

Using Pico with SSD1306 based LCD and SPI

Aug 2026



Introduction



So, I found another 0.96" LCD to play with, just lying around at home. I figured it would be a bit rude to not give it a go. It is however, slightly different to the one we previously setup.

About the LCD:

- It is still an ssd1306 based display, got this from the label on the packaging.
- It does support I2C, 3 wire SPI and 4 wire SPI. The default resistor settings put it at 4 wire SPI.
- SPI will use 7 wires instead of the 4 wires for I2C.
- SPI has faster refresh than I2C.
- Same 128×64 resolution.
- This version of the LCD does not have the Yellow Blue split.

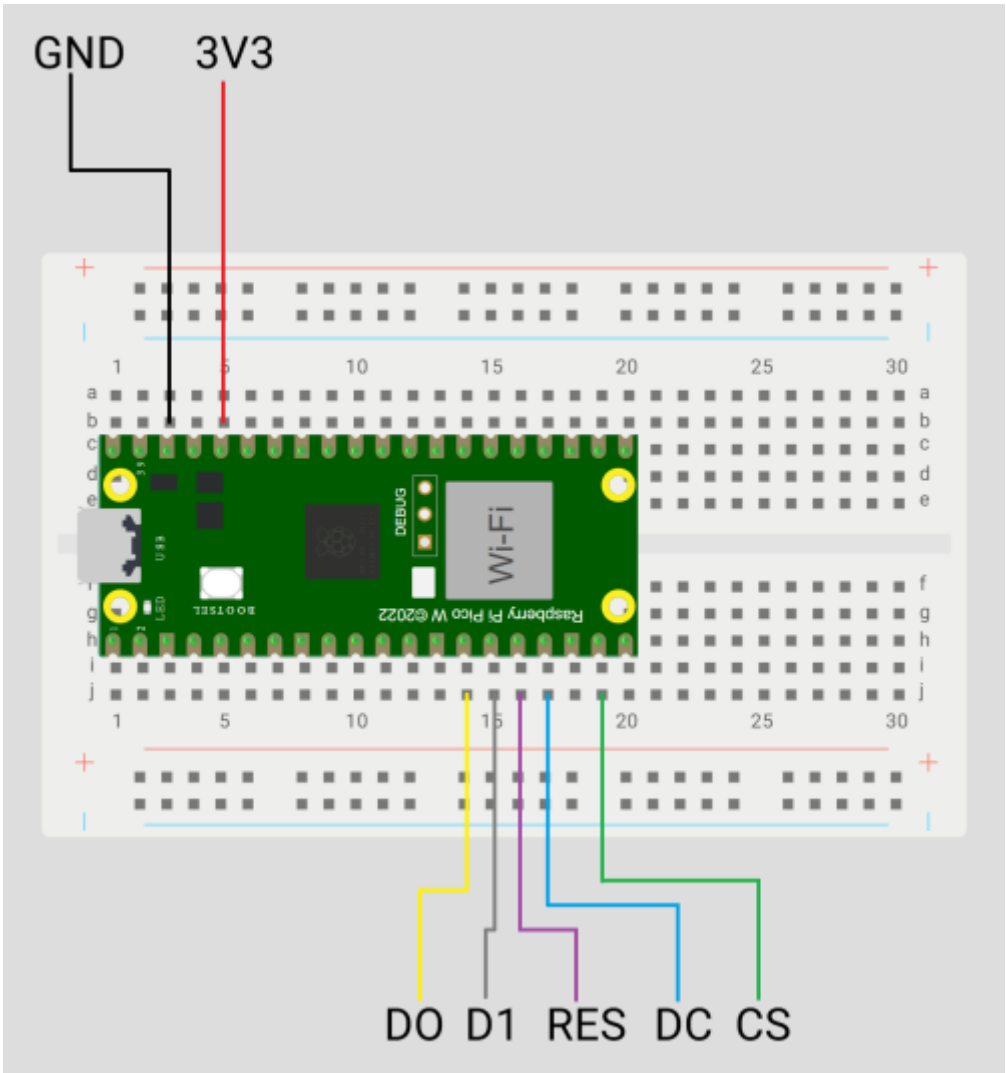
Because we are using SPI, the Pico will need a new driver as the original driver is for I2C. Before we get in to the Python code, let's have a look at the wiring.

The Wiring

Please see below for the SPI wiring to the Pico Pins.

OLED Pin	Connect To (Pico 2 W)	Function
GND	GND	Ground
VCC	3V3	Power
D0	GP10	SPI SCK
D1	GP11	SPI MOSI
RES	GP12	Reset
DC	GP13	Data/Command
CS	GP14	Chip Select

Below is a more diagrammatic view.



Once you have cabled up your LCD, we can move on to the software.

Python Code

Before we look at the code, I want to share that there are a few differences between SPI and I2C, have a look at this table below.

Feature	I ² C OLED	SPI OLED
Device addressing	Yes (0x3C, 0x3D)	No
How device is selected	Address on bus	CS pin
Bus type	Shared	Point-to-point
Pins	4	7
Speed	Slower	Faster

So SPI is faster, but there is no addressing, so you cannot chain multiple SPI devices from a single set of Pico GPIO Pins. If you need to run multiple displays, your choice is to either use I2C LCDs, or use multiple sets of SPI pins for each LCD.

The Driver for SPI (1306 driver chip)

Below is the driver you will need for the LCD. You need to put it in to the **lib** folder on your Pico.

[| download](#)

```
from machine import Pin, SPI
from ssd1306_spi import SSD1306_SPI
import time

# SPI1 setup
spi = SPI(1, baudrate=8000000, polarity=0, phase=0)

dc = Pin(13, Pin.OUT)
res = Pin(12, Pin.OUT)
cs = Pin(14, Pin.OUT)

oled = SSD1306_SPI(128, 64, spi, dc, res, cs)

oled.fill(0)
oled.text("Hello Alan!", 0, 0)
oled.text("SSD1306 SPI OK", 0, 16)
oled.text("Pico 2 Ready", 0, 32)
oled.show()

# call the function below from your main code with the value 255 (max brightness)
oled.set_brightness(255)
```

==== LCD Test Code —

I want to look at the old I2C code and compare to the SPI code to point out some differences, this is the Python code that just prints 8 static lines on the LCD for a simple test.

Here is the original I2C code

[| download](#)

```
from machine import Pin, I2C    #Gives access to the I2C Pins. Without it, you couldn't use GPIO pins or I2C.
import ssd1306
import time

i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=200000)
lcd = ssd1306.SSD1306_I2C(128, 64, i2c)

lcd.fill(0)
lcd.text("Yo Tuesday Club!", 0, 0)
lcd.text("SSD1306 LCD", 0, 8)
lcd.text("Address 0x3C", 0, 16)
lcd.text("Cool :)", 0, 24)
lcd.text("Another Line :)", 0, 32)
lcd.text("One More? :)", 0, 40)
lcd.text("Line 7", 0, 48)
lcd.text("Line 8", 0, 56)
lcd.show()

time.sleep(300)
lcd.fill(0)
lcd.show()
```

Here is the same code, but modified to run the SPI LCD Screen.
Use this code for your SPI LCD Display.

[| download](#)

```
from machine import Pin, SPI    #Gives access to the SPI Pins. Without it, you couldn't use GPIO pins or SPI.
from ssd1306_spi import SSD1306_SPI
import time

spi = SPI(1, baudrate=8000000, polarity=0, phase=0)

dc = Pin(13, Pin.OUT)
res = Pin(12, Pin.OUT)
cs = Pin(14, Pin.OUT)

oled = SSD1306_SPI(128, 64, spi, dc, res, cs)

oled.fill(0)
oled.text("Yo Tuesday Club!", 0, 0)
oled.text("SSD1306 oled", 0, 8)
oled.text("Address 0x3C", 0, 16)
oled.text("Cool :)", 0, 24)
oled.text("Another Line :)", 0, 32)
oled.text("One More? :)", 0, 40)
oled.text("Line 7", 0, 48)
oled.text("Line 8", 0, 56)
oled.show()

time.sleep(300)
oled.fill(0)
oled.show()
```

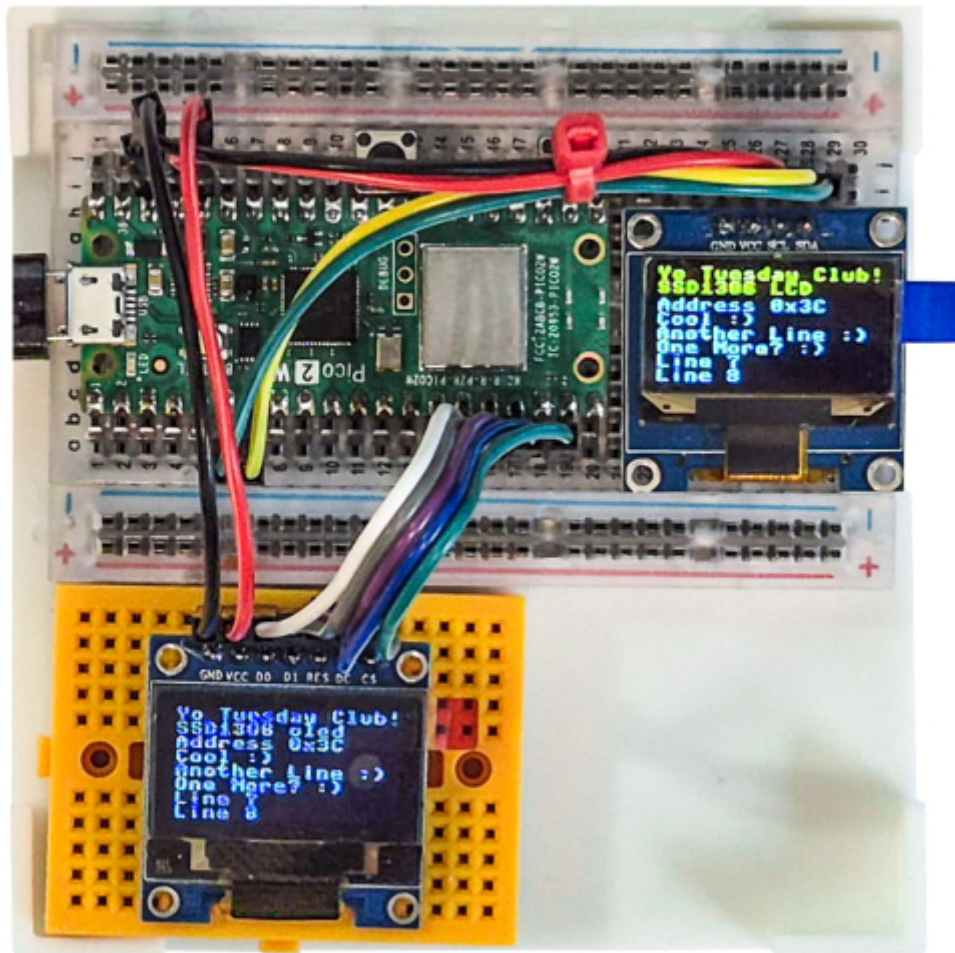
The second line of each Python script loads some code, there is some difference in how they work:

- **import ssd1306** - Loads the entire module, and you have access to all classes within that module.
- **from ssd1306_spi import SSD1306_SPI** - Loads just the SSD1306_SPI class from the module ssd1306_spi.
- **In the first example we used the name lcd** - lcd = ssd1306.SSD1306_I2C(128, 64, i2c)
- **In the second example we used the name oled** - oled = SSD1306_SPI(128, 64, spi, dc, res, cs)

The rest is pretty much the same, from a code perspective there really isn't much difference in the amount of work to get the I2C or SPI LCD displays working.

The Running LCDs

Below we can see both the I2C LCD (on the right) and the SPI LCD (Below the Pico).



They both work really well and for simple text based applications will be perfect.

From:

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

Permanent link:

https://wiki.alanwalker.uk/doku.php?id=using_pico_with_ssd1306_based_lcd_and_spi

Last update: 2026/08/24 20:01

