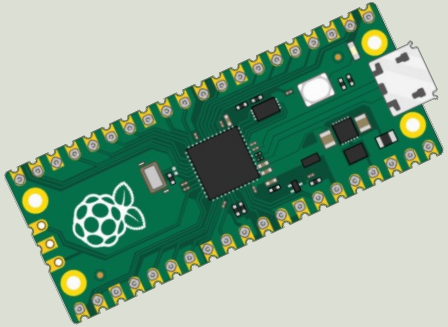
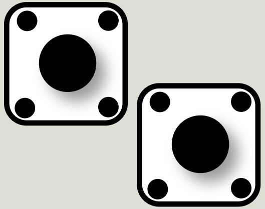
https://www.halvorsen.blog



Raspberry Pi Pico

Push Buttons



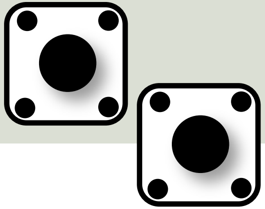
Hans-Petter Halvorsen

Contents

- Introduction
  - Raspberry Pi Pico
  - Thonny Python Editor
  - MicroPython
- Push Buttons
  - Push Button with Pull-up Resistor
  - Push Button with Pull-down Resistor
  - Push Button and LED

Introduction

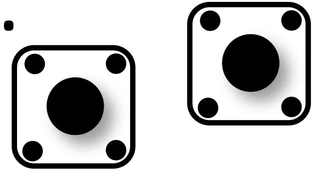
Introduction



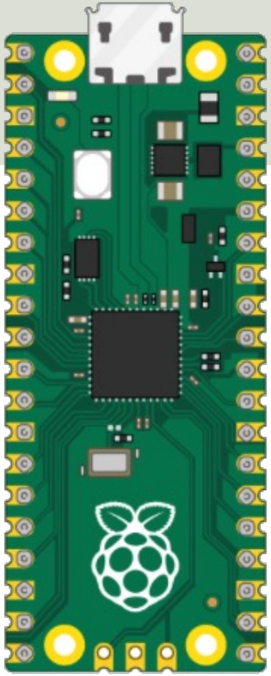
- In this Tutorial we will show how we can use a Push Buttons with Raspberry Pi Pico
- We will use MicroPython
- Pushbuttons or switches connect two points in a circuit when you press them.
- You can use it to turn on a Light/LED when holding down the button, etc.
- Push Buttons have many Applications, we will show some basic examples here

What do you need?

- Raspberry Pi Pico
- A Micro-USB cable
- A PC with Thonny Python Editor (or another Python Editor)
- Breadboard
- Electronics Components like LED, Resistors, Jumper wires, etc.
- Push Buttons



Raspberry Pi Pico



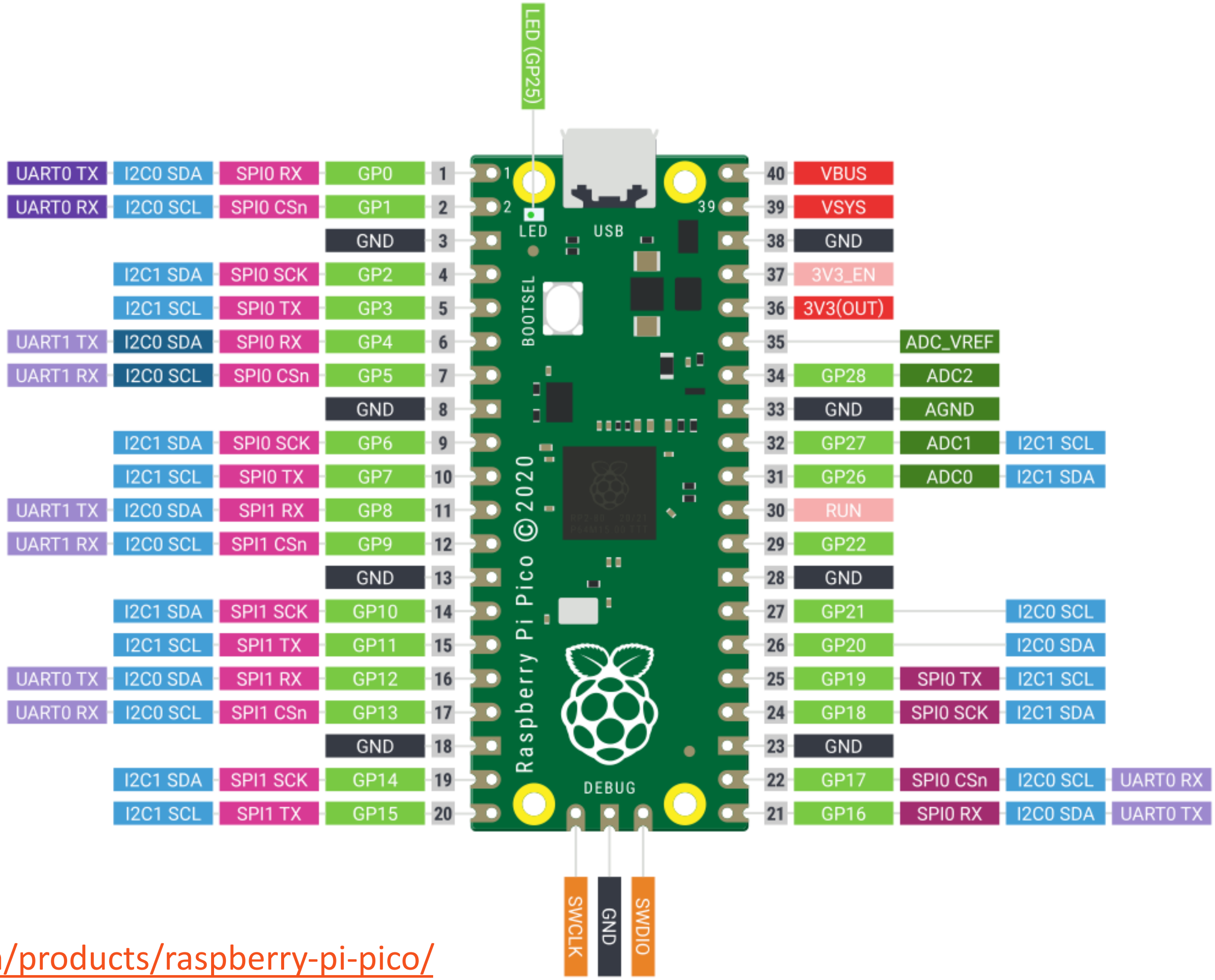
- Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation
- Raspberry Pi Pico has similar features as Arduino devices
- Raspberry Pi Pico is typically used for Electronics projects, IoT Applications, etc.
- You typically use MicroPython, which is a downscaled version of Python, in order to program it

https://www.raspberrypi.com/products/raspberry-pi-pico/

https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico

Pico Pinout

- 🟥 Power
- ⬛ Ground
- 🟪 UART / UART (default)
- 🟩 GPIO, PIO, and PWM
- 🟩 ADC
- 🟥 SPI / SPI (default)
- 🟦 I2C / I2C (default)
- 🟥 System Control
- 🟧 Debugging

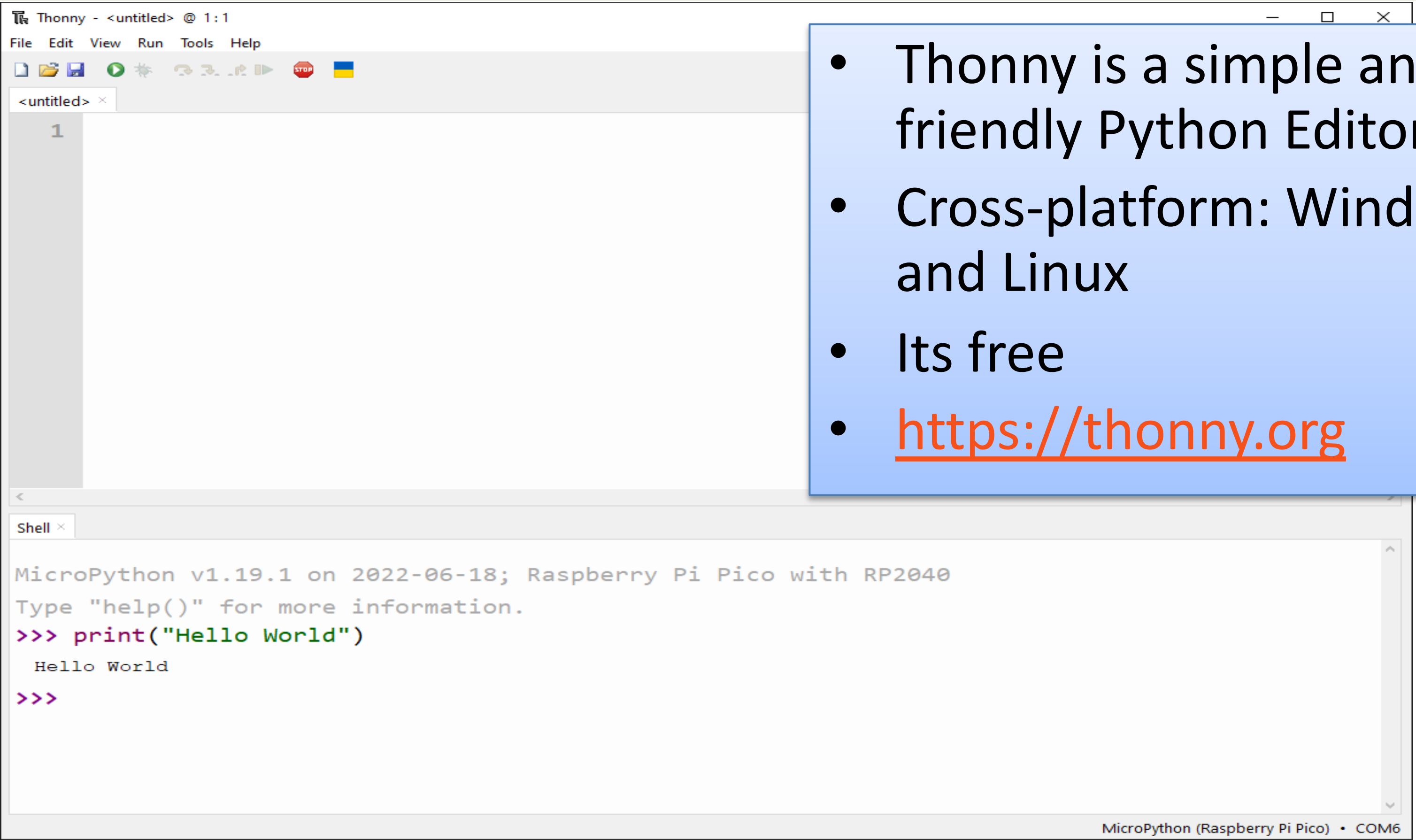


UART0 TX	I2C0 SDA	SPI0 RX	GP0	1	40	VBUS			
UART0 RX	I2C0 SCL	SPI0 CSn	GP1	2	39	VSYS			
			GND	3	38	GND			
	I2C1 SDA	SPI0 SCK	GP2	4	37	3V3_EN			
	I2C1 SCL	SPI0 TX	GP3	5	36	3V3(OUT)			
UART1 TX	I2C0 SDA	SPI0 RX	GP4	6	35		ADC_VREF		
UART1 RX	I2C0 SCL	SPI0 CSn	GP5	7	34	GP28	ADC2		
			GND	8	33	GND	AGND		
	I2C1 SDA	SPI0 SCK	GP6	9	32	GP27	ADC1	I2C1 SCL	
	I2C1 SCL	SPI0 TX	GP7	10	31	GP26	ADC0	I2C1 SDA	
UART1 TX	I2C0 SDA	SPI1 RX	GP8	11	30	RUN			
UART1 RX	I2C0 SCL	SPI1 CSn	GP9	12	29	GP22			
			GND	13	28	GND			
	I2C1 SDA	SPI1 SCK	GP10	14	27	GP21		I2C0 SCL	
	I2C1 SCL	SPI1 TX	GP11	15	26	GP20		I2C0 SDA	
UART0 TX	I2C0 SDA	SPI1 RX	GP12	16	25	GP19	SPI0 TX	I2C1 SCL	
UART0 RX	I2C0 SCL	SPI1 CSn	GP13	17	24	GP18	SPI0 SCK	I2C1 SDA	
			GND	18	23	GND			
	I2C1 SDA	SPI1 SCK	GP14	19	22	GP17	SPI0 CSn	I2C0 SCL	UART0 RX
	I2C1 SCL	SPI1 TX	GP15	20	21	GP16	SPI0 RX	I2C0 SDA	UART0 TX

https://www.raspberrypi.com/products/raspberry-pi-pico/

Thonny

- Thonny is a simple and user-friendly Python Editor
- Cross-platform: Windows, macOS and Linux
- Its free
- https://thonny.org


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
STOP
<untitled>
1
Shell
MicroPython (Raspberry Pi Pico) • COM6

```
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> print("Hello World")
 Hello World
>>>
```

MicroPython

- MicroPython is a downscaled version of Python
- It is typically used for Microcontrollers and constrained systems

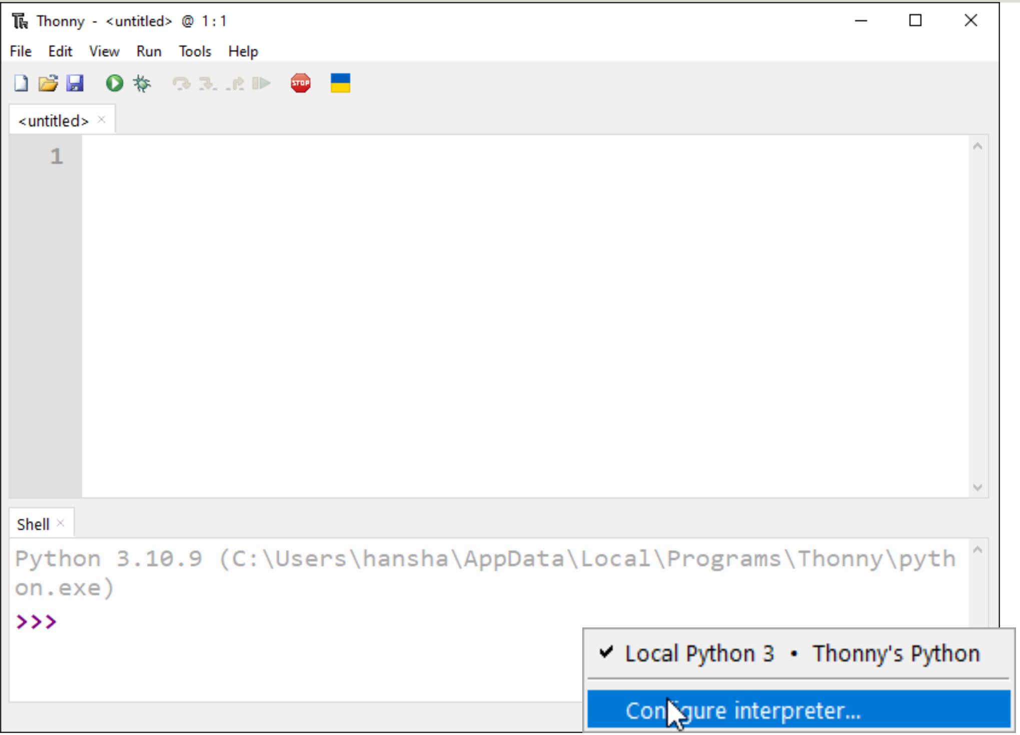
https://docs.micropython.org/en/latest/index.html

https://micropython.org

MicroPython Firmware

- The first time you need to install the MicroPython Firmware on your Raspberry Pi Pico
- You can install the MicroPython Firmware manually or you can use the Thonny Editor

Install MicroPython Firmware using Thonny

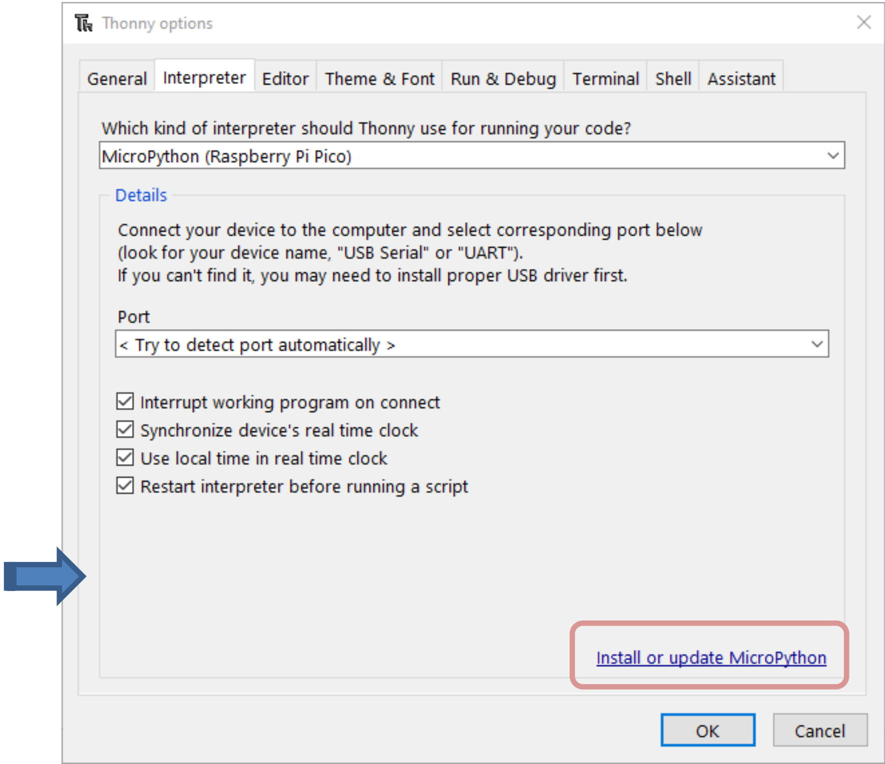


Thonny - <untitled> @ 1 : 1

```
Python 3.10.9 (C:\Users\hansha\AppData\Local\Programs\Thonny\python.exe)
>>>
```

✓ Local Python 3 • Thonny's Python

Configure interpreter...



Thonny options

Which kind of interpreter should Thonny use for running your code?

MicroPython (Raspberry Pi Pico)

Details

Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Port

< Try to detect port automatically >

☑ Interrupt working program on connect
☑ Synchronize device's real time clock
☑ Use local time in real time clock
☑ Restart interpreter before running a script

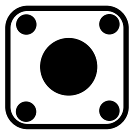
Install or update MicroPython

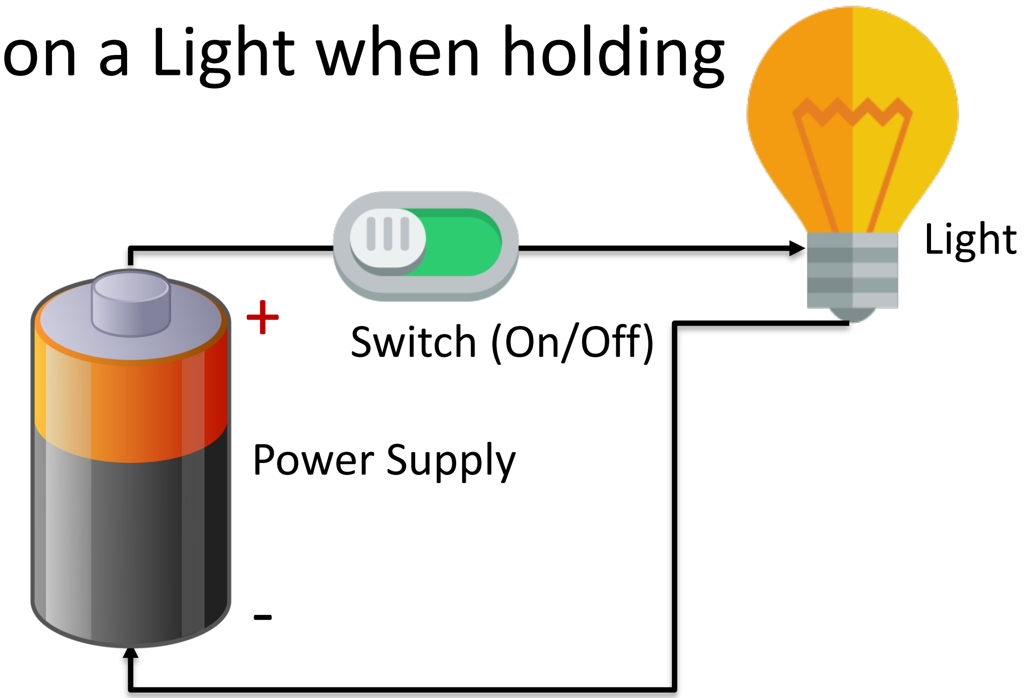
https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/3

Push Buttons

Push Button/Switch

- Pushbuttons or switches connect two points in a circuit when you press them.
- You can use it to turn on a Light when holding down the button, etc.





Push Button Examples

The Tutorial will go through different Examples where we use a Push Button and MicroPython:

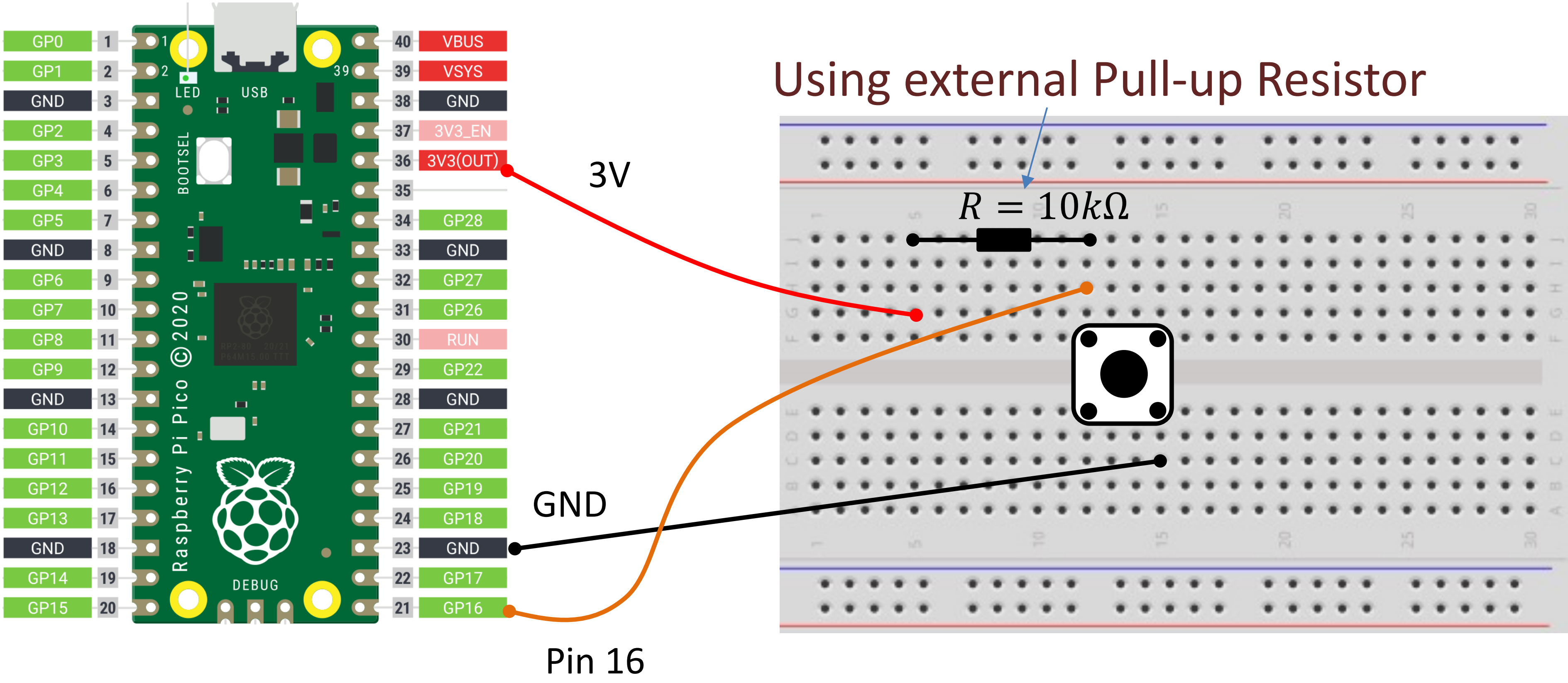
- Push Button with Pull-up Resistor
- Push Button with Pull-down Resistor
- Push Button and LED

Push Button with Pull-up Resistor

Hans-Petter Halvorsen

Table of Contents

Wiring (Pull-up Resistor)

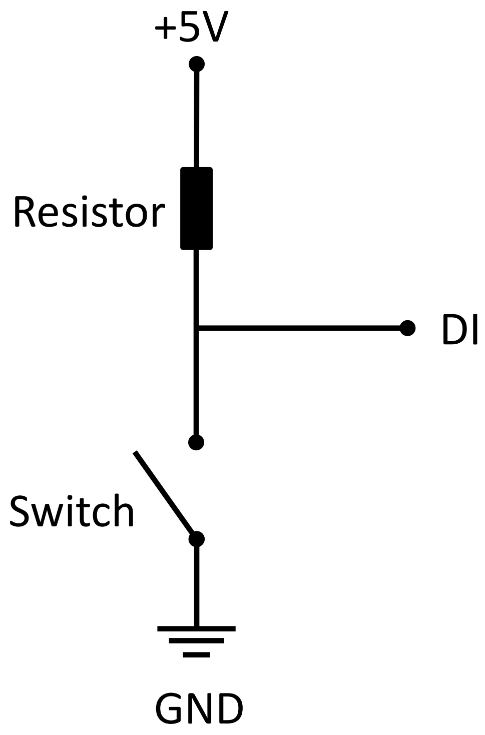


Pull-down/Pull-up Resistor

Why do we need a pull-up or pull-down resistor in the circuit?

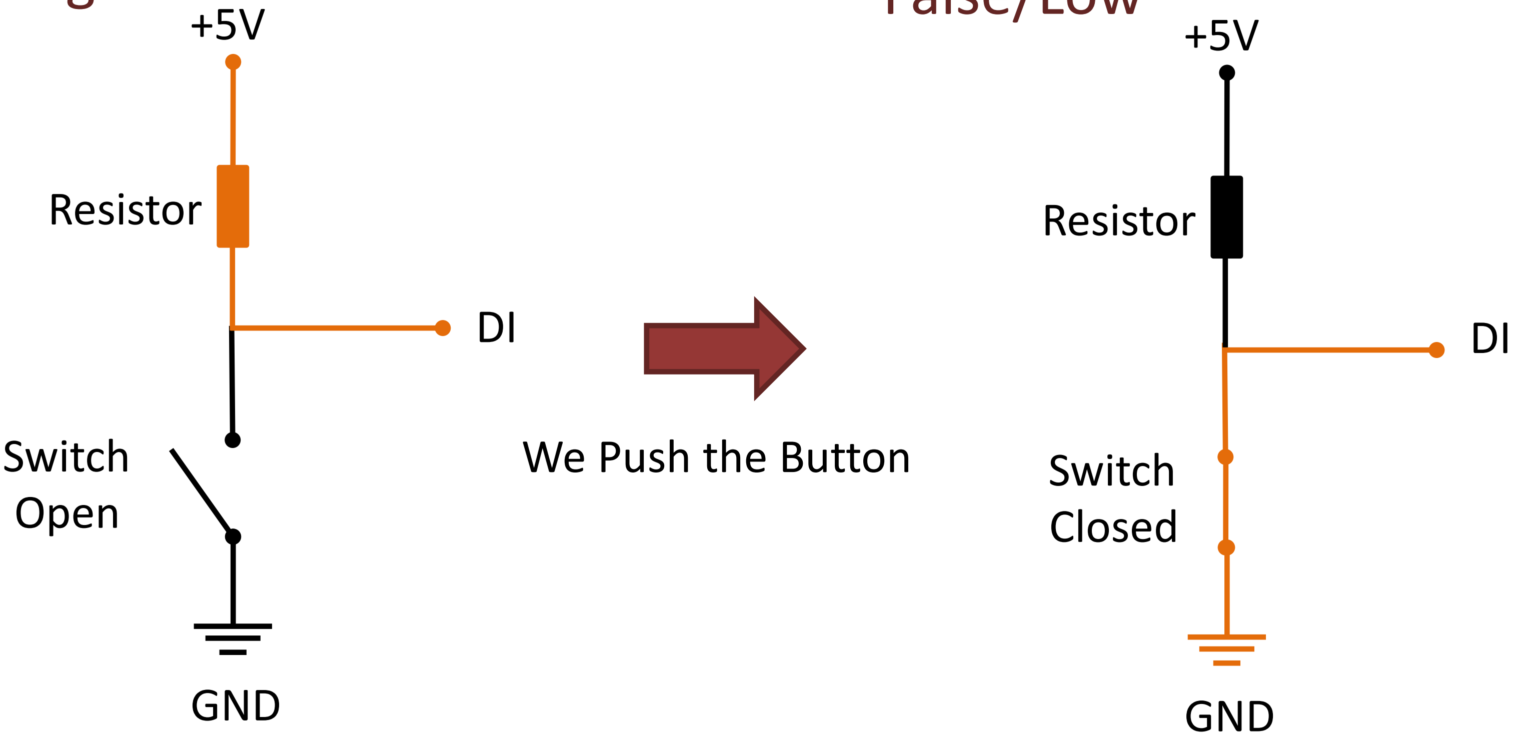
- If you disconnect the digital I/O pin from everything, it will behave in an irregular way.
- This is because the input is "floating" - that is, it will randomly return either HIGH or LOW.
- That's why you need a pull-up or pull-down resistor in the circuit.

Pull-up Resistor



- When the pushbutton is open (unpressed) there is a connection between 5V and the DI pin.
- This means the default state is **True** (High).
- When the button is closed (pressed), the state goes to **False** (Low).

Pull-up Resistor



True/High

False/Low

We Push the Button

Push Button (Pull-up Resistor)

```
import machine
import time

pin = 16
button = machine.Pin(pin, machine.Pin.IN, machine.Pin.PULL_UP)

while True:
    if button.value() == 1:
        print("Please Push the Button")
    else:
        print("Button is Pushed")
    time.sleep(1)
```

Button is NOT Pushed => True/High

Button is Pushed => False/Low

Adding “NOT”

```
from machine import Pin
from time import sleep

pin = 16
button = Pin(pin, Pin.IN, Pin.PULL_UP)

while True:
    if not button.value():
        print("Button is Pushed")
    else:
        print("Please Push the Button")
    sleep(1)
```

Push Button with Pull-down Resistor

Hans-Petter Halvorsen

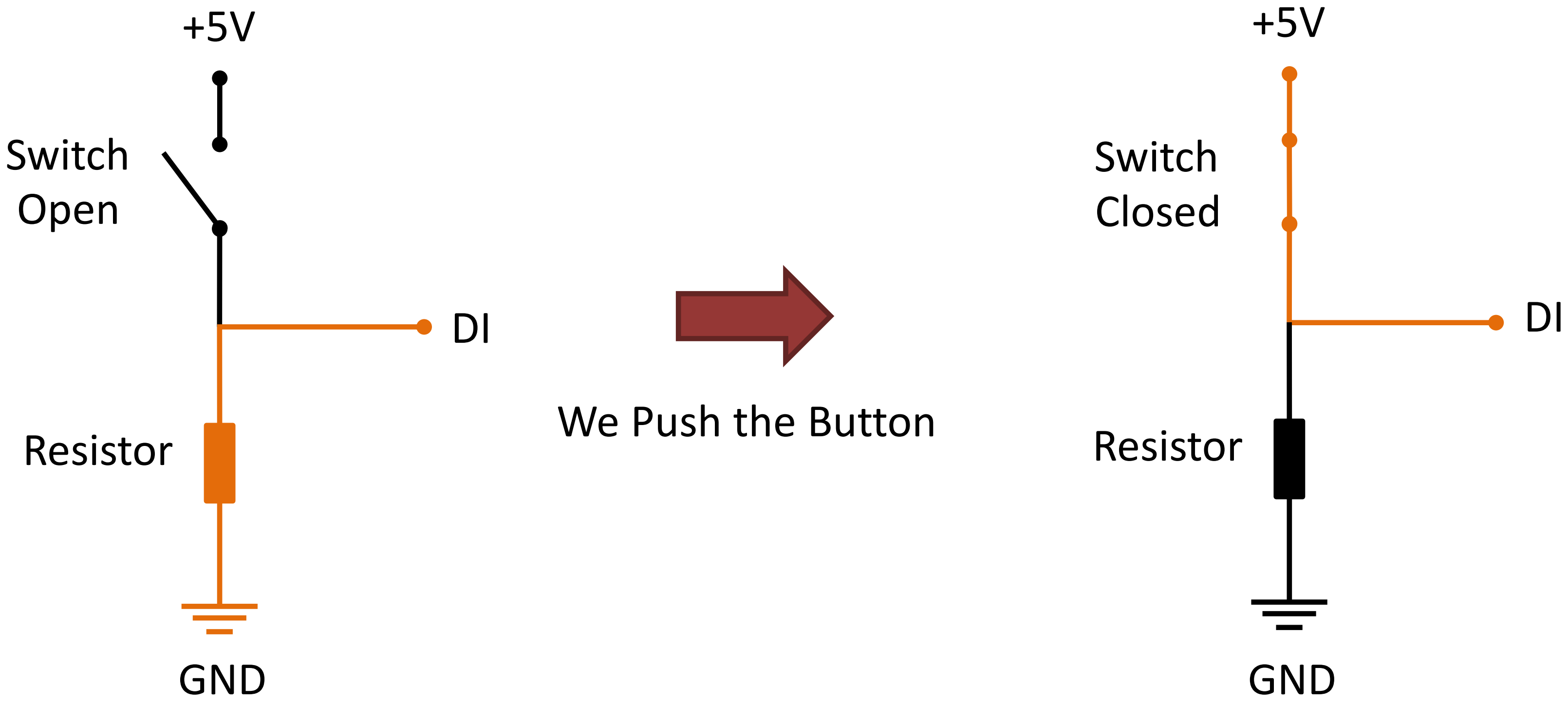
Table of Contents

Pull-down Resistor

We could also have wired according to a “Pull-down“ Resistor

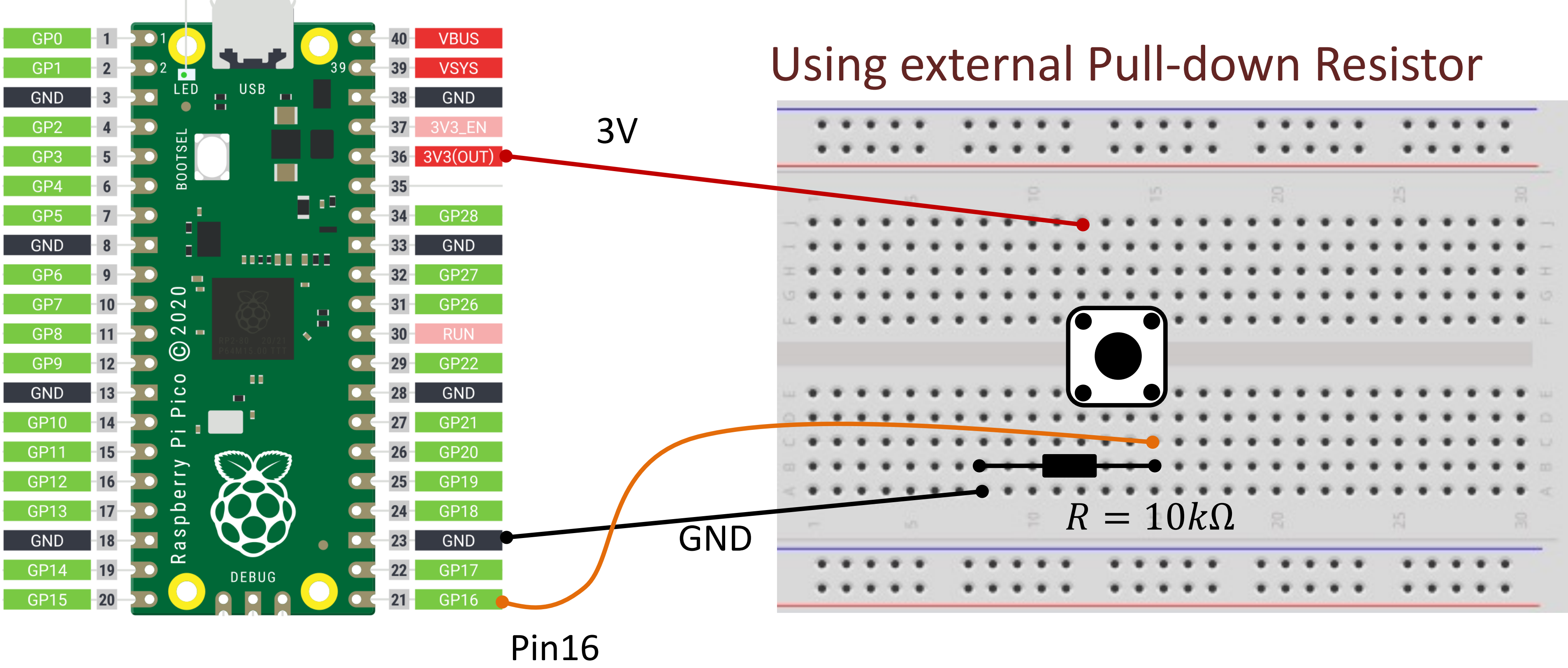
False/Low

True/High



Wiring (Pull-down Resistor)

Using external Pull-down Resistor

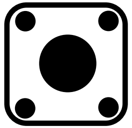


Push Button (Pull-down Resistor)

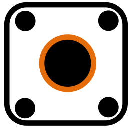
```
from machine import Pin
from time import sleep

pin = 16
button = Pin(pin, Pin.IN, Pin.PULL_DOWN)

while True:
    if button.value():
        print("Button is Pushed")
    else:
        print("Please Push the Button")
    sleep(1)
```



Button is NOT Pushed => False/Low



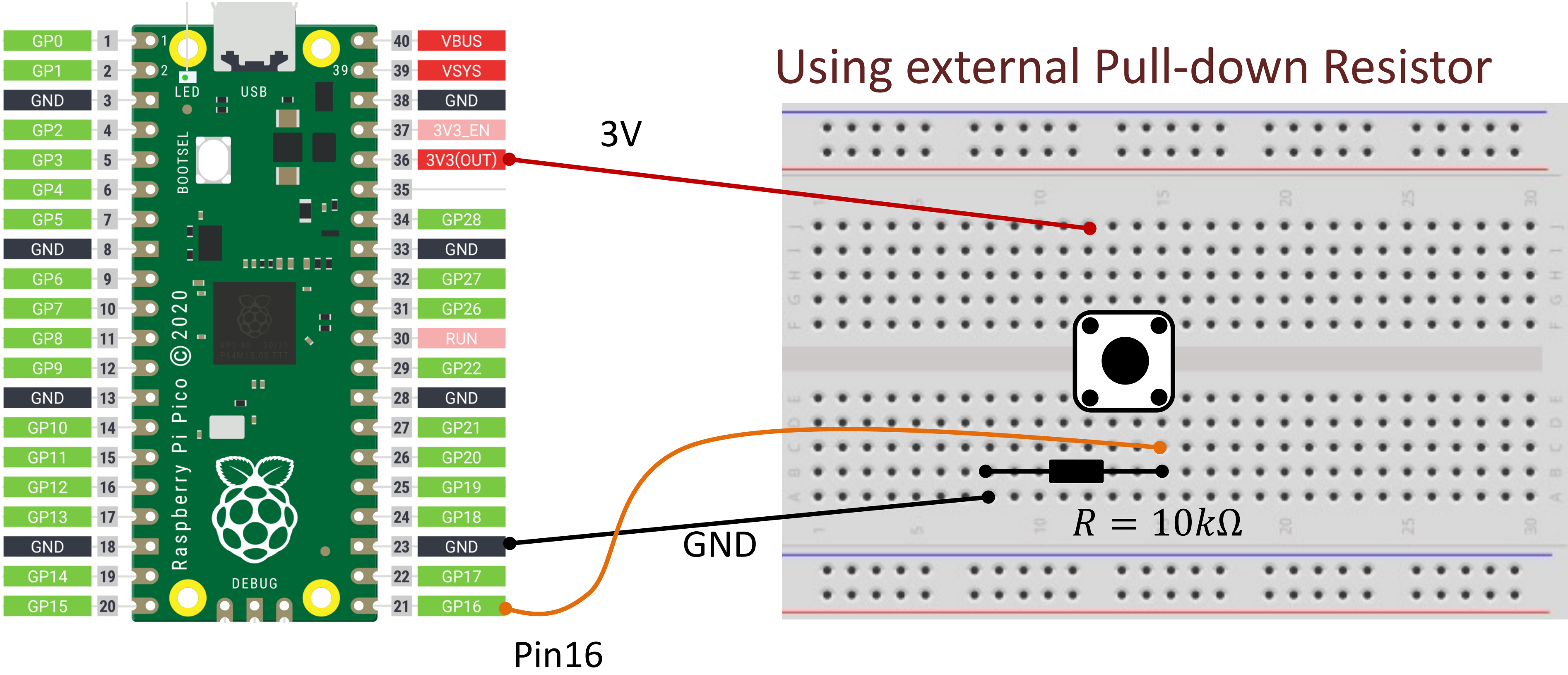
Button is Pushed => True/High

Push Button and LED

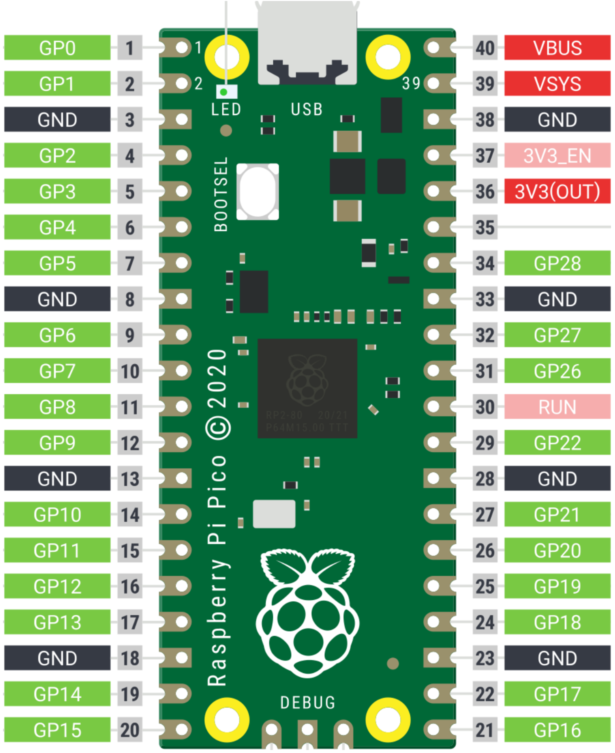
Table of Contents

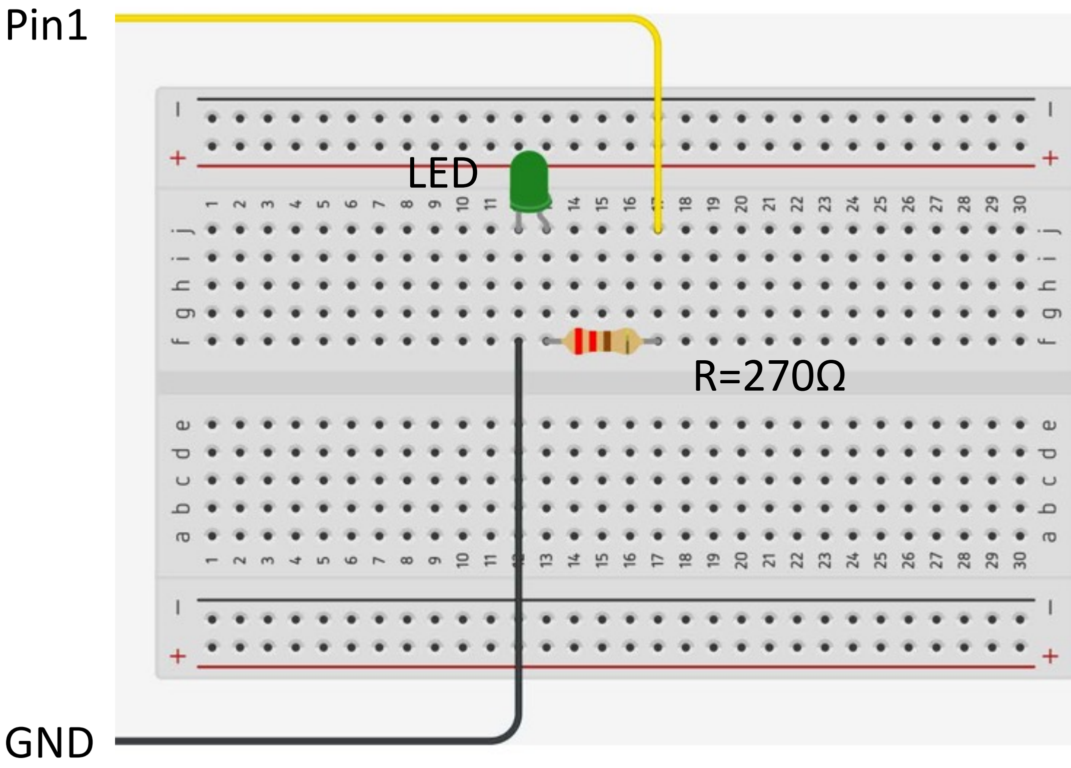
Button (Pull-down Resistor)

Using external Pull-down Resistor



Wiring LED


GP0 1
GP1 2
GND 3
GP2 4
GP3 5
GP4 6
GP5 7
GND 8
GP6 9
GP7 10
GP8 11
GP9 12
GND 13
GP10 14
GP11 15
GP12 16
GP13 17
GND 18
GP14 19
GP15 20
40 VBUS
39 VSYS
38 GND
37 3V3_EN
36 3V3(OUT)
35
34 GP28
33 GND
32 GP27
31 GP26
30 RUN
29 GP22
28 GND
27 GP21
26 GP20
25 GP19
24 GP18
23 GND
22 GP17
21 GP16
LED
USB
BOOTSEL
Raspberry Pi Pico ©2020
DEBUG


Pin1
LED
R=270Ω
GND

Python

This basic program turns the LED on when you push the button

```
from machine import Pin
from time import sleep

pin = 1
led = Pin(pin, Pin.OUT)

pin = 16
button = Pin(pin, Pin.IN, Pin.PULL_DOWN)

while True:
    if button.value():
        print("Button is Pushed")
        led.value(1)
    else:
        print("Please Push the Button")
        led.value(0)
    sleep(1)
```

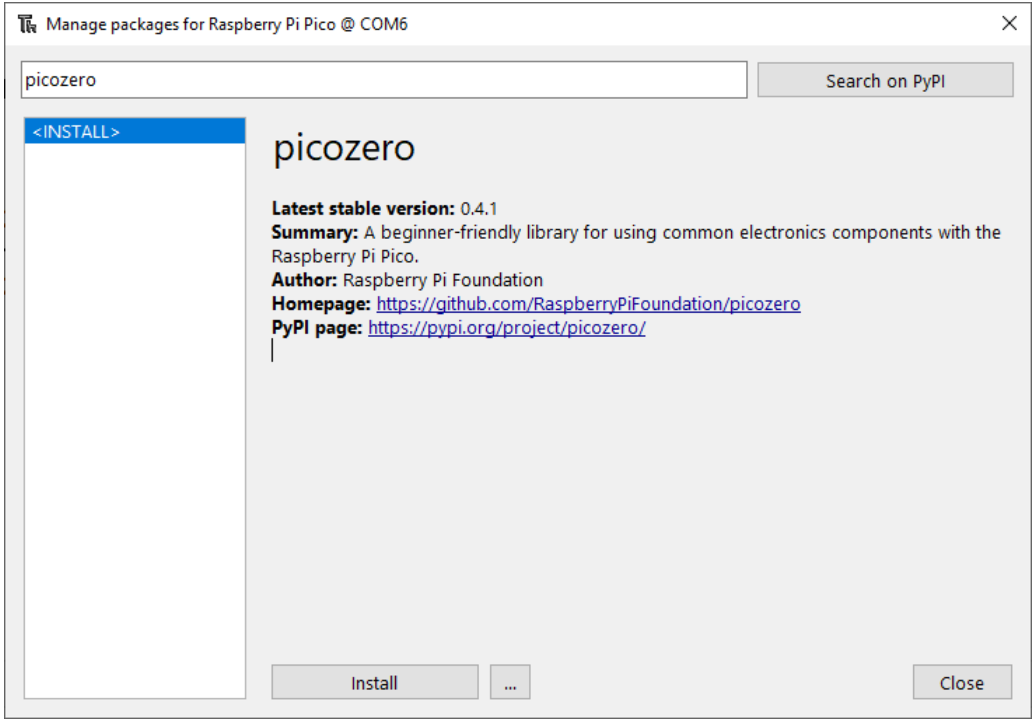
PicoZero

Hans-Petter Halvorsen

Table of Contents

PicoZero

- The **picozero** Python Library is intended to be a beginner-friendly library for using common electronics components with the Raspberry Pi Pico
- It can be used instead of the machine Library in many cases
- You install it like an ordinary Python Library using “pip install picozero” or from the “Manage Packages” window in the Thonny editor



picozero

Latest stable version: 0.4.1
Summary: A beginner-friendly library for using common electronics components with the Raspberry Pi Pico.
Author: Raspberry Pi Foundation
Homepage: https://github.com/RaspberryPiFoundation/picozero
PyPI page: https://pypi.org/project/picozero/

https://pypi.org/project/picozero/
https://picozero.readthedocs.io
https://github.com/RaspberryPiFoundation/picozero

Picozero + Push Button

https://picozero.readthedocs.io/en/latest/api.html#button

Button + LED

```
from picozero import LED, Button
from time import sleep

ledpin = 1
led = LED(ledpin)

buttonpin = 16
button = Button(buttonpin, pull_up=False)

while True:
    if button.is_pressed:
        print("Button is Pushed")
        led.on()
    else:
        print("Please Push the Button")
        led.off()
    sleep(1)
```

This basic program turns the LED on when you push the button

This examples uses the **picozero** library instead of the machine library

Raspberry Pi Pico Resources

- Raspberry Pi Pico:
  https://www.raspberrypi.com/products/raspberry-pi-pico/
- Raspberry Pi Foundation:
  https://projects.raspberrypi.org/en/projects?hardware[]=pico
- Getting Started with Pico:
  https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico
- MicroPython:
  https://docs.micropython.org/en/latest/index.html

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog