Google API기반의 Raspberry Pi 프로젝트

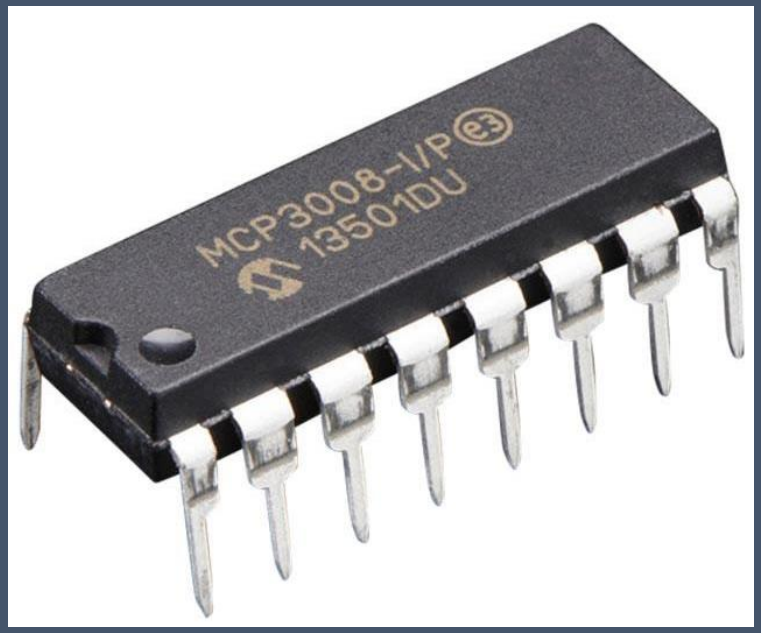
dorek99@naver.com
2019.05.02

SPI

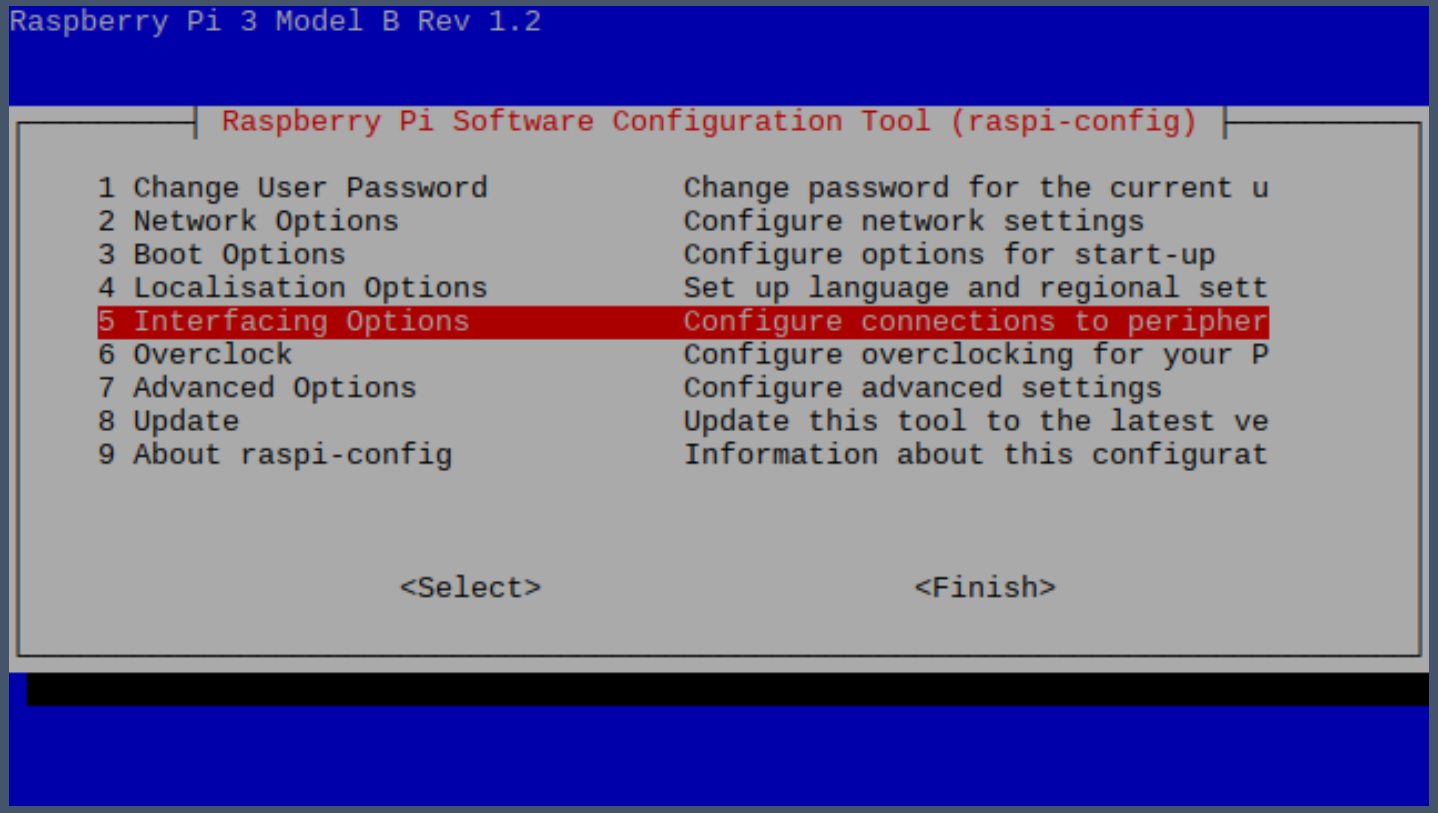
- Serial Peripheral Interface (SPI) - MCP3008
- 주변 인터페이스와 직렬 방식으로 통신하는 장치
- 아날로그 -> 디지털 컨버터(ADC) 및 포트 확장 칩 등으로 주로 사용
- 기본적으로 4개의 시그널 핀으로 구성
  - MOSI : Master Out Slave In
  - MISO : Master In Slave Out
  - SS : Slave Select
  - SCK: Serial Clock
- 마스터(라즈베리파이)는 하나이지만 슬레이브(주변기기, 각종 센서)는 복수 개를 사용할 수 있음

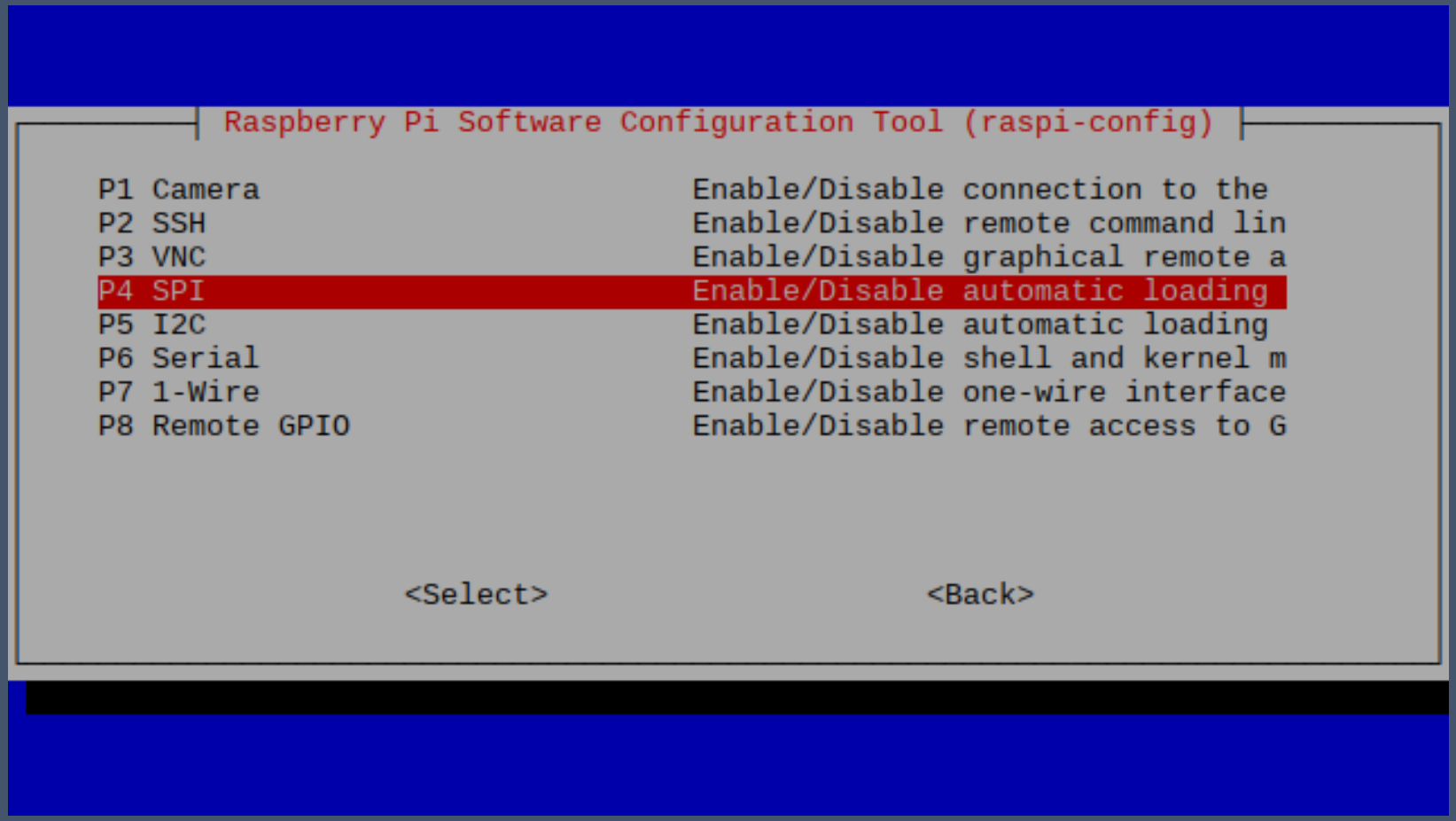
SPI

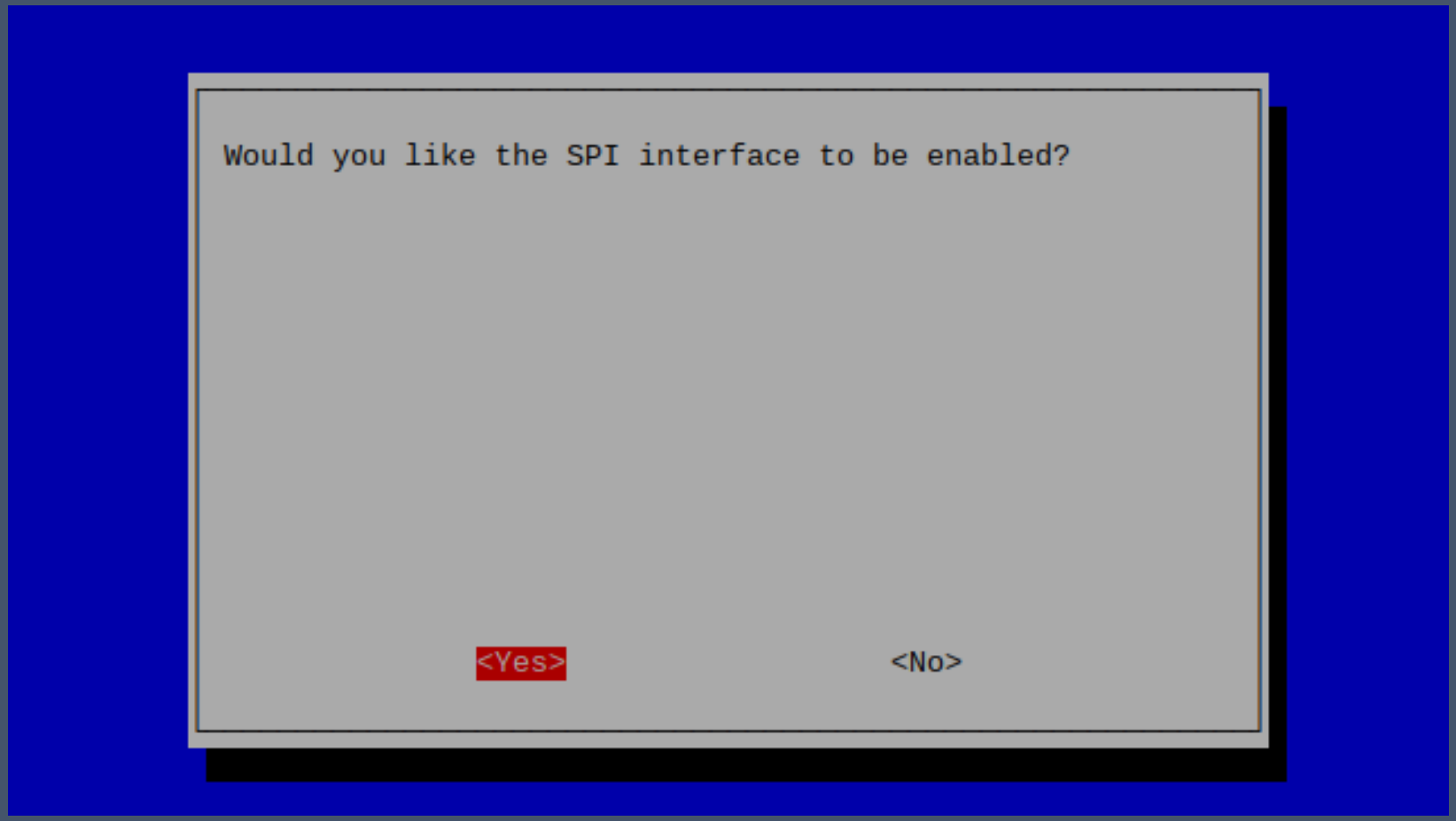
- SPI로 가장 일반적으로 사용되는 MCP3008은 데이터 10비트, 아날로그 입력 8채널을 지원함.

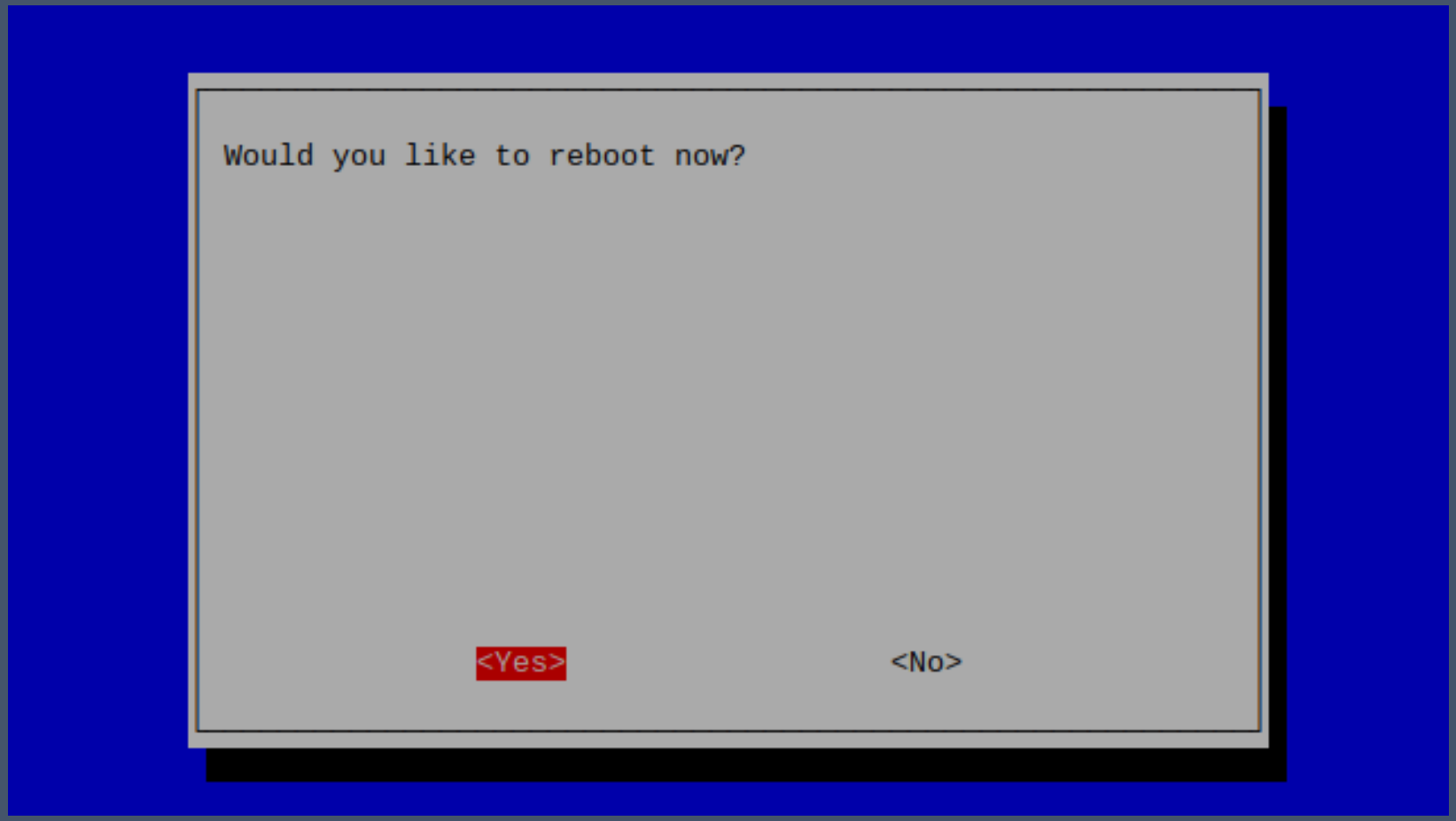


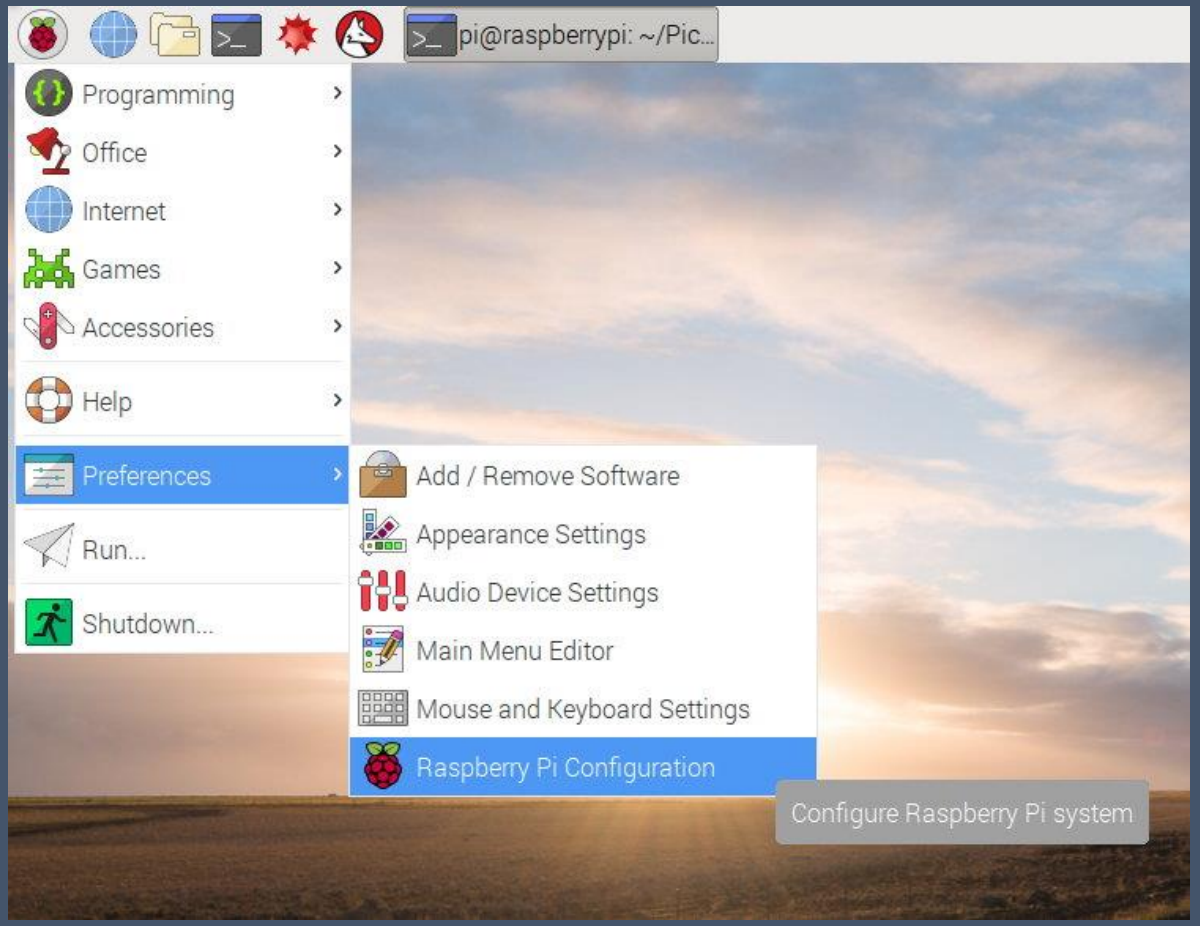
```
sudo raspi-config
```

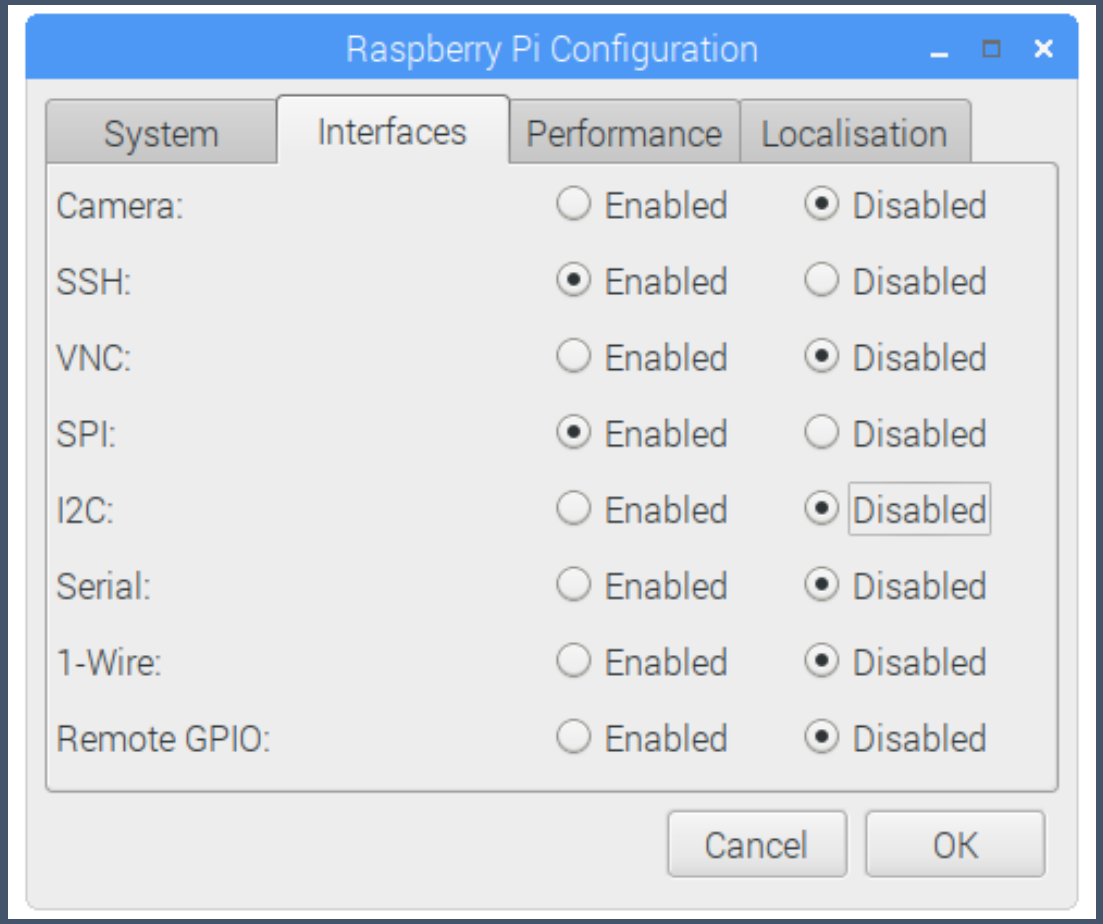


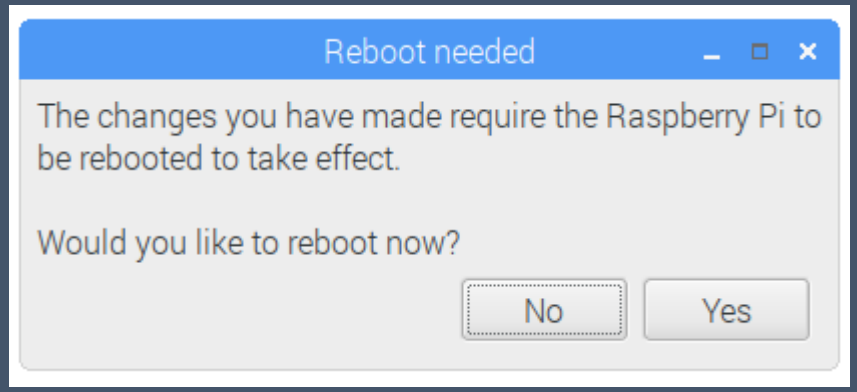

Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera Enable/Disable connection to the
P2 SSH Enable/Disable remote command lin
P3 VNC Enable/Disable graphical remote a
P4 SPI Enable/Disable automatic loading
P5 I2C Enable/Disable automatic loading
P6 Serial Enable/Disable shell and kernel m
P7 1-Wire Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to G
<Select>
<Back>


Would you like the SPI interface to be enabled?
<Yes>
<No>


Would you like to reboot now?
<Yes>
<No>


pi@raspberrypi: ~/Pic...
Programming
Office
Internet
Games
Accessories
Help
Preferences
Run...
Shutdown...
Add / Remove Software
Appearance Settings
Audio Device Settings
Main Menu Editor
Mouse and Keyboard Settings
Raspberry Pi Configuration
Configure Raspberry Pi system


Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera: Enabled Disabled
SSH: Enabled Disabled
VNC: Enabled Disabled
SPI: Enabled Disabled
I2C: Enabled Disabled
Serial: Enabled Disabled
1-Wire: Enabled Disabled
Remote GPIO: Enabled Disabled
Cancel
OK


Reboot needed
The changes you have made require the Raspberry Pi to be rebooted to take effect.
Would you like to reboot now?
No
Yes

```
• sudo apt-get update
• sudo apt-get upgrade
• sudo nano /boot/config.txt
• dtparam=spi=on
• sudo reboot
• lsmod
```

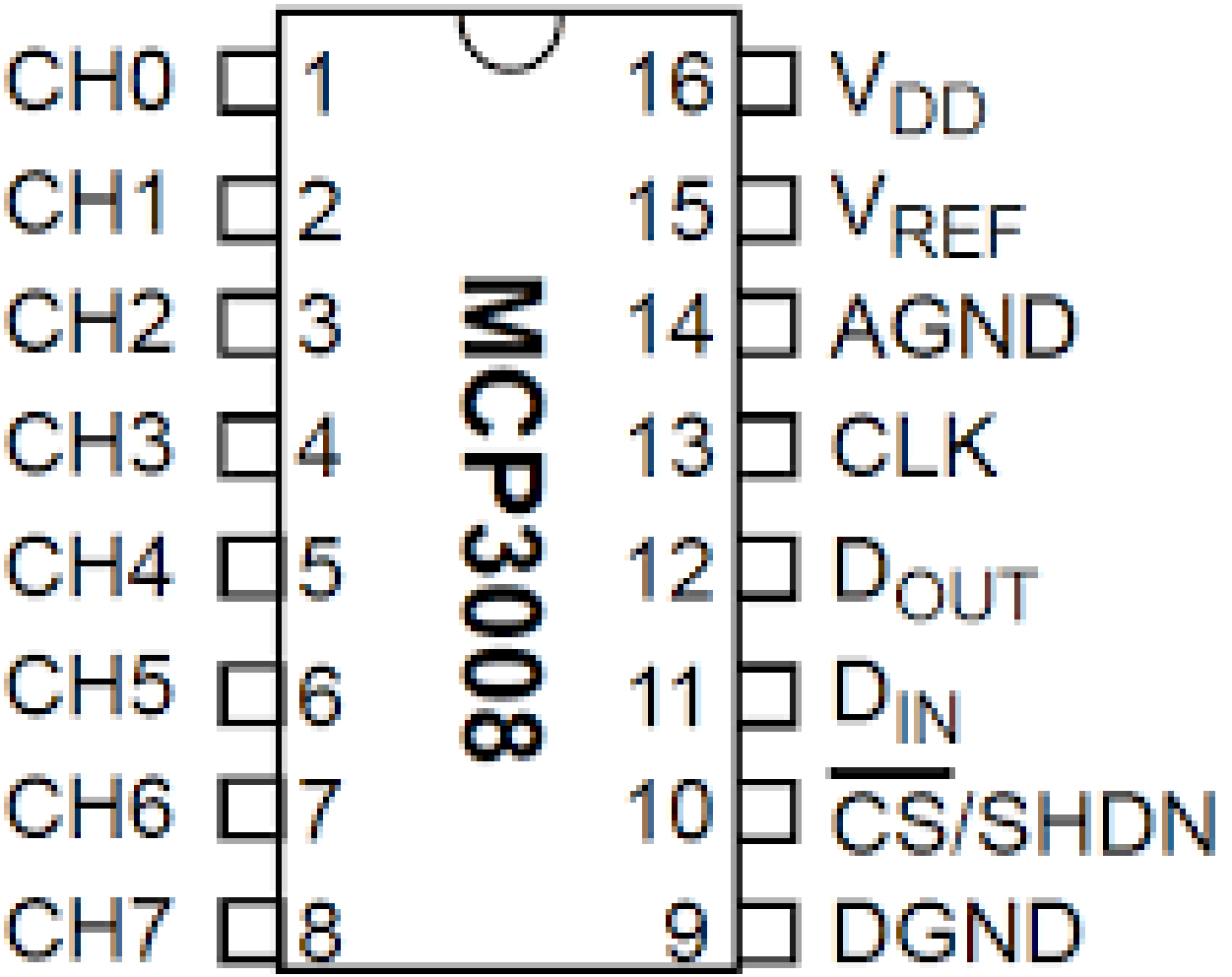
lsmod 입력 후 출력되는 리스트 중에 “spi_bcm2708” 또는 “spi_bcm2835” 가 나타나야함.

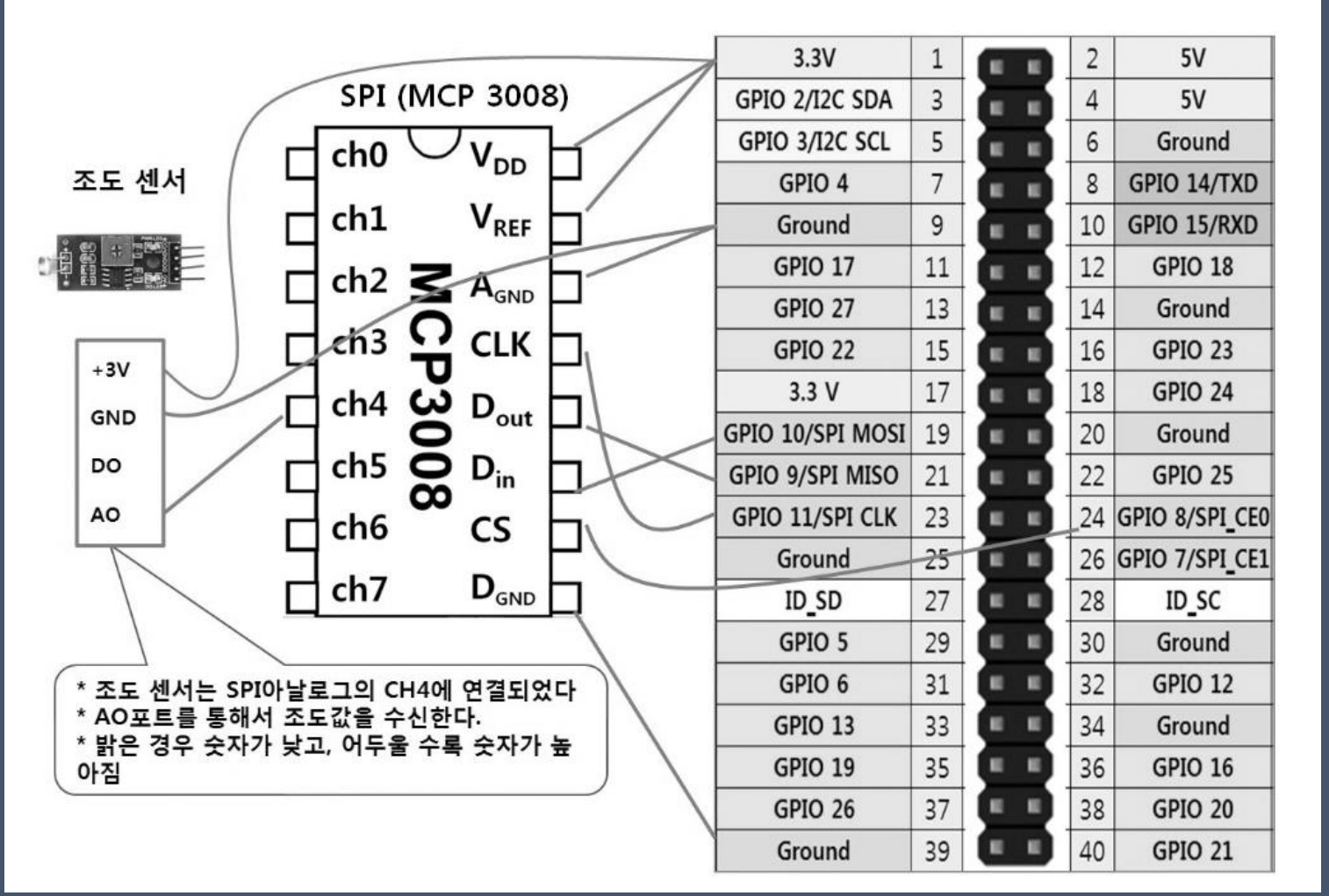
```
• sudo apt-get install -y python-dev python3-dev
• sudo apt-get install -y python-spidev python3-spidev
• cd Desktop
• git clone https://github.com/Gadgetoid/py-spidev.git
• cd py-spidev
• sudo python setup.py install
• sudo python3 setup.py install
```

조도 센서

- SPI와 조도 센서를 연결하여 아날로그 방식으로 현재 조도에 대한 센싱 값을 측정하여 이에 따라 LED 밝기 조절

- https://github.com/HakjunLee1/raspberry
- 에서 spi_cds_led.py 다운로드


CH0
1
CH1
2
CH2
3
CH3
4
CH4
5
CH5
6
CH6
7
CH7
8
MCP3008
16
V_{DD}
15
V_{REF}
14
AGND
13
CLK
12
D_{OUT}
11
D_{IN}
10
$\overline{CS}$/SHDN
9
DGND

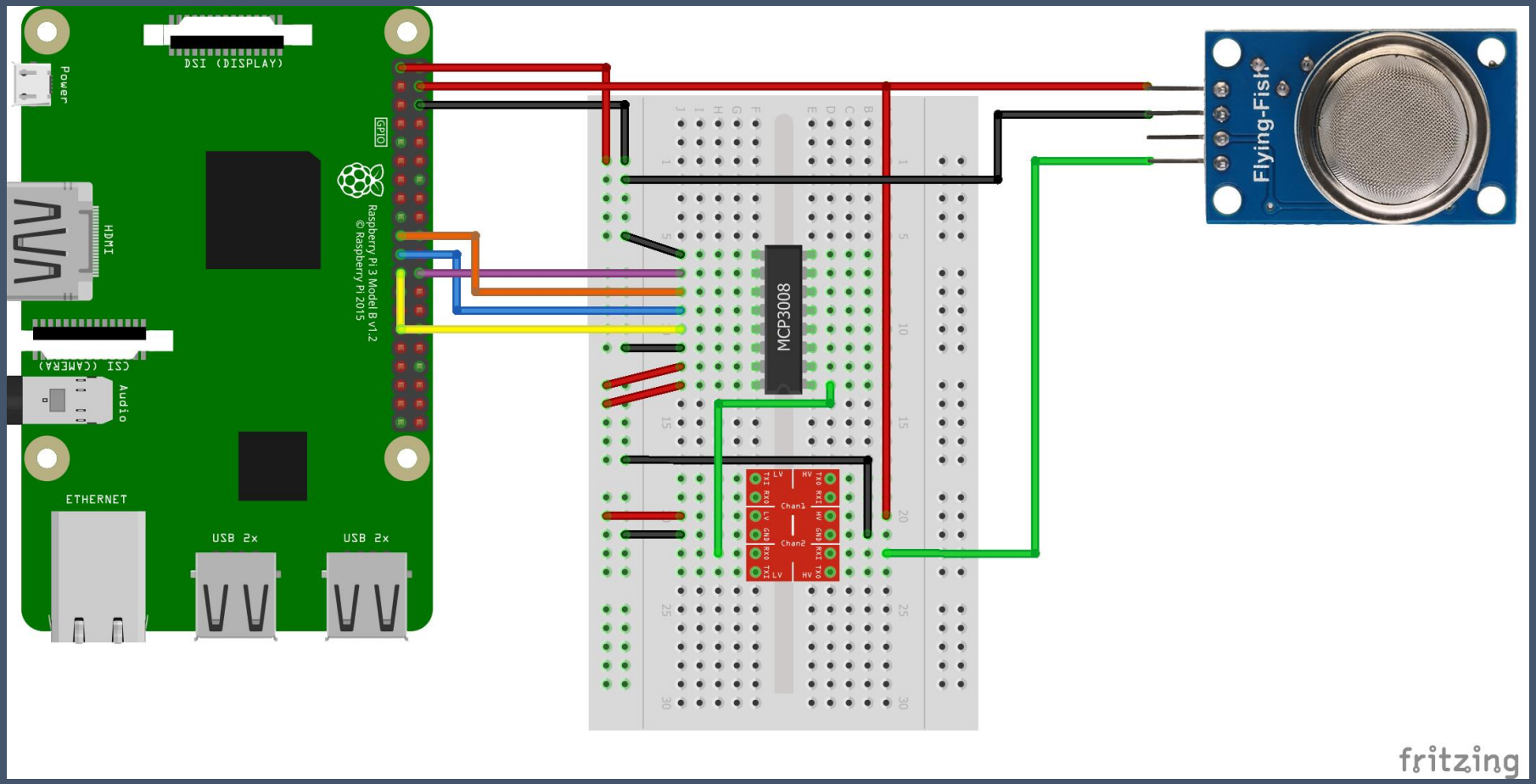


- 조도센서 ch1 에 연결
- LED는 GPIO21번 연결, + 21 GPIO Pin, - GND
  (소스코드에서는 21번 pin을 가리키나 편의에 따라 변경 가능)

알콜센서

- 알코올, 에탄올의 농도에 따라 출력 전압 증가
- 가스가 감지되면 뒷면에 LED 불빛 ON
- VCC ↔ 5.0V
- GND ↔ power supply ground
- AOUT ↔ MCU.IO (analog output)
- DOUT ↔ MCU.IO (digital output)

알콜센서

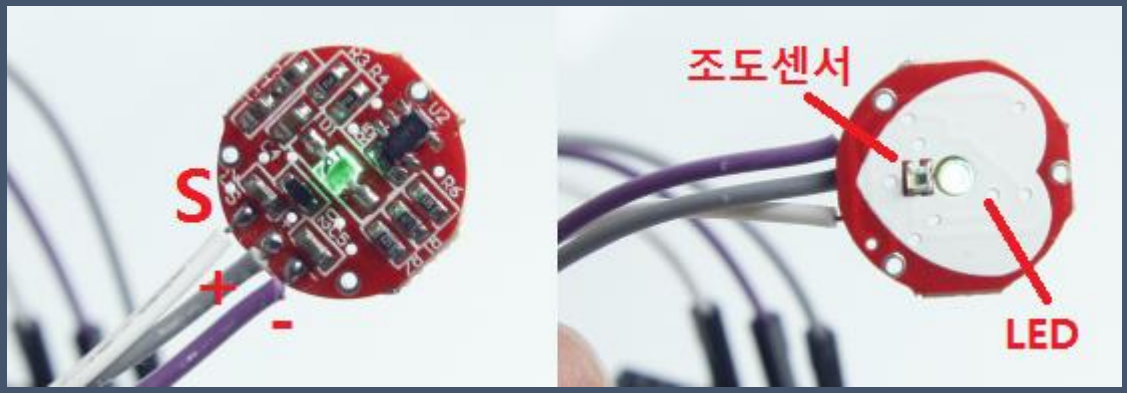

DSI (DISPLAY)
Power
GPIO
HDMI
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
MCP3008
Chan1
Chan2
LV
HV
Flying-Fish
fritzing

알콜센서

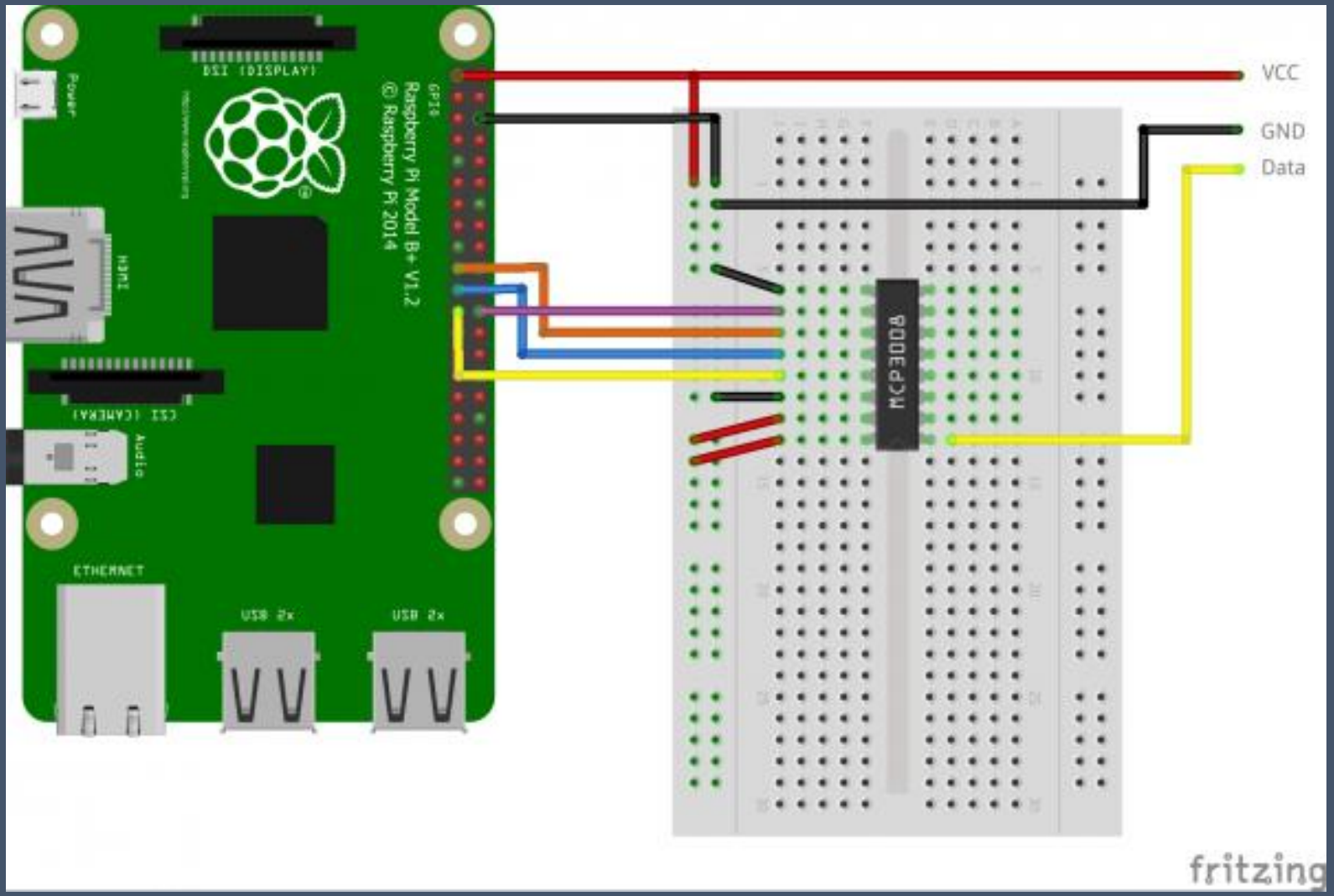
- 참고 사이트
  - https://tutorials-raspberrypi.com/configure-and-read-out-the-raspberry-pi-gas-sensor-mq-x/
- 다운로드 소스코드
- cd Desktop
- git clone https://github.com/tutRPi/Raspberry-Pi-Gas-Sensor-MQ
- cd Raspberry-Pi-Gas-Sensor-MQ
- sudo python example.py

심박센서

- LED로 빛을 내뿜고 반사되는 빛의 양을 측정하여 이를 전압으로 바꾸어 출력함.
- 피부 표면에서, 심장이 박동될 때 혈류가 달라짐으로 인해 빛의 반사양이 바뀜
- 이는 작은 변화이지만 크고 작음을 측정하면 심박을 알아낼 수 있다.
- 센싱 환경에서 잡음이 다소 끼어있으므로, 빛의 변화량을 잘 파악해야 정확한 측정이 가능하지만 3500원밖에 안하는 이 센서에 ....



심박센서


VCC
GND
Data
MCP3008
DSI (DISPLAY)
Power
HDMI
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
Raspberry Pi Model B+ V1.2
© Raspberry Pi 2014
fritzing

심박센서

- sudo apt-get update
- sudo apt-get upgrade sudo apt-get install git

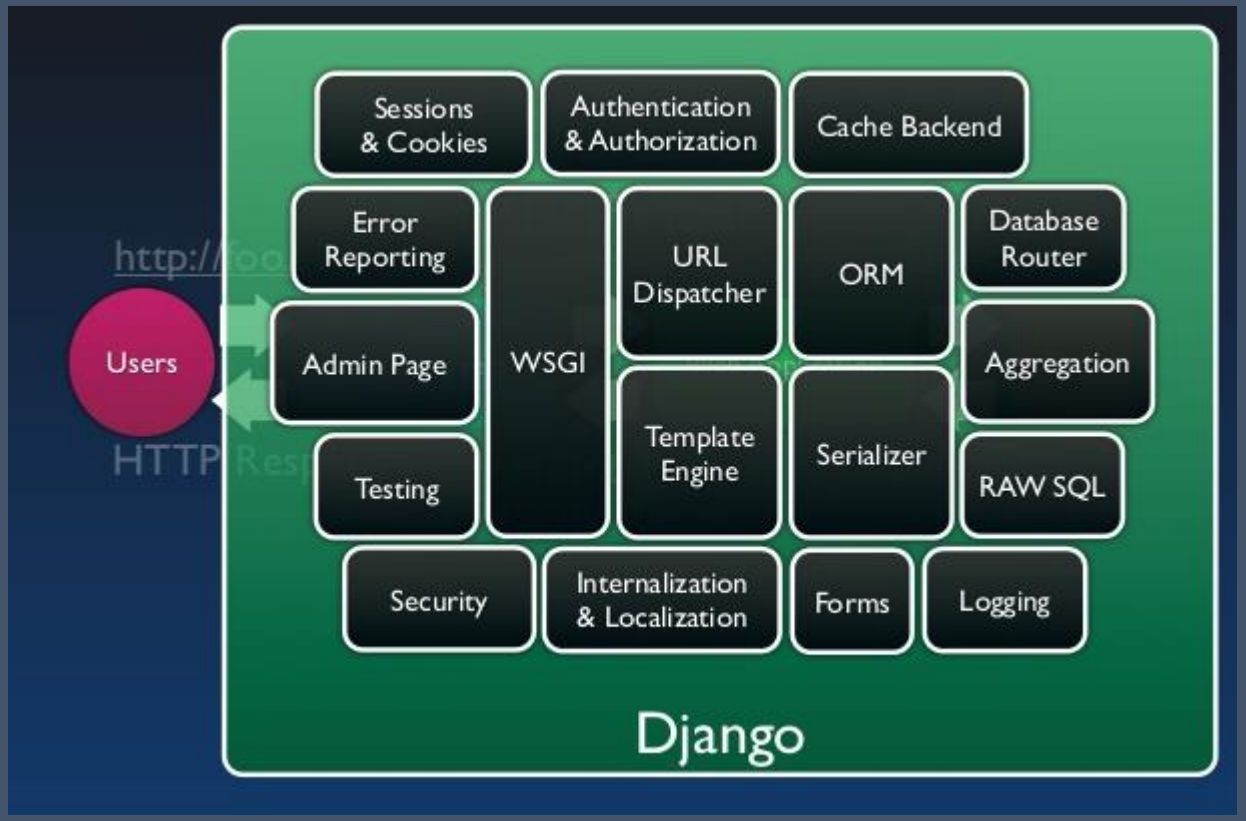
- cd Desktop

- git clone https://github.com/tutRPi/Raspberry-Pi-Heartbeat-Pulse-Sensor

- cd Raspberry-Pi-Heartbeat-Pulse-Sensor/
- sudo python example.py

장고(Django)

- 파이썬 기반 웹서버 프레임워크



장고(Django)

- **Step 1: 장고 설치하기**
- **1. 패키지 업데이트**
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt-get autoremov
- **2. 파이썬 Mysql 데이터베이스 설치**
  - sudo apt-get install python-mysqldb -y
- **3. LAMP 서버 설치하기**
  - sudo apt-get install apache2 -y
  - sudo apt-get install mysql-server mysql-client -y
  - sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-mysql -y
  - sudo service apache2 restart

장고(Django)

- **4. Phpmyadmin 설치후 아파치 구성**
  - sudo apt-get install phpmyadmin -y
  - sudo nano /etc/apache2/apache2.conf
  - 위 코드 실행 후, 빈곳에 다음의 코드 넣습니다.
    - Include /etc/phpmyadmin/apache.conf

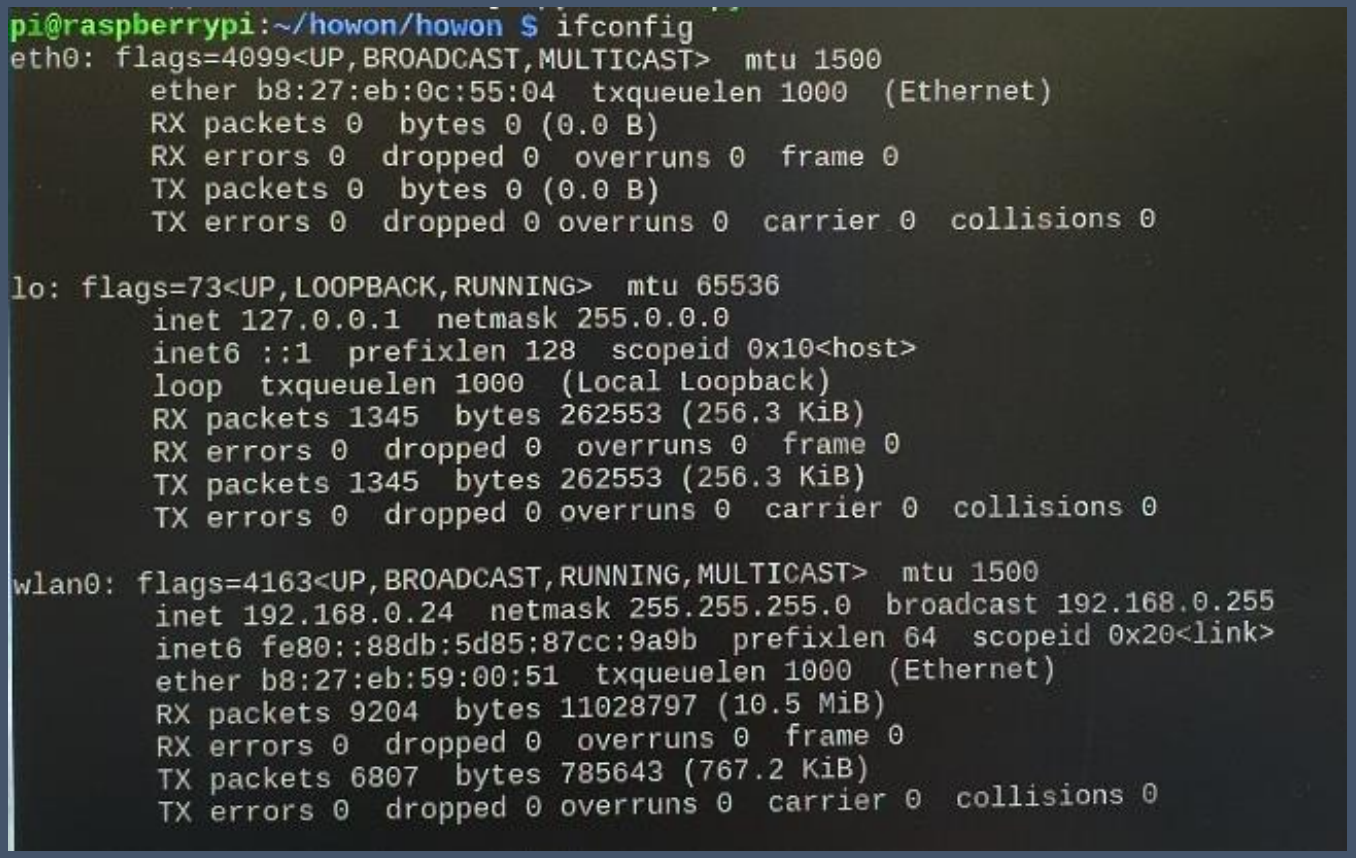
  - sudo service apache2 restart
  - sudo apt-get install python-setuptools -y
  - wget https://bootstrap.pypa.io/get-pip.py
  - sudo python get-pip.py
  - sudo rm -rf get-pip.py

장고(Django)

- **5. 장고 설치하기**
- Sudo pip3 uninstall Django
- sudo pip3 install Django

- **6. 프로젝트 시작하기**
- cd~
- cd /Desktop
- django-admin startproject howon
- cd howon
- **세팅 변경**
- cd howon/howon
- nano settings.py

- Ifconfig 명령 후 IP 확인



- settings.py에는 다음의 변경이 있습니다.
- 1) ALLOWED_HOSTS에는 자신의 IP 입력

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '&r=@4w6ly^u(@l-x=xv#-=fd=lzcg(t17okv_bt*_o+&du*-##'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['192.168.0.24']


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

- 변경: urls.py (nano urls.py)

```
…. 중략
from howon import index
urlpatterns = [
…..중략
        path('index/', index.first),
]
```

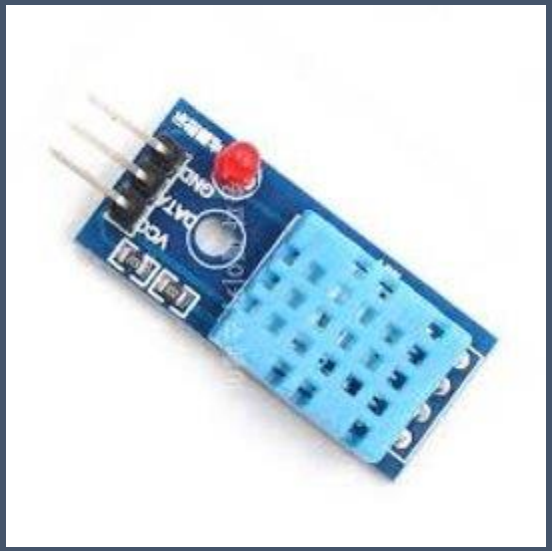
howon/howon/index.py

```
from Django.shortcuts import render
from Djnago.http import HttpResponse
def first(request):
        return HttpResponse(“Hell world”)
```

- cd Desktop/howon
- python3 manage.py migrate
- python3 manage.py runserver 192.168.0.24:8000

http://xxx.xxx.xxx.xxx:8000/index

- 온습도 센서 + 장고



- 변경: urls.py (nano urls.py)

```
…. 중략
from howon import dht11
urlpatterns = [
…..중략
        path('dht11/', dht.show_result),
]
```

- DHT11 (GND) ---- 라즈베리파이(GND)
- DHT11 (VCC) --- 라즈베리파이(5V)
- DHT11 (DAT) --- 라즈베리파이(#26)

/howon/howon/dht11.py

```
1  from django.shortcuts import render
2  from django.http import HttpResponse
3  import Adafruit_DHT
4  import datetime
5  from imp import reload
6
7  sensor = Adafruit_DHT.DHT22
8
9  #pin = 26
10 pin = 26
11 humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
12 s=datetime.datetime.now()
13
14 def show_result(request):
15     if humidity is not None and temperature is not None:
16         return HttpResponse('Temp=%0.2f*C  Humidity=%0.2f Time=%s' %(temperature, humidity,s))
17     else:
18         return HttpResponse('Failed to get reading. Try again!')
19
20
```

http://xxx.xxx.xxx.xxx:8000/dht11

- 변경: urls.py (nano urls.py)

```
…. 중략
from howon import dht11
From howon import inputbyuser
urlpatterns = [
…..중략
        path('dht11/', dht.show_result),
        path('getInput/<int:num>', inputbyuser.get_input),
]
```

/howon/howon/inputbyuser.py

```python
from django.shortcuts import render
from django.http import HttpResponse

def get_input(request, num=1):
    print("username : ", num)
    return HttpResponse("Your input is {} ".format(num))

# Create your views here.

```

Your input is 12345

http://xxx.xxx.xxx.xxx:8000/**getInput/12345**

- 장고를 이용하여 원격으로 LED를 ON/OFF 하는 코드를 작성해 보세요
  - URL을 통해 num 파라미터에 1을 전달하면 ON
  - 이외 OFF
  - time.sleep 사용
- Hint
  - try:
    - GPIO.setmode(GPIO.BCM)
    - GPIO.setup(pin, GPIO.OUT)
    - 조건문….
  - except KeyboardInterrupt:
    - Print(“end”)
  - Finally:
    - GPIO.cleanup()

QnA