
MCP3008 & TMP36

을 이용한 온도측정 및 응용

K M U C S
시스템최신기술
라즈베리사조

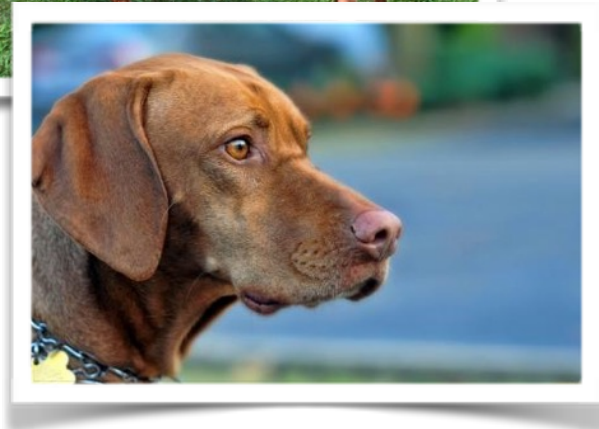
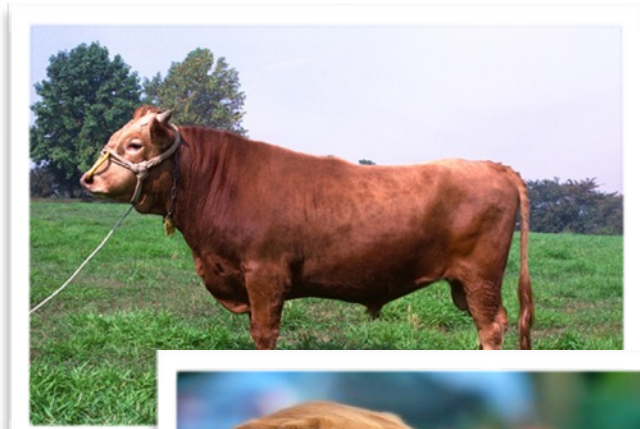
| | |
|-----|-----|
| 박성우 | 최승혁 |
| 이두나 | 허성실 |
| | 최윤승 |



1. MCP3008은 무엇이고, TMP36은 무엇인가?
 - 1.1. 소개
 - 1.2. 회로배선
2. 이걸 어떻게 써먹는가?
 - 2.1. 시스템 아키텍처 & 프로그램 설명
 - 2.2. 온도측정/화재경보 소스코드 설명 (까지만)
3. 데모 & QnA

1. MCP3008은 무엇이고, TMP36은 무엇인가?

1.1.



1. MCP3008은 무엇이고, TMP36은 무엇인가?

1.1.



- 우리가 느끼는 **온도**는 아날로그
- 그러나 컴퓨터는 **디지털**
- 애들끼리 뭔가 소통가능한
일종의 **컨버터**가 필요해!

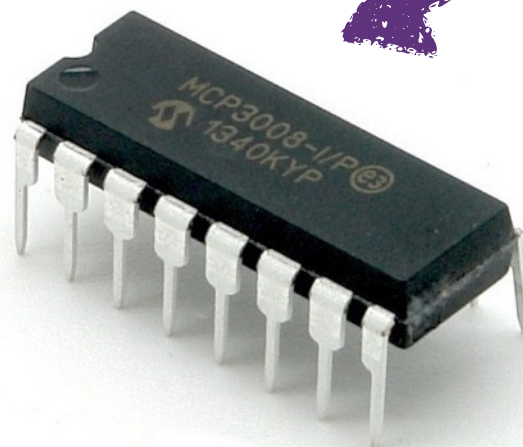
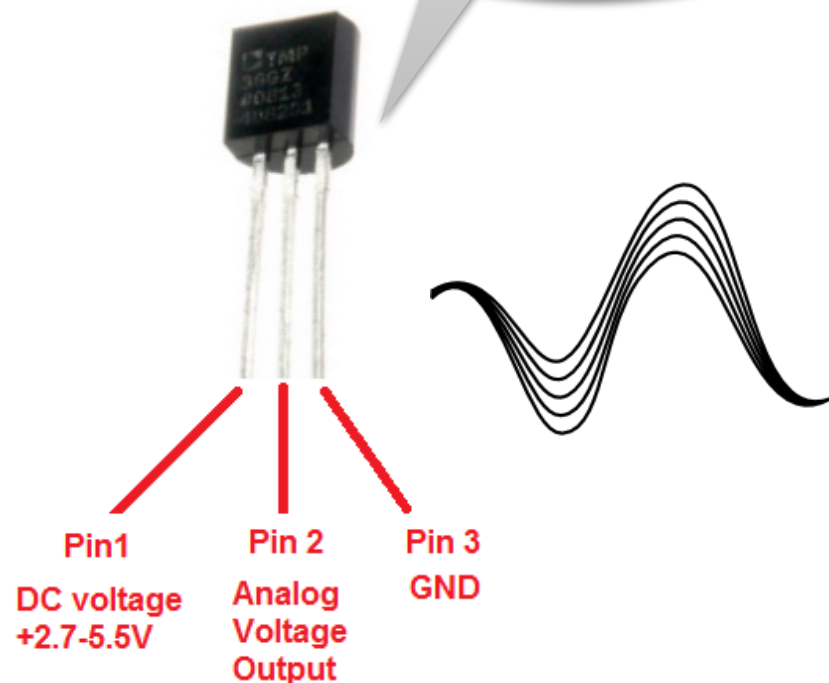
1. MCP3008은 무엇이고, TMP36은 무엇인가?

1.1.

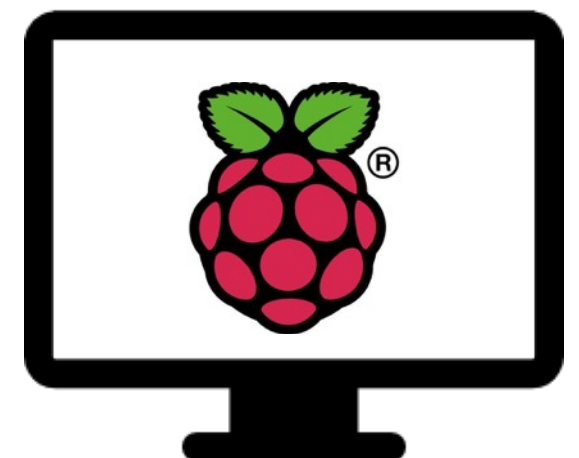


얘가 바로
'TMP36'
온도정보를 아날로그
신호로 출력한다.

- 우리가 느끼는 **온도**는 아날로그
- 그러나 컴퓨터는 **디지털**
- 애들끼리 뭔가 소통가능한
일종의 **컨버터**가 필요해!



0 1 0 0
0 0 0 0
1 0 1 1

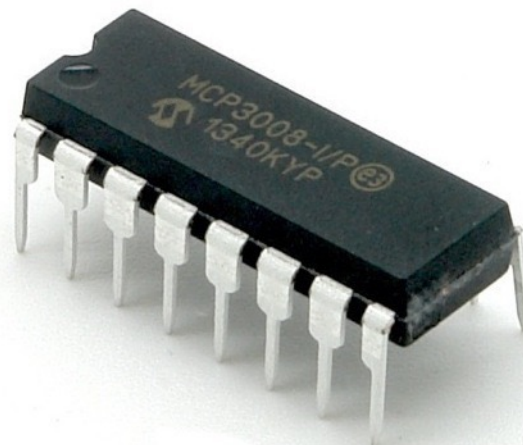
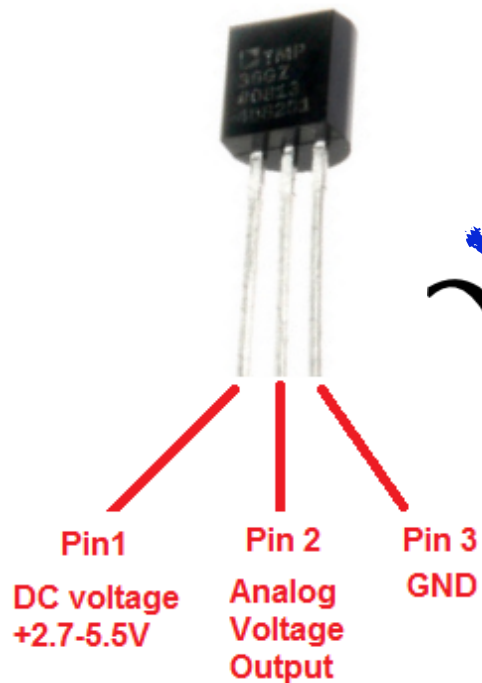
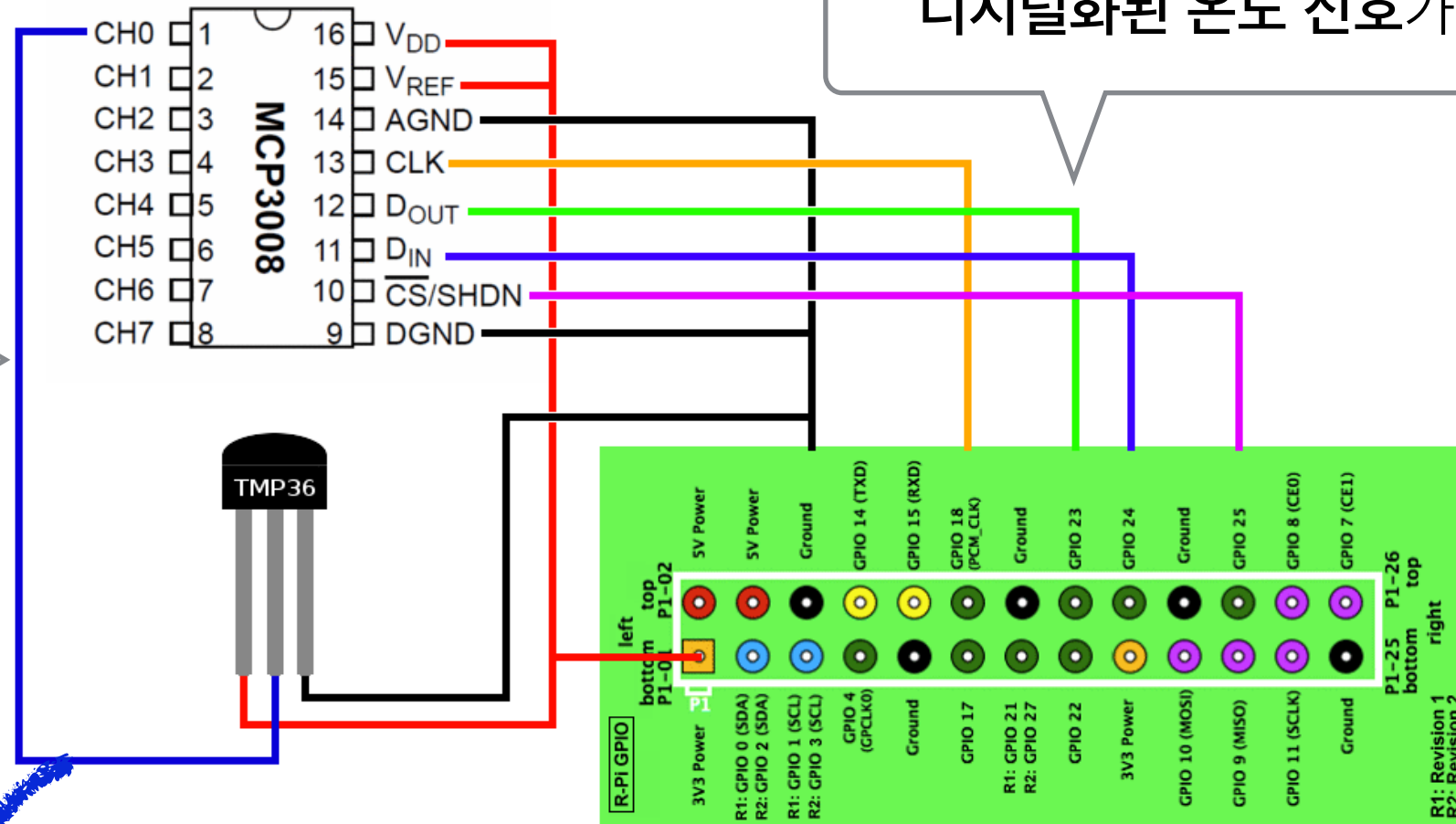


1. MCP3008은 무엇이고, TMP36은 무엇인가?

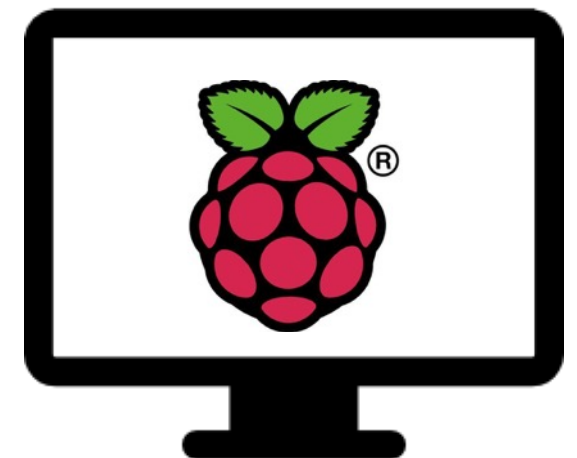
1.2. 회로배선

MCP3008의
channel 0번으로,
TMP36이 보내는
온도 신호가
전송된다.

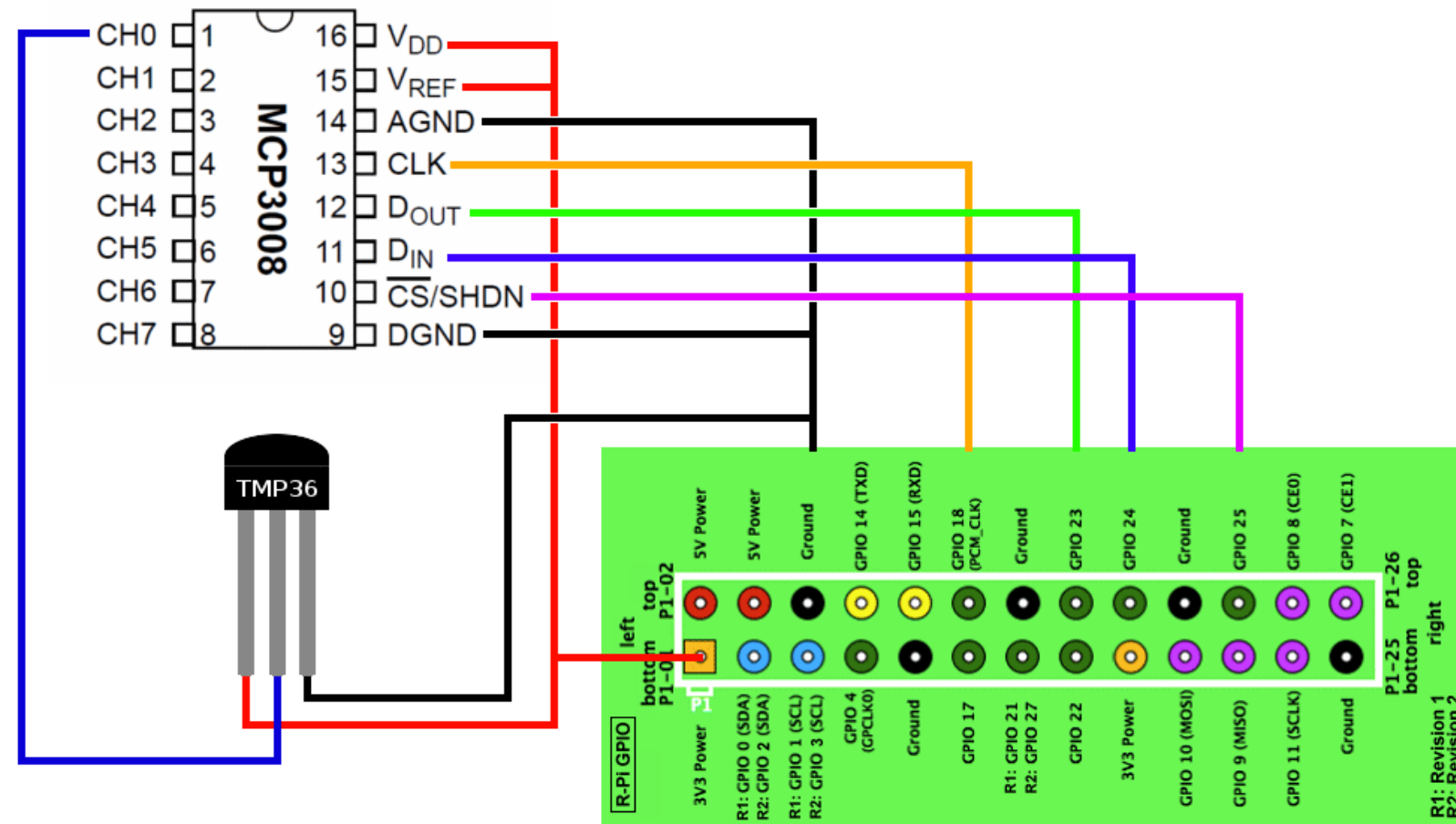
General-purpose input/output을 통해
디지털화된 온도 신호가 전송된다.



0 1 0 0
0 0 0 0
1 0 1 1



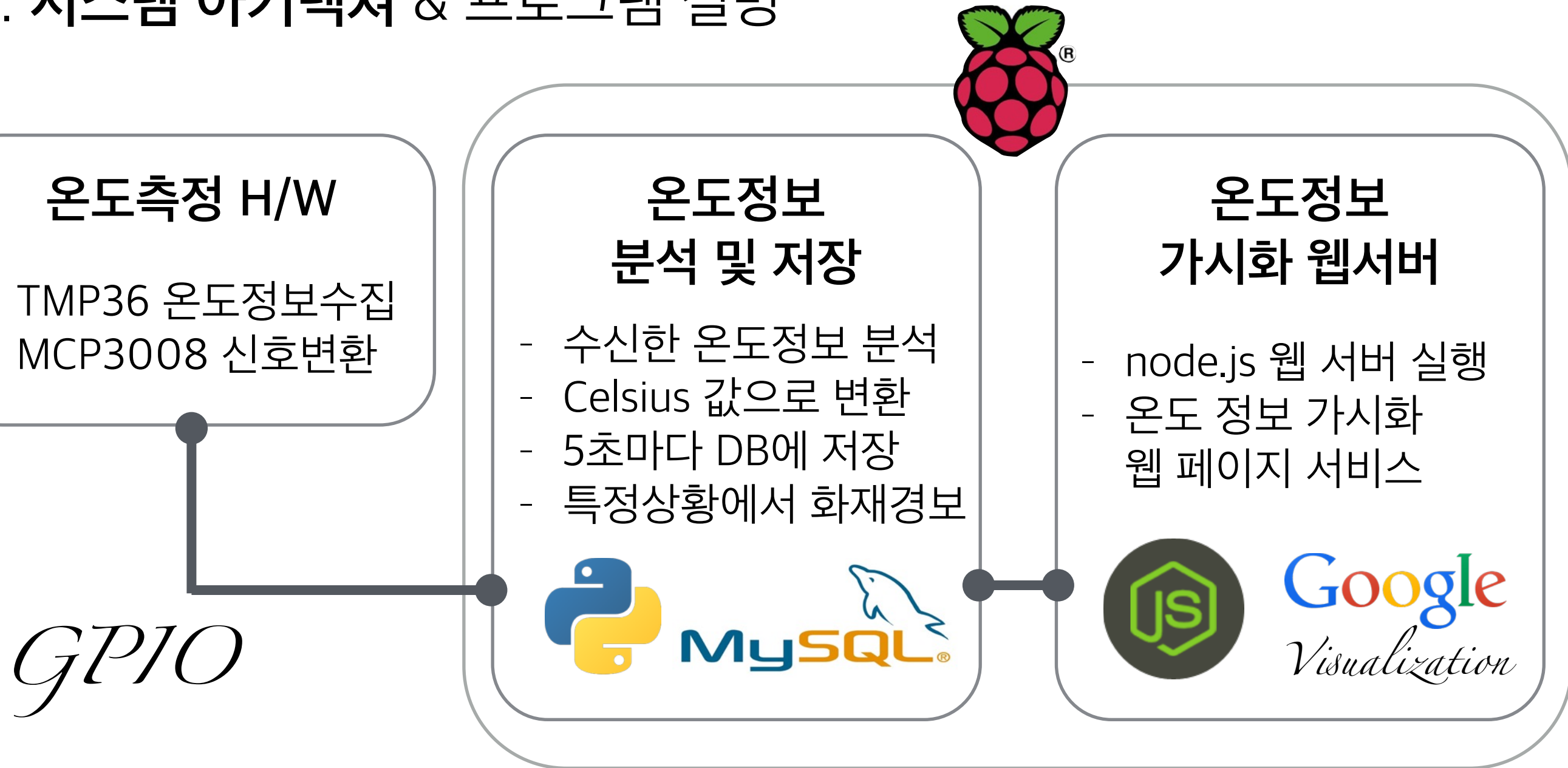
1. MCP3008은 무엇이고, TMP36은 무엇인가?



이제 온도 측정은 어떻게든 하겠는데.....
이걸 어디다가 어떻게 써먹나요?

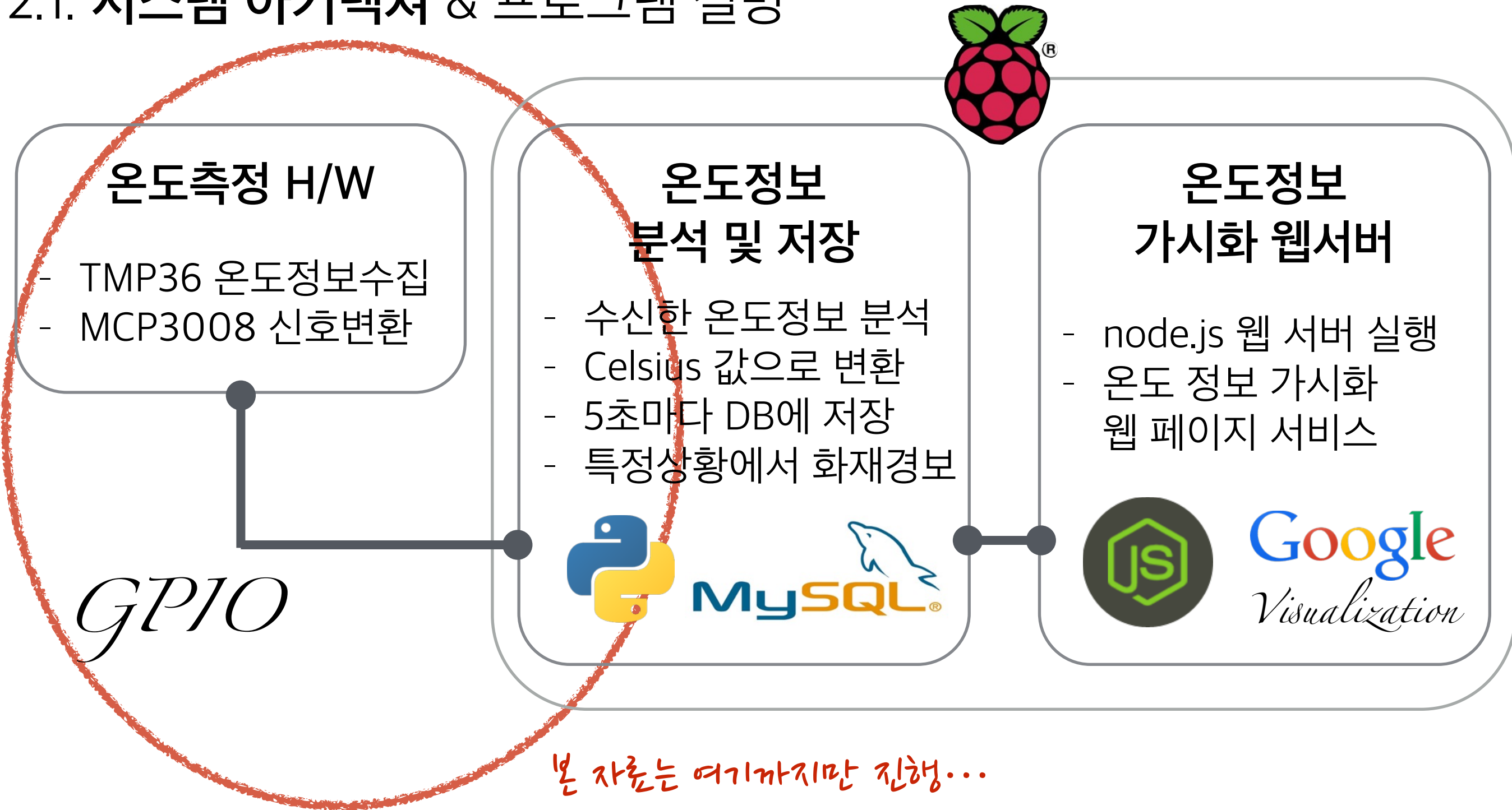
2. 이걸 어떻게 써먹는가?

2.1. 시스템 아키텍처 & 프로그램 설명



2. 이걸 어떻게 써먹는가?

2.1. 시스템 아키텍처 & 프로그램 설명



2. 이걸 어떻게 써먹는가?

2.1. 시스템 아키텍처 & 프로그램 설명



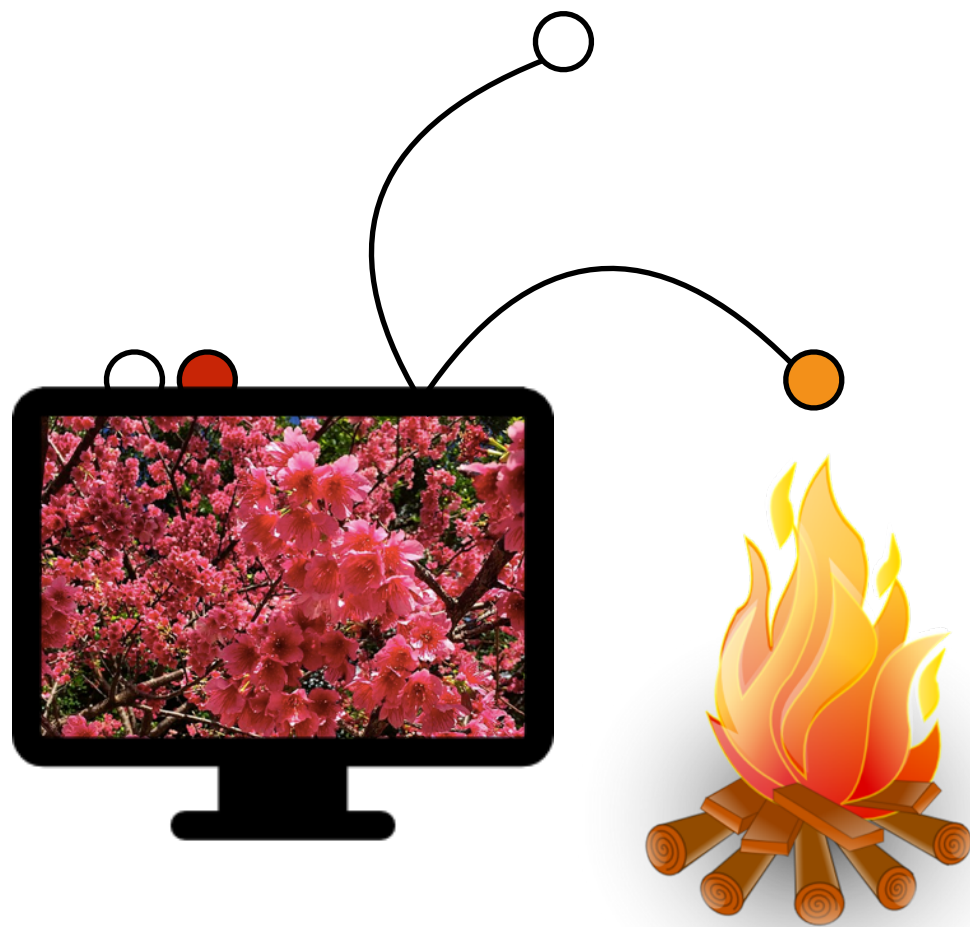
2. 이걸 어떻게 써먹는가?

2.1. 시스템 아키텍처 & 프로그램 설명



2. 이걸 어떻게 써먹는가?

2.1. 시스템 아키텍처 & 프로그램 설명



그리고 ‘봄노래’가
재생된다.

2. 이걸 어떻게 써먹는가?

2.2. 온도측정/화재경보 소스코드 설명

그 전에, MCP3008을 사용하기 위해, 터미널에서 다음 명령어를 실행하세요.

```
apt-get install python-dev
```

최신의 라즈비안은 기본적으로 GPIO를 사용하는 라이브러리를 포함하고 있지만, 가끔 없는 경우도 있습니다. 다음 명령어도 그냥 따지지 말고 실행하세요.

```
sudo easy_install -U distribute  
sudo apt-get install python-pip  
sudo pip install rpi.gpio
```

```
#!/usr/bin/env python
```

```
import time  
import RPi.GPIO as GPIO
```

```
GPIO.setwarnings(False)  
GPIO.setmode(GPIO.BCM)
```

```
# initialise les ports du GPIO  
GPIO.setup(24, GPIO.OUT)  
GPIO.setup(23, GPIO.IN)  
GPIO.setup(18, GPIO.OUT)  
GPIO.setup(25, GPIO.OUT)
```

```
# lit les donnees SPI du MCP3008 et renvoie un entier entre 0 et 1023  
def readspi(mcp3008_input):
```

```
    GPIO.output(25, True)  
    GPIO.output(18, False)  
    GPIO.output(25, False)  
    commandout = mcp3008_input  
    commandout |= 0x18  
    commandout <= 3
```

```
    for i in range(5):  
        if (commandout & 0x80):  
            GPIO.output(24, True)  
        else:  
            GPIO.output(24, False)  
        commandout <= 1
```

사용할 GPIO
핀 세팅


```
commandout <=<= 1
GPIO.output(18, True)
GPIO.output(18, False)
bitout = 0
for i in range(12):
    GPIO.output(18, True)
    GPIO.output(18, False)
    bitout <=<= 1
    if (GPIO.input(23)):
        bitout |= 0x1
GPIO.output(25, True)
bitout /= 2
return bitout
```

```
# 150315 # for insert temperature data to MySQL database
```

```
import MySQLdb as db
import datetime
tmpDB = db.connect('<<ip_address>>', '<<username>>', '<<password>>',
'<<schema_name>>')
```

```
cur = tmpDB.cursor()
```

```
# 150328 # for LED & sound playing
import pygame #doona
```

```
GPIO.setup(17, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)
```

사용할 GPIO
핀 세팅

Embedded
SQL을 위한
세팅

```
# 150328 # for LED & sound playing
```

```
import pygame #doona
```

```
GPIO.setup(17, GPIO.OUT)
```

```
GPIO.setup(22, GPIO.OUT)
```

```
GPIO.output(17, True)
```

```
GPIO.output(22, True)
```

```
time.sleep(1)
```

```
GPIO.output(17, False)
```

```
GPIO.output(22, False)
```

```
pygame.mixer.init()
```

```
background = pygame.mixer.Sound("testsound.wav")
```

```
flag = 0
```

```
## import Settings end. ##
```

```
# boucle principale qui realise les calculs et affiche le resultat
```

```
toutes les 5 secondes
```

```
while True:
```

```
    try:
```

```
        bitout = readspi(0) # lit les donnees de l'entree 0 du MCP3008
```

```
        mV = bitout * ( 3300.0 / 1024.0) # convertit la valeur en mV
```

```
        tC = (mV/10.0) - 50.0 # convertit les mV en degre Celsius
```

```
        tC = "%.1f" % tC # ne garde que une decimale
```

화재경보
데모를 위한
LED 설정

화재경보
데모를 위한
음악재생
설정

```

while True:
    try:
        bitout = readspi(0) # lit les donnees de l'entree 0 du MCP3008
        mV = bitout * ( 3300.0 / 1024.0) # convertit la valeur en mV
        tC = (mV/10.0) - 50.0 # convertit les mV en degre Celsius
        tC = "%.1f" % tC # ne garde que une decimale

        # 150318 # second TMP38
        mV2 = bitout2 * ( 3300.0 / 1024.0) # convertit la valeur en mV
        tC2 = (mV2/10.0) - 50.0 # convertit les mV en degre Celsius
        tC2 = "%.1f" % tC2 # ne garde que une decimale

        # for alert
        if(abs(float(tC)-float(tC2)) <= 3.0):
            GPIO.output(17, True)
            GPIO.output(22, False)
            print("green")

        else:
            GPIO.output(22, True)
            GPIO.output(17, False)
            print("red")
            if (flag == 0) :
                flag = 1
                background.play()

    print(tC, tC2)

    # 150315 # insert to database

```

TMP36 2개
설정, Celsius
단위로 변환

화재경보를
위한 조건분기

```
GPIO.output(22, False)
print("green")

else:
    GPIO.output(22, True)
    GPIO.output(17, False)
    print("red")
    if (flag == 0) :
        flag = 1
        background.play()

print(tC, tC2)

# 150315 # insert to database
current_time = datetime.datetime.now()
#print current_time, tC

query = "INSERT INTO Gathering(time, celsius)" \
        "VALUES(%s,%s)"
data = [(current_time, tC)]
cur.executemany(query,data)
tmpDB.commit()

## Embedded SQL statements end. ##

time.sleep(5)
except KeyboardInterrupt:
    quit()
```

온도값을
Database에
전송

3. 데모 & QnA



1. 실온 상태에서는 녹색불이 들어온다.
2. 화재경보(센서의 체온감지) 상황에서 적색불 점등과 동시에 봄노래 재생

감사합니다 !