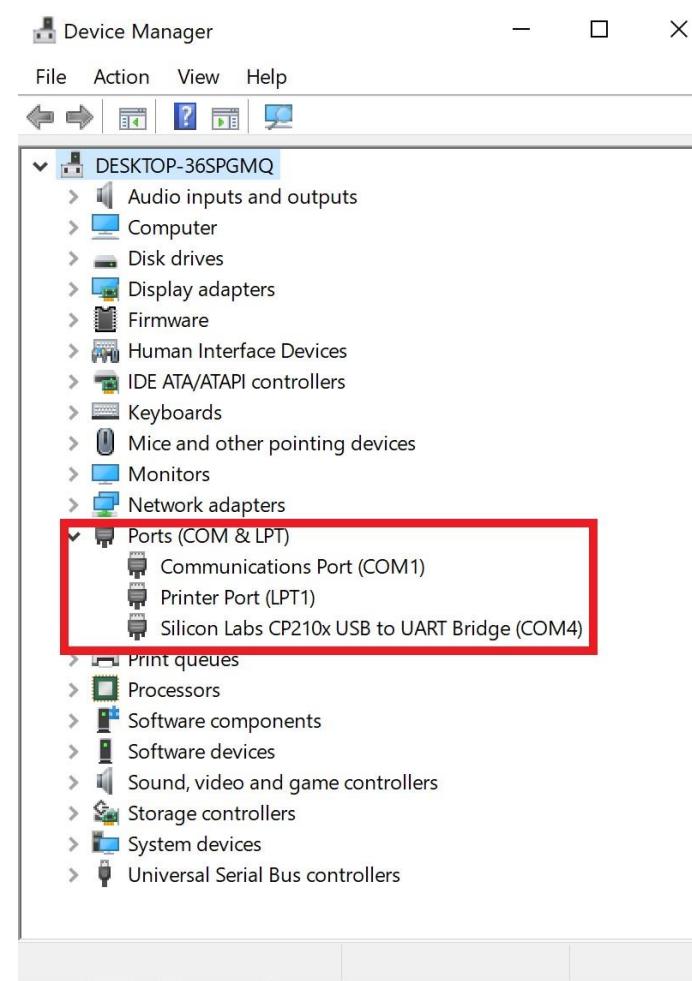
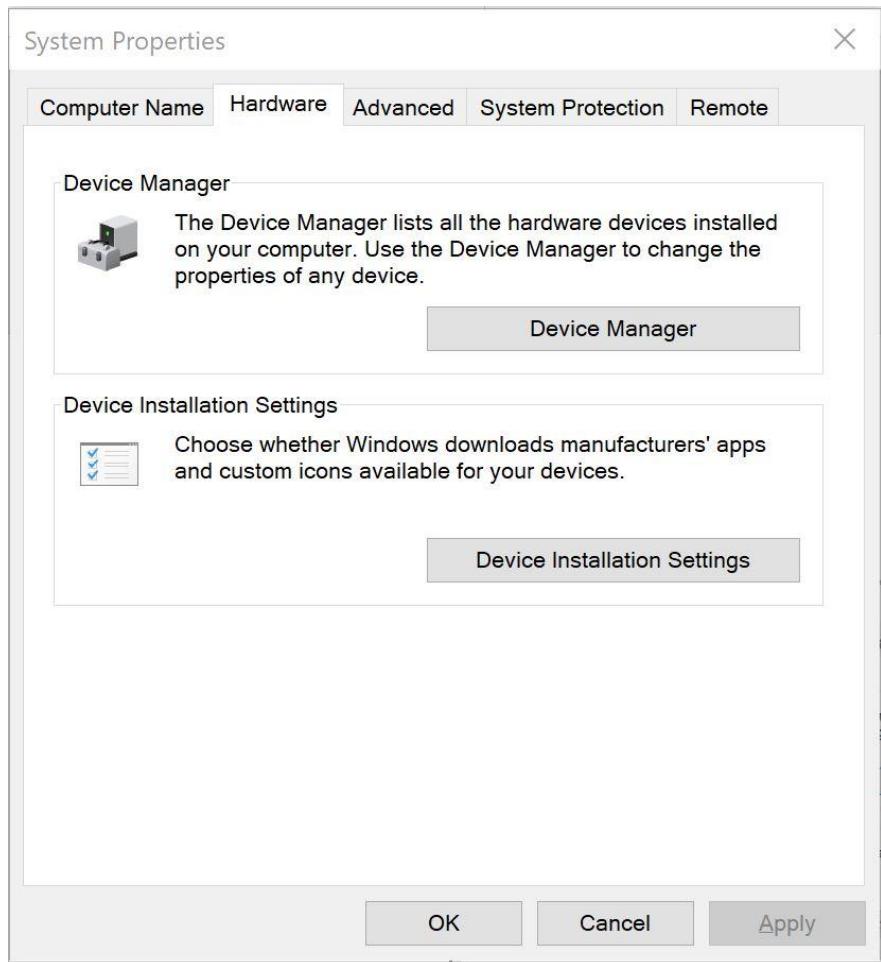


เริ่มต้นอุปกรณ์ + Micropython



ตรวจสอบหมายเลขคอมพอร์ต (com port)

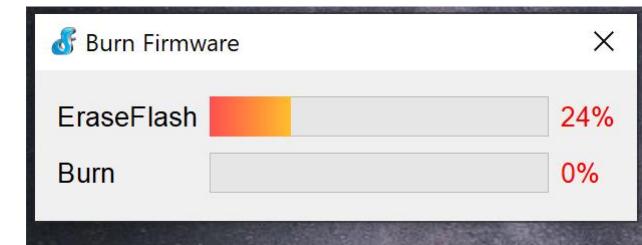
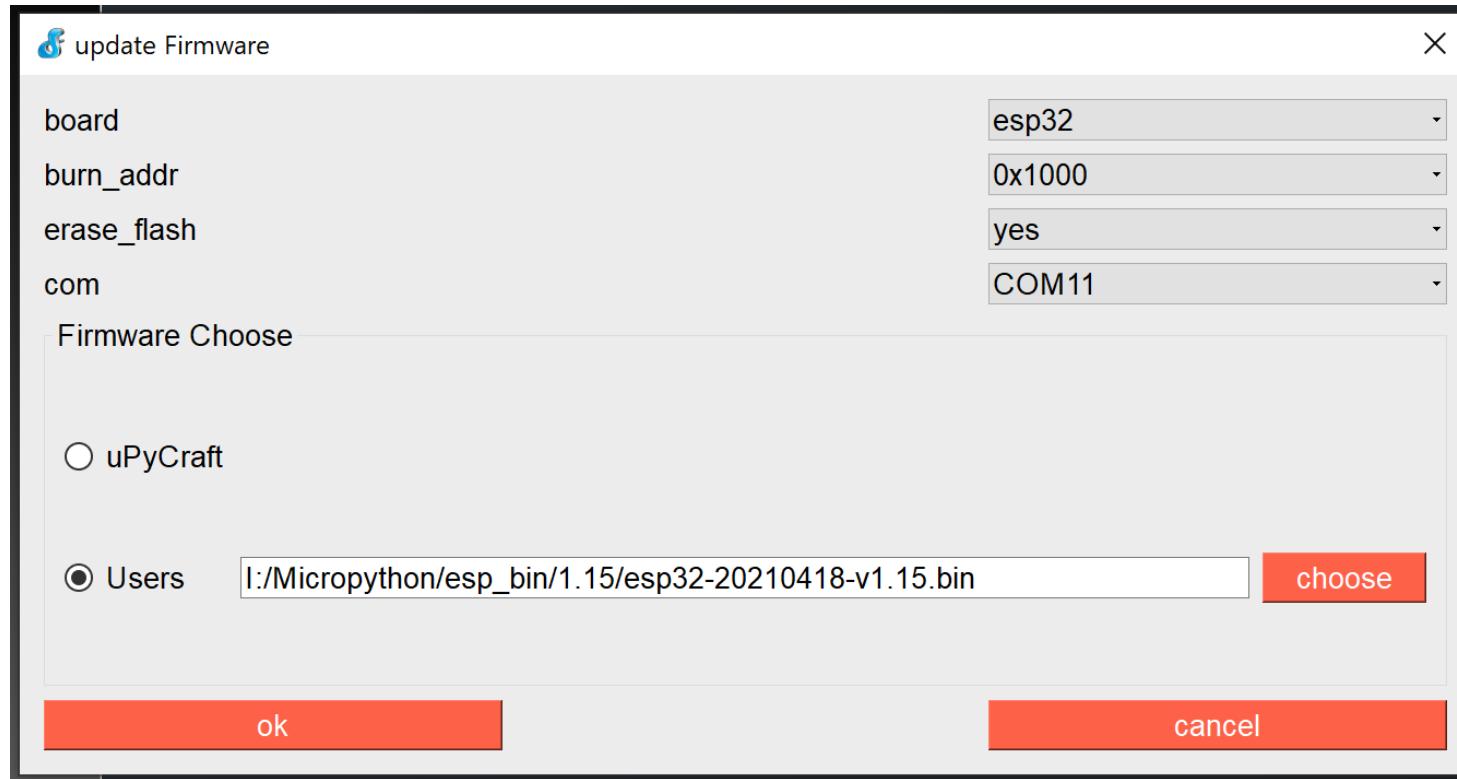


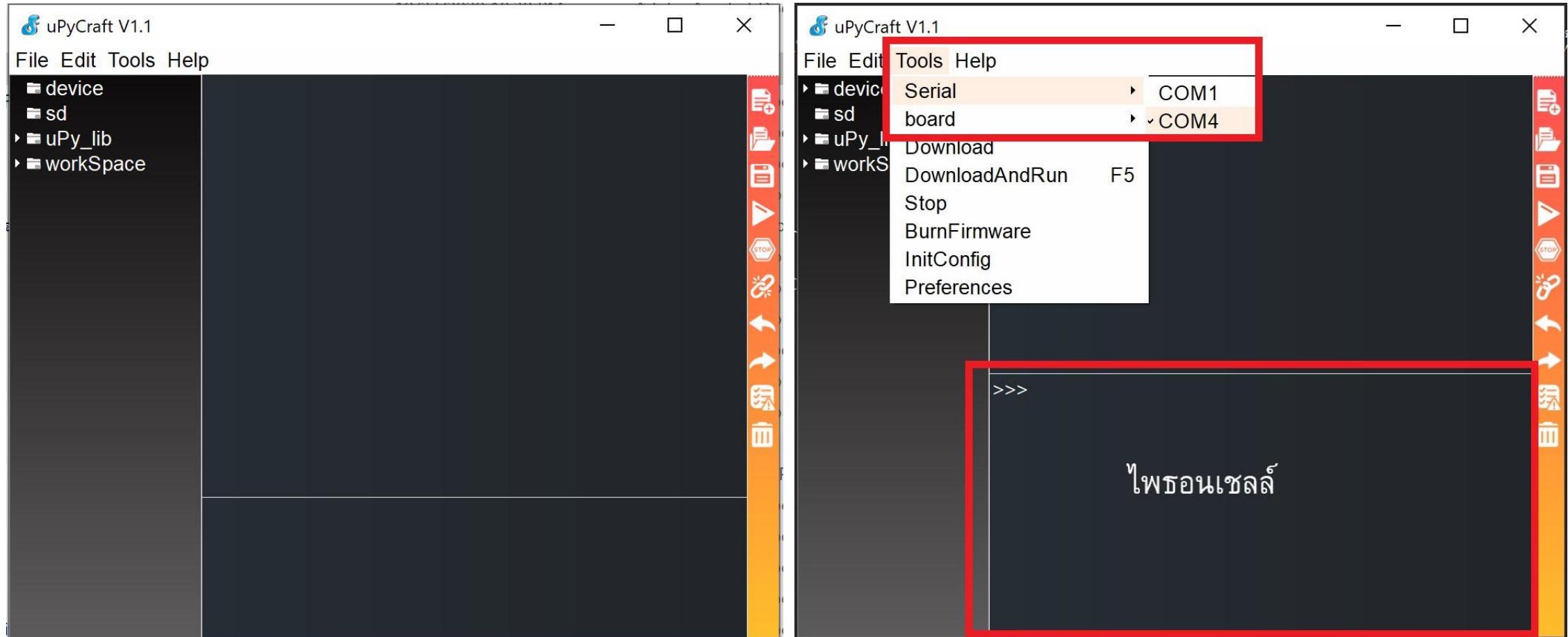
หากไม่พบ หมายเลข com port
ให้ติดตั้งไดร์เวอร์ก่อน

- โปรแกรม uPyCraft IDE สำหรับ Windows



การติดตั้ง Micropython ผ่าน editor





เริ่มต้นใช้งาน

โมดูล Machine สำหรับไมโครไฟฟอน

- from module import xxxx
- PWM ใช้สร้างสัญญาณ PWM ช่วงความถี่จาก 1 Hz ถึง 40 MHz
- ADC ใช้แปลงจากสัญญาณอนาลอกเป็นดิจิทัล อยู่ที่ขาระหว่าง 32 ถึง 39
- DAC ใช้แปลงจากสัญญาณดิจิทัลเป็นอนาลอก
- Pin ใช้ควบคุมการทำงานของชุด I/O
 - ขา 1 และ 3 ใช้สำหรับ REPL UART TX and RX respectively
 - ขา 6, 7, 8, 11, 16, และ 17 สำหรับเชื่อมต่อ embedded flash
 - ขา 34-39 ใช้เป็น input เท่านั้น ไม่มีตัวต้านทาง pull-up

ปรับแรงดัน HW-131 ให้ถูกต้อง

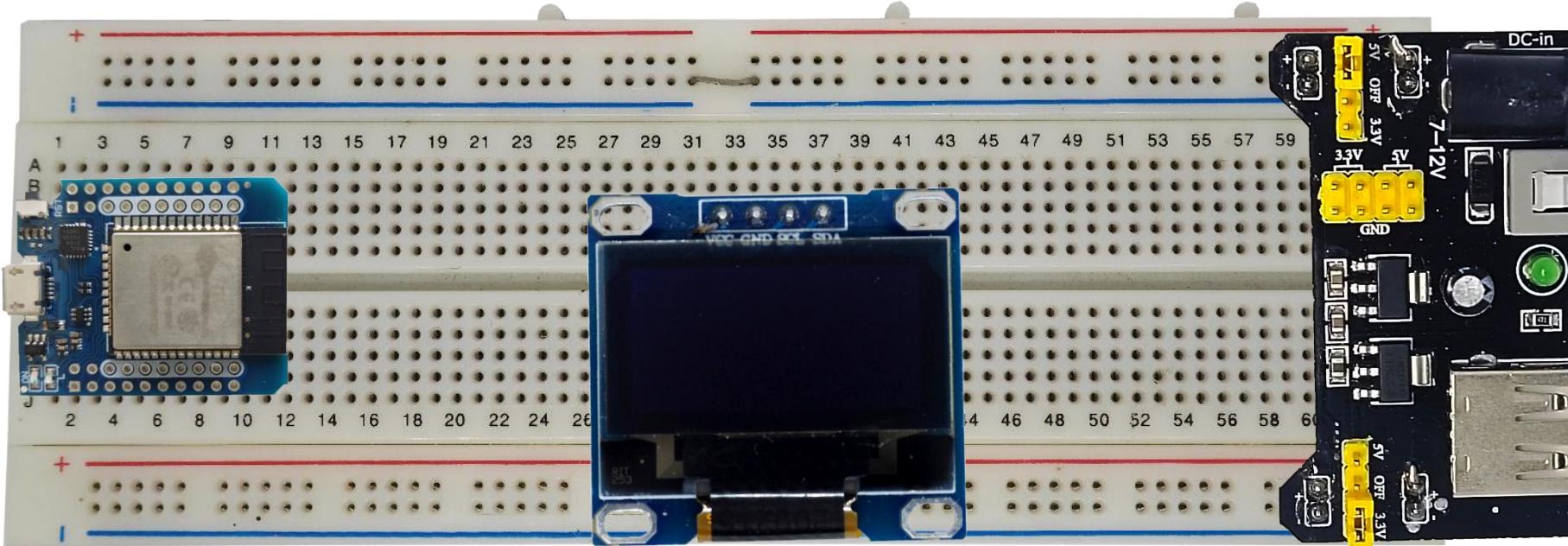


5V สำหรับ ด้านบน

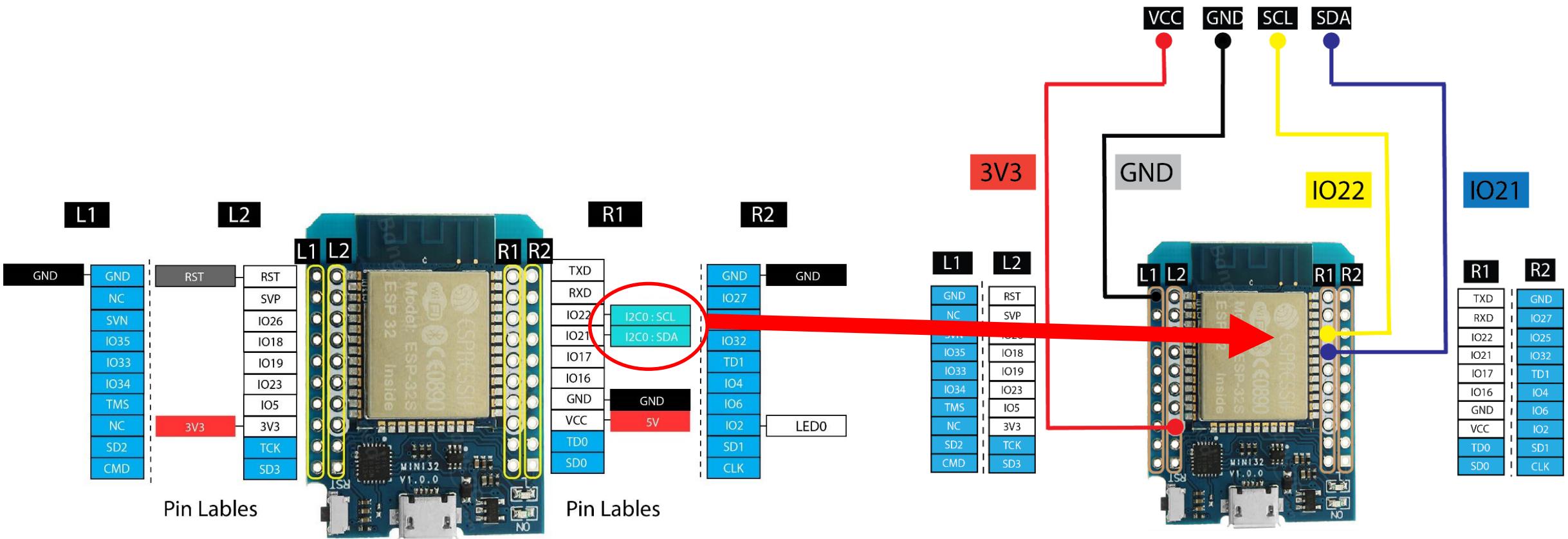
3.3 V สำหรับ ด้านล่าง



จัดวางอุปกรณ์ดังรูป



ເຮືອມຕ່ອສໍາລັບລາຍ



The screenshot shows the uPyCraft V1.1 IDE interface. The menu bar includes File, Edit, Tools, Help, and a toolbar with icons for Serial, Download, Stop, BurnFirmware, InitConfig, and Preferences. A red box highlights the 'Download' option in the Tools menu. The central workspace displays the code for the ssd1306.py file, which is described as the "D1306 OLED driver, I2C and SPI interface". The code lists various command definitions, such as SET_CONTRAST, SET_ENTIRE_ON, SET_NORM_INV, SET_DISP, SET_MEM_ADDR, SET_COL_ADDR, SET_PAGE_ADDR, SET_DISP_START_LINE, SET_SEG_REMAP, SET_MUX_RATIO, and SET_COM_OUT_DIR, each assigned a specific constant value. Below the code, the terminal window shows the message "Ready to download this file, please wait!" followed by "download ok".

```
ssd1306.py x
D1306 OLED driver, I2C and SPI interface

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()

```

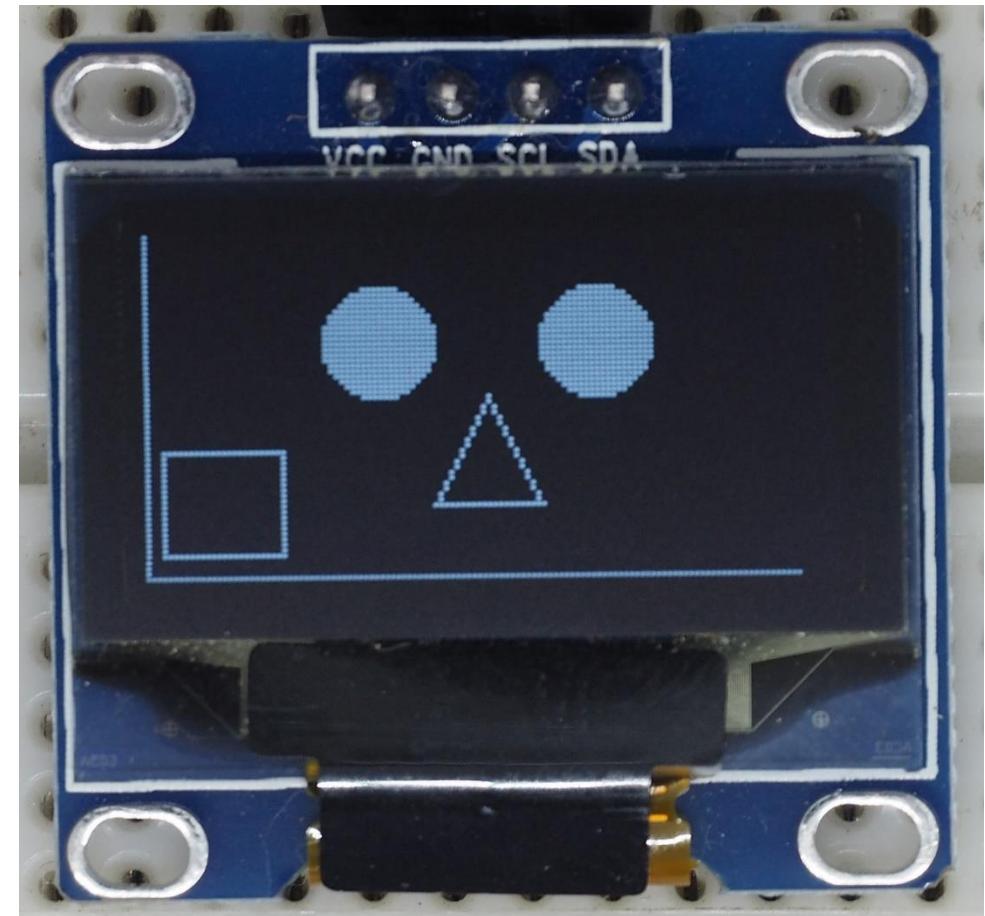


ฟังก์ชัน	รายละเอียด
fill(x)	$x = 0$ ลบหน้าจอทั้งหมด $x = 1$ เขียนหน้าจอทั้งหมด
contrast(x)	ปรับความสว่างหน้าจอ ค่า x ต่ำจะให้ความสว่างที่ต่ำ
poweroff	ปิดหน้าจอ

ผลที่ควรได้

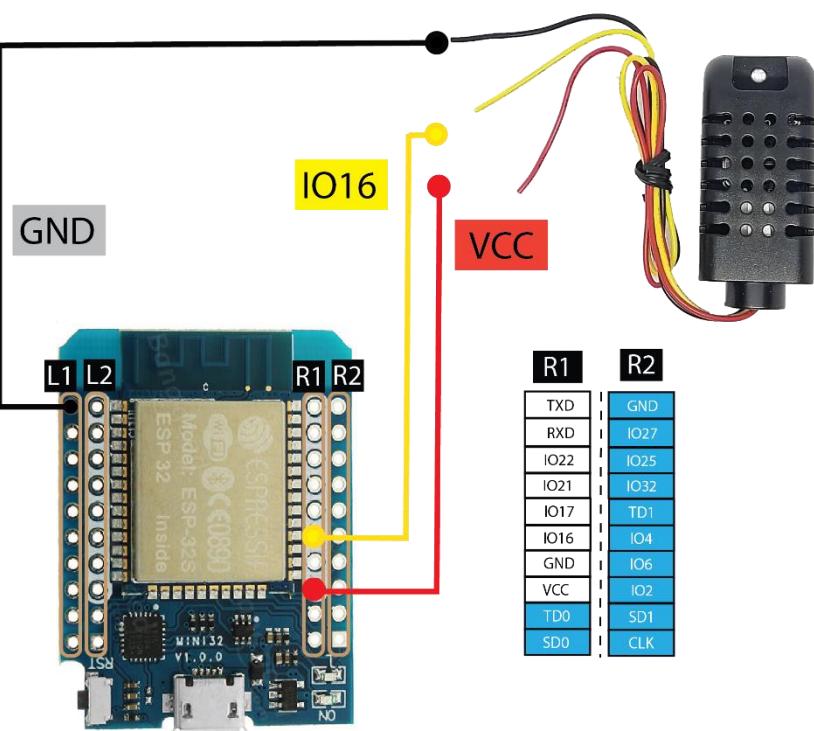
ໂນໂລຢີເພີ່ມເຕີມ gfx ສໍາຮັບວາດຮູ້ປາກ ບ. Adufruit

```
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)
```

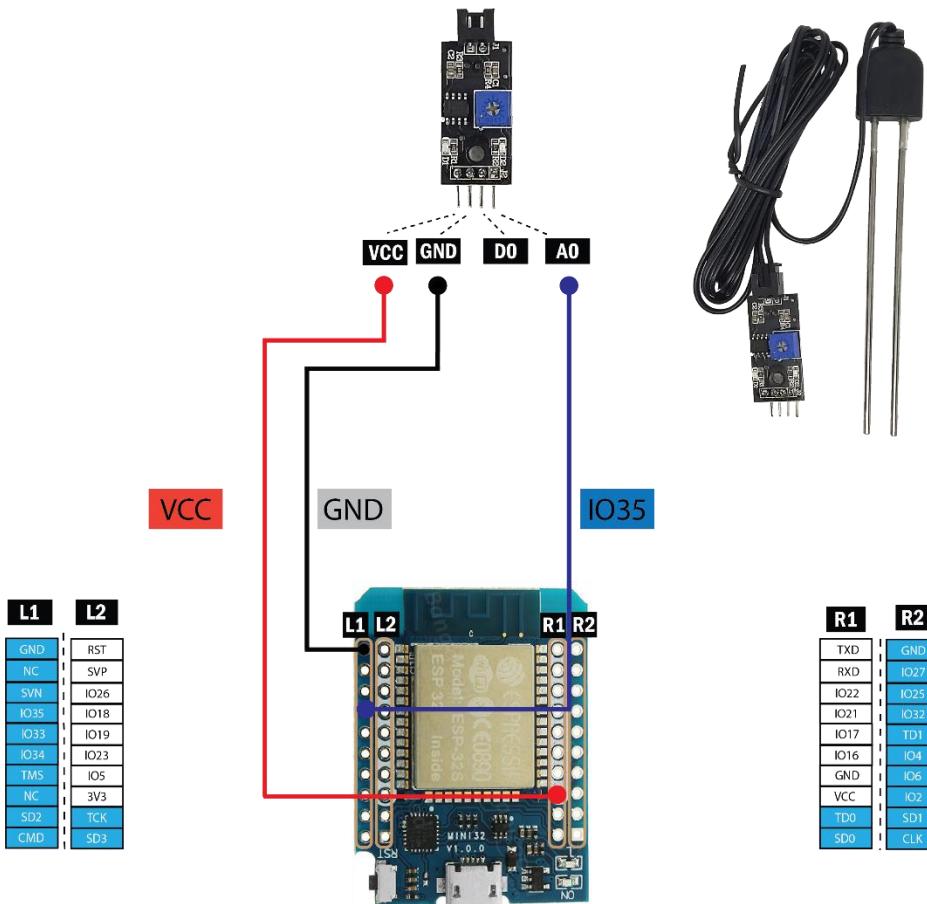
ต่อสายไม่ดี อ่านไม่ได้ครับ

```
>>> 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() |
```

ทดสอบอ่านค่าอනalog จากเซ็นเซอร์วัดความชื้นดิน



```
1 from machine import ADC, Pin  
2  
3 adc35 = ADC(Pin(35))  
4 adc35.read()  
5  
6 adc35.atten(ADC.ATTN_11DB)  
7 adc35.width(ADC.WIDTH_12BIT)  
8 adc35.read()  
9
```

การลดทอน สัญญาณ

- ADC.ATTN_0DB: 0dB รองรับแรงดันสูงสุด 1.0 ไวโอล์(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)



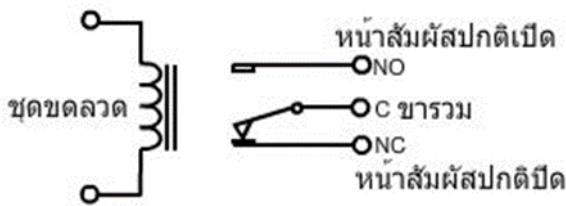
รีเลย์ (Relay)	โซลิดสเตตอรีเลย์ (Solid State Relay)
อายุการใช้งานสั้น เนื่องจากหน้าสัมผัสแบบแมคคานิค	มีอายุการใช้งานนาน
มีเสียงระหว่างการตัดต่อ	ไม่มีเสียงเวลาตัดต่อ
อาจเกิดสัญญาณรบกวน เนื่องจากการตัดต่อ	ไม่เกิดการรบกวน
สามารถตรวจสอบได้ง่าย	เกิดความร้อนการใช้งานเป็นเวลานาน ควรมีการใช้ชีสซิงค์เพื่อบาധความร้อน

ส่วนประกอบของรีเลย์ทั่วไป

1. ขดลวด (Coil)



2. หน้าสัมผัส (Contact)

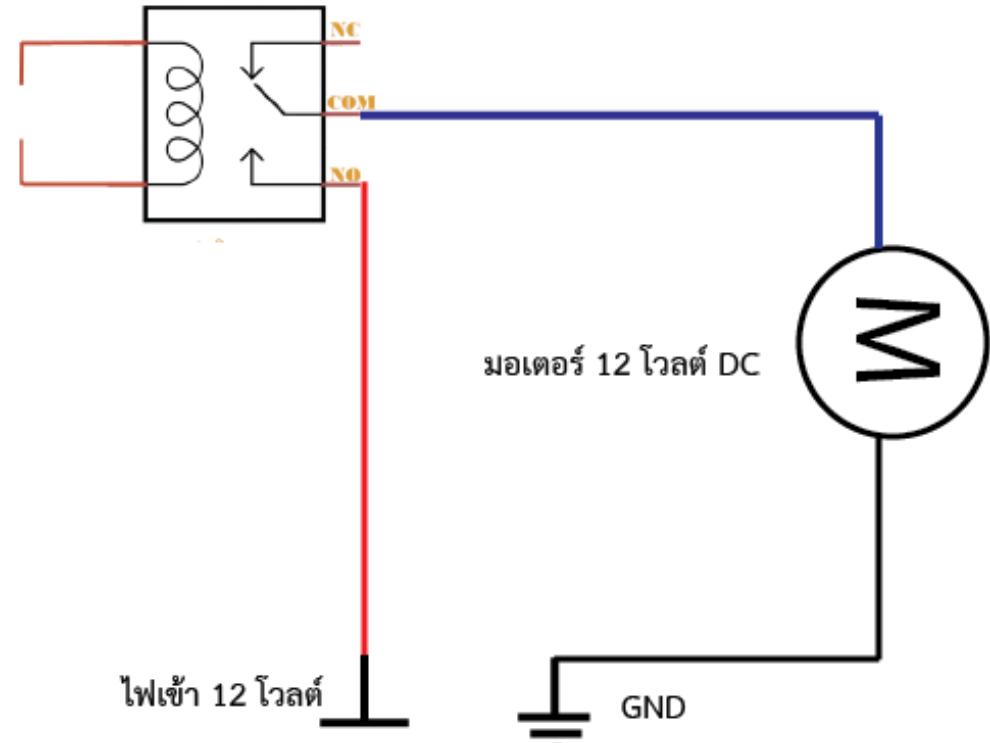


- จุดต่อ NC : จุดต่อ NC ย่อมาจาก normal close หมายความว่า ปกติปิดหรือหากยังไม่จ่ายไฟให้ขดลวดเห็นี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา
- จุดต่อ NO : จุดต่อ NO ย่อมาจาก normal open หมายความว่า ปกติเปิดหรือหากยังไม่จ่ายไฟให้ขดลวดเห็นี่ยวนำหน้าสัมผัสจะไม่ติดกัน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

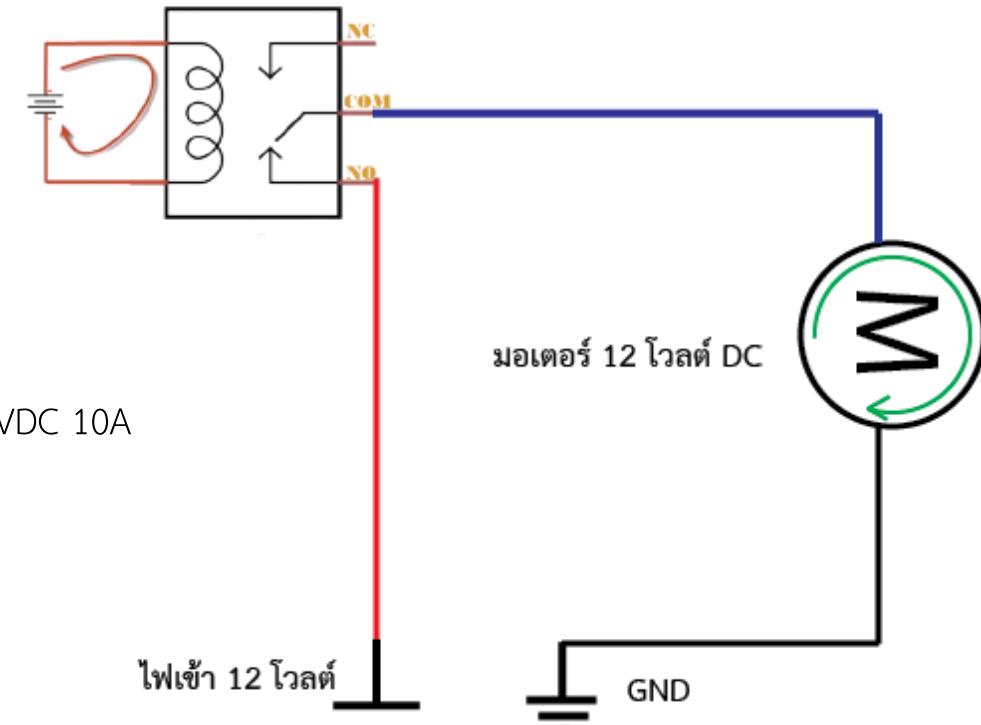
ข้อคำถึงในการ เลือกซื้อหรือใช้ งานรีเลย์ทั่วไป

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

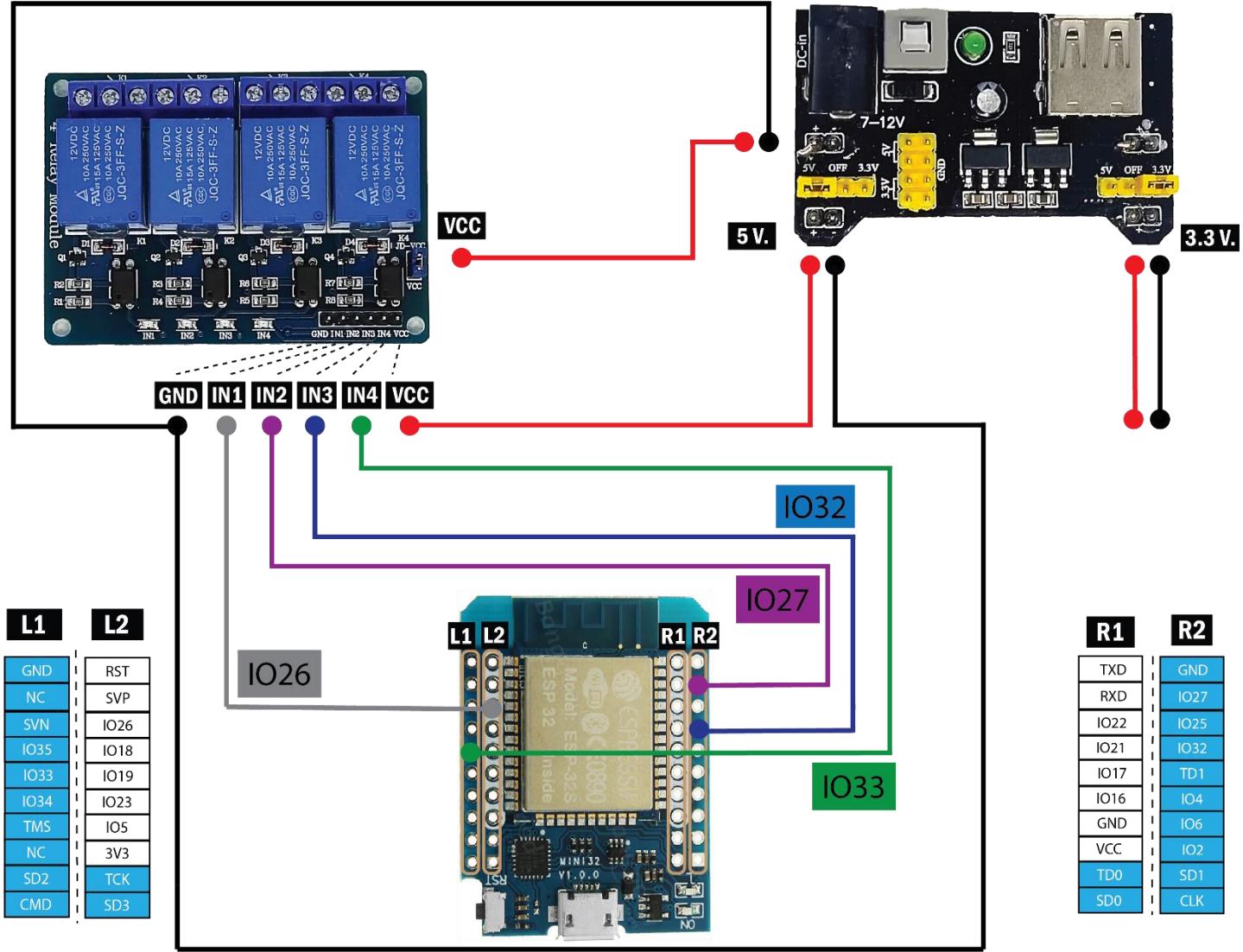
รีเลย์บันบอร์ด



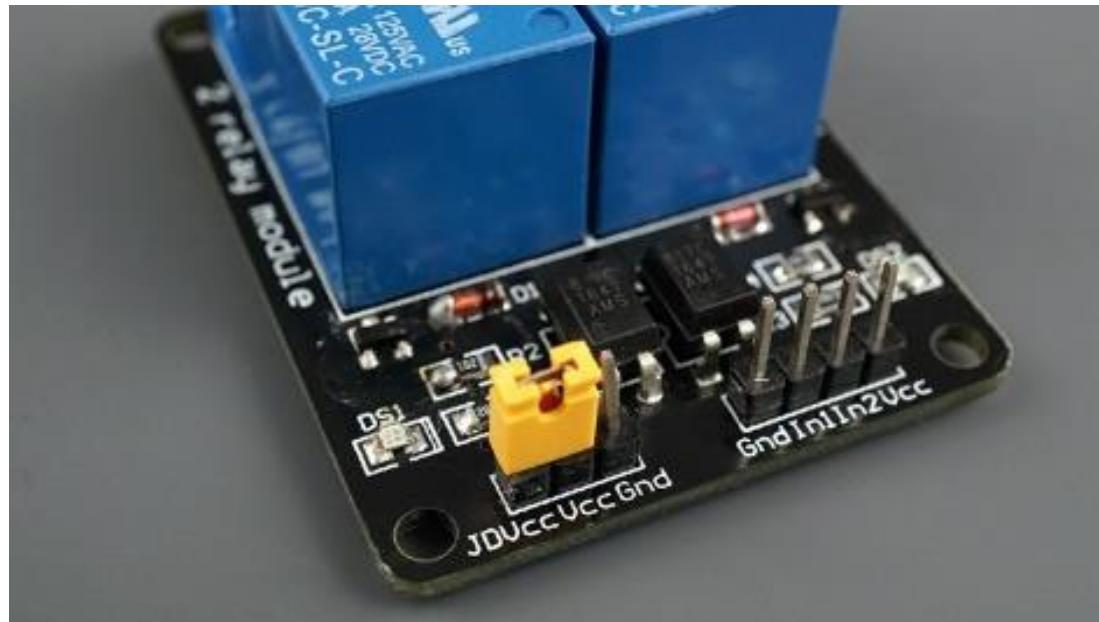
- กระแสเตอร์ขนาด 12 โวลต์
- แบบ SPDT
- หน้าสัมผัสขนาด 120VAC/24VDC 10A
หรือ 250VDC 10A/6A



การเชื่อมต่อรีเลย์

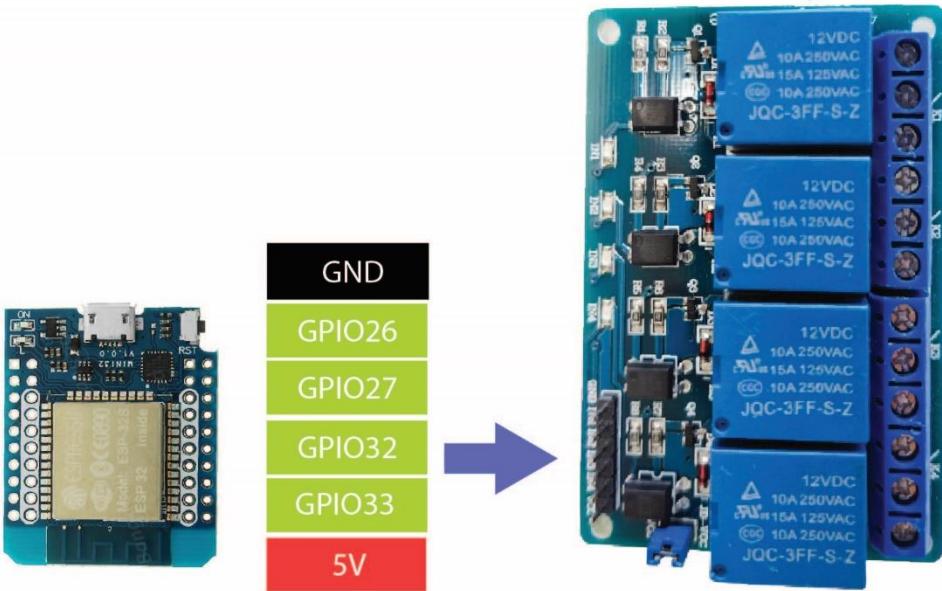


การจ่ายไฟให้กับรีเลย์



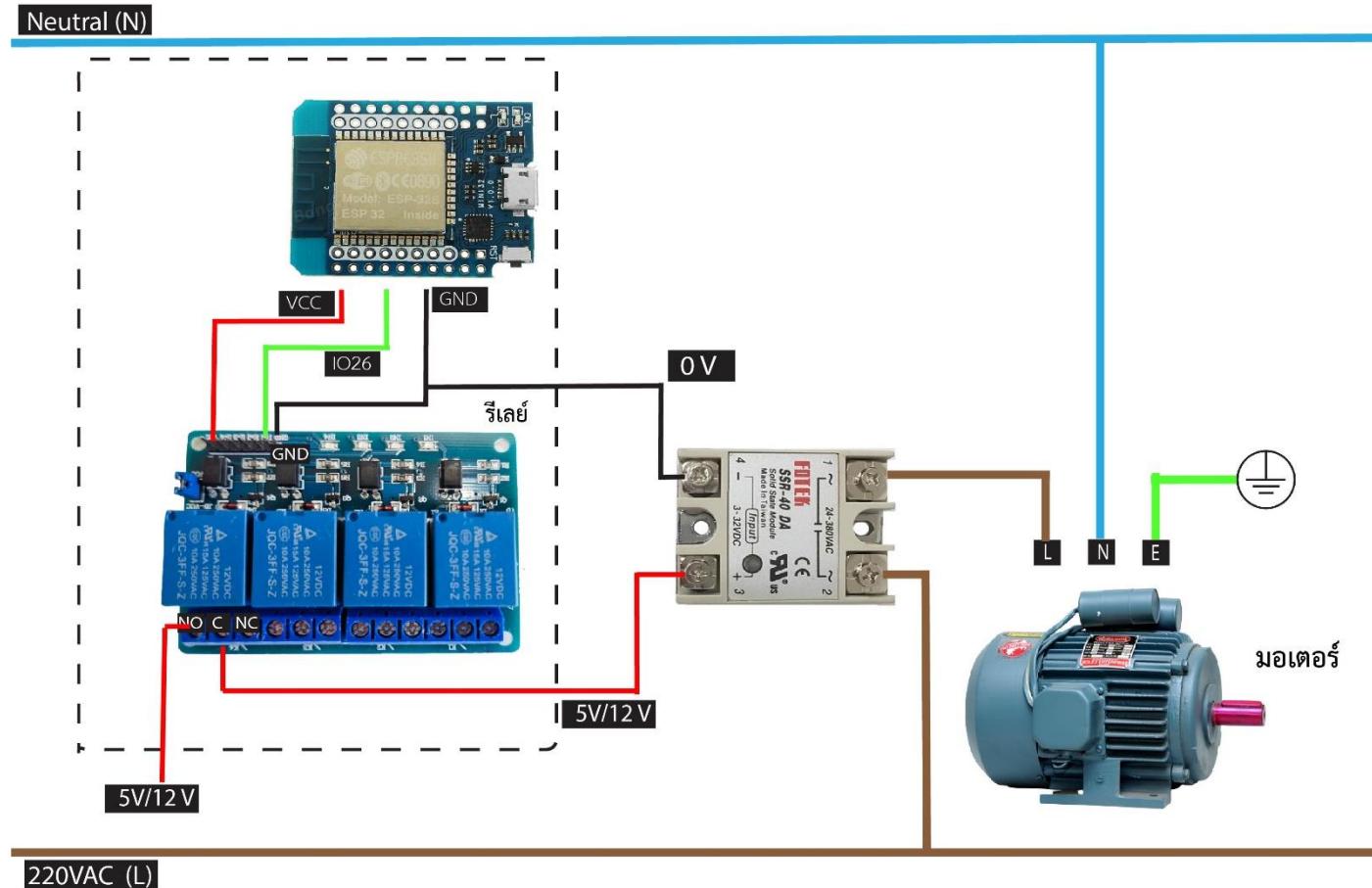
ข	หน้าที่	หมายเหตุ
1	GND	เชื่อมต่อกราวน์ด์ร่วมกับวงจรภายใน
2	VCC	ขาไฟเลี้ยงบนบอร์ดวงจร ร่วมไฟขาเข้า
3	JD-VCC	ขาไฟเลี้ยงเพื่อขับรีเลย์

ทดลองบอร์ด 4 ช่องสัญญาณ

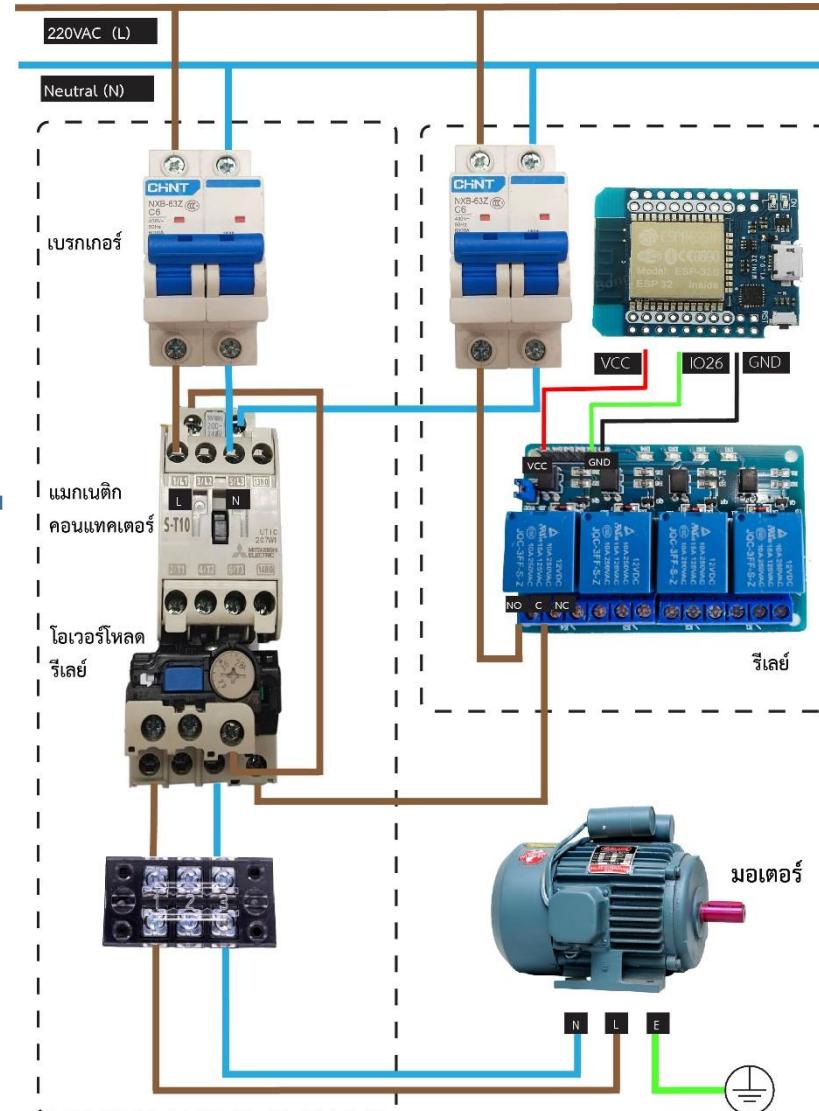
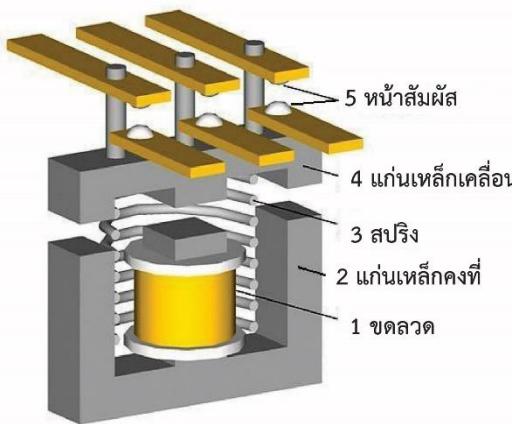


```
from machine import Pin  
relay1 = Pin(26, Pin.OUT)  
relay2 = Pin(27, Pin.OUT)  
relay3 = Pin(32, Pin.OUT)  
relay4 = Pin(33, Pin.OUT)  
relay1.value(0) # on  
relay1.value(1) # off
```

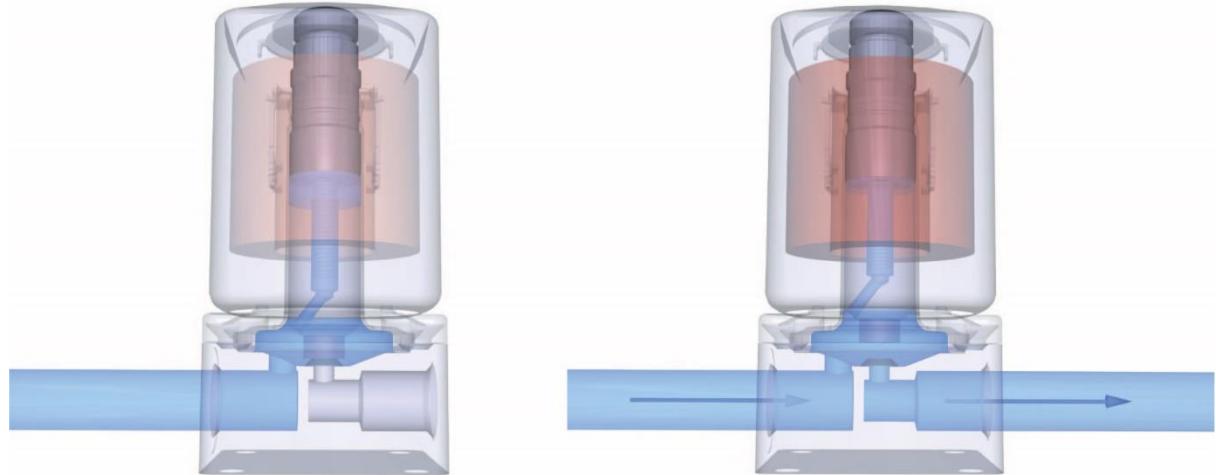
การใช้งาน Solid State Relay เพื่อขับมอเตอร์



การใช้งาน Magnetic Relay และ Overload เพื่อขับมอเตอร์



ໂჟლິນອຍດ້ວລ້າ



ສភາວະປກຕີ ແລະ ພັງຈາກສັ່ງທຳງານ

