



초소형(신용카드 크기) 저사양(Armv6: 700Mhz ~ 1000Mhz)으로 낮은 램 메모리(512MB)

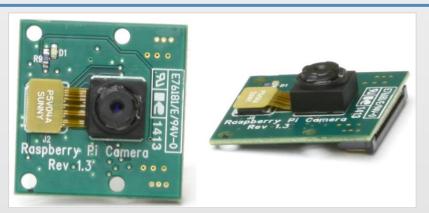
- 소형화 및 저전력 설계에 기반한 제품으로 최대 750mA를 사용이 가능함
- 고화질 FULL HD(1080p) 500만 화소의 라즈베리파이 전용 카메라가 출시됨
 - ➡ 적외선 촬영이 가능하며, 50mA의 전류가 소모됨



MMAL Driver를 사용하는 mjpg-streamer를 실행함

2) Camera 센서

- SONY IMX219 이미지 센서 탑재
- 고정 초첨렌즈
- 8M pixel
- 정지화상: 3280X2464
- 동영상: 1080p30fps, 720p60fps, 640X480p60fps/90fps





2) Camera 센서



1080p video recording to SD flash memory cards

Simultaneous output of 1080p live video via HDMI while recording

Removable lens

Adapters for Canon EF, and Nikon F mount lens interchange

Higher-resolution still-image capture (2592 x 1944 native resolution, 5 megapixels)

3) Camera 센서 인터페이스



MIPI Mobile Industry Processor Interface

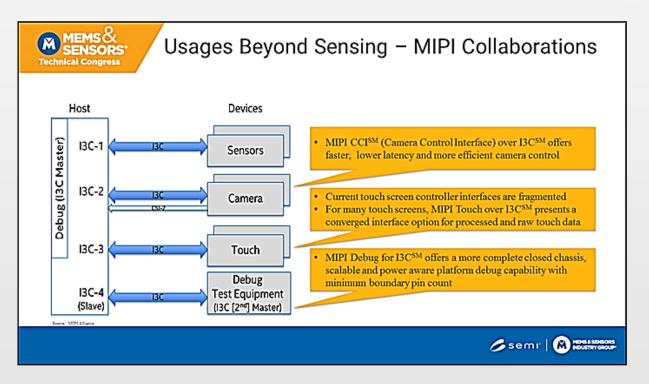
모바일 기기를 구성하는 구성 요소들 사이의 인터페이스를 규정하기 위하여 설립함

모바일 기기는 성능이 빠른 속도로 향상되고 있으며, 다양한 기능과 성능의 구성 요소들의 결합으로 인하여 인터페이스에 관한 협력이 필요함

ARM, 인텔, 노키아, 삼성, ST 마이크로 일렉트로닉스 및 텍사스 인스트루먼트사에 의해서 설립되었고, 그 외에도 LG전자, 매그나칩이 참여 중임

3) Camera 센서 인터페이스

MIPI Mobile Industry Processor Interface



4) CCD Camera 센서 인터페이스





인터페이스에서 카메라는 CSI Transmitter와 CCI Slave로써 동작함



4) CCD Camera 센서 인터페이스

카메라를 컨트롤하는 디바이스는 I2C 버스를 사용하여 카메라를 컨트롤함

카메라에서 촬영한 이미지는 S-LVDS를 통하여 받음

4) CCD Camera 센서 인터페이스



LVDS (Low-Voltage Differential Signaling)

고속 디지털 인터페이스를 하기 위하여 고속 데이터 전송 속도, 낮은 전력 소비 및 뛰어난 노이즈 내성을 갖도록 설계함

LVDS는 2개의 페어라인에 의해서 신호를 전달함

드라이버에서는 2개의 드라이브 신호에 의해서 전달하고 리시버에서는 100옴의 터미네이션 저항을 통하여 받음

신호 변화에 따른 전압의 변화량이 적기 때문에 고속의 신호 전송이 가능함

인접한 두 개의 페어라인에 흐르는 전류의 방향이 다르기 때문에 그로 인한 자계가 상쇄되어 전자파 방출이 적음



mjpg-streamer 컴파일을 위해 영상관련 라이브러리 및 cmake 패키지를 설치함

```
🔊 🕮 📵 pi@raspberrypi: ~
Reading state information... Done
git is already the newest version.
git set to manually installed.
The following extra packages will be installed:
 cmake-data emacsen-common imagemagick-common libdivulibre-text
 libdivulibre21 libexiv2-12 libilmbase6 liblensfun-data liblensfun0
  liblgr-1-0 libmagickcore5 libmagickcore5-extra libmagickwand5 libnetpbm10
 libopenexr6 libwmf0.2-7 libxmlrpc-core-c3 netpbm ufraw-batch
Suggested packages:
 imagemagick-doc autotrace enscript ffmpeg gimp gnuplot grads hp2xx html2ps
 libwmf-bin mplayer povray radiance sane-utils texlive-base-bin transfig
 exiv2 ufraw
The following packages will be REMOVED:
 libjpeg62-dev
The following NEW packages will be installed:
 cmake cmake-data emacsen-common imagemagick imagemagick-common
  libdjvulibre-text libdjvulibre21 libexiv2-12 libilmbase6 libjpeg8-dev
 liblensfun-data liblensfun0 liblqr-1-0 libmagickcore5 libmagickcore5-extra
 libmagickwand5 libnetpbm10 libopenexr6 libwmf0.2-7 libxmlrpc-core-c3 netpbm
 ufraw-batch
0 upgraded, 22 newly installed, 1 to remove and 1 not upgraded.
Need to get 13.0 MB of archives.
After this operation, 31.9 MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

pi@openmake ~ \$ sudo apt-get install git cmake libjpeg8-dev imagemagick-y



videodev2.h 헤더파일을 링크함

mjpg-streamer를 컴파일 진행 시에는 요구하는 videodev.h 헤더파일을 변경된 videodev2.h 파일로 링크함

pi@openmake ~ \$ sudo ln-s/usr/include/linux/videodev2.h /usr/include/linux/videodev.h



mjpg-stream svn 서버와 raspberrypi pi-cam 소스 공개 배포처를 통해 sjdmkd 소스를 내려받고 mmal driver에 맞추어 일부 내용을 수정함

pi@raspberrypi:~/project \$ git clone https://github.com/jacksonliam/mjpg-streamer.git

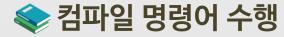
라즈베리파이는 이미 많은 유저들에 의해 수정된 소스가 git hub 또는 bitbucket 서버를 배포 중임

pi@raspberrypi: ~ \$ git clone https://github.com/liamfraser/mjpg-streamer pi@raspberrypi: ~ \$ cd ~/mjpg-streamer/mjpg-streamer-experimental

mjpg−streamer 관련 디렉토리 이동

```
🔊 🔘 🕦 pi@raspberrypi: ~/mjpg-streamer/mjpg-streamer-experimental
pi@raspberrypi ~ $ git clone https://github.com/liamfraser/mjpg-streamer
Cloning into 'mipg-streamer'...
remote: Reusing existing pack: 1829, done.
remote: Counting objects: 168, done.
remote: Compressing objects: 100% (132/132), done.
remote: Total 1997 (delta 34), reused 162 (delta 31)
Receiving objects: 100% (1997/1997), 3.30 MiB | 777 KiB/s, done.
Resolving deltas: 100% (1157/1157), done.
pi@raspberrypi ~ $ ls
Desktop mjpg-streamer ocr_pi.png python_games
pi@raspberrypi ~ $ cd mjpg-streamer
pi@raspberrypi ~/mjpg-streamer $ ls
mjpg-streamer-experimental README.md
pi@raspberrypi ~/mjpg-streamer $ cd mjpg-streamer-experimental
pi@raspberrypi ~/mjpg-streamer/mjpg-streamer-experimental $
```

```
🔊 🗐 📵 pi@raspberrypi: ~/mjpg-streamer/mjpg-streamer-experimental
make[3]: Leaving directory `/home/pi/mjpg-streamer/mjpg-streamer-experimental/p
ugins/input raspicam/build'
make[3]: Entering directory `/home/pi/mjpg-streamer/mjpg-streamer-experimental/p
lugins/input_raspicam/build'
[100%] Building C object CMakeFiles/input raspicam.dir/input raspicam.c.o
/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input raspicam/input r
aspicam.c: In function 'worker thread':
/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input_raspicam/input_r
aspicam.c:722:9: warning: assignment makes pointer from integer without a cast |
enabled by default]
  nking C shared library libinput raspicam.so
make[3]: Leaving directory '/home/pi/mjpg-streamer/mjpg-streamer-experimental/pl
ugins/input raspicam/build'
[100%] Built target input raspicam
make[2]: Leaving directory `/home/pi/mjpg-streamer/mjpg-streamer-experimental/pl
ugins/input_raspicam/build'
make[1]: Leaving directory `/home/pi/mjpg-streamer/mjpg-streamer-experimental/pl
ugins/input raspicam/build'
cp plugins/input raspicam/build/libinput raspicam.so ./input raspicam.so
       0m24.993s
        0m19.400s
        0m2.800s
pi@raspberrypi ~/mjpg-streamer/mjpg-streamer-experimental $
```



pi@openmake ~/mjpg-streamer/mjpg-streamer-experimental/ \$ make clean all



mjpg-streamer를 실행할 수 있는 환경이 갖추어졌으면 라즈베리파이에서 mjpg-streamer를 손쉽게 이용할 수 있도록 스크립트를 작성하여 웹 스트리밍을 구현함

pi@openmake ~/mjpg-streamer/mjpg-streamer-experimental/ \$ cd ~ pi@openmake ~ \$ sudo vi mjpg.sh



```
export STREAMER_PATH=$HOME/mjpg-streamer/mjpg-streamer-experimental
```

export LD_LIBRARY_PATH=\$STREAMER_PATH

\$STREAMER_PATH/mjpg_streamer -i "input_raspicam.so-d 200" -o "output_http.so-w

\$STREAMER_PATH/www"



pi@openmake ~ \$ sh mjpg.sh

```
🕽 🗐 🤘 pi@raspberrypi: ~
pi@raspberrypi ~ $ sh mjpg.sh
MJPG Streamer Version: svn rev: Unversioned directory
DBG(/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input_raspicam/in
put raspicam.c, input init(), 116): argv[0]=raspicam input plugin
DBG(/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input raspicam/in
put raspicam.c, input init(), 116): argv[1]=-d
DBG(/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input raspicam/in
put_raspicam.c, input_init(), 116): argv[2]=200
DBG(/home/pi/mjpg-streamer/mjpg-streamer-experimental/plugins/input_raspicam/in
put raspicam.c, input init(), 183): case 6, 7
i: delav..... 200
i: resolution.....: 640 x 480
i: camera parameters....:
Sharpness 0, Contrast 0, Brightness 50
Saturation 0. ISO 400. Video Stabilisation No. Exposure compensation 0
Exposure Mode 'auto', AWB Mode 'auto', Image Effect 'none'
Metering Mode 'average'. Colour Effect Enabled No with U = 128. V = 128
Rotation 0, hflip No, vflip No
o: www-folder-path...: /home/pi/mjpg-streamer/mjpg-streamer-experimental/www/
o: HTTP TCP port....: 8080
o: username:password.: disabled
o: commands....: enabled
```



mjpg-streamer 라즈베리파이 카메라를 통해 jpeg 파일로 생성된 이미지를 파노라마처럼 스틸샷으로 만들어 빠른 속도로 재생함으로써 영상으로 보이도록 함

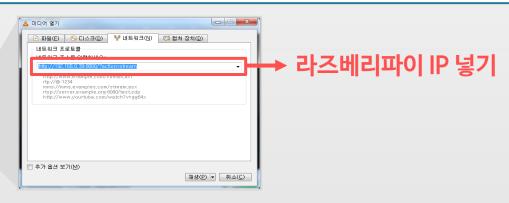
스틸샷 영상은 "http://라즈베리파이아이피:8080/?action=stream" 파일로 저장이 됨

웹 이외에도 VLC Player 등의 불을 이용해 스트리밍 영상을 확인할 수 있음

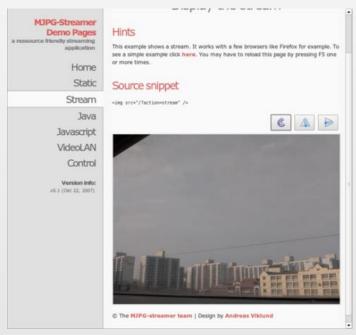
- 🔷 mjpg-streamer 웹 스트리밍 영상확인
 - ▶ VLC Player 다운로드 및 설정 (http://www.videolan.org/)

미디어 → 네트워크 스트림 열기 → 네트워크 주소에 라즈베리파이 IP 넣기 (http://라즈베리파이아이피:8080/?action=stream)

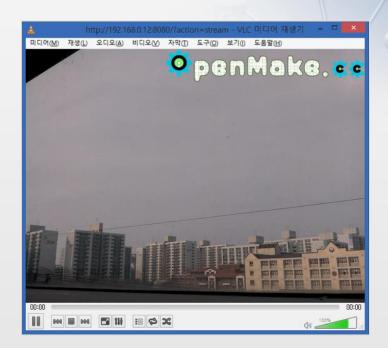




❤ mjpg-streamer 웹 스트리밍 영상확인



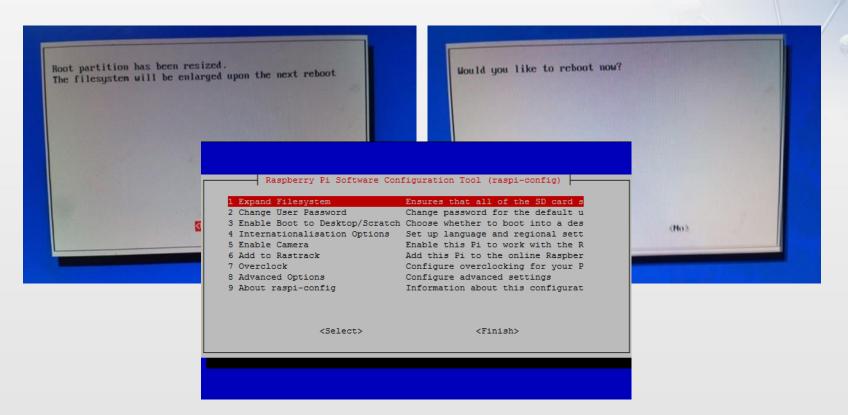
[Web을 통한 영상]



[VLC Player를 통한 영상]

- ◆ 라즈베리파이 환경설정(raspi-config)
 - Change User Password
 - 라즈베리파이의 비밀번호 수정 탭
 - 운영체제 설치 시 초기 비밀번호는 raspberry이며, 탭을 선택한 후 변경해줌
 - Configure Keyboard
 - Generic 105-key(Intl) PC가 디폴트됨
 - 3 Expand Filesystem
 - SD카드 사용 영역을 설정하기 위한 탭
 - SD카드의 모든 영역을 사용하기 위해서 반드시 설정해줘야 하며, 탭을 선택하여 OK를 누른 후 재부팅하면 적용됨

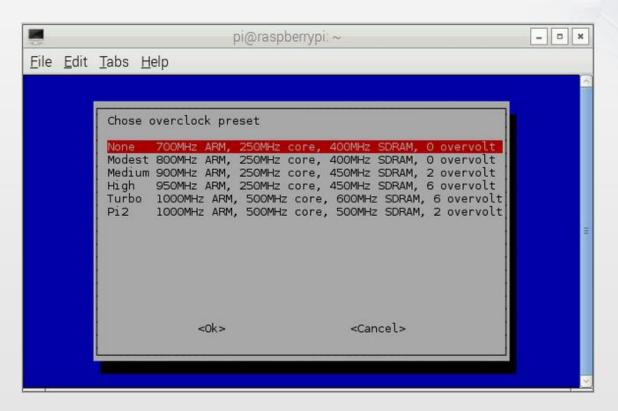
◆ 라즈베리파이 환경설정 (raspi-config)



- ◆ 라즈베리파이 환경설정(raspi-config)
 - 4 Enable Boot to Desktop/Scratch
 - 라즈베리파이를 켰을 때, 데스크탑, 커맨드 라인, 스크래치 화면 중 어느 화면으로 부팅할지 선택이 가능함
 - ※ 데스크탑으로 부팅 시, 아이디/비밀번호 입력을 안 해도 되니 설정을 해두면 간편해짐
 - 5 Internationalisation Options
 - 나라(국제화) 설정: 언어, 시간, 키보드 레이아웃 설정이 가능함
 - 한국 언어 설정 방법: Change Locale → ko_KR.UTF-8 UTF-8
 - 한국 시간 설정 방법 : Change Timezone → Asia → Seoul

- ◆ 라즈베리파이 환경설정(raspi-config)
 - 6 Enable Camera
 - 라즈베리파이 카메라 사용여부를 설정하는 탭
 - Add to Rastrack
 - 전 세계에 라즈베리파이 사용자의 분포를 알려주는 탭
 - (8) Overclock
 - 정해진 클럭을 오버시키는 기능을 함
 - 사용하면 속도가 빨라질 수는 있으나, 발열이 생기고 기기의 안전성을 저해함
 - 방열판을 이용하면 발열을 줄일 수 있음

◆ 라즈베리파이 환경설정(raspi-config)



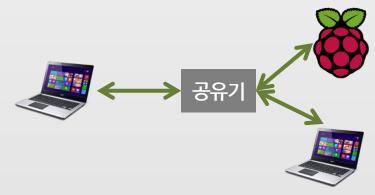


DHCP 고정할당 설정

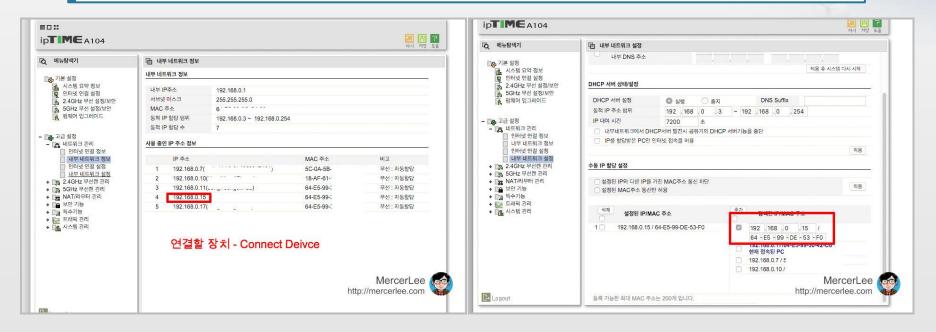
■ 기기 재시작에 따른 IP 주소 변경을 방지하고, 특정 MAC 주소에 대해 내부 IP를 고정할 수 있음

포트포워딩

SSH로 라즈베리파이를 외부에서 접속하려면 내부 네트워크 중
 SSH 포트 접속을 허용하는 기기로 인터넷망에서 포워딩할 수 있어야 함

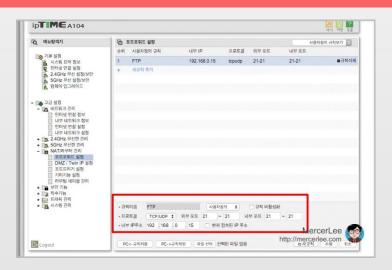


AP 관리자 모드로 접속하면 현재 raspberrypi라는 기기에 어떤 IP 주소가 할당되었는지 확인할 수 있음



포트 포워딩을 PuTTY 같은 SSH 서비스로 라즈베리파이에 접근하고 싶거나, FTP 연결을 외부 네트워크에서도 연결을 희망한다면?

네트워크 외부에서 공유기에 접속했을 때 특정 포트로 접근을 하면, 요청한 포트 번호에 따라서 내부 네트워크에 연결된 기기들에 할당된 내부 IP 주소로 포워딩 함



규칙이름은 원하는대로 정함

※ 22번 포트이면 'ssh~'로 작성하는게 좋음

내부 IP 주소는 라즈베리파이가 할당받은 IP 주소임

FTP 서버 설치

 내부 포트와 외부 포트에 별다른 일이 없다면 동일하게 세팅하면 되고, 기관에서 보안상 필요한 경우 이를 변경할 필요도 있음

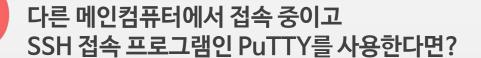
❖ 컴파일 명령어 수행

pi@openmake ~/ \$ sudo apt-get install proftpd-basic pi@openmake ~/ \$ vi /etc/proftpd/proftpd.conf

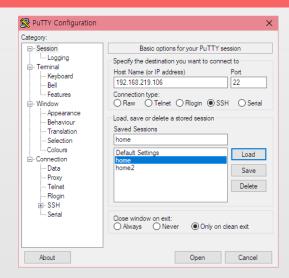
proftpd.conf 파일을 열어 "#DefaultRoot ~" 부분을 찾아 "DefaultRoot ~" 로 변경함

pi@openmake ~/ \$ sudo service proftpd restart

| MultilineRFC2228 | on |
|----------------------------------|-----------------|
| DefaultServer | on |
| ShowSymlinks | on |
| | |
| TimeoutNoTransfer | 600 |
| TimeoutStalled | 600 |
| TimeoutIdle | 1200 |
| | |
| DisplayLogin | welcome.msg |
| DisplayChdir | .message true |
| ListOptions | "-1" |
| | |
| DenyFilter | *.*/ |
| | |
| # Use this to jail all users in | their homes |
| <mark>D</mark> efaultRoot | ~ |
| | |
| # Users require a valid shell li | |
| # Use this directive to release | that constrain. |
| | |



- 라즈베리파이의 IP는 공유기가 할당한 내부 IP 주소를 적어도 되고, 포트 포워딩을 마쳤으므로 공유기 자체가 가지고 있는 외부 IP 주소를 적어도 됨
- 22포트로 접근을 하면자동으로 라즈베리파이에 연결됨







\$ sudo apt-get update



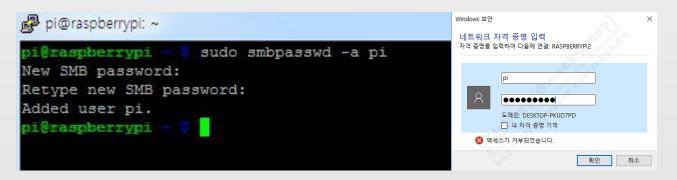
apt-get install samba samba-common-bin

```
pi@raspberrypi - 🔻 sudo apt-get install samba samba-common-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
 tdb-tools
Suggested packages:
 openbsd-inetd inet-superserver smbldap-tools ldb-tools ctdb
The following NEW packages will be installed:
 samba samba-common-bin tdb-tools
 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.
Need to get 6,398 kB of archives.
After this operation, 36.1 MB of additional disk space will be used.
Do you want to continue [Y/n]? v
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main samba armhf 2:3.6.6-6+deb7u3 [3,318 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main samba-common-bin armhf 2:3.6.6-6+deb7u3 [3,054 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main tdb-tools armhf 1.2.10-2 [25.9 kB]
Fetched 6,398 kB in 5s (1,243 kB/s)
```

🧇 samba 설치

\$ sudo su

- ⇒ #을 사용해도 되고 sudo명령어를 계속 사용하여도 됨
- # smbpasswd -a pi
- ⇒ # a 옵션을 사용하여 사용자를 추가함





Vi/etc/samba/smb.conf

```
pi@raspberrypi: ~
  GNU nano 2.2.6
                                                          File: /etc/samba/smb.conf
        cdrom share is accesed. For this to work /etc/fstab must contain
        an entry like this:
        /dev/scd0
                    /cdrom iso9660 defaults, noauto, ro, user
 The CD-ROM gets unmounted automatically after the connection to the
  If you don't want to use auto-mounting/unmounting make sure the CD
        is mounted on /cdrom
    preexec = /bin/mount /cdrom
    postexec = /bin/umount /cdrom
[pi]
comment = raspberry pi folder
path = /
valid user = pi
writable = yes
browseable = yes
```



```
# vi/etc/samba/smb.conf
  [global]
  workgroup = WORKGROUP // 네트워크 공유 그룹 설정
  encrypt password = true // 암호 설정
  unix charset = utf-8
  charset = utf-8
  #dos charset = cp949
  [pi] # 추가한 계정
  comment = Raspberry pi
  path = /home #공유할 폴더 경로
```

🧇 samba 설치

```
valid user = pi #smbpasswd 에서 설정한 유저
browseable = yes
comment = max/samba
read only = no #읽기 전용 여부
writable = yes #쓰기 권한 여부
printable = no
public = yes
guest ok = no #게스트 계정 접근 허용 여부
```



service samba restart or # /etc/init.d/samba restart



윈도우에 접속하여 확인함

- 윈도우즈 내컴퓨터의 왼쪽 메뉴에 "네트워크" 부분을 클릭하면 라즈베리파이에 설정된 hostname으로 컴퓨터가 있음을 확인함
- 실행창에서 ₩₩라즈베리파이 ip 입력으로 접속하여 확인함

❤ ifconfig로 라즈베리파이 ip 확인



- 윈도우에서 리눅스 (라즈베리파이) 폴더가 공유됨
- 복사가 가능하여 작업하기 수월함

❤ ifconfig로 라즈베리파이 ip 확인

