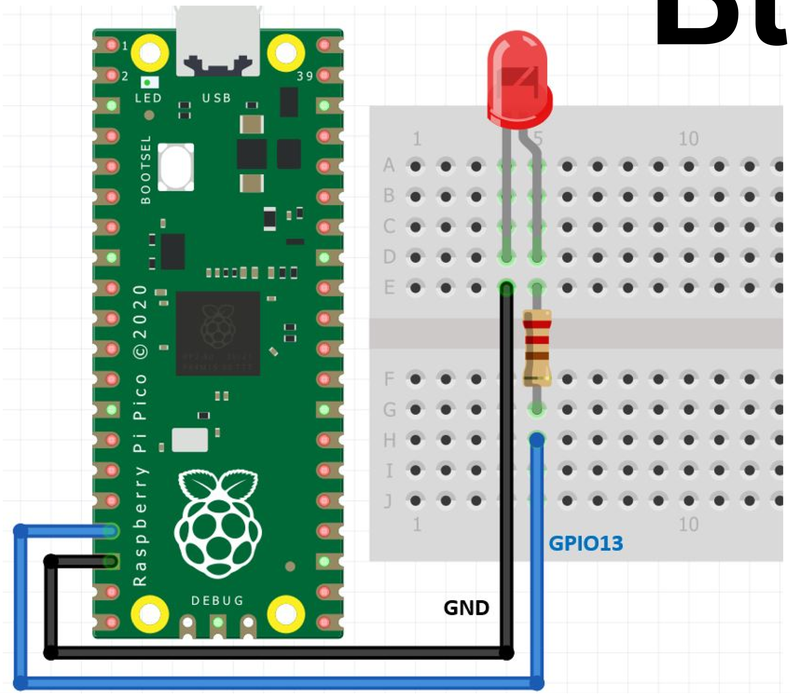
Gearbox Academy

"One step at a time.
You'll get there."

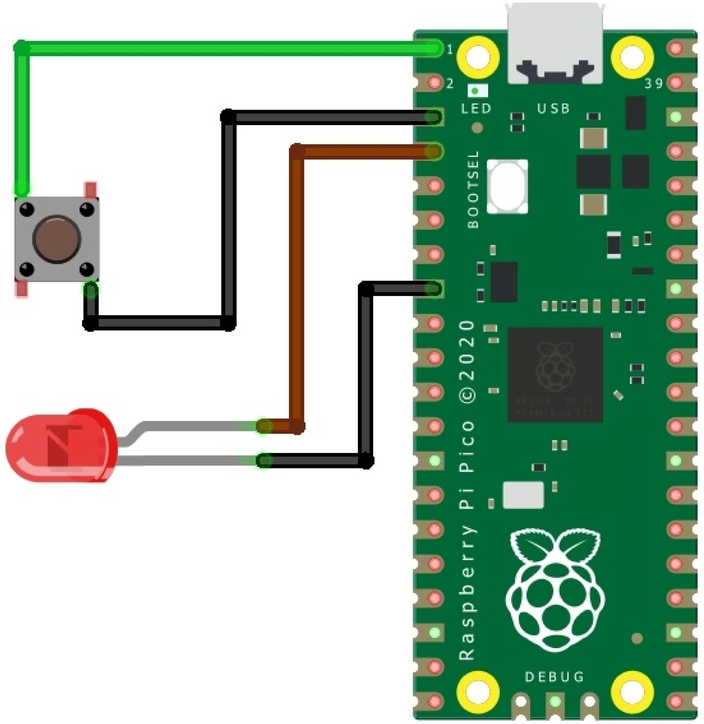
Blink Project



LED	Raspberry Pi Pico
Cathode (-)	GND
Anode (+)	GPIO13

```
import time
from machine import Pin
led=Pin(13,Pin.OUT)          #create LED object from
pin13,Set Pin13 to output

while True:
  led.value(1)              #Set led turn on
  time.sleep(0.5)
  led.value(0)              #Set led turn off
  time.sleep(0.5)
```

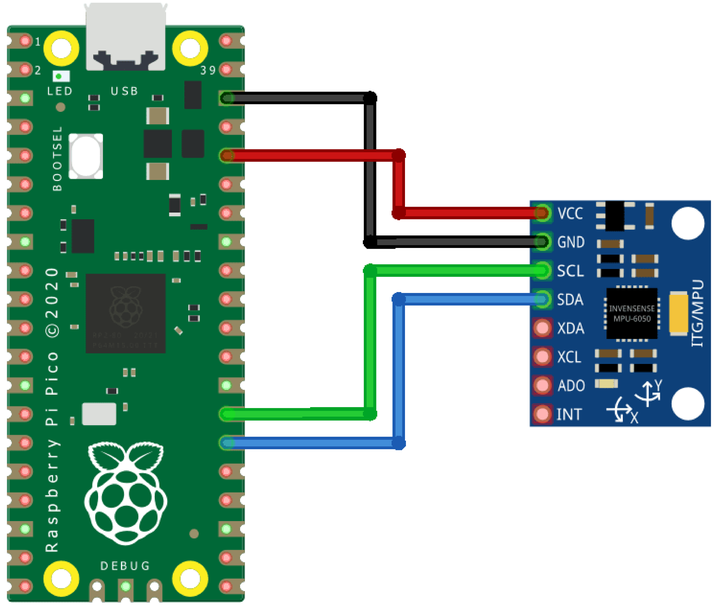
LED Control using Push Button with Raspberry Pi Pico



A Push button, also known as a tact switch, is a kind of electronic component. Its internal structure is connected and disconnected by the force change of metal shrapnel. When in use, press the switch, The switch is closed and the circuit is turned on. When released, the switch bounces and the circuit is disconnected.

```
from machine import Pin
from utime import sleep_ms
button = Pin(0, Pin.IN, Pin.PULL_UP)   #Internal pull-up
led = Pin(1, Pin.OUT)
State=0                                #0 means that the
light is currently off
if __name__ == '__main__':
    while True:
        print(button.value())
        if button.value() == 0:       #key press
            if State==0:
                led.value(1)
                sleep_ms=100
                while button.value() == 0:
                   State=1
            else:
                led.value(0)
                sleep_ms=100
                while button.value()== 0:
                    State=0
```

Measuring Acceleration with the MPU6050 accelerometer via I2C

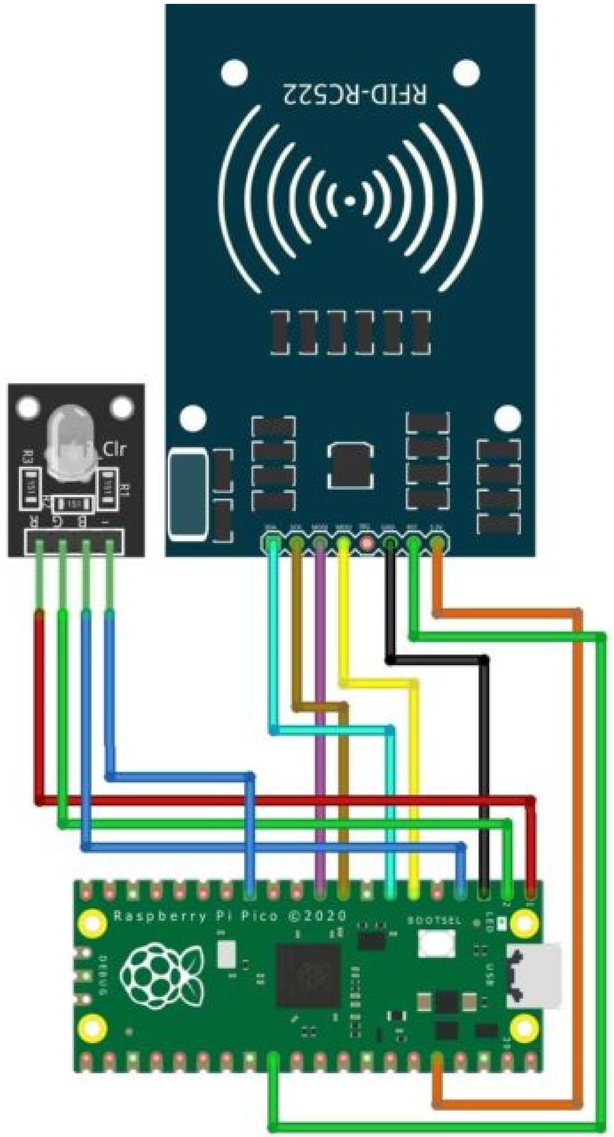


The sensor combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP) in a single integrated circuit (IC) package. The MPU6050 is widely used in various applications such as robotics, smartphones, gaming devices, and wearable technology, thanks to its accurate motion tracking capabilities and high sensitivity.

```
from machine import Pin, I2C
import utime

PWR_MGMT_1 = 0x6B
SMPLRT_DIV = 0x19
CONFIG = 0x1A
GYRO_CONFIG = 0x1B
ACCEL_CONFIG = 0x1C
TEMP_OUT_H = 0x41
ACCEL_XOUT_H = 0x3B
GYRO_XOUT_H = 0x43

def init_mpu6050(i2c, address=0x68):
    i2c.writeto_mem(address, PWR_MGMT_1,
b'\x00')
    utime.sleep_ms(100)
    i2c.writeto_mem(address, SMPLRT_DIV,
b'\x07')
    i2c.writeto_mem(address, CONFIG,
b'\x00')
    i2c.writeto_mem(address, GYRO_CONFIG,
b'\x00')
    i2c.writeto_mem(address, ACCEL_CONFIG,
b'\x00')

def read_raw_data(i2c, addr, address=0x68):
    high = i2c.readfrom_mem(address, addr,
1)[0]
    low = i2c.readfrom_mem(address, addr +
1, 1)[0]
    value = high << 8 | low
    if value > 32768:
        value = value - 65536
    return value

    }
```

```
def get_mpu6050_data(i2c):
    temp = read_raw_data(i2c,
TEMP_OUT_H) / 340.0 + 36.53
    accel_x = read_raw_data(i2c,
ACCEL_XOUT_H) / 16384.0
    accel_y = read_raw_data(i2c,
ACCEL_XOUT_H + 2) / 16384.0
    accel_z = read_raw_data(i2c,
ACCEL_XOUT_H + 4) / 16384.0
    gyro_x = read_raw_data(i2c,
GYRO_XOUT_H) / 131.0
    gyro_y = read_raw_data(i2c,
GYRO_XOUT_H + 2) / 131.0
    gyro_z = read_raw_data(i2c,
GYRO_XOUT_H + 4) / 131.0

    return {
        'temp': temp,
        'accel': {
            'x': accel_x,
            'y': accel_y,
            'z': accel_z,
        },
        'gyro': {
            'x': gyro_x,
            'y': gyro_y,
            'z': gyro_z,
        }
```

Interfacing RC522 RFID Reader Module with Raspberry Pi Pico



Raspberry Pi Pico	MFRC522 RFID module
GND	GND
3V3	3.3V
GP5	SDA
GP6	SCK
GP7	MOSI
GP4	MISO
—	IRQ
GPIO22	RST
RGB LED Module	**Raspberry Pi Pico**
R	GP13
G	GP12
B	GP11
–	GND

For code check out.

https://microcontrollerslab.com/raspberry-pi-pico-rfid-rc522-micropython/