



**Raspberry Pi**

**Weather Monitor System**

<Camosun College Capstone Project>

Authors: Brendan Caldwell, Caleb Olsen, Pei-Fan Ho

Course: ENGL273 - Technical Communication

Instructor: Tom Nienhuis

Date: July 16th, 2024

# **Table Of Contents**

[**Table Of Contents 2**](#_uisqmm9bsx6n)

[**Introduction 3**](#_dwtqbm7ispkl)

[**Requirements 4**](#_lj92ygx9f12z)

[Hardware 4](#_kscu7marmf8c)

[**Instructions 5**](#_4eg8pdrqonum)

[Raspberry Pi 5](#_9zjt22h6v7cp)

[Download The Source Code 5](#_p201cgr8g3l4)

[Create the Python Virtual Environment 7](#_2l97ksrlenbd)

[Start The API Server 8](#_z6f1ofyu5s4u)

[Raspberry Pi Pico 9](#_6o8sunv4bten)

[Assembling the Pi Pico with the BME280 Sensor 9](#_63abkirqhd3o)

[Downloading Thonny 10](#_tymgxs6j5dp7)

[Installing Circuit Python on the Pi Pico 11](#_omdrey5ae70)

[Getting the Scripts onto the Pi Pico 15](#_ds4liksbwxok)

[Adjusting the Scripts for Personal Use 16](#_7agbtm7mg01g)

[**Conclusion 18**](#_hrptgd184r37)

# **Introduction**

The purpose of the Raspberry Pi Weather Station is to be able to record weather data in a remote location and display it on a web page. This product is meant for people who are looking to learn more about both the Raspberry Pi and the Raspberry Pi Pico and see what you can accomplish with them.

The scope of this manual is to show how to get the API server up and running on the Raspberry Pi. This is not meant to show how to get the other components of the project working like the web page or the Pi Pico.

Any **red and bold** text indicates a potential exception to the normal instruction that you may encounter.

The following are tasks that **are** covered in this manual:

* How to clone the Github Repository
* How to setup the Python virtual environment and install all the dependencies for the API server
* How to start the API server
* How to setup the Pi Pico

The following are tasks that are **not** covered in this manual:

* How to setup the Raspberry Pi
* How to use the command line
* How to navigate the Raspberry Pi (i.e open a web browser, open the command line, etc.)
* How to install git
* How to setup the web page

# **Requirements**

## **Hardware**

* Personal Computer (Windows Recommended)
* Raspberry Pi 3B or 4B
  + Micro SD card (32GB Minimum)
  + Micro SD card reader for personal computer
* Raspberry Pi Pico W Microcontroller
  + Micro USB adapter to connect Pi Pico to personal computer
* BME280 temperature/humidity/pressure sensor
* Breadboard with minimum 6 wires

## 

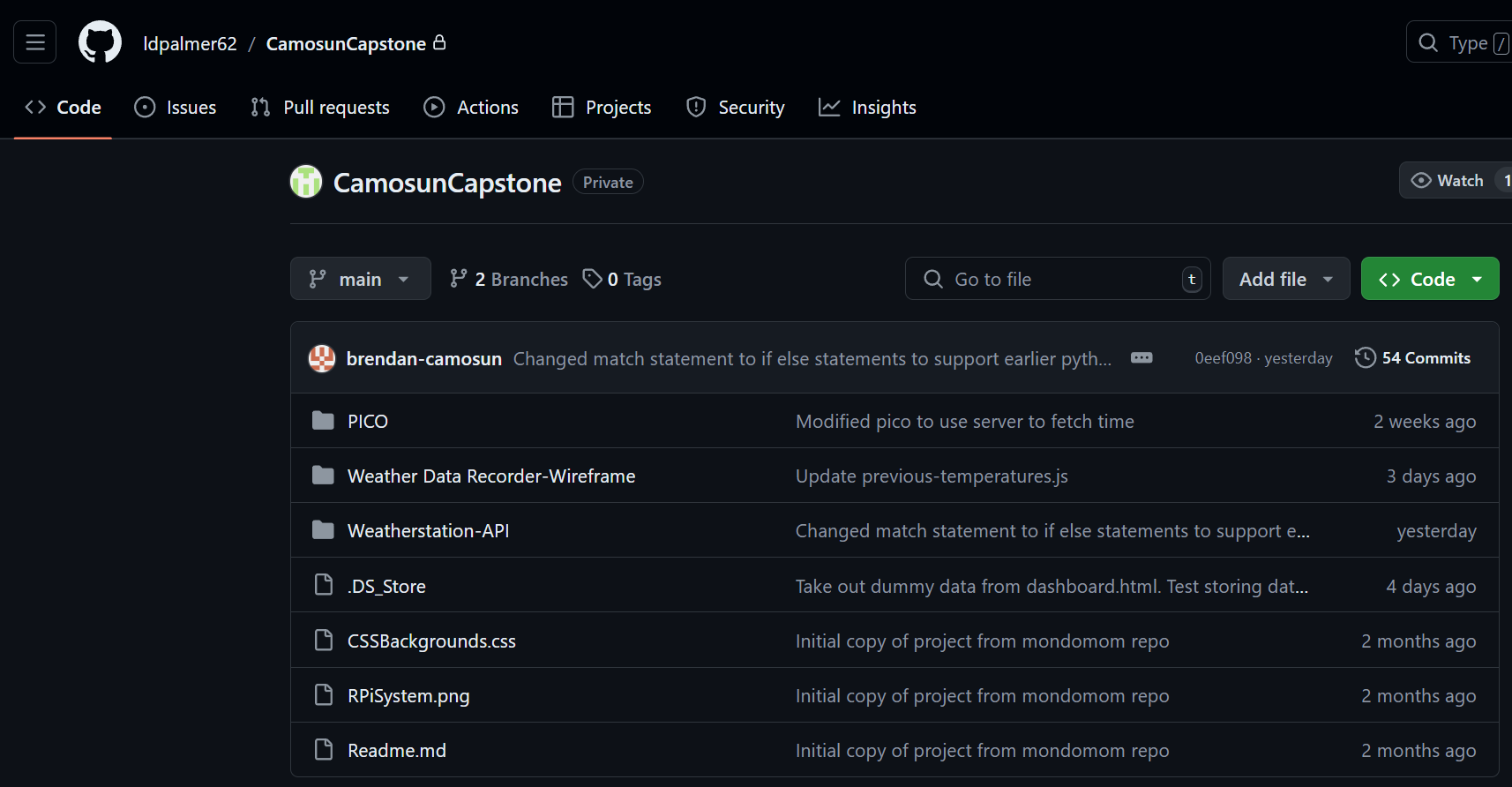
* Python installed on the Raspberry Pi
* Basic knowledge on how to use the Raspberry Pi, this includes tasks such as being able to open programs like the command line or a web browser.
* Basic knowledge of Python
* Basic knowledge on how to use the command line. Some examples include:
  + Change directories using the *cd* command
  + Activate a Python virtual environment
  + Execute a Python script
* Basic understanding of networking (i.e IP addresses and ports)

# **Instructions**

## **Raspberry Pi**

### Download The Source Code

1. Login to your Raspberry Pi
2. Open a web browser
3. Navigate to the following URL: <https://github.com/ldpalmer62/CamosunCapstone>



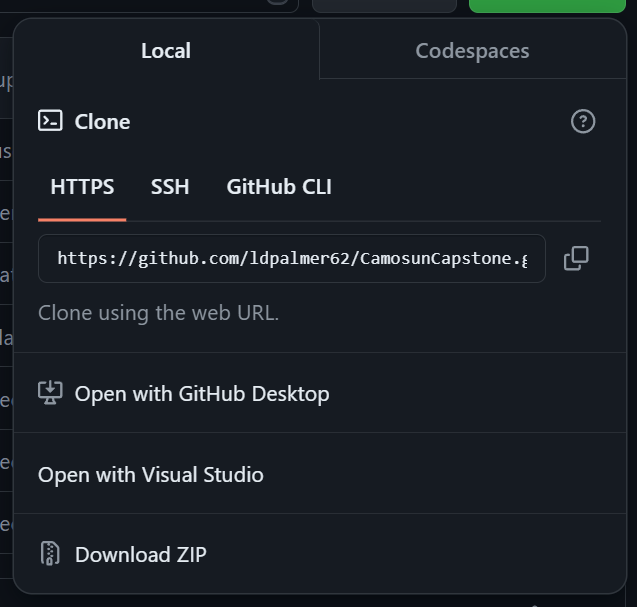
**Figure 1 - The Github repository web page**

1. Click on the green dropdown button that says *<> Code*



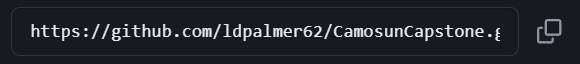
**Figure 2 - <> Code dropdown button**

1. Click on the tab that says HTTPS



**Figure 3 - HTTPS menu**

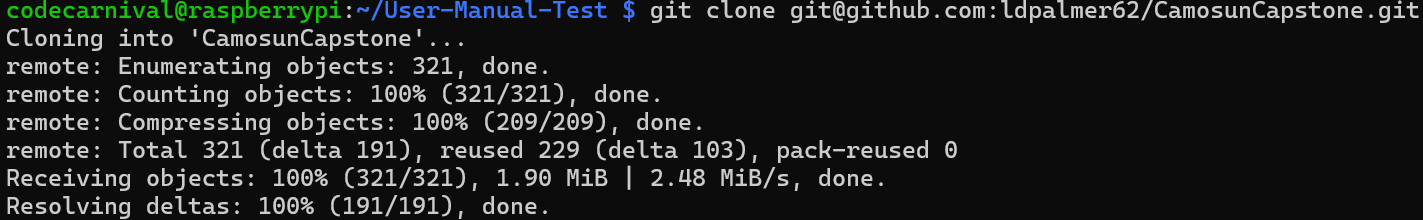
1. Copy the link displayed underneath the HTTPS tab



**Figure 4 - Link to copy**

1. Open the command line
2. Navigate to a directory where you want to download the source code
3. execute the following command:

*git clone https://github.com/ldpalmer62/CamosunCapstone.git*

**

**Figure 5 - git clone command being executed**

**Note: In the screenshot above, the URL I’m using is copied from underneath the SSH tab rather than the HTTPS tab. The reason for this is because at the time I’m making this, the repository is private, so I need to use the SSH URL to authenticate with Github.**

## 

### Create the Python Virtual Environment

1. Execute the following command: *cd ./CamosunCapstone/Weatherstation-API*



**Figure 6 - cd command example**

1. Execute the following command: *python3 -m venv ./env*



**Figure 7 - Python virtual environment creation command**

1. Execute the following command: *source ./env/bin/activate*



**Figure 8 - Python virtual environment being activated**

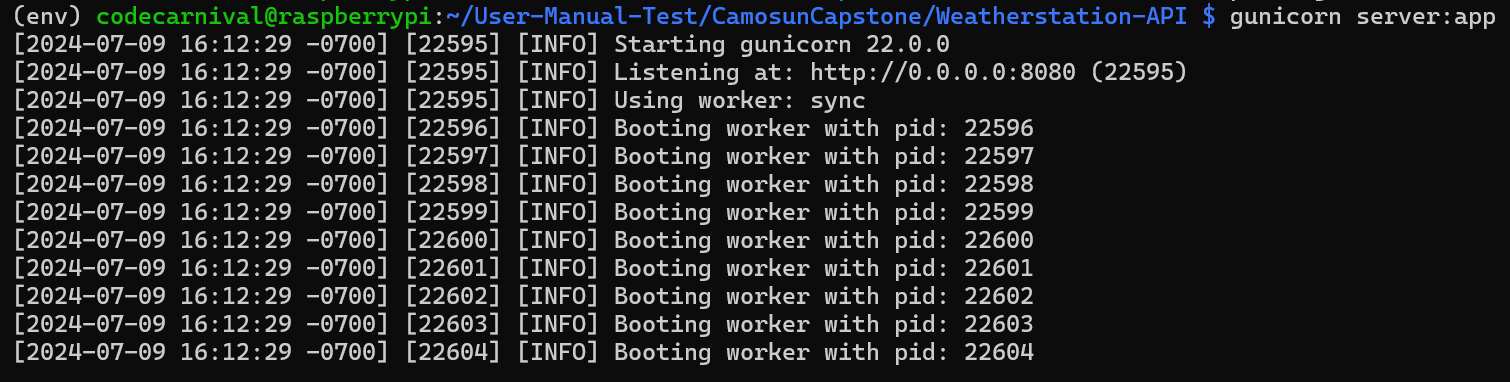
1. Execute the following command: *pip install -r ./requirements.txt*



**Figure 9 - Installation of all the dependencies for the API server**

## 

### Start The API Server

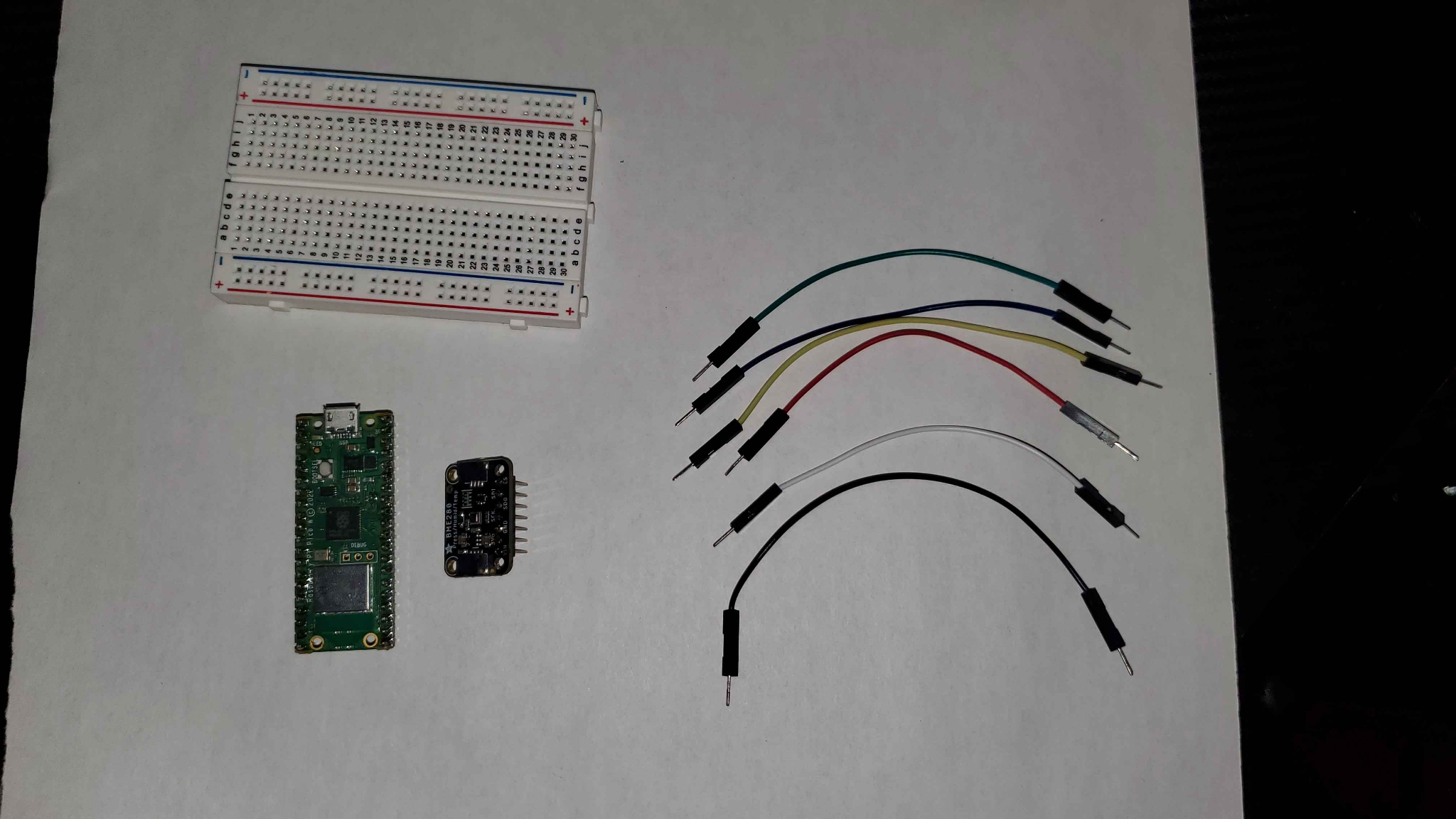
1. Execute the command: *gunicorn server:app*

**Figure 10 - Starting the API server using Gunicorn**

## **Raspberry Pi Pico**

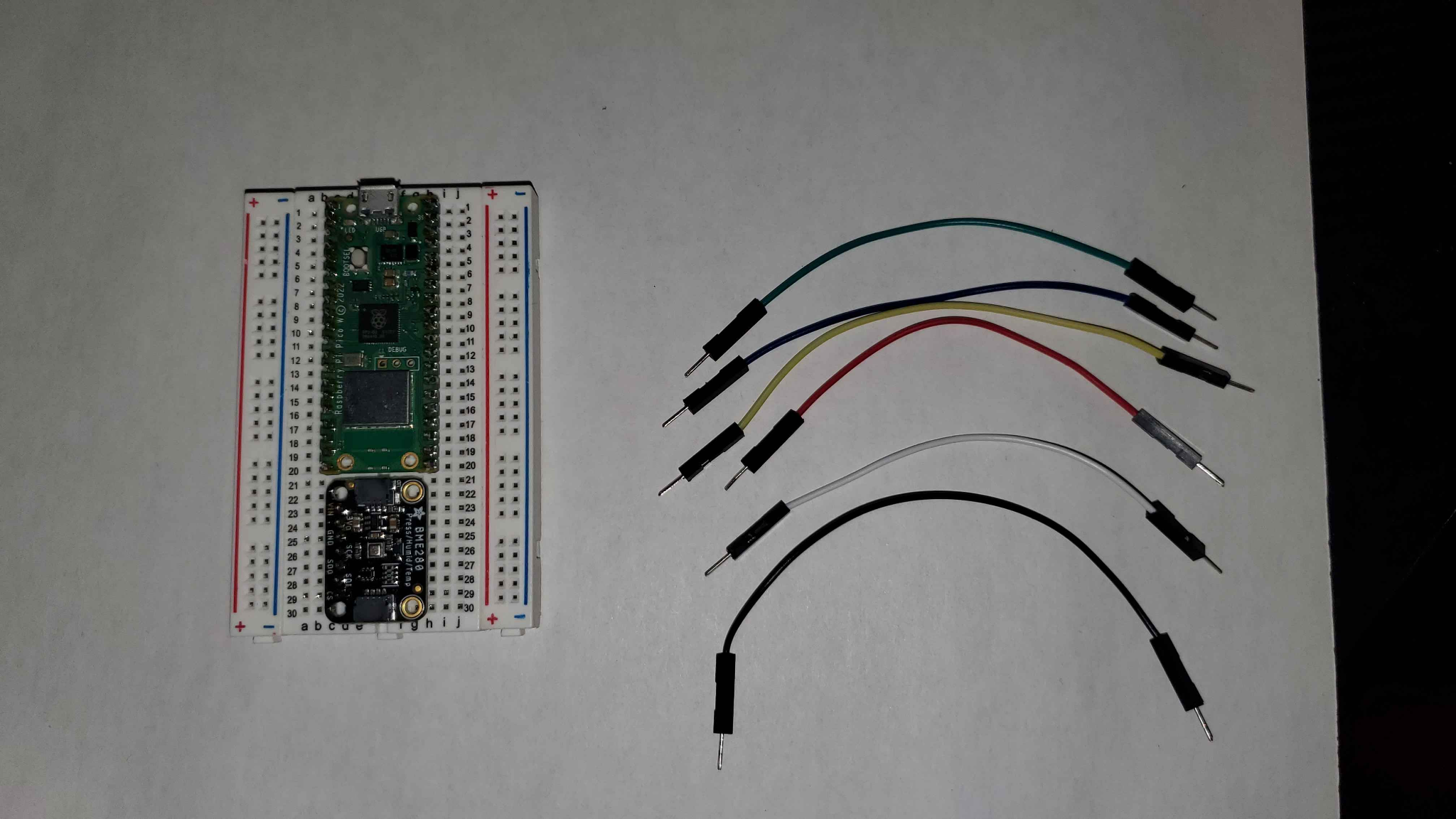
### Assembling the Pi Pico with the BME280 Sensor

1. Get your Pi Pico, BME280, Breadboard, and wires.



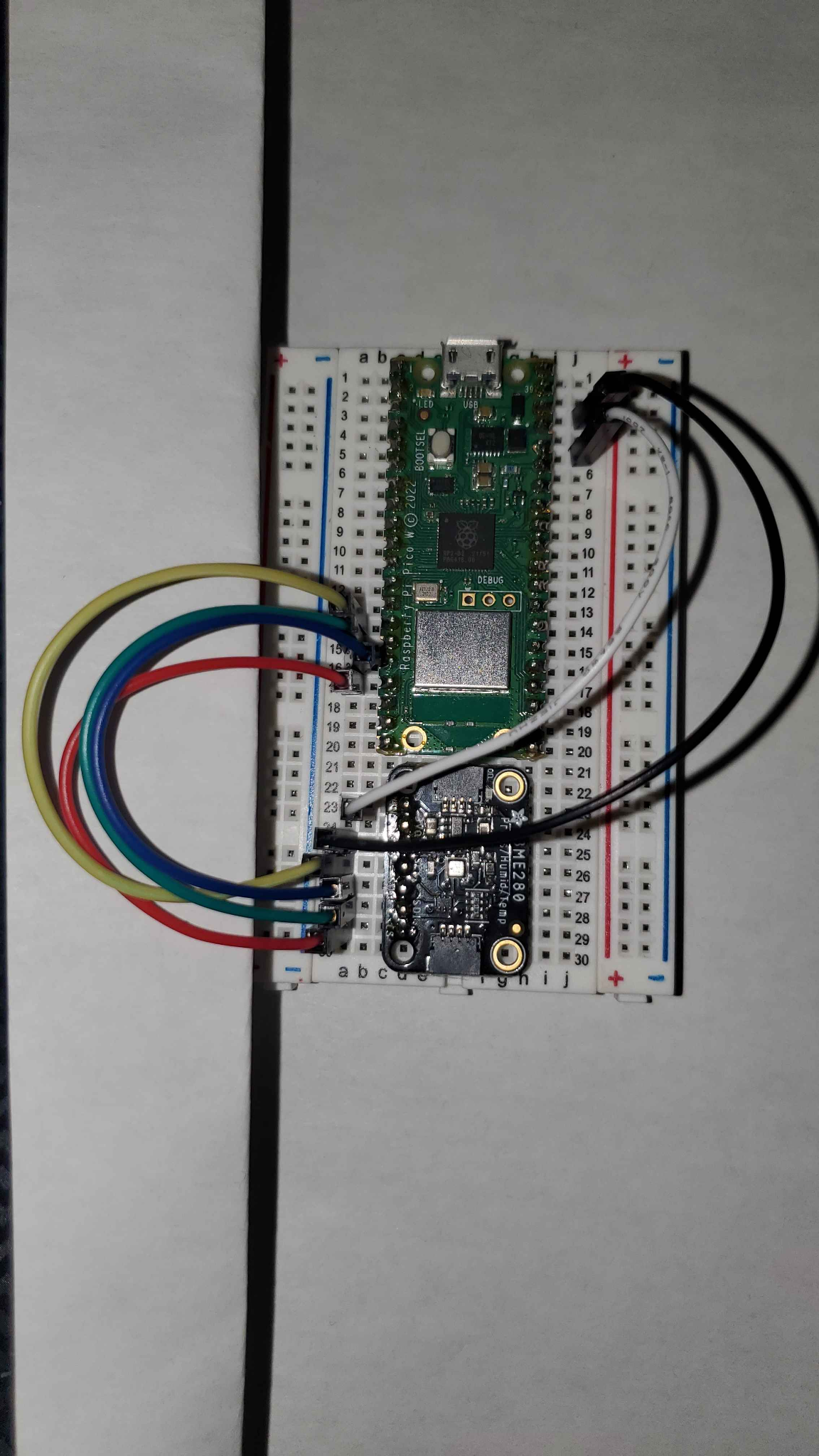
**Figure 11 - Pi Pico, BME280, Breadboard, and wires**

1. Place your Pi Pico and BME280 on the breadboard as shown below



**Figure 12 - Pi Pico and BME280 on Breadboard**

1. Use your wires to connect your Pi Pico and BME280 as shown below



**Figure 13 - Fully assembled Pi Pico and BME280**

**Wiring based on setup above:**

**Black: 3R -> 25L**

**White: 5R -> 23L**

**Yellow: 14L -> 26L**

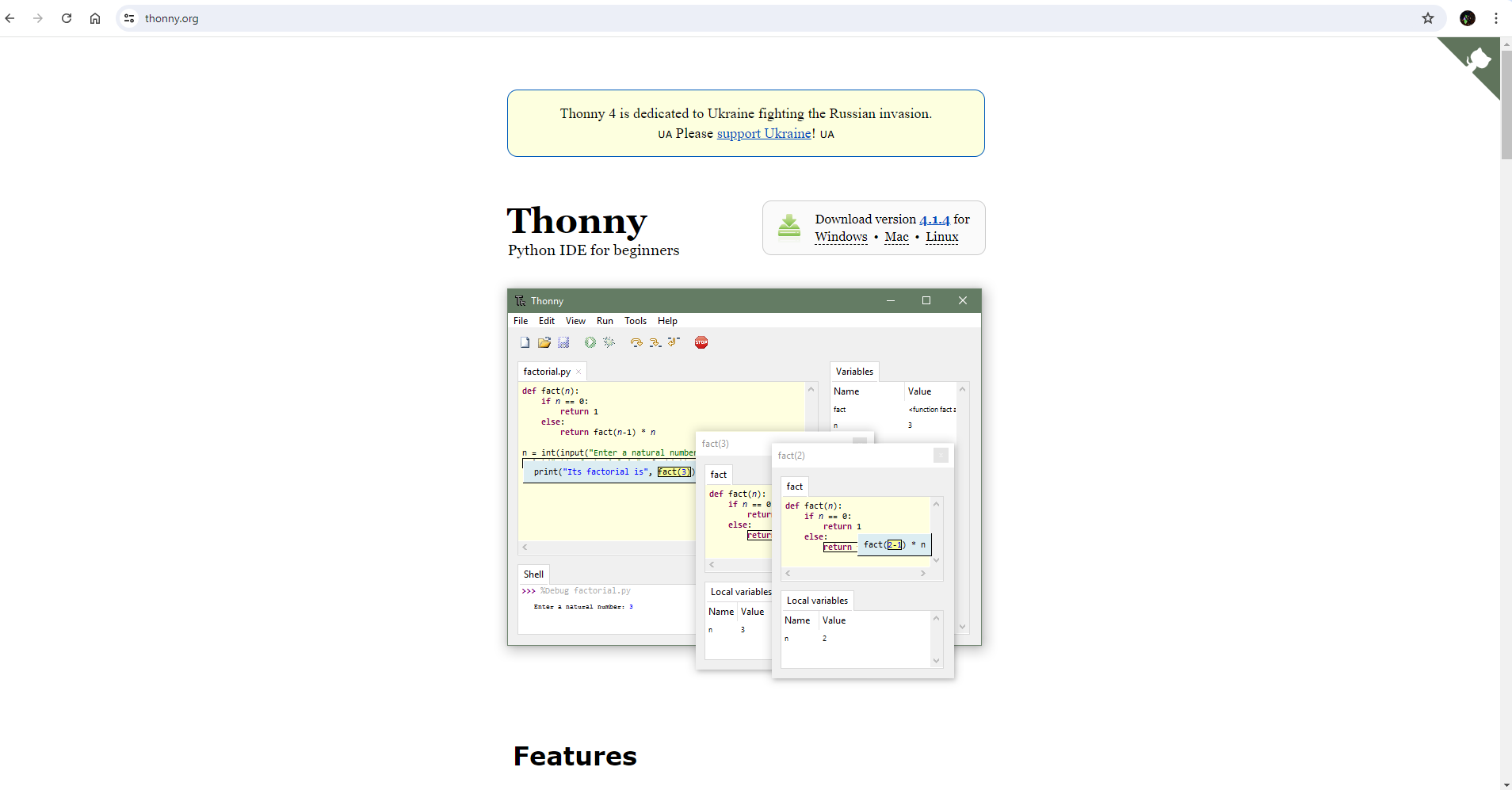
**Green: 15L -> 28L**

**Blue: 16L -> 27L**

**Red: 17L -> 29L**

### Downloading Thonny

1. On your personal computer, open a web browser and go to [thonny.org](https://thonny.org/)



**Figure 14 - Thonny.org web page**

1. Download the latest version for your operating system

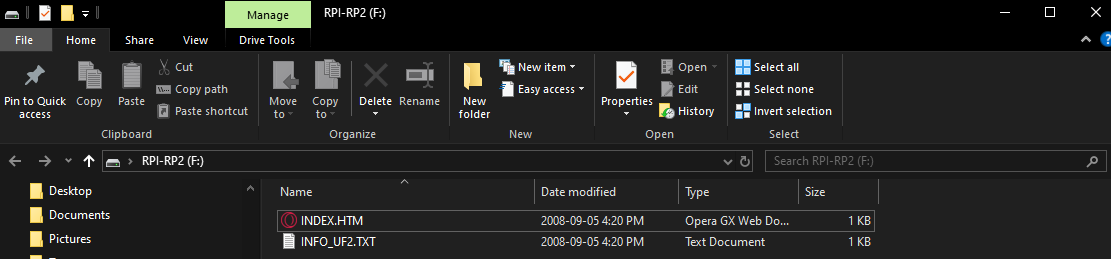
  
**Figure 15 - Hovering on Windows to show download options**

1. Run Thonny-(installed version).exe

### Installing Circuit Python on the Pi Pico

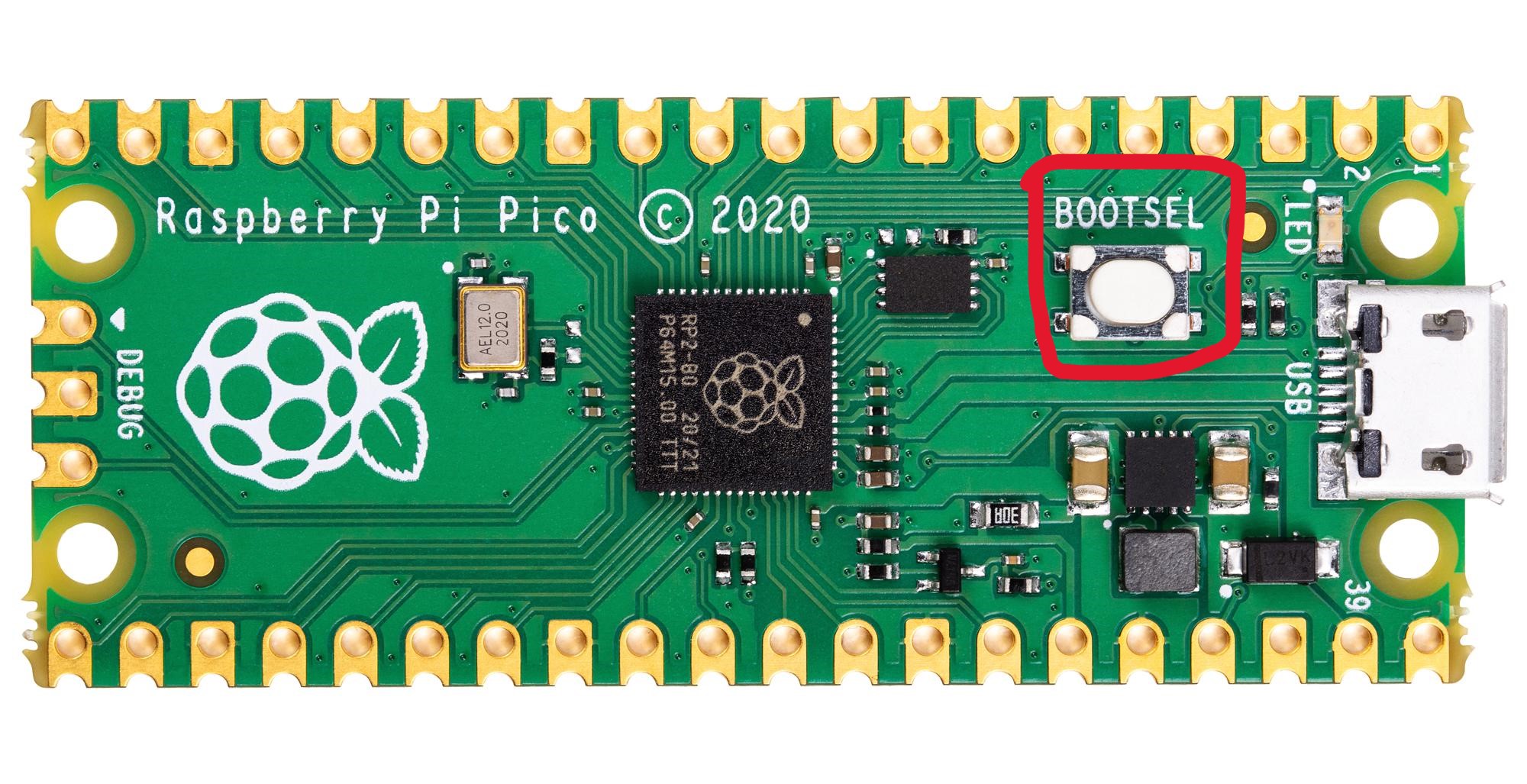
1. Plug your Pi Pico into your personal computer via micro-USB

Your personal computer will detect it as a storage drive and after a couple seconds it will open up a folder

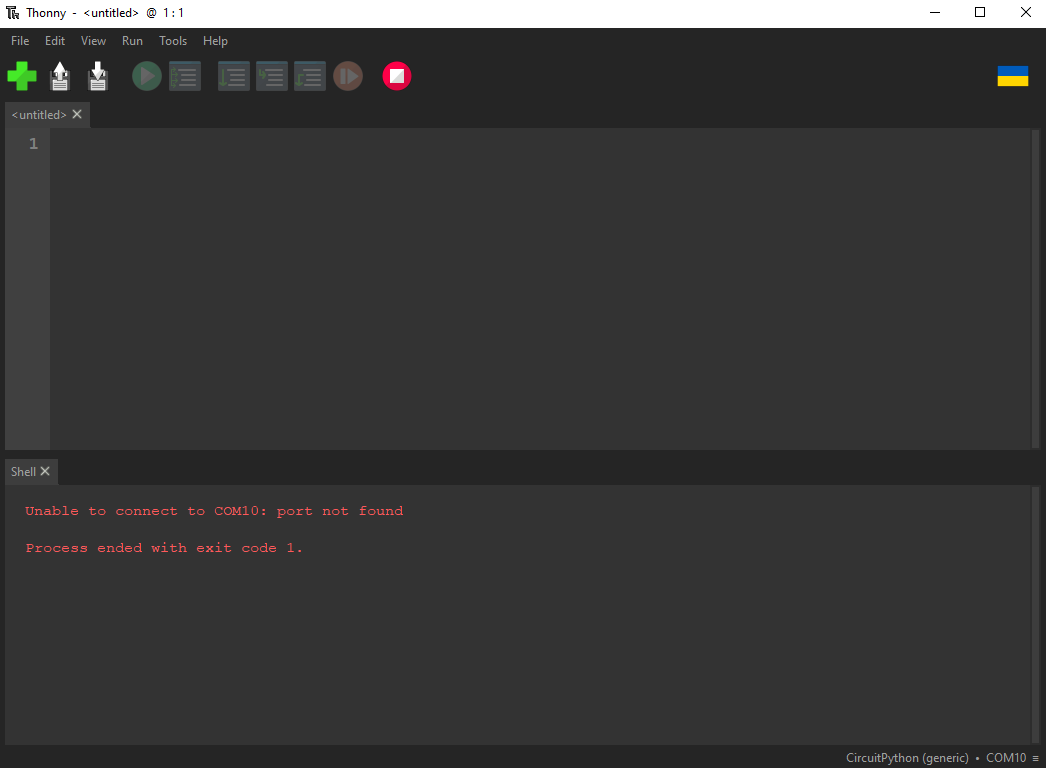


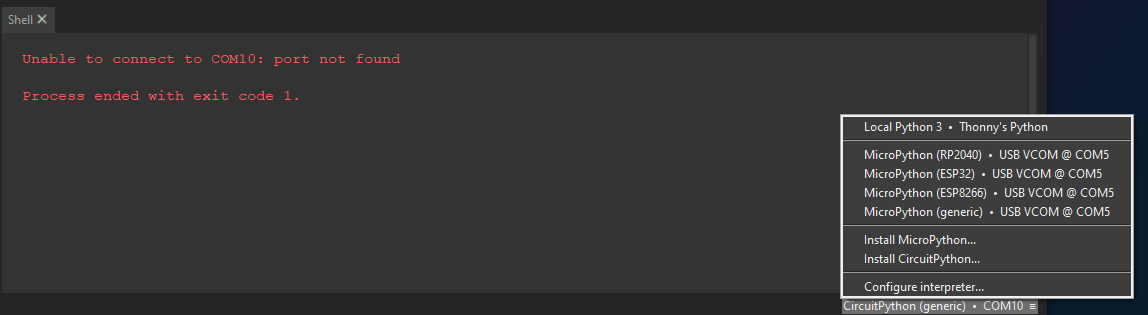
**Figure 16 - File explorer opened after plugging in Pi Pico**

**If your personal computer doesn’t detect it as expected, try unplugging and plug it back in while holding the little white button on the top of the Pi Pico labeled BOOTSEL**



**Figure 17 - BOOTSEL button on Pi Pico**

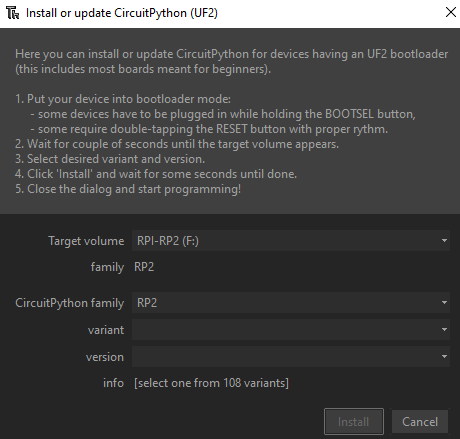
1. Open Thonny (if not already open)  
     
   
2. **Figure 18 - Thonny UI**
3. Click on the button at the bottom left of the window (In this case, “CircuitPython (generic) - COM10”



**Figure 19 - Selecting install options on Thonny**

1. Click on “Install CircuitPython”

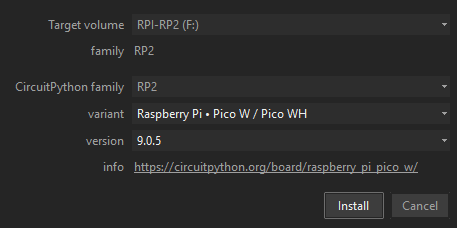
You’ll be met with the following window



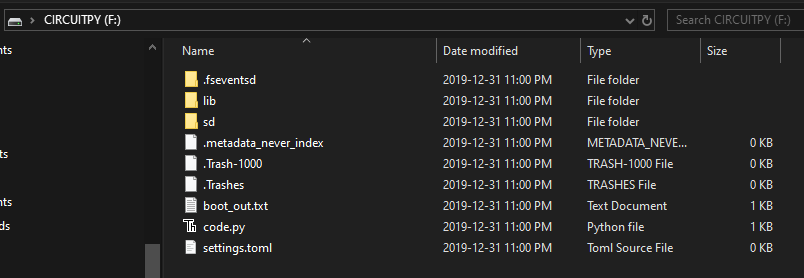
**Figure 20 - Install configuration UI**

1. Fill in the variant fields as shown below then click install  
   ***Note: make sure you use version 9.X.X. this has been tested with 9.0.5 and 9.1.0***

***If you wish to use versions 8.X.X you will need to find their respective versions library variants at*** [*https://circuitpython.org/libraries*](https://circuitpython.org/libraries)

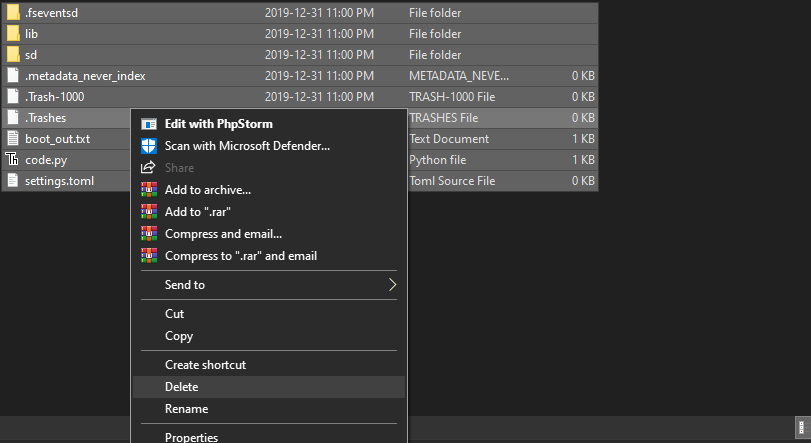


**Figure 21 - Filled configuration options**

*After roughly a minute, the install will finish and automatically open a folder for your Pi Pico  
  
*

**Figure 22 - Folder opened after installing CircuitPython**

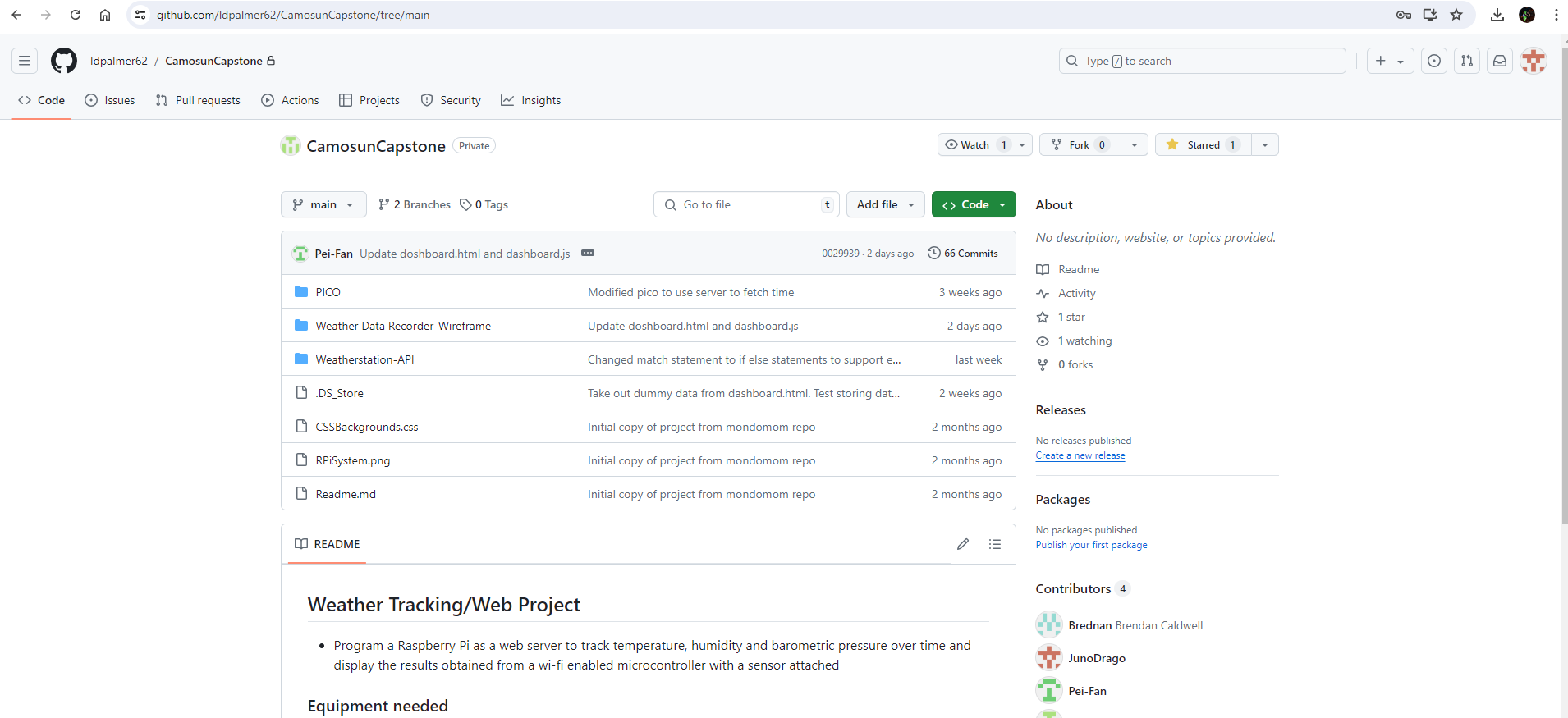
1. Delete all the files in the directory, we will not need them



**Figure 23 - Deleting all files in Pi Pico directory**

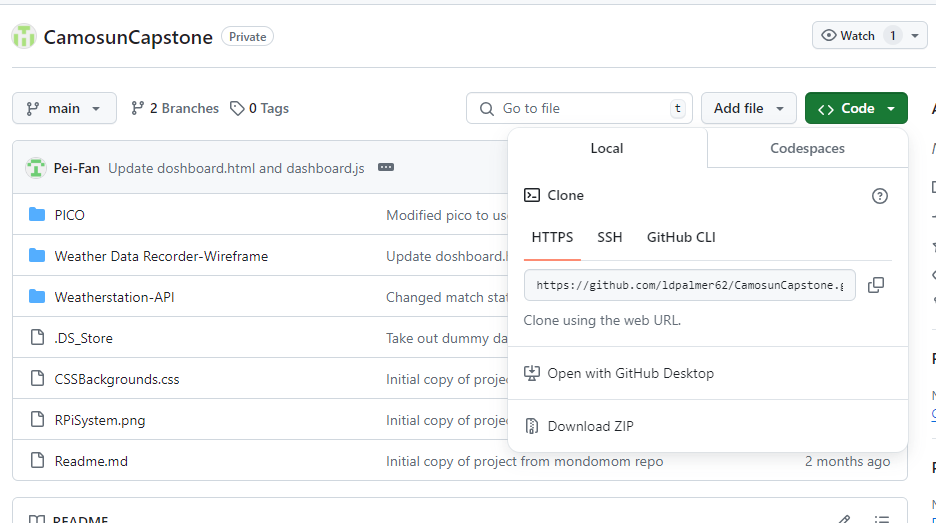
### Getting the Scripts onto the Pi Pico

1. On a web browser go to [github.com/ldpalmer62/CamosunCapstone](http://github.com/ldpalmer62/CamosunCapstone)



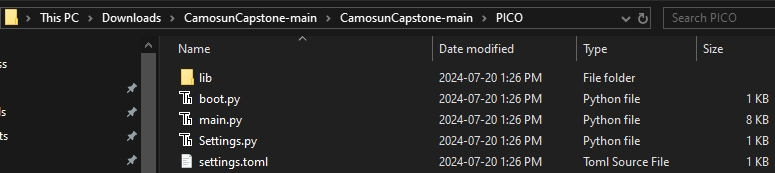
**Figure 24 - Github page**

1. Click on the **<> Code** dropdown
2. Click on **Download ZIP**

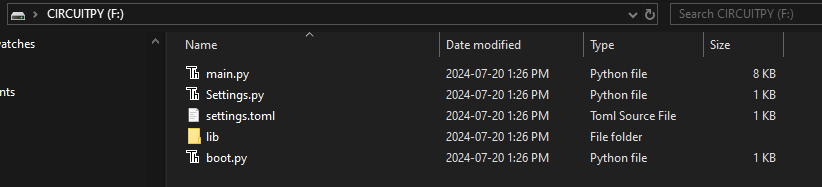


**Figure 25 - Downloading the Github repository**

1. Unzip the “CamosunCapstone-Main.zip” folder
2. In the new unzipped folder, navigate into the PICO folder

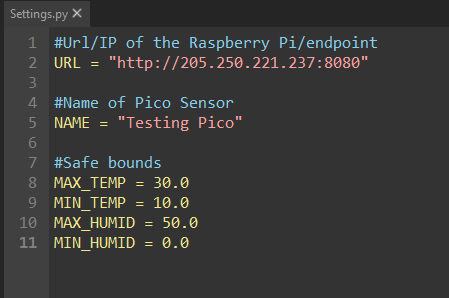


**Figure 26 - PICO folder**

1. Copy all the files into the Pi Picos directory (likely named “CIRCUITPY (D:)”)

**Figure 27 - Pi Pico folder filled with new files**

### Adjusting the Scripts for Personal Use

1. Double click on Settings.py  
     
   

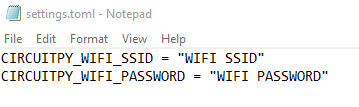
**Figure 28 - Settings.py opened with Thonny**

1. Change the URL to the URL or IP:PORT of your web server
2. Change the NAME to what you wish the name of the sensor to be
3. Adjust the Safe bounds to what you want your boundaries to be

Note that temp is in celsius and humidity is in %relative humidity

1. Save and close out of Thonny
2. Double click on settings.toml

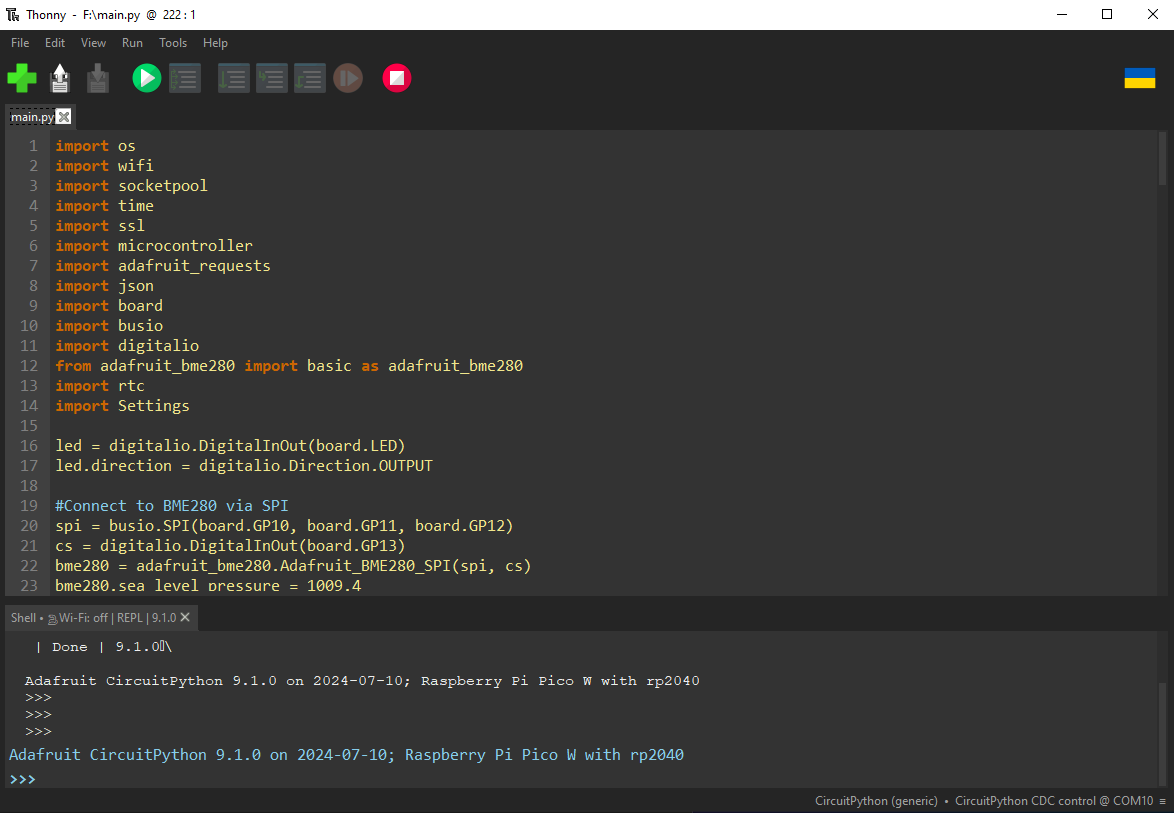
**You may need to choose what to open the file with, just choose notepad**



**Figure 29 - settings.toml opened with notepad**

1. Change the WIFI SSID and WIFI PASSWORD to your networks SSID and Password

**Note: your SSID is your wifi name, example: “Shaw 8964”**

1. Save and close the notepad
2. Double click on Main.py  
     
   

**Figure 30 - Main.py opened with Thonny**

1. Click on the green play button at the top of the screen

**Note: There’s a good chance it will not work, this is because the Pico is in read-only mode and is trying to write to a file called ID.txt, this is normal.**

**If it doesn’t work, Unplug and plug back in the Pi Pico, then repeat steps 9 and 10**

1. Congrats! your Pi Pico is now posting data to your web server!

# **Conclusion**

Congratulations! Your Raspberry Pi weather monitor is up and running. If you encounter any issues or need assistance with configuring settings not mentioned in this manual, feel free to contact us at:   
[CalebOlsen34@gmail.com](mailto:CalebOlsen34@gmail.com)   
[brendanshauncaldwell@gmail.com](mailto:brendanshauncaldwell@gmail.com)

[npoint122@hotmail.com](mailto:npoint2021@gmail.com)