INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

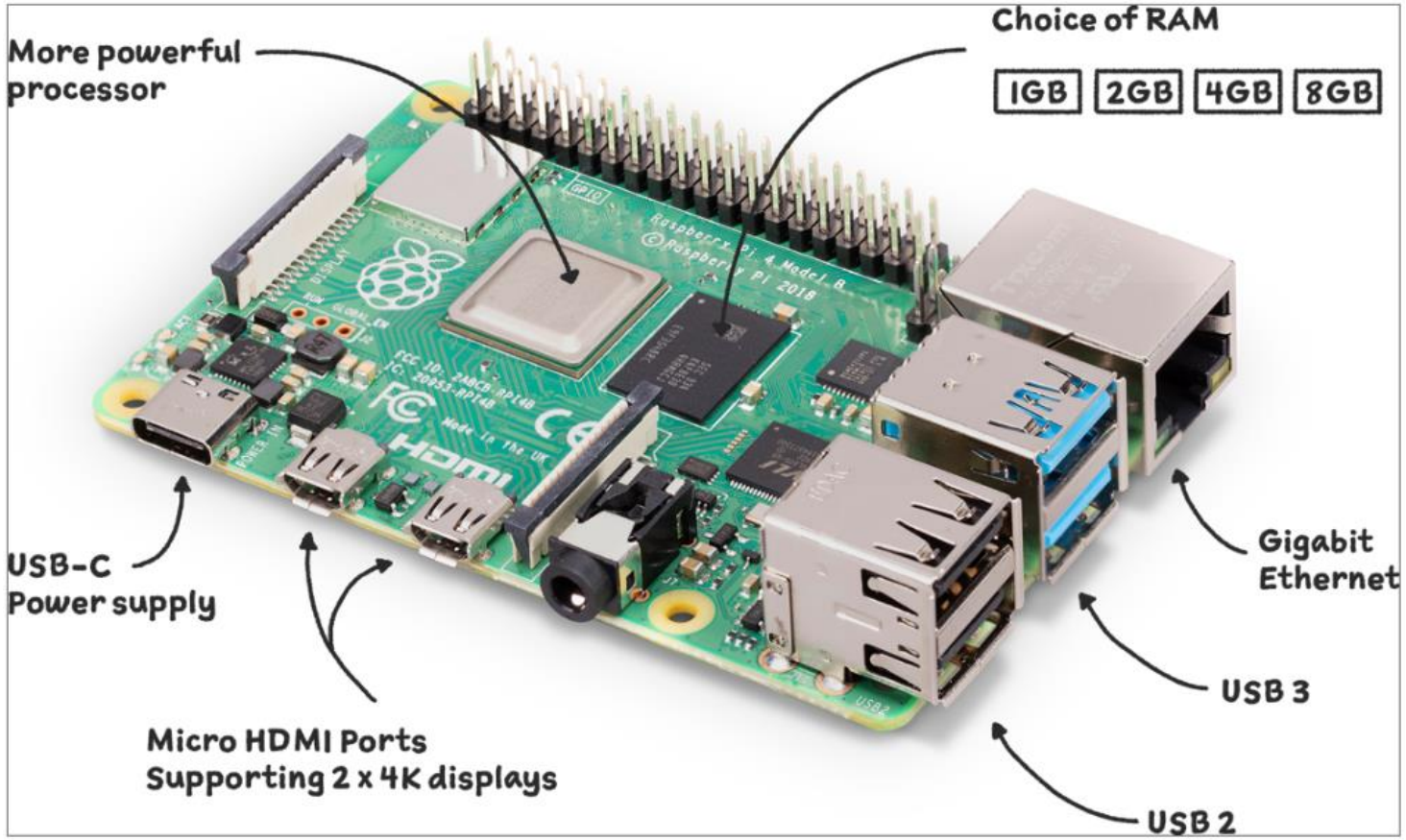
Modern IoT Technology

An Introduction to Raspberry

What is a Raspberry Pi?

- A Raspberry Pi is a mini computer based on the ARM architecture designed to increase accessibility to computing. It is designed to be cheap and easy to use, and is a great way to learn about computers and programming.
- Various applications involving robots, remote monitoring, citizen science, and so on

Raspberry Pi



RP2040 microcontroller

- A dual-core ARM Cortex-M0+ microcontroller
- Static random-access memory (SRAM) 264 kilobytes (KB)
- Volley of peripherals: Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI), and Programmable Input/Output (PIO).
- Enables you to design your own interface: universal asynchronous receiver-transmitter (UART) interface or a video interface.

References for RP2040

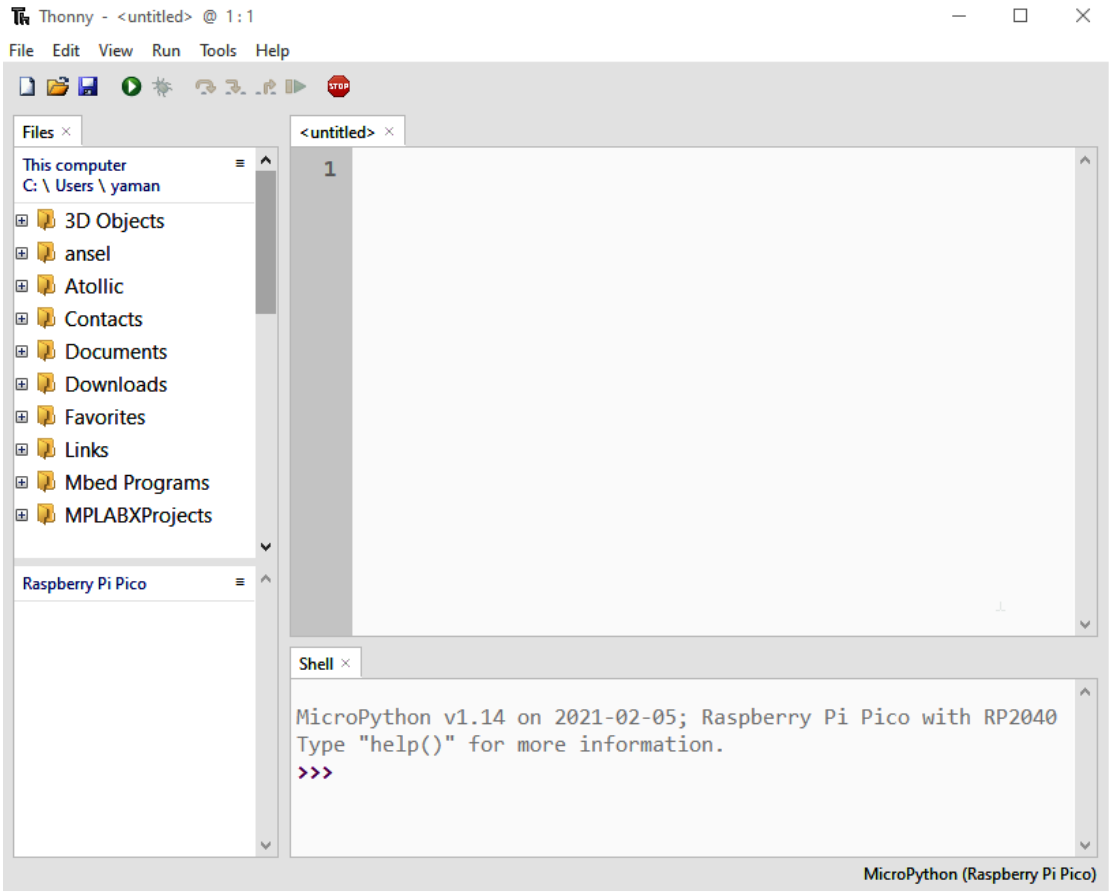
- The datasheet for the RP2040
- https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf
- The datasheet for the Raspberry Pi Pico
- https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf
- Resources for RP2040
- https://www.raspberrypi.com/documentation/microcontrollers/

Raspberry Pi Installer

- https://www.raspberrypi.com/documentation/computers/getting-started.html

MicroPython

Thonny IDE (IDE stands for integrated development environment)

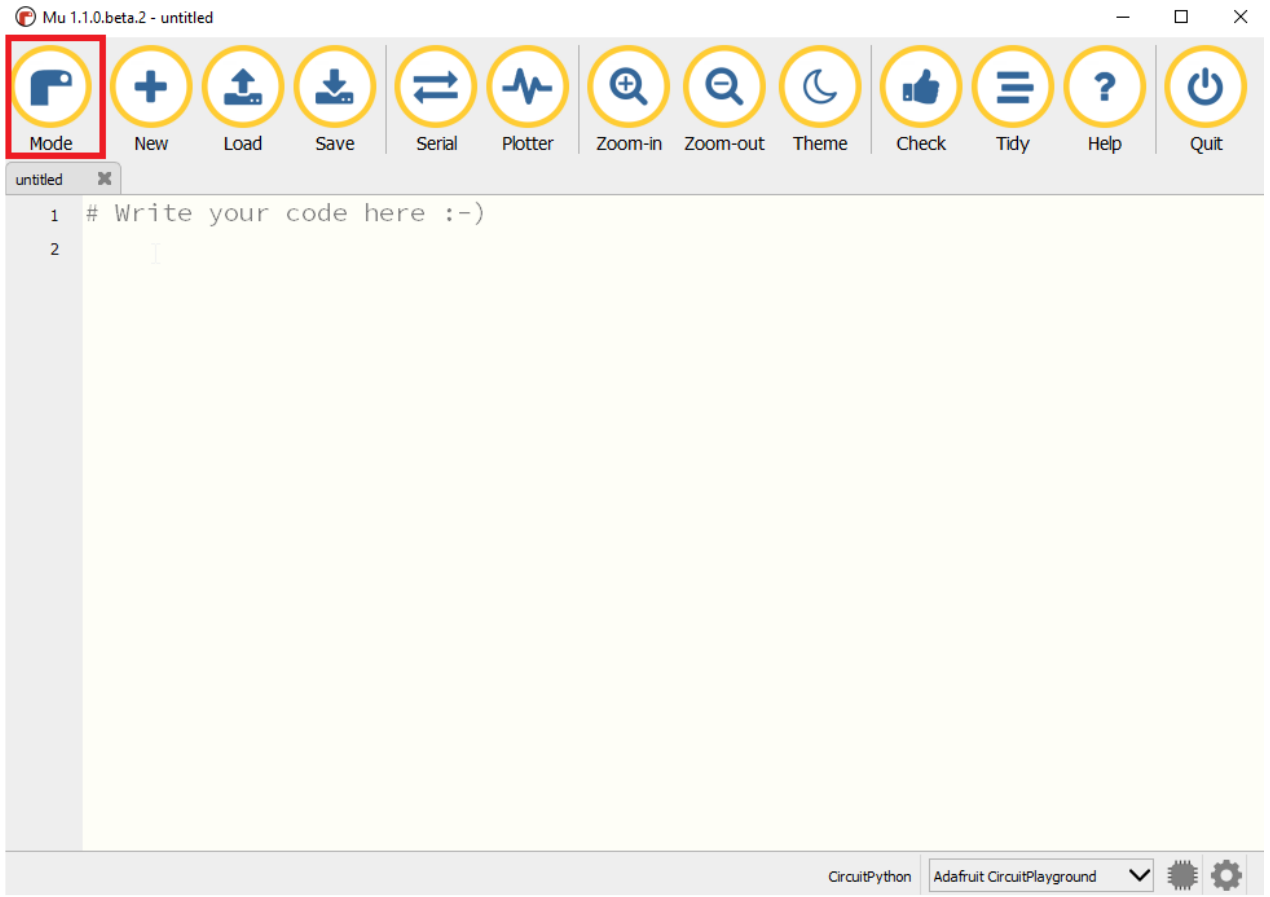


https://thonny.org

Windows, Mac, and Linux operating systems

CircuitPython

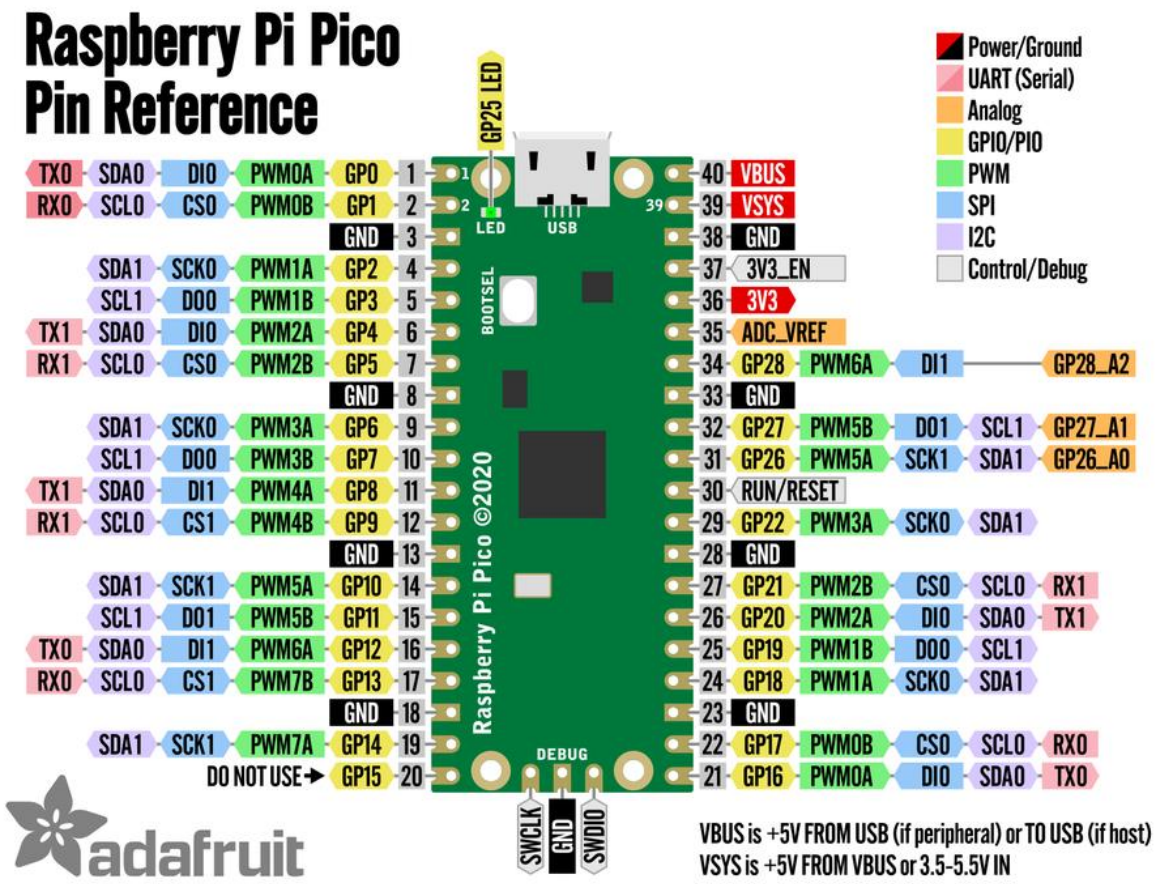
Mu IDE



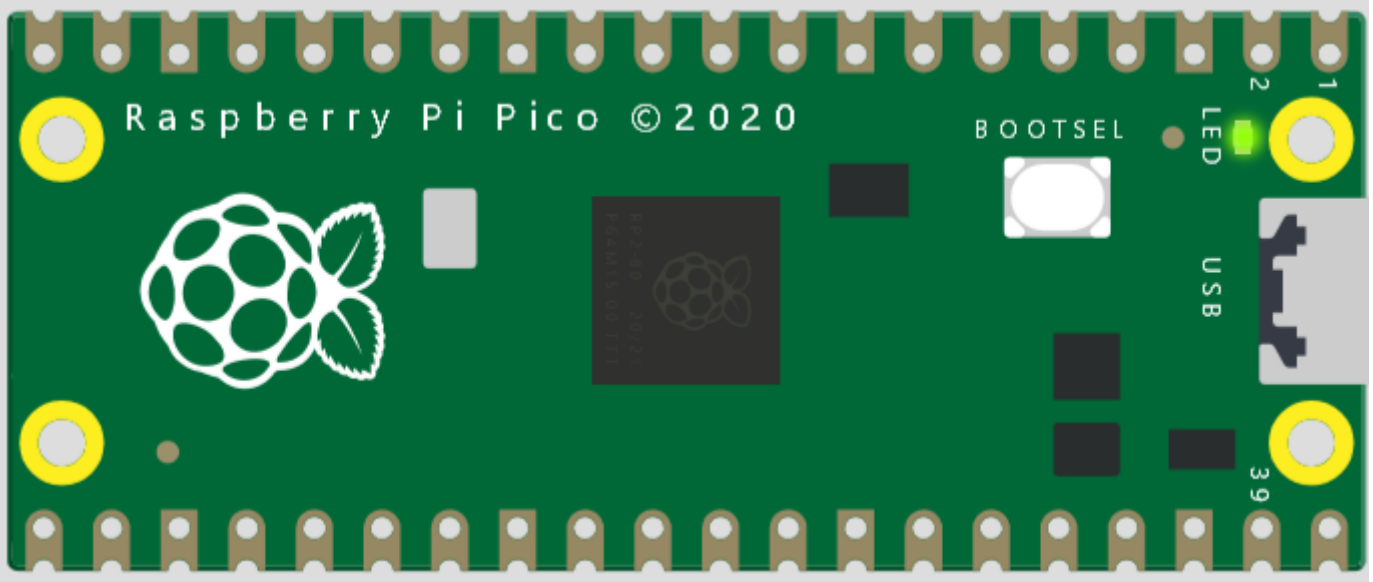
Windows, Mac, and Linux operating systems

https://codewith.mu/en/download

Pinout and design



Blink Built-in Led



CircuitPython

```
from machine import Pin
import utime
led = Pin(25, Pin.OUT)
while True:
    led.toggle()
    utime.sleep(1)
```

CircuitPython

```
import time
import board
import digitalio
led = digitalio.DigitalInOut(board.LED)
led.direction = digitalio.Direction.OUTPUT
while True:
        led.value = True
        time.sleep(1)
        led.value = False
        time.sleep(2)
```

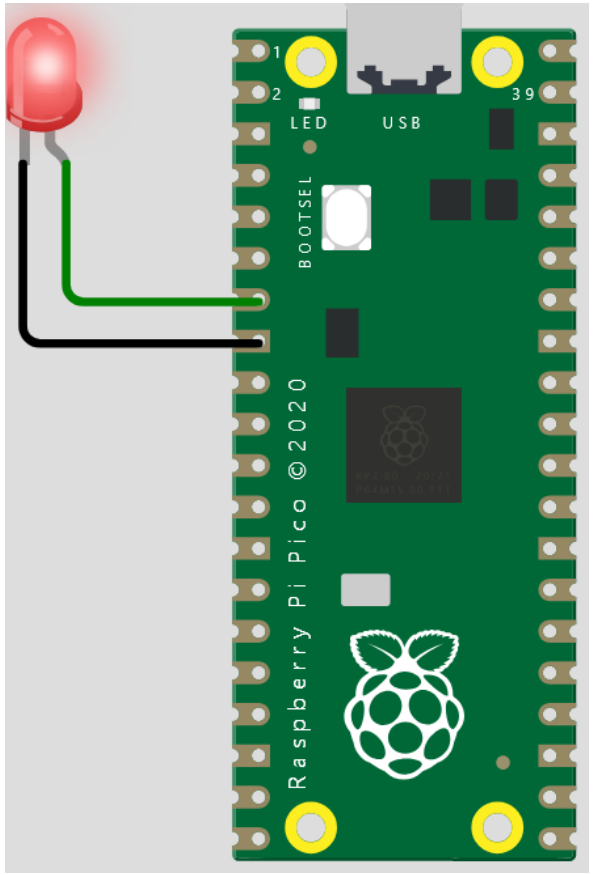
Code C-Python

```
ledpy = machine.Pin("LED", machine.Pin.OUT)
```

```
#include "pico/stdlib.h"

int main() {
#ifndef PICO_DEFAULT_LED_PIN
#warning blink example requires a board with a
regular LED
#else
    const uint LED_PIN = PICO_DEFAULT_LED_PIN;
    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);
    while (true) {
        gpio_put(LED_PIN, 1);
        sleep_ms(250);
        gpio_put(LED_PIN, 0);
        sleep_ms(250);
    }
#endif
}
```

Built-in Led

```
from machine import Pin
from utime import sleep

print("Hello, Pi Pico!")

led = Pin(5, Pin.OUT)
while True:
  led.toggle()
  sleep(0.5)
```

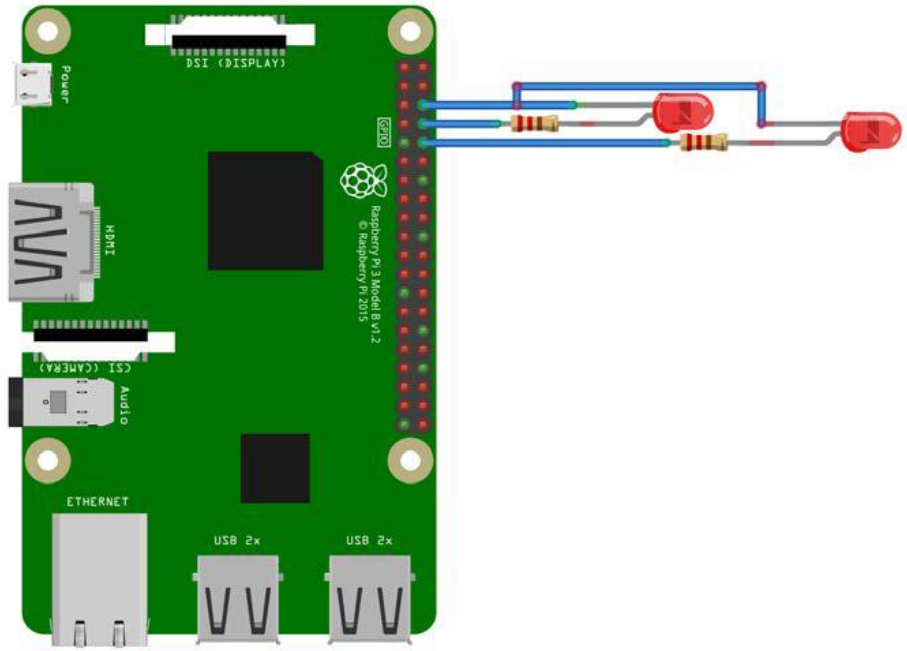


Choosing between IoT hardware

- sudo apt-get install python3-rpi.gpio -y

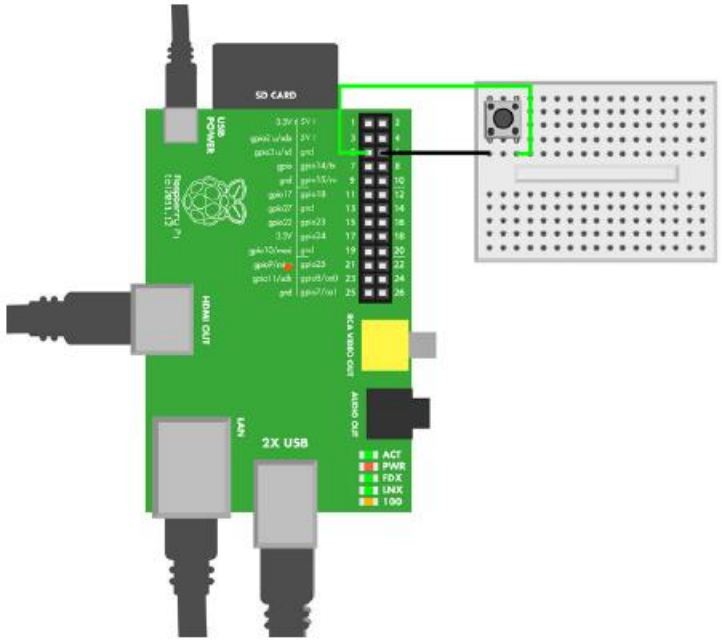
```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)
while True:
    GPIO.output(8, GPIO.HIGH)
    sleep(1)
    GPIO.output(8, GPIO.LOW)
    sleep(1)
```

Exercise

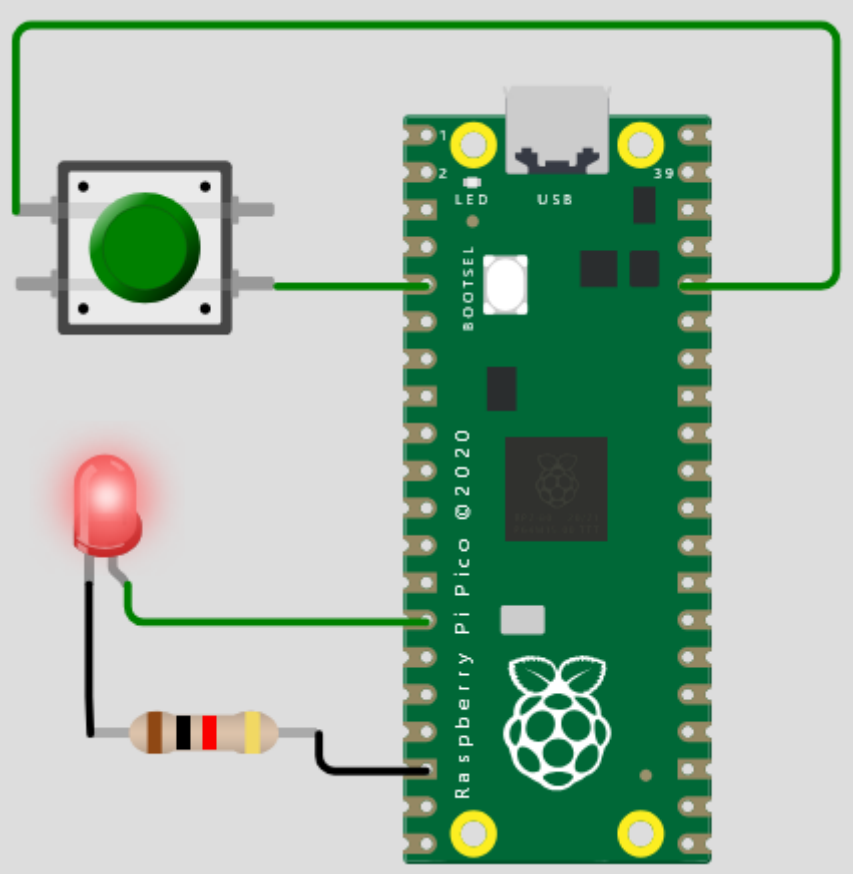

DSI (DISPLAY)
Power
HDMI
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015

Push-button programming

GPIO pin (BCM)	Board pin (place in header)
GPIO 3	5
GND	6

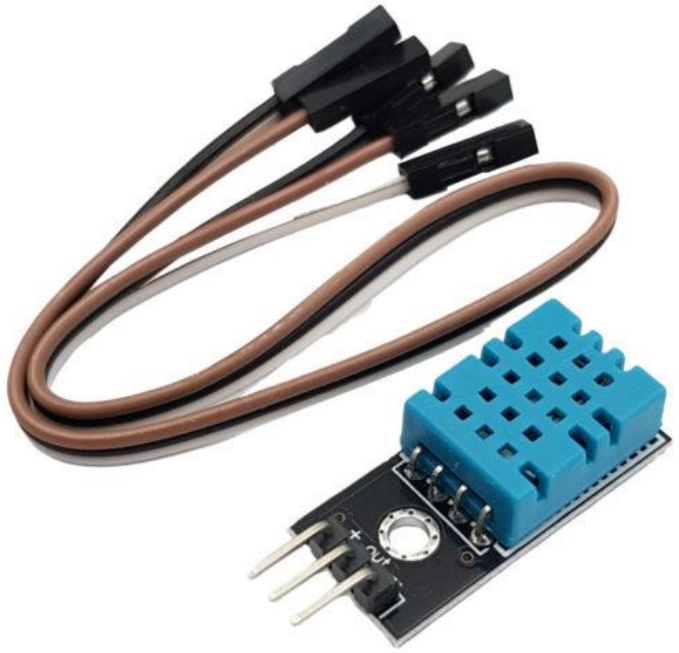


Pull-Down Button

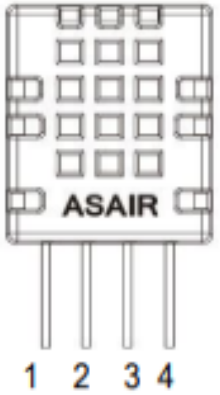



```
import machine
import time
print("Hello, Pi Pico!")
led = machine.Pin(10, machine.Pin.OUT)
button = machine.Pin(3,machine.Pin.IN,
machine.Pin.PULL_DOWN)
while True:
    if button.value():
        print(button.value())
        led.toggle()
        time.sleep(0.5) # Wait for USB to
become ready
```

DHT20 temperature sensor

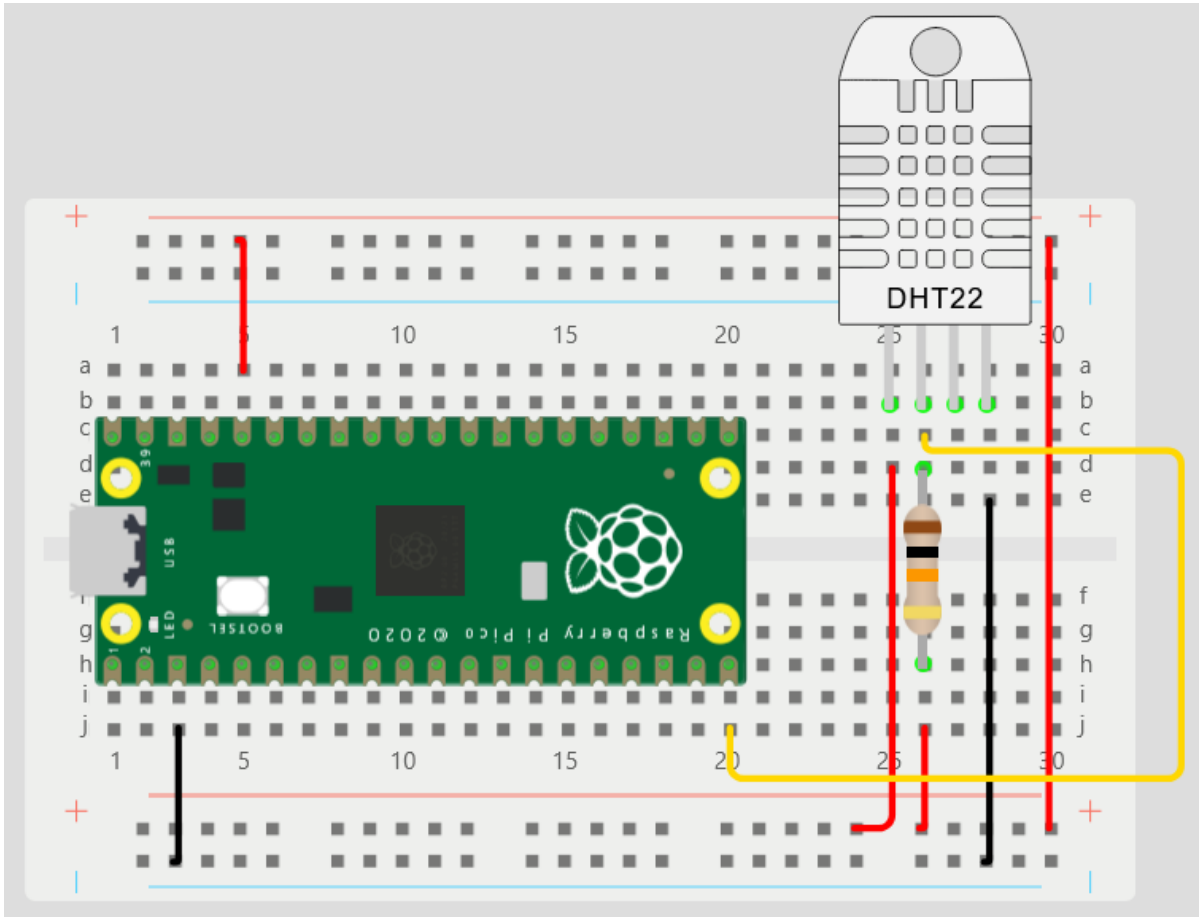


Pins	Name	Describe
1	VDD	Power supply (2.2v to 5.5v)
2	SDA	Serial data bidirectional port
3	GND	Ground
4	SCL	Serial clock bidirectional port



Testing the DHT20 temperature sensor

```
# Hardware connections used:
#    DHT22 VCC Pin to 3.3V
#    DHT22 SDA Pin to GPIO Pin 15
#    10k ohm pull-up resistor
from DHT22 SDA Pin to 3.3V
#    DHT22 GND Pin to GND
```



Testing the DHT20 temperature sensor

```
from machine import Pin
from time import sleep
from dht import DHT22
# creating a DHT object
dht = DHT22(Pin(15))

# continuously get sensor readings while the board has
power
while True:
    # getting sensor readings
    dht.measure()
    temp = dht.temperature()
    hum = dht.humidity()

    # displaying values to the console
    print(f"Temperature: {temp}°C   Humidity: {hum}% ")

    sleep(2)
```

CircuitPython REPL

```
 Temperature: 22.4 C
Humidity: 43.4 %

 Temperature: 22.4 C
Humidity: 43.3 %

 Temperature: 22.4 C
Humidity: 43.5 %

 Temperature: 22.4 C
Humidity: 43.6 %
```