[1]--VNC--

1. Download “VNC-Viewer-7.10.0-Windows.exe”

Windows에 설치 (https://www.realvnc.com/en/connect/download/viewer/)

1. <https://developer.nvidia.com/embedded/learn/tutorials/vnc-setup> 참조

아래 명령 차례로 orin nx terminal에서 입력

cd /usr/lib/systemd/user/graphical-session.target.wants

sudo ln -s ../vino-server.service ./.

gsettings set org.gnome.Vino prompt-enabled false

gsettings set org.gnome.Vino require-encryption false

# -- set password ex = 'wia' 🡸 VNC-Viewer에서 Password

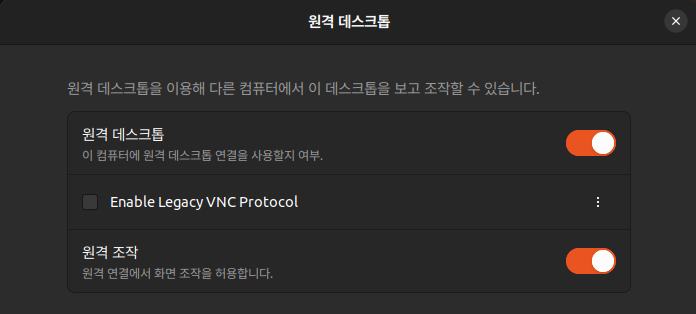
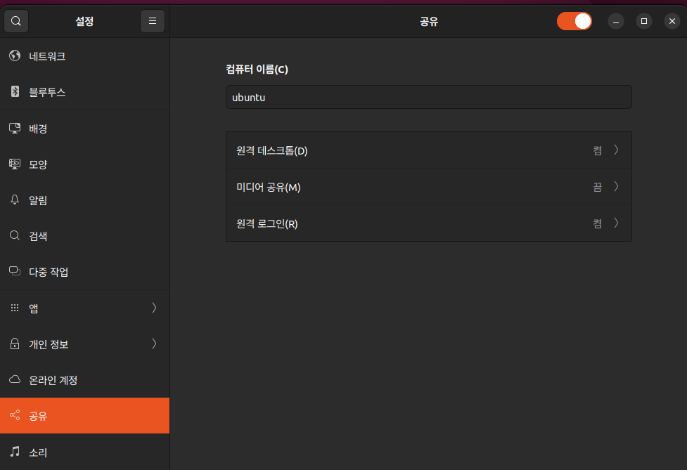
# Replace thepassword with your desired password

gsettings set org.gnome.Vino authentication-methods "['vnc']"

gsettings set org.gnome.Vino vnc-password $(echo -n 'wia'|base64)

1. Orin NX에서 세팅

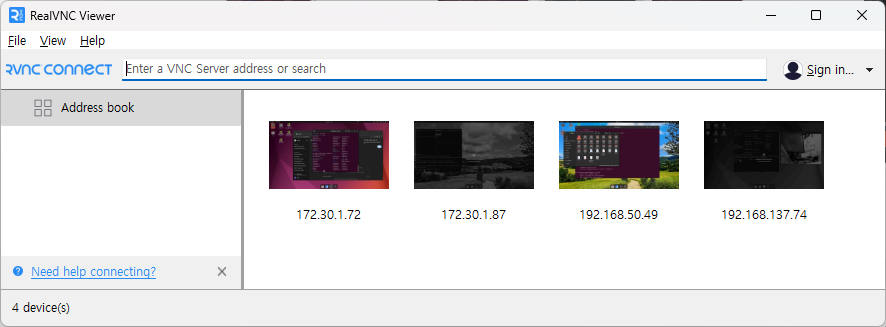
setting - sharing : screen sharing [on] - remote read/write [on]



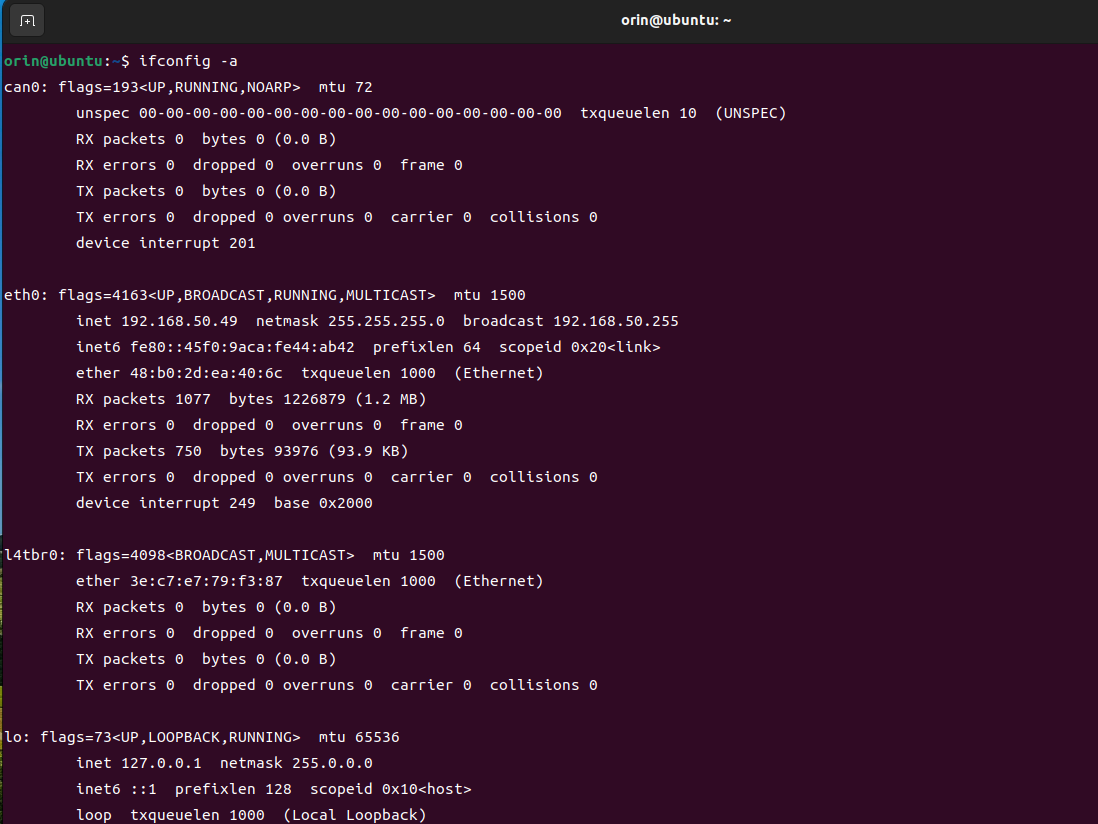
한번 접속하면 아래와 같이 주소가 남는다.

주소창에 “192.168.50.49” 🡸 아래 IP 주소 확인할 것.

Password를 물어 보면 [1].2에서 세팅한 wia



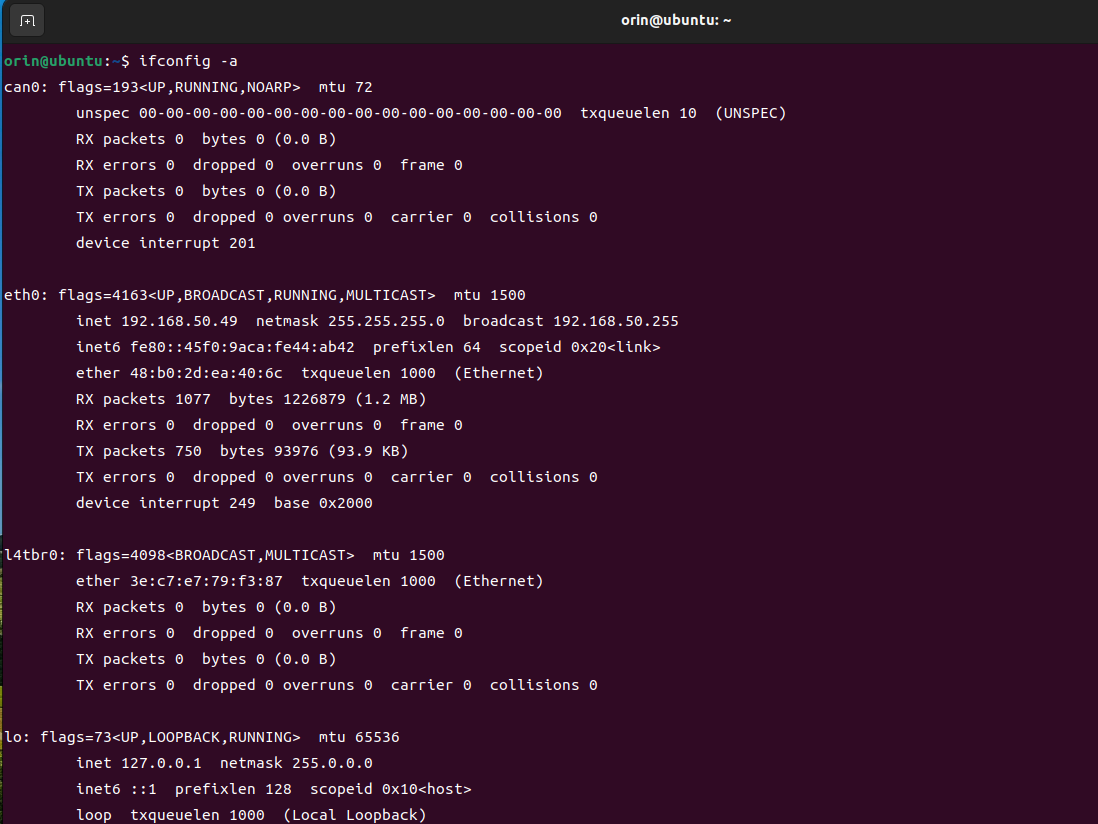
접속할 주소는 Orin NX에서 ifconfig –a 로 확인 “아래: eth0-192.168.50.49”



[2] CAN

1. CAN이 network 드라이버로 정상 인식했는지 확인

Can0로 정상 인식 확인



2. PinMux (안해줘도 동작할지도 모른다.)

sudo apt-get install busybox 🡺 busybox 설치

busybox devmem 0x0c303018 w 0xc458 🡺 CAN\_IN pin mux

busybox devmem 0x0c303010 w 0xc400 🡺 CAN\_OUT pin mux

3. To load the CAN kernel drivers

$ sudo modprobe can

$ sudo modprobe can\_raw

$ sudo modprobe fan\_fd

$ sudo modprobe mttcan 🡸 내가 가지고 있는 보드는 되지 않음/ 안해도 됨

==🡺 위 설정은 재부팅하면 다시 입력해야 함.

해결방법 : 시작시 실행하는 방법, 프로그램 내에서 실행하는 방법

프로그램 내에서 실행하는 것은 python에서는 잘 되는데 C에서는 잘 안되었음

1. 시작프로그램에 등록

rc-local service는 부팅이 끝나고 자동으로 root 권한 으로 시작하는 프로그램을 담고 있음

ls /dec/rc.local로 파일이 있는지 확인

없다면 sudo gedit /etc/rc/local 로 파일 만듬

#!/bin/bash

# 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.

exit 0

일반적으로 위와 같은데 만든 파일이라면 비어 있음

sudo modprobe can

sudo modprobe can\_raw

sudo modprobe can\_fd

sudo ip link set can0 up type can bitrate 500000 dbitrate 500000 berr-reporting on fd on <= bitrate, datarate 모두 500K, fd on

sudo ip link set up

Exit 0

입력하고 저장

sudo chmod +x /etc/rc.local

권한 설정 후 재부팅

1. sudo apt-get install can-utils 샘플 코드 및 can util 설치
2. 첨부한 파일 can0 폴더

Orin에 ftp나 usb를 이용 복사 및 압축 풀고

Cd can0

Make cansend

* 송신하는 샘플
* ./cansend할 것. 그냥 cansend하면 can-util이 실행됨
* ./cansend 123#12345678 하면 그 이후에 test 10 packet이 송신됨
* ./cansend 567##1AABBCCDDEEFF1122 하면 ; ##1 – BRS 옵션으로 송신

Make candump

* 수신하는 샘플
* 원래 표준은 read 함수를 사용하는데 recvmsg를 사용하였다.
* Recvmsg는 TX, Error Event도 모두 받을 수 있으므로 옵션을 보고
* 그에 맞는 처리를 할 수 있다.

1. 부가적으로 python 예제도 첨부.

py\_can\_rx.py, py\_can\_tx.py 🡺 python3 file.py로 실행

[3] UART

1. ls /dev/ttyTHS\*로 device driver 확인

Ls /dev/ttyUSB\*로 usb2serial driver 확인

2. 권한 설정 변경

sudo usermod -a -G dialout $USER

sudo chmod a+rw /dev/ttyTHS0

부팅시 자동 적용

sudo gedit /etc/udev/rules.d/99-ttyths.rules

# 아래 내용 입력 후 저장

KERNEL=="ttyTHS\*", MODE="0666"

[jetson linux]

<https://engcang.github.io/2024/01/06/nvidia_jetson_ttyTHS_permission.html>

[일반 linux]

<https://ahnbk.com/?p=1370>

https://blog.dev-colson.com/hardware/maxon\_motor\_udev\_rule/

같은 방법으로 권한 설정 변경을 한다.

uart\_rxtx.c 샘플 참조

// compile : gcc uart\_rxtx.c -o uart\_test

// execute : ./uart\_test

[4] GPIO

제일 쉽게 될 것 같았던 GPIO가 제일 잘 안되었다.

다행히 저번에 언급한 <https://github.com/Rubberazer/JETGPIO> git lib가 완벽하게 동작한다.

1. Download

git clone <https://github.com/Rubberazer/JETGPIO.git>

1. Install

압축풀고, 읽기 전용 해제하고 압축해제한 폴더에서

sudo make

sudo make install

1. gpioWrite(3, 1); // 40pin header의 3번핀 module connector 기준 191번핀을 1로
2. gpioWrite(15, 1);

15번 (GPIO12, 모듈 connector 218번핀)을 1로

level = gpioRead(7); //7번(GPIO9,211pin)의 Level Logic을 level 변수에 넣음.

* 샘플 예제에 interrupt도 있으니 용도에 맞게 사용하면 된다.

[4] UVC 관련

UVC는 단순 command windows에 display하는 것으로 OpenCV가 설치되어 있다는 가정하에서

동작시켰습니다.

영상지연 Delay는 NobufferingUVC가 가장 적었습니다.