 ttl=64 time=0.517 ms
64 bytes from 172.21.73.29: icmp_seq=2 ttl=64 time=0.420 ms
64 bytes from 172.21.73.29: icmp_seq=3 ttl=64 time=0.430 ms
64 bytes from 172.21.73.29: icmp_seq=4 ttl=64 time=0.431 ms
64 bytes from 172.21.73.29: icmp_seq=5 ttl=64 time=0.431 ms

--- 172.21.73.29 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.420/0.445/0.517/0.044 ms
root@wise710a1:~#
```

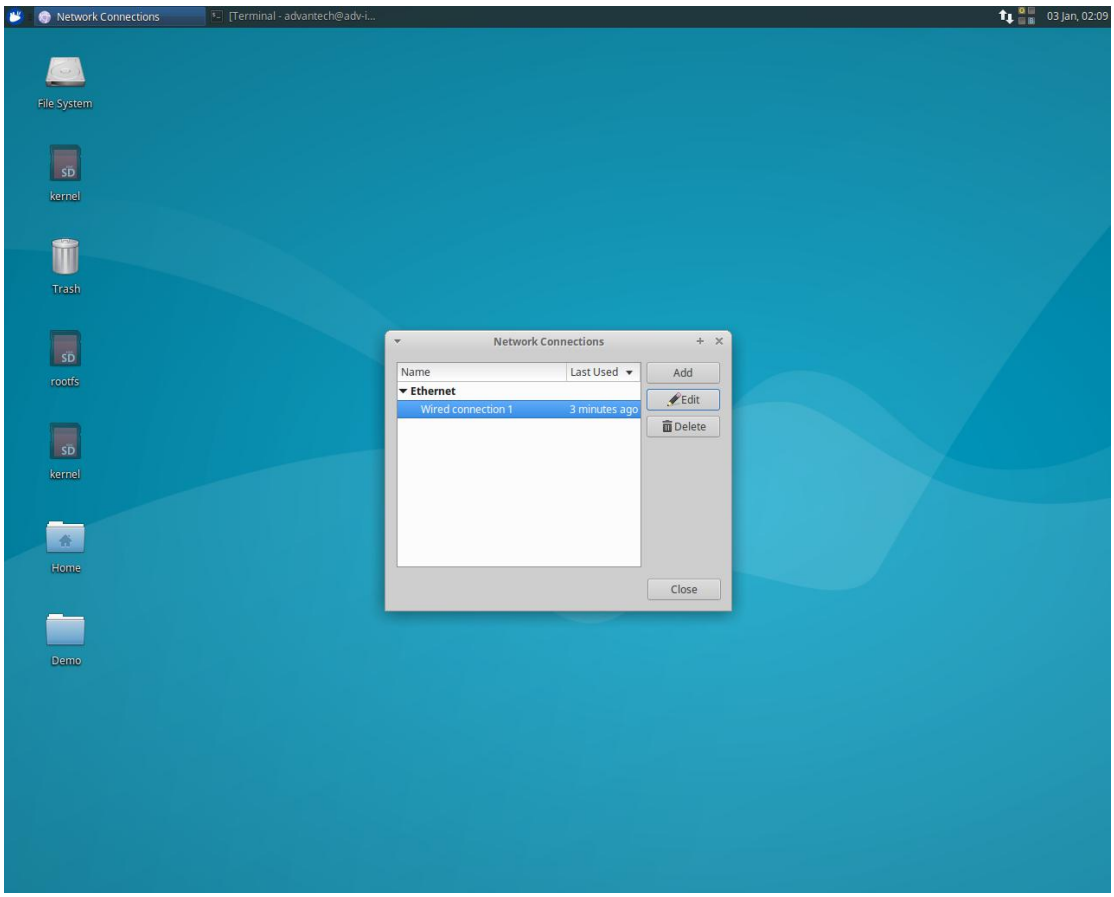
Note! *The target computer (Client) firewall need close.*

4.5.2 图形配置静态 IP

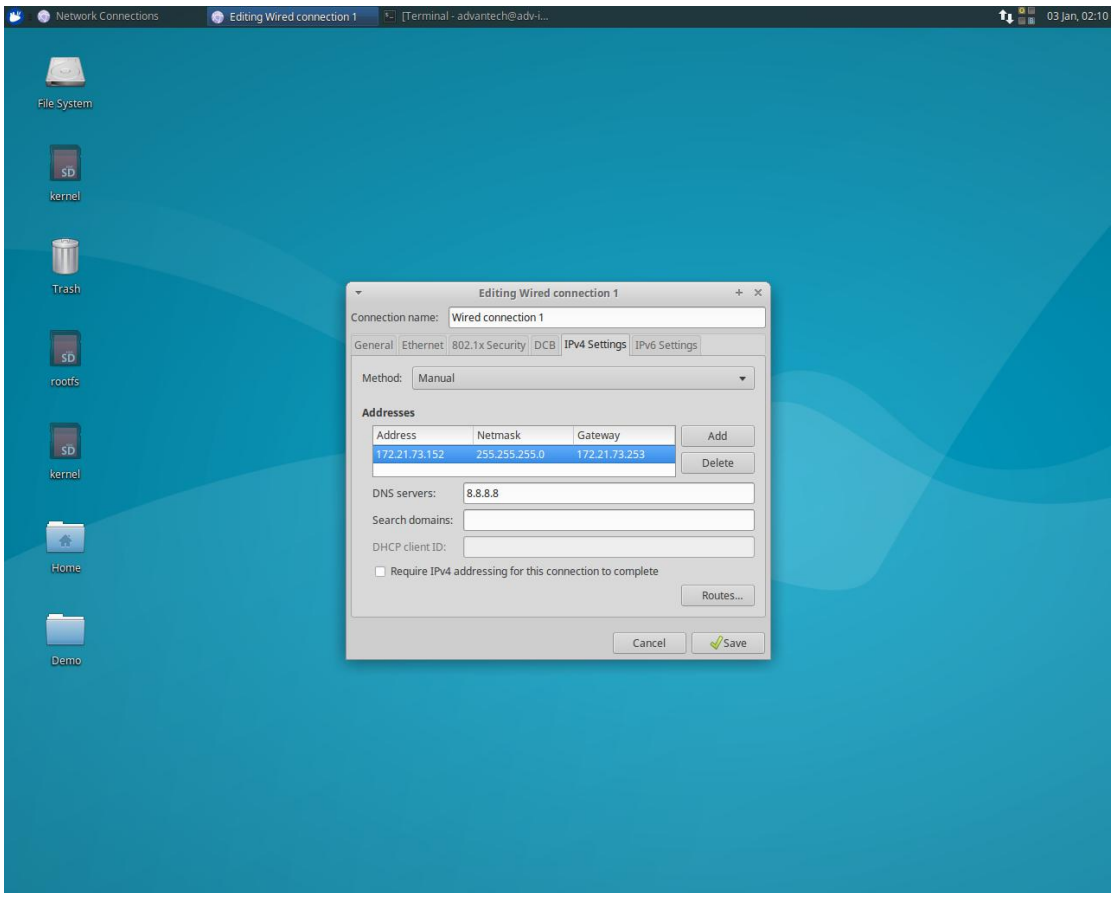
Step1：右上角点击 Edit Connections..



Step2：点击 Edit



Step3：设置 ip，子网掩码，网关，DNS；点击保存



Step4：插拔网线，通过 ifconfig 查看

4.6. 外接 WIFI 模块

4.6.1 支持模块

WiFi 型号：968AD00259（RTL8188EE）

4.6.2 WIFI 配置与连接（968AD00259）

命令行配置

Setp1：使能无线网络

```
root@wise710a1:~# nmcli radio wifi on
```

Setp2：扫描 WiFi 网络

```
root@wise710a1:~# nmcli device wifi
```

Setp3：连接 WPA2/PSK 加密网络

```
root@wise710a1:~# nmcli device wifi connect <WIFI_SSID> <WIFI_PASSWD> name wlan0 ifname wlan0
```

Setp4：连接无密码网络

```
root@wise710a1:~# nmcli device wifi connect <WIFI_SSID> “” name wlan0 ifname wlan0
```

Setp5：关闭当前连接

```
root@wise710a1:~# nmcli connection down wlan0
```

Setp6：删除当前的连接

```
root@wise710a1:~# nmcli connection delete wlan0
```

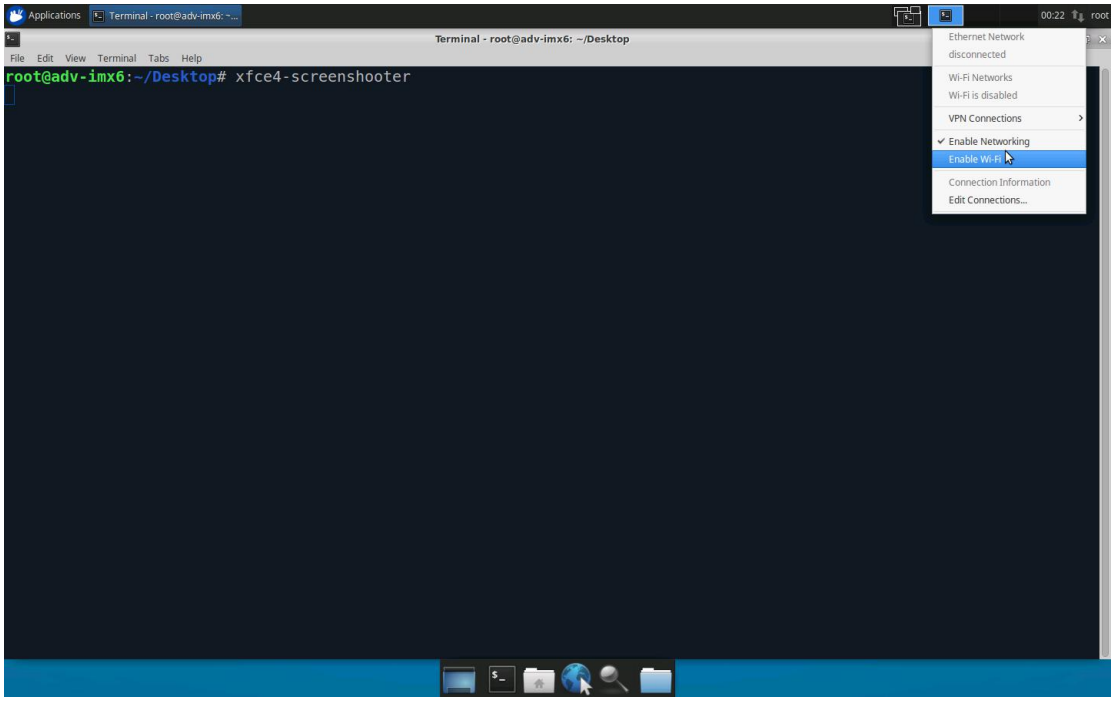
Setp7：关闭无线网络

```
root@wise710a1:~# nmcli radio wifi off
```

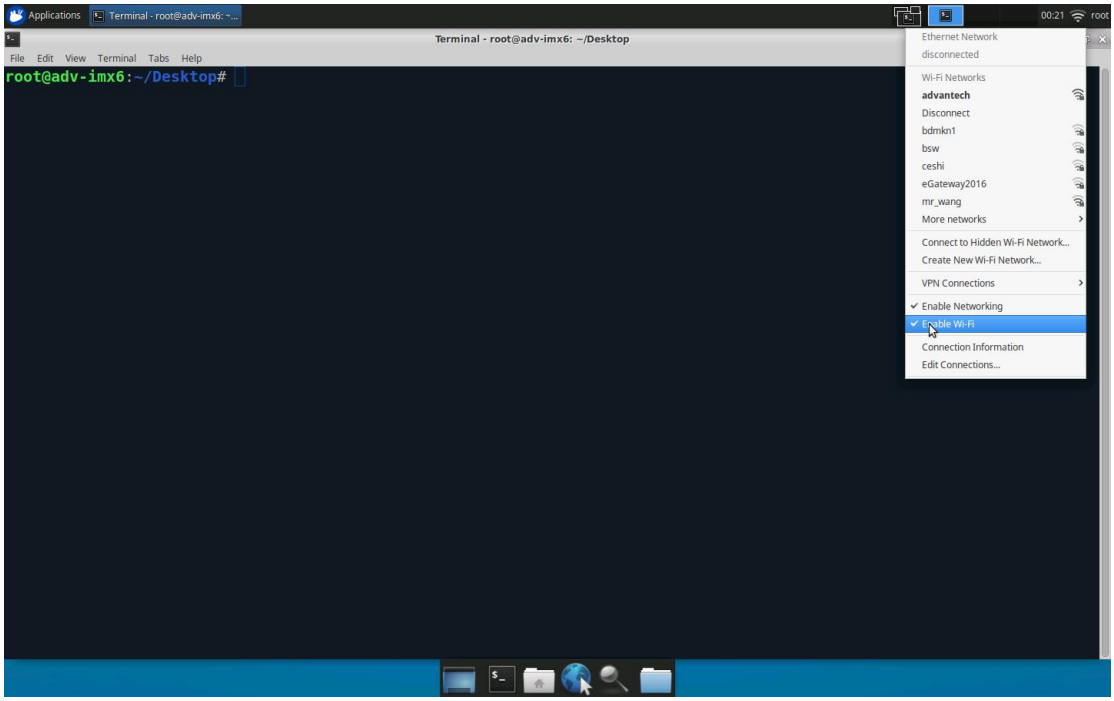
图形配置方式

Setp1：系统主界面的右上角有网络连接图标 

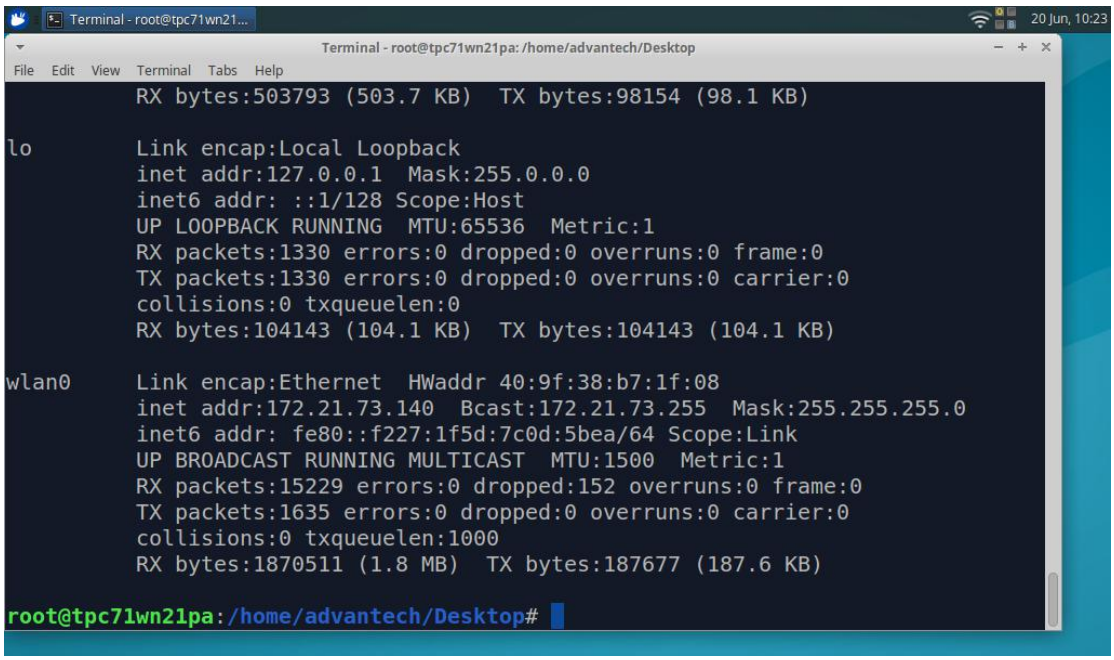
Setp2：点击图标将 WIFI 功能打开 Enable Wi-Fi



Setp3：使能 WiFi 后可以查看周围的无线网络，选择一个进行连接



Setp4：连接成功后可以正常上网



4.7. 外接 4G/LTE 模块测试

4.7.1 支持模块

4G/LTE 模块型号：EWM-C117FL01E 系列

4.7.2 4G/LTE 模块配置和连接(型号 EWM-C117FL01E)

命令行配置

Setp1：使能移动数据网络

```
root@wise710a1:~# nmcli radio wwan on
```

Setp2：连接移动数据网络

```
root@wise710a1:~# nmcli connection add con-name "ppp" type gsm ifname ttyUSB1 apn 3gnet user uninet password "111111"
```

或

```
root@wise710a1:~# nmcli connection add con-name "usb" type ethernet ifname usb0
```

Setp3：断开移动数据网络连接

```
root@wise710a1:~# nmcli connection down "ppp"
```

Setp4：删除移动数据网络连接

```
root@wise710a1:~# nmcli connection delete "ppp"
```

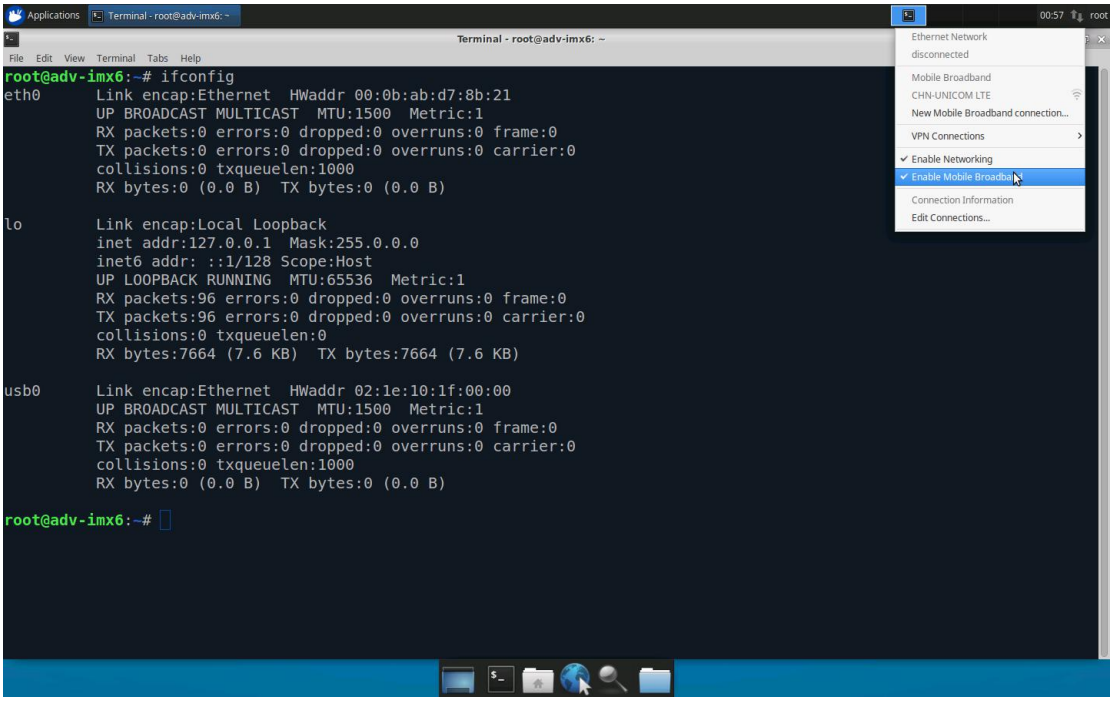
Setp5：关闭移动数据网络连接

```
root@wise710a1:~# nmcli radio wwan off
```

图形配置方式

Step1：点击系统主界面右上角连接图标 

Step2：点击图标将 3G 功能打开 Enable Mobile Broadband



Step3：点击 New Mobile Broadband connection 后，弹出的对话框输入任意密码点击下一步验证即可。

4.8. Time And Date Setting

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

```
root@wise710a1:~# date -s “2019/01/01 13:25:00”
```

Synchronize time from the NTP server:

```
root@wise710a1:~# ntpdate <NTPSERVERIP>
```

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

```
root@wise710a1:~# hwclock -w
```

Reset system time (use RTC hardware clock time):

```
root@wise710a1:~# hwclock -s
```

Set system time zone (use Shanghai time):

```
root@wise710a1:~# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
root@wise710a1:~# sync
```

4.9. DIO Test

As you can see below, there are 4 DI/DO supported by WISE-710 internal.

HW	Default value	System node	Software node ID
DO1	low	/sys/class/gpio/gpio1/value	1
DO2	low	/sys/class/gpio/gpio2/value	2
DO3	low	/sys/class/gpio/gpio3/value	3
DO4	low	/sys/class/gpio/gpio4/value	4
DI1	-	/sys/class/gpio/gpio5/value	5
DI2	-	/sys/class/gpio/gpio6/value	6
DI3	-	/sys/class/gpio/gpio7/value	7
DI4	-	/sys/class/gpio/gpio8/value	8

Please use Advantech EAPI api & example to test DIO.

Set DO1 output value to high:

```
root@wise710a1:~# cd /usr/Advantech/EAPI_test
root@wise710a1: /usr/Advantech/EAPI_test# ./testdl_gpio 5 1 1
GPIOSetLevel Id: 1
Level: 1
```

Set DO2 output value to low:

```
root@wise710a1:~# cd /usr/Advantech/EAPI_test
root@wise710a1: /usr/Advantech/EAPI_test# ./testdl_gpio 5 2 0
GPIOSetLevel Id: 2
Level: 0
```

Get DI1 output value:

```
root@wise710a1:~# cd /usr/Advantech/EAPI_test
root@wise710a1:/usr/Advantech/EAPI_test# ./testdl_gpio 4 5
GPIOGetLevel Id: 5
level: 0
```

Get DO1 output value:

```
root@wise710a1:~# cd /usr/Advantech/EAPI_test
root@wise710a1:/usr/Advantech/EAPI_test# ./testdl_gpio 4 1
GPIOGetLevel Id: 1
level: 1
```

4.10. CAN Test

As you can see below, there are 1 flexCAN supported by WISE-710 internal.

HW	DEVICE	MODE
flexCAN0	can0	socket can

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

```
root@wise710a1:~# ip link set can0 down
root@wise710a1:~# ip link set can0 up type can bitrate 125000 loopback off
root@wise710a1:~# ip link set can0 up
root@wise710a1:~# 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@wise710a1:~# 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@wise710a1:~# cansend can0 123#11
```

Recv message from socket can0:

```
root@wise710a1:~# candump can0
```

4.11. 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      @#
```

Step4: Remote desktop (use VNC Viewer 6.18.625)


VNC Viewer
VNC® Viewer
VNC Server: 172.21.73.131:5900
Encryption: Let VNC Server choose
About...
Options...
Connect


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.
Continue
Cancel


tpc71wn21pa:0 - VNC Viewer
kernel
Trash
data
File System
recovery
Home
rootfs
Demo
52 MB Volume

5. 系统配置

5.1. 终端命令行

Linux 下很多系统操作和配置都是在命令行下进行的，启动命令行有以下两种方法：
方式一：启动一个终端可以选择 Applications -> Accessories -> Terminal
方式二：快捷方式：Ctrl + Alt + T
在命令行，可以运行各种 shell 命令和脚本，常用命令

- 查看目录： ls
- 创建目录： mkdir（目录名）
- 切换目录： cd（directory/location）
- 复制文件/目录： cp（源文件或目录名）（目标目录或文件名）
- 删除文件/目录： rm （文件或目录名）
- 重命名文件/目录： mv （文件或目录名）
- 查找文件/目录： locate （文件或目录名）
- pwd 显示当前目录
- ifconfig 显示系统的网络

命令行是 Linux 下最基本的操作，是每个 Linux 开发者和使用者的必备技能，这边不详细说明。

5.2. 添加用户，密码

Step1：创建新用户

```
# useradd -d /home/test -g root -m test      //新建 test 用户并指  定用户组为 root 用户组,并自动建立登录目录
# passwd  test                  //给 test 用户设置密码
```

Step2：给已有的用户增加工作组

```
# usermod -G root test   //给 test 用户设置 root 附属用户组
# gpasswd -a test root    //给 test 用户设置 root 用户组
```

5.3. 开机自动运行特定程序

系统默认开机自启动文件在/etc/rc.local 下，客户将运行的特定程序以脚本的形式写入，1.在 rc.local 文件中运行该脚本即可。
例如：
(1)客户需要开机运行 demo 程序,首先创建一个 sh 脚本文件,将运行 demo 编写到脚本当中。

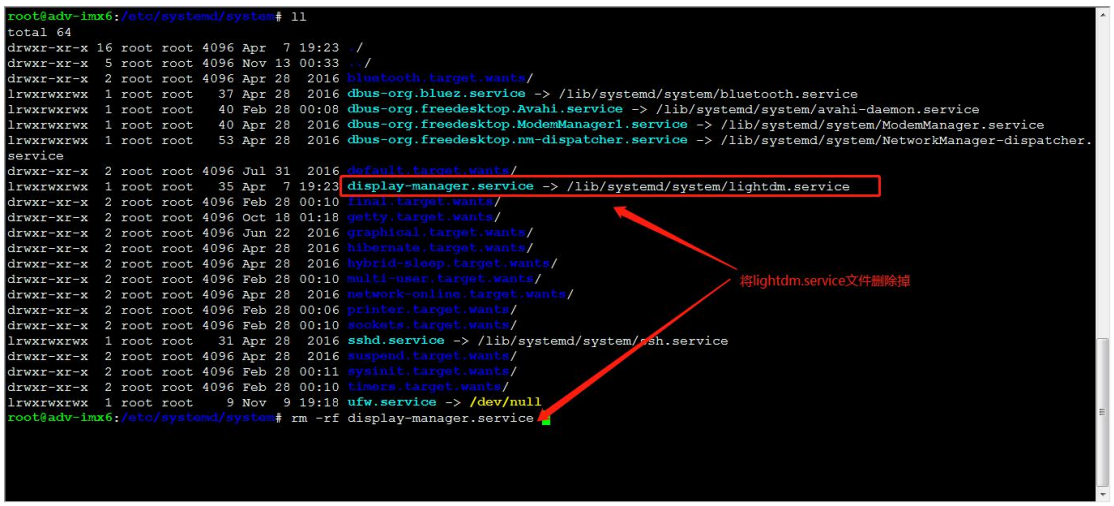
```
# vi demo.sh
cd  demo/  &&  ./demo
```

(2)执行脚本编写完成之后，接下来在/etc/rc.local 中写入运行该脚本的命令。

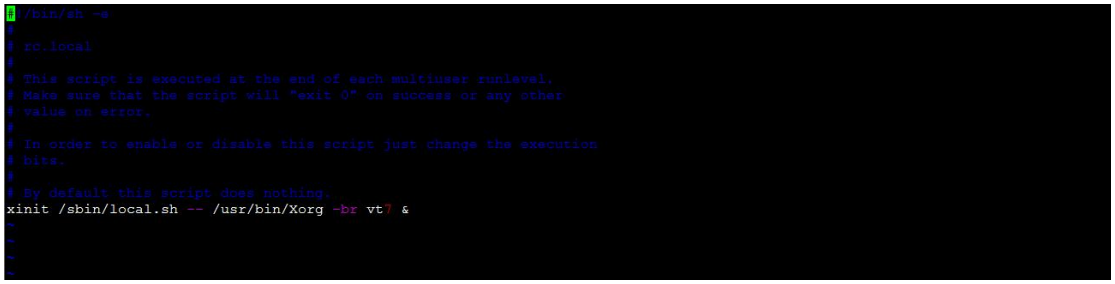
 # vi /etc/rc.local
/root/demo.sh

2.带图形方式开机自启动

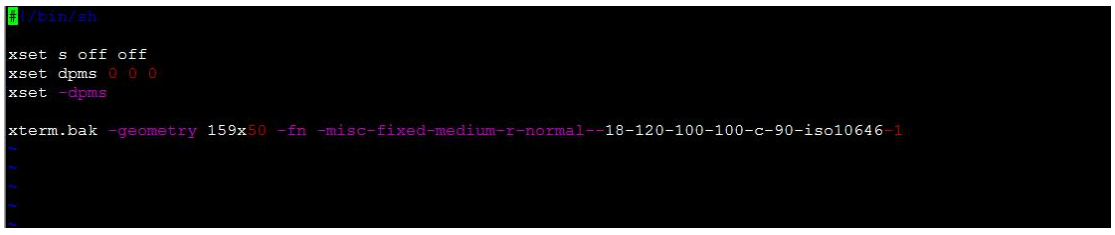
（1）首先删除掉登陆管理器的 service 文件



（2）通过在/etc/rc.local 中添加启动 Xorg 命令



（3）在/sbin/local.sh 中创建脚本添加需要启动的程序



（4）通过 Systemd 启动 rc.local 的服务文件

 # systemctl enable rc-local.service

 # systemctl restart rc-local.service

 # reboot

（5）开机启动程序效果

6. 二次开发客制系统的备份和部署

针对具体的应用，开发人员肯定需要在我们提供的系统上进行二次开发和客制，已满足特定应用方案的需求，但客户在 WISE-710 上完成二次开发客制后，如何将该系统备份出来，批量部署的其他的机器，本章节给出了相应的方法：
当用户将系统安装在 emmc 上面，并且整合自己的应用，完成调试后，需要将该系统备份出来进行批量部署时，可以使用我们提供的备份脚本程序进行简备份和部署。
注：是备份 emmc 上的系统到 SD，再通过 SD 烧录到其他设备的 emmc,所以二次开发客制的系统是在 emmc 上。

6.1. 备份目标系统 image

1) 提前准备一张 SD 并烧录好我们提供的原始 ARM Ubuntu 系统，并通过 SD 卡启动。
   注：此系统使用我们提供的烧录到 SD 的系统 image。
2) 系统启动起来进入（/mk_inand/scripts/）目录执行 backup_emmc_rootfs_to_sdcard.sh 脚本备份系统

```
imx6dlwise710 login:
root@wise710a1:~# cd /mk_inand/scripts
root@wise710a1:~/mk_inand/scripts#./backup_emmc_rootfs_to_sdcard.sh
/dev/mmcblk0
```

3) 等待备份完成后，sync 同步
4) poweroff 关机，拔出 SD 卡

6.2. 部署到其他设备

此时 SD 卡中的备份了最新的系统，可以将新系统烧录到其他 WISE-710 设备上，具体烧录步骤和之前 Ubuntu 烧录到 emmc 相同，具体请参考章节 3.6。

7. 加值客制解决方案

以下解决方案，研华已有相应的方案，但这些方案需要进行专门的客制化，其中有些客制是需要付费的。具体请联系产品 PM，获得更进一步的信息。

7.1. 远程 OTA 更新

WISE-PaaS/OTA, 可以通过远程方式，更新 Ubuntu 系统或者应用程序。

7.2. 跨网段远程桌面

默认 VNC 只能在局域网内进行远程桌面，通过客制，可以支持跨网段的远程桌面。

7.3. 系统备份

对系统进行备份，在损坏是还原到初始版本

7.4. 只读文件系统

为确保断电可靠性，提供只读文件系统方案，确保系统分区只读，用户数据记录到另一个可读写分区。