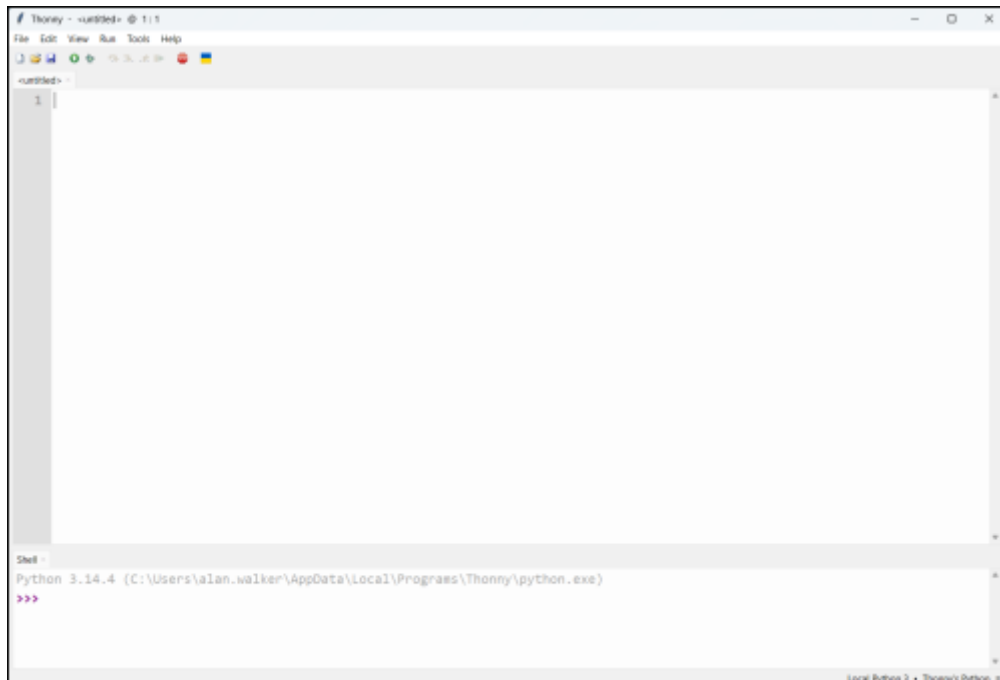


Thonny

Jun 2026



Introduction

Thonny, an IDE (Integrated Development Environment) is a software application that provides a central interface for programmers to write, test, and debug code. It combines essential tools—such as a code editor, compiler, and debugger—into a single program to make software development much more efficient.

What Thonny actually is

Thonny is a Python development environment that includes:

- A built-in Python interpreter
- A simple code editor
- A debugger
- Tools for uploading code to microcontrollers (like the Pico)

It's designed so you can focus on learning or experimenting without being overwhelmed by complex menus or settings.

Why people use Thonny for Raspberry Pi Pico

This is where Thonny really shines.

- It can detect a Pico automatically
- It lets you install MicroPython firmware onto the Pico
- You can run code directly on the board
- You can save files to the Pico's flash storage
- It shows the Pico's serial output in real time

It's basically the easiest way to get started with MicroPython on a Pico.

What you can do with Thonny

- Write Python scripts
- Upload code to a Pico or Pico W
- Debug your code step-by-step
- Inspect variables
- Use the REPL (interactive Python prompt)
- Manage files on the Pico

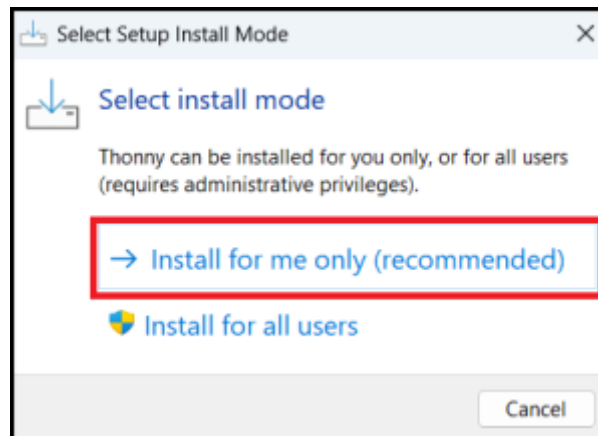
It's intentionally simple, but powerful enough for real projects.

Installing Thonny

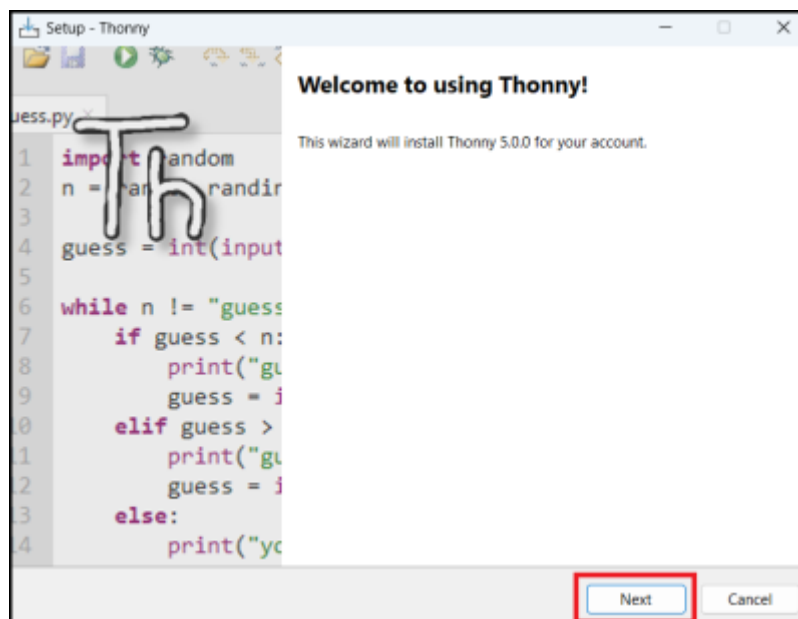
To download Thonny, head on over to their [Website](#) and download the latest version for your operating system. At the time of writing (June 2026) the current version is 5.0.0)

Once downloaded, double click the installer to start of the process.

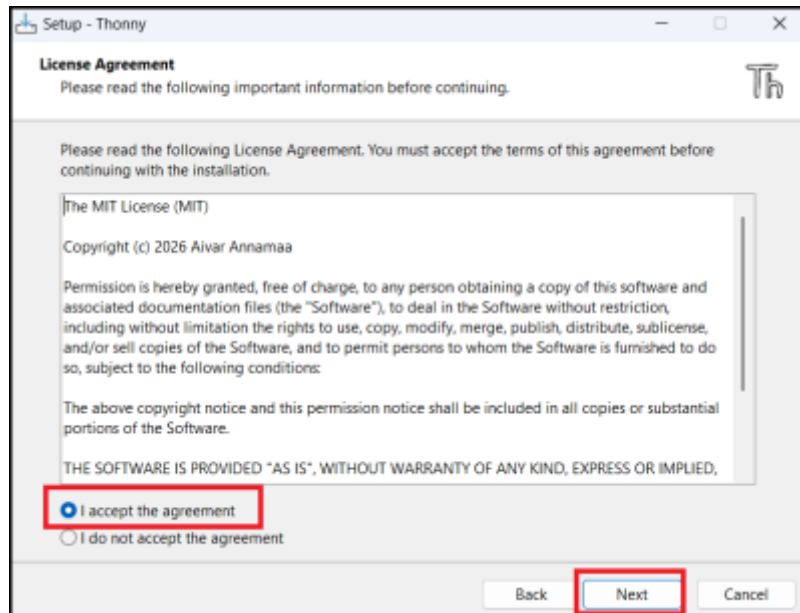
From the installation page, select Install for me only (unless you have a good reason to use (Install for all users).



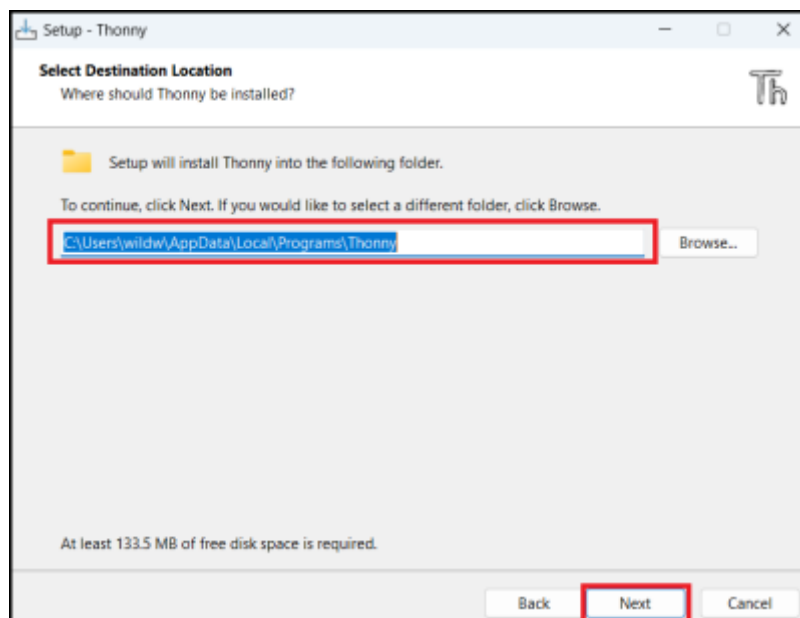
From the Welcome page, select Next.



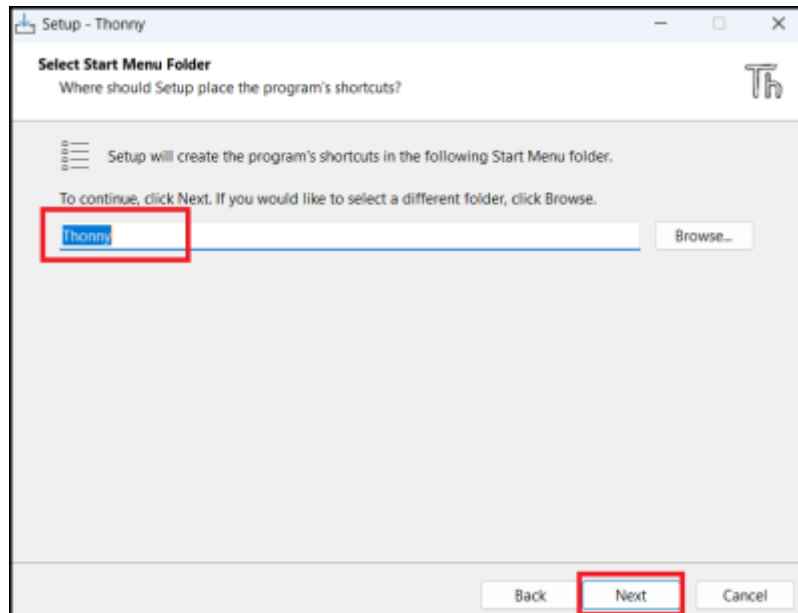
From the License Agreement page, select 'I accept the agreement' and click Next.



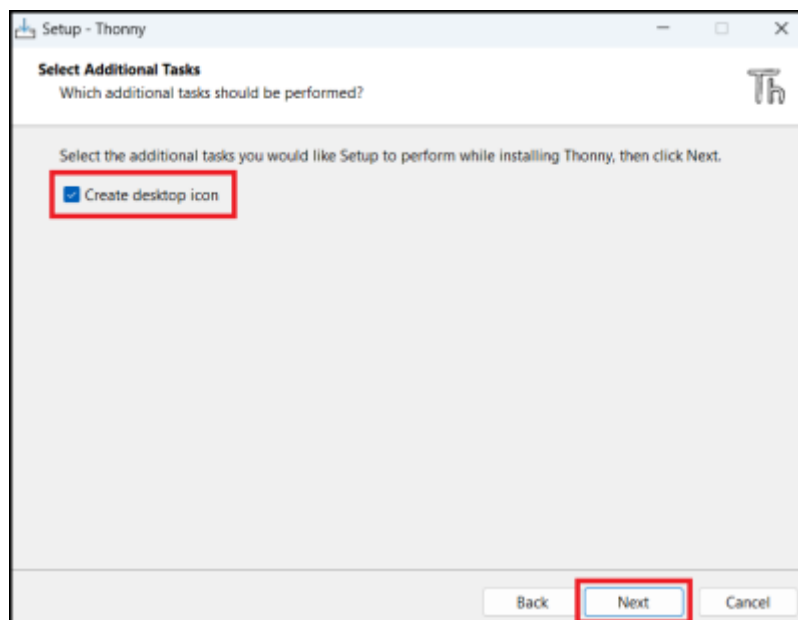
Use the default destination, unless you have a good reason not to. Click Next.



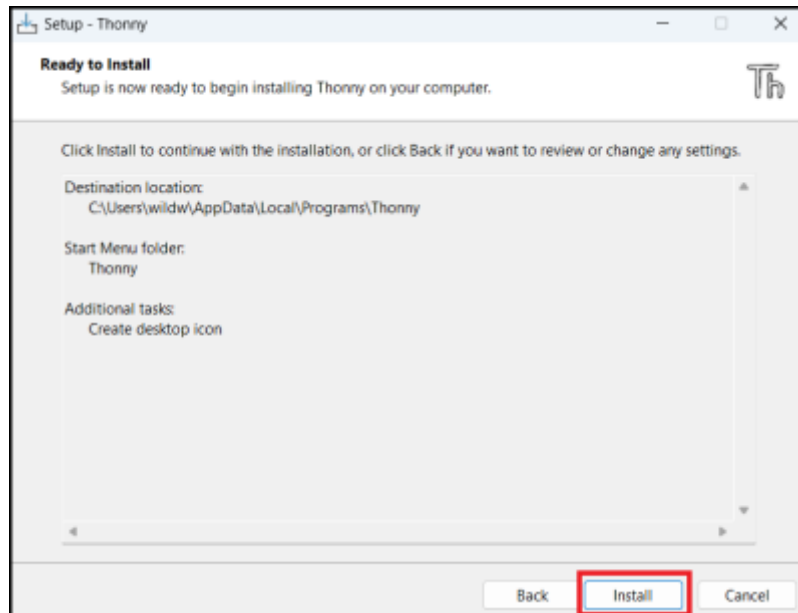
For the Start Folder name, just use the default of Thonny, click Next.



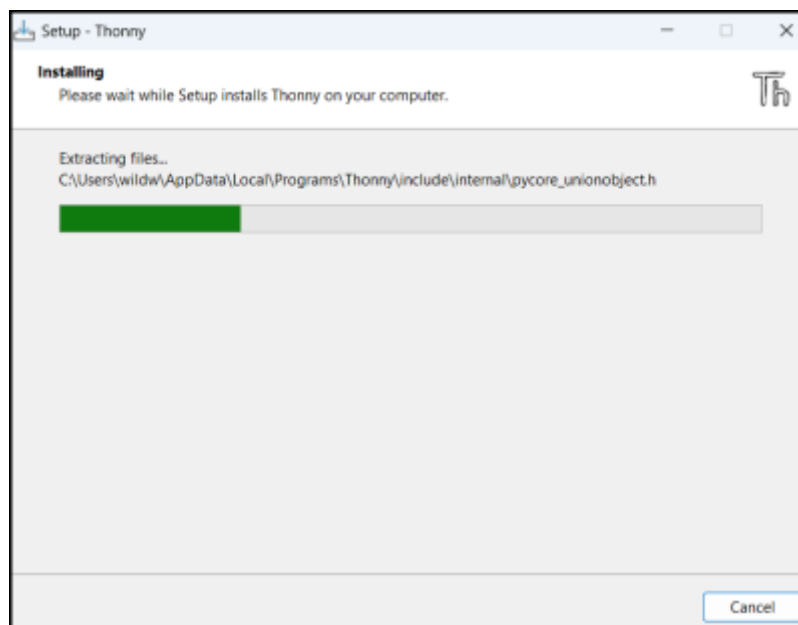
Tick the 'Create Desktop Icon' and click Next.



From the Ready to install page, click the Install button.

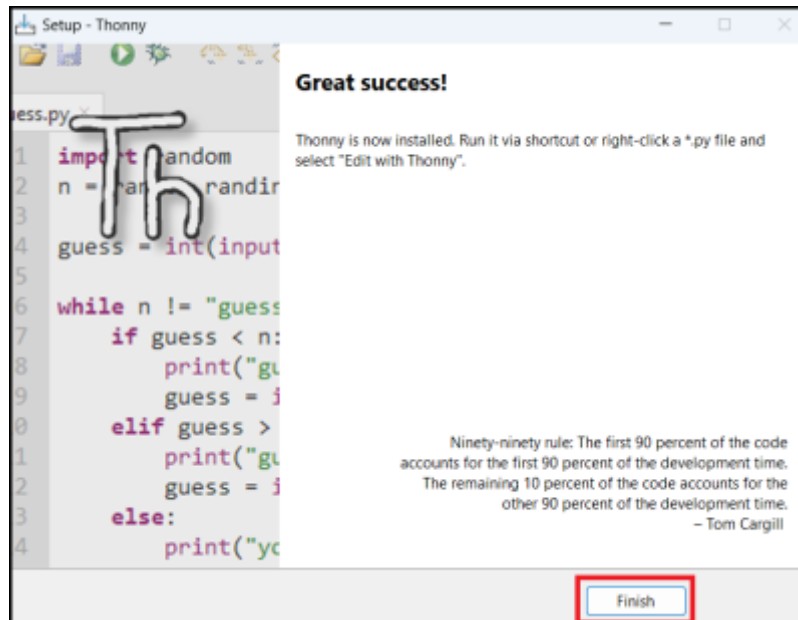


Thonny will start to install, nothing to do here but wait.



There can be quite a delay here, just be patient.

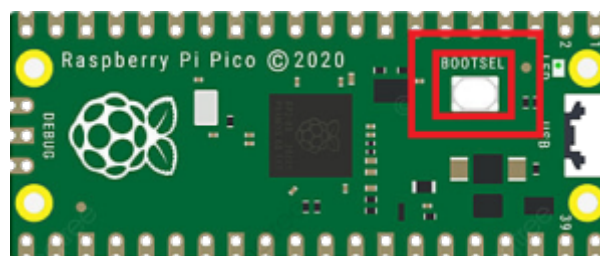
Once installed, you will see the following page. Click Finish to close the installer.



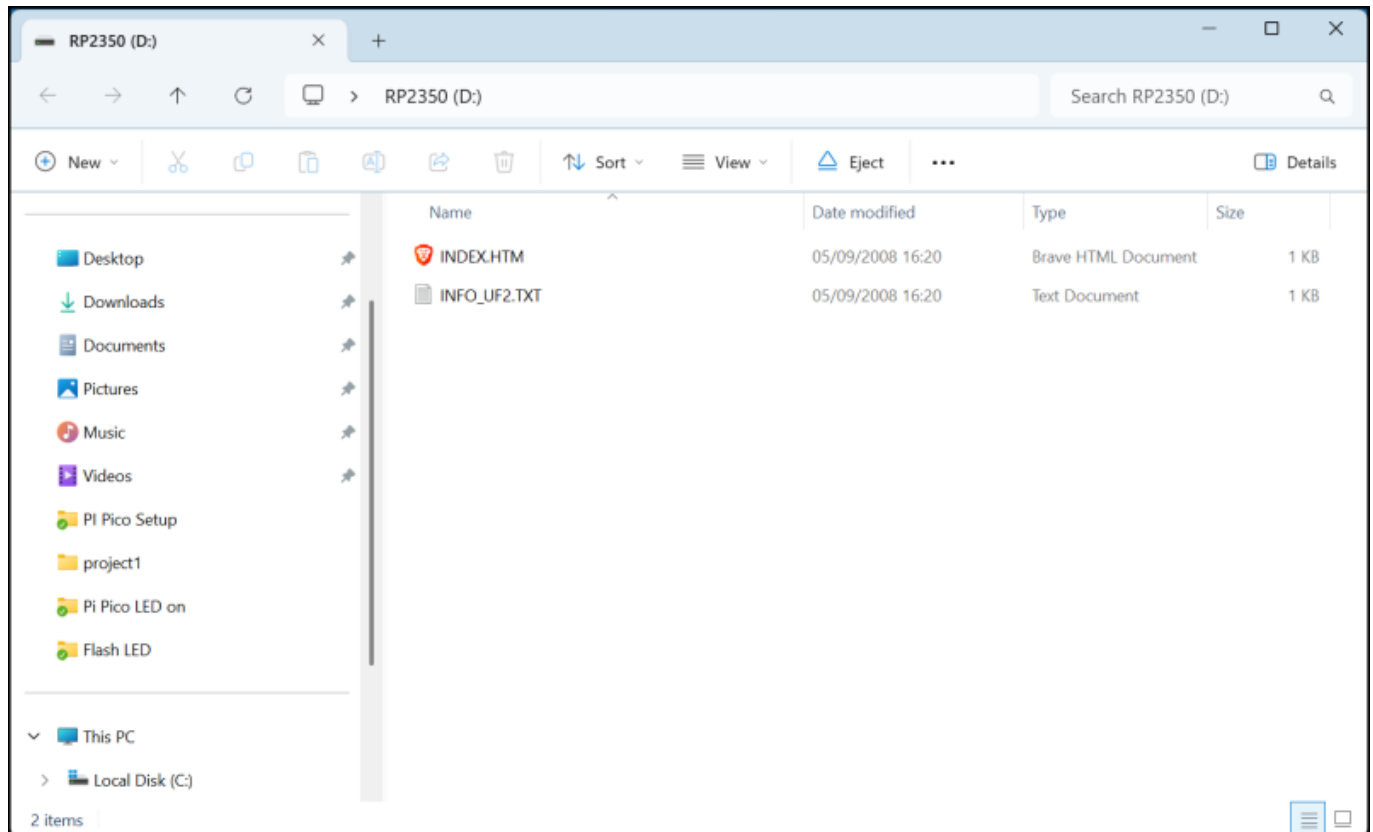
That is it, Thonny is now installed.

Running and Initial Setup

Now is the time to plug in the Pico to your computer, the one you have installed Thonny on to. You will connect the Pico to the PC using a USB cable. Before you connect the Pico, you need to hold down the BOOT SEL button on your Pico, this is the only button on the Pico so is simple to find.



While holding down the BOOTSEL button, plug the USB cable from your computer in to the Raspberry Pi Pico. The Pico will power up and a folder will appear in your Explorer.

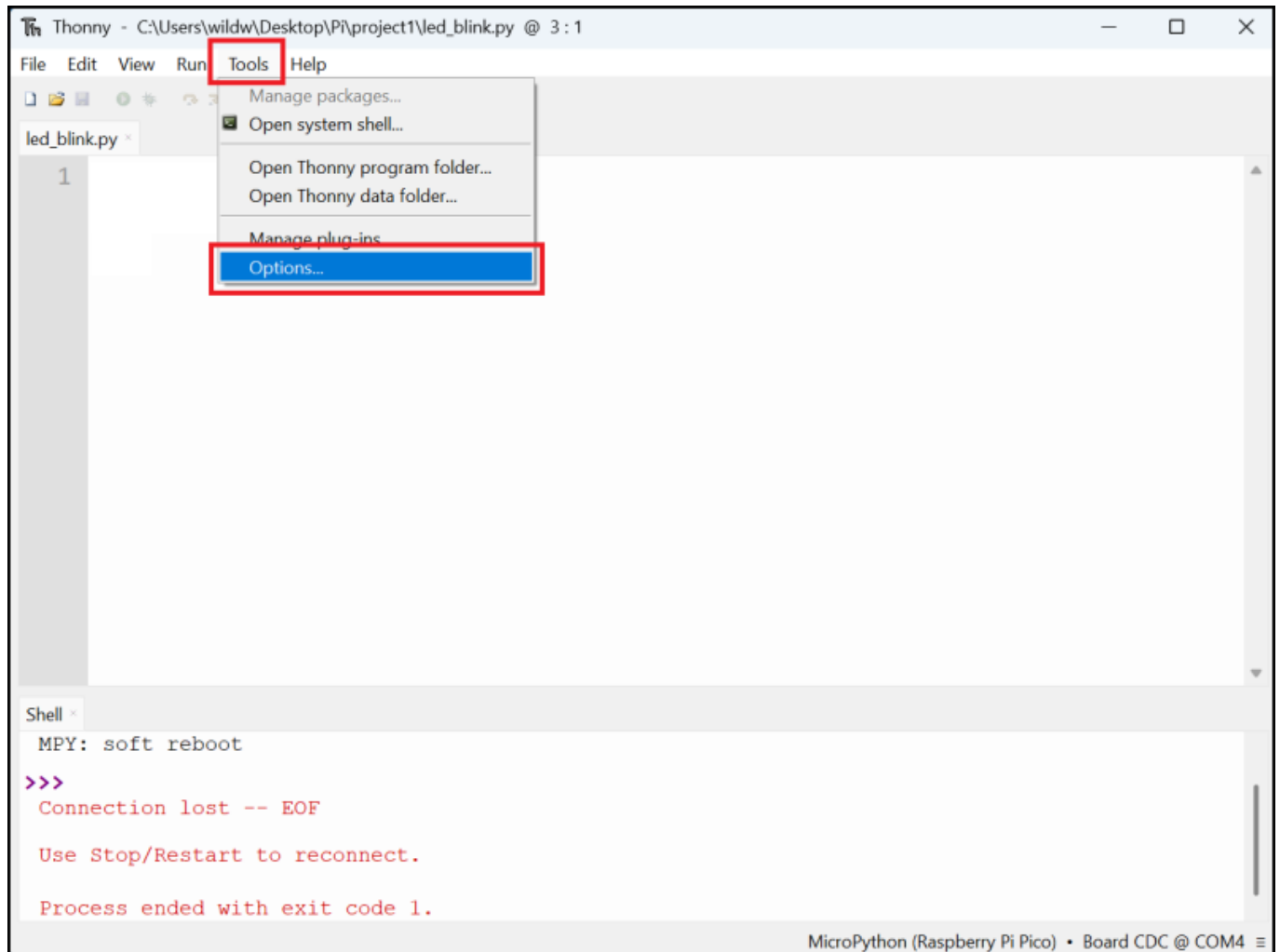


There will be two files in this folder:

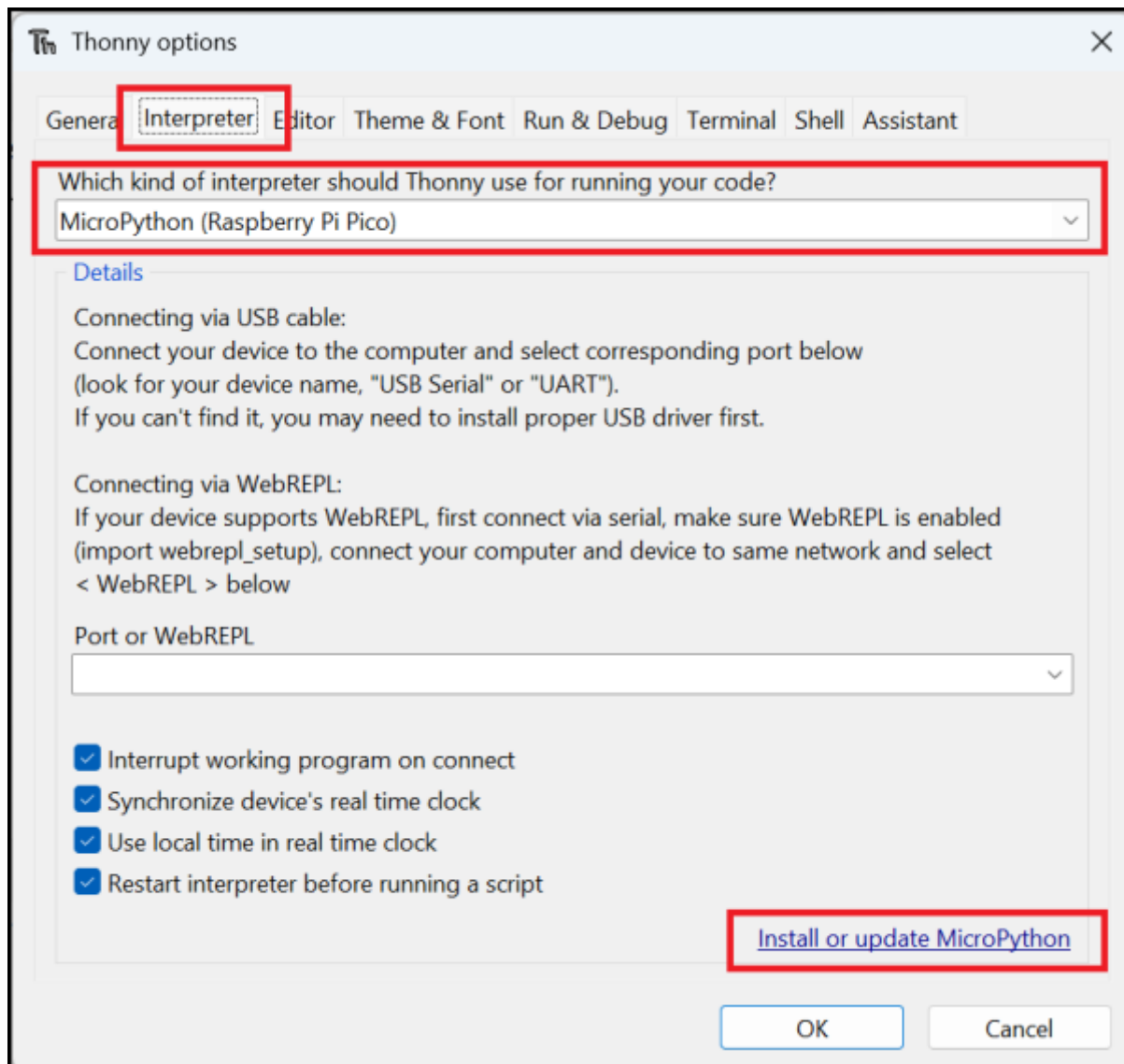
INDEX.HTM
INFO_UF2.TXT

Please note that this is a brand new Pico, so we would not expect to see any other files.

Run your Thonny application that you installed earlier.

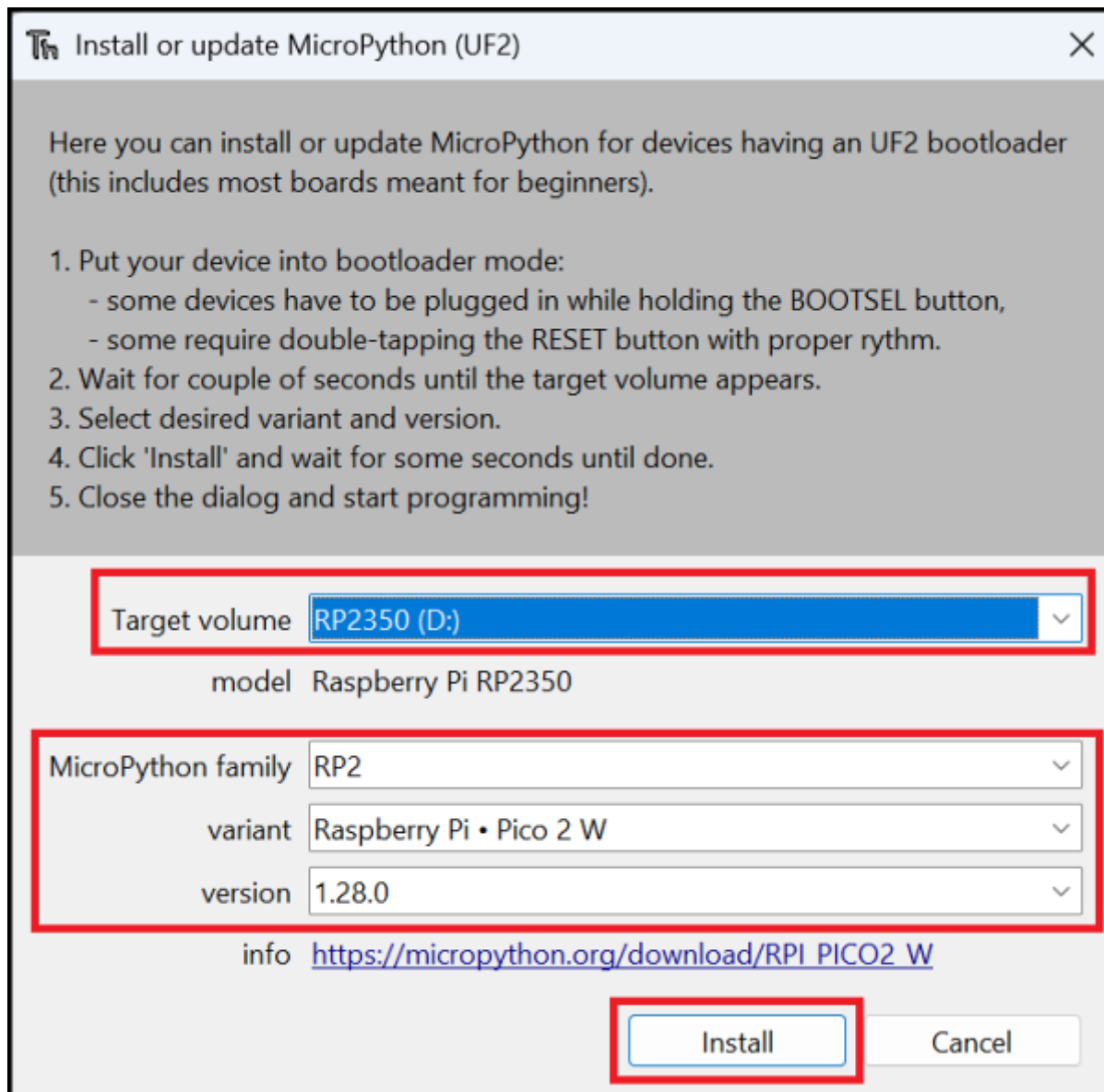


Select from Thonny, Tools then Options...



Select the Interpreter tab, ensure the MicroPython (Raspberry Pi Pico) Interpreter is selected. From the bottom of this page click 'Install or update MicroPython'

You will be presented with some options for the MicroPython install (or update).

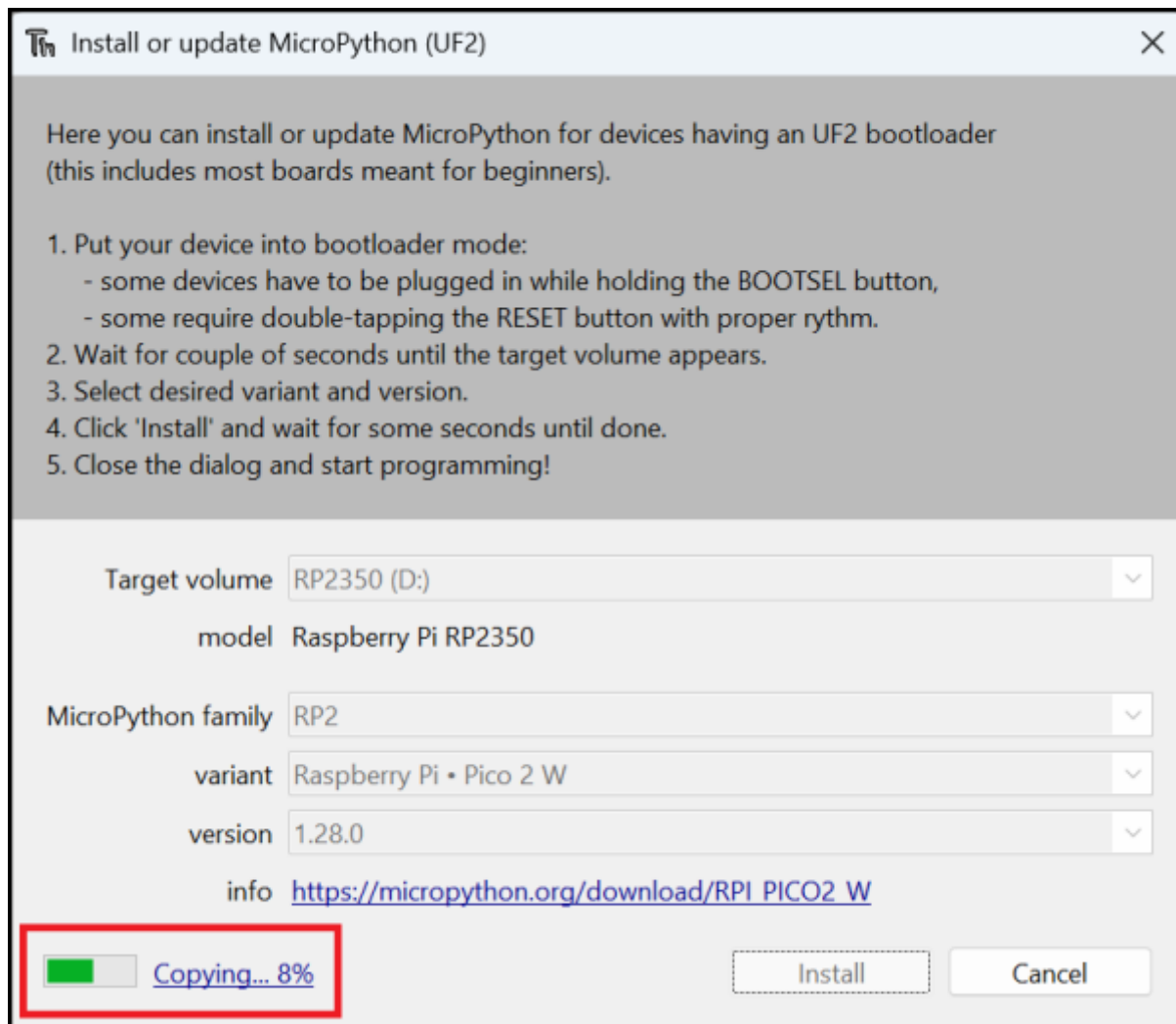


We need to set some of the options here to match our Pico device.

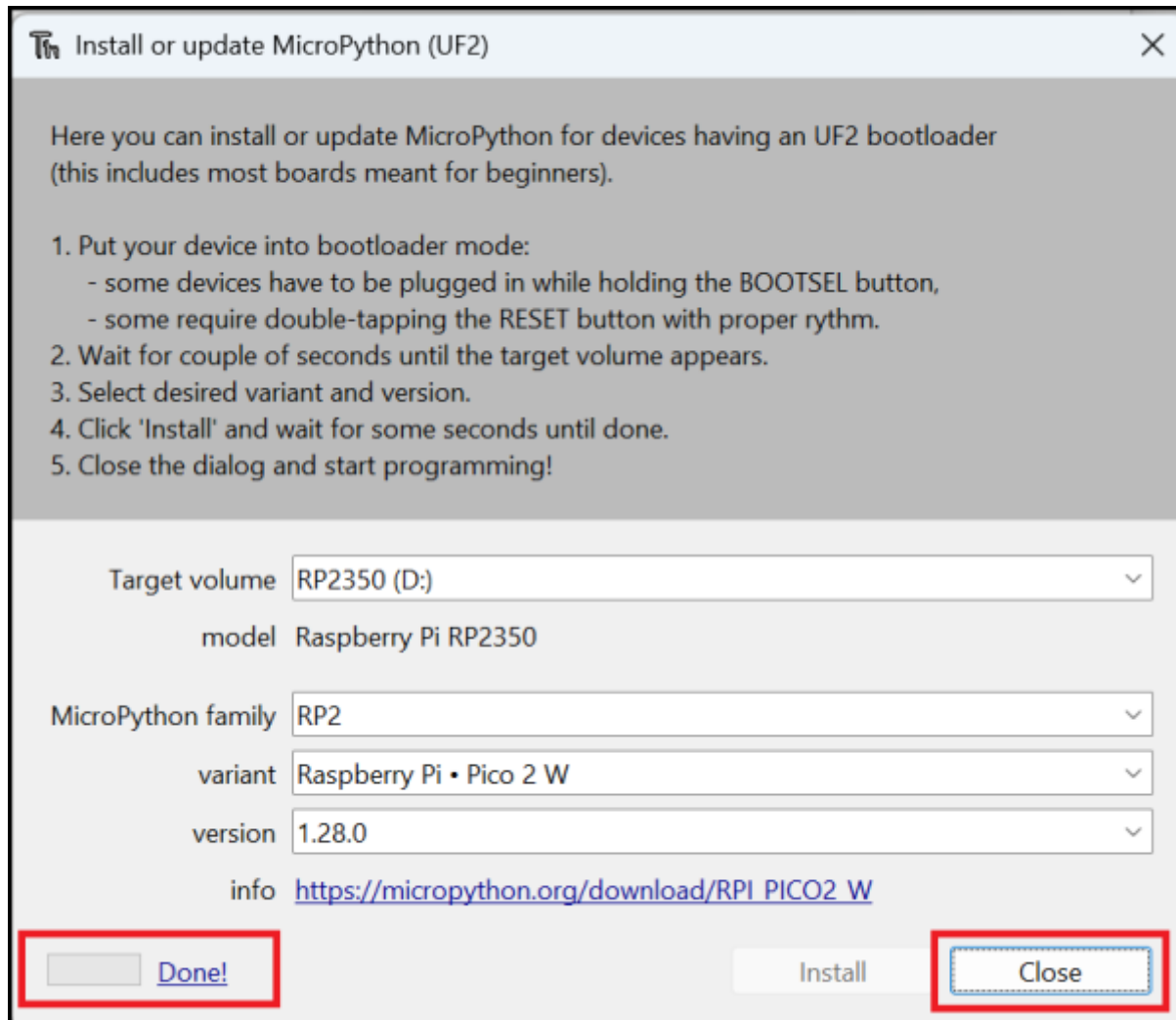
- Target Volume - This is the drive in explorer your Pico connected as.
- MicroPython Family - RP2
- Variant - Raspberry Pi Pico (your Pico type) I am using a Pico 2 W in this example
- Version - Select the latest version (unless you have a compelling reason not to)

Once set, click the [Install](#) button.

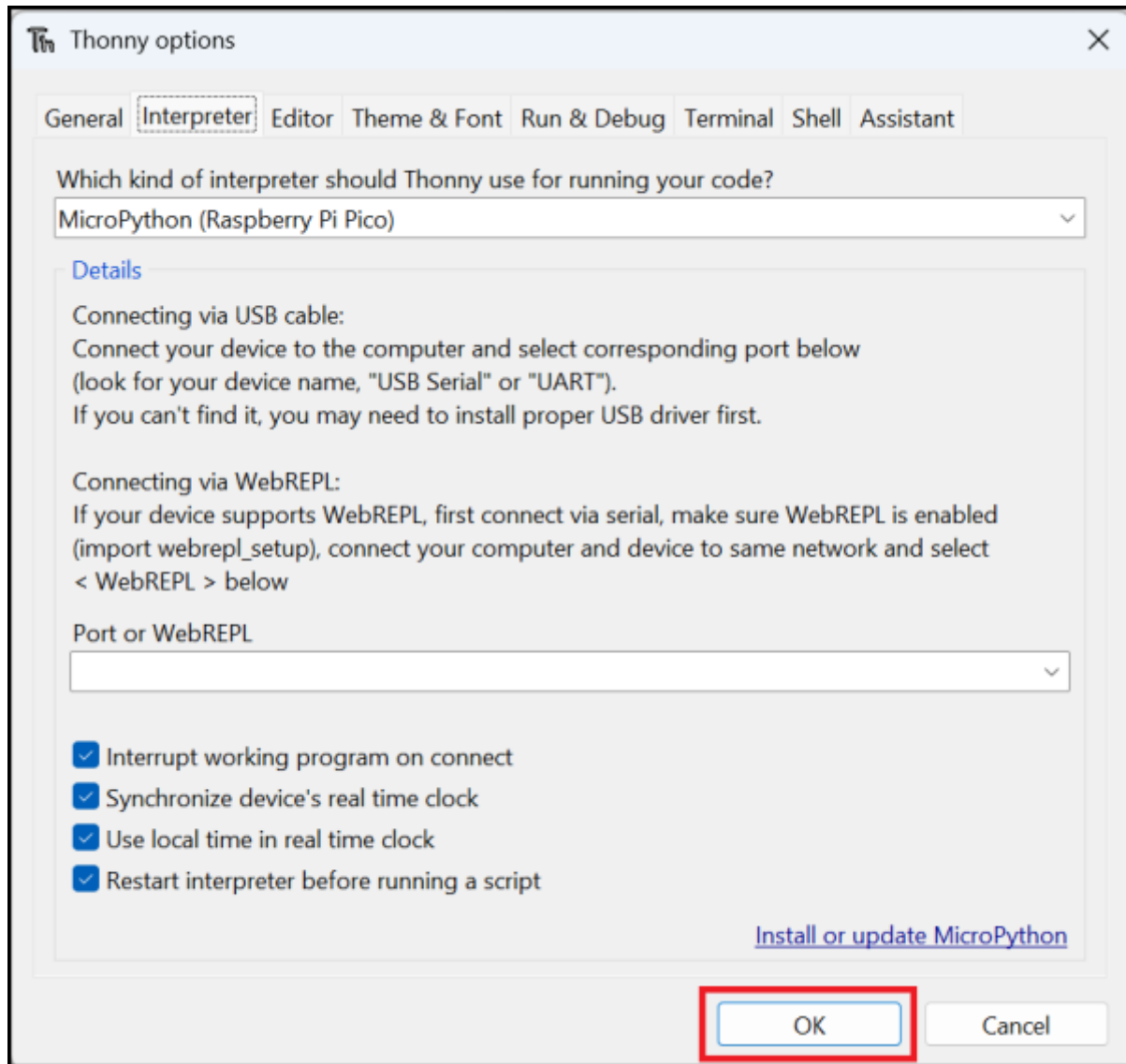
You will see a progress bar to show the installation is running.



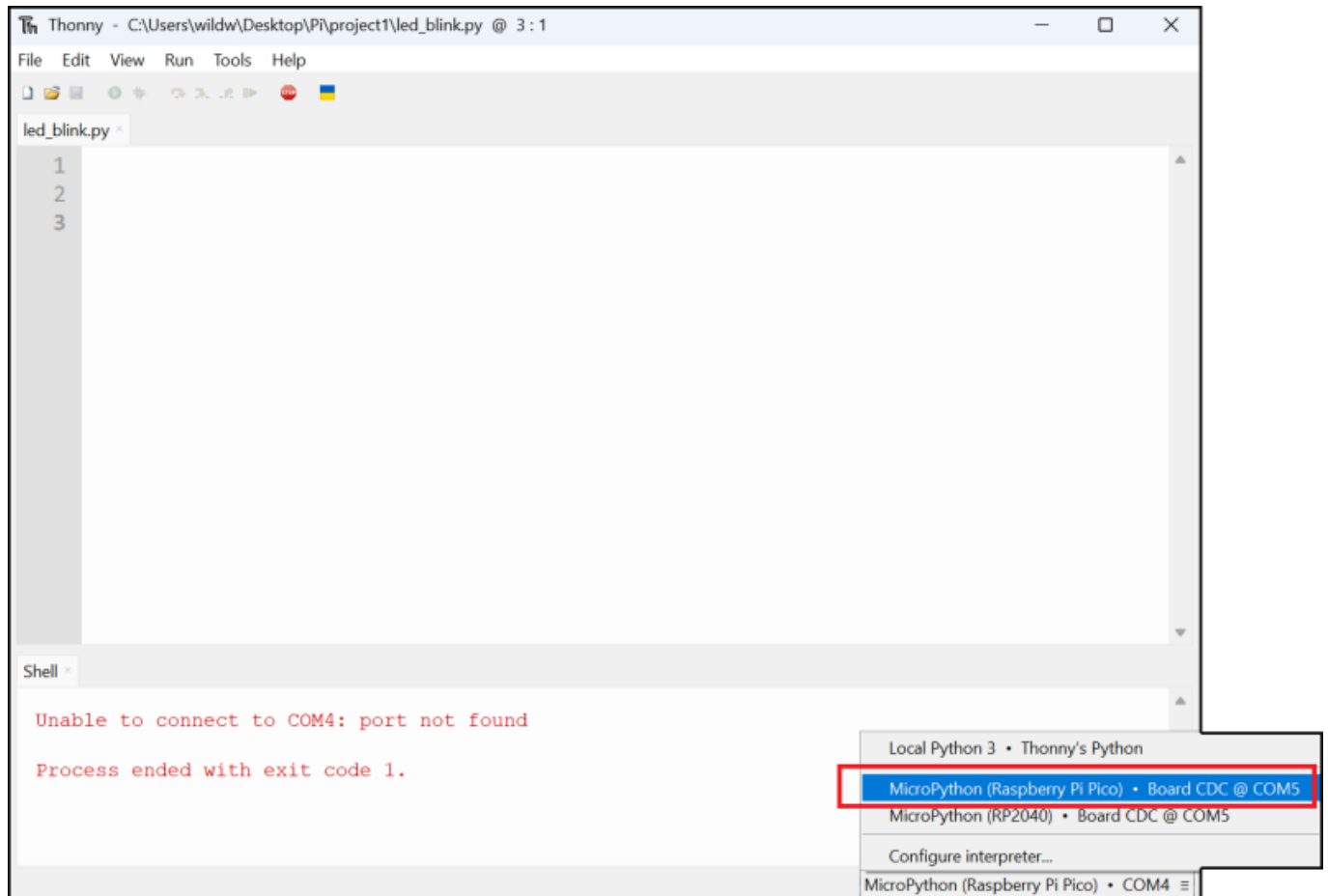
Once the installation is complete, the progress bar will change to **Done!**



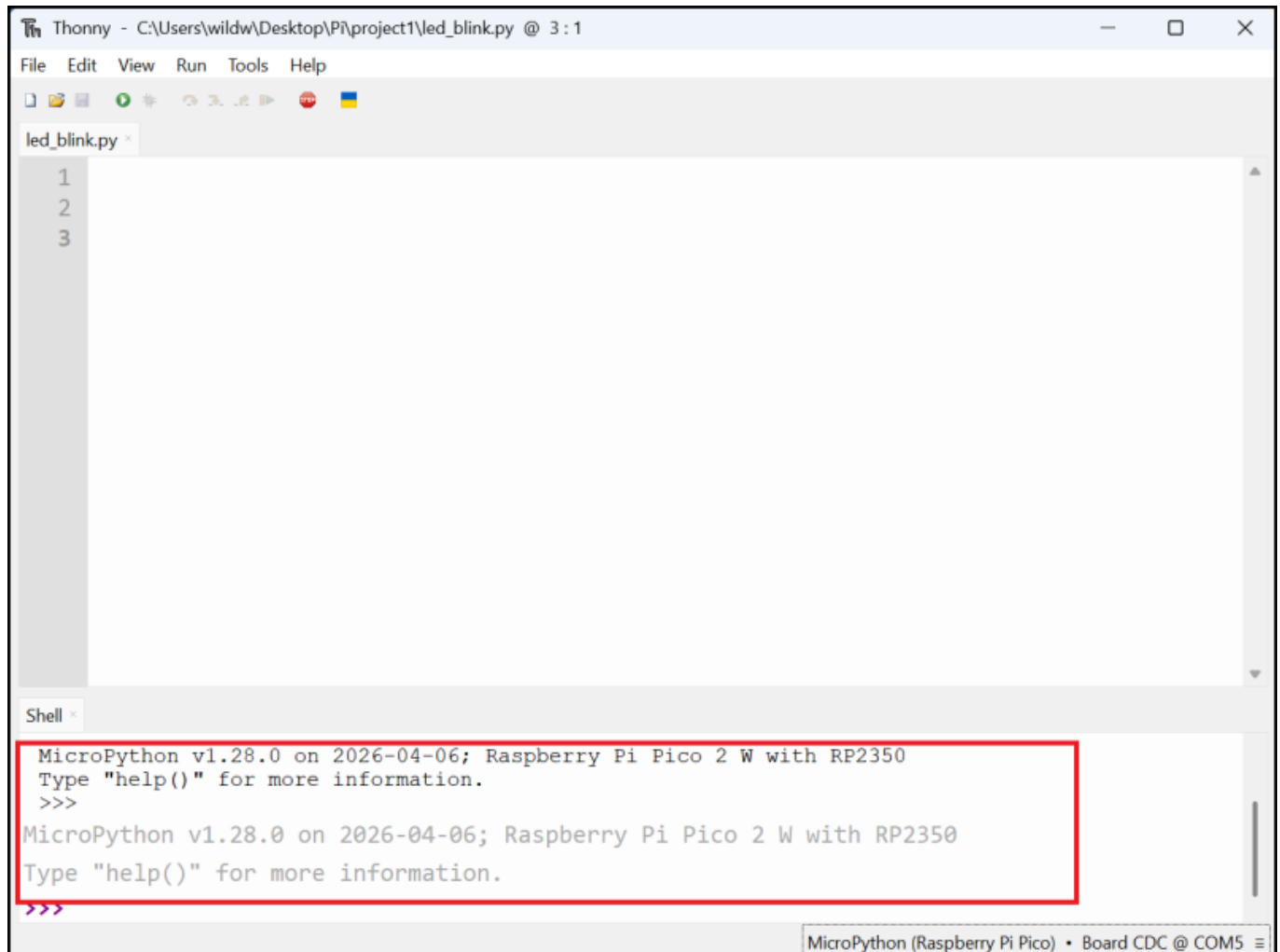
Once back at the Interpreter tab you can just select OK



You will now be back at the main Thonny interface. It will have taken just a few seconds for the Pico to reboot.



From the bottom right of Thonny select your Pico Device.



You will see that at the bottom of Thonny, it lists your Pico Device and the version of MicroPython that has just been installed on to it.

Install Pico Zero Library

Library files are like device drivers, they allow you to run commands to do something with a device. The picozero library is a simplified driver file for the Pi Pico.

You should now have a Thonny icon on your desktop. Double click the icon to start Thonny.

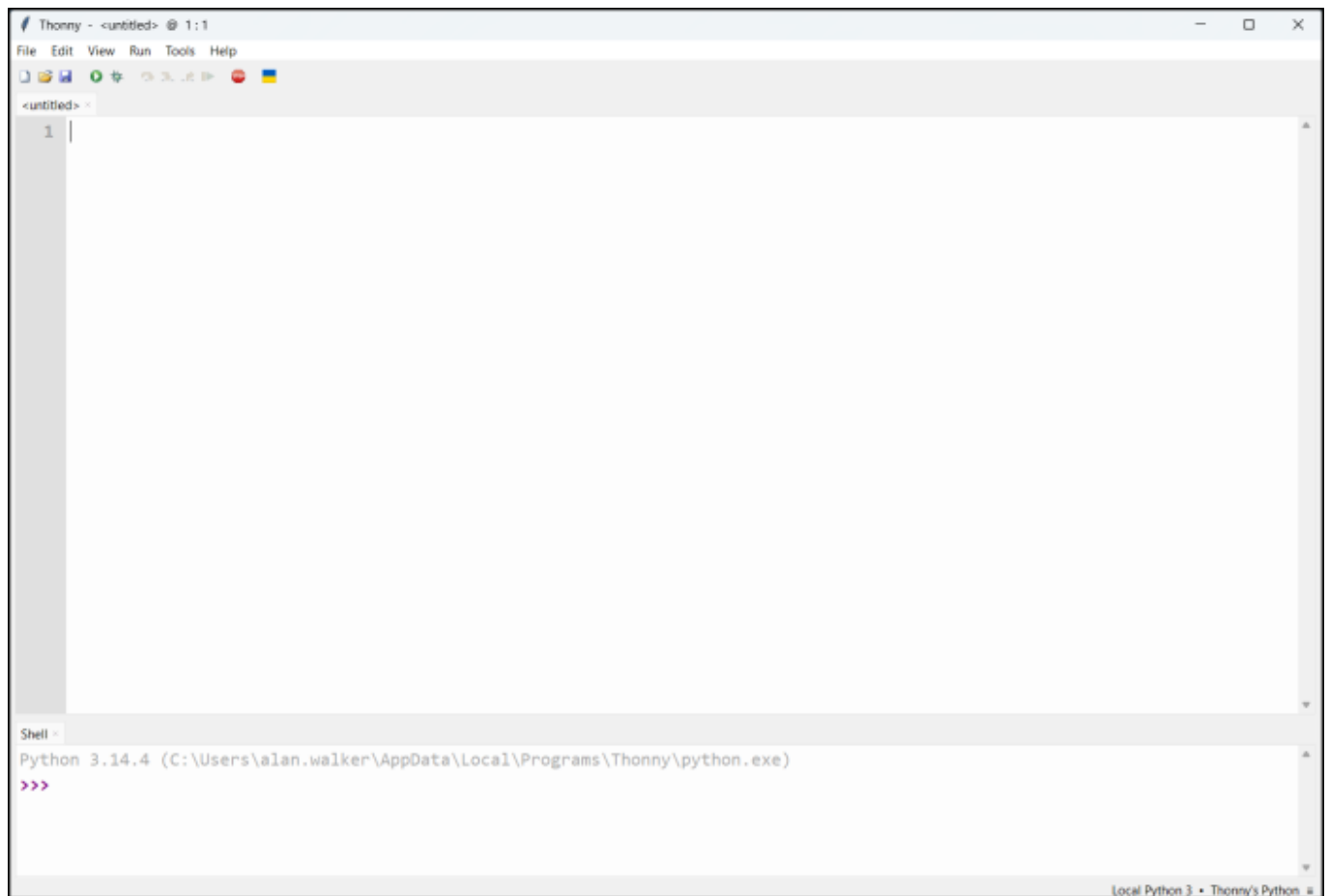


First page you will see is the options page. From here select your preferred language. Leave the 'Initial settings' at Standard.



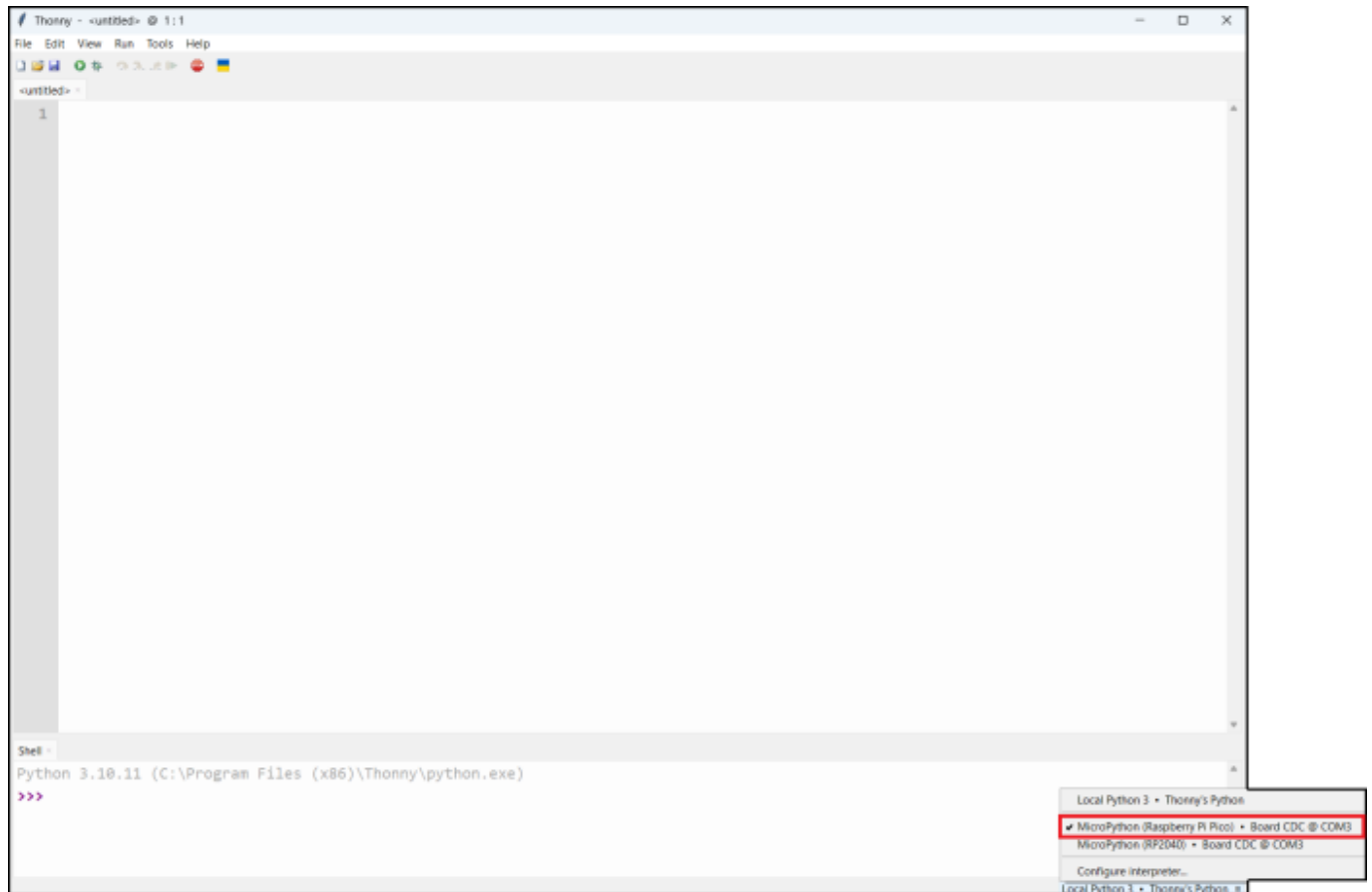
Click the 'Lets Go' button.

Thonny will now open.



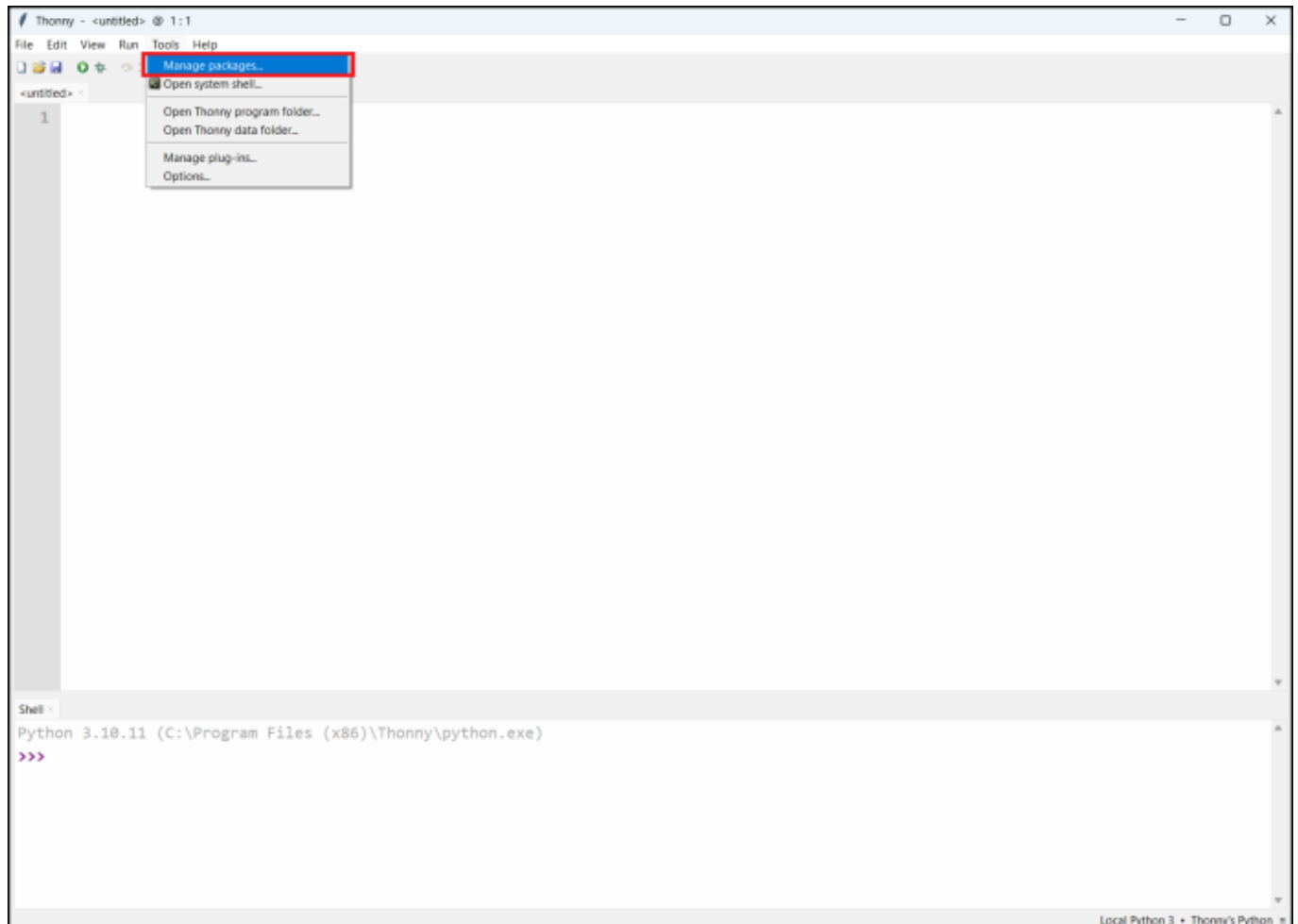
**** At this point, please connect your Raspberry Pi Pico to your computers USB Port ****

First thing we want to do with Thonny is select the device we are working with from the bottom right hand corner of the Thonny Application.

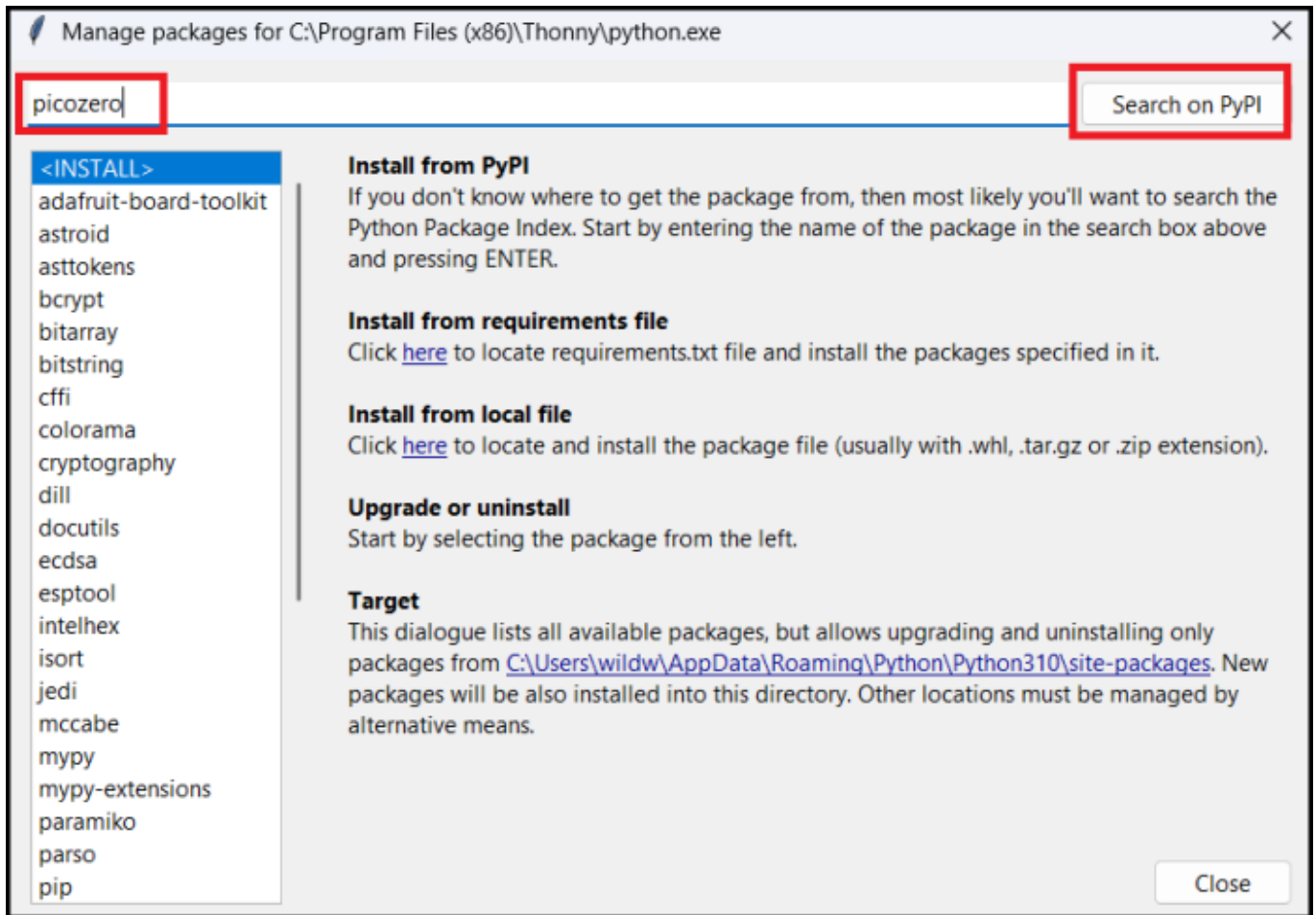


If you pico is not listed, then your computer has not recognised the device.

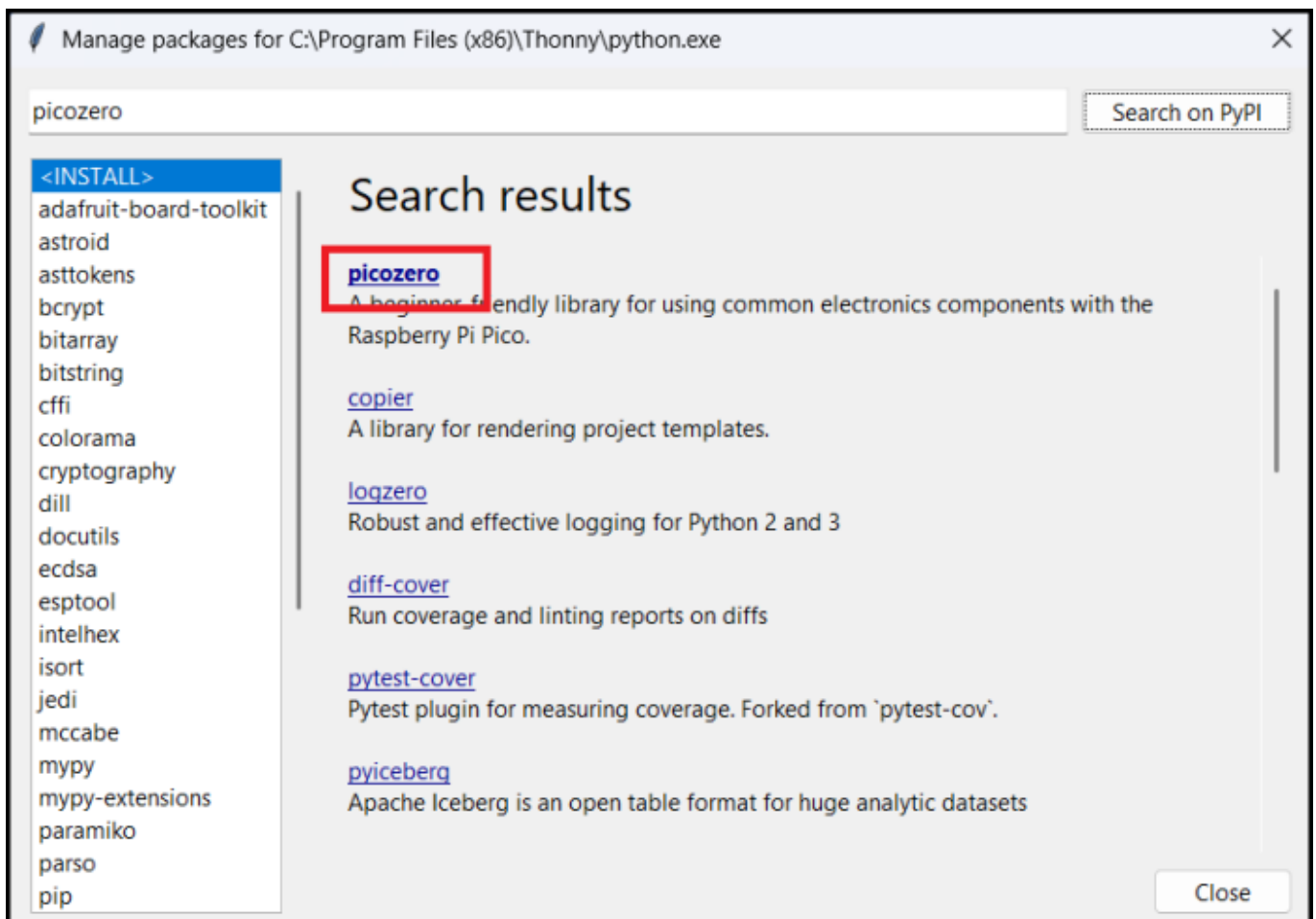
Next, from the Thonny toolbar, we are going to select Tools and then Manage Packages.



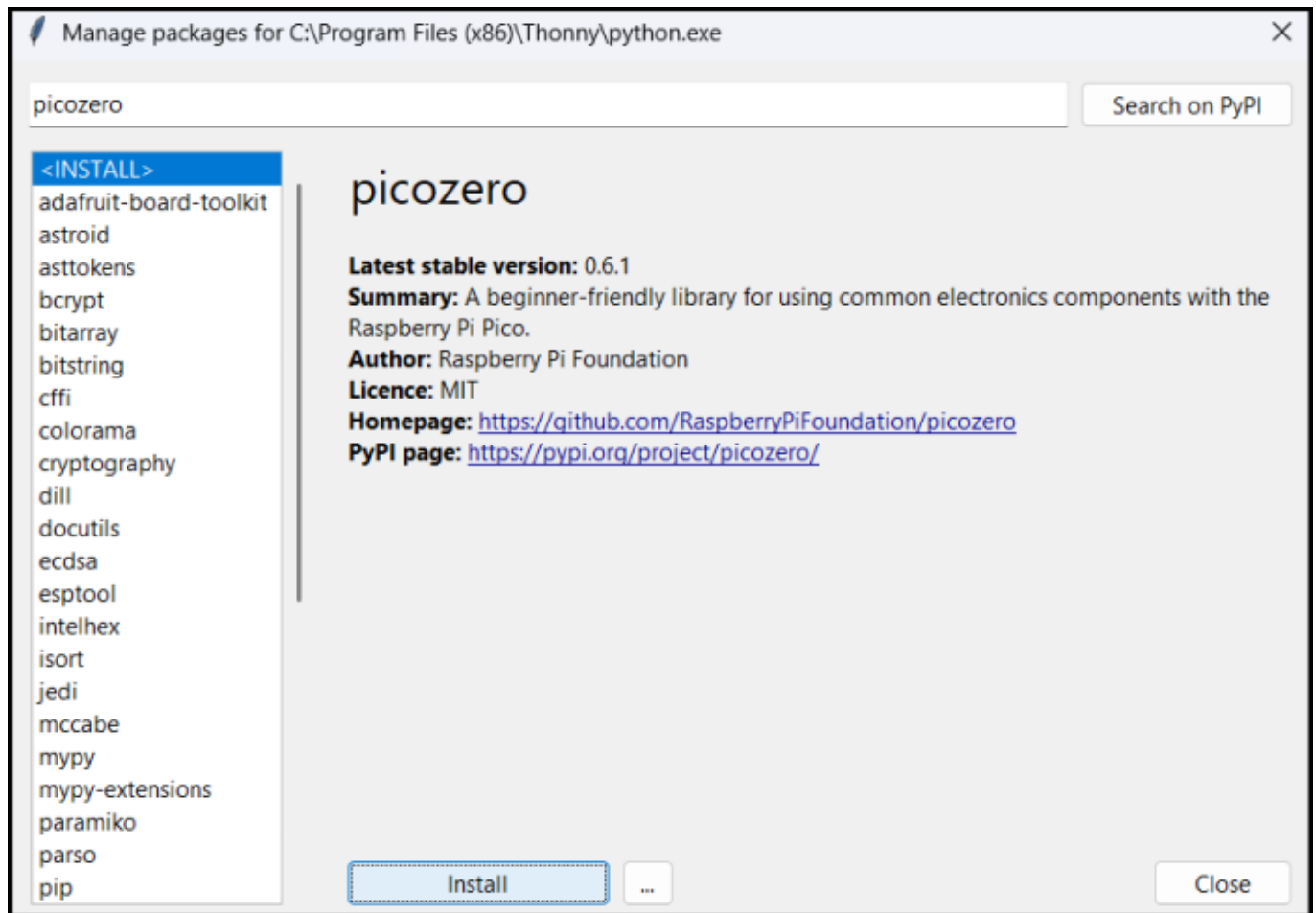
A box will open. In the search bar enter [picozero](#) then click the Search on PyPI button.



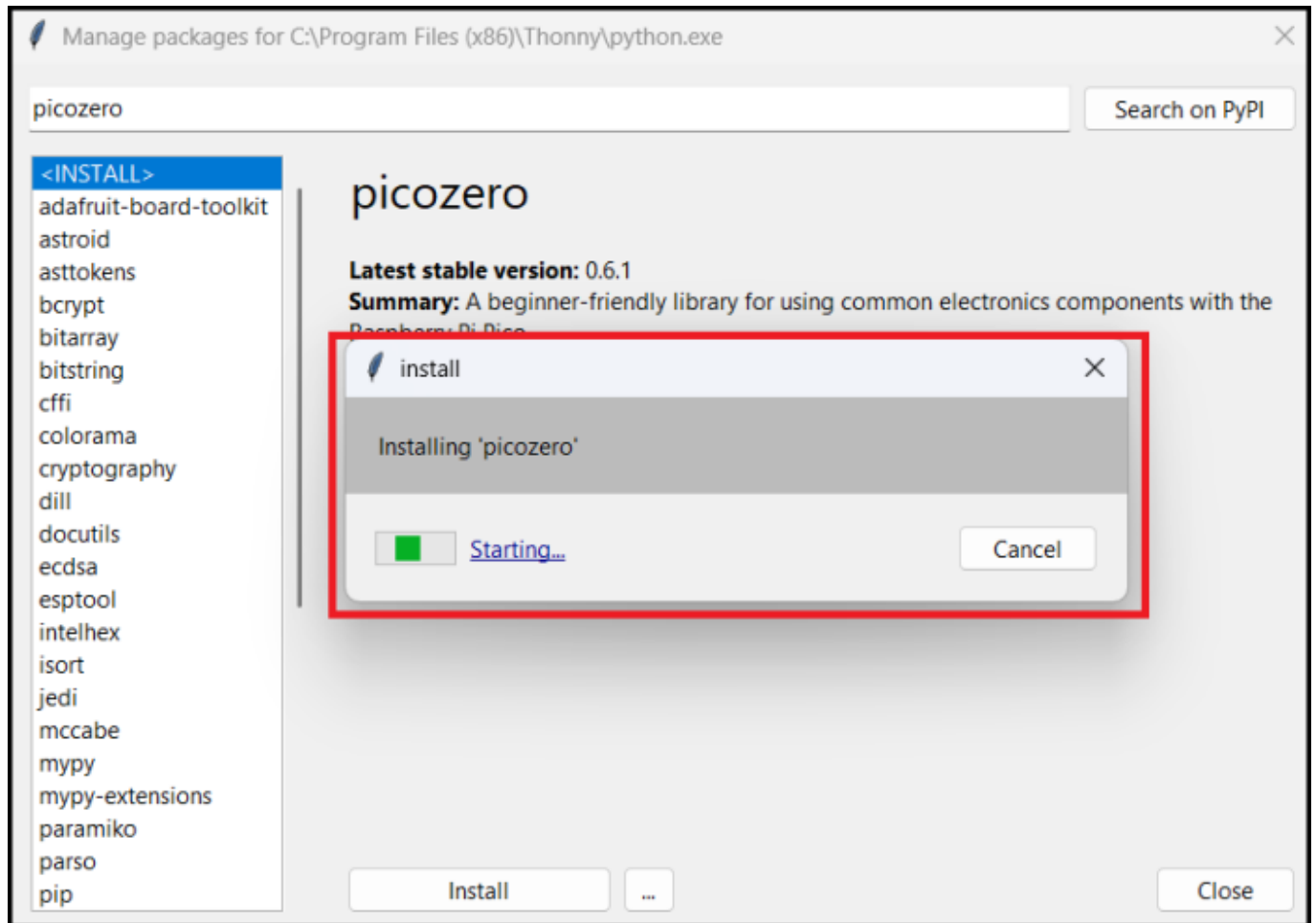
In the Search results, you should see the entry [picozero](#). Click on this entry (it's a hyperlink).



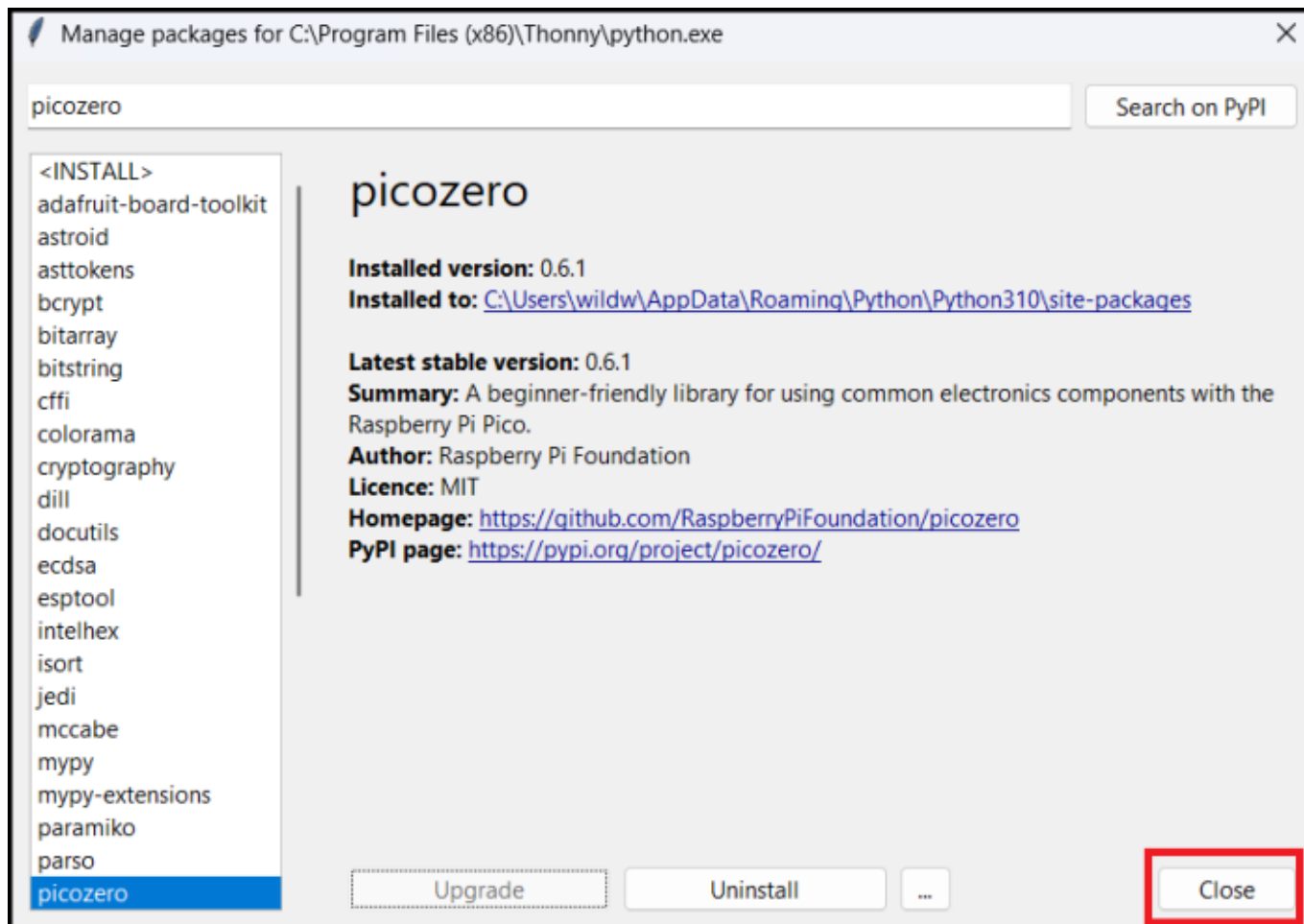
This will take you to the [picozero](#) page where you can click install.



You will see the picozero installer running.



Now you can click the Close button.



That is it, you have installed and done some basic configuration of Thonny.

So now you can now write some code.

For example when writing the code to flash or "blink" the onboard LED you can use:

[download](#)

```
from picozero import pico_led  
pico_led.blink(0.5)
```

This is a simple way and great to get started but it is limited, you can get the same result using other library's

From:

<https://wiki.alanwalker.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:

<https://wiki.alanwalker.uk/doku.php?id=thonny&rev=1781645361>

Last update: **2026/06/16 21:29**

