

OnBoard NeoPixel

Jun 2026



Introduction

The RP2040-Zero mini has an onboard NeoPixel. So we cannot use the Raspberry Pi Pico code to use this LED. Instead we need to use some Micro Python specific to the Waveshare (or compatible) device.

The NeoPixel is situated between the Boot and Reset buttons.



As this is a NeoPixel, we need to provide the RGB value for it to work, this is the main difference between the RPi Pico. We create RGB value, write it to memory and then turn on the LED.

NeoPixel Blink Code

What the code below does is simple turn on and off the NeoPixel, first we can see the Micro Python code, there are no comments so as not to create too much clutter, but the code with comments is below.

[| download](#)

```
from machine import Pin
import neopixel
import time

LED_PIN = 16

np = neopixel.NeoPixel(Pin(LED_PIN), 1)

while True:

    np[0] = (255, 0, 0)
    np.write()
    time.sleep(0.5)

    np[0] = (0, 0, 0)
    np.write()
    time.sleep(0.5)
```

First we import three libraries - Pin, neopixel and time.

We set the LED Pin to be 16 (this is the GPIO Pin that the NeoPixel uses).

Next we create a NeoPixel (WS2812/WS2812B) LED controller object, telling MicroPython:

- Which pin the NeoPixel data line is connected to (16).
- How many LEDs are in the strip (in this case, just 1).

We start a loop

- Set NeoPixel value - np[0] = (255, 0, 0) (this is red)
- Write the value - np.write()
- Wait half a second - time.sleep(0.5)

We then write zeros to the NeoPixel (turn it off) and wait half a second and repeat.

NeoPixel Blink Code with Comments

[| download](#)

```
# Import the Pin class from the machine module.
# This allows us to control the RP2040's GPIO (General Purpose Input/Output) pins.
from machine import Pin

# Import the neopixel module.
# This provides functions for controlling WS2812 (NeoPixel) RGB LEDs.
import neopixel

# Import the time module.
# We'll use this to create delays with sleep().
import time

# The GPIO pin connected to the data input of the WS2812 LED.
# On many Waveshare RP2040 boards this is GPIO 16.
LED_PIN = 16

# Create a NeoPixel object.
#
# Pin(LED_PIN) tells MicroPython which GPIO pin is connected
# to the LED.
#
# The second parameter (1) tells it there is only ONE RGB LED
# connected to this pin.
np = neopixel.NeoPixel(Pin(LED_PIN), 1)

# Start an infinite loop.
# Everything inside this loop will repeat forever.
while True:

    # Set LED number 0 (the first and only LED)
    # to the colour Red.
    #
    # Colours are stored as:
    # (Red, Green, Blue)
    #
    # Each value can range from:
    # 0    = Off
    # 255 = Maximum brightness
    #
    # So this means:
    # Red   = 255 (full brightness)
    # Green = 0
```

```
# Blue = 0
np[0] = (8, 0, 8)

# Send the colour data to the LED.
#
# Changing np[0] only changes the value in memory.
# The LED won't actually change until write() is called.
np.write()

# Wait for half a second.
time.sleep(0.5)

# Set the LED colour to black (off).
#
# Since all three colours are zero,
# the LED turns off.
np[0] = (0, 0, 0)

# Update the LED so it turns off.
np.write()

# Wait another half second before repeating.
time.sleep(0.5)

#other colours
#(255, 0, 0),    # Red
#(0, 255, 0),    # Green
#(0, 0, 255),    # Blue
#(255, 255, 0),  # Yellow
#(255, 0, 255),  # Magenta
#(0, 255, 255),  # Cyan
#(255, 255, 255),# White
#(0, 0, 0)       # Off
```

From:

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

Permanent link:

https://wiki.alanwalker.uk/doku.php?id=onboard_neopixel&rev=1784467459

Last update: 2026/07/19 13:24

