

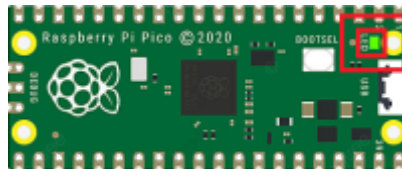
LED On & Off

Jun 2026



Introduction

The Raspberry Pi Pico actually has an onboard LED that we can use for some simple Micro Python testing. Now one slight complication here is that the original Pico and the Pico 2 W use different ways to address the LED. This guide is using the Pico, where the LED is on GPIO pin 25. The way we will address the LED means that you don't really need to know the pin number.



In the above picture we can see where the LED is. Once powered on the LED by default will be off, and we will need to use some micro python to turn it on.

Lets run Thonny, and use some Micro Python code to turn on and turn off our onboard Pi Pico LED.

A screenshot of the Thonny IDE interface. The title bar reads 'Thonny - Raspberry Pi Pico :: /led_on.py @ 6:1'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Tools', and 'Help'. The toolbar shows icons for file operations, running, and debugging. The editor window shows a file named 'led_on.py' with the following code:

```
1 # Import the piczero LED class
2 from piczero import pico_led
3
4 #Turn on the pico LED
5 pico_led.on()
6
```

The Shell window at the bottom shows the MicroPython environment: 'MicroPython v1.28.0 on 2026-04-06; Raspberry Pi Pico with RP2040', 'Type "help()" for more information.', and a prompt '>>>>'. The status bar at the bottom right indicates 'MicroPython (Raspberry Pi Pico) • Board CDC @ COM4'.

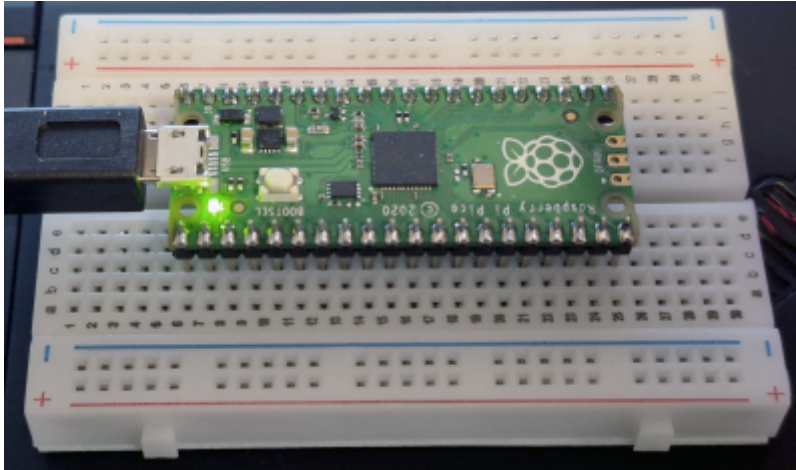
Here we can see the code that we need to enter in to Thonny.

[download](#)

```
# Import the picozero LED class
from picozero import pico_led

#Turn on the pico LED
pico_led.on()
```

To run this code, click the green play button on the Thonny toolbar. This of course turns the Pi Pico LED on.



This code does a couple of things:

from picozero import pico_led

This line imports the `pico_led` object from the `picozero` library. `picozero` is a beginner-friendly library for controlling hardware on the Raspberry Pi Pico. `pico_led` is a built-in object that represents the on-board LED (the little LED already on the Pico board).

So this line is basically saying: "Give me access to the Pico's built-in LED so I can control it."

pico_led.on()

This is the line that actually turns the LED on. `pico_led` is the LED object. `.on()` is a function that switches it on. The LED lights up and stays on until you turn it off or reset the board. It's the simplest possible way to control hardware on the Pico.

Now that you have turned the LED on, you might want to turn it off, and the program is almost the same except for the argument for the `pico_led` line.

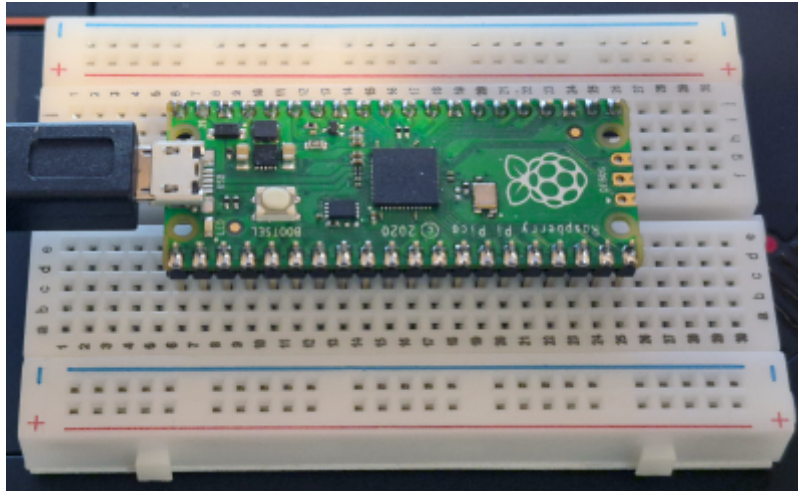
[download](#)

```
# Import the picozero LED class
from picozero import pico_led

#Turn on the pico LED
pico_led.off()
```

As you can see in the code above, all we have done is to change the `pico_led.on()` to `pico_led.off()`

This now turns off the Pi Pico LED.



That is it, we can now successfully turn on and off the onboard LED on the Pi Pico, as well as creating our first piece of Micro Python in Thonny.

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

Permanent link:
https://wiki.alanwalker.uk/doku.php?id=led_on_and_off&rev=1781471019

Last update: **2026/06/14 21:03**

