

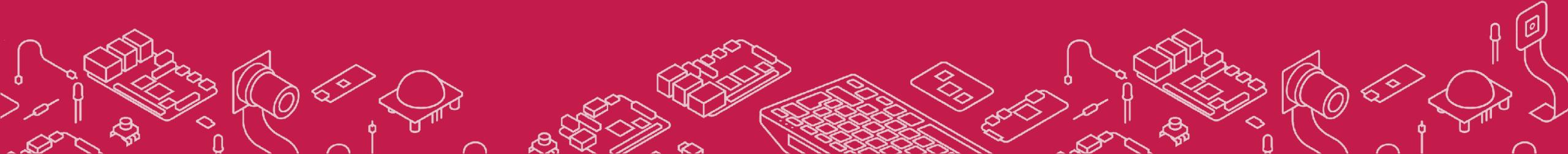
Internet of Things with Raspberry Pi Pico

June 2024 in Lagos, Nigeria - Day One



Wi-Fi: Nordic Hotel Lagos

Password: Nordic.21.0slo



Welcome!

A few words of introduction before we get started

Overview

Here's what to expect over the next two days

- Learning
- Getting to know each other
- Getting hands-on with hardware and software
- Making something

Overview

Here's what we expect from you

- Participation
- Helping each other out
- No fear of failure
- Seeking help from facilitators
- Patience and feedback

Overview

Here's what to expect over the next two days

Day 1:

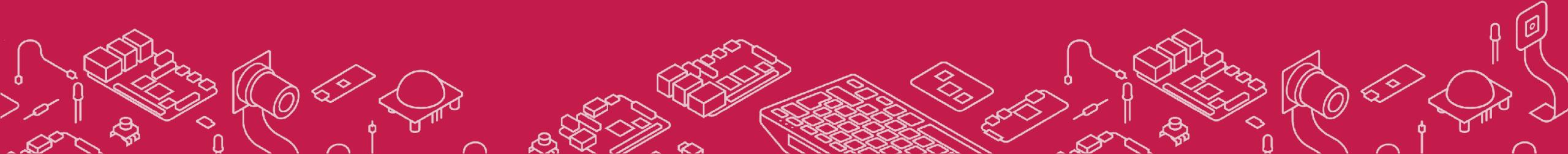
- Raspberry Pi Pico basics
- MicroPython basics
- Digital input and output
- Sensors

Day 2:

- Cloud
- **Making**
- Sharing
- Certificates (and badge!)

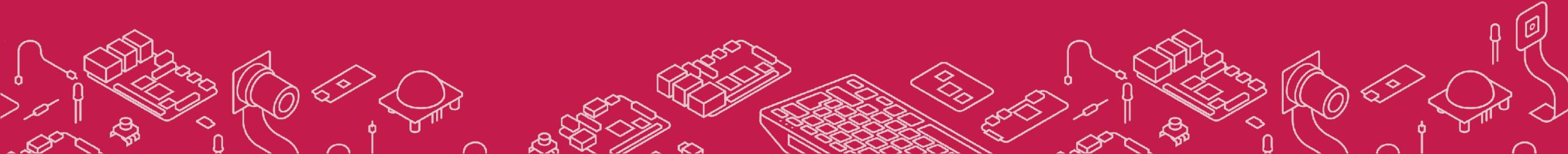
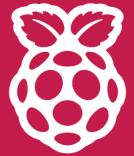
throughout: plenty of breaks

It's time for a lightning talk!



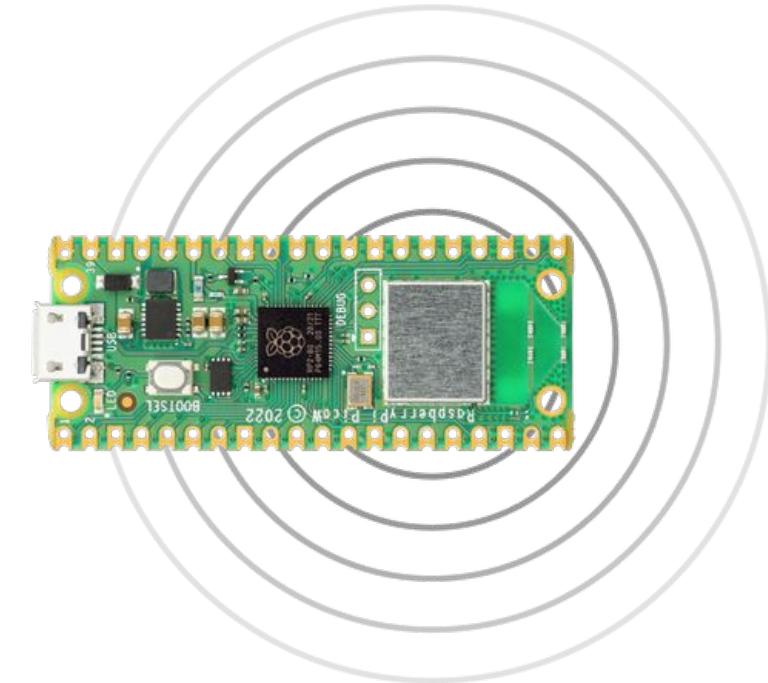
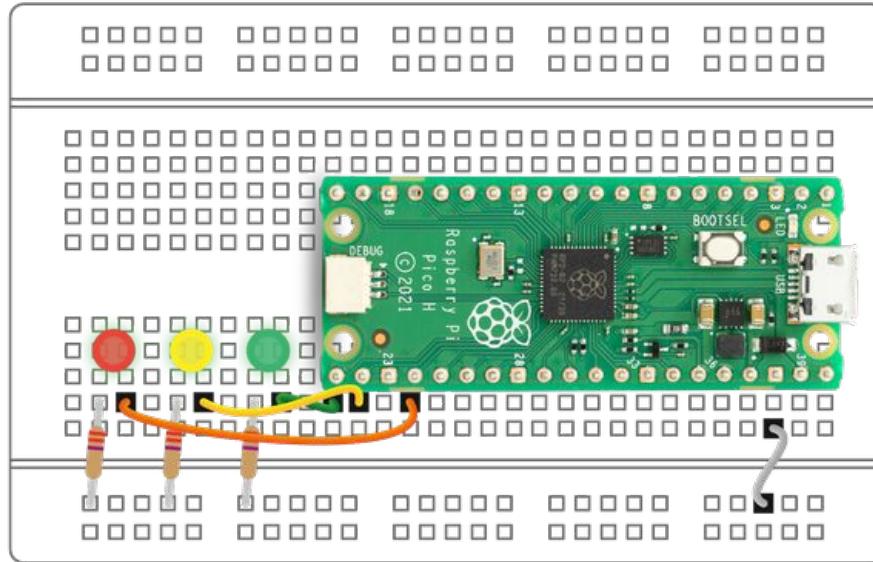
Technology Overview

IoT, microcontrollers, MicroPython, Thonny



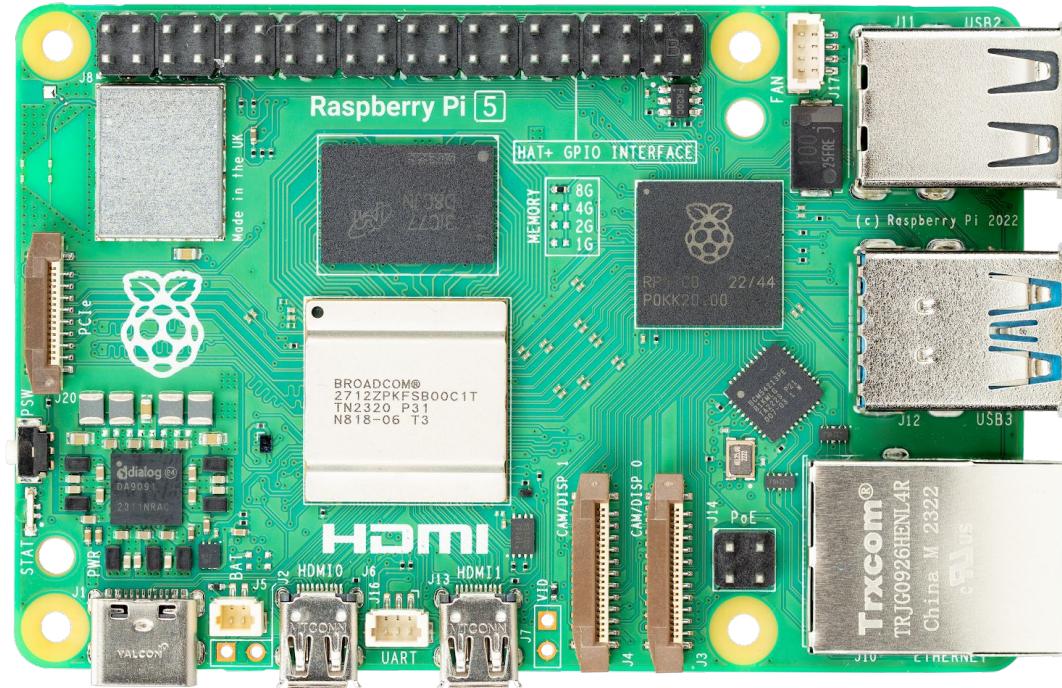
The Internet of Things

What is it, how is it used, why are we learning about it?



Microcontrollers and computers

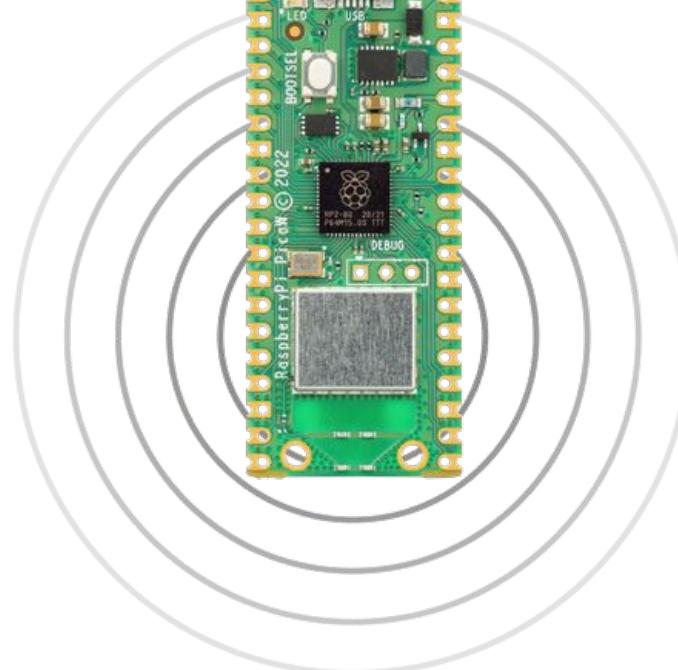
What is the difference between them?



Raspberry Pi Pico and RP2040

What is the difference between them?

Raspberry Pi Pico



W variant



RP2040

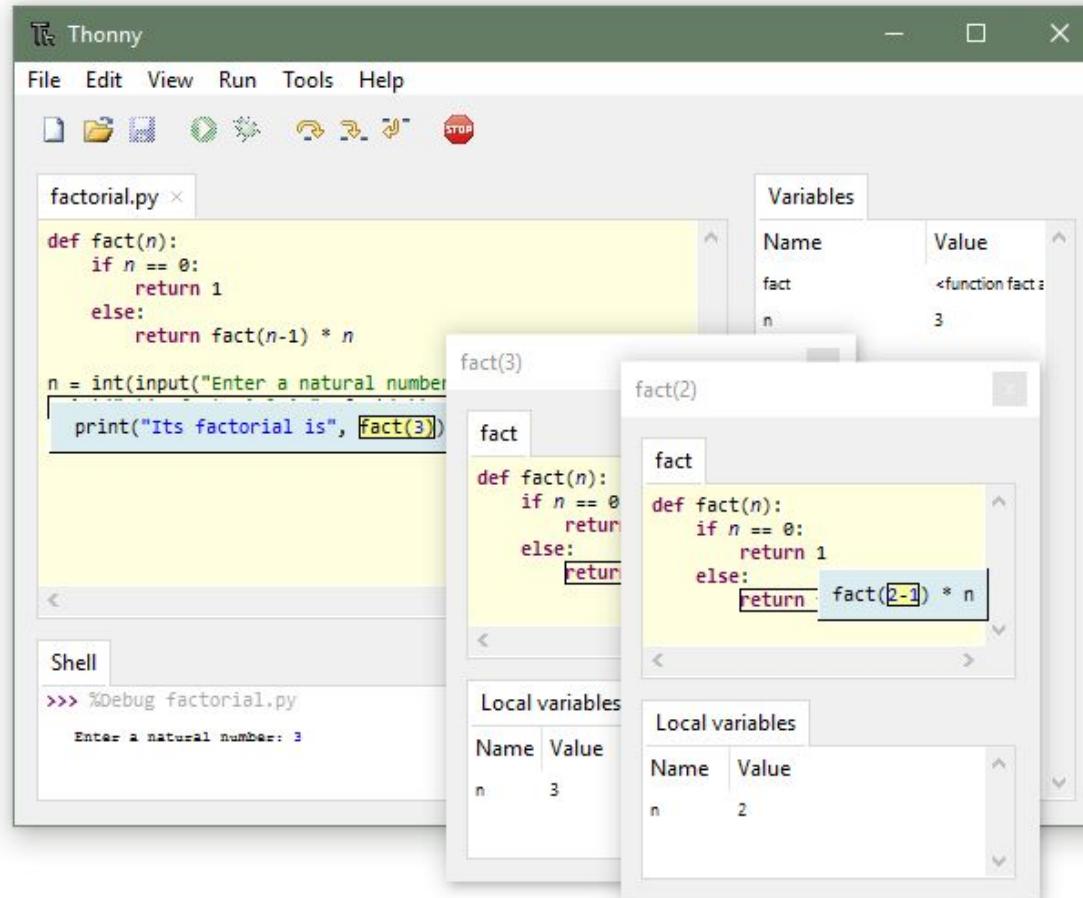
A word about programming languages



C / C++

Thonny

A Python integrated development environment (IDE)



<https://thonny.org/>



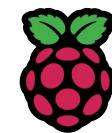
Prototyping components

What's in your kit?

- Raspberry Pi Pico WH
- USB Cable
- Breadboard
- Assorted LEDs
- Buttons
- Assorted resistors
- Jumper wires
- Pico reference sheet (pinout)

Sensors:

- AHT10 - humidity and temperature
- HC-SR501 - motion
- HC-SR04 - distance



 *swift*berry

The logo features the word "swift" in a blue, italicized sans-serif font, followed by "berry" in a larger, gray, italicized sans-serif font. To the left of "swift", there is a small icon consisting of three horizontal bars of increasing length from left to right, with a small vertical bar above the middle bar.

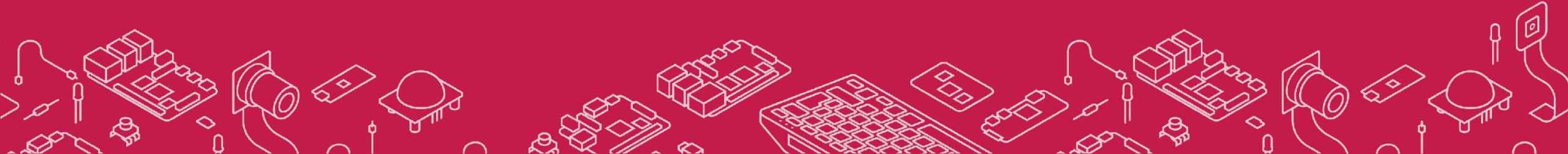
bit.ly/iot-pico-examples



Wi-Fi: Nordic Hotel Lagos
Password: Nordic.21.Oslo

MicroPython and Raspberry Pico basics

It's time for "Hello, World!"



Installing MicroPython on your Raspberry Pi Pico

Thonny makes it easy

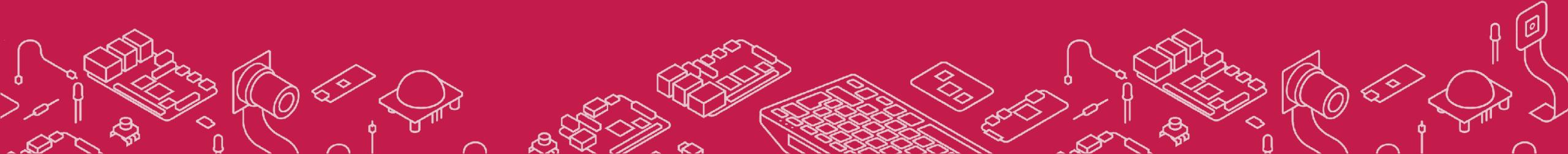
- Hold down the **BOOTSEL** button while you plug in your Raspberry Pi Pico
- Go to the **Tools** menu and click **Options**
- Go to the **Interpreter** tab.
- Click the link in the bottom right which says “**Install or update MicroPython**”
- For **Target volume**, choose whichever includes RPI-RP2
- For **variant**, choose Pico W
- For **version**, choose 1.20.0
- Click **Install**

MicroPython basics

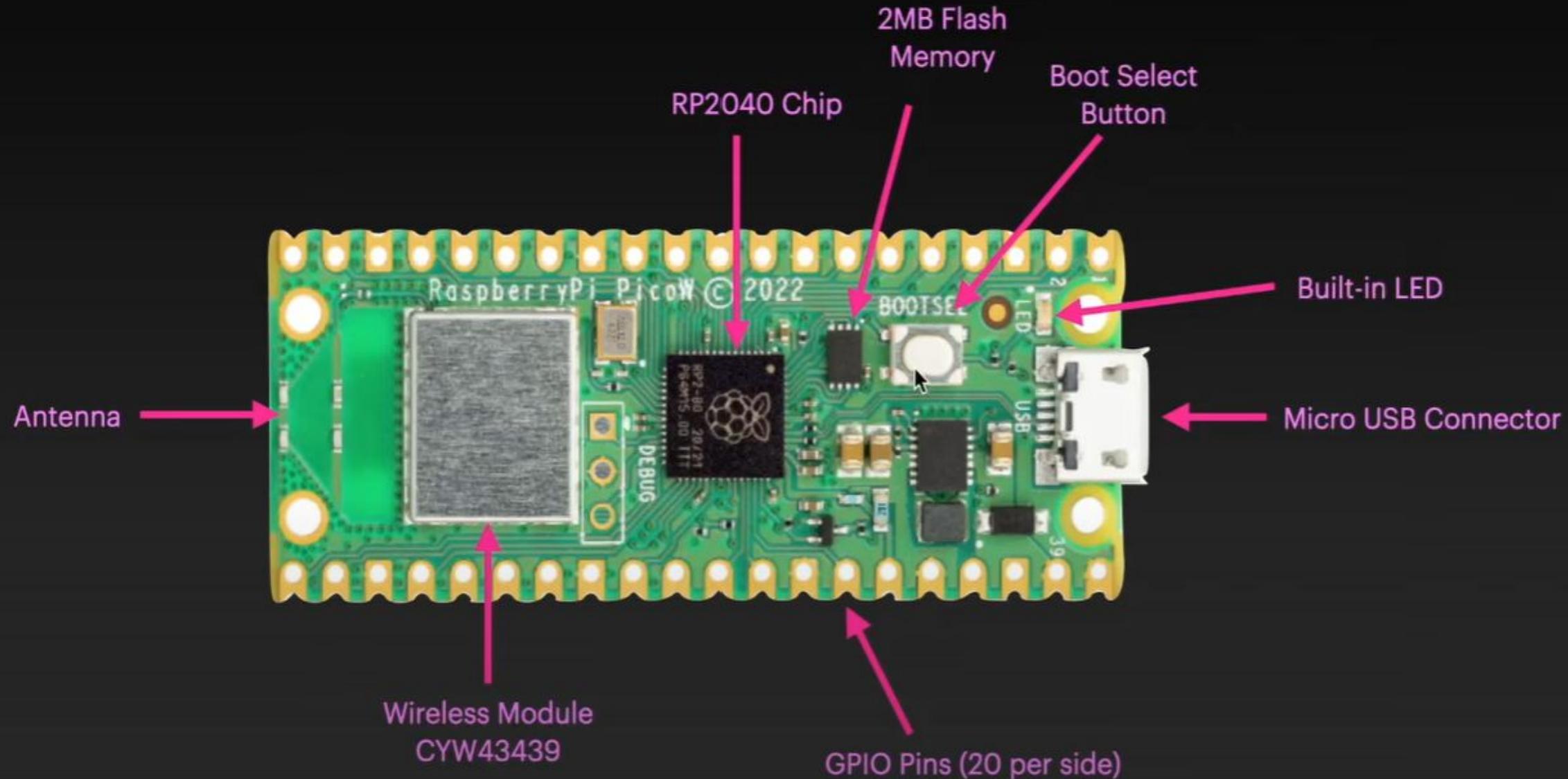
It's time for a live demo with Thonny

Digital input and output

It's time to start wiring some stuff up!

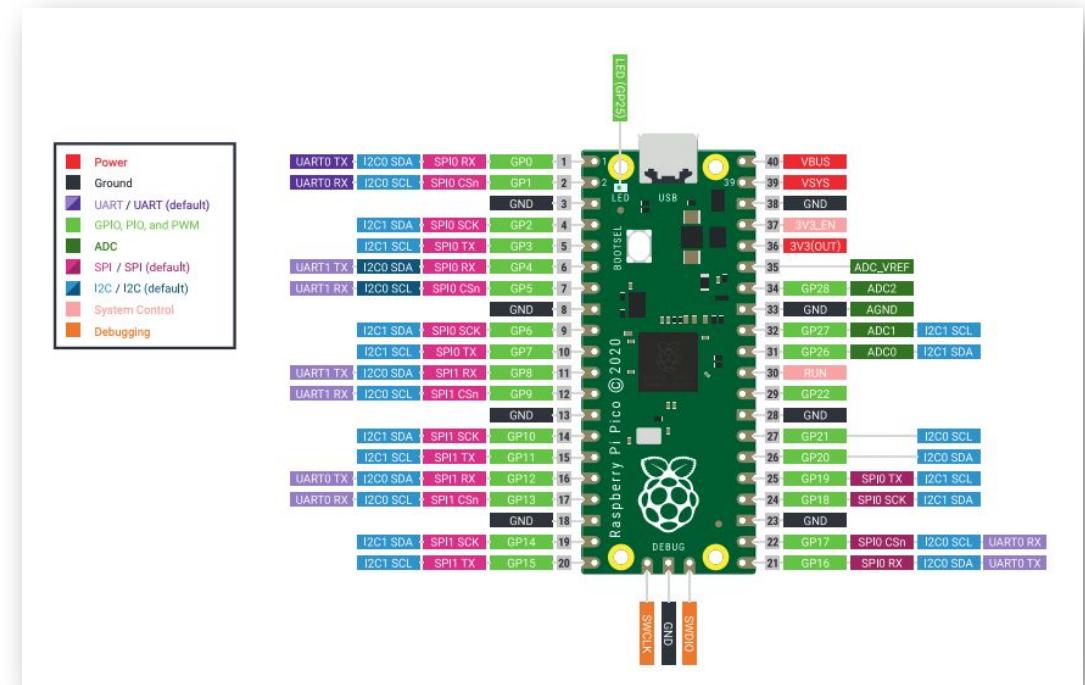
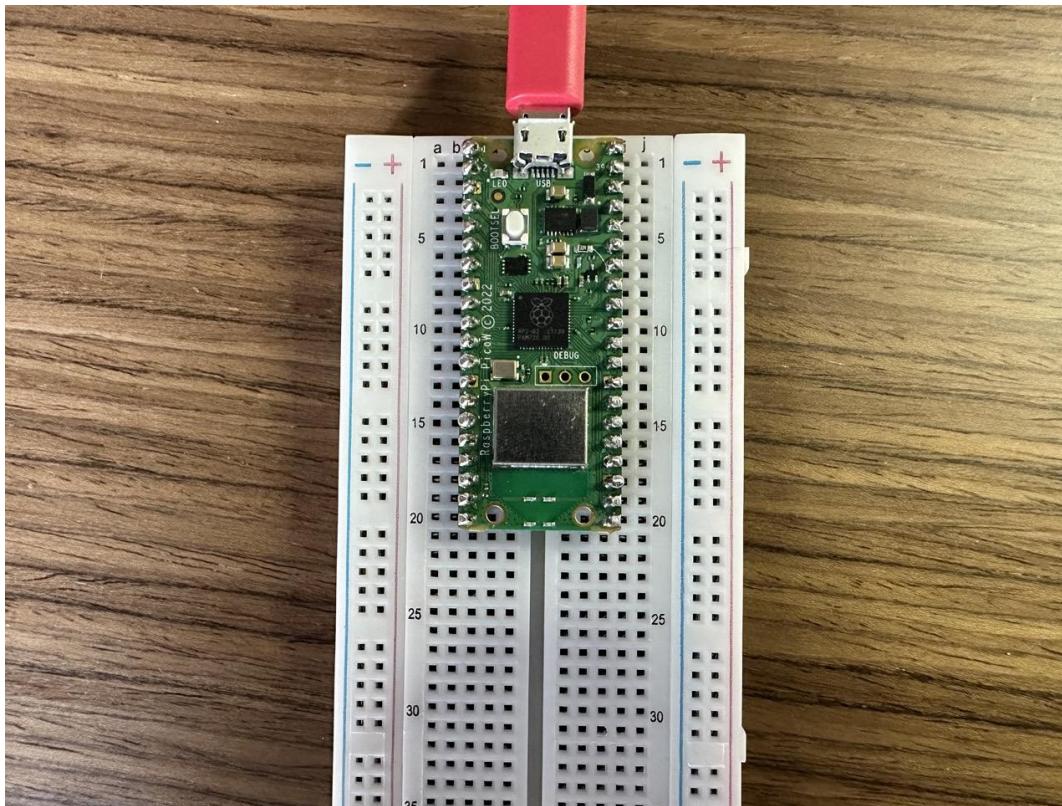


Raspberry Pi Pico W and pin basics



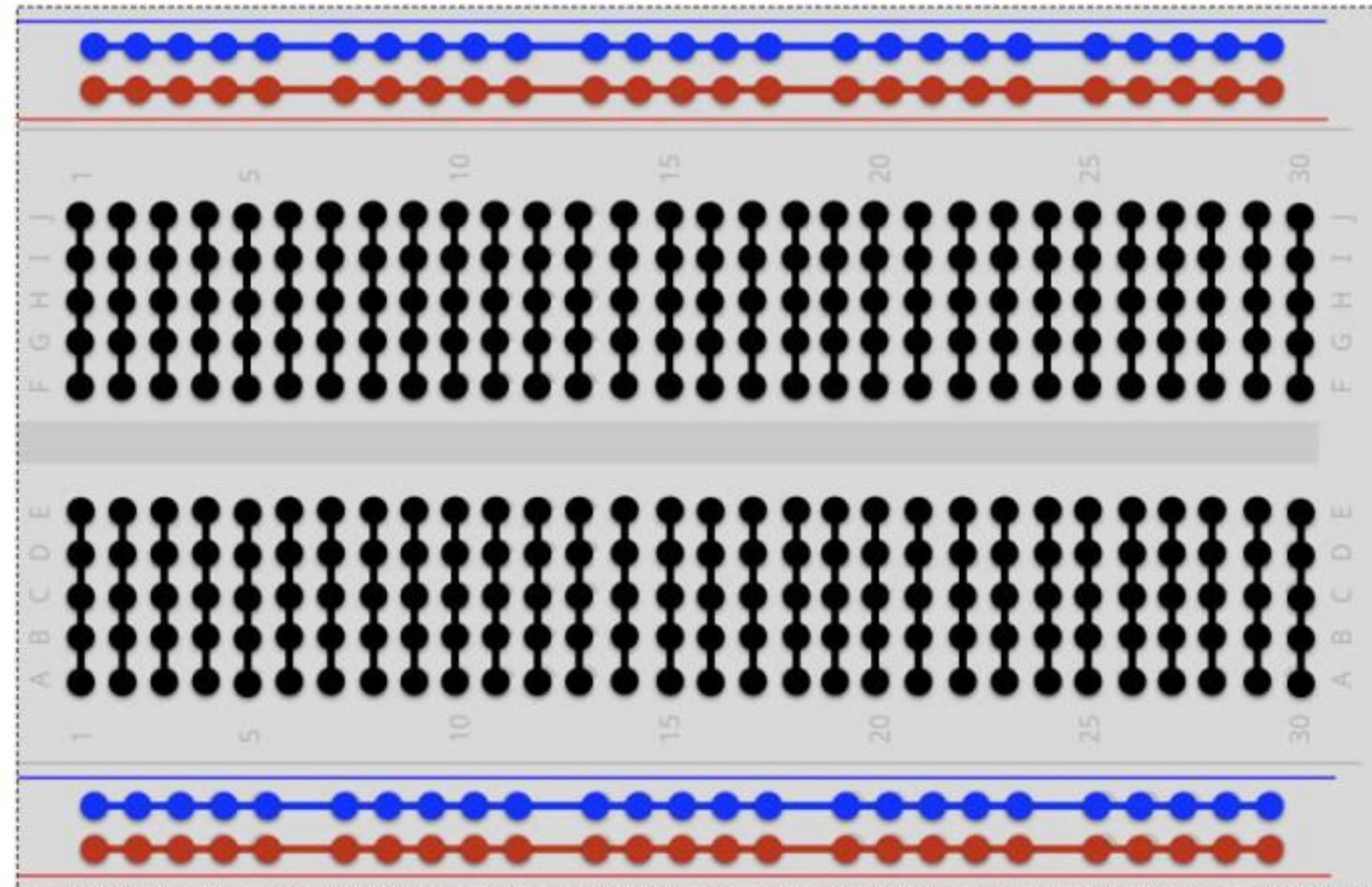
Breadboard and pin basics

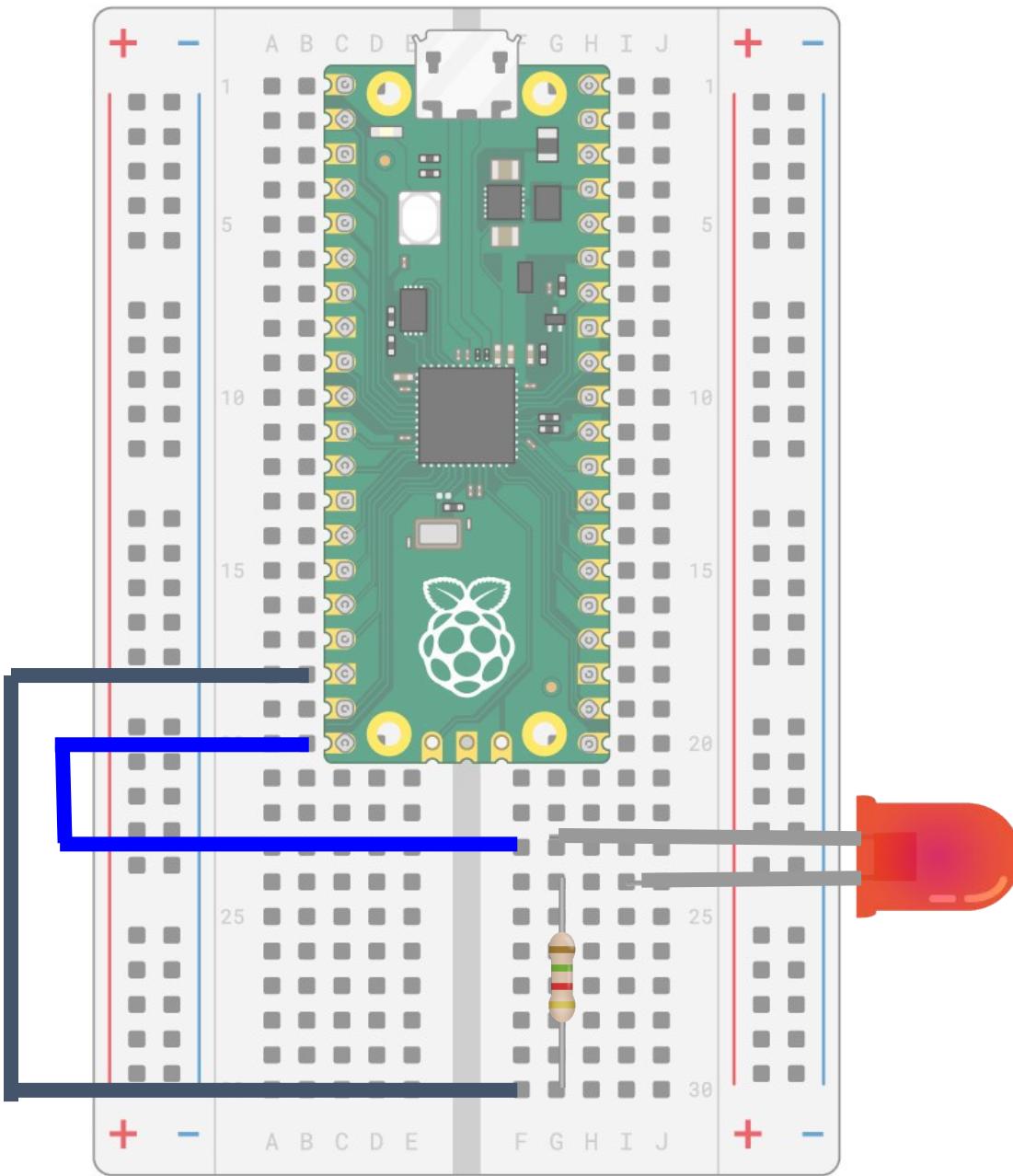
The breadboard will make prototyping much easier



Breadboard and pin basics

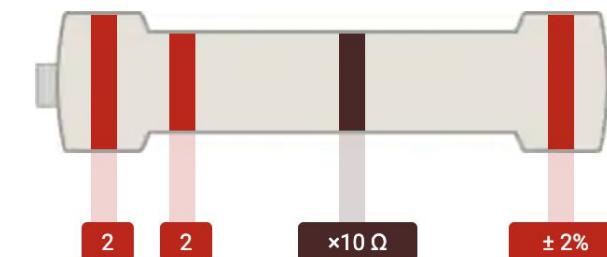
The breadboard will make prototyping much easier



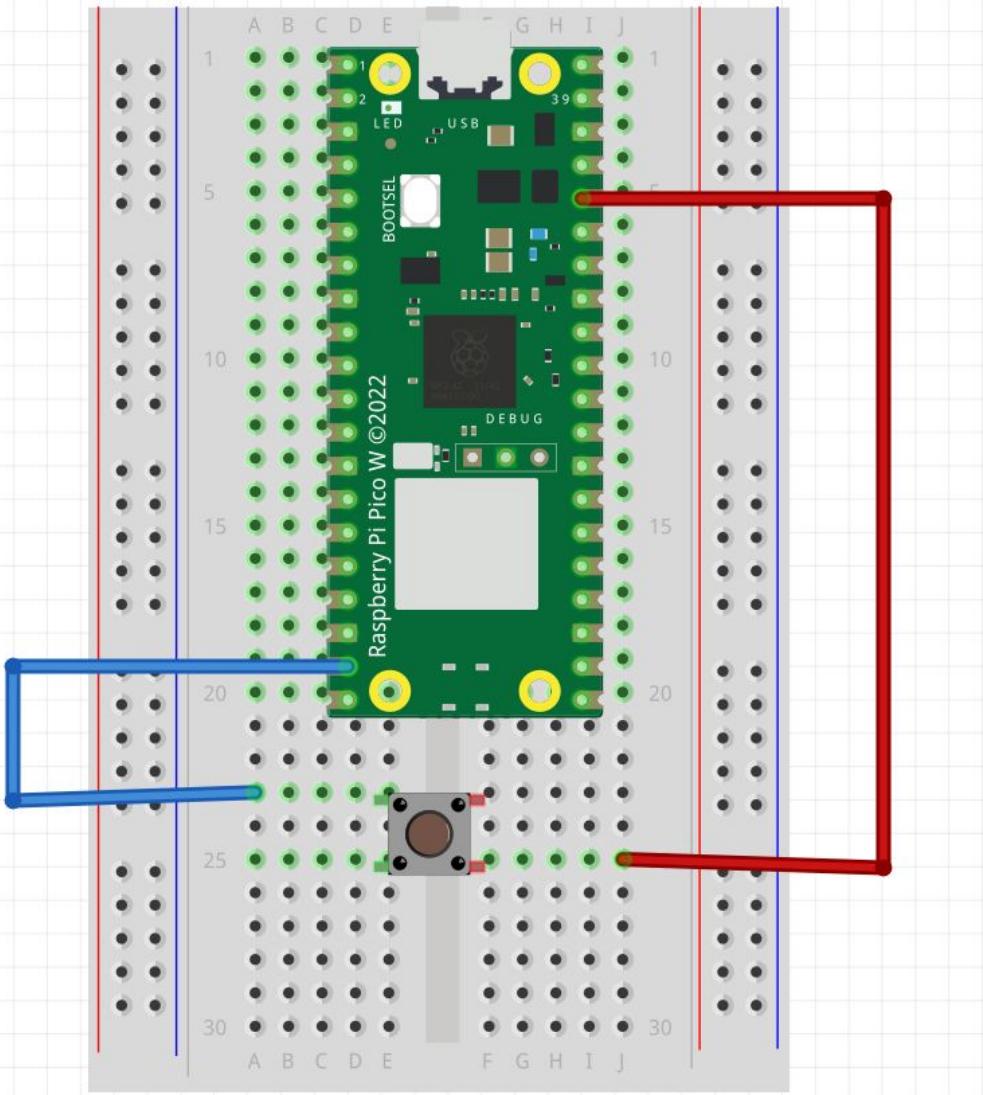


Wiring up an LED (digital output)

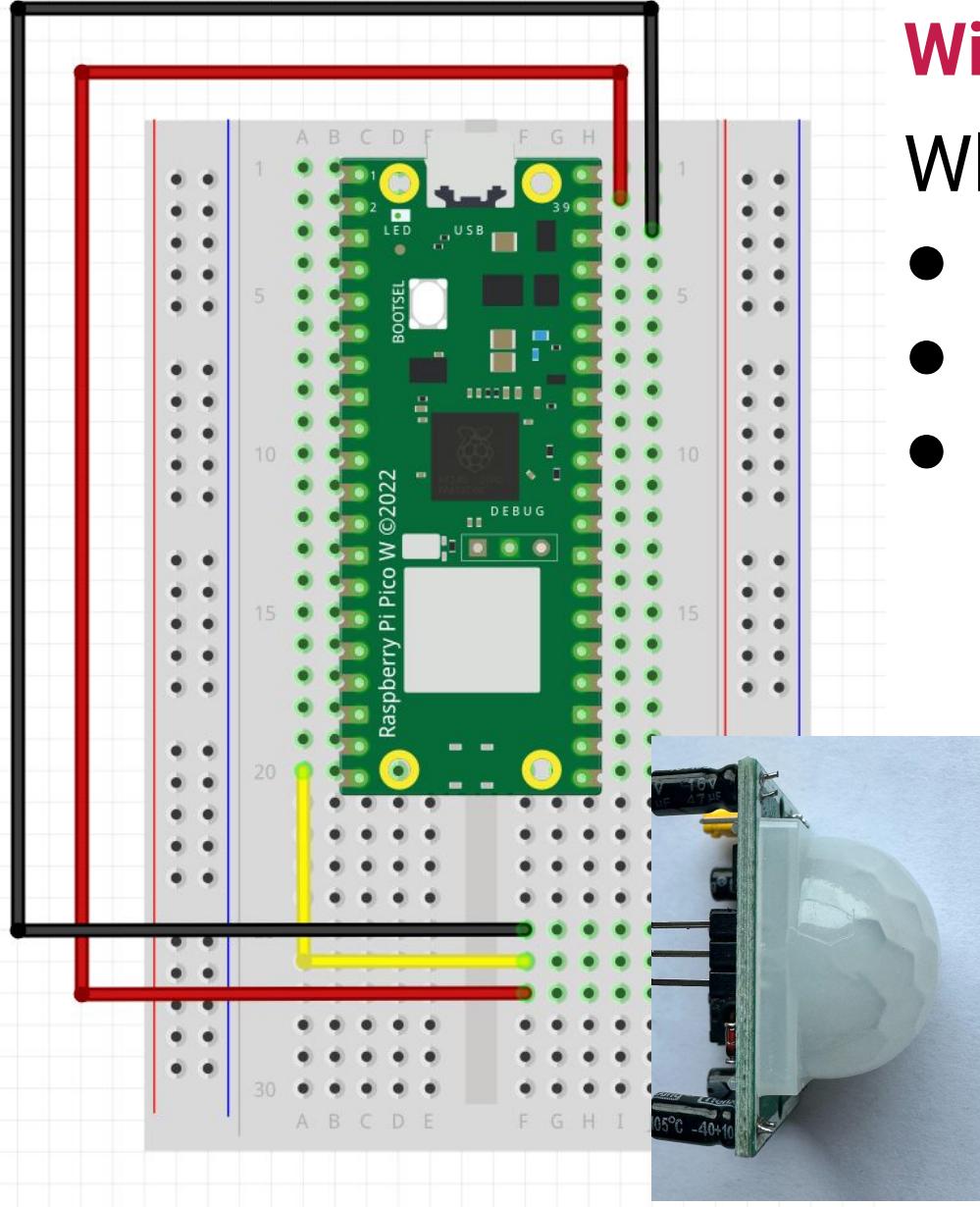
- The long leg of the LED connects to GPIO 15
- The short leg of the LED connects to a 220Ω resistor
- The other side of the resistor connects to GND



Wiring up a button (digital input)



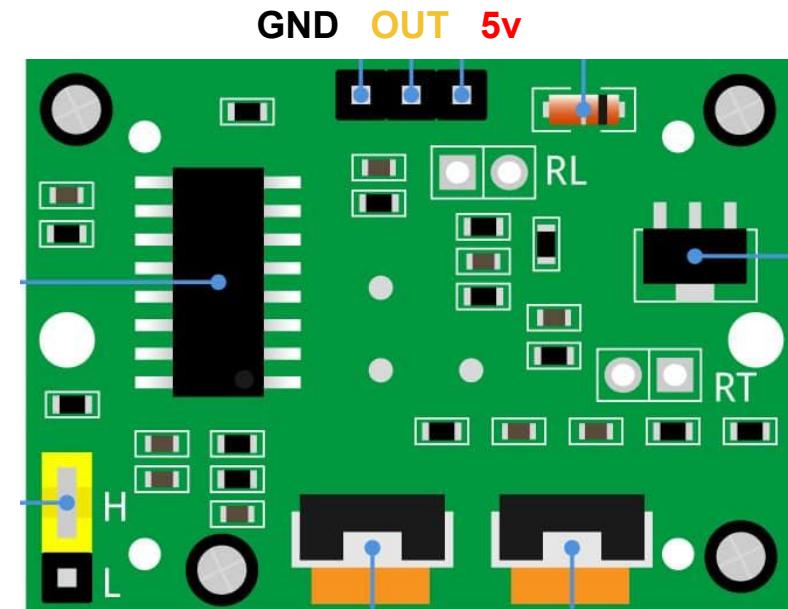
- One side of the button connects to **3v3**
- The others side of the button connect to **GPIO 14**



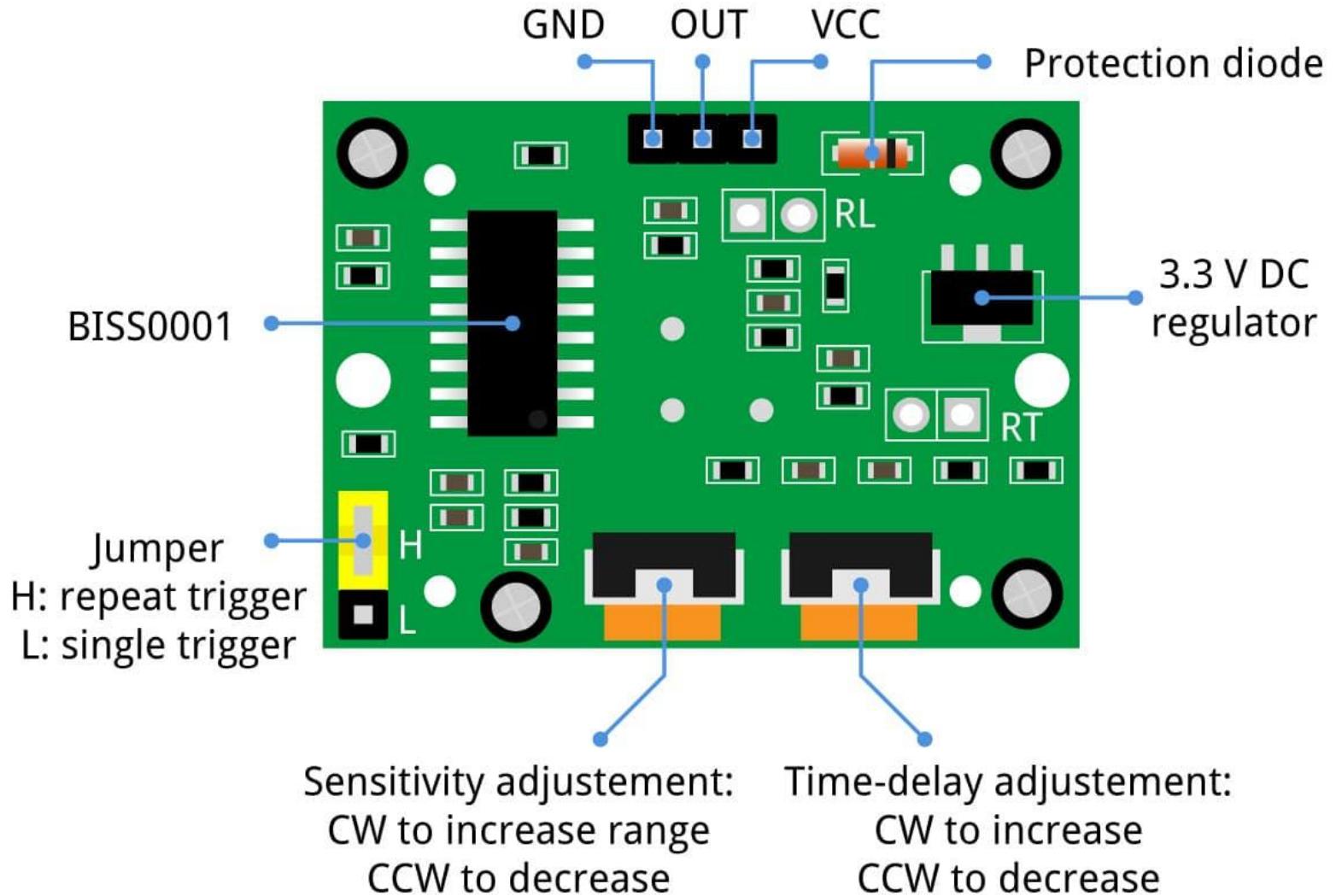
Wiring up a motion sensor (HC-SR501)

When orienting the sensor as shown:

- The top pin connects to **GND**
- The middle pin connects to **GP15**
- The bottom pin connects to **VSYS**

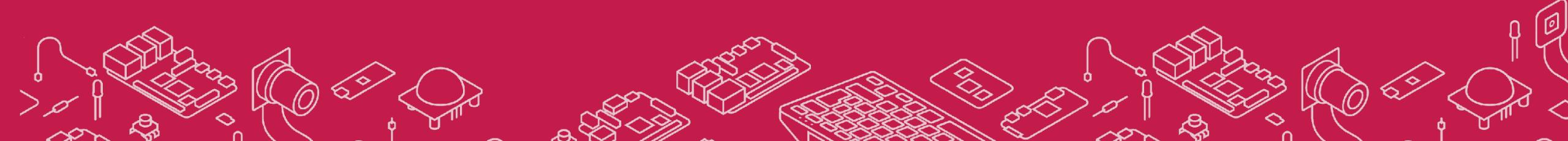


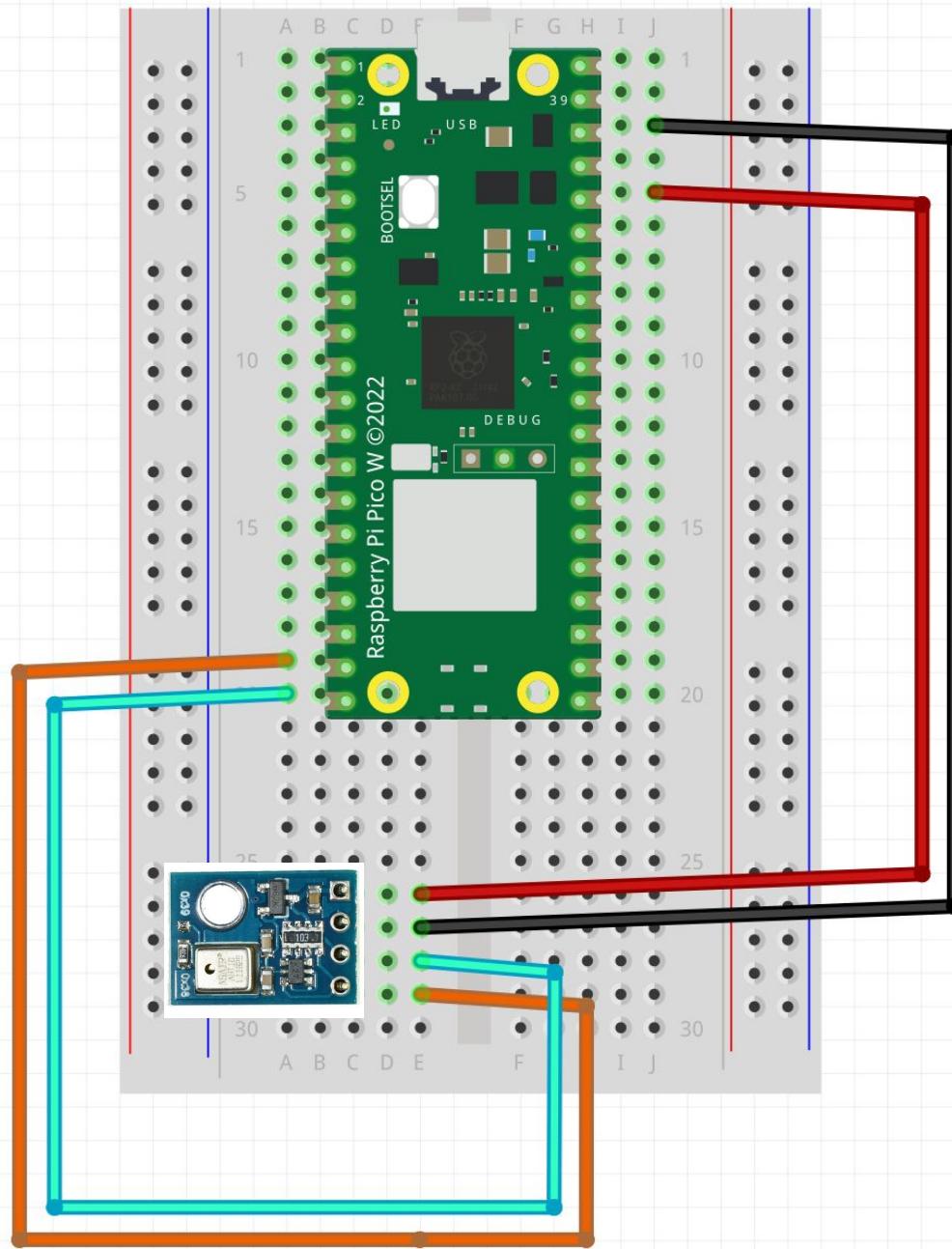
More information about motion sensor HC-SR501



Using digital sensors with i2c

Wiring up a temperature and humidity sensor





Wiring up an AHT10 sensor

- AHT10's VIN connects to Pico's **3v3 OUT**
- AHT10's GND connects to Pico's **GND**
- AHT10's SCL connects to Pico's **GP15 (I2C1 SCL)**
- AHT10's SDA connects to Pico's **GP14 (I2C1 SDA)**

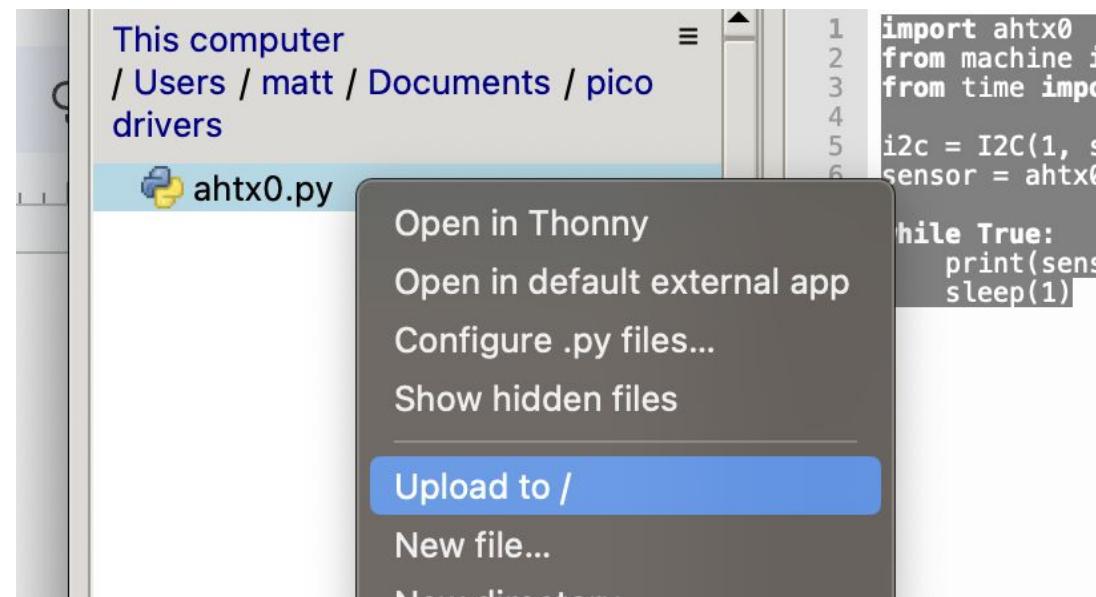


bottom of AHT10

Download the AHTX0 MicroPython driver and save it on your Pico

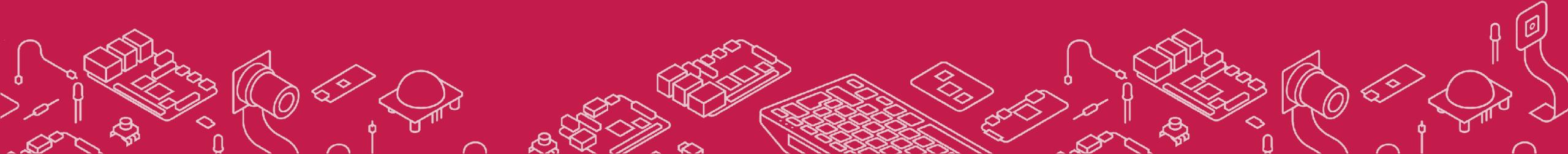
Someone did the hard work for us already!

- Download the driver from bit.ly/aht10pico
- Save that file as **ahtx0.py** on your Pico
 - option one: copy the text into a new file in Thonny and then save it
 - option two: download, find it in Thonny, and then right click it to upload it to the Pico

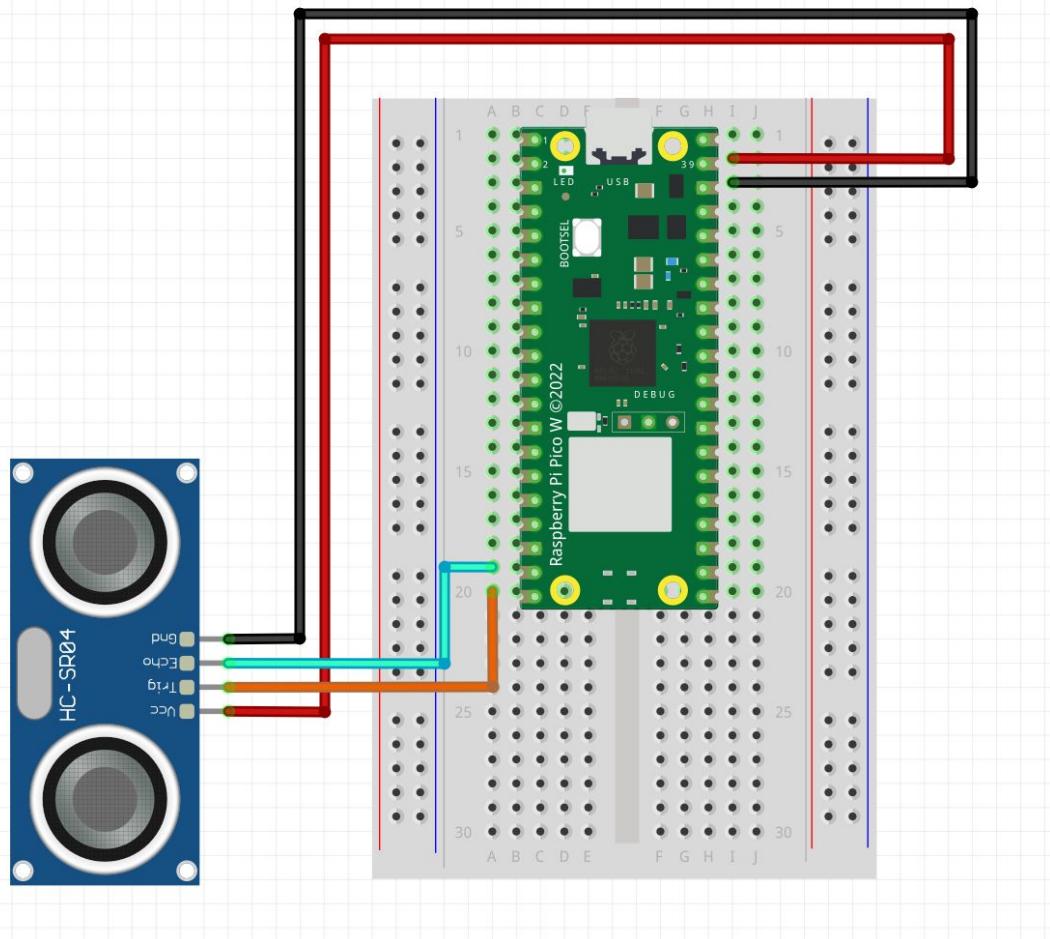


Using an ultrasonic distance sensor

Using the HC-SR04 with GPIO



Wiring up an HC-SR04

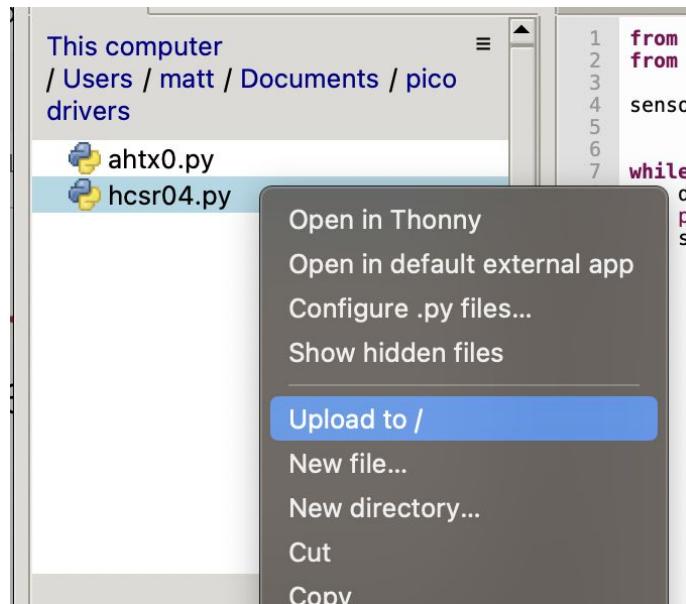


- HC-SR04's VCC connects to Pico's **VSYS**
- HC-SR04's GND connects to Pico's **GND**
- HC-SR04's ECHO connects to Pico's **GP14**
- HC-SR04's TRIGGER connects to Pico's **GP15**

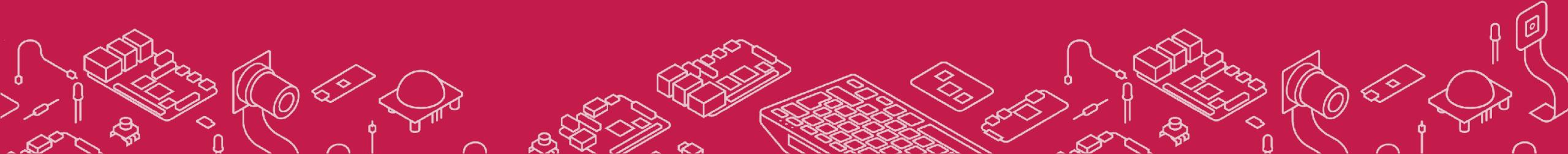
Download the HC-SR04 MicroPython driver and save it on your Pico

Someone did the hard work for us already!

- Download the driver from bit.ly/hcsr04pico
- Save that file as **hcsr04.py** on your Pico
 - option one: copy the text into a new file in Thonny and then save it
 - option two: download, find it in Thonny, and then right click it to upload it to the Pico



Connecting to Wi-Fi and installing MQTT library



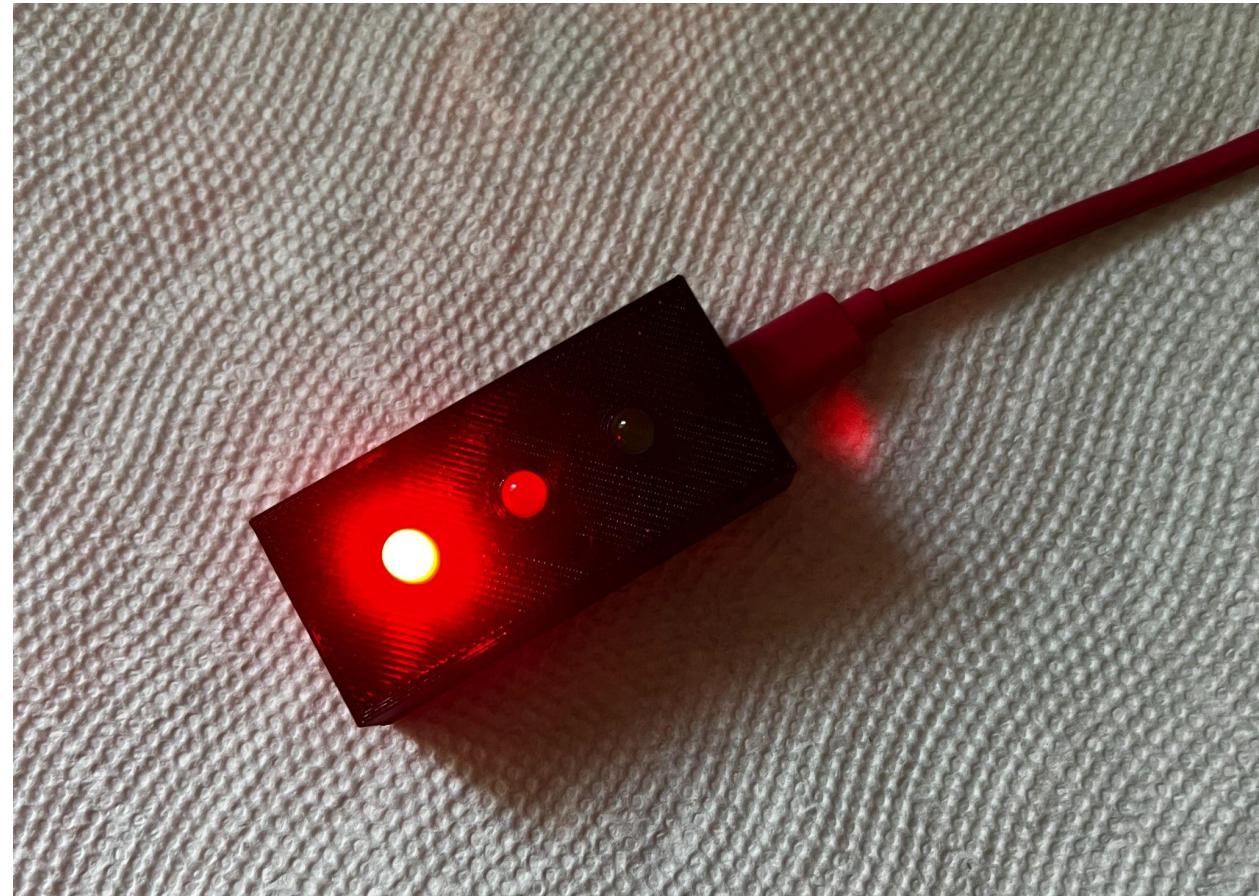
What is MQTT?

Before we share, what do you know?

- It's a messaging protocol
- It's used heavily for Internet of Things and M2M
- It uses a publish/subscribe model with "topics" as feeds
- It can be used on the LAN or via Internet
- It's designed to work on "lightweight" devices like Raspberry Pi Pico
- An MQTT server is called a "broker"
- There are lots of options for client and broker software
- We're going to use `umqtt-simple`, a client library for MicroPython

Garage / Shed indicator project

My MQTT project example

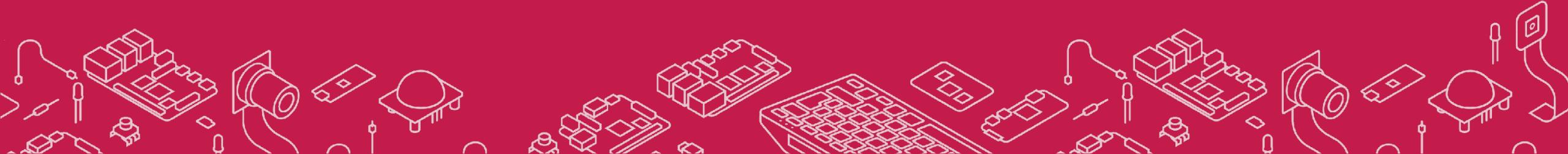


Connect to WiFi and download MQTT library

- Get the code from bit.ly/pico-iot-wifi
- Paste it into a new Thonny file
- Edit the wifi network info to your phone hotspot or:
 - ssid = 'Nordic Hotel Lagos'
 - password = 'Nordic.21.Oslo'
- Save the file on the Pico as **wificonnect.py**
- In the console (">>>>"), type:

```
import wificonnect
import umqtt
umqtt.simple.install("umqtt.simple")
```
- If it works, **import umqtt.simple** should run without errors.

Connecting to Adafruit IO



Create an Adafruit IO account and a new feed (aka topic)

- Go to <https://io.adafruit.com/>
- Create a free account
- After logging into your account, go to feeds and create a new feed.
- Give it a very simple name, like temperature
- Click the name of the feed and then click feed info to see the MQTT endpoint. This will be your MQTT topic.
- Close that window and click the yellow key at the top.
 - Adafruit IO Username == umqtt user
 - Adafruit IO Key == umqtt password

Get the Pico to send data to Adafruit IO

- Get the example code from
bit.ly/adafruit-io-example
- Change the required parts: username, key, and feedname
- Note: feedname will always be in the pattern:
[username] / feeds / [feedname]
- The code transmits the value 10 every 20 seconds. Now update it to use sensor data of your choice.
- Watch for the changes in your Adafruit IO account.
- **Do not send more than 30 values per minute!** Your account will be suspended.