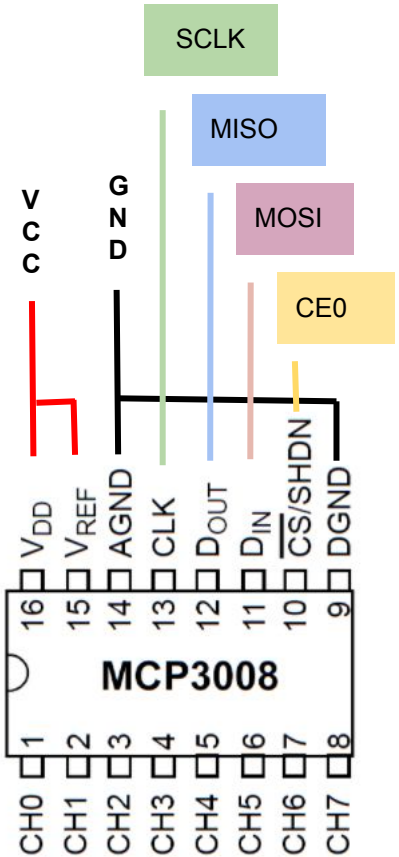
# MCP3008

SPI AC-to-DC轉換器

## 硬體連接




**不使用的CHANNEL接地**

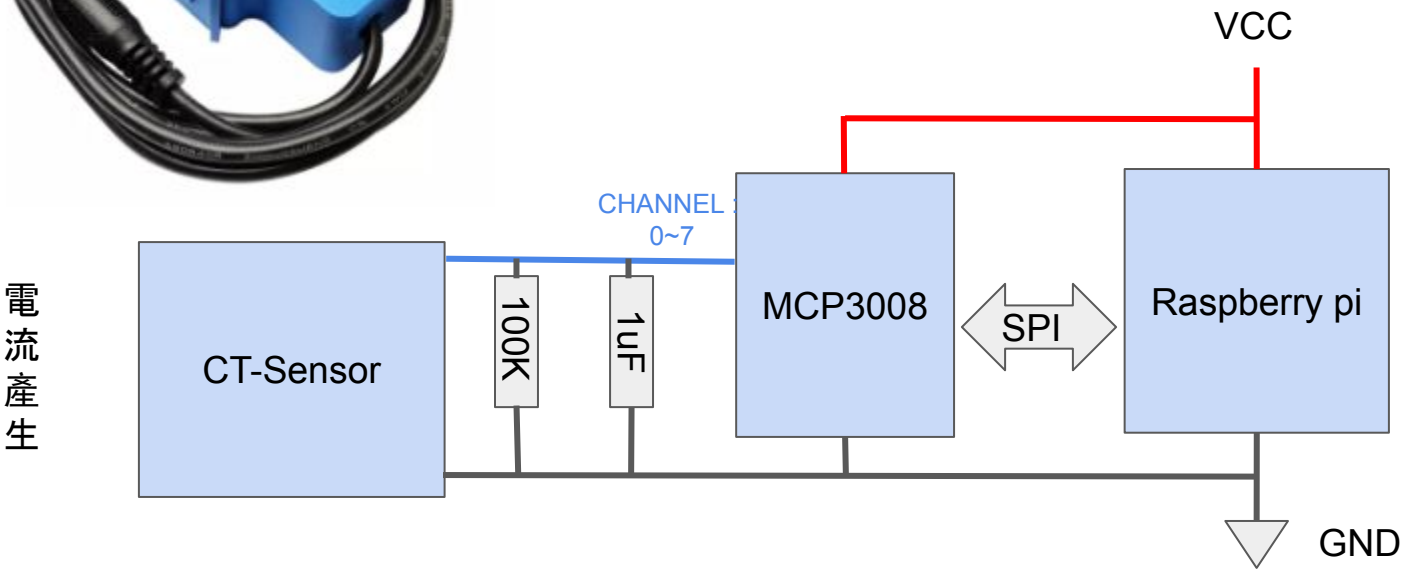
## 範例程式碼 參考網址

```
import time
import datetime

import Adafruit_GPIO.SPI as SPI
import Adafruit_MCP3008

# Software SPI configuration:
CLK  = 11
MISO = 9
MOSI = 10
CS   = 8
mcp = Adafruit_MCP3008.MCP3008(clk=CLK, cs=CS, miso=MISO, mosi=MOSI)
print(datetime.datetime.now())
print('Reading MCP3008 values, press Ctrl-C to quit...')
print('| {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(*range(8)))
print('-' * 57)

# Main program loop.
while True:
    # Read all the ADC channel values in a list.
    values = [0]*8
    for i in range(8):
        # The read_adc function will get the value of the specified channel (0-7).
        values[i] = mcp.read_adc(i)
    # Print the ADC values.
    print('| {0:>4} | {1:>4} | {2:>4} | {3:>4} | {4:>4} | {5:>4} | {6:>4} | {7:>4} |'.format(*values))
    # Pause for half a second.
    time.sleep(0.5)
```

CT-Sensor (CH1) - 透過RLC並聯電路

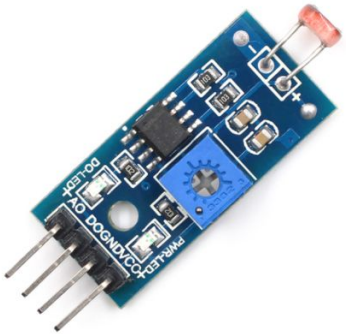
```
def CTsensor():
    global MCP3008_Ch
    Number=1
    Sensor=["CT-sensor","10","30"] #NAME , <Value=0 , >Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values));#print(AV)
    if (AV<int(Sensor[1]) and AV==1):
        MCP3008_Ch[Number] = 0 ;   #print("OFF")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 1 ;   #print("ON")
```

實際感測器皆有誤差需做調整



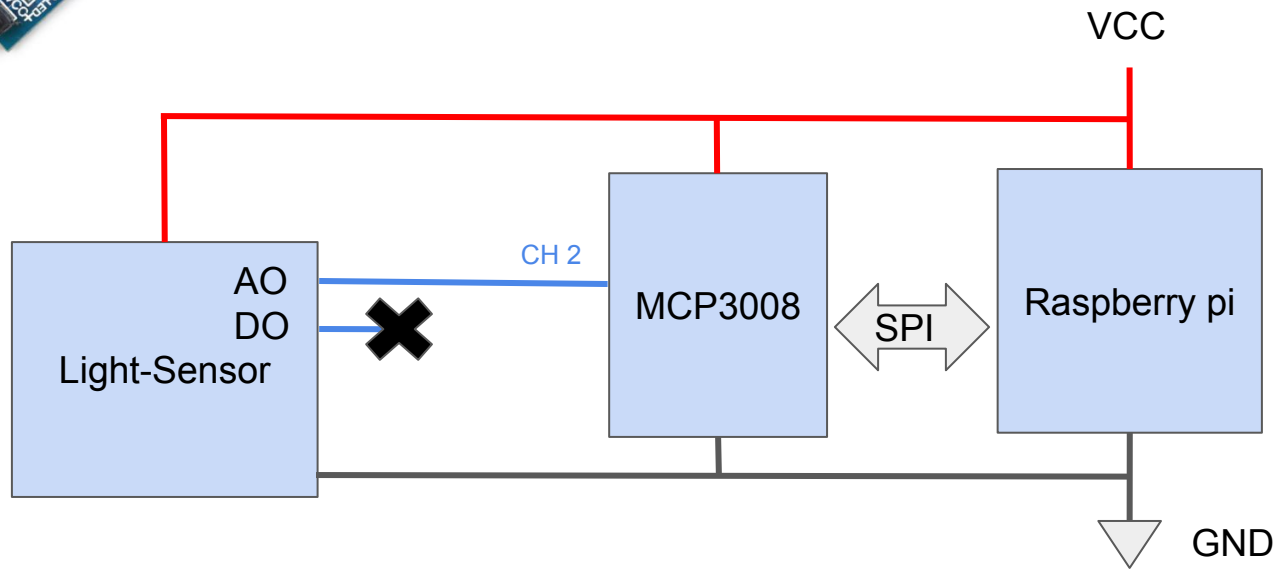

電流產生

Light-Sensor (CH2)

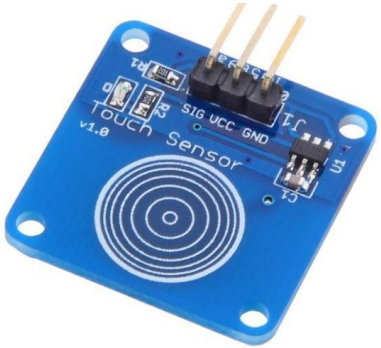


```
def Lightsensor():
    global MCP3008_Ch
    Number=2
    Sensor=["Light-sensor","200","200"] #NAME , <Value=1 , >Value=0
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values));#print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 1 ;   #print("Bright")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 0 ;   #print("DARK")
```

實際感測器皆有誤差需做調整

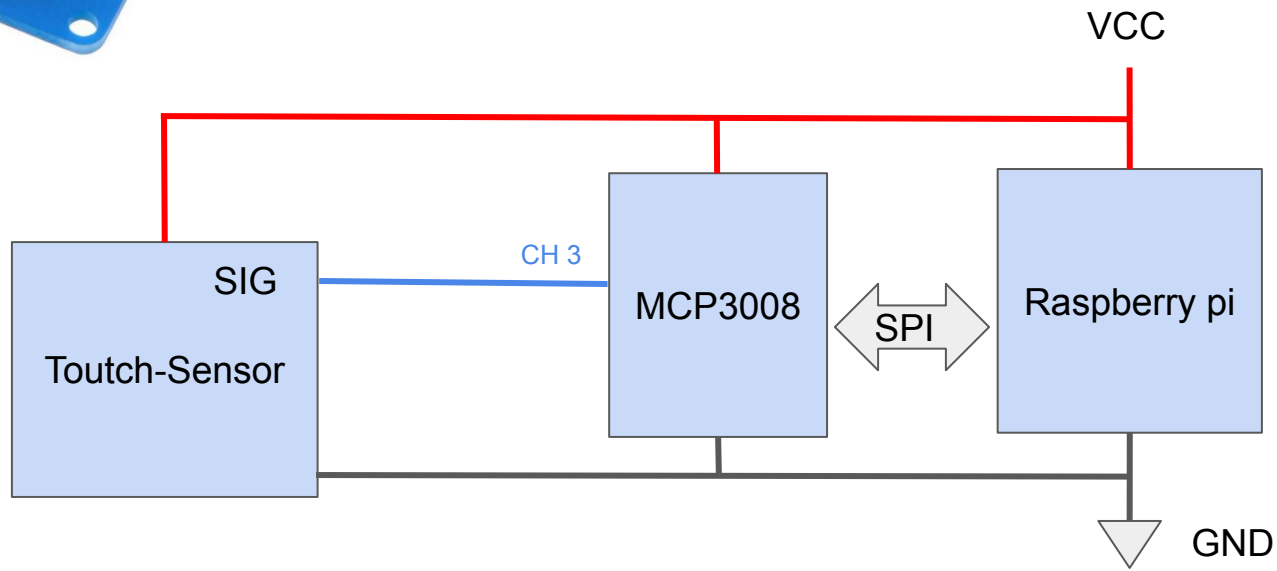


Light-Sensor (CH3)

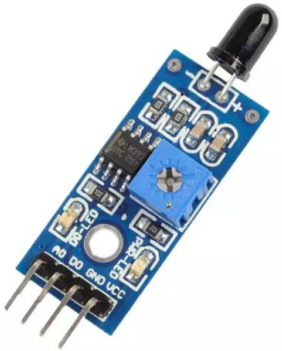



```
def Toutchsensor():
    global MCP3008_Ch
    Number=3
    Sensor=["Toutch-sensor","500","600"] #NAME , <Value=0 , >Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values));#print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 0 ;   #print("NO CONTACT")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 1 ;   #print("CONTACT")
```

實際感測器皆有誤差需做調整

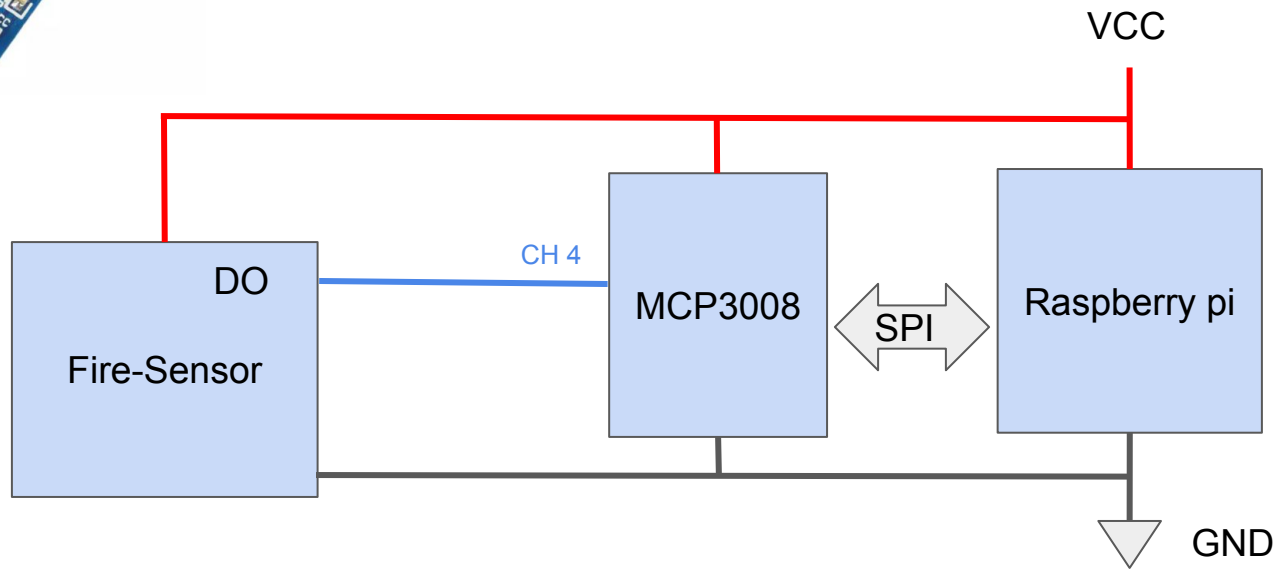


Fire-Sensor (CH4)




```
def Firesensor():
    global MCP3008_Ch
    Number=4
    Sensor=["Fire-sensor","100","1000"] #NAME , <Value=0 , >Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values));#print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 1 ;   #print("Fire")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 0 ;   #print("no Fire")
```

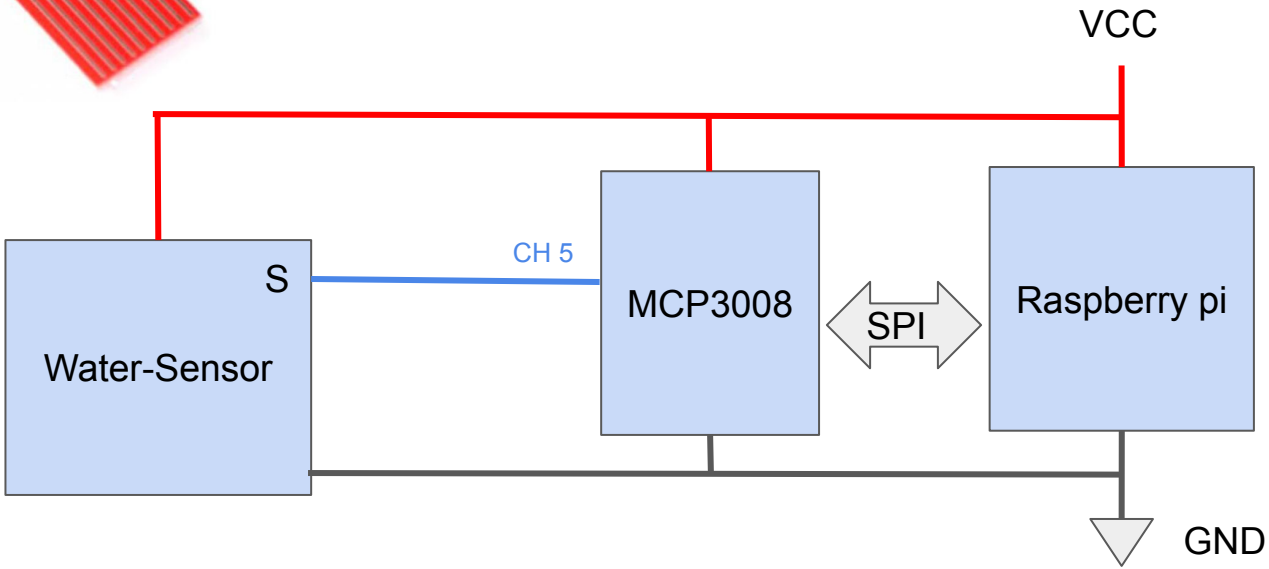
實際感測器皆有誤差需做調整



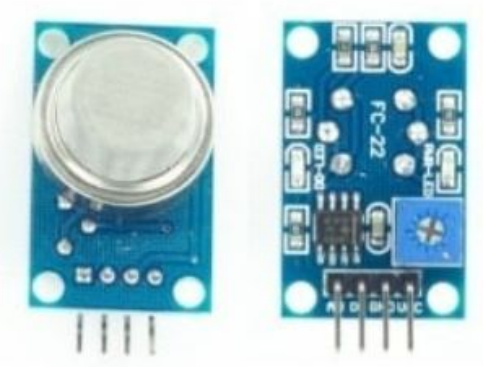
# Water-Sensor (CH5)


```
def Watersensor():
    global MCP3008_Ch
    Number=5
    Sensor=["Water-sensor","10","100"] #NAME , <Value=0 , >Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values));#print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 0 ;   #print("leaking")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 1 ;   #print("no leaking")
```

實際感測器皆有誤差需做調整

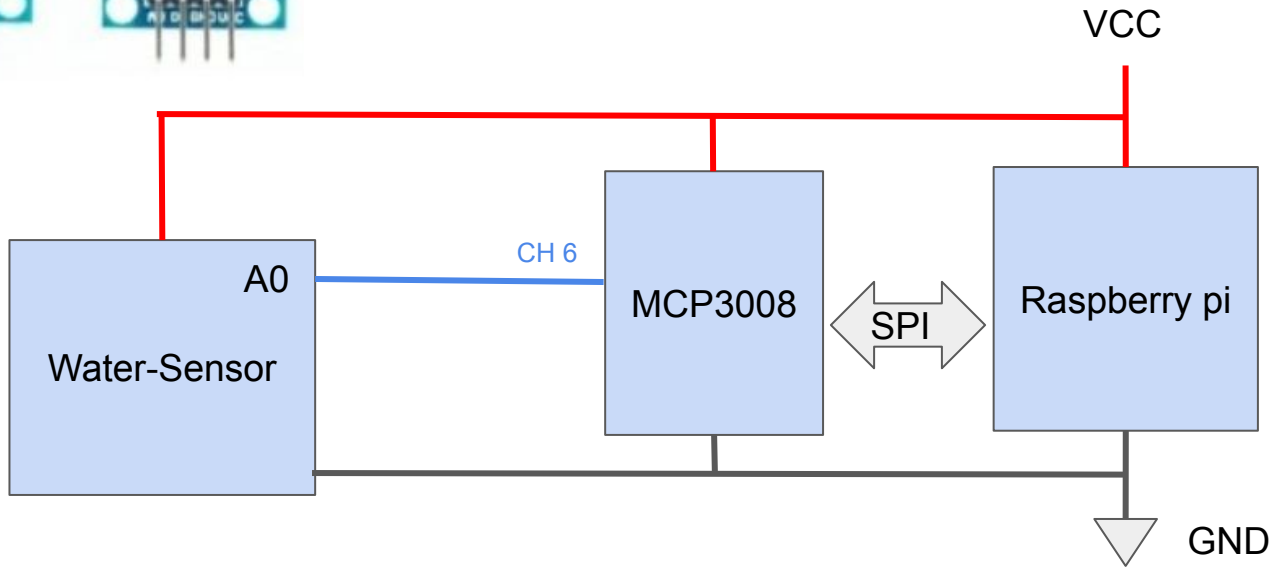


GAS-Sensor (CH6)

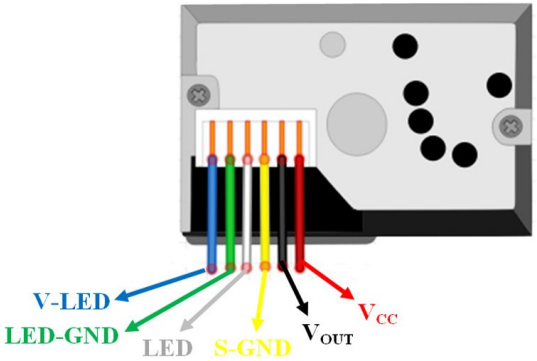



```
def GASsensor():
    global MCP3008_Ch
    Number=6
    Sensor=["GAS-sensor","30","100"] #NAME , <Value=0 ,
>Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values)); #print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 0 ;   #print("no pollution ")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 1 ;   #print("Air pollution")
```

實際感測器皆有誤差需做調整



Dust-Sensor (CH7)




```
def Dustsensor():
    global MCP3008_Ch
    Number=7
    Sensor=["Dust-sensor","30","100"] #NAME , <Value=0 , >Value=1
#Below no need to adjust
    AV = int(("{%s}"%Number).format(*values)); #print(AV)
    if (AV<int(Sensor[1])):
        MCP3008_Ch[Number] = 0 ;   #print("no dust ")
    if (AV>int(Sensor[2])):
        MCP3008_Ch[Number] = 1 ;   #print("pm2.5 hight")
```

實際感測器皆有誤差需做調整