The CamJam EduKit Worksheets

CamJam EduKit 1 is the first kit in the CamJam EduKit series. You can download the worksheets, for free, at https://camjam.me/edukit.

The EduKits are compatible with all versions of Raspberry Pi Pico. You can also use EduKit 1 with all versions of the Raspberry Pi computer.

Notes and Assumptions

Throughout these worksheets, it is assumed that you are using a Windows PC, and you are editing your code with the Thonny code editor. Using an Apple Mac or any Linux computer, including a Raspberry Pi computer, is also possible.

You will eventually be using the “picozero” MicroPython Library to interact with the Pico’s GPIO pins. This is a user-friendly library that hides some of the complexities of working with GPIO pins.

You should not copy and paste the code from the worksheets themselves since the spacing at the beginning of lines will not necessarily be pasted correctly, and the code will not always work.

Indentation at the beginning of MicroPython code is important; it is MicroPython’s way of recognising how code should be grouped in ‘condition’, ‘loop’ and ‘function’ code blocks. Instead of cutting-and-pasting, type in the code or download it from GitHub using the instructions at the end of this worksheet.

It is assumed that your Pico has not been used before and that there are no useful files stored on the Pico.

CamJam EduKit – Worksheet 1

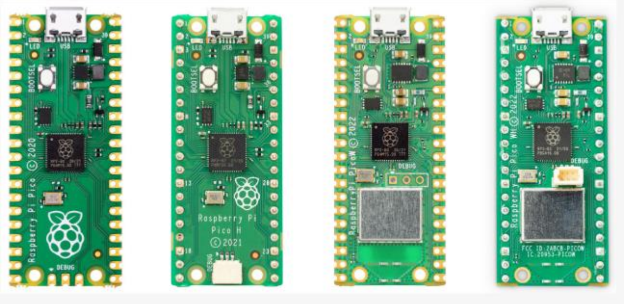
Project Setting up the Raspberry Pi Pico

Description Set up and run your first MicroPython program to print “Hello World” to the screen. You will not be connecting any of the contents of the CamJam EduKit to the Pico for this exercise.

Equipment Required

For the worksheets, you will require:

- A Raspberry Pi Pico with header pins. The Pico WH or H is ideal, but you can also solder your own pins to a Pico. There are numerous guides on how to do this on the internet.
- A Windows PC or a computer that can run the Thonny editor.
- A micro-USB data cable. Do not use a power-charging-only cable.
- CamJam EduKit 1 from The Pi Hut (https://thepihut.com/edukit).
- An additional 4 Male-Male (M-M) jumpers. These *may* be included in your EduKit if it was bought very recently. If they are not included, they are available from various sources including The Pi Hut: search for “male/male jumper wires”.



The Raspberry Pi Pico family currently consists of four boards; the original Raspberry Pi Pico (far left), the Pico H (middle left), the Pico W (middle right), and the Pico WH (far right).

The ‘H’ means the Pico has pre-soldered headers (pins), and ‘W’ means it has Wi-Fi and Bluetooth to enable it to connect to networks and other devices.

Setting up your Raspberry Pi Pico

The Raspberry Pi Foundation has excellent instructions for installing MicroPython onto the Pico, and Thonny onto your computer:

https://projects.raspberrypi.org/en/projects/introduction-to-the-pico/2

If you can access that page, follow their instructions, and skip down this worksheet to the ‘Writing Code’ section.

MicroPython is a version of Python that runs on small micro-controllers like the Pico. Within these worksheets it will sometimes be referred to as Python, since it is almost identical to the version that runs on computers such as the Raspberry Pi and Windows PCs.

Downloading MicroPython

You will need to download MicroPython for the Pico from the Raspberry Pi website. Visit the following link:

https://www.raspberrypi.com/documentation/microcontrollers/micropython.html

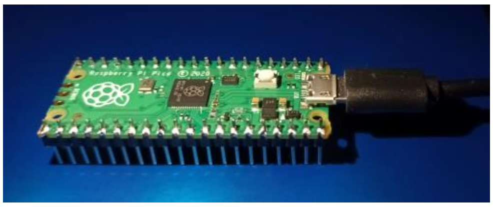
Download the appropriate version of the UF2 file to your computer.

NOTE: If your Pico does not have Wi-Fi you need to download the Raspberry Pi Pico version, otherwise choose the Raspberry Pi Pico W version. Look at the label on your Pico board to find out what you have.

Setting up the MicroPython Firmware on the Raspberry Pi Pico

When plugging in and unplugging the micro-USB cable, hold the Raspberry Pi Pico's micro-USB connector with two fingers, one on top of the connector and one on the underside of the board. Excessive force can accidentally break the connector off if you hold the board by its edges.

With the USB cable unplugged from your computer, plug the micro-USB end into the Pico.



Hold down the white 'BOOTSEL' button on the top of the Pico. While still holding it down, connect the other end of the micro-USB cable to your computer. Count to three, then let go of the 'BOOTSEL' button.

After a few seconds, you should see the Raspberry Pi Pico appear as a removable drive on your computer, as though you had connected a USB flash drive or an external hard drive.

Open two windows in the File Manager. In one, navigate to where you saved the UF2 file, and in the other to the Pico that appeared as a USB drive above.

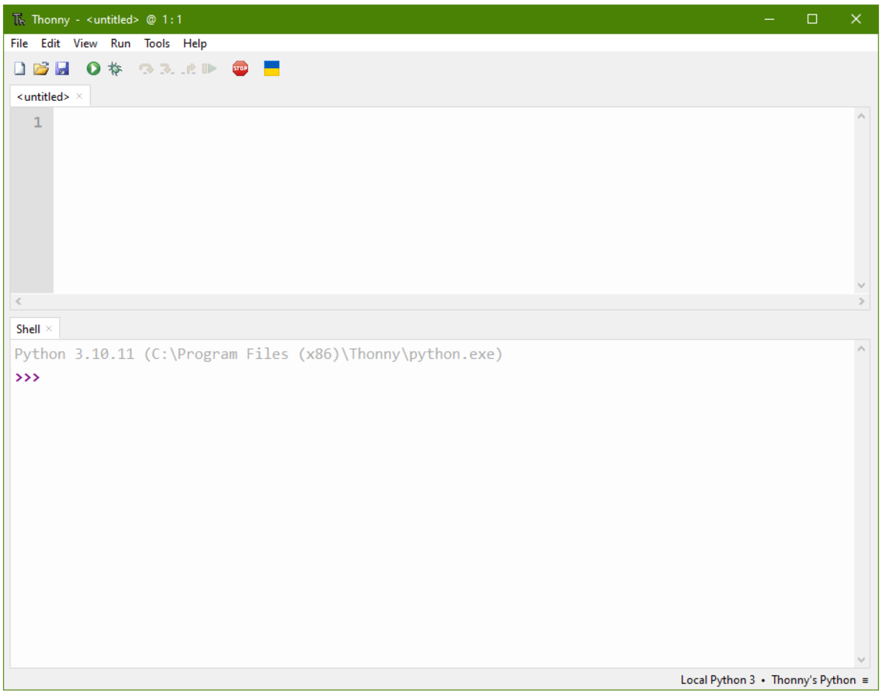
Click and hold the left mouse button on the UF2 file and drag it to the other File Manager window showing the Raspberry Pi Pico as a removable storage drive. When you release the mouse button to drop the UF2 file onto the Pico, the MicroPython firmware will be installed. After a few seconds the Raspberry Pi Pico will disappear from File Manager. You may also see a warning that a drive was removed without being ejected. Do not worry, that is supposed to happen!

Installing the Thonny Python editor on your computer

If you are using a Raspberry Pi as your computer, Thonny should already be installed.

If it is not installed, open https://thonny.org in your web browser. There are several versions suiting different computers. Download the appropriate version for the computer you are using and install it.

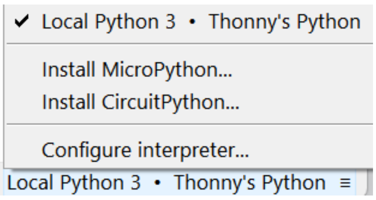
Open Thonny. It should look something like the picture below:



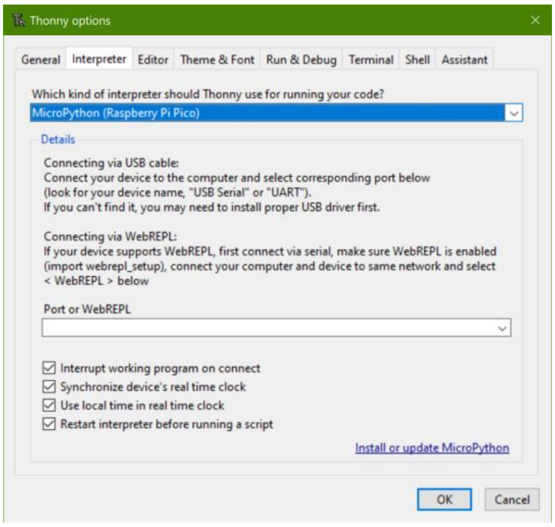
```
Python 3.10.11 (C:\Program Files (x86)\Thonny\python.exe)
>>>
```

On the bottom right corner, click the three horizontal bars:

- Select Configure Interpreter:



- Select 'MicroPython (Raspberry Pi Pico)':



Which kind of interpreter should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)

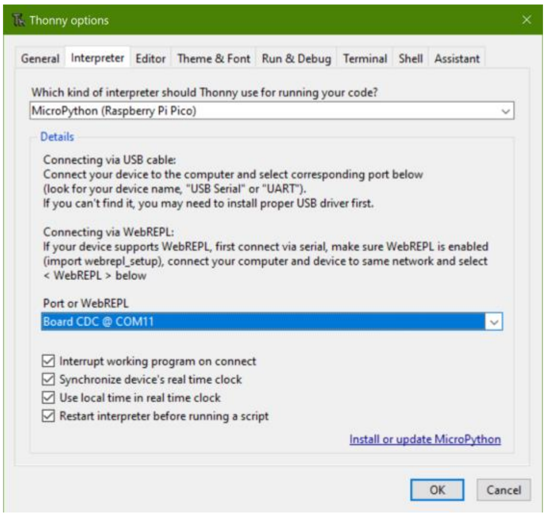
Connecting via USB cable:
Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Connecting via WebREPL:
If your device supports WebREPL, first connect via serial, make sure WebREPL is enabled
(import webrepl_setup), connect your computer and device to same network and select
< WebREPL > below

Port or WebREPL

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

- Select the COM port for the Raspberry Pi Pico under 'Port or WebRepl':



Which kind of interpreter should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)

Connecting via USB cable:
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.

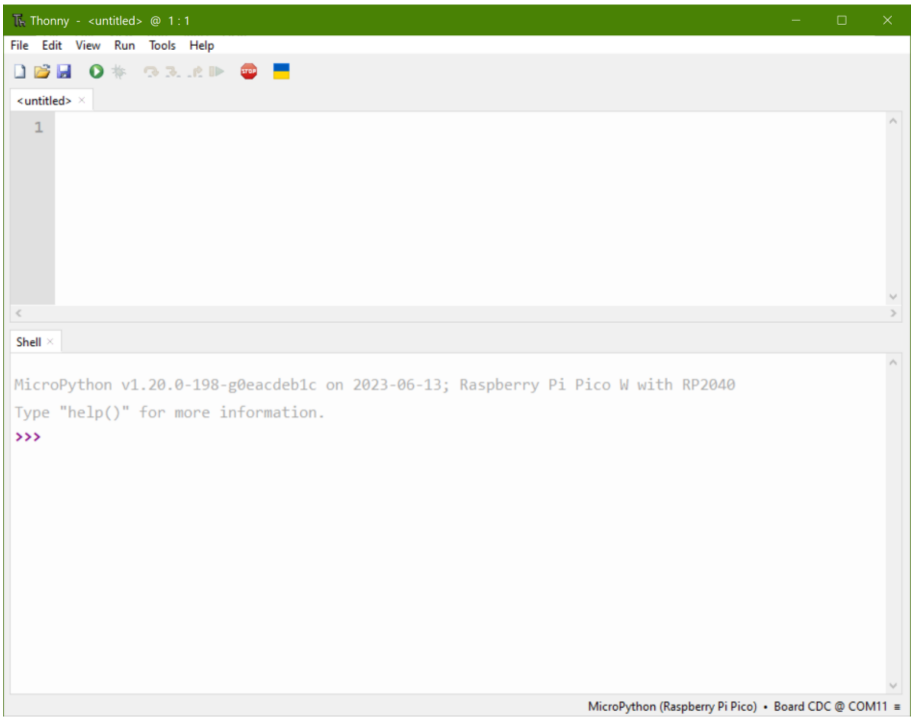
Connecting via WebREPL:
If your device supports WebREPL, first connect via serial, make sure WebREPL is enabled
(import webrepl_setup), connect your computer and device to same network and select
< WebREPL > below

Port or WebREPL
Board CDC @ COM11

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

- Click on OK

MicroPython will run and display version information in the Shell area. ‘Board CDC @ COM port’ will be shown in the bottom right corner. Your COM port number may differ.



```
MicroPython v1.20.0-198-g0eacdeb1c on 2023-06-13; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>
```

MicroPython (Raspberry Pi Pico) • Board CDC @ COM11 ≡

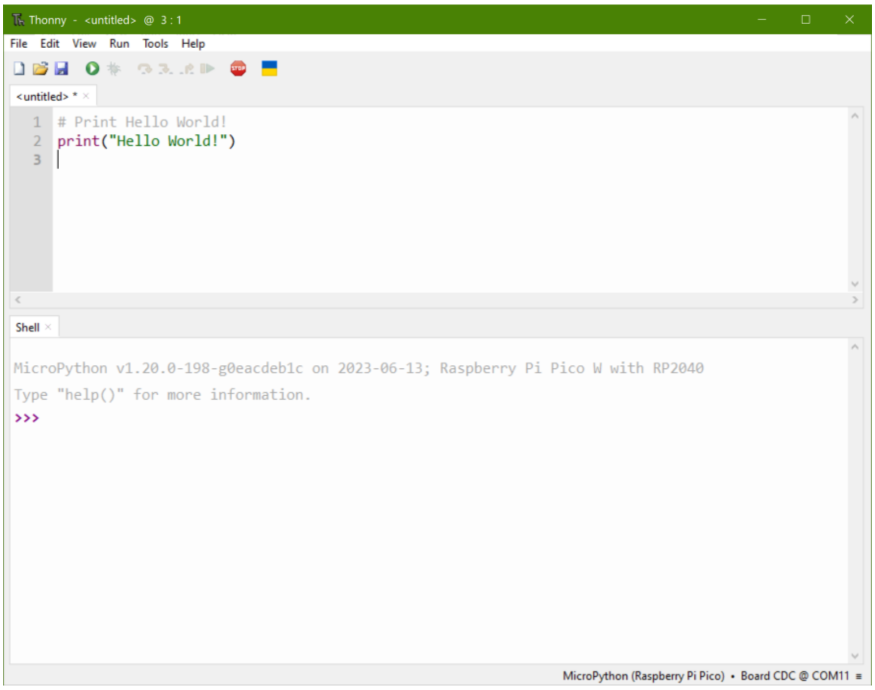
Writing Code

You are now going to create your first piece of MicroPython code which will simply print "Hello World" to the screen.

Thonny will have already opened an empty file named <untitled>. Type the following code exactly as shown into the Script Area of Thonny:

```
# Print Hello World!
print("Hello World!")
```

Thonny should look like this:



Thonny - <untitled> @ 3 : 1
File Edit View Run Tools Help
<untitled> *
Shell

```
# Print Hello World!
print("Hello World!")
```

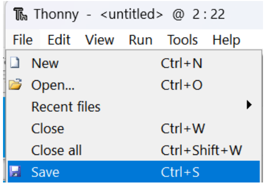
```
MicroPython v1.20.0-198-g0eacdeb1c on 2023-06-13; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>
```

MicroPython (Raspberry Pi Pico) • Board CDC @ COM11

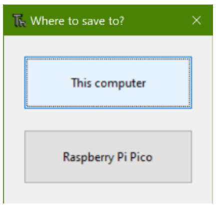
Everything on the same line after a '#' is a 'comment' that the coder has written to give useful information to the reader and will be ignored by MicroPython.

Saving your code

In Thonny, navigate the menu to File, Save.



A window will appear asking you where you want to save the file to. Choose 'This computer':



Create a new folder called 'EduKit 1' on your computer. Save the file in the computer folder, calling the file `1-helloworld.py`.

Running the Code

Click the green Run icon on top menu bar. Alternatively, select the 'Run current script' menu option, under the Run menu. Or you can just press the F5.

The file will be saved to your Pico where it will execute your code.

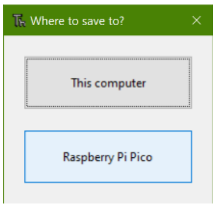
You will see "Hello World!" printed to the Shell section of Thonny.

If you find that the code does not run correctly, there may be an error in the code you have typed. Check and edit the code, save again, and re-run it.

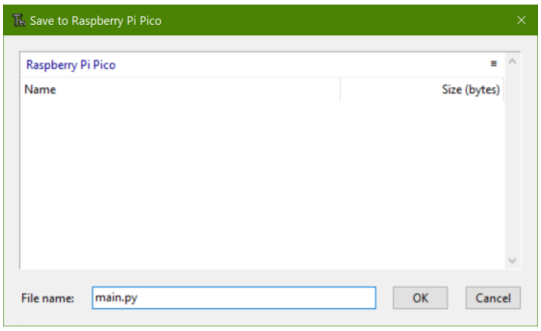
Running the code when the Pico is plugged in

Your Pico can run the code saved to it every time it is plugged in. This is easy to achieve simply by saving your code with the special name '`main.py`'.

You can try this by re-saving the Hello World example. Select 'Save as…' from the File menu. This time, select 'Raspberry Pi Pico':



Save the file as '`main.py`':



File name: main.py

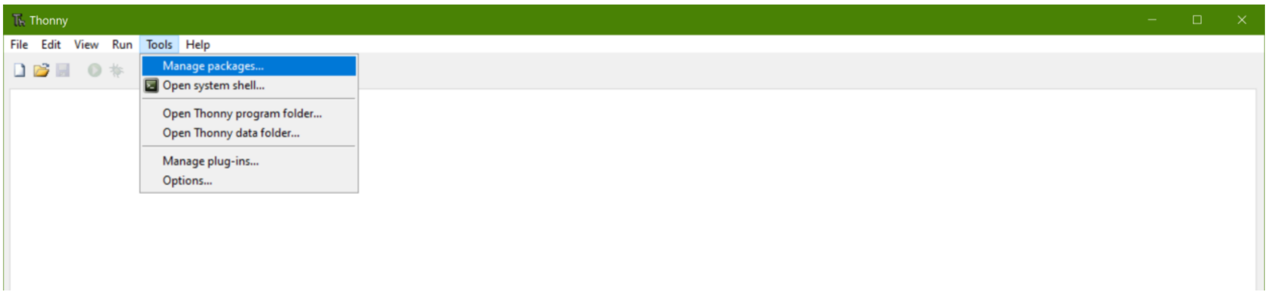
Now unplug your Pico and plug it in again. The code will run, although you will not see anything on screen.

NOTE: If your code runs forever, or is in a loop, you may not be able to run your code again unless you click on the 'Stop' icon.

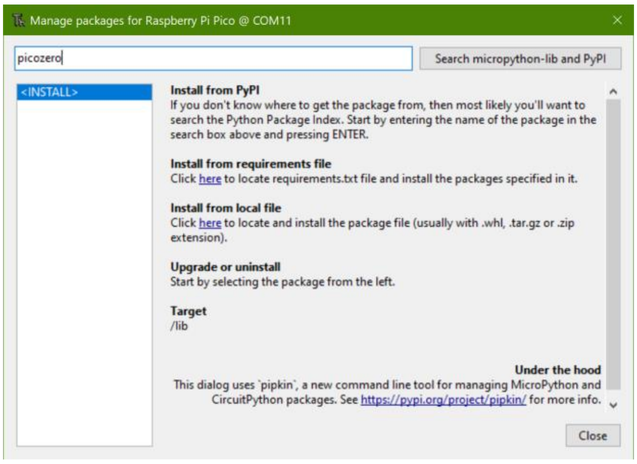
Install the "picozero" library

You will need to install the "picozero" library before you proceed with the rest of the worksheets. 'picozero' is a 'library' of pre-written code that you can run from within your own code. Libraries provide you with extra functionality which you would otherwise have to write yourself. This saves you both time and the need to learn much harder code.

- In Thonny, choose ‘Tools > Manage packages…’.


Thonny
File Edit View Run Tools Help
Manage packages...
Open system shell...
Open Thonny program folder...
Open Thonny data folder...
Manage plug-ins...
Options...

- In the pop-up window titled ‘Manage packages for Raspberry Pi Pico’, type ‘picozero’ and click ‘Search micropython-lib and PyPI’:


Manage packages for Raspberry Pi Pico @ COM11
picozero
Search micropython-lib and PyPI
<INSTALL>
Close

Install from PyPI
If you don't know where to get the package from, then most likely you'll want to search the Python Package Index. Start by entering the name of the package in the search box above and pressing ENTER.

Install from requirements file
Click here to locate requirements.txt file and install the packages specified in it.

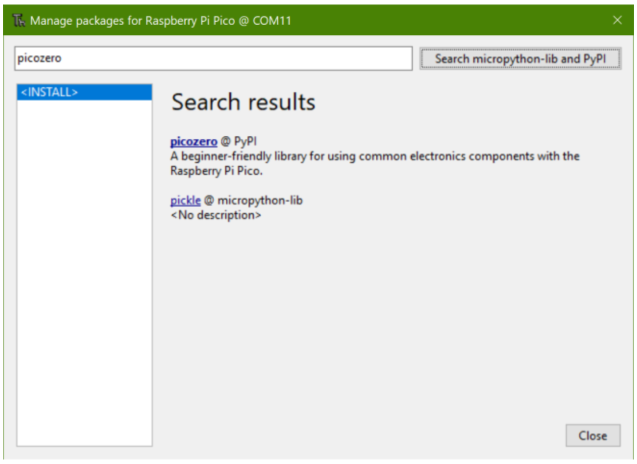
Install from local file
Click here to locate and install the package file (usually with .whl, .tar.gz or .zip extension).

Upgrade or uninstall
Start by selecting the package from the left.

Target
/lib

Under the hood
This dialog uses `pipkin`, a new command line tool for managing MicroPython and CircuitPython packages. See https://pypi.org/project/pipkin/ for more info.

- Click on ‘picozero’ in the search results:

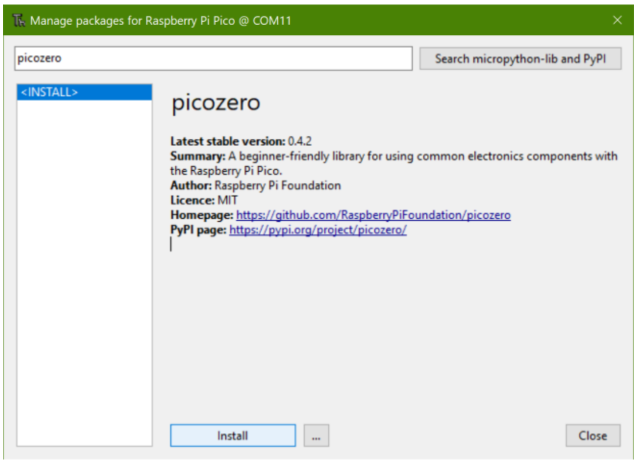

Manage packages for Raspberry Pi Pico @ COM11
picozero
Search micropython-lib and PyPI
<INSTALL>
Close

Search results

picozero @ PyPI
A beginner-friendly library for using common electronics components with the Raspberry Pi Pico.

pickle @ micropython-lib
<No description>

- Then click on 'Install':



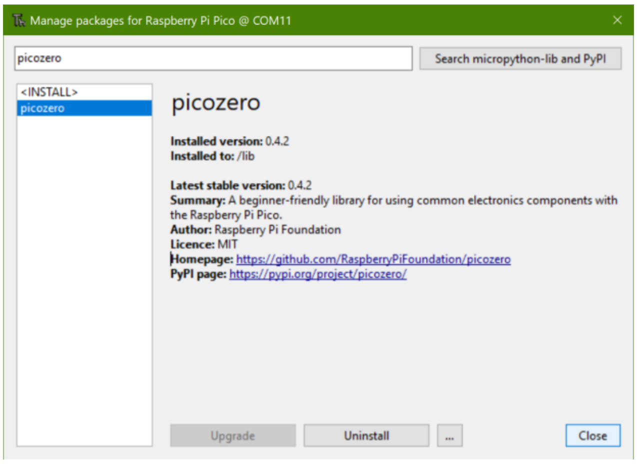
Manage packages for Raspberry Pi Pico @ COM11

picozero

picozero

Latest stable version: 0.4.2
Summary: A beginner-friendly library for using common electronics components with the Raspberry Pi Pico.
Author: Raspberry Pi Foundation
Licence: MIT
Homepage: https://github.com/RaspberryPiFoundation/picozero
PyPI page: https://pypi.org/project/picozero/

- Wait for the library to be downloaded and installed, then click on 'Close':



Manage packages for Raspberry Pi Pico @ COM11

picozero

picozero

Installed version: 0.4.2
Installed to: /lib

Latest stable version: 0.4.2
Summary: A beginner-friendly library for using common electronics components with the Raspberry Pi Pico.
Author: Raspberry Pi Foundation
Licence: MIT
Homepage: https://github.com/RaspberryPiFoundation/picozero
PyPI page: https://pypi.org/project/picozero/

Download the EduKit Worksheets and Code from GitHub

The code written for the CamJam EduKit, and listed in the worksheets, is also stored on GitHub. Follow these instructions to download.

GitHub Repository

All the code and worksheets for each EduKit can be found online at https://github.com/CamJam-EduKit. You may download individual files or whole "repositories" from there.

There are multiple versions of the code and worksheets, depending on whether you are using a Raspberry Pi computer or the Raspberry Pi Pico. You should download the Pico version.