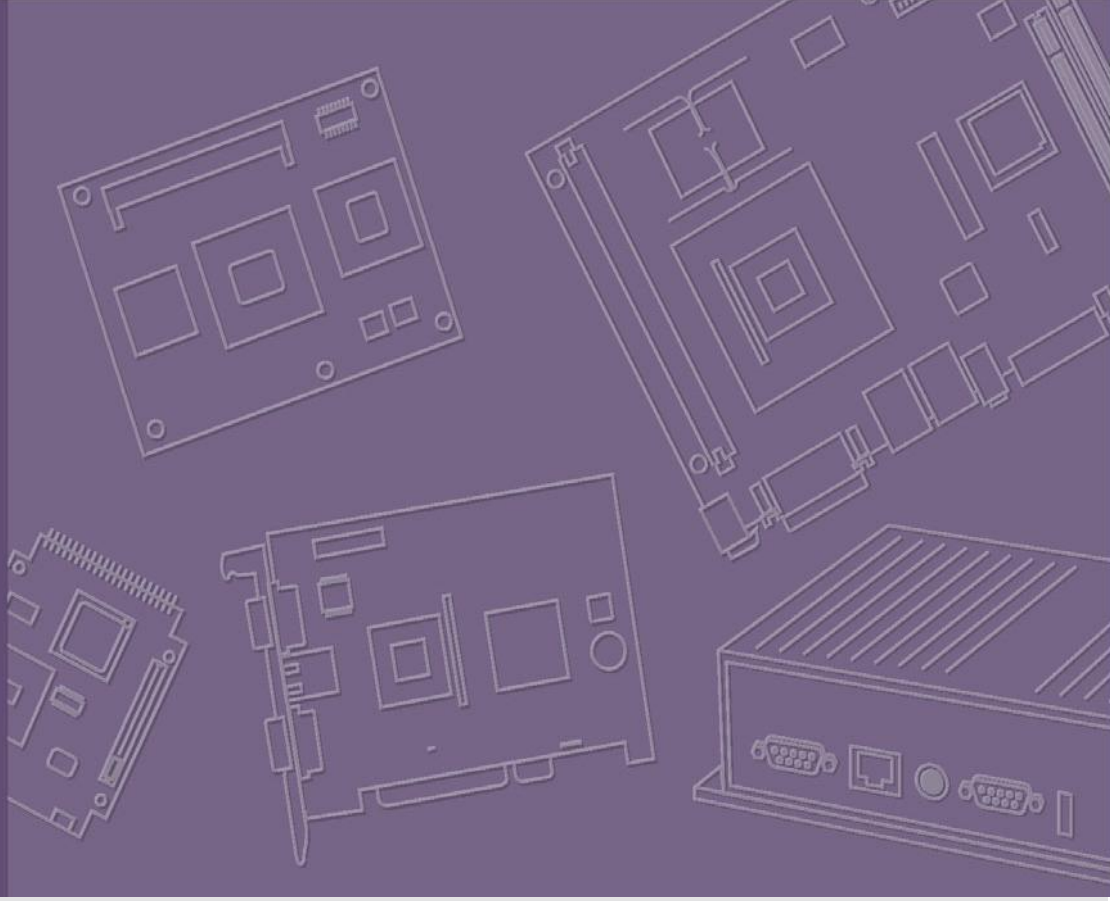


Software User Manual

WISE-710-N600A
工業通訊網關

Industrial Protocol Gateway with Freescale i.MX 6 DualLite CPU, Dual GbE, 3 x COM, 4 x DI/O, 1 x Micro USB, and 1 x Micro SD Slot

ADVANTECH

Enabling an Intelligent Planet

Contents

目录

1 Introduction

WISE-710 platform is an embedded system with Linux kernel 4.1.15 inside. It contains all system-required shell commands and drivers ready. We do not offer IDE developing environment in WISE-710 BSP, users can evaluate and develop under Ubuntu 16.04LTS 64bit environment.

There are 9 major boot components for Linux, “adv_logo_1024x600_32bpp.bmp”, “u-boot_crc.imx”, “u-boot_crc.bin”, “u-boot_crc.bin.crc”, “SPL”, “zImage”, ”imx6dl-wise710-a1.dtb”, “yocto21.tar.gz” ,“8111g-cfg” and “EdgeLink”.

It will not be able to boot into Linux environment successfully if one of above 10 files is missing from booting media (SD card or onboard flash).

The purpose of this chapter is to introduce software configuration and development of WISE-710 to you, so that you can develop your own application(s) efficiently.

WISE-710 application development is only in Linux host PC and you cannot develop your application on Windows/Android host PC. For now the official supported host version is Ubuntu 16.04 LTS, host PC in any other Linux version may have compatibility issue. In this case, we strongly recommend to have Ubuntu 16.04 LTS installed to your host PC before start WISE-710 evaluation/development.

2 Package Content

We would offer you pre-built system image for WISE-710 system recovery. Supports installation in Linux environments.

2.1 Pre-built System Image

You are able to find the pre-built image WISE-710-rx_yyyymmdd.tar.gz in WISE-710 Evaluation Kit DVD image downloaded from Advantech website. WISE-710 supports booting from SD card so you can extract the image to SD card then dump the image file to onboard eMMC to complete system recovery. For more detail, please refer to section 4.3 System Recovery.

2.2 Development Kit Package

WISE-710 development kit package contains cross compiler and some scripts used in OS development. It can be downloaded from the open source community.

3 System Recovery

This section provides detail procedures of restoring the eMMC image. You can do system recovery following these steps if you destroy onboard flash image by accident.

3.1 Make Boot SD On PC Ubuntu16.04

Copy "WISE-710-rx_yyyymmdd.tar.gz" package to your desktop.

Open "Terminal" on Ubuntu 16.04 LTS 64bit.

```
user@ubuntu:/home/user# sudo su (Change to "root" authority)
```

Input your password.

```
root@ubuntu:/home/user# cd Desktop/
root@ubuntu:/home/user/Desktop# tar -zxvf WISE-710-rx_yyyymmdd.tar.gz
```

Insert one SD card to your developing computer

Check the SD card location, like /dev/sdx

```
root@ubuntu:/home/user# cd ./WISE-710-rx_yyyymmdd/scripts
root@ubuntu:/home/user/WISE-710-rx_yyyymmdd/scripts# ./mksd_recovery-linux.sh /dev/sdx
```

Please choose ‘y’ and wait until dump disk is done.

3.2 Make Boot SD On WISE-710 Linux

On WISE-710 platform
Copy "WISE-710-rx_yyyymmdd.tar.gz" package to WISE-710 “/home/root/”.
Decompression "WISE-710-rx_yyyymmdd.tar.gz" package

```
root@wise710a1:~# cd /home/root/
root@wise710a1:~# tar -zxvf WISE-710-rx_yyyymmdd.tar.gz
```

Insert one SD card to WISE-710
Check the SD card location, like /dev/mmcblk1

```
root@wise710a1:~# cd ./WISE-710-rx_yyyymmdd/scripts
root@wise710a1:~/WISE-710-rx_yyyymmdd/scripts# ./mksd_recovery-linux.sh /dev/mmcblk1
```

Please choose ‘y’ and wait until dump disk is done.

3.3 Restoring eMMC From Boot SD

On WISE-710 platform

```
Freescale i.MX Release Distro 4.1.15-2.0.0 Yocto 2.1 (krogoth) imx6dlwise710 /dev/ttymxc0
imx6dlwise710 login: root
root@wise710a1:~# cd /mk_inand/scripts
root@wise710a1:~/mk_inand/scripts# ./mkinand-linux.sh /dev/mmcblk0
```

Please wait until dump disk is done.
Power off and remove this SD card.

3.4 Backup eMMC Rootfs To SD Card

On WISE-710 platform

```
Freescale i.MX Release Distro 4.1.15-2.0.0 Yocto 2.1 (krogoth) imx6dlwise710 /dev/ttymxc0
imx6dlwise710 login: root
root@wise710a1:~# cd /mk_inand/scripts
root@wise710a1:~/mk_inand/scripts# ./backup_emmc_rootfs_to_sdcard.sh /dev/mmcblk0
```

Please wait for the script to finish
After the script is executed, you can use the SD card to burn the backup system to other devices.

4 Set up Compiler toolchain Environment

All instructions in this guide are based on Ubuntu 16.04 LTS developing environment. Please install the Ubuntu 16.04 LTS at your PC/NB in advance. You can use the cross compiler toolchain to compile the related applications (arm-poky-linux-gnueabi-

gcc version is 5.3.0). When you obtain the compiler toolchain package, please refer to following instructions to extract to your developing environment.

```
root@ubuntu:/home/user# tar -zxvf fsl-imx-x11.tar.bz2
root@ubuntu:/home/user# mv fsl-imx-x11 /opt
root@ubuntu:/home/user# cd /opt/fsl-imx-x11/4.1.15-2.0.0
root@ubuntu:/opt/fsl-imx-x11/4.1.15-2.0.0#
. environment-setup-cortexa9hf-neon-poky-linux-gnueabi
```

Or you can download the cross compilation tool chain from the open source community(http://releases.linaro.org/components/toolchain/binaries/5.4-2017.01/arm-linux-gnueabihf/) and refer to the following steps for installation.

```
root@ubuntu:/home/user# xz -d gcc-linaro-5.4.1-2017.01-i686_arm-linux-gnueabihf.tar.xz
root@ubuntu:/home/user# tar -xvf gcc-linaro-5.4.1-2017.01-i686_arm-linux-gnueabihf.tar
root@ubuntu:/home/user# export PATH=/gcc-linaro-arm-linux-gnueabihf-5.4.1-2017.01_linux/bin:$PATH
```

5 Test tools

All test tools must be verified on WISE-710 Yacto2.1 file system “/usr/Advantech/”, please prepare required test fixtures before verifying each specified I/O. If you have any problem to get the test fixture, please contact your Advantech contact window for help.

5.1 eMMC Test

Step1: **When booting from SD**, erase and check

```
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 |................|
```

Step2: Write and check

```
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|
```

Note!

1. *This operation **may damage the data stored** in eMMC flas. Please make sure there is no critical data in the eMMC flash being used for this test.*

5.2 SD Test

Step 1: **When booting from eMMC**, you would see only below directories

```
root@wise710a1:~# ls -l /dev/mmcblk*
brw-rw---- 1 root disk 179,    0 Jan 25 15:50 /dev/mmcblk0
```

```
brw-rw---- 1 root disk 179,   8 Jan 25 15:50 /dev/mmcblk0boot0
brw-rw---- 1 root disk 179, 16 Jan 25 15:50 /dev/mmcblk0boot1
brw-rw---- 1 root disk 179,   1 Jan 25 15:50 /dev/mmcblk0p1
brw-rw---- 1 root disk 179,   2 Jan 25 15:50 /dev/mmcblk0p2
brw-rw---- 1 root disk 179,   3 Jan 25 15:50 /dev/mmcblk0p3
brw-rw---- 1 root disk 179,   4 Jan 25 15:50 /dev/mmcblk0p4
brw-rw---- 1 root disk 179, 24 Jan 25 15:50 /dev/mmcblk0rpmb
```

Step 2: Insert SD card to SD card slot and check your device again. You should be able to see more directories. /dev/mmcblk1 is the SD card storage (Ex. SD Card have 2 partitions).

```
root@wise710a1:~# ls -l /dev/mmcblk*
brw-rw---- 1 root disk 179,   0 Jan 25 15:50 /dev/mmcblk0
brw-rw---- 1 root disk 179,   8 Jan 25 15:50 /dev/mmcblk0boot0
brw-rw---- 1 root disk 179, 16 Jan 25 15:50 /dev/mmcblk0boot1
brw-rw---- 1 root disk 179,   1 Jan 25 15:50 /dev/mmcblk0p1
brw-rw---- 1 root disk 179,   2 Jan 25 15:50 /dev/mmcblk0p2
brw-rw---- 1 root disk 179,   3 Jan 25 15:50 /dev/mmcblk0p3
brw-rw---- 1 root disk 179,   4 Jan 25 15:50 /dev/mmcblk0p4
brw-rw---- 1 root disk 179, 24 Jan 25 15:50 /dev/mmcblk0rpmb
brw-rw---- 1 root disk 179, 32 Jan 25 15:57 /dev/mmcblk1
brw-rw---- 1 root disk 179, 33 Jan 25 15:57 /dev/mmcblk1p1
brw-rw---- 1 root disk 179, 34 Jan 25 15:57 /dev/mmcblk1p2
```

Step 3: Erase and check

```
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 |................|
```

Step 4: Write and check

```
root@wise710a1:~# echo -n "0123456789ABCDEF" | dd of=/dev/mmcblk1
bs=1024 count=1 seek=1
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|
```

Note!

1. *This operation **may damage the data stored** in SD card. Please make sure there is no critical data in the SD card being used for this test. If your SD card size is small, the **seek value need to be small**.*

5.3 LAN Test

Setting: Check current IP config.

```
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)
```

Setting: Enable eth0 connect

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

Setting: Disable eth0 connect

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

Setting: Get current network **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  --
```

Setting: Delete eth0 connect

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

Or

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

Setting: Set eth0 to DHCP mode

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

Setting: Set eth0 to STATIC mode

```
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"
```

Setting: Check current device status

```
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 test: Here is a real case for your reference. The hosts(WISE-710) IP is 172.21.73.179; the target(A desktop computer) IP is 172.21.73.29. So we can use below command to see if we can get any response from the client.

```
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.*

5.4 UART Test

As you can see below, there are 4 UART supported by WISE-710. /dev/ttymxc0 is reserved for WISE-710 debug port, the rest /dev/ttyAP0~2 ports could be applied by user.

HW	SW	DEVICE
COM1	Debug port	/dev/ttymxc0
COM1	232 / 485	/dev/ttyAP0
COM2	485	/dev/ttyAP1
COM3	485	/dev/ttyAP2

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/ttyAP0
```

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/ttyAP0
```

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/ttyAP0
```

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/ttyAP1
```

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/ttyAP2
```

Note! *The switching between the RS232 and RS485 working modes of COM1 is realized by SW9 and software. For the operation of SW9, please refer to the hardware instruction manual. It should be noted that after changing SW9, it needs to be powered on twice and then the working mode of COM1 will be successfully switched.*

5.5 4G/LTE Test

WISE-710 supports Advantech EWM-C117FL01E series modules throught mini-PCIe.

Dial as:
Enable mobile networking:

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

Connect to mobile network:

```
root@wise710a1:~# wan.sh on
```

Or

```
root@wise710a1:~# nmcli connection add con-name "ppp" type gsm ifname
ttyUSB1 apn 3gnet user uninet password "111111" nmcli connection down wlan0
```

Or

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

Disconnect from mobile network:

```
root@wise710a1:~# wan.sh down
```

Disable mobile networking:

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

5.6 WIFI Test

WISE-710 supports Realtek RTL8188EE 968AD00259 wifi modules throught mini-PCIe.

method 1:

Enable wifi networking:

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

Scan wifi networking:

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

Connect to WPA2/PSK encrypted network:

```
root@wise710a1:~# wlan.sh up <WIFI_SSID> <WIFI_PASSWD>
```

Or

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

Connect to open network:

```
root@wise710a1:~# wlan.sh up <WIFI_SSID>
```

Or

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

Disconnect from wifi network:

```
root@wise710a1:~# wlan.sh down
```

Or

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

Disable wifi networking:

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

5.7 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
```

5.8 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
```

5.9 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:

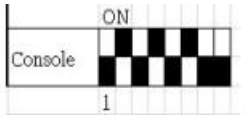
```
root@wise710a1:~# candump can0
```

6 Debug console
6 Test tools

6.1 Prepare

Before testing WISE-710, please install the putty tool on the host PC.
https://www.putty.org/

Connect the PC to WISE-710 (using RS232) and set COM1(Hardware SW9) to console mode.

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

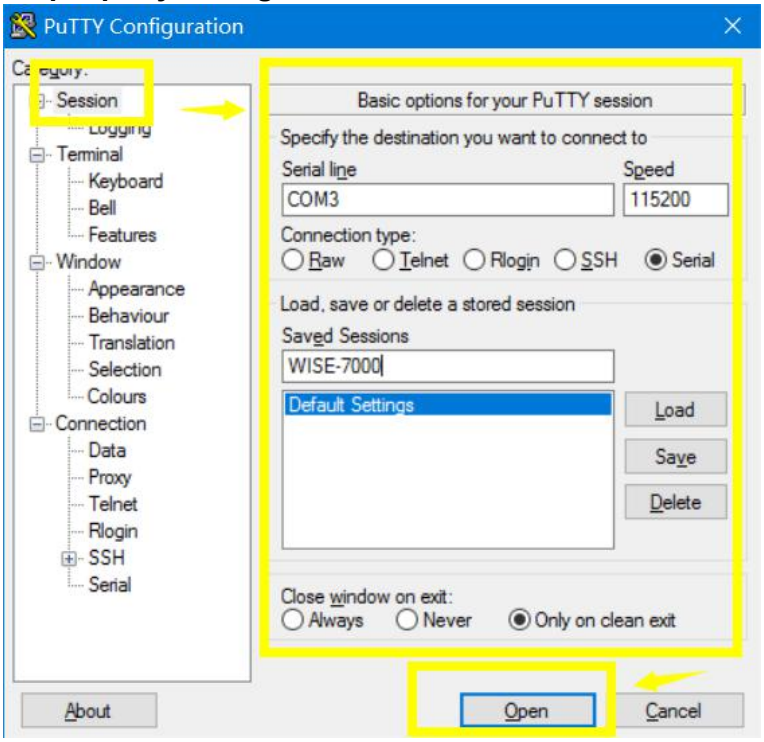
Then you can use putty to connect to the WISE-710 in following step.

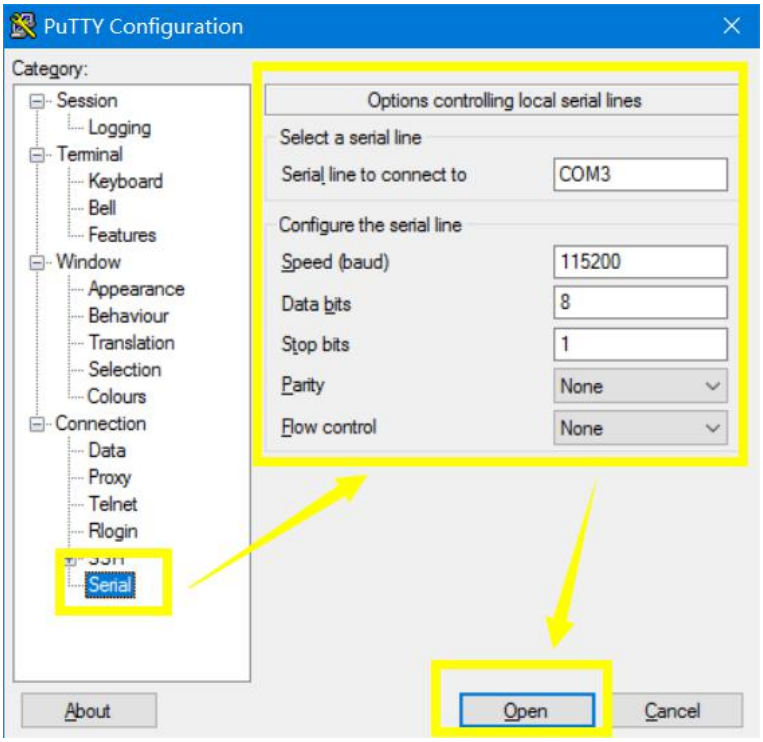
6.2 Use debug console

Step1: Check the debug COM port you connect

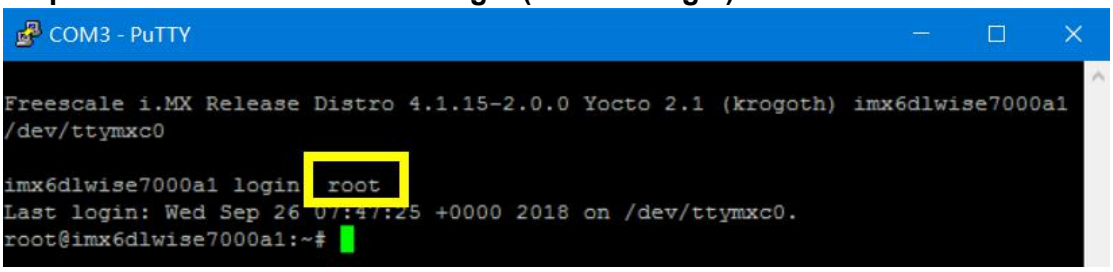
Desktop -> my computer -> property -> device manager -> COM&LPT

Step2: putty Configure



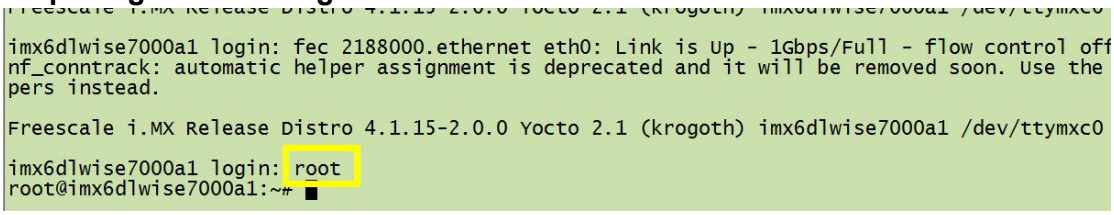


Step3: Power on WISE-710 and login (use root login)

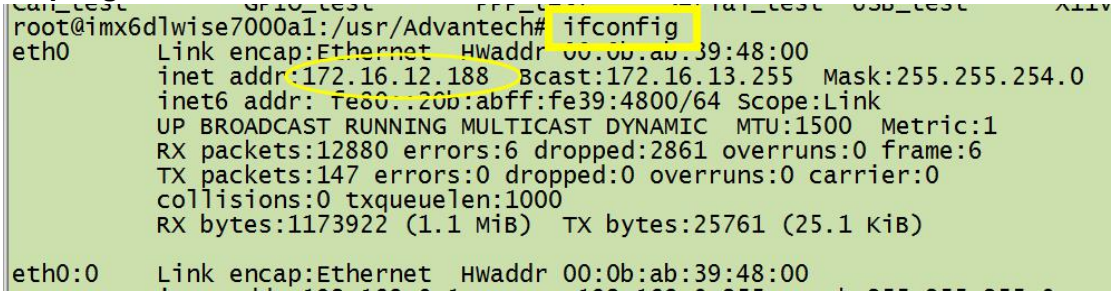


7 X11vnc remote

Step1: login with debug console



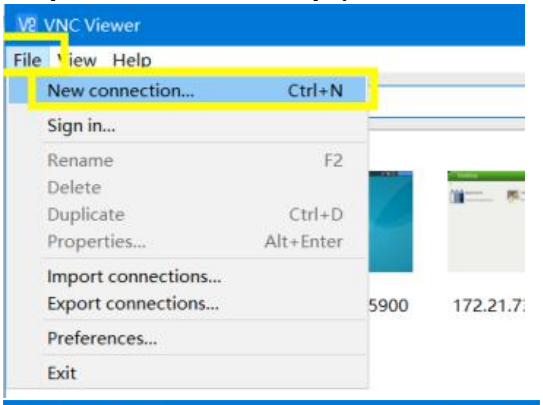
Step2: get current ethernet IP

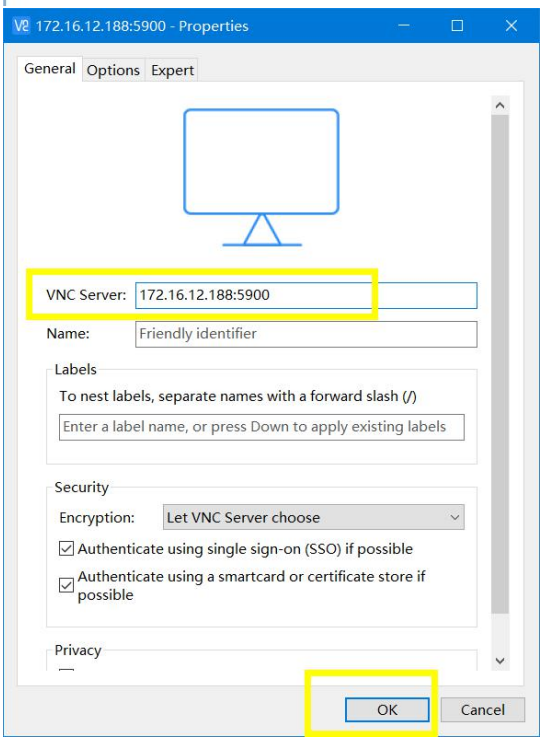


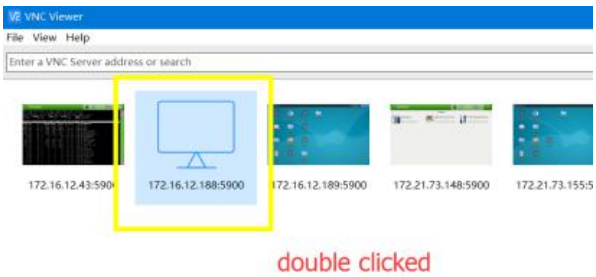
Step3: start x11vnc server

```
root@imx6dlwise710a1-47f7:~# /usr/Advantech/X11vnc_test/start_x11vnc.sh &
[1] 1424
root@imx6dlwise710a1-47f7:~# 02/01/2019 04:54:32 passing arg to libvncserver: -rf
02/01/2019 04:54:32 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!! <<<           @#
#@                                                                      @#
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#
```

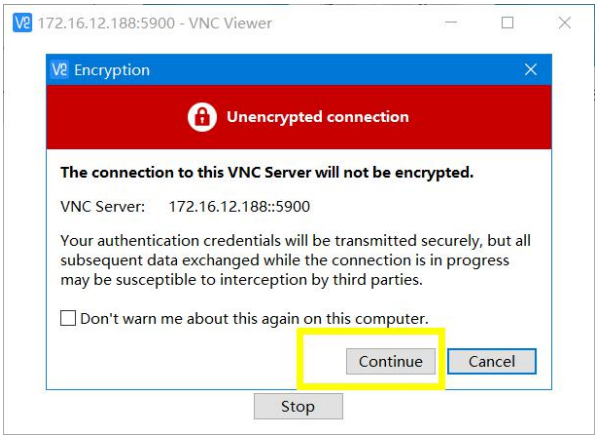
Step4: Remote desktop (use VNC Viewer 6.18.625)

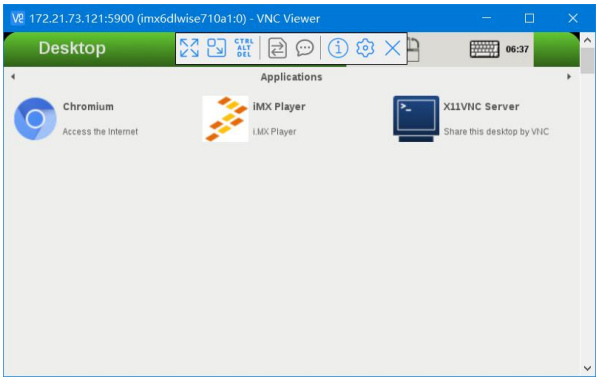






double clicked





8 Others

8.1 System log

To saved system space and improve system performance, yocto system will delete the system log files(**/var/volatile/log/syslog** and **/var/volatile/log/user.log**) at 8:30 by default every day. Automatic cleanup actions can be managed by **/etc/rc5.d/S90crond** service.

```
root@wise710a1:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

#   m    h    dom mon dow user          command
#   1    *      * * *     root          cd / && run-parts /etc/cron.hourly
# 30     7      * * *     root          cd / && run-parts /etc/cron.daily
# 42     7      * * 7     root          cd / && run-parts /etc/cron.weekly
# 55     7      1 * *     root          cd / && run-parts /etc/cron.monthly
30       8      * * *     root          /usr/bin/clean-log.sh
```

8.2 x11vnc security

X11vnc does not set the login password by default. You can set the login password for x11vnc in the following ways.

```
# Set the x11vnc login password to <PASSWD>
root@wise710a1:~# x11vnc -storepasswd <PASSWD> /etc/x11vnc.pass

# Replace x11vnc startup script
root@wise710a1:~# mv /usr/Advantech/X11vnc_test/start_x11vnc-usepasswd.sh
/usr/Advantech/X11vnc_test/start_x11vnc.sh

# Example: set password 123456 for x11vnc.
root@wise710a1:~# x11vnc -storepasswd 123456 /etc/x11vnc.pass
root@wise710a1:~# mv /usr/Advantech/X11vnc_test/start_x11vnc-usepasswd.sh
/usr/Advantech/X11vnc_test/start_x11vnc.sh
```