ST7735 display – Set up guide.

This is a cheap 1.8 inch colour display with a backlight.

Getting Started:

Unlike most other recent guides, this guide uses **circuit python**.

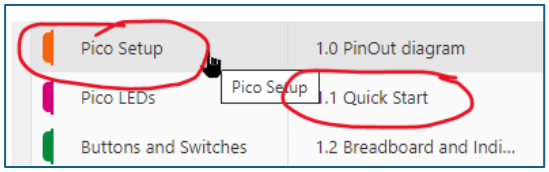
Your Pico device needs to have circuit python installed on it (not micropython).

Circuit Python is installed the same way you set up Micropython.

See the quick start guide

(in the Pi Club Tuts notebook)

if you need to remind yourself how to set up up the Pico



Plug in the Pico device to a PC.

Open up the Thonny app.

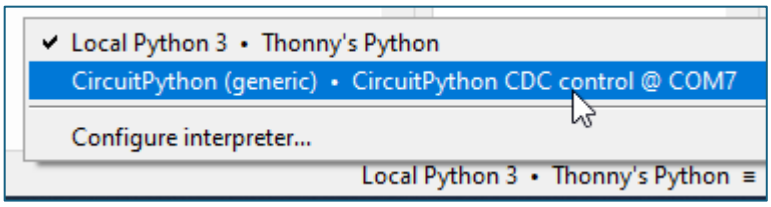
In Thonny, click in the bottom right corner.

A new and unused Pico will ask you whether you want to install MicroPython or CircuitPython.

If the Pico already has Micro Python installed, hold down the BootSel button on the Pico, before you connect it to the PC.

This should allow you to to install Circuit Python.

Once Circuit Python has installed you should be able to click in the bottom right corner of Thonny and select this option.



Circuit Python code is very similar to micro python code.

Use the test code below to check your Pico is connected to the PC

Test Code – make on board green LED flash

```
import board
import digitalio
import time

led = digitalio.DigitalInOut(board.LED)
led.direction = digitalio.Direction.OUTPUT

for x in range(4):
    led.value = True
    time.sleep(0.5)
    led.value = False
    time.sleep(0.5)
```

Assuming the LED did flash, we can now move onto wiring up the screen.

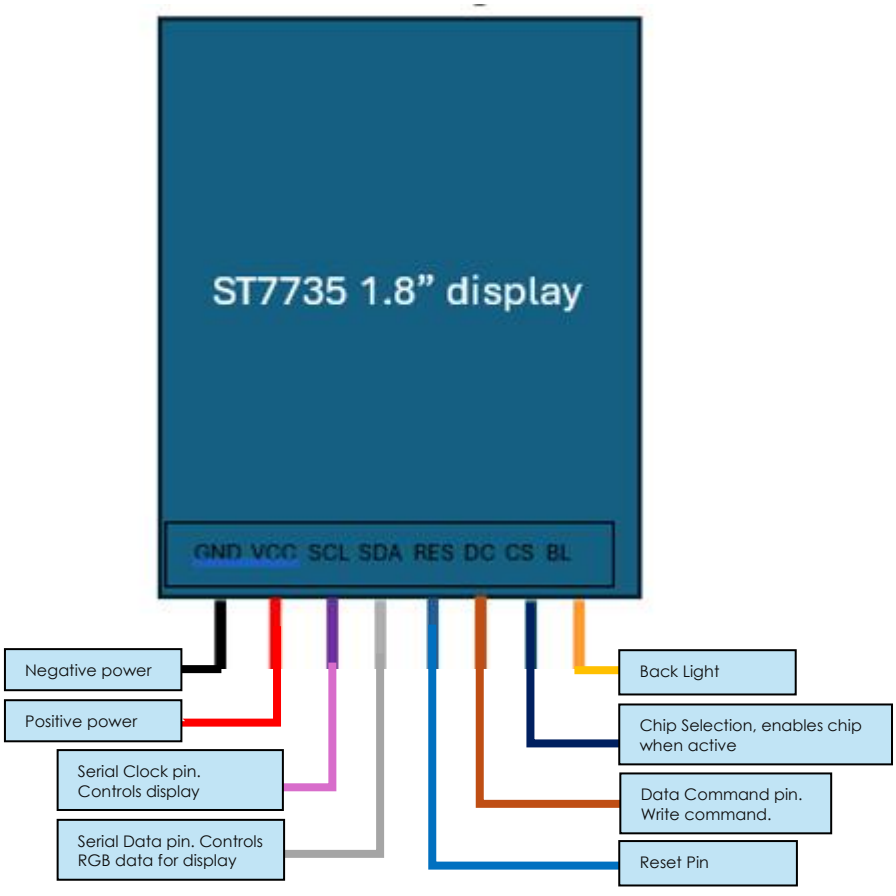
Hardware Setup

There are several wires and connections to deal with, but this is not hard to implement.

ST7735 Pins explained:

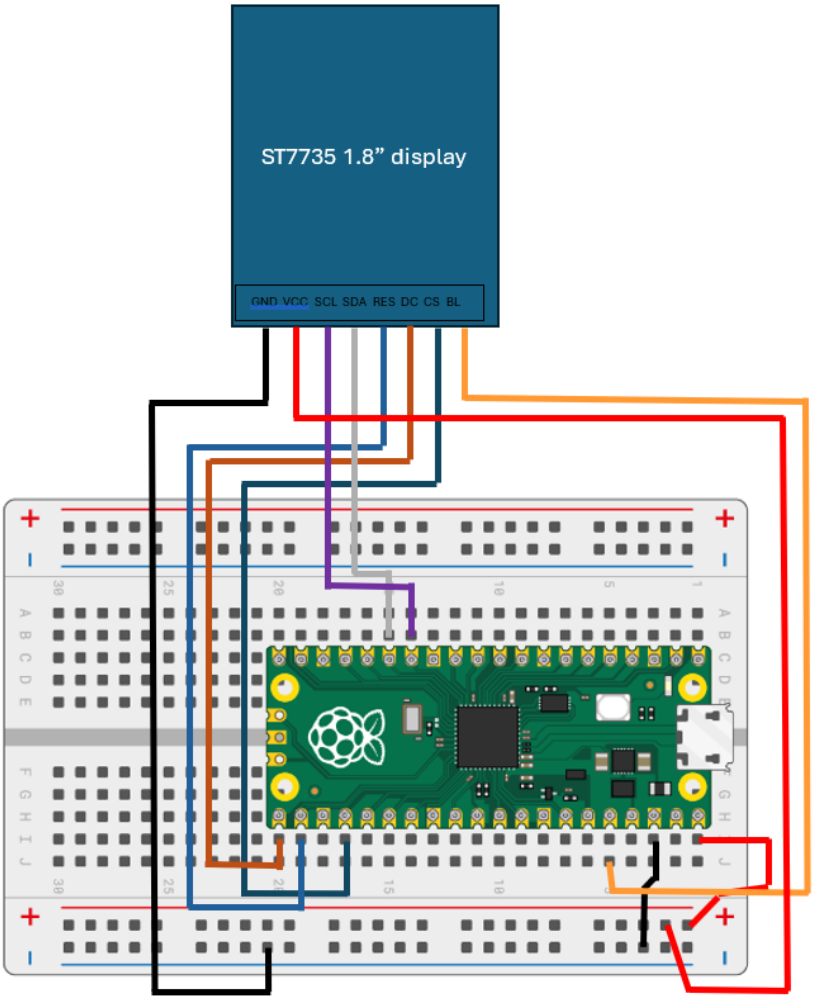
There are quite a few wires but be methodical and it should all go smoothly.

The ST7735 display has the following pins. You do not need to know exactly what they do, but it is nice to have a reference as you get used to using this screen.



Wiring Diagram

This is how to wire the ST7735 screen to a Pico using a breadboard.



This table identifies the Pico pins by number.

Display Pins	Pico Pins	Position on Pico
GND	GND	Anywhere on GND rail
VCC	5v	Anywhere on + rail
SCL	GP10	7 in from top left.
SDA (or MOSI)	GP11	6 in from top left.
RES	GP17	One in from bottom left.
DC (or AO)	GP16	Bottom left
CS	GP18	4th in from bottom left.
BL (or LED)	3.3v	5th Pin in from bottom right.

BL stands for Backlight and is labelled LED on some displays.

Software Setup

We need to download and install a few libraries to run this screen.

When we used the LED strips we had to install a neopixel library.

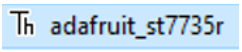
To use this display, we need several libraries installed on the Pico.

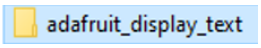
Download and install libraries

The following files can be downloaded from the Pi Club Tuts OneNote page.

Download them into your own area.

This file and folder need to be copy/pasted into the "lib" folder on the pico:





Open file explorer.

Browse to where you downloaded the above files.

Copy the Adafruit_st7735r file

Browse to the Pico device connected to your PC

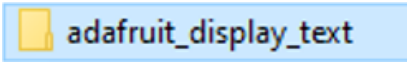
It looks like this

Go into the "lib" folder on the pico

Paste in the file you copied.

Locate the "Adafruit_display_text" folder you have already downloaded (and unzipped).

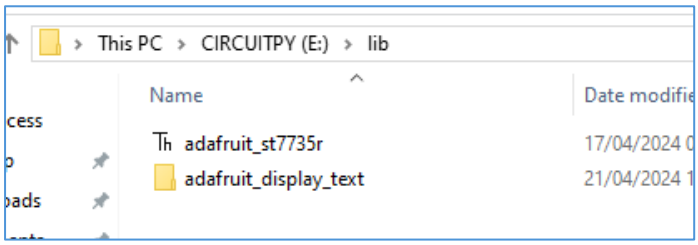
You want to copy the whole folder.



Copy the whole folder.

Browse to the "lib" folder on the pico and paste in the copied folder.

This is what the "lib" folder on your Pico should look like:



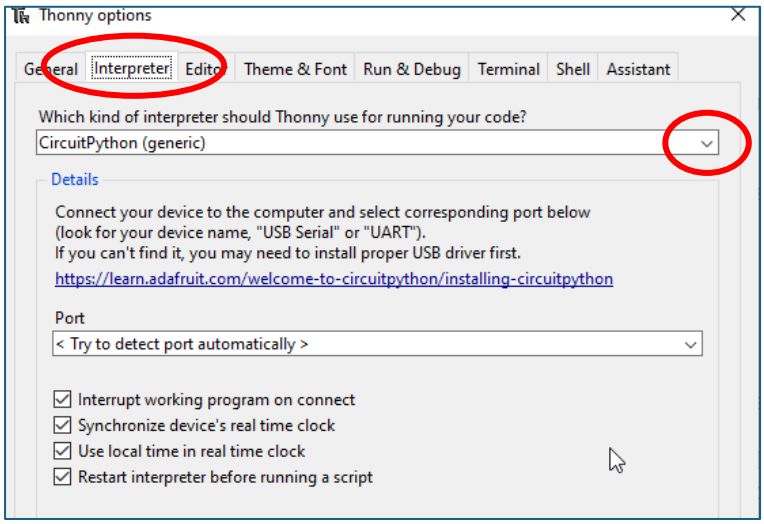
Install Libraries from the Thonny App

Open up the Thonny app.

Select Tools / Options

Select the "interpreter" tab

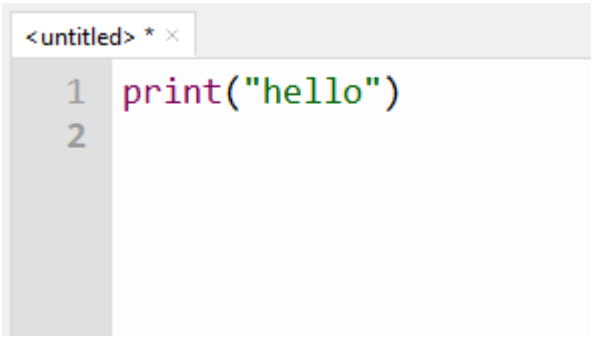
Select "circuitPython (generic)" from dropdown



Click OK.

Close any open .py pages in Thonny

In a new page write this



RUN

Output appears in shell (but not yet on display)

Go back to this Git page

Go to this Github page

https://github.com/educ8s/CircuitPython-ST7735-Examples

Select “raspberry pi pico” folder

Select “simple test” folder

Paste code into new Thonny page

```
# Raspberry Pi Pico

import board,busio
from time import sleep
from adafruit_st7735r import ST7735R
import displayio
import terminalio
from adafruit_display_text import label

mosi_pin = board.GP11
clk_pin = board.GP10
reset_pin = board.GP17
cs_pin = board.GP18
dc_pin = board.GP16
```

Run it

You may get this error (see the Thonny Shell)

```
>>> %Run -c $EDITOR_CONTENT
  Traceback (most recent call last):
    File "<stdin>", line 25, in <module>
  AttributeError: .show(x) removed. Use .root_group = x

>>>
```

The problem is with this line of code:

```
24 splash = displayio.Group()
25 display.show(splash)
```

The command **.show** is no longer used. It has been depreciated.

Instead of .show, we now use the command **root_group.**

Edit line 25 to look as follows:

```
25 display.root_group = splash
```

Source of information:

https://github.com/adafruit/circuitpython/issues/8499

If you are having a problem downloading or editing the above code, use the test code below. This is the same as above, with the changes made for you.

Test Code (for simple test):

```
# Raspberry Pi Pico

import board,busio
from time import sleep
from adafruit_st7735r import ST7735R
import displayio
import terminalio
from adafruit_display_text import label

mosi_pin = board.GP11
clk_pin = board.GP10
reset_pin = board.GP17
cs_pin = board.GP18
dc_pin = board.GP16

displayio.release_displays()

spi = busio.SPI(clock=clk_pin, MOSI=mosi_pin)

display_bus = displayio.FourWire(spi, command=dc_pin, chip_select=cs_pin, reset=reset_pin)

display = ST7735R(display_bus, width=128, height=160, bgr = True)

splash = displayio.Group()
display.root_group = splash

color_bitmap = displayio.Bitmap(128, 160, 1)

color_palette = displayio.Palette(1)
color_palette[0] = 0x00FF00  # Bright Green

bg_sprite = displayio.TileGrid(color_bitmap, pixel_shader=color_palette, x=0, y=0)
splash.append(bg_sprite)

# Draw a smaller inner rectangle
inner_bitmap = displayio.Bitmap(118, 150, 1)
inner_palette = displayio.Palette(1)
inner_palette[0] = 0x000000  # Black
inner_sprite = displayio.TileGrid(inner_bitmap, pixel_shader=inner_palette, x=5, y=5)
splash.append(inner_sprite)

# Draw a label
text_group = displayio.Group(scale=1, x=11, y=24)
text = "Hello World!\n\nThis is a sample\ntext!\n\nEverything is \nworking fine."
text_area = label.Label(terminalio.FONT, text=text, color=0xFFFFFF)
text_group.append(text_area)  # Subgroup for text scaling
splash.append(text_group)

while True:
    pass
```

Output a different message.

```
43 text_group = displayio.Group(scale=1, x=11, y=24)
44 text = "Hello World!\n\nThis is a sample\ntext!\n\nEverything is \nworking fine."
45 text_area = label.Label(terminalio.FONT, text=text, color=0xFFFFFF)
```

Changed to...

```
43 text_group = displayio.Group(scale=1, x=11, y=24)
44 text = "Love this!"
45 text_area = label.Label(terminalio.FONT, text=text, color=0xFFFFFF)
```

Trying Gauge code again

CircuitPython-ST7735-Examples / raspberry pi pico /

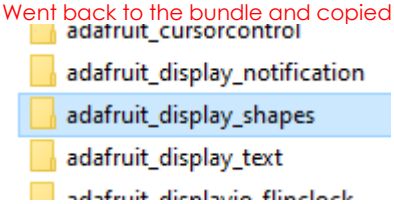
- gauge
- simple_test
- slideshow

Getting this error

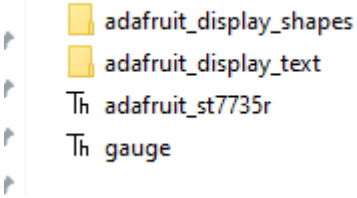
Shell

```
>>> %Run -c $EDITOR_CONTENT

  Traceback (most recent call last):
    File "<stdin>", line 6, in <module>
    File "/lib/gauge.py", line 1, in <module>
  ImportError: no module named 'adafruit_display_shapes'

>>>
```

Went back to the bundle and copied this



Paste into the Pico "lib" folder



Now getting different error

```
Traceback (most recent call last):
  File "<stdin>", line 35, in <module>
AttributeError: .show(x) removed. Use .root_group = x

>>>
```

This is the problem code

```
35 display.show(group)
36 display.auto_refresh = True
37
```

Changed to this

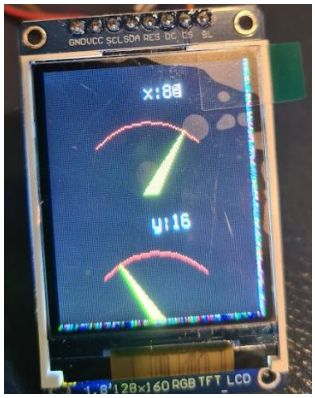
```
35 display.root_group = group
```

Now working!
Working test code for Gauge.py

```
#Raspberry Pi Pico Version

import board, busio
from adafruit_st7735r import ST7735R
import displayio
from gauge import Gauge #get the library here: https://github.com/benevpi/Circuit-Python-Gauge

mosi_pin = board.GP11
clk_pin = board.GP10
reset_pin = board.GP17
cs_pin = board.GP18
dc_pin = board.GP16

displayio.release_displays()

spi = busio.SPI(clock=clk_pin, MOSI=mosi_pin)

display_bus = displayio.FourWire(spi, command=dc_pin, chip_select=cs_pin, reset=reset_pin)

display = ST7735R(display_bus, width=128, height=160, bgr = 1 ) #bgr = 1 is needed for RGB color codes

gauge = Gauge(0,100, 64, 80, value_label="x:", arc_colour=0xFF0000, colour=0xFFFF00, outline_colour=0xFFFF00)
gauge.x = 32
gauge.y = 0

gauge2 = Gauge(0,100, 64, 80, value_label="y:", arc_colour=0xFF0000, colour=0xFFFF00, outline_colour=0xFFFF00)
gauge2.x = 32
gauge2.y = 81

group = displayio.Group(scale=1)

group.append(gauge)
group.append(gauge2)

display.root_group = group
display.auto_refresh = True

x = 0
y = 100

while True:
    while x < 100:
        x += 2
        y -= 2
        gauge.update(x)
        gauge2.update(y)

    while x > 0:
        x -= 2
        y += 2
        gauge.update(x)
        gauge2.update(y)

    while x < 100:
        x += 5
        y -= 5
        gauge.update(x)
        gauge2.update(y)

    while x > 0:
        x -= 5
        y += 5
        gauge.update(x)
```

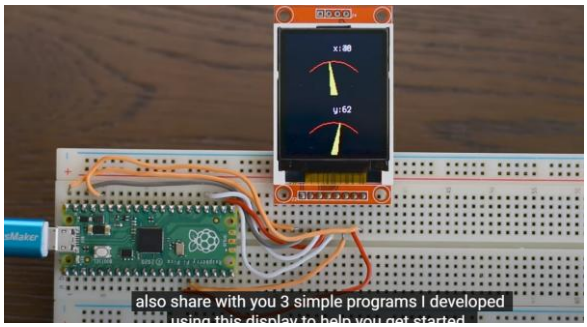
```
gauge2.update(y)
```

Could use this for water level sensor!



Sources and Further Reading

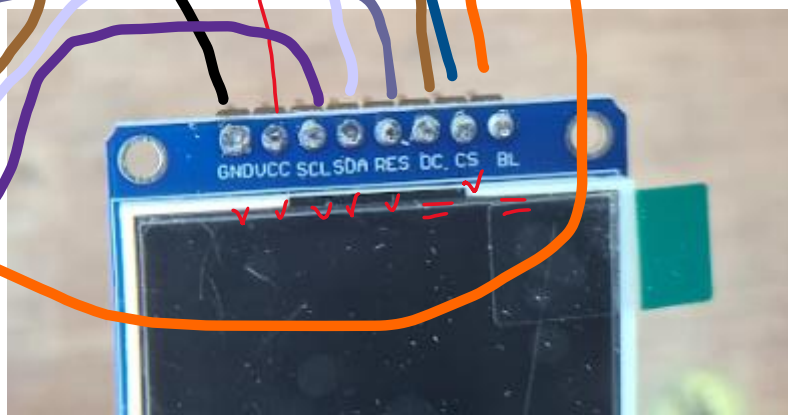
https://learn.adafruit.com/getting-started-with-raspberry-pi-pico-circuitpython/blinky-and-a-button



https://www.youtube.com/watch?v=KaGHxvVnKQ4

Display Pins	Board Pins
VCC	5V
GND	GND
CS	GP18 (CS)
RESET	GP17 (RESET)
A0	GP16 (DC)
SDA	GP11 (MOSI)
SCK	GP10 (SCLK)
LED	3.3V

Sth in bet. re.



The pin marked **LED** serves the backlight of the display. Connected to the 3.3V pin of the ESP32 one gets full backlight. If connected to pin 32 of the ESP32, the backlight can be switched on or off, while the intensity can be regulated via pulse width modulation (PWM).

SCL (clock) and **SDA** (data, on other displays named **MOSI**) connect to pins D18 and D23 of the ESP32, respectively.

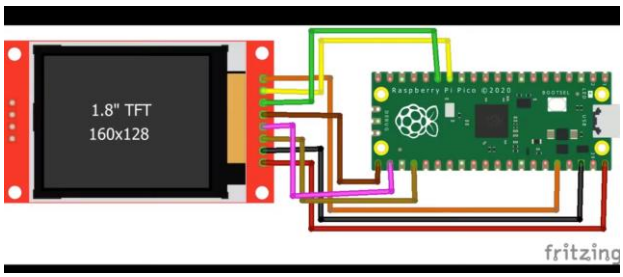
A0 is connected to pin D2,

RST (reset) to pin D4 and

CS (chip select) to pin D15 of the ESP32.

https://www.zonnepanelen.wouterlood.com/31-1-8-inch-128160-pixel-spi-tft-wiring-to-an-esp32-microcontroller/

https://www.amazon.co.uk/dp/B07FKX67WN?psc=1&ref=ppx_yo2ov_dt_b_product_details



https://www.youtube.com/watch?v=9rDXPcwuXLA

Raspberry Pi Pico	ST7735
VBUS (PIN No – 40)	VCC
GND (PIN No – 38)	GND
GP18 (PIN No – 24)	CS
GP17 (PIN No – 22)	RESET
GP16 (PIN No – 21)	A0
GP11 (PIN No – 15)	SDA
GP10 (PIN No – 14)	SCK
3.3V (PIN No – 36)	LED

The above video is to display pictures.

https://www.youtube.com/watch?v=Xw0mGUgh-k4