

เริ่มต้นอุปกรณ์ + Micropython



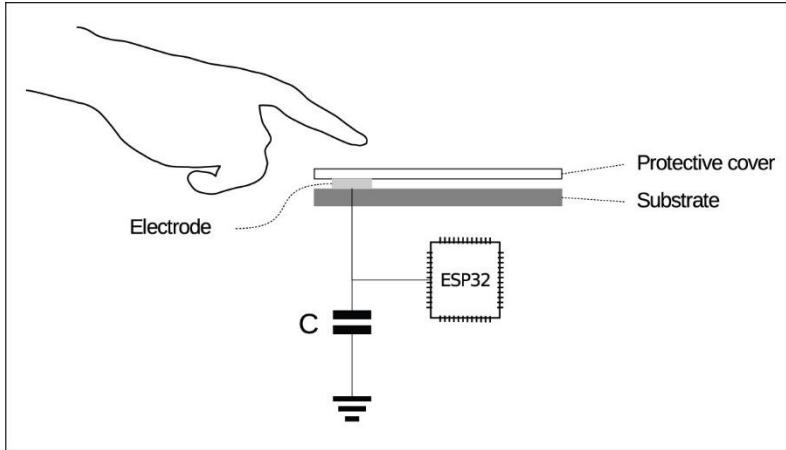
การใช้งานเซนเซอร์ภายใน

- ESP32 ประกอบด้วย เซนเซอร์ภายใน ได้แก่
 - อุณหภูมิภายใน
 - เซนเซอร์สัมผัส
 - เซนเซอร์ตรวจจับสถานะแม่เหล็ก (Hall Sensor)

ឧណអ្នមិតាយីន

```
1 import esp32
2 fahrenheit = esp32.raw_temperature()
3 celsius = (fahrenheit -32.0)/1.8
4 print("Temp %d F = %5.2f C" %(fahrenheit, celsius))
```

เซนเซอร์สัมผัส (Capacity Touch Sensor)



```
1 from machine import TouchPad, Pin
2 import time
3 pinlist= [4,0, 15, 13, 12, 14, 27, 33, 32]
4 for i in pinlist:
5     pin = TouchPad(Pin(i, Pin.IN))
6     value = pin.read()
7     print('Pin {}: Value = {}'.format(i, value))
8     time.sleep_ms(500)
```

เซนเซอร์สัมผัส	พิน GPIO
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	GPIO15 (MTDO)
T4	GPIO13 (MTCK)

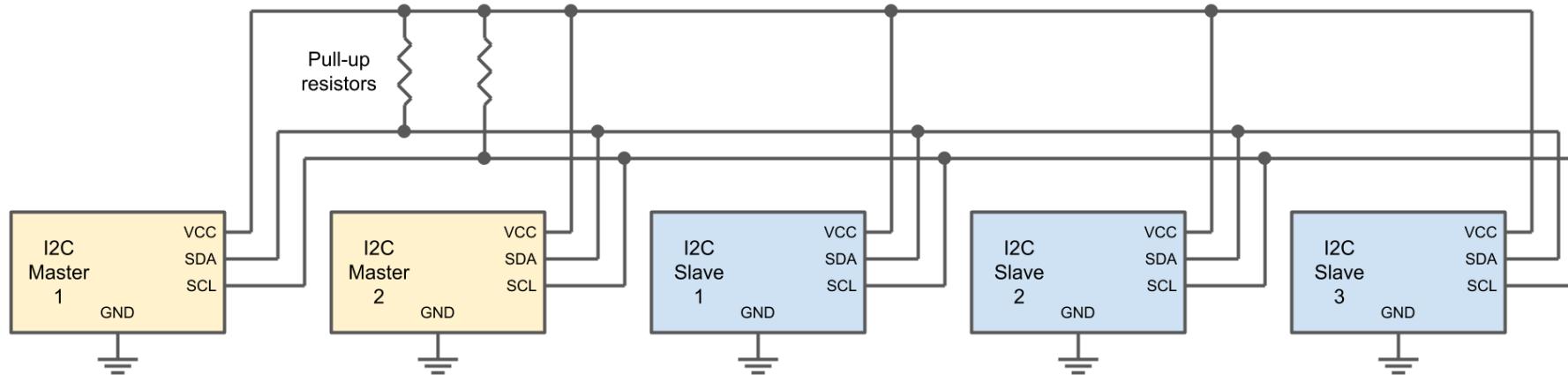
เซนเซอร์สัมผัส	พิน GPIO
T5	GPIO12 (MTDI)
T6	GPIO14 (MTMS)
T7	GPIO27
T8	GPIO33 (32K_XN)
T9	GPIO32 (32K_XP)

เซนเซอร์ตรวจจับสนามแม่เหล็ก (Hall Sensor)

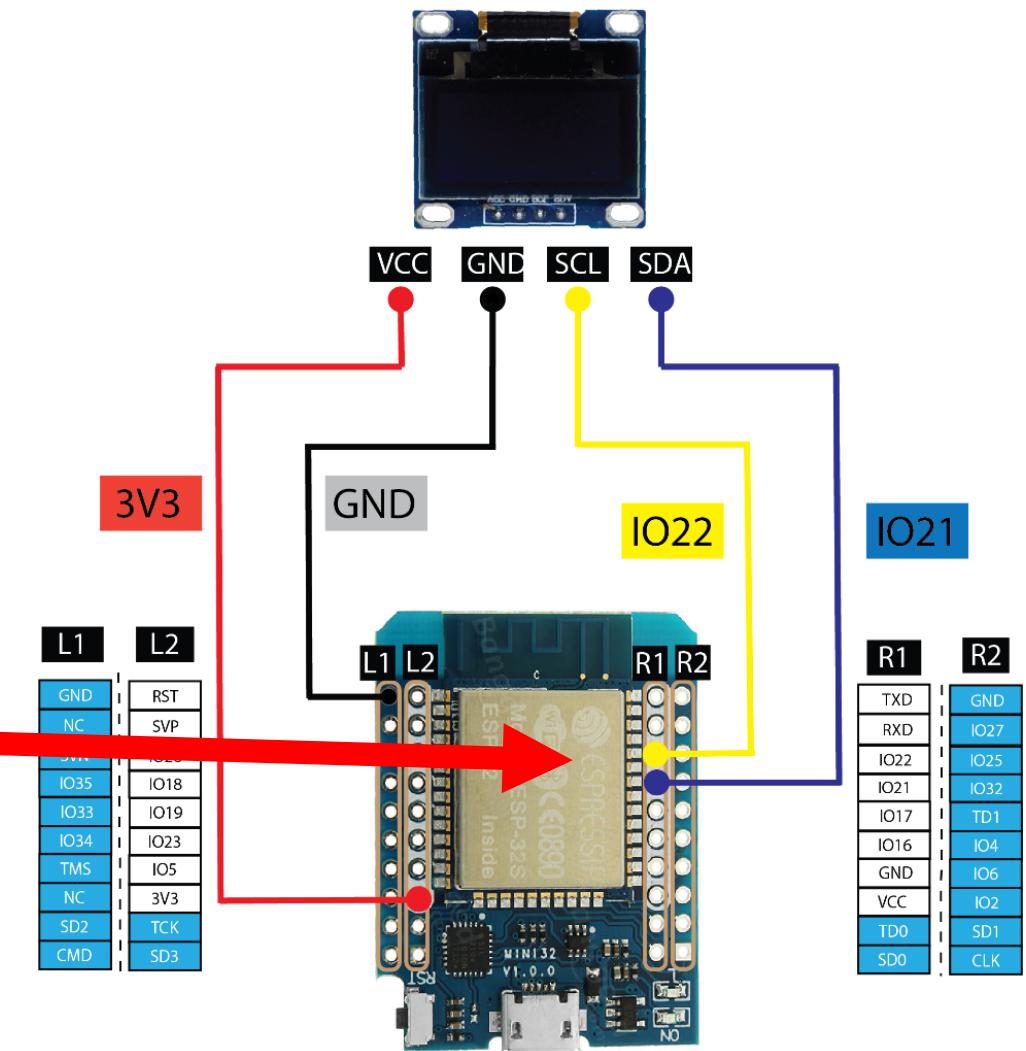
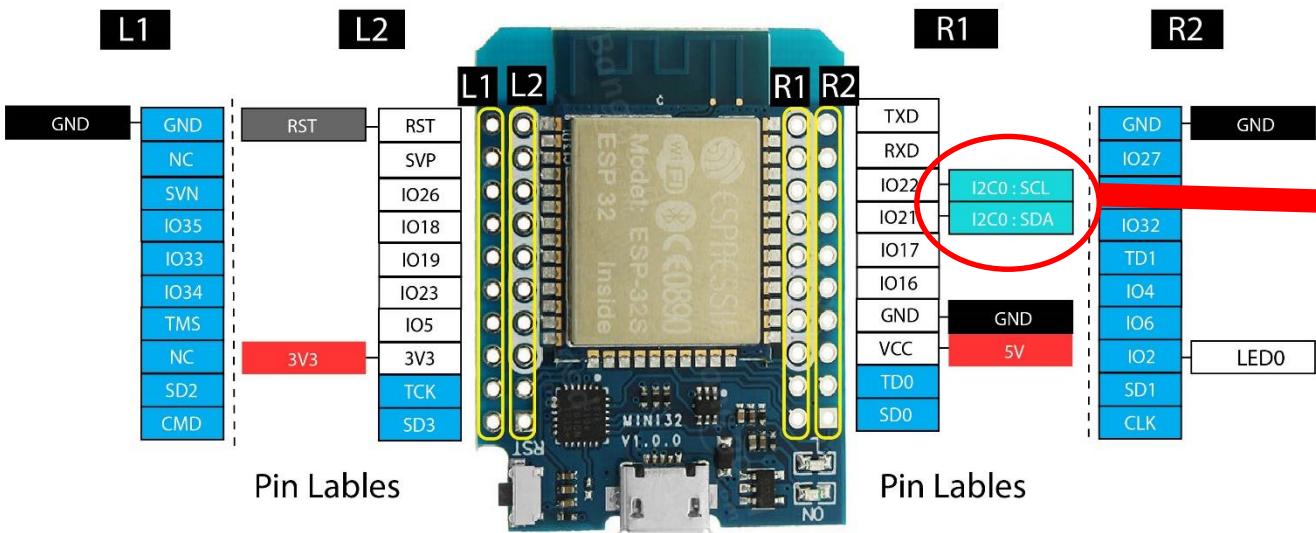
```
1 import esp32
2 esp32.hall_sensor()
3 i = 0
4 for i in range(5):
5     print("Value = %d" %esp32.hall_sensor())
```

การใช้งาน การสื่อสาร I2C

- Inter-Integrated Circuit (I2C) เป็นการสื่อสารรูปแบบบัสสัญญาณ โดยอุปกรณ์ทุกตัวจะเชื่อมบนบัส



ເຮືອມຕ່ອສໍາລັບລາຍ



ทดสอบจอ OLED

วิธีต่อสายใช้งานจอ OLED ผ่าน ESP32 mini

uPyCraft V1.1

File Edit Tools Help

device Serial ssd1306.py

sd board D1306 OLED driver, I2C and SPI interface

uPy... Download DownloadAndRun F5

work Stop BurnFirmware InitConfig Preferences

```
    1 SET_CONTRAST      = const(0x81)
    2 SET_ENTIRE_ON     = const(0xa4)
    3 SET_NORM_INV      = const(0xa6)
    4 SET_DISP           = const(0xae)
    5 SET_MEM_ADDR       = const(0x20)
    6 SET_COL_ADDR       = const(0x21)
    7 SET_PAGE_ADDR      = const(0x22)
    8 SET_DISP_START_LINE = const(0x40)
    9 SET_SEG_REMAP      = const(0xa0)
    10 SET_MUX_RATIO     = const(0xa8)
    11 SET_COM_OUT_DIR    = const(0xc0)

>>>
>>>
>>>

Ready to download this file,please wait!
-----
download ok
```

ดาวน์โหลด ssd1306.py ลงไปยังบอร์ด

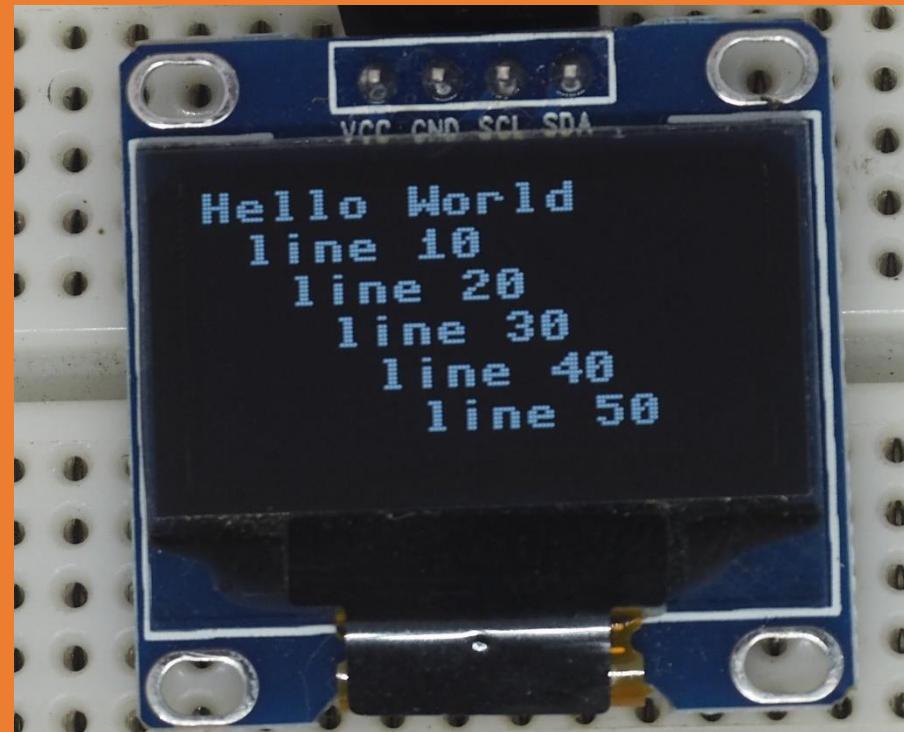
```

from machine import Pin, SoftI2C
import ssd1306

i2c = SoftI2C(scl=Pin(22), sda=Pin(21))

oled = ssd1306.SSD1306_I2C(128, 64, i2c)
oled.text('Hello World ', 0, 0)
for i in range(10, 60, 10):
    oled.text("line {}".format(i), i, i)
oled.show()

```



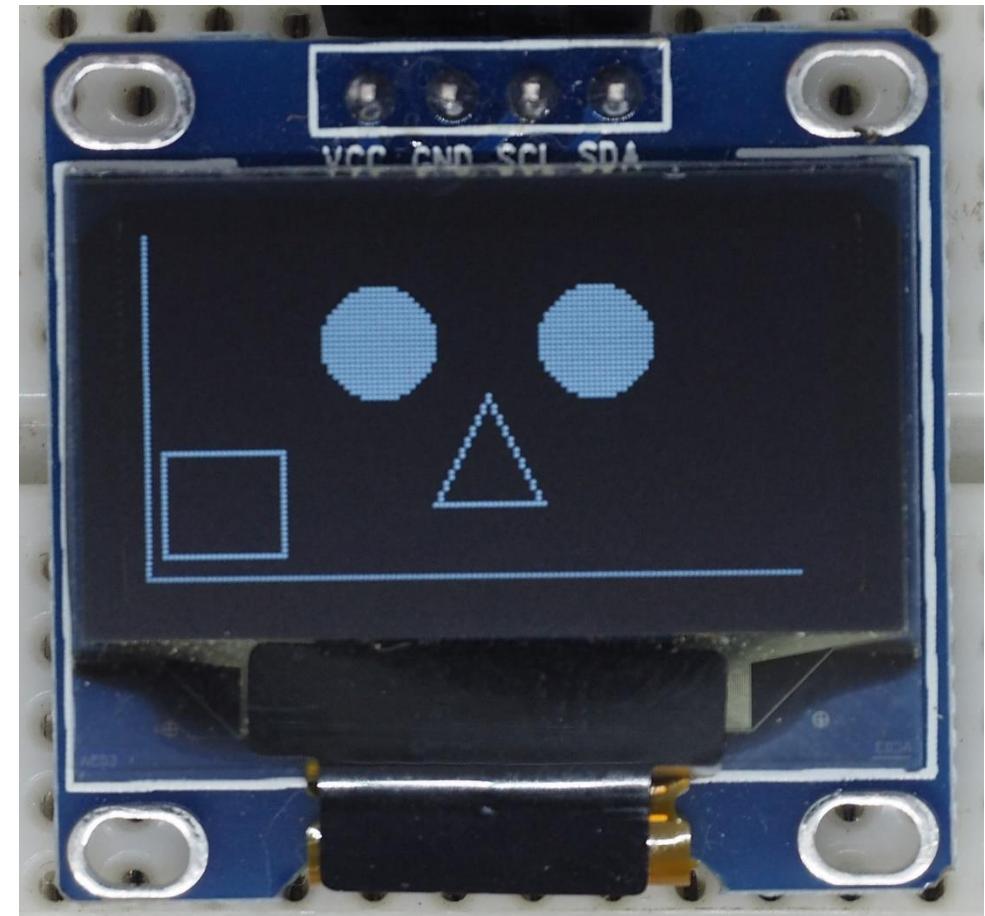
File: https://github.com/ckboa/smartfarm_training-/blob/main/oled_hello.py

ผลที่ควรได้

ฟังก์ชัน	รายละเอียด
fill(x)	$x = 0$ ลบหน้าจอทั้งหมด $x = 1$ เขียนหน้าจอทั้งหมด
contrast(x)	ปรับความสว่างหน้าจอ ค่า x ต้องให้ความสว่างที่ต้อง
poweroff	ปิดหน้าจอ

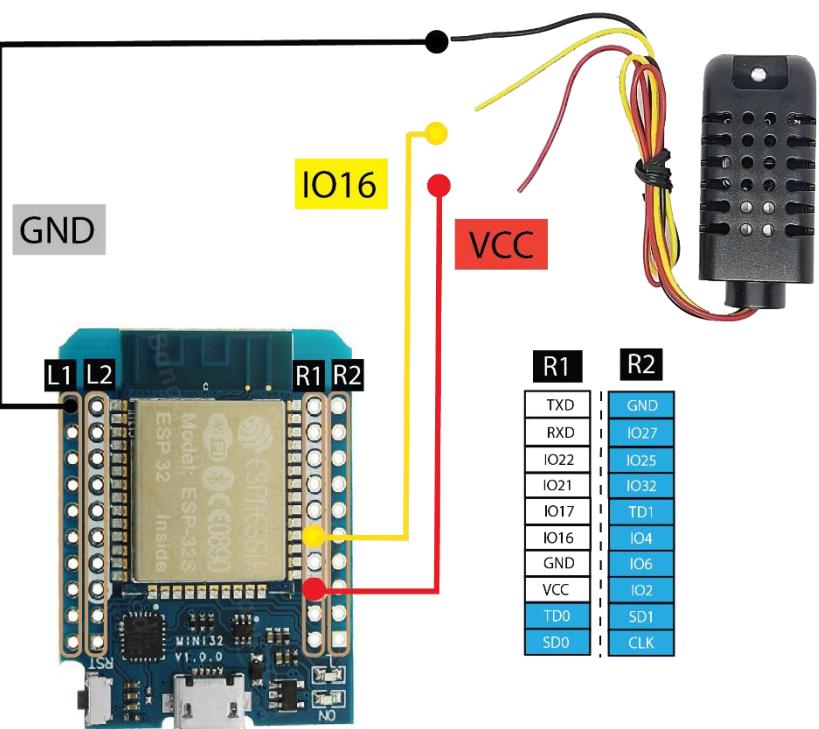
ໂນໂລຢີເພີ່ມເຕີມ gfx ສໍາຮັບວາດຮູ້ປາກ ບ. Adufruit

```
from machine import Pin, SoftI2C  
import ssd1306  
import gfx  
  
i2c = SoftI2C(scl=Pin(22), sda=Pin(21))  
display = ssd1306.SSD1306_I2C(128, 64, i2c)  
  
draw = gfx.GFX(128, 64, display.pixel)  
draw.line(0, 0, 0, 63, 1)  
draw.line(0, 63, 120, 63, 1)  
draw.rect(3,40,23,20,1)  
draw.fill_circle(43, 20, 10, 1)  
draw.fill_circle(83, 20, 10, 1)  
draw.triangle(53,50,73,50,63,30,1)  
display.show()
```



ทดสอบอุณหภูมิ ความชื้น

L1	L2
GND	
NC	SVP
SVN	IO26
IO35	IO18
IO33	IO19
IO34	IO23
TMS	IO5
NC	3V3
SD2	TCK
CMD	SD3



- ป้อนโค้ดต่อไปนี้

```
1 from machine import Pin
2 import dht
3 sensor = dht.DHT22(Pin(16))
4 sensor.measure()
5 temp = sensor.temperature()
6 humi = sensor.humidity()
7 print("temperature: %3.1f" %temp)
8 print("Huminity: %3.1f" %humi)
```

[https://github.com/ckboa/smartfarm_training-
blob/main/DHT_Test.py](https://github.com/ckboa/smartfarm_training/blob/main/DHT_Test.py)

กรณีการต่อสายไม่ดี

```
>>> sensor.measure()
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "dht.py", line 17, in measure
OSError: [Errno 110] ETIMEDOUT
```

ทดสอบการแสดงผลร่วมกับจอ OLED

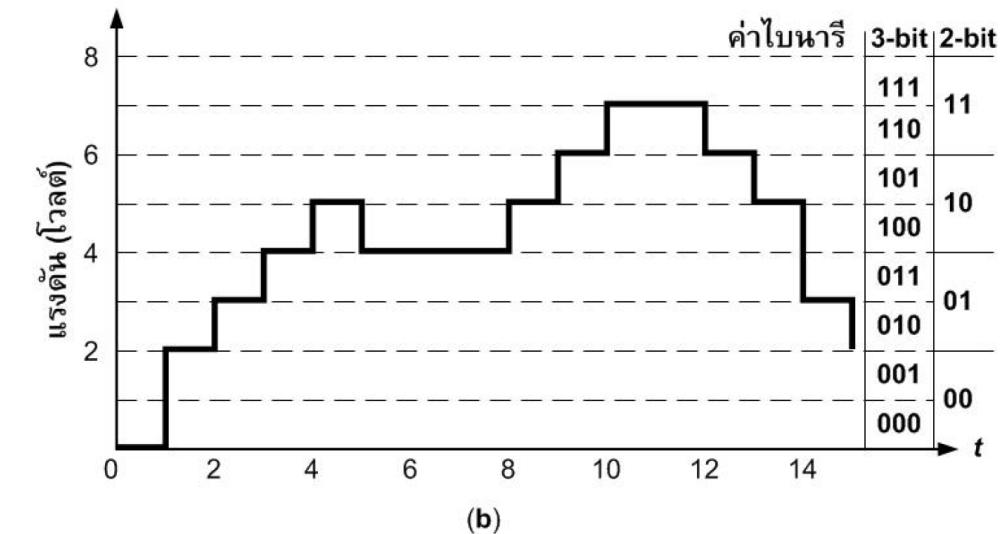
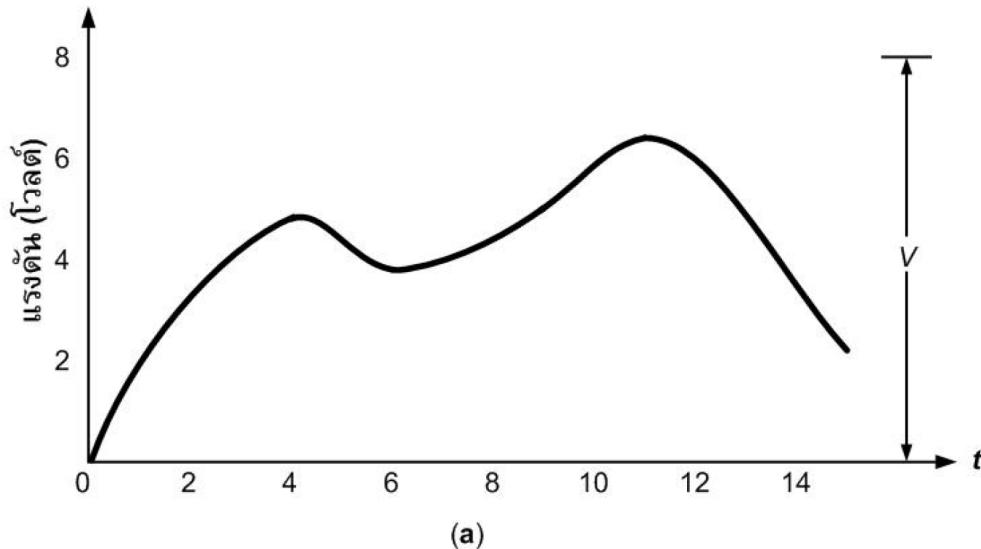
```
1 from machine import Pin, SoftI2C  
2 import ssd1306  
3 import dht  
4  
5 # display setting  
6 i2c = SoftI2C(scl=Pin(22), sda=Pin(21))  
7 display = ssd1306.SSD1306_I2C(128, 64, i2c)  
8  
9 # sensor reading  
10 sensor = dht.DHT22(Pin(16))  
11 sensor.measure()  
12  
13 temp = sensor.temperature()  
14 humi = sensor.humidity()  
15  
16 temp_str = "Temp: {0:3.1f}".format(temp)  
17 humi_str = "Humi: {0:3.1f}".format(humi)  
18  
19 display.text(temp_str, 0, 0)  
20 display.text(humi_str, 0, 10)  
21  
22 display.show() |
```

- เพิ่มเติมส่วน ในการล็อก 4 เหลี่ยม เพื่อการแสดงผลออกทางจอภาพ

ให้อ่านค่าอุณหภูมิต่อเนื่องและแสดงผลที่ได้บนจอ OLED
(อาจแสดงค่าอื่น ๆ พร้อมกันด้วย)

ADC

- Analog to Digital Conversion (ADC) เป็นการใช้เพื่อแปลงสัญญาณอนาล็อกไปเป็นดิจิทัล



การลดทอน สัญญาณ

(สำหรับอนาล็อก)

- ADC.ATTN_0DB: 0dB รองรับแรงดันสูงสุด 1.0 ไวโอล์ต(ดีฟอลท์)
- ADC.ATTN_2_5DB: ลดthon 2.5dB รองรับแรงดันสูงสุดประมาณ 1.34v
- ADC.ATTN_6DB: ลดTHON 6dB รองรับแรงดันสูงสุดประมาณ 2.00v
- ADC.ATTN_11DB: ลดTHON 11dB รองรับแรงดันสูงสุดประมาณ 3.6v

การกำหนด จำนวนบิตของ สัญญาณ

- ADC.WIDTH_9BIT: ข้อมูล 9 บิต
- ADC.WIDTH_10BIT: ข้อมูล 10 บิต
- ADC.WIDTH_11BIT: ข้อมูล 11 บิต
- ADC.WIDTH_12BIT: ข้อมูล 12 บิต (ดีพอลท์)

รีเลย์

ประเภทของรีเลย์

1. รีเลย์หัวไป

2. โซลิดสเตตอรีเลย์ (Solid State Relay)



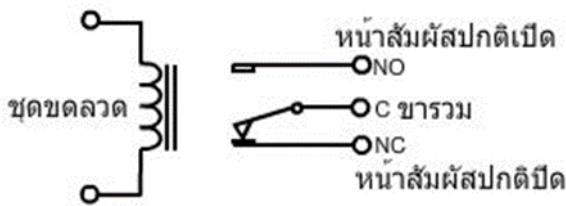
รีเลย์ (Relay)	โซลิดสเตตอรีเลย์ (Solid State Relay)
อายุการใช้งานสั้น เนื่องจากหน้าสัมผัสแบบแมคคานิค	มีอายุการใช้งานนาน
มีเสียงระหว่างการตัดต่อ	ไม่มีเสียงเวลาตัดต่อ
อาจเกิดสัญญาณรบกวน เนื่องจากการตัดต่อ	ไม่เกิดการรบกวน
สามารถตรวจสอบได้ง่าย	เกิดความร้อนการใช้งานเป็นเวลานาน ควรมีการใช้ชีสซิงค์เพื่อบาധความร้อน

ส่วนประกอบของรีเลย์ทั่วไป

1. ขดลวด (Coil)



2. หน้าสัมผัส (Contact)

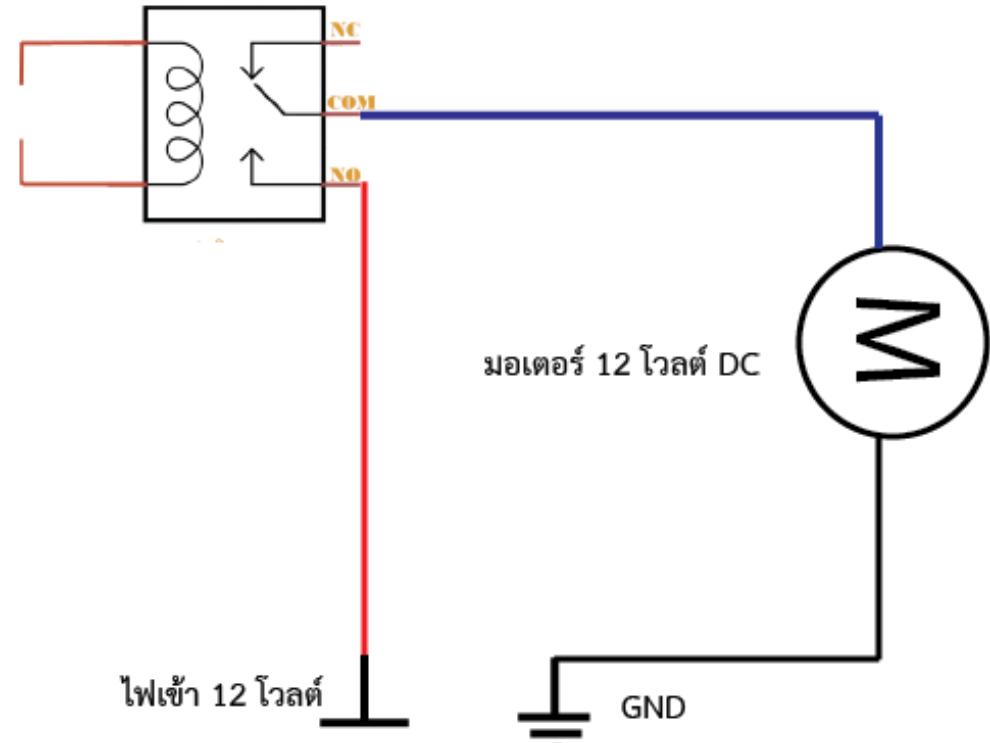


- จุดต่อ NC : จุดต่อ NC ย่อมาจาก normal close หมายความว่า ปกติปิดหรือหากยังไม่จ่ายไฟให้ขดลวดเห็นี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา
- จุดต่อ NO : จุดต่อ NO ย่อมาจาก normal open หมายความว่า ปกติเปิดหรือหากยังไม่จ่ายไฟให้ขดลวดเห็นี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิด
- จุดต่อ C : Common (C) หมายถึง จุดร่วมที่ต่อมาจากการแหล่งจ่ายไฟ

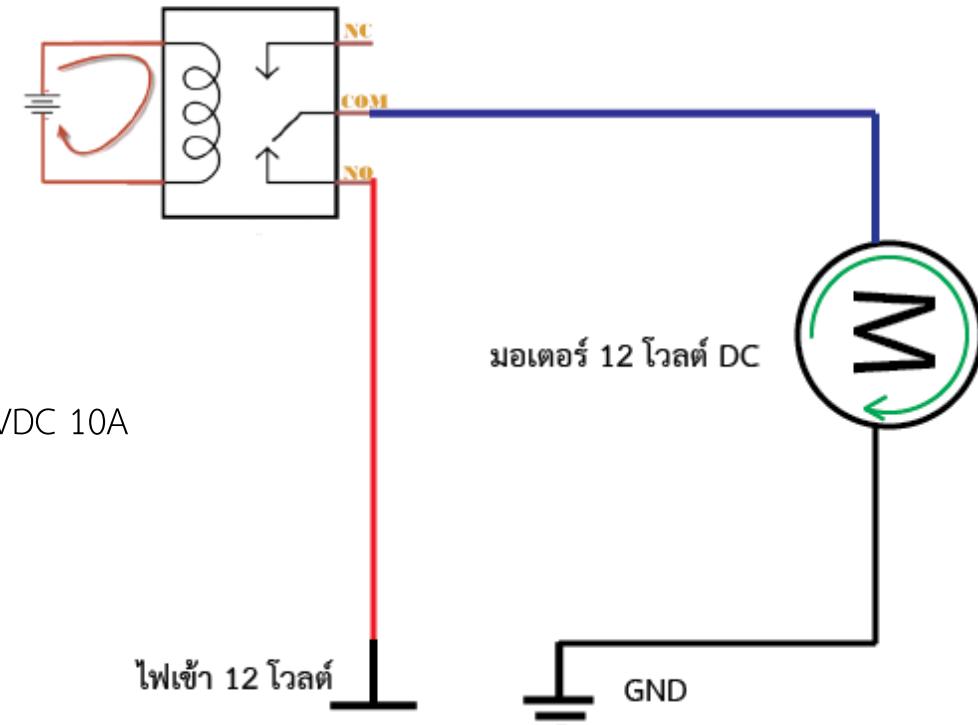
ข้อคำถึงในการ เลือกซื้อหรือใช้ งานรีเลย์ทั่วไป

1. พิกัดแรงดัน
2. ชนิดของโหลด
3. พิกัดโหลด
4. ขั้วต่อของรีเลย์ (Terminals)
5. ตัวบอกรสถานะการทำงาน (Indicator)

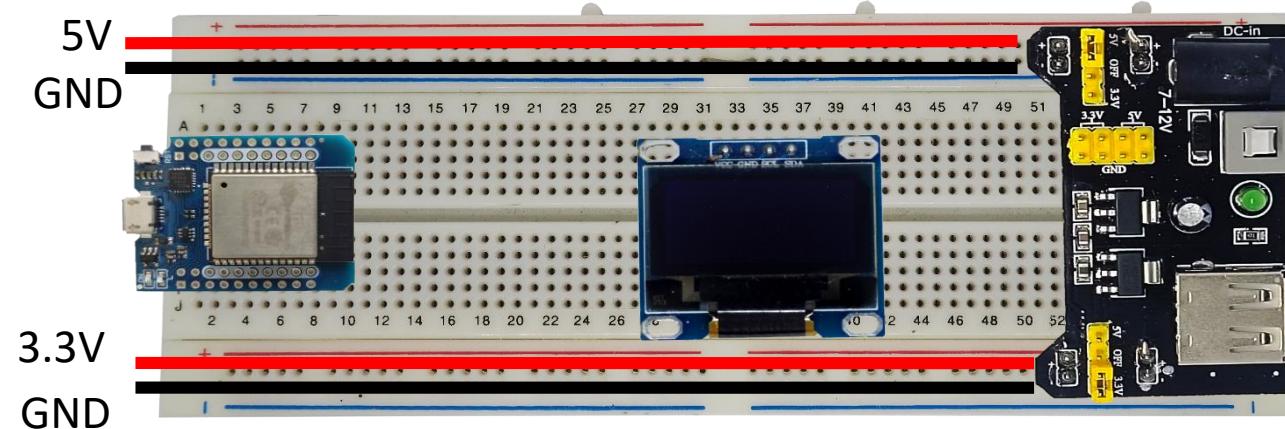
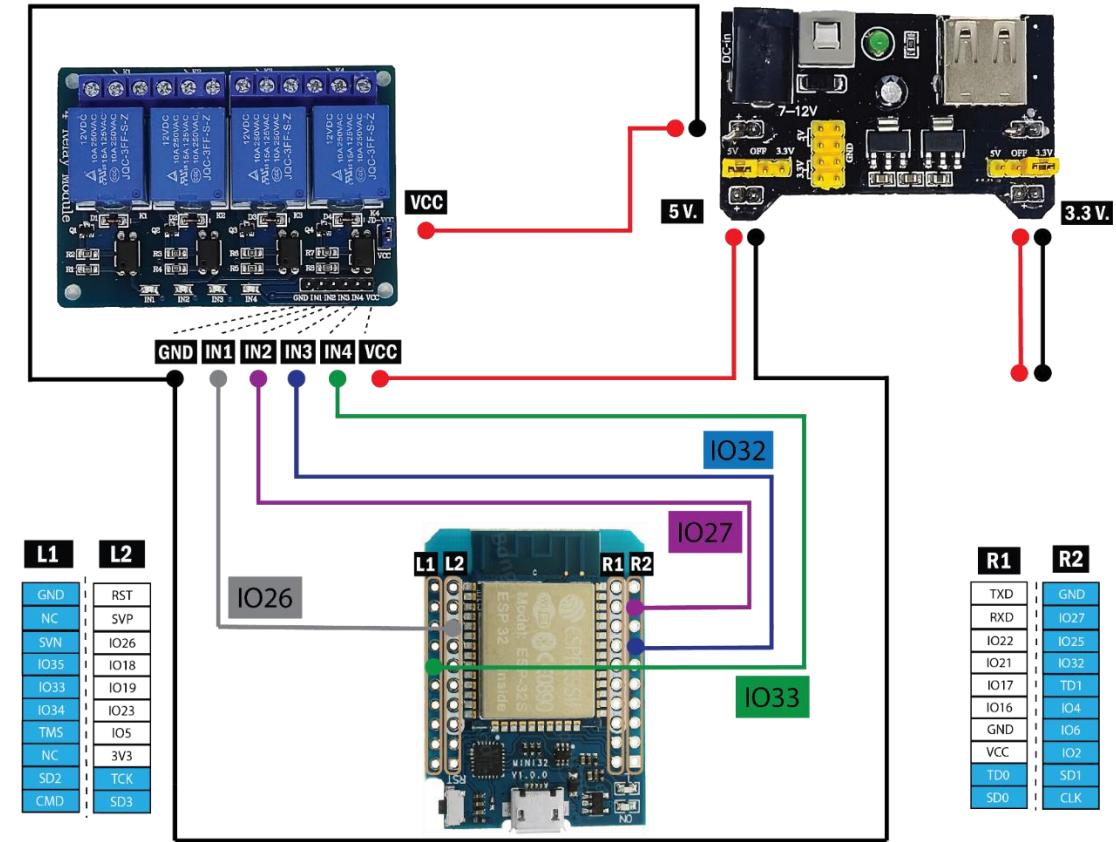
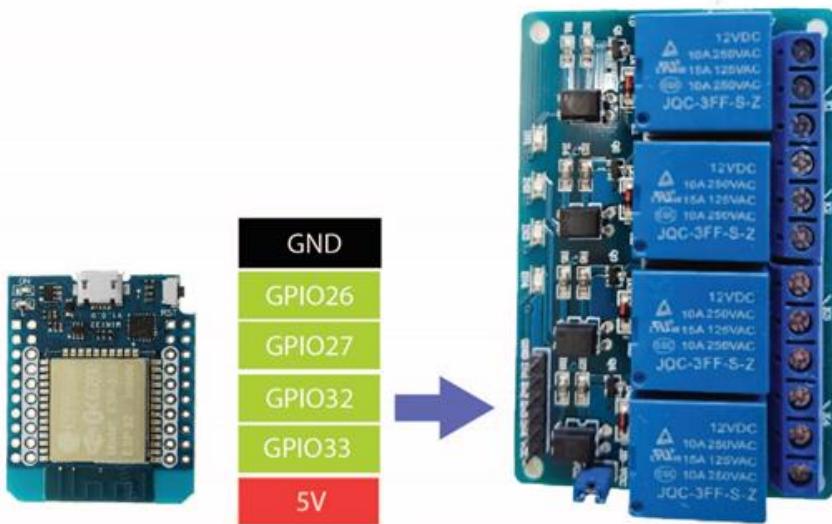
รีเลย์บันบอร์ด



- กระแสเต็มข่านาด 12 โวลต์
- แบบ SPDT
- หน้าสัมผัสขนาด 120VAC/24VDC 10A
หรือ 250VDC 10A/6A



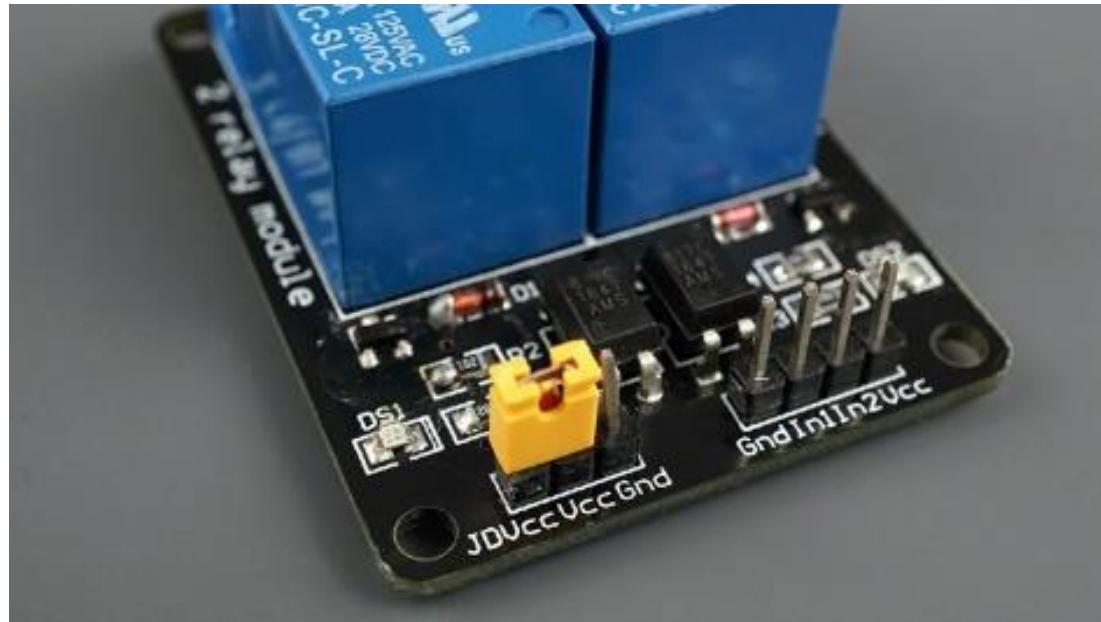
การเชื่อมต่อรีเลย์



การต่อวีเลย์

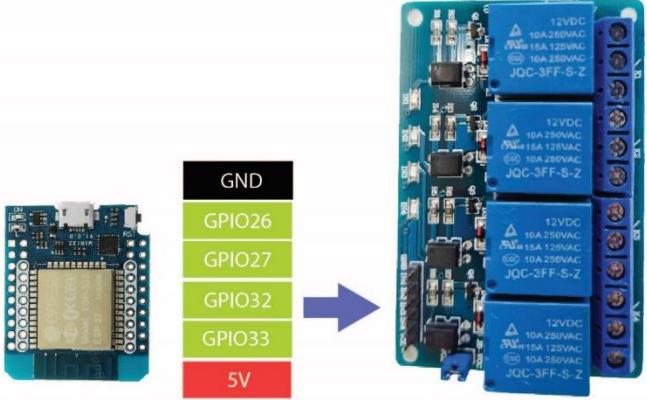
วิธีการต่อสายใช้งานวีเลย์

การจ่ายไฟให้กับรีเลย์



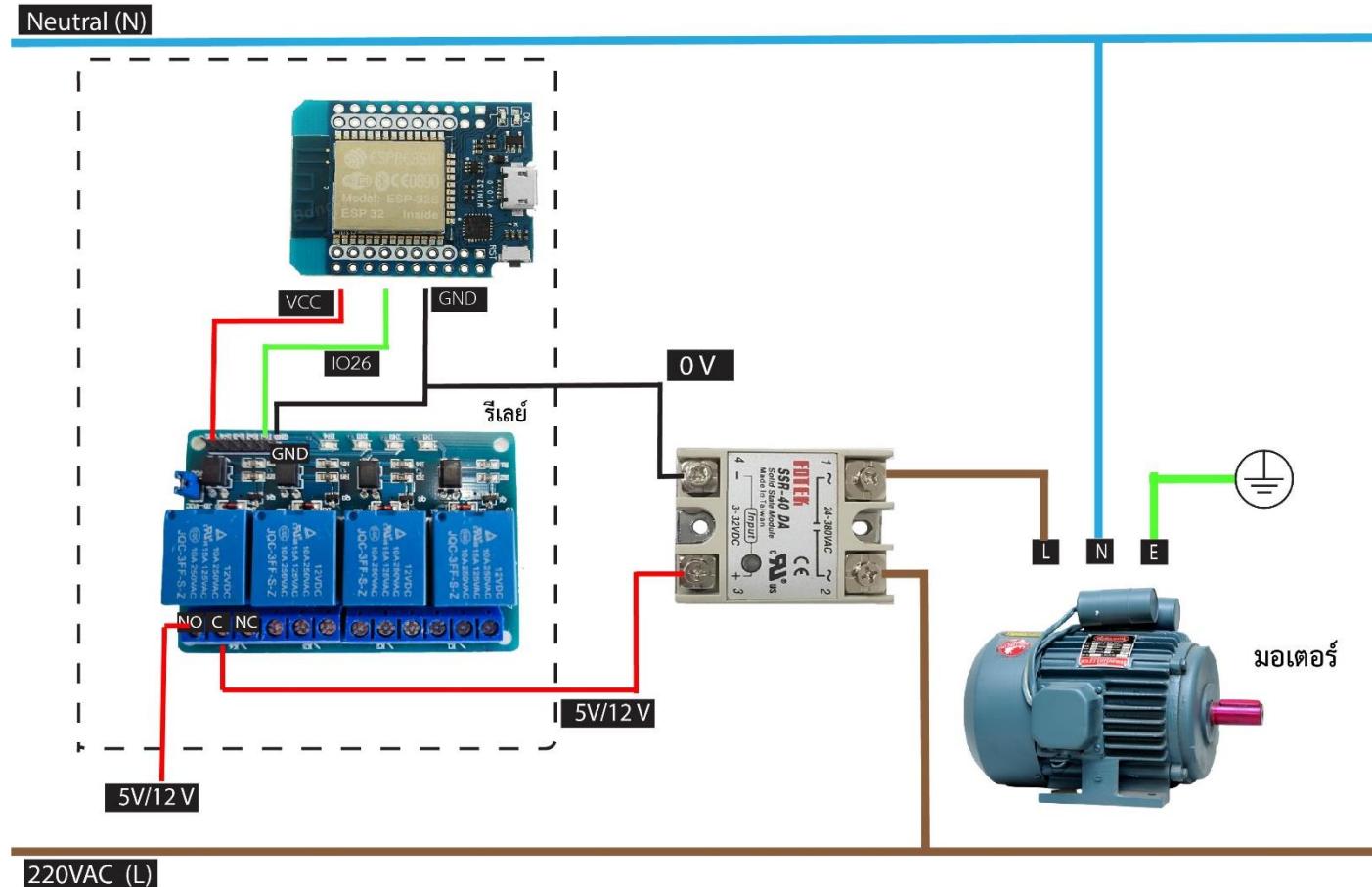
ข	หน้าที่	หมายเหตุ
1	GND	เชื่อมต่อกราวน์ด์ร่วมกับวงจรภายใน
2	VCC	ขาไฟเลี้ยงบนบอร์ดวงจร ร่วมไฟขาเข้า
3	JD-VCC	ขาไฟเลี้ยงเพื่อขับรีเลย์

ทดสอบบอร์ด 4 ช่องสัญญาณ

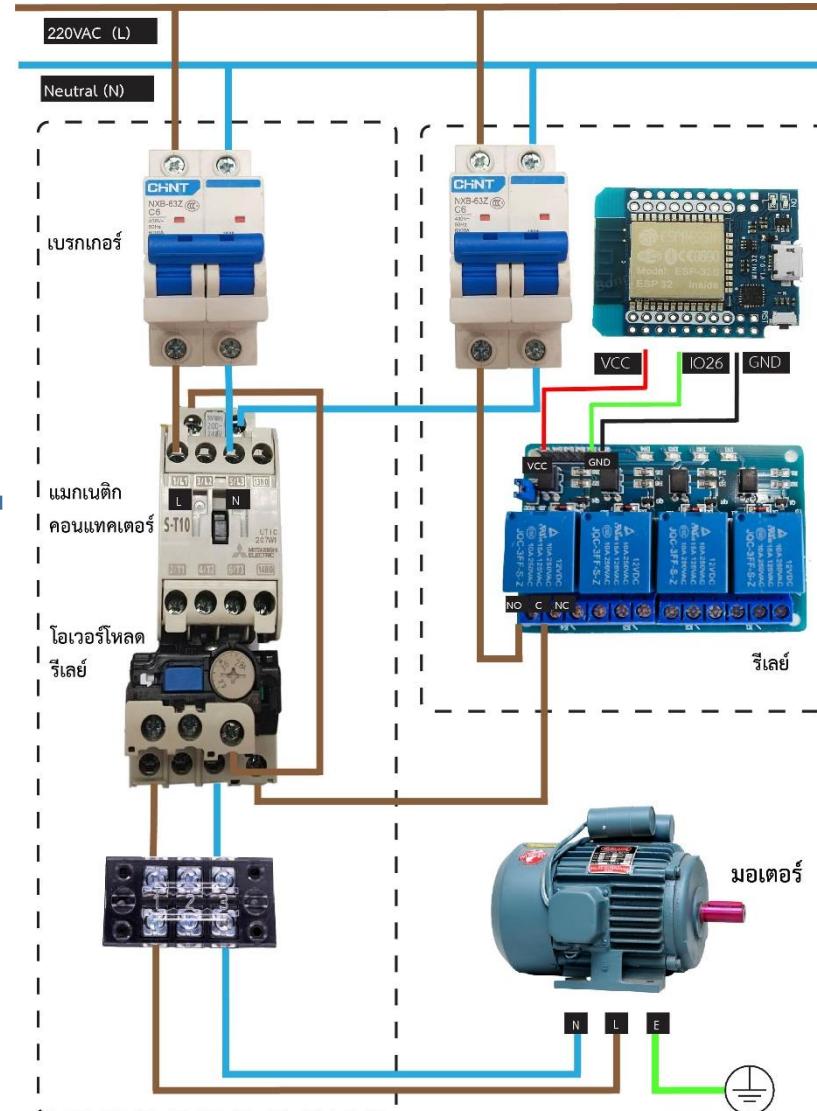
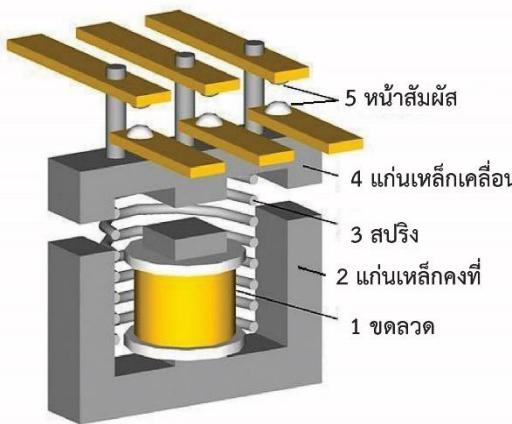


```
from machine import Pin  
relay1 = Pin(26, Pin.OUT)  
relay2 = Pin(27, Pin.OUT)  
relay3 = Pin(32, Pin.OUT)  
relay4 = Pin(33, Pin.OUT)  
relay1.value(0) # on  
relay1.value(1) # off
```

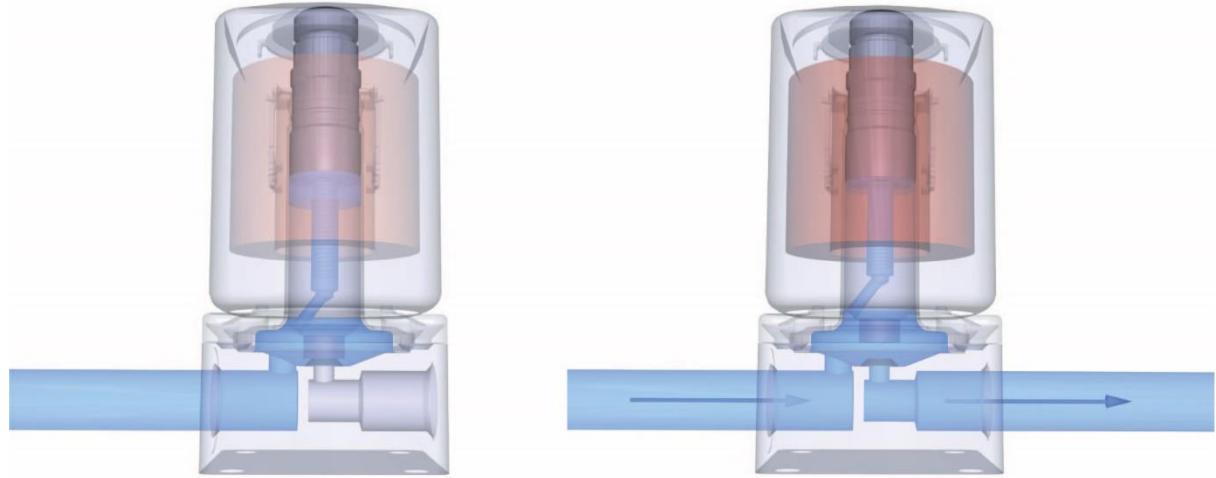
การใช้งาน Solid State Relay เพื่อขับมอเตอร์



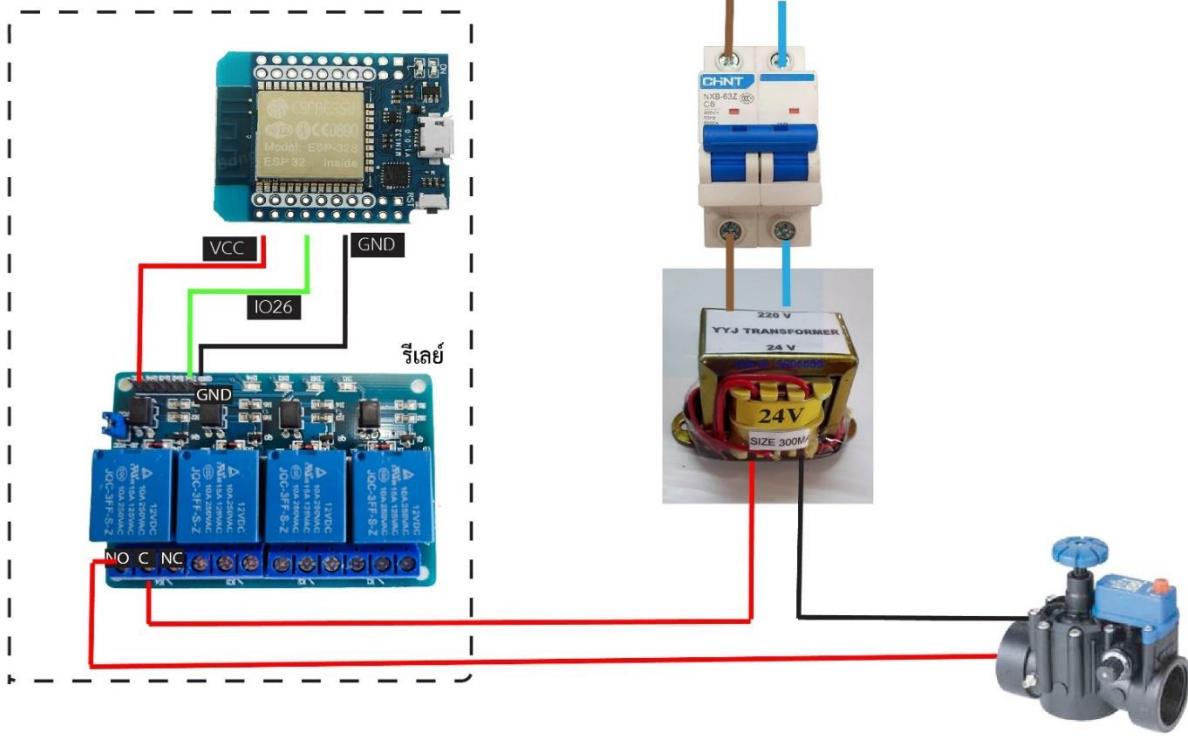
การใช้งาน Magnetic Relay และ Overload เพื่อขับมอเตอร์



ໂჟლິນອຍດ້ວລ້າ

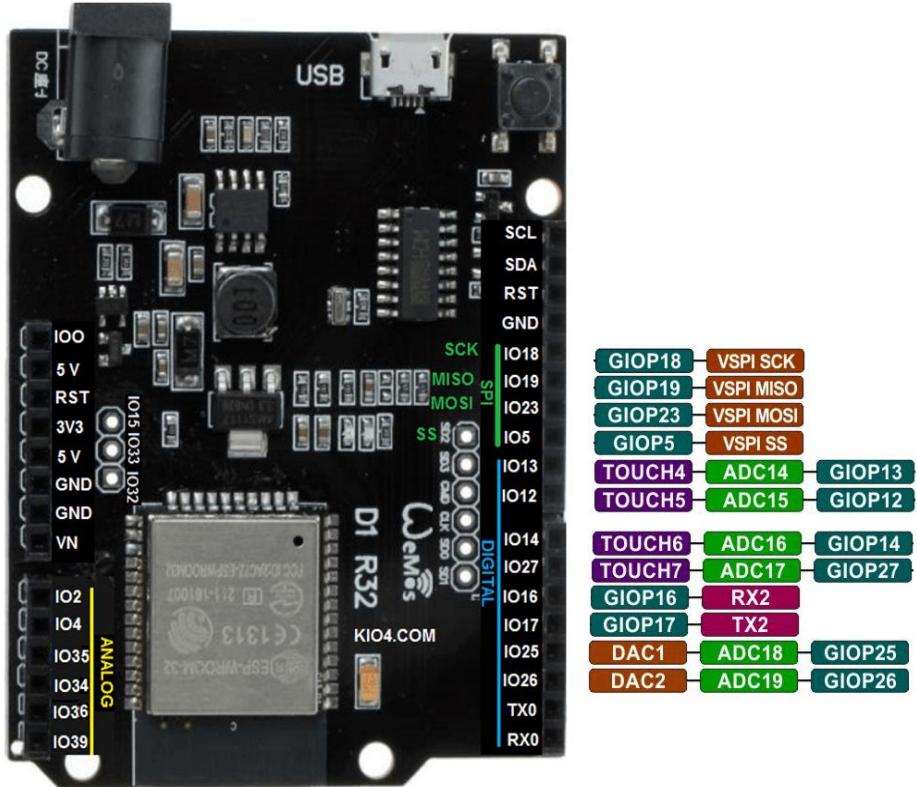
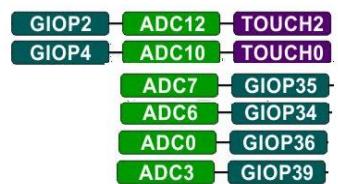


ສភາວະປກຕີ ແລະ ພັງຈາກສັ່ງທຳງານ



ทดลอง

- ให้เขียนโปรแกรมควบคุมการทำงานของรีเลย์
 - 1. ใช้ Touch sensor เมื่อสัมผัสแล้วให้ รีเลย์ทำงาน
 - 2. แสดงสถานะ การทำงานของรีเลย์บนหน้าจอ
 - 3. ต่อรีเลย์เพื่อเปิดปิดพัดลมขนาดเล็ก/หลอดไฟ LED



Hint (Touch)

```
1 from time import *
2 from machine import TouchPad, Pin
3
4 touch = TouchPad(Pin(14))
5 print("Pin 14 touch value: {}".format(touch.read()))
6 while True:
7     sleep(3)
8     print("Touch value : {}".format(touch.read()))
```

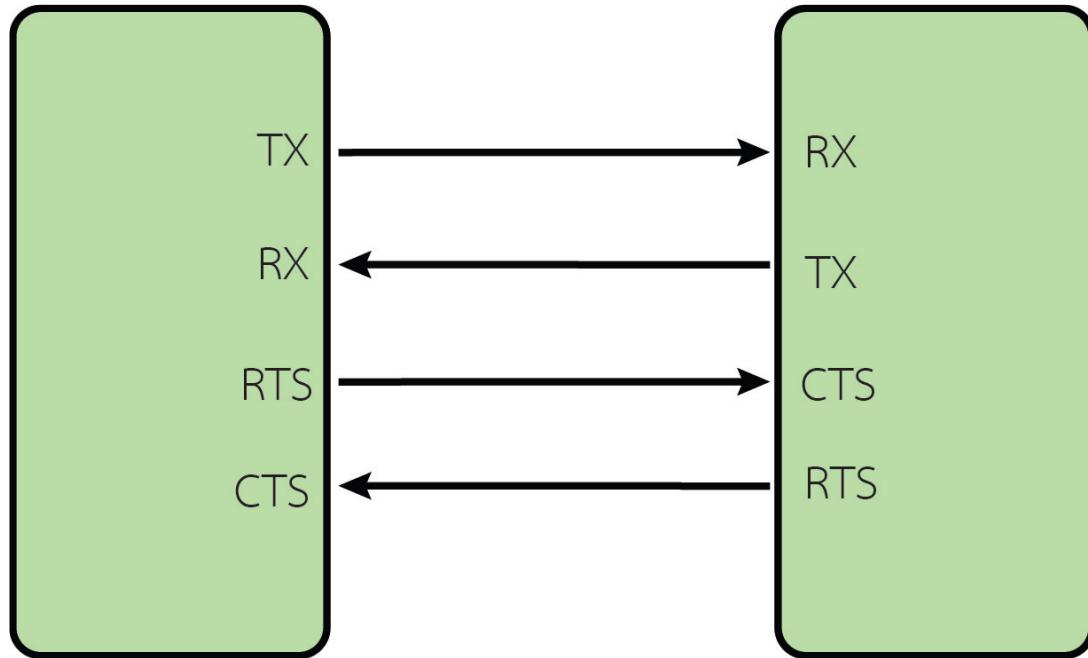
การสื่อสาร UART

- Universal Asynchronous ReceiverTransmitter (UART) เป็นช่องทางการสื่อสารแบบอะซิงโครนัสแบบอนุกรม



การใช้งาน UART

- ESP32 UART PIN

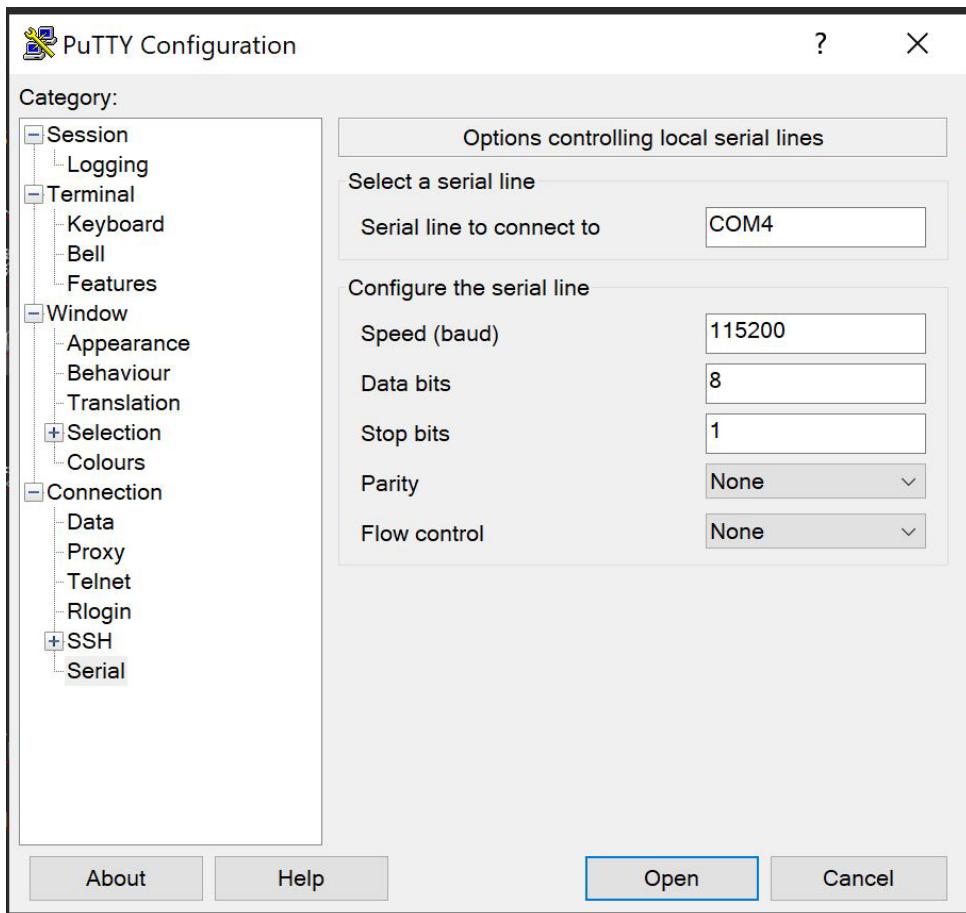
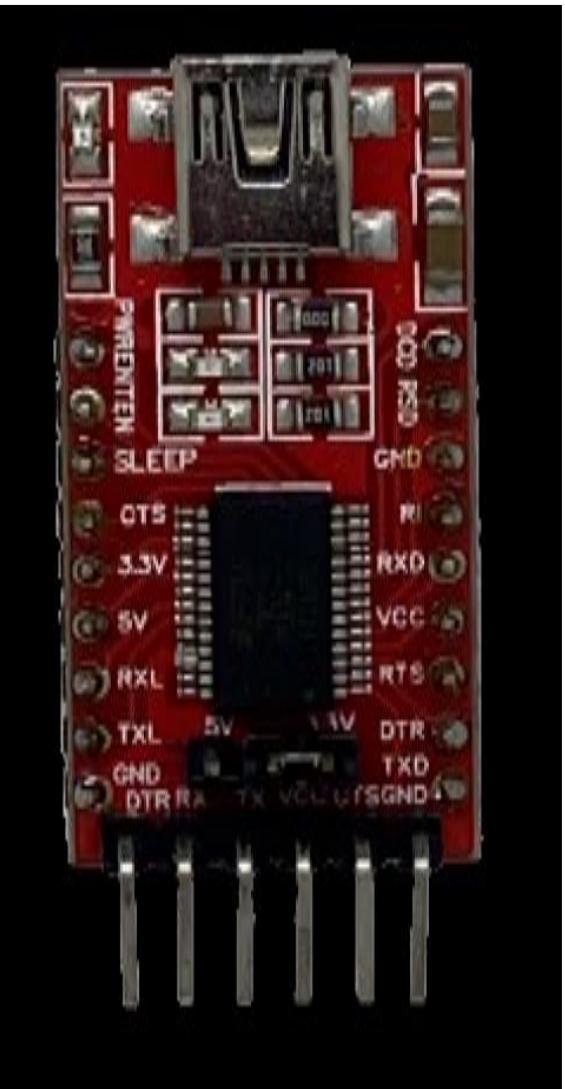


UART	RX	TX	CTS	RTS
UART0	GPIO3	GPIO1	N/A	N/A
UART1	GPIO9	GPIO10	GPIO6	GPIO11
UART2	GPIO16	GPIO17	GPIO8	GPIO7

- ตัวอย่างการกำหนดค่า

```
1 from machine import UART
2 uart = UART(1, 9600)
3 uart.init(9600, bits=8, parity=None, stop=1)
4
5 uart = UART(2, baudrate=115200, bits=8, parity=None, stop=1, timeout=5000)
```

การส่งค่าไปยังคอมพิวเตอร์



การอ่านเขียน UART

```
1 from machine import UART
2 uart = UART(2, baudrate=115200, bits=8, parity=1, stop=1, timeout=10)
3 uart.write("From TX PIN")
4 print(uart.read())
```

แบบฝึกหัด

ให้จับคู่สิ่งที่อยู่ในบอร์ด A กับบอร์ด B ที่ส่งค่าเซนเซอร์อุณหภูมิและความชื้นของจากบอร์ด A ไปแสดงผลยังบอร์ด B

