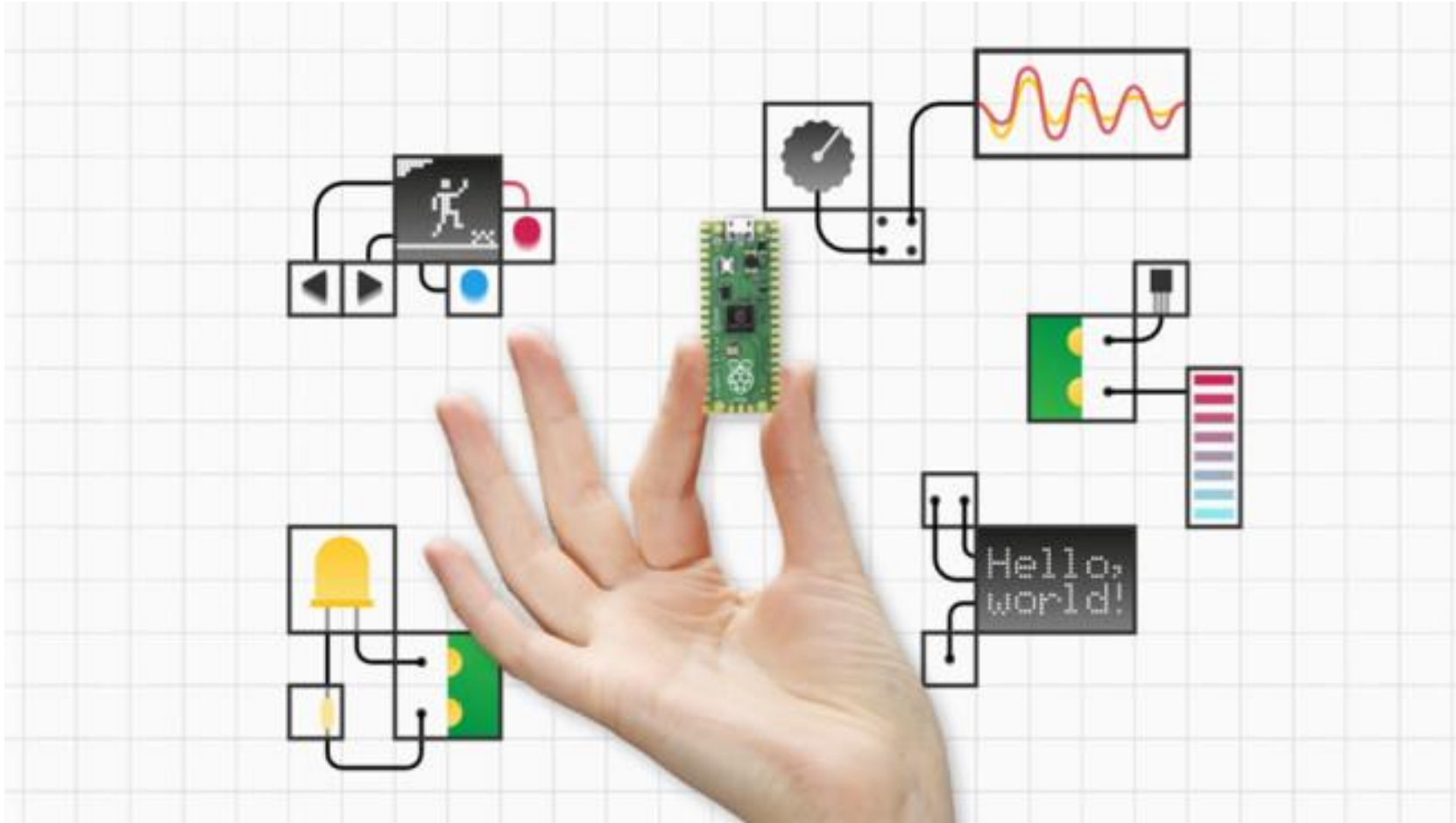


# การเขียนโปรแกรม Raspberrypi Pico

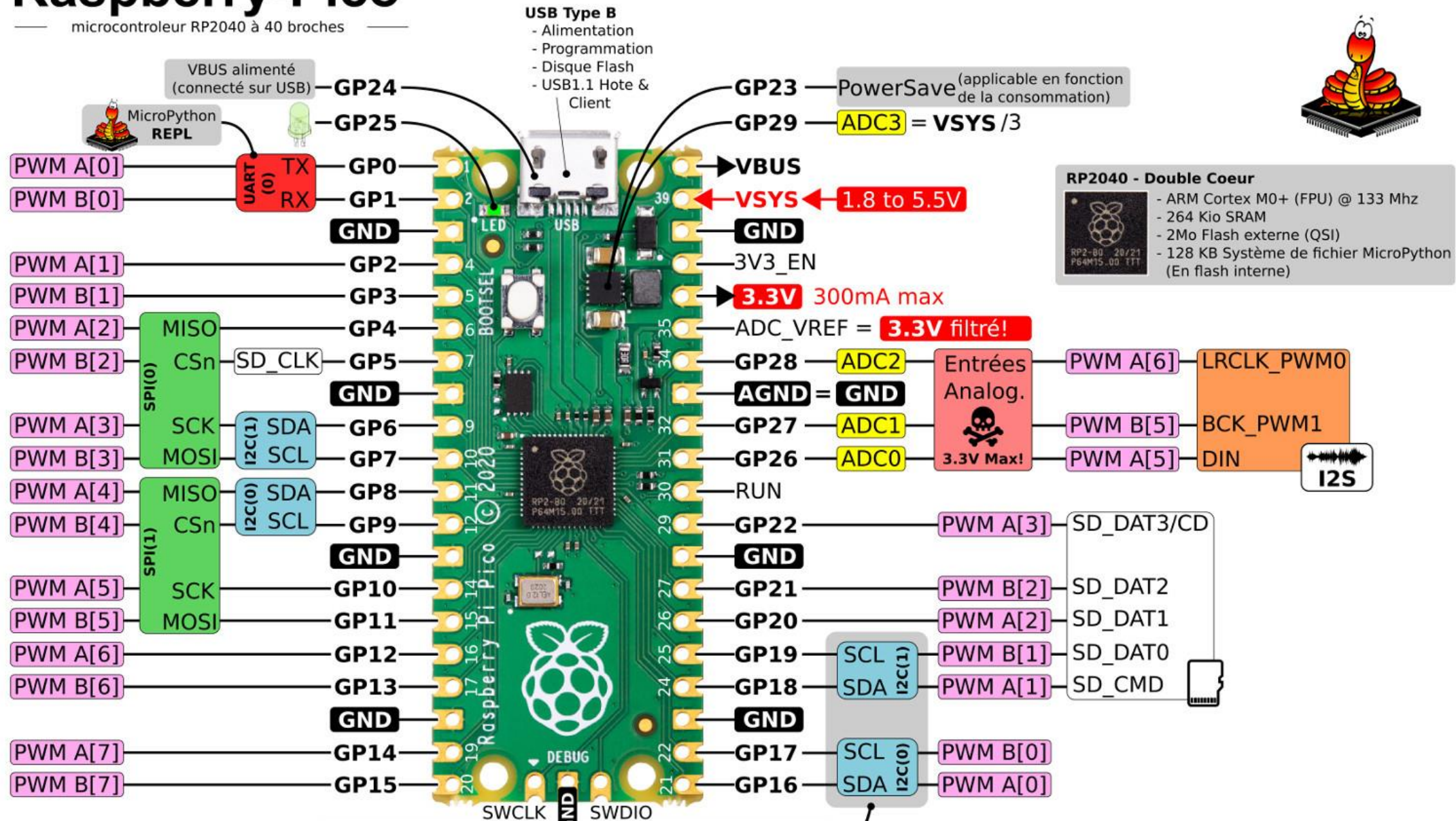
อ.เห่ง + อ.หนึ่ง

# What is Raspberry Pi Pico ?



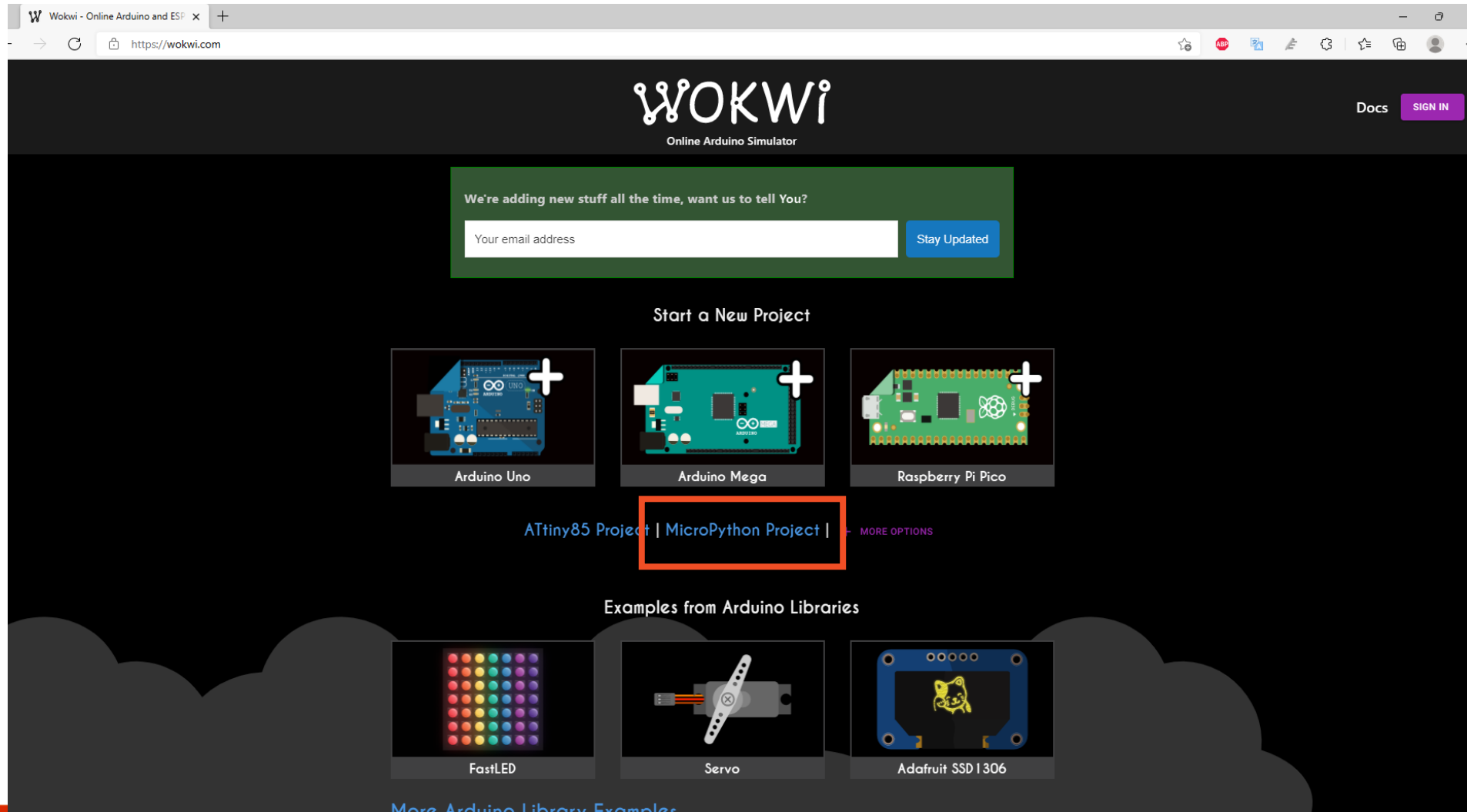
# Raspberry-Pico

microcontrôleur RP2040 à 40 broches



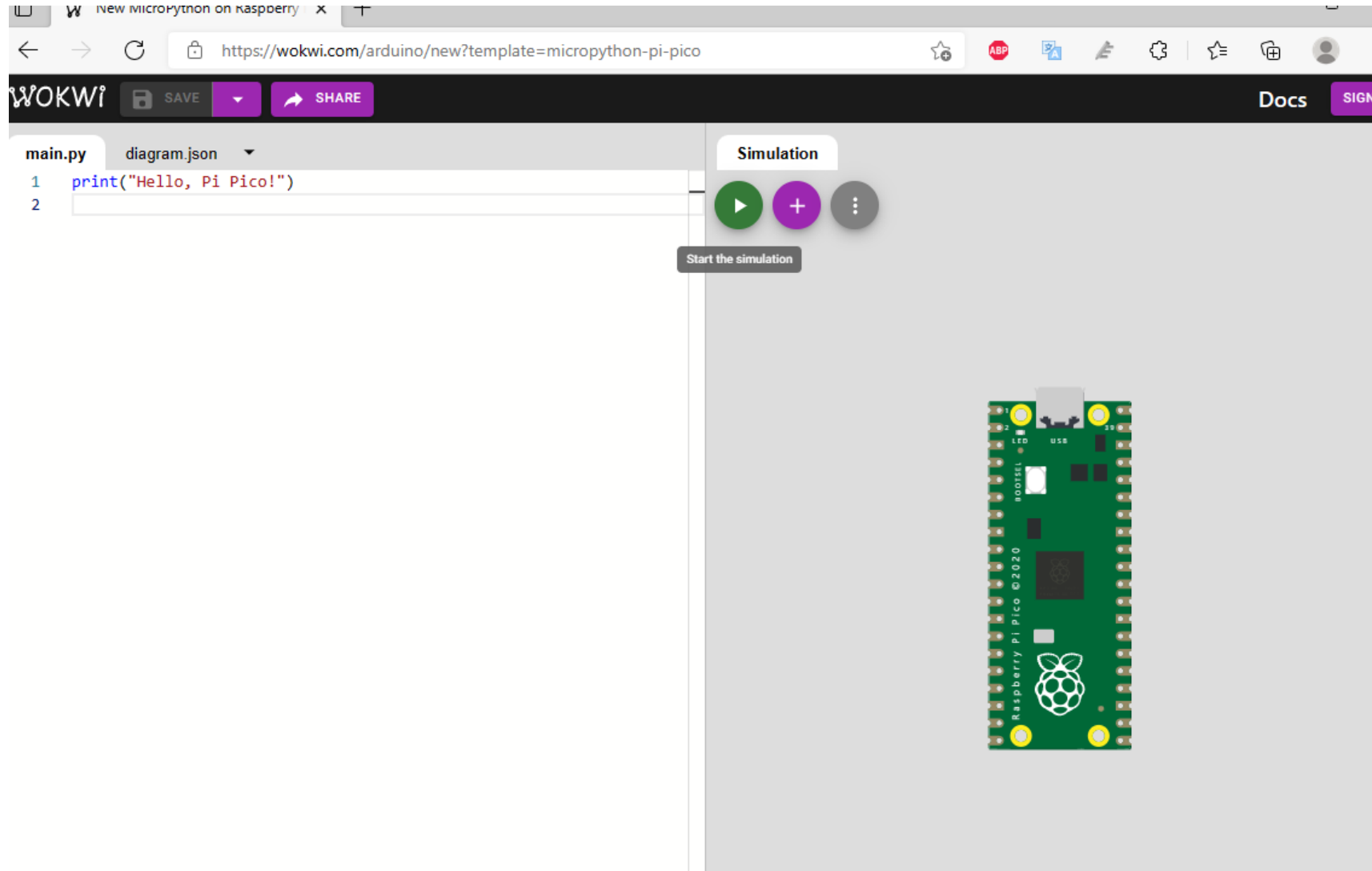
# Laboratory 1

การใช้งาน Raspberry Pi Pico Simulation ด้วย  
MicroPython บน Wokwi

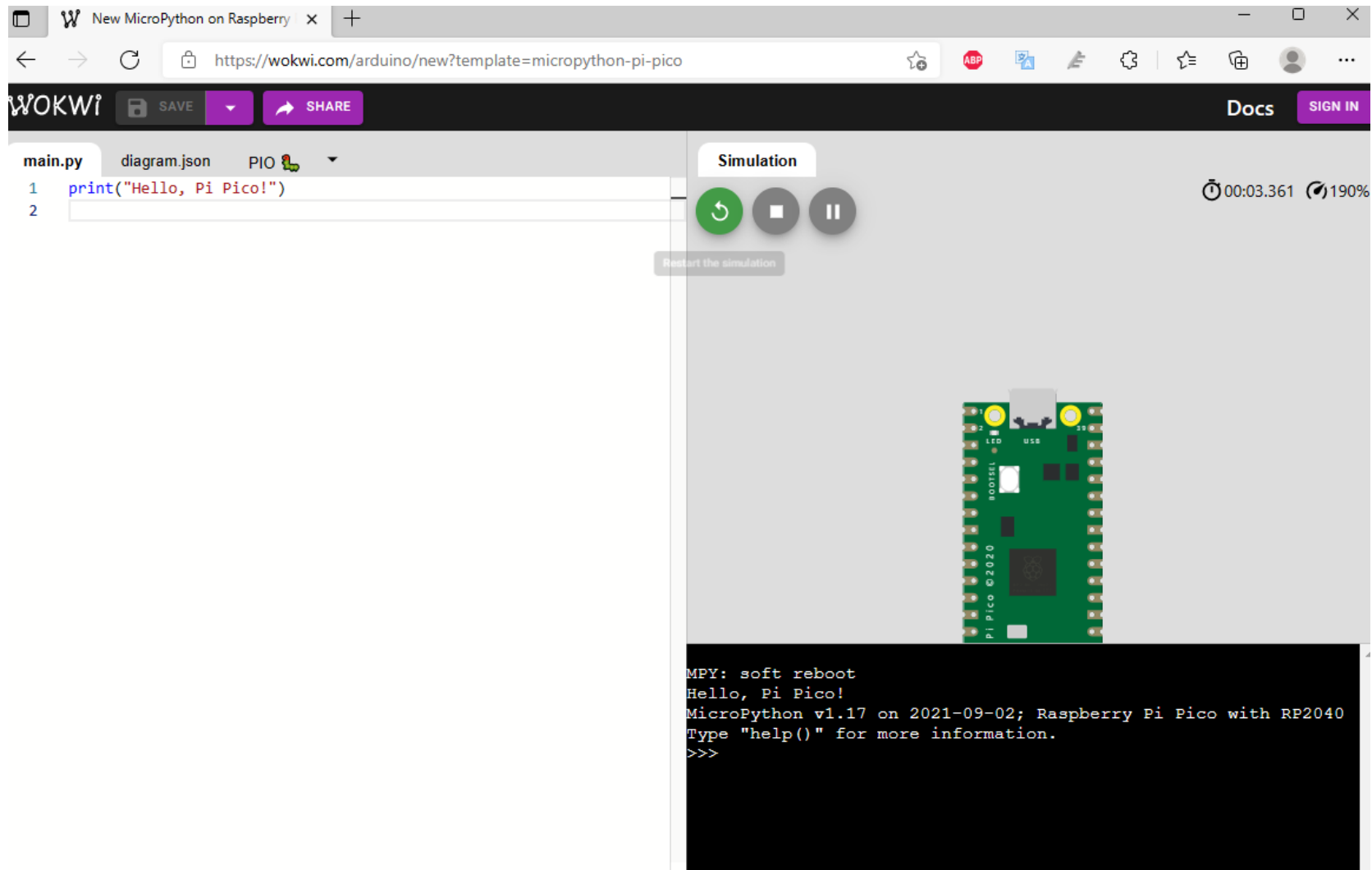




# ปรากฏหน้าต่าง Wokwi Simulator สำหรับ Pi Pico มา



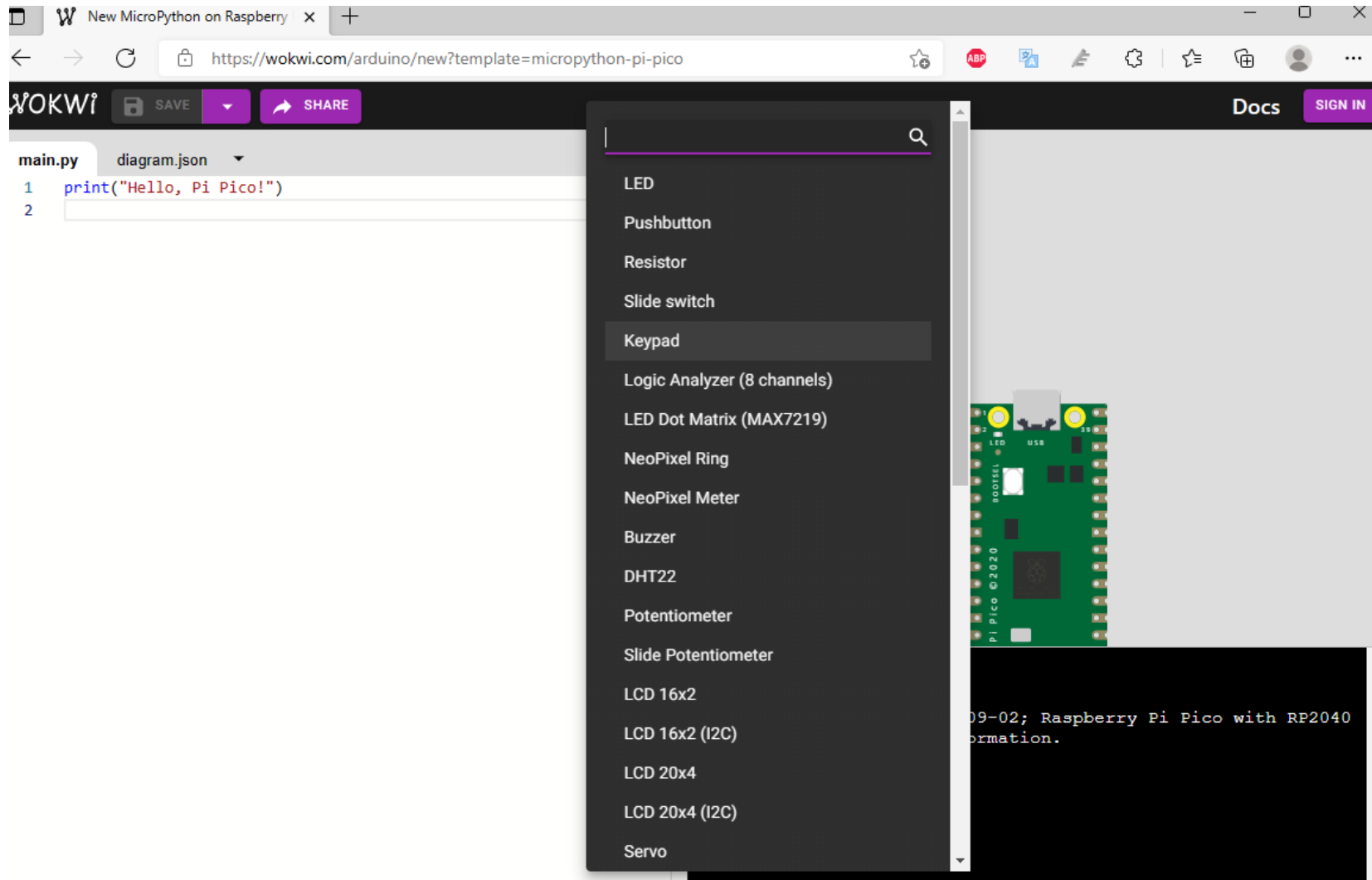
# ทดลองกดปุ่ม Play เพื่อทำการรันโค้ด

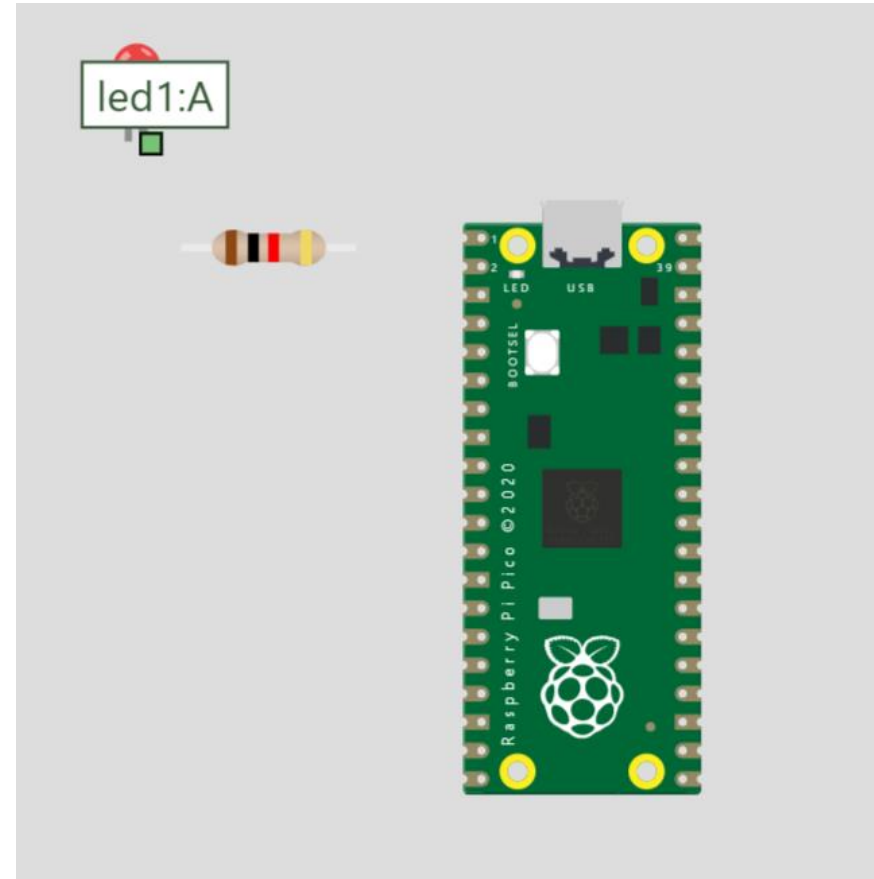
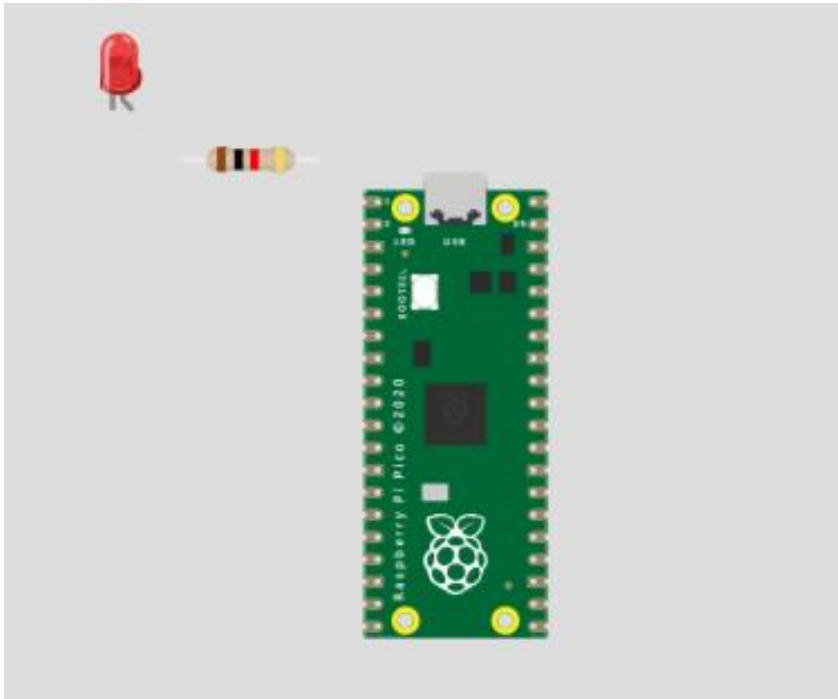


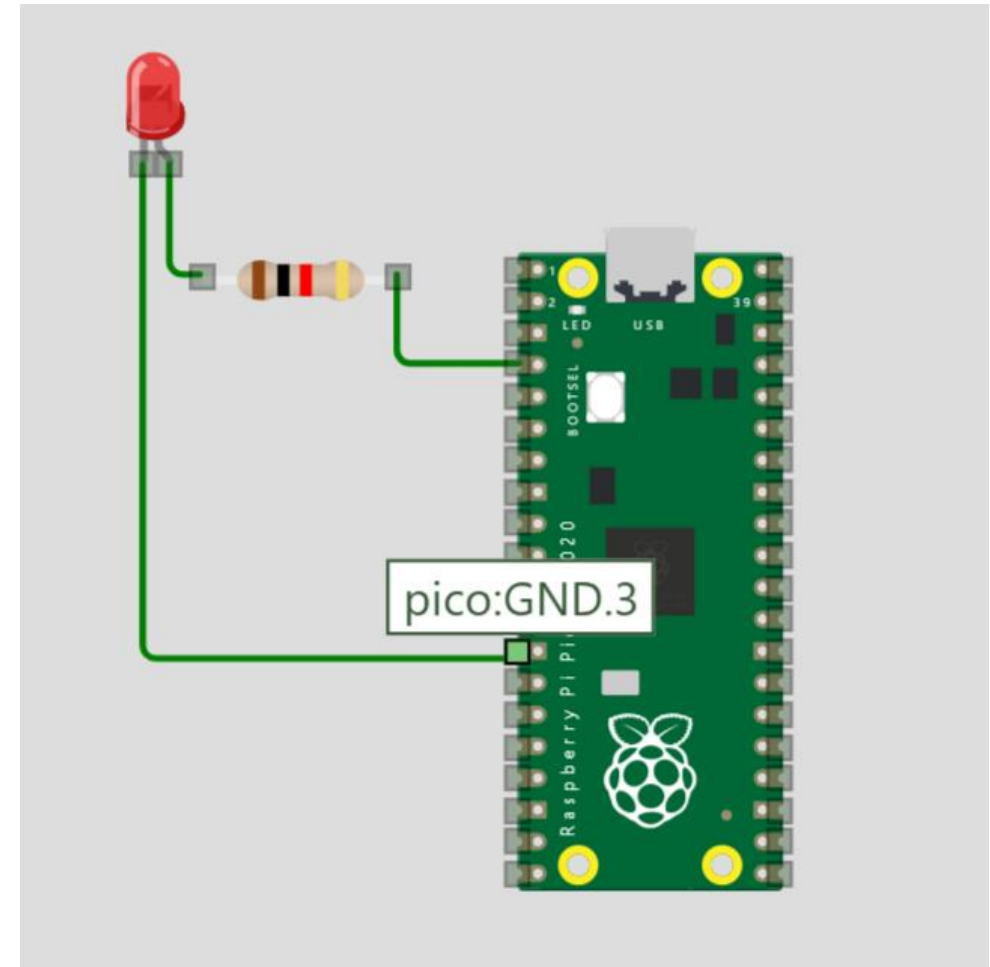
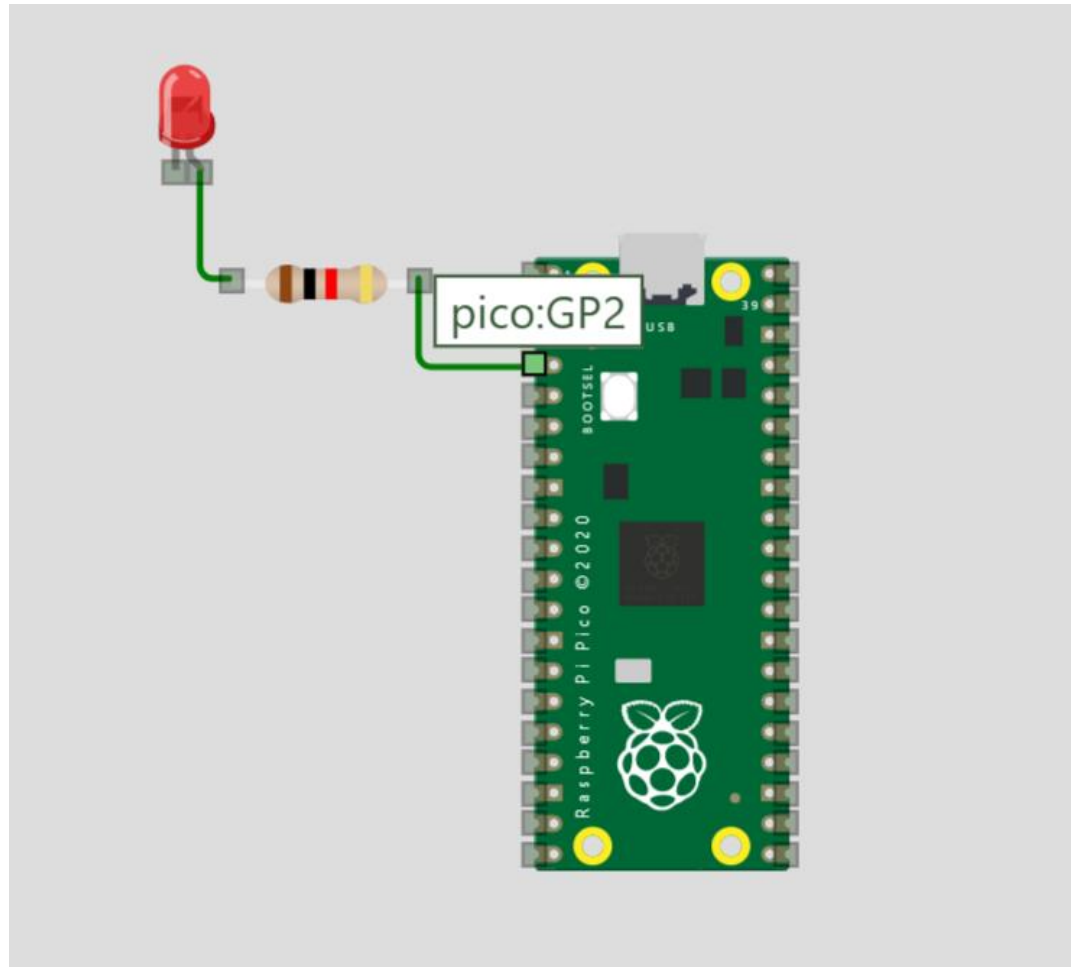
# ทดลอง Print เสน่ห์ตัวเอง



# Wokwi Simulator







# ทำการเขียนโค้ดลงไป main.py ดังนี้

```
from machine import Pin
import utime

led = Pin(2,Pin.OUT)

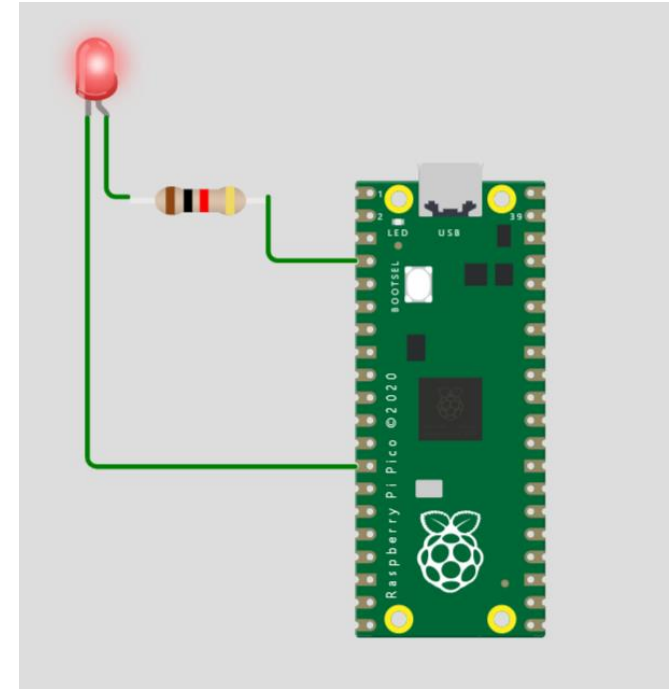
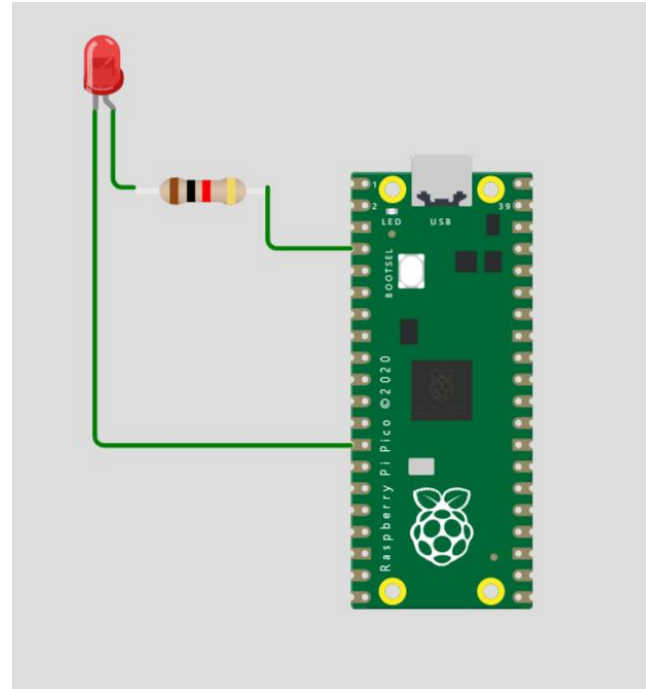
while True:

    led.on()

    utime.sleep(1)

    led.off()

    utime.sleep(1)
```



จากนั้นกดปุ่ม Play เพื่อ  
Start Simulation

หากต้องการให้กะพริบถี่ขึ้น หรือช้าลง จะต้องทำอย่างไร ?

# ทำการเขียนโค้ดลงไปที่ main.py ดังนี้

```

from machine import Pin
import utime

led = Pin(2,Pin.OUT)

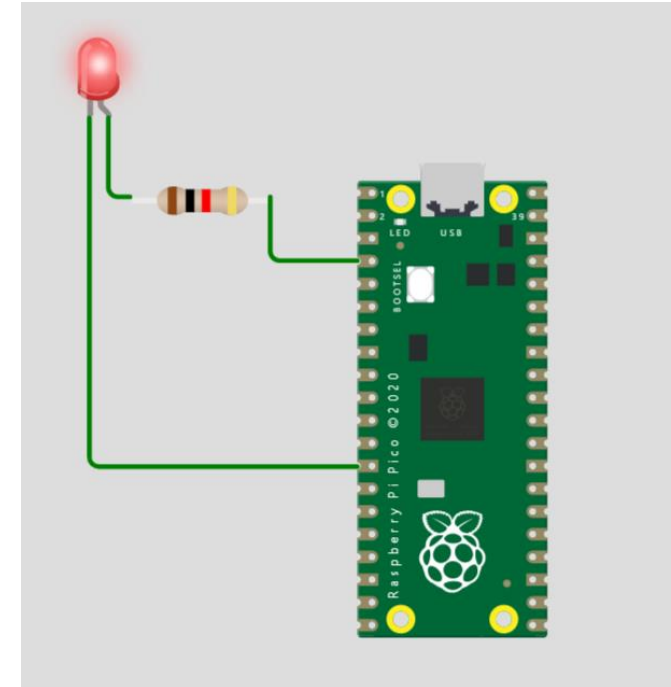
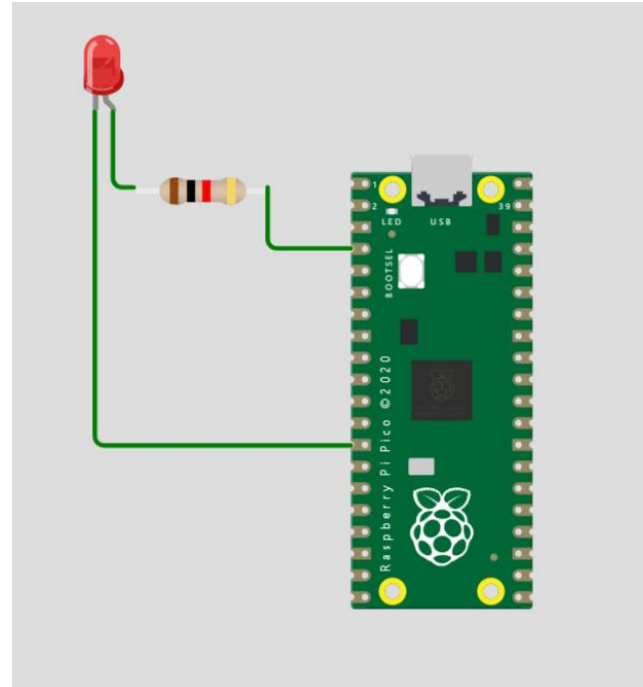
while True:

    led.on()

    utime.sleep(1)

    led.off()

    utime.sleep(1)
  
```



จากนั้นกดปุ่ม Play เพื่อ  
Start Simulation

# Laboratory 2

## การใช้งาน Sensor



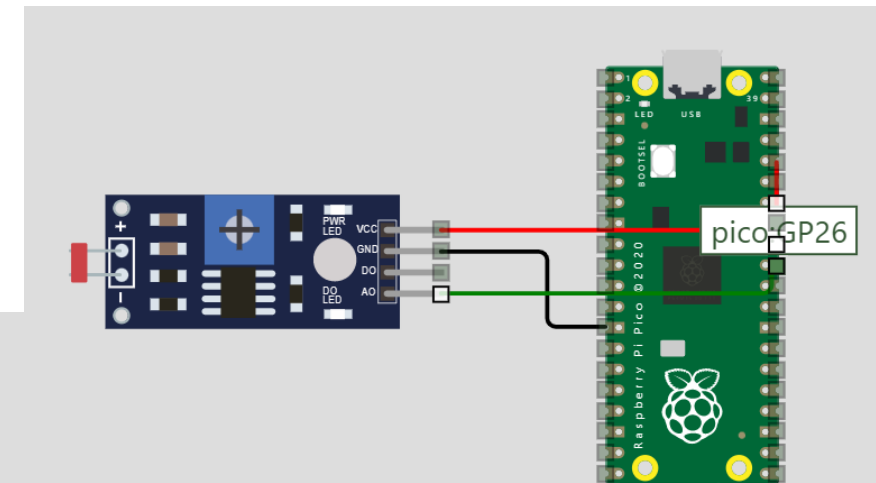
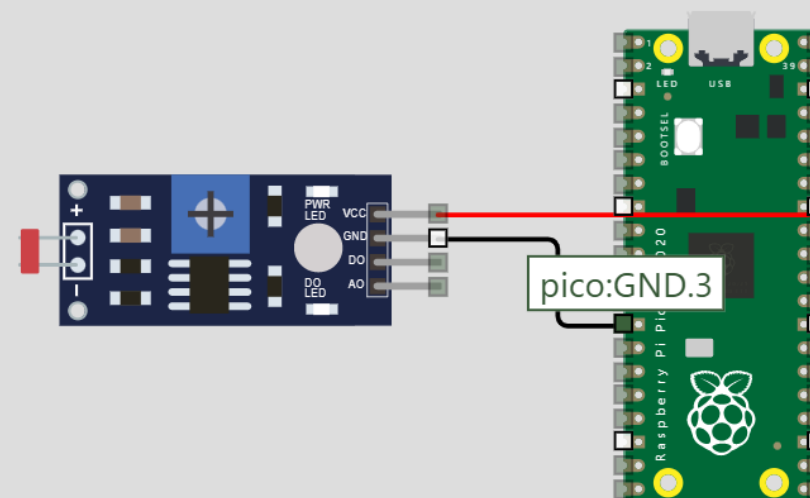
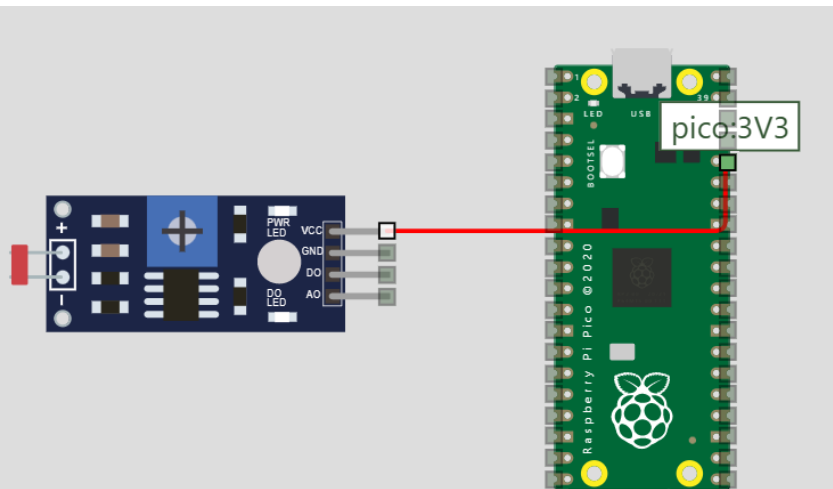
# Light Sensor (เซ็นเซอร์แสง)



74HC595 Shift Register

Photoresistor (LDR) Sensor

KY-040 Rotary Encoder



# เขียนโค้ดดังนี้

```
import machine
```

```
import utime
```

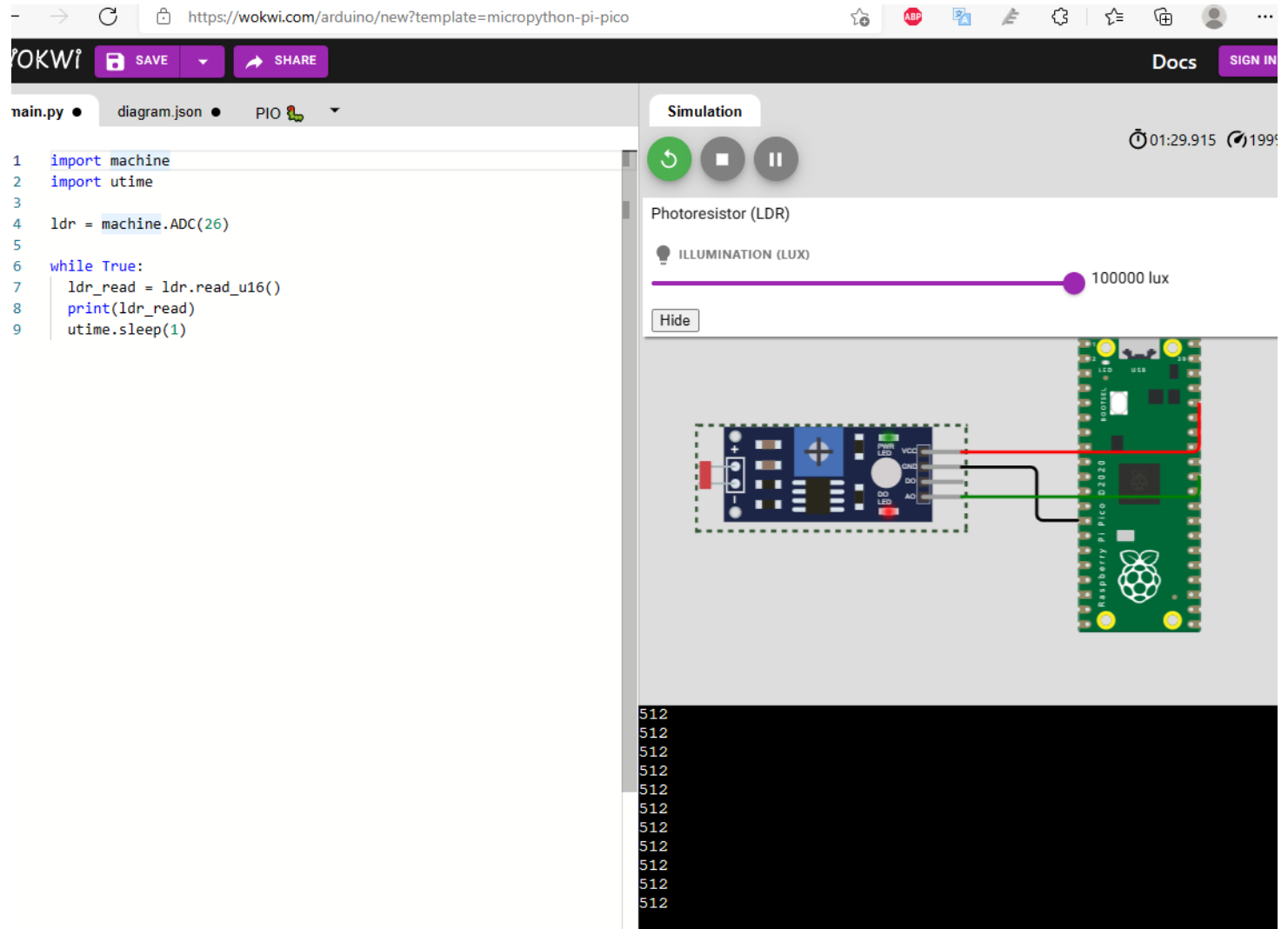
```
ldr = machine.ADC(26)
```

```
while True:
```

```
    ldr_read = ldr.read_u16()
```

```
    print(ldr_read)
```

```
    utime.sleep(1)
```

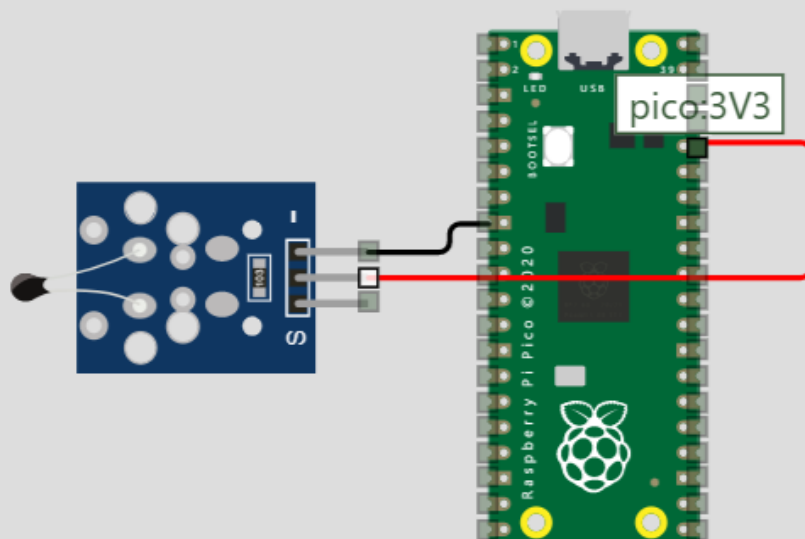
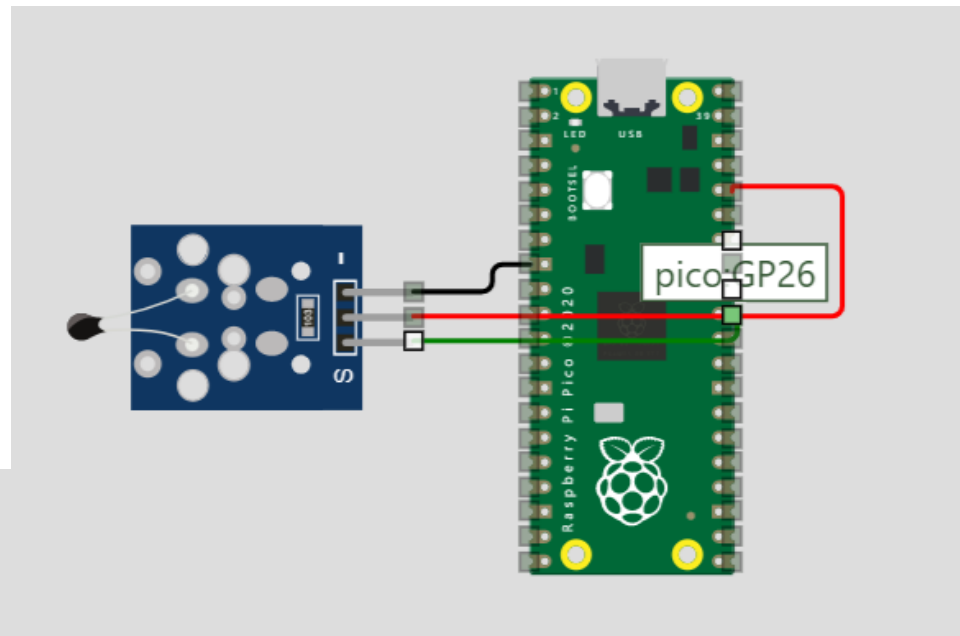
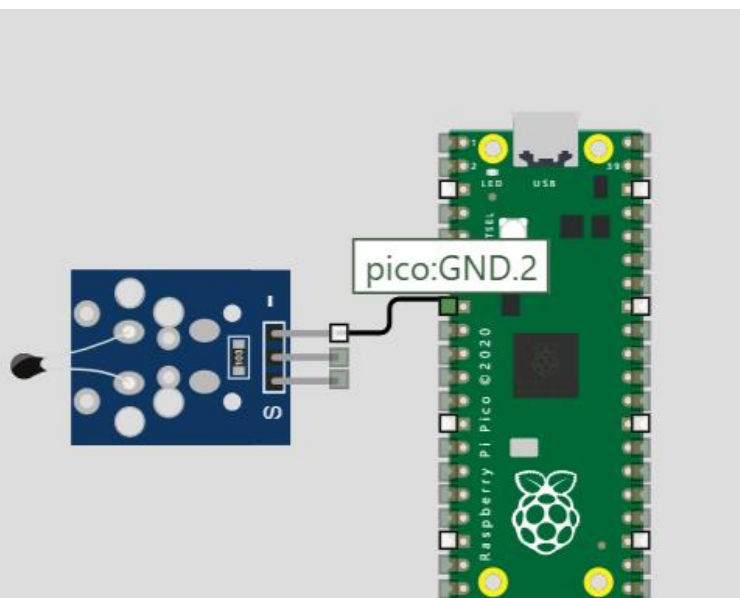


0.1 lux คือค่า 65007  
100000 lux คือค่า 512

ทำอย่างไรจึงจะแสดงผลค่าความสว่างจากการ print ที่ถูกต้อง ?

# Temperature Sensor

## Analog Temperature Sensor (NTC)



# เขียนโค้ดดังนี้

```
import machine
```

```
import utime
```

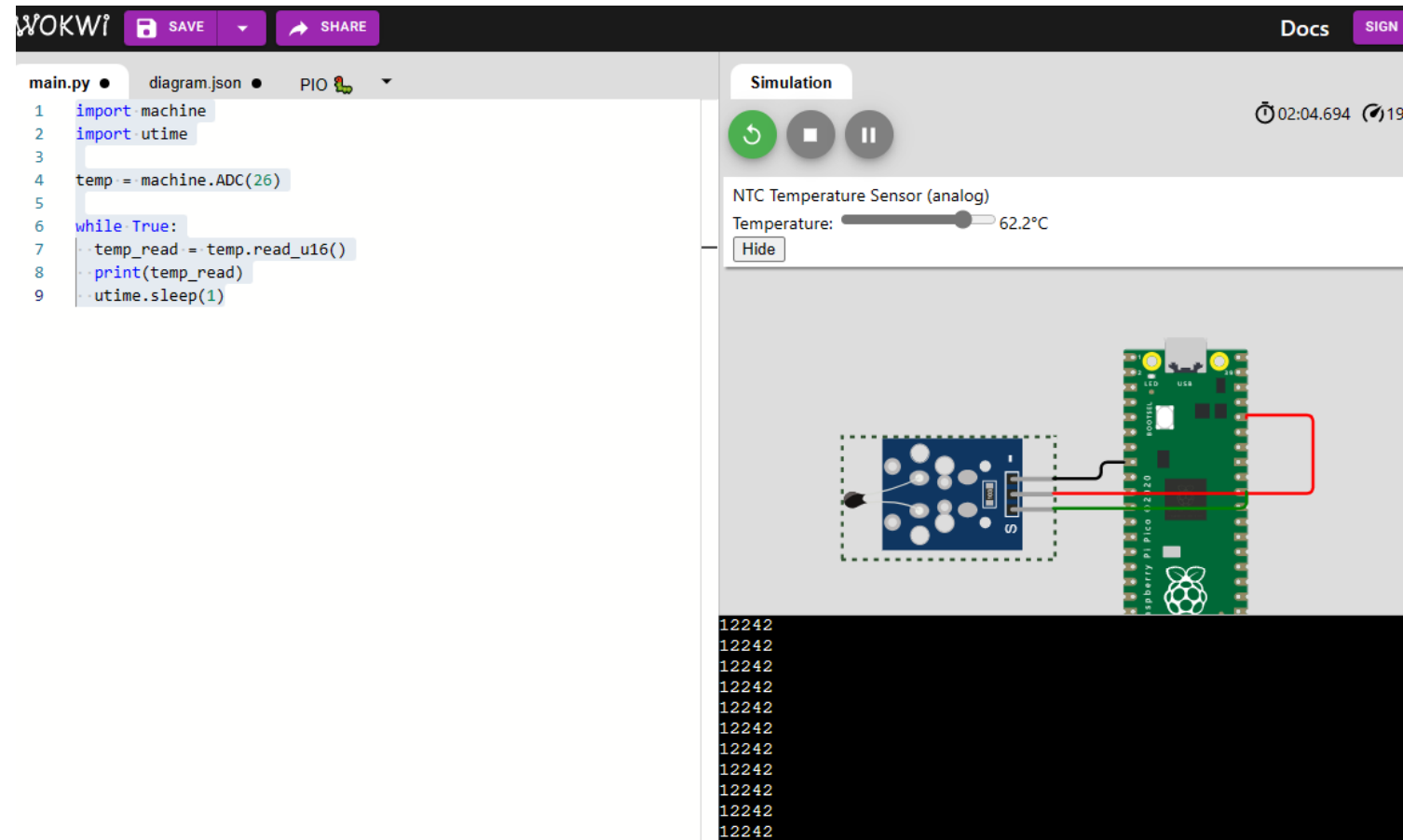
```
temp = machine.ADC(26)
```

```
while True:
```

```
    temp_read = temp.read_u16()
```

```
    print(temp_read)
```

```
    utime.sleep(1)
```



WOKWI

SAVE SHARE

Docs SIGN

main.py • diagram.json • PIO

```

1 import machine
2 import utime
3
4 temp = machine.ADC(26)
5
6 while True:
7     temp_read = temp.read_u16()
8     print(temp_read)
9     utime.sleep(1)

```

Simulation

NTC Temperature Sensor (analog)

Temperature: 62.2°C

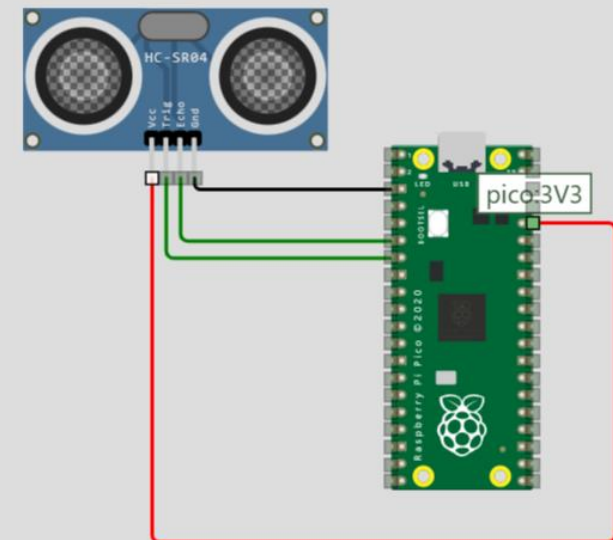
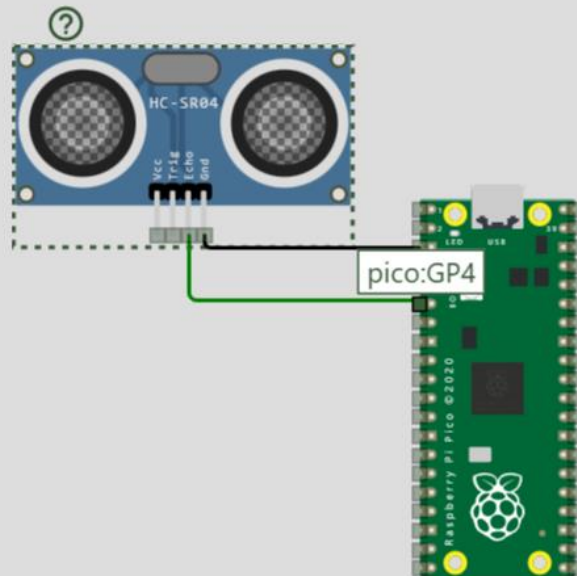
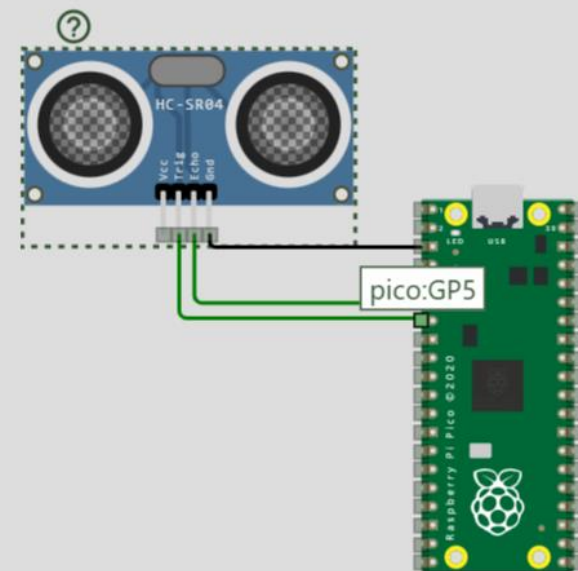
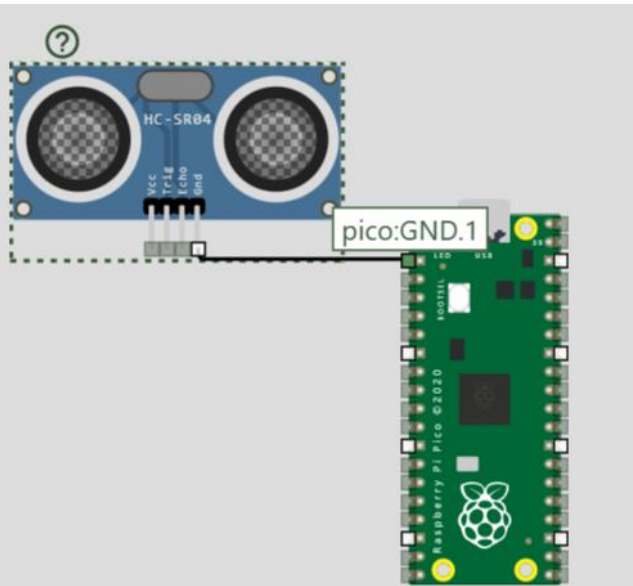
Hide

12242  
12242  
12242  
12242  
12242  
12242  
12242  
12242  
12242  
12242  
12242

ทำอย่างไรจึงจะแสดงผลค่าอุณหภูมิจากการ print ที่ถูกต้อง ?

# Ultrasonic Sensor

## HC-SR04 Ultrasonic Distance Sensor





# เขียนโค้ดดังนี้

```
import machine
```

```
import utime
```

```
while True:
```

```
    trig=machine.Pin(5, machine.Pin.OUT)
```

```
    trig.off()
```

```
    utime.sleep_us(2)
```

```
    trig.on()
```

```
    utime.sleep_us(10)
```

```
    trig.off()
```

```
    echo=machine.Pin(4, machine.Pin.IN)
```

```
    while echo.value() == 0:
```

```
        pass
```

```
    t1 = utime.ticks_us()
```

```
    while echo.value() == 1:
```

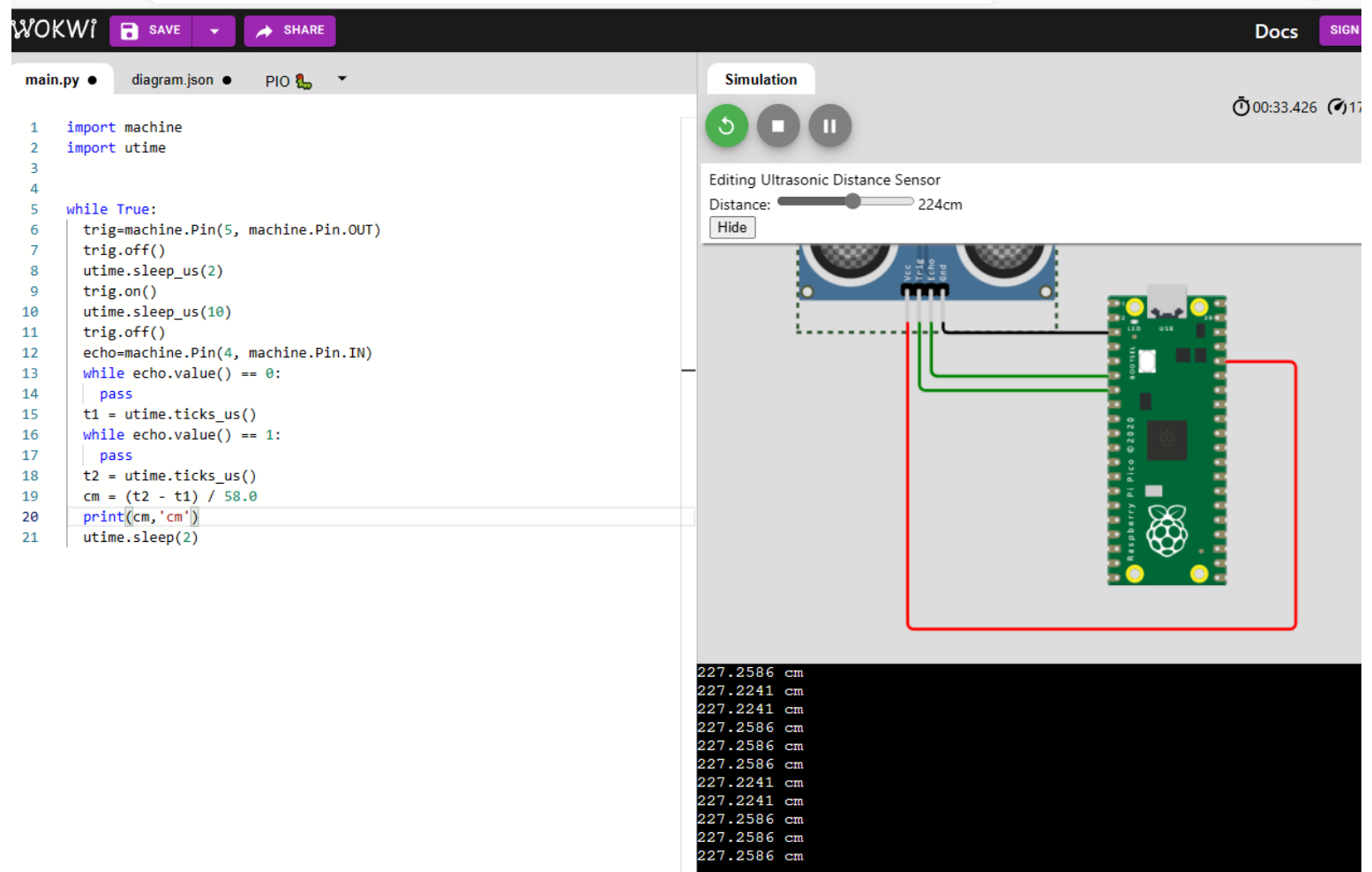
```
        pass
```

```
    t2 = utime.ticks_us()
```

```
    cm = (t2 - t1) / 58.0
```

```
    print(cm,'cm')
```

```
    utime.sleep(2)
```



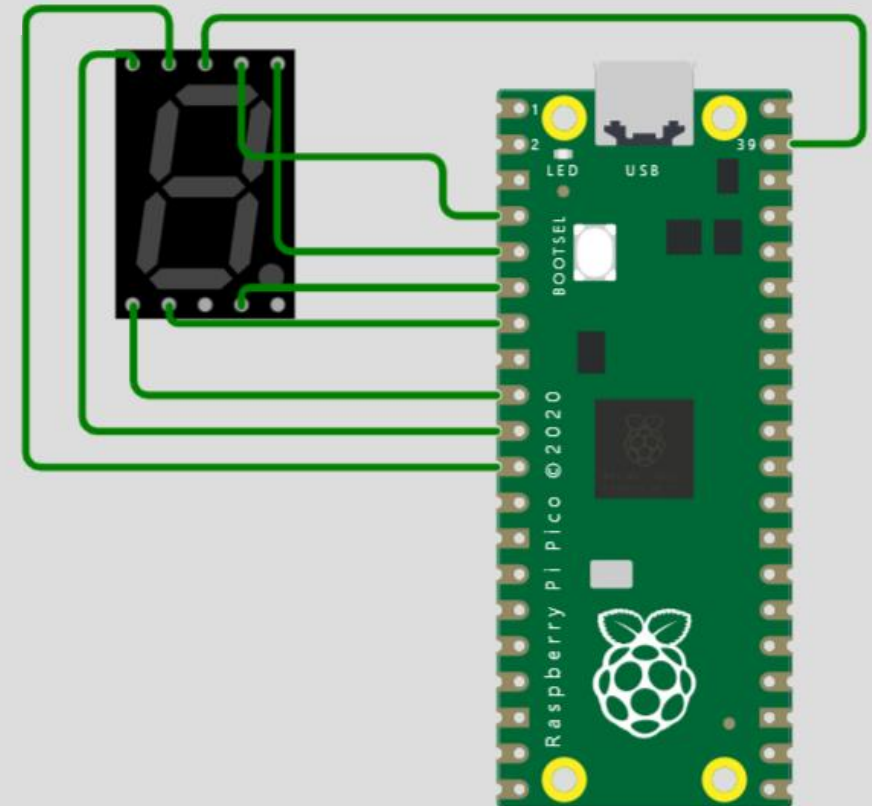
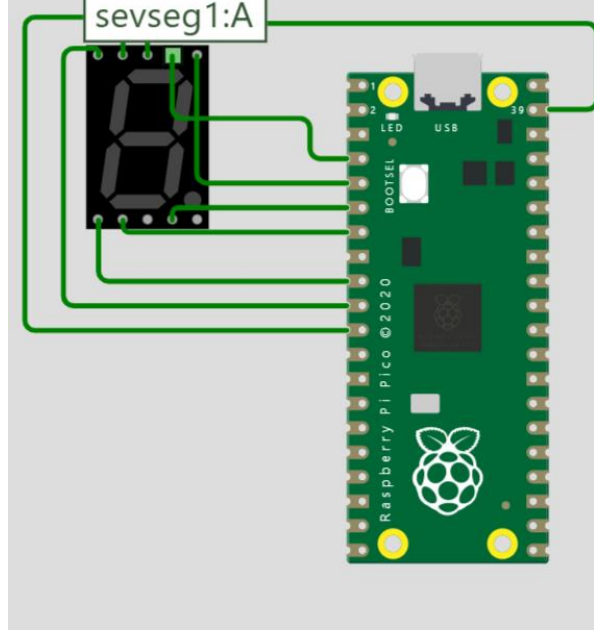
อยากให้เห็นผลเป็นหน่วยนิ้ว ต้องทำอย่างไร ?

# Laboratory 3

การแสดงผลและควบคุมอุปกรณ์

# Seven Segment

Pin(2, Pin.OUT), # A  
 Pin(3, Pin.OUT), # B  
 Pin(4, Pin.OUT), # C  
 Pin(5, Pin.OUT), # D  
 Pin(6, Pin.OUT), # E  
 Pin(8, Pin.OUT), # F  
 Pin(7, Pin.OUT), # G



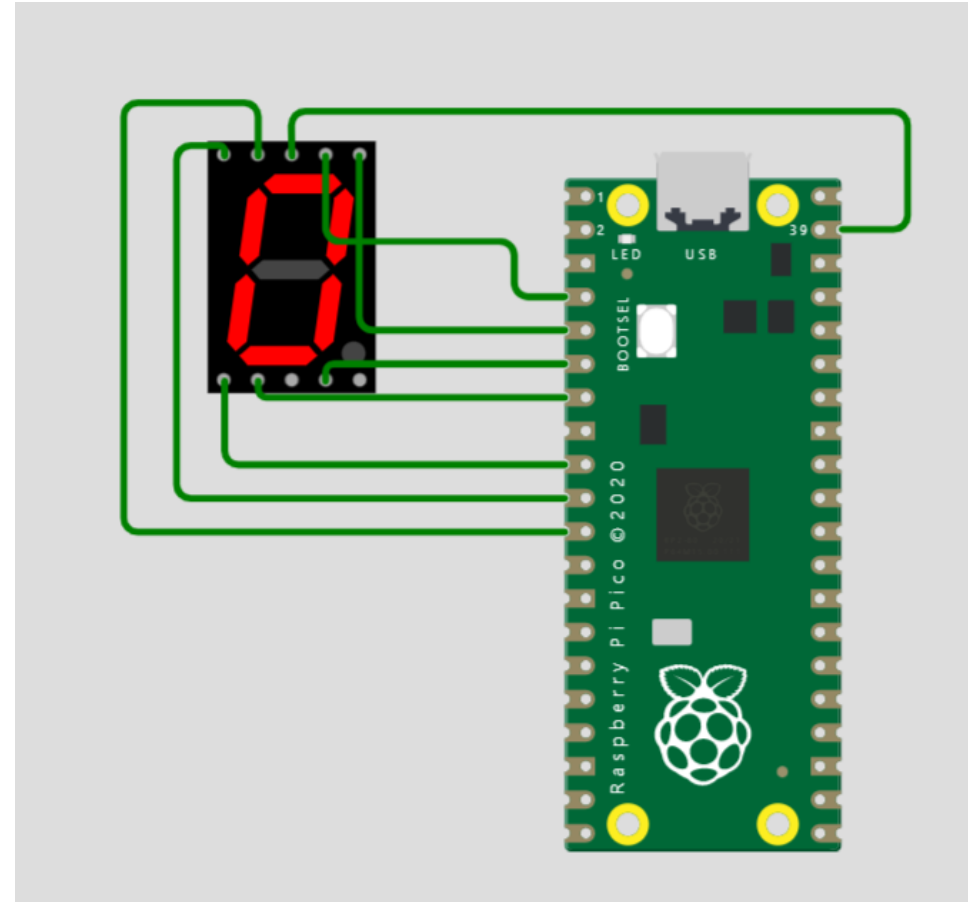
# เขียนโค้ดดังนี้

```
from machine import Pin
from utime import sleep
```

```
#      A
#      ---
#  F |  G  | B
#      ---
#  E |      | C
#      ---
#      D
```

```
A = Pin(2,Pin.OUT) # A
B = Pin(3,Pin.OUT) # B
C = Pin(4,Pin.OUT) # C
D = Pin(5,Pin.OUT) # D
E = Pin(6,Pin.OUT) # E
F = Pin(8,Pin.OUT) # F
G = Pin(7,Pin.OUT) # G
```

```
while True:
    G.on()
    sleep(1)
```

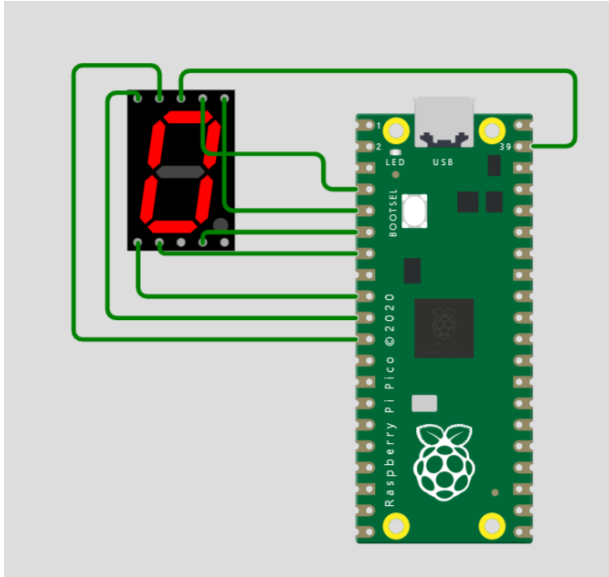


# เขียนโค้ดดังนี้

```
from machine import Pin
from utime import sleep

pins = [
    Pin(2, Pin.OUT), # A
    Pin(3, Pin.OUT), # B
    Pin(4, Pin.OUT), # C
    Pin(5, Pin.OUT), # D
    Pin(6, Pin.OUT), # E
    Pin(8, Pin.OUT), # F
    Pin(7, Pin.OUT), # G
    Pin(0, Pin.OUT) # DP (not connected)
]

digits = [
    [0, 0, 0, 0, 0, 0, 1, 1], # 0
    [1, 0, 0, 1, 1, 1, 1, 1], # 1
    [0, 0, 1, 0, 0, 1, 0, 1], # 2
    [0, 0, 0, 0, 1, 1, 0, 1], # 3
    [1, 0, 0, 1, 1, 0, 0, 1], # 4
    [0, 1, 0, 0, 1, 0, 0, 1], # 5
    [0, 1, 0, 0, 0, 0, 0, 1], # 6
    [0, 0, 0, 1, 1, 1, 1, 1], # 7
    [0, 0, 0, 0, 0, 0, 0, 1], # 8
    [0, 0, 0, 1, 1, 0, 0, 1], # 9
    [0, 0, 0, 1, 0, 0, 0, 1], # a
    [1, 1, 0, 0, 0, 0, 0, 1], # b
    [0, 1, 1, 0, 0, 0, 1, 1], # c
    [1, 0, 0, 0, 0, 1, 0, 1], # d
    [0, 1, 1, 0, 0, 0, 0, 1], # e
    [0, 1, 1, 1, 0, 0, 0, 1], # f
]
```



```
def reset():
    for pin in pins:
        pin.value(1)

reset()

switch = Pin(11, Pin.IN)

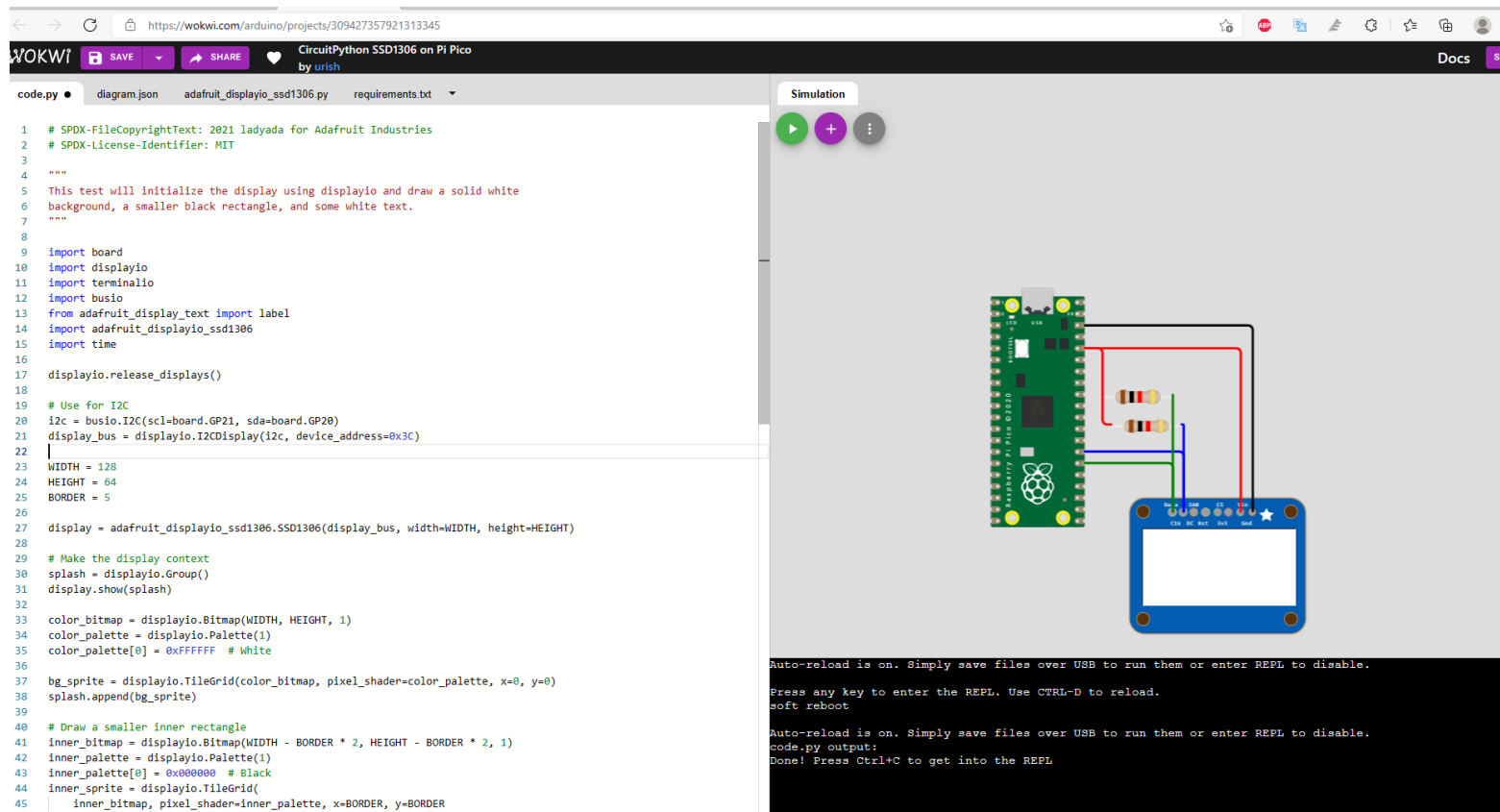
while True:
    for i in range(len(digits)):
        if switch.value() == 0:
            break;
        for j in range(len(pins) - 1):
            pins[j].value(digits[i][j])
        sleep(1.5)
```

# OLED Display

<https://docs.wokwi.com/guides/circuitpython>

## Project examples

- [Blink with CircuitPython](#)
- [CircuitPython SSD1306 Example](#)



WOKWI

SAVE SHARE

CircuitPython SSD1306 on Pi Pico by urish

code.py diagram.json adafruit\_displayio\_ssd1306.py requirements.txt

```

1 # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2 # SPDX-License-Identifier: MIT
3
4
5 This test will initialize the display using displayio and draw a solid white
6 background, a smaller black rectangle, and some white text.
7
8
9 import board
10 import displayio
11 import terminalio
12 import busio
13 from adafruit_display_text import label
14 import adafruit_displayio_ssd1306
15 import time
16
17 displayio.release_displays()
18
19 # Use for I2C
20 i2c = busio.I2C(scl=board.GP21, sda=board.GP20)
21 display_bus = displayio.I2CDisplay(i2c, device_address=0x3C)
22
23 WIDTH = 128
24 HEIGHT = 64
25 BORDER = 5
26
27 display = adafruit_displayio_ssd1306.SSD1306(display_bus, width=WIDTH, height=HEIGHT)
28
29 # Make the display context
30 splash = displayio.Group()
31 display.show(splash)
32
33 color_bitmap = displayio.Bitmap(WIDTH, HEIGHT, 1)
34 color_palette = displayio.Palette(1)
35 color_palette[0] = 0xFFFFFF # White
36
37 bg_sprite = displayio.TileGrid(color_bitmap, pixel_shader=color_palette, x=0, y=0)
38 splash.append(bg_sprite)
39
40 # Draw a smaller inner rectangle
41 inner_bitmap = displayio.Bitmap(WIDTH - BORDER * 2, HEIGHT - BORDER * 2, 1)
42 inner_palette = displayio.Palette(1)
43 inner_palette[0] = 0x000000 # Black
44 inner_sprite = displayio.TileGrid(
45     inner_bitmap, pixel_shader=inner_palette, x=BORDER, y=BORDER
46 )
  
```

Simulation

Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.

Press any key to enter the REPL. Use CTRL-D to reload.

soft reboot

Auto-reload is on. Simply save files over USB to run them or enter REPL to disable.

code.py output:

Done! Press Ctrl+C to get into the REPL



# PROJECT



**KMITL**  
**FIGHT**