



Raspberry - HW2

임베디드스쿨1기

Lv1과정

2020. 11.03

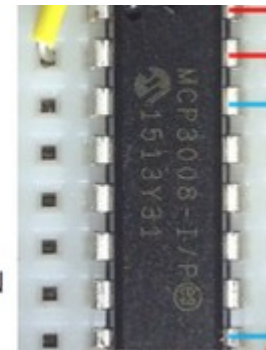
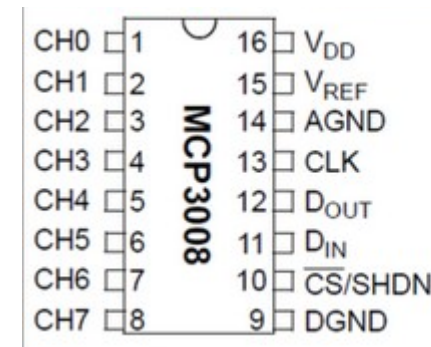
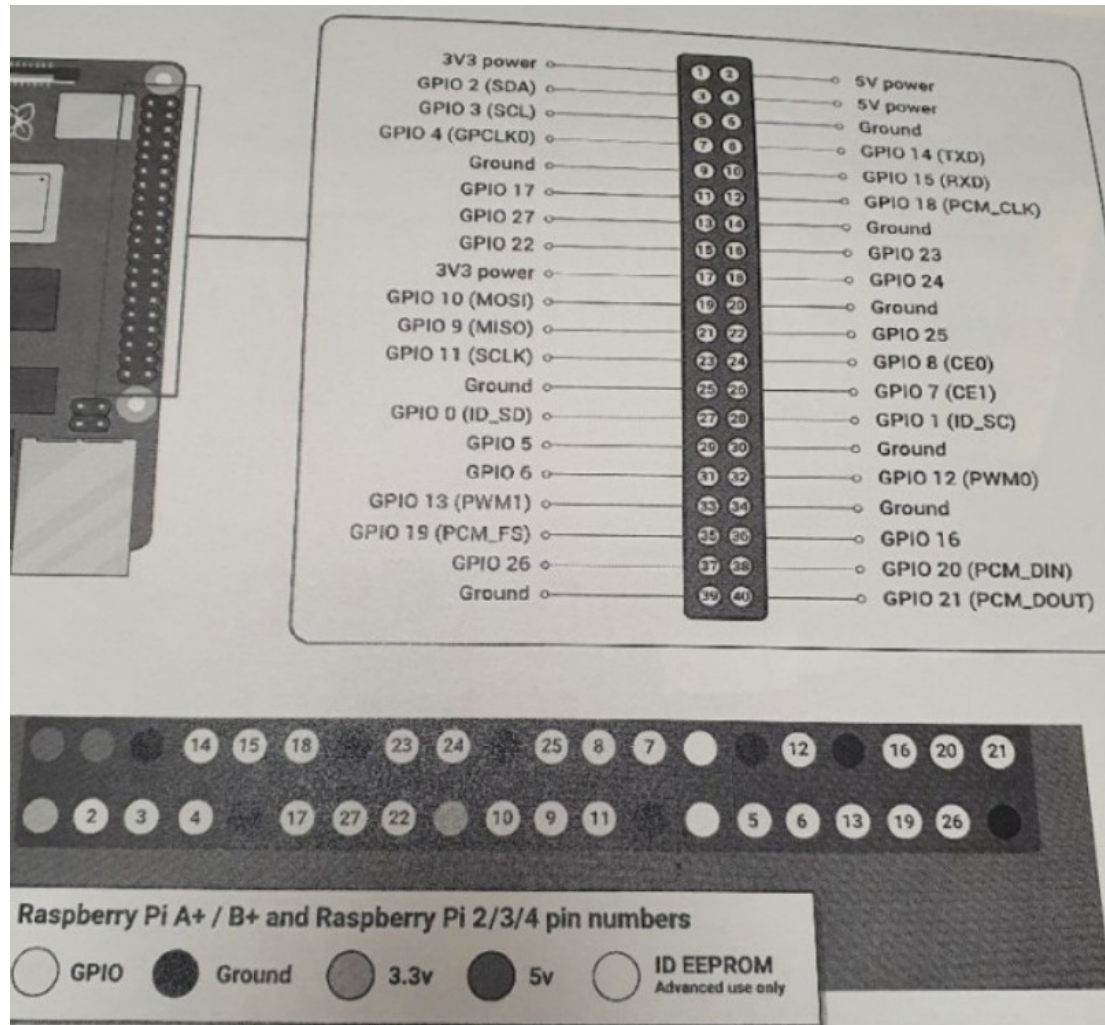
박하늘

1. LDR Sensor Control

1) Raspberry Pin & MCP3008

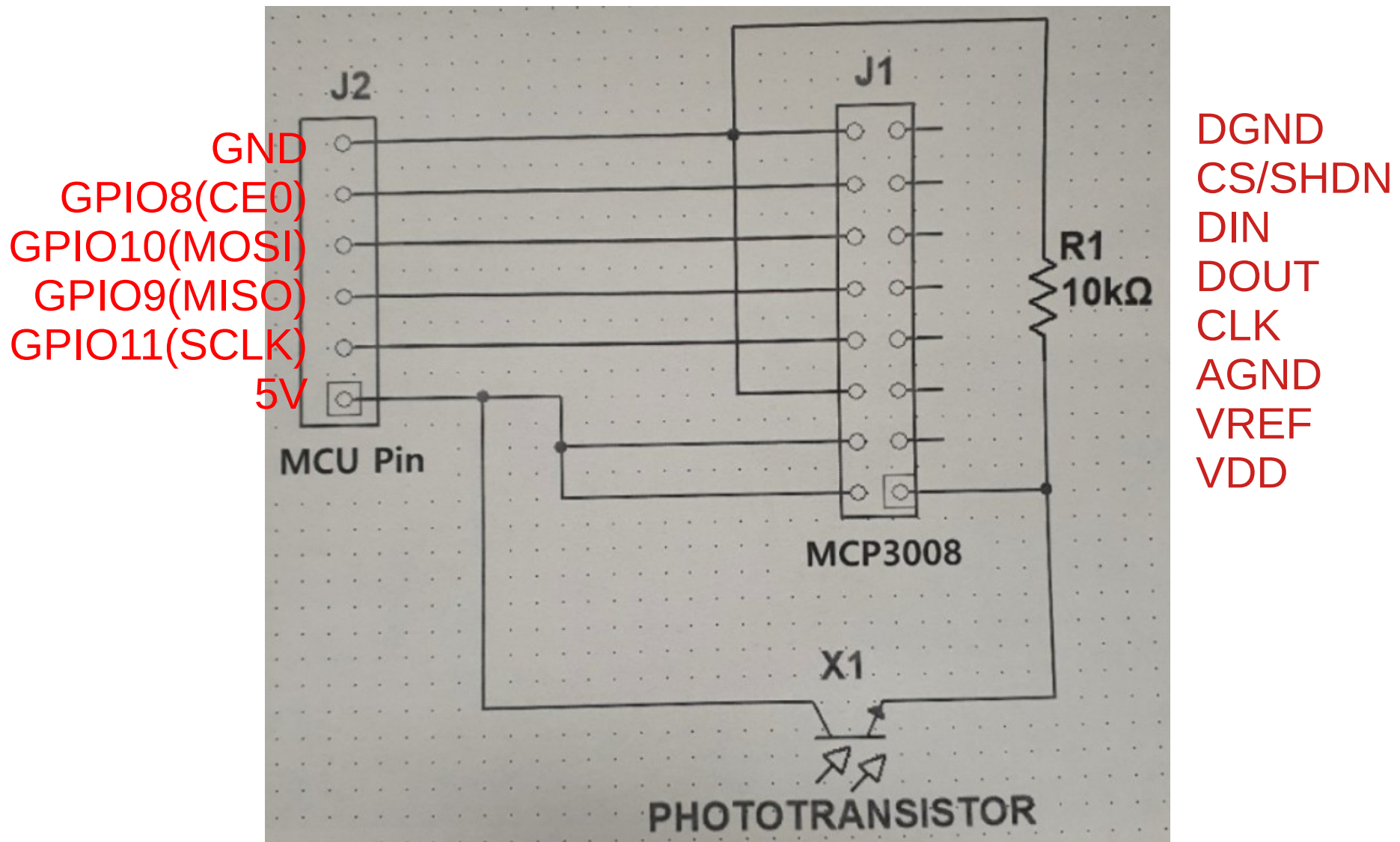
: SPI는 마스터(라즈베리파이)와 슬레이브(MCP3008)로 나뉨

마스터가 슬레이브에 정해진 명령을 보내면 그에 해당 되는 값을 전송해주는 시스템



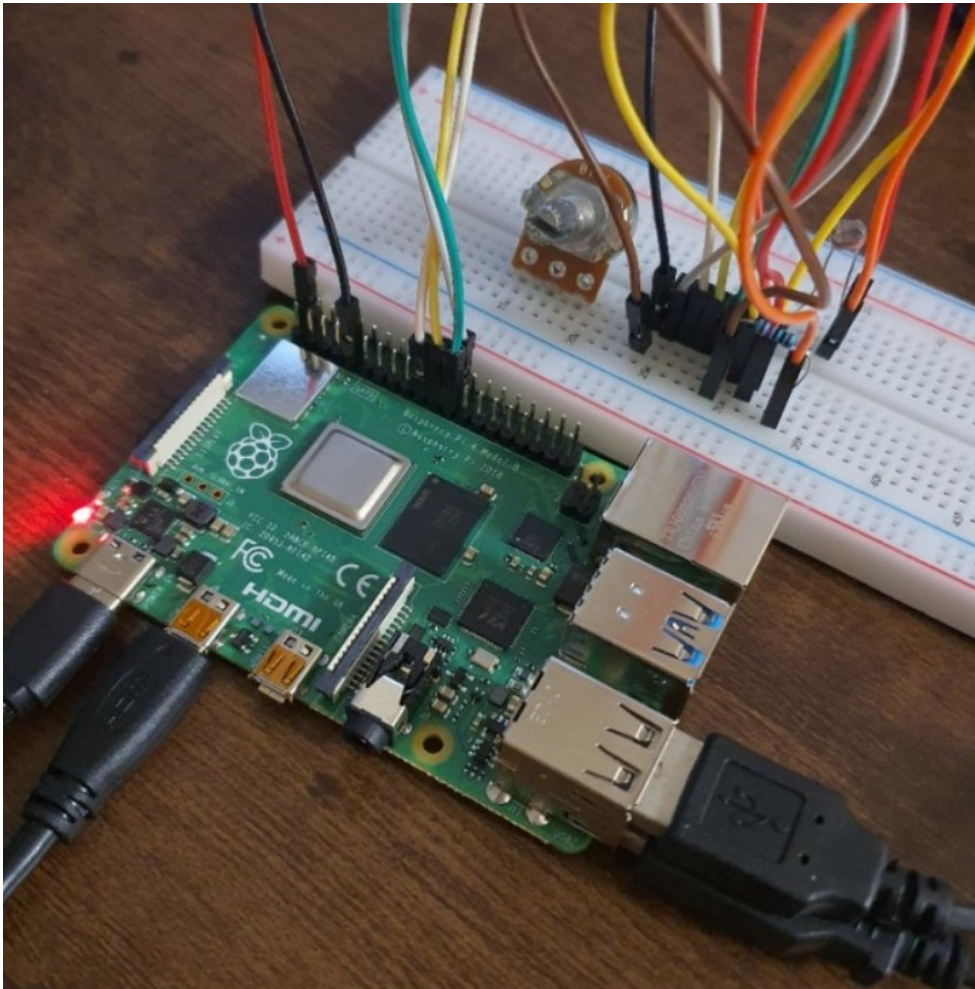
1. LDR Sensor Control

2) LDR Sensor 회로 구성



1. LDR Sensor Control

3) 회로 구성 & 결과



```
LDR Value : 1021
-----
LDR Value : 1022
-----
LDR Value : 1007
-----
LDR Value : 816
-----
LDR Value : 957
-----
LDR Value : 794
-----
LDR Value : 923
-----
LDR Value : 668
-----
LDR Value : 906
-----
LDR Value : 899
-----
LDR Value : 888
-----
LDR Value : 879
-----
LDR Value : 875
-----
LDR Value : 850
-----
```

1. LDR Sensor Control

1) 소스분석

```
1 import spidev
2 import time
3
4 delay = 0.5
5 ldr_channel = 0
6
7 #인스턴스 생성
8 spi = spidev.SpiDev()
9 #spi 통신 시작
10 spi.open(0,0)
11 #spi 통신 속도 설정
12 spi.max_speed_hz = 10000
13
14 #0~7까지 8개의 채널에서 spi 데이터를 읽는다.
15
16 def readadc(adcnum):
17     if adcnum > 7 or adcnum < 0:
18         return -1
19
20     r = spi.xfer2([1,8+adcnum << 4, 0])
21     data = ((r[1] & 3) << 8) + r[2]
22     return data
23
24 while True:
25     ldr_value = readadc(ldr_channel)
26     print("-----")
27     print("LDR Value : %d" % ldr_value)
28     time.sleep(delay)
```

- 센서 값을 ADC하여 라즈베리파이가 수신할 필요가 있으므로 다음과 같은 규칙으로 전송해야 한다.

- spi.xfer2라는 함수는 안에 있는 데이터를 쓰고 리턴된 값을 반환 (각각 시작비트, 몇번째 데이터를 불러올 것인지, 그리고 자릿수를 맞추기 위한 값)

라즈베리파이는 총 3개의 바이트를 전송해주는데,

첫째 바이트 0 0 0 0 0 Start SGL D2

둘째 바이트 D1 D0 X X X X X X

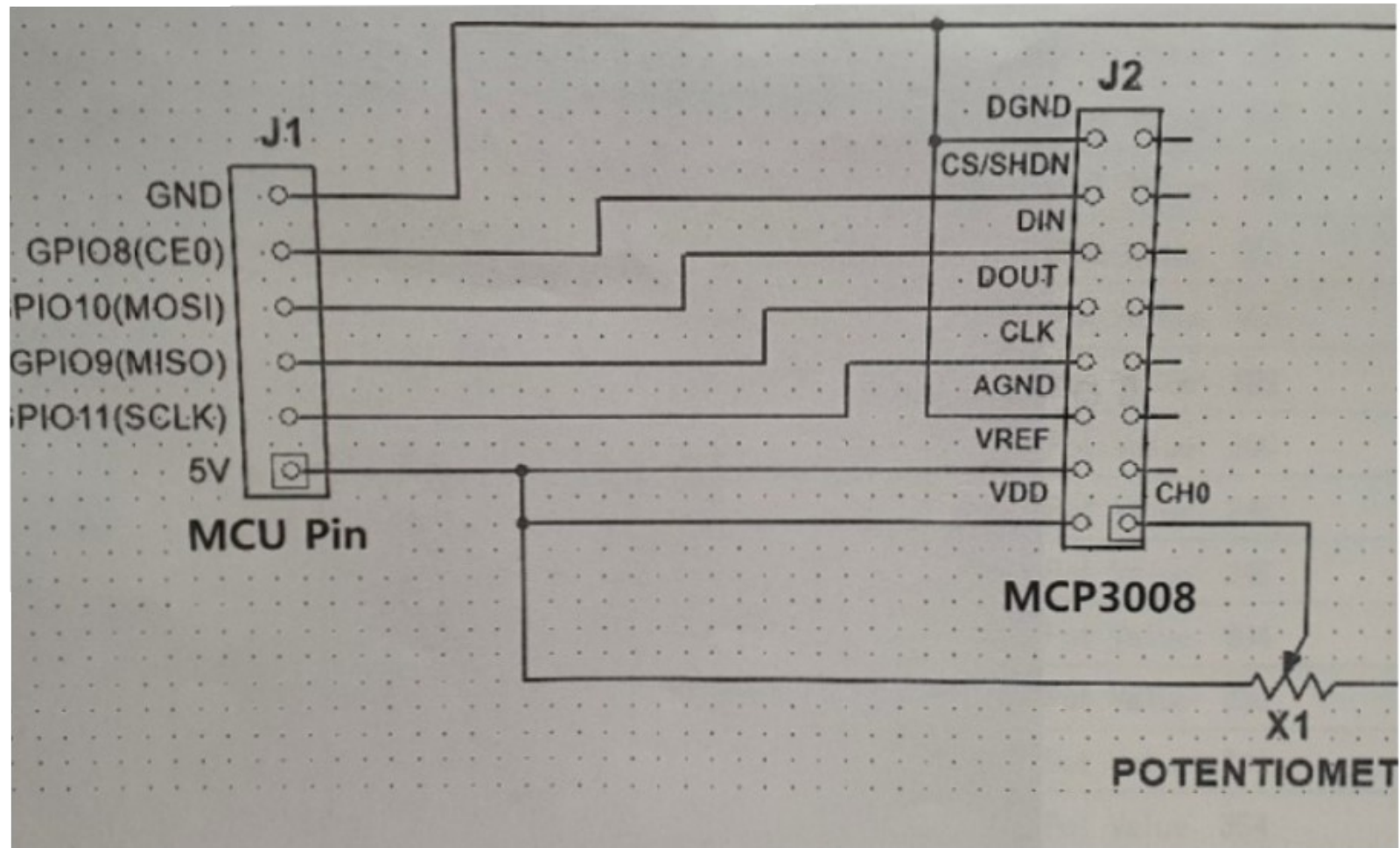
셋째 바이트 X X X X X X X X

데이터 중 아래에서 12비트가 ADC결과값이다.

반환되는 r은 첫 8비트는 비어있으며
두번째 8비트중 1, 2번째 비트는 ADC의 9, 10번째 비트,
세번째는 하위 8비트를 의미합니다.
이를 더함으로써 총 10비트의 데이터가 완성

2. SPI Control

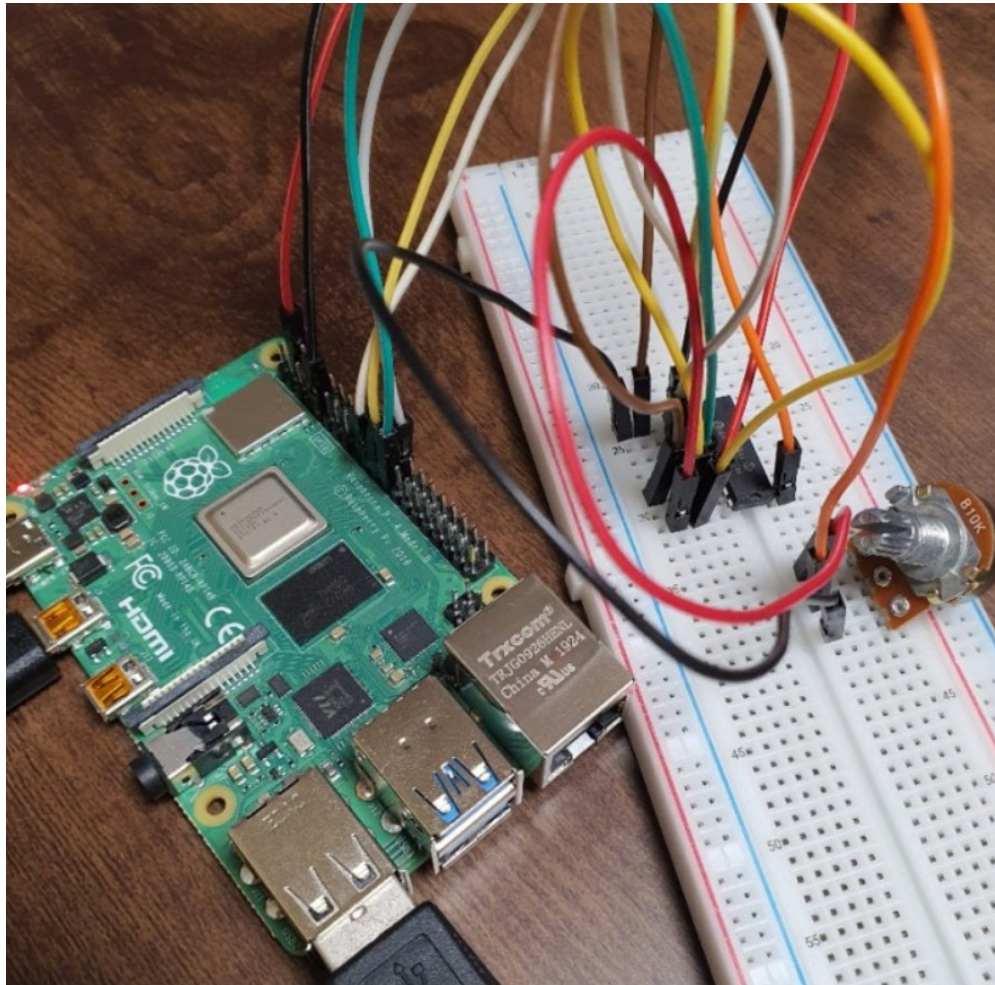
1) SPI Control 회로 구성



2. SPI Control

1) 회로 구성 & 소스 & 결과

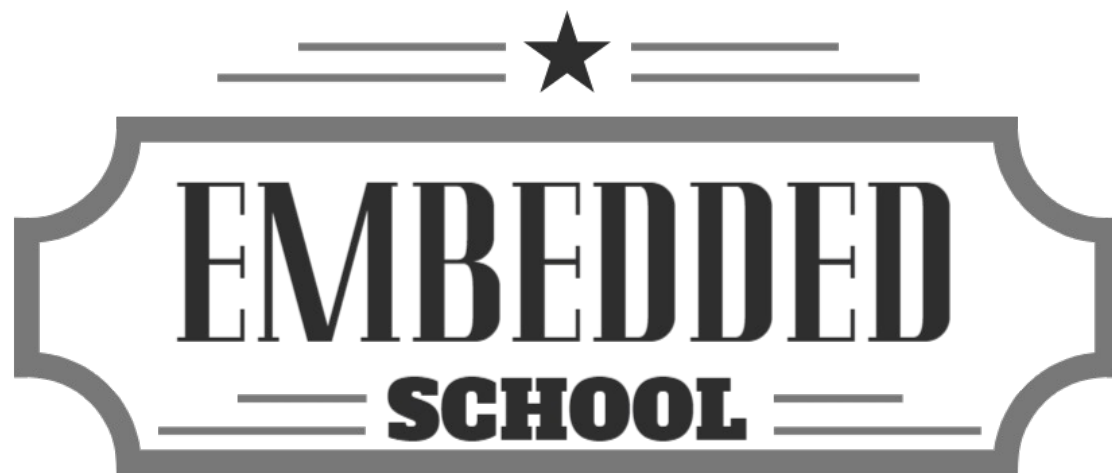
(소스설명 pg.4참고)



MCP3003 SPI Control.py

```
1 import spidev
2 import time
3
4 delay = 0.5
5 pot_channel = 0
6
7 #인스턴스 생성
8 spi = spidev.SpiDev()
9 #spi 통신 시작
10 spi.open(0,0)
11 #spi 통신 속도 설정
12 spi.max_speed_hz = 10000
13
14 #0-7까지 8개의 채널에서 spi 데이터를 읽는다.
15
16 def readadc(adcnun):
17     if adcnun > 7 or adcnun < 0:
18         return -1
19
20     r = spi.xfer2([1,8+adcnun << 4, 0])
21     data = ((r[1] & 3) << 8) + r[2]
22     return data
23
24 while True:
25     pot_value = readadc(pot_channel)
26     print("-----")
27     print("LDR Value : %d" % pot_value)
28     time.sleep(delay)
29
```

```
-----
LDR Value : 259
-----
LDR Value : 145
-----
LDR Value : 16
-----
LDR Value : 187
-----
LDR Value : 421
-----
LDR Value : 132
-----
LDR Value : 28
```



감사합니다.