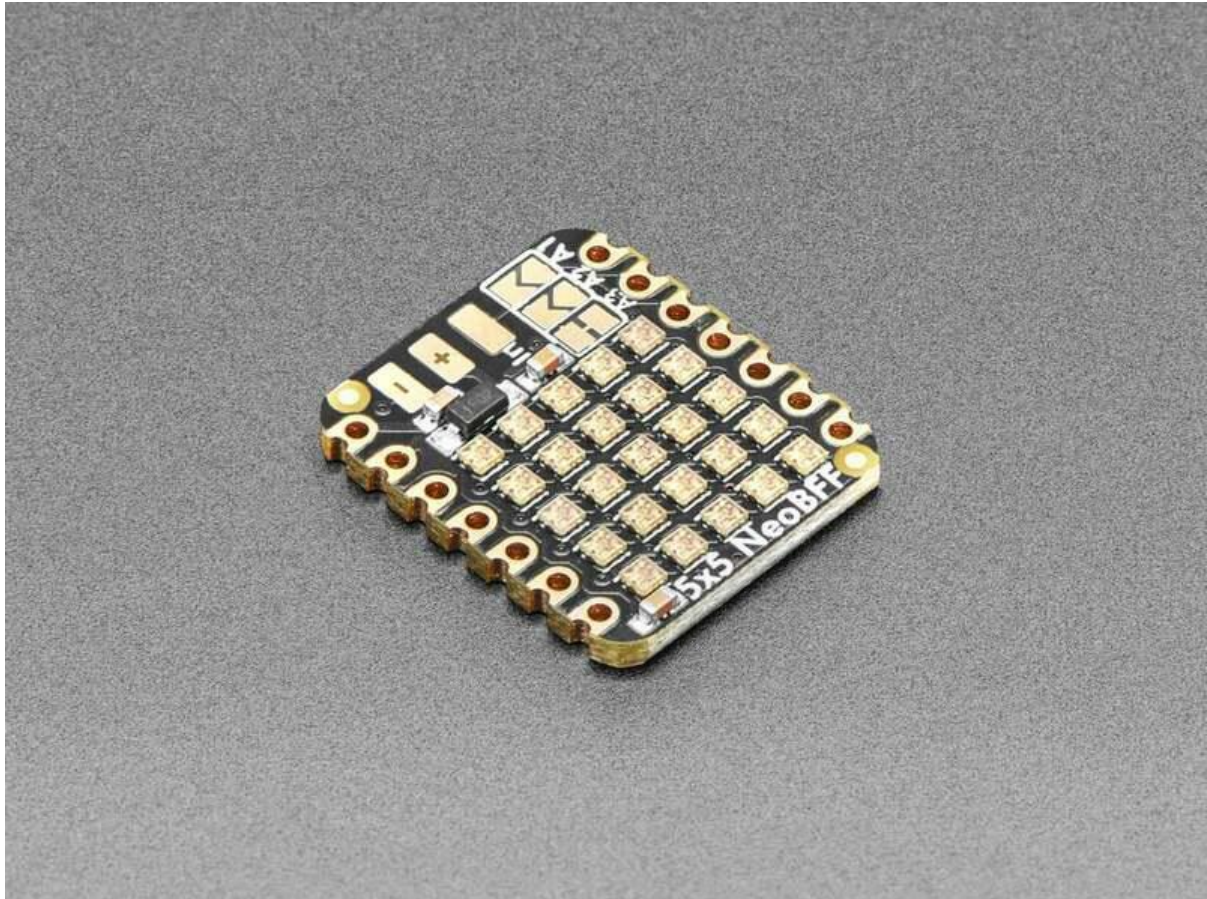




Adafruit 5x5 NeoPixel Grid BFF

Created by Liz Clark



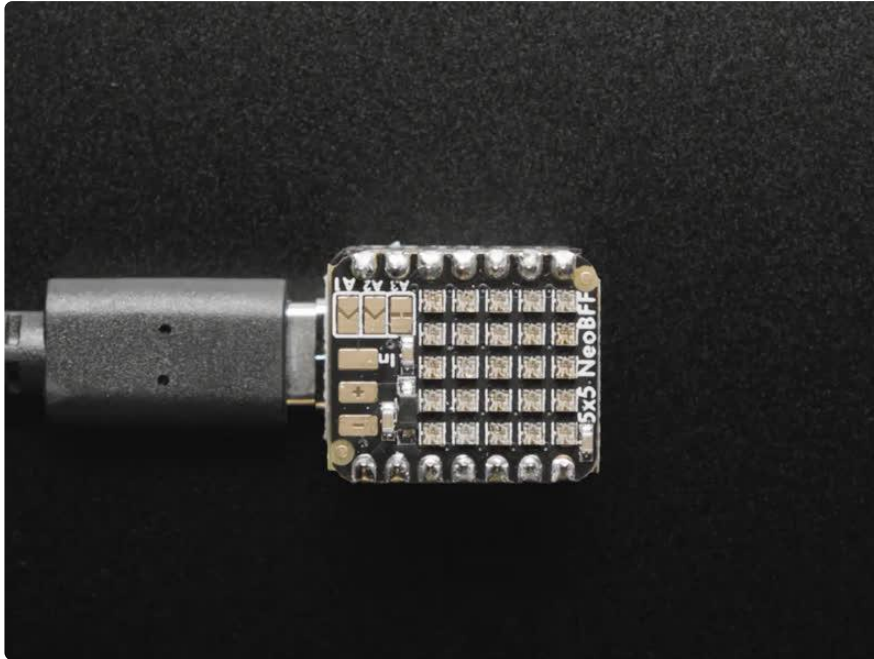
<https://learn.adafruit.com/adafruit-5x5-neopixel-grid-bff>

Last updated on 2022-12-01 04:14:57 PM EST

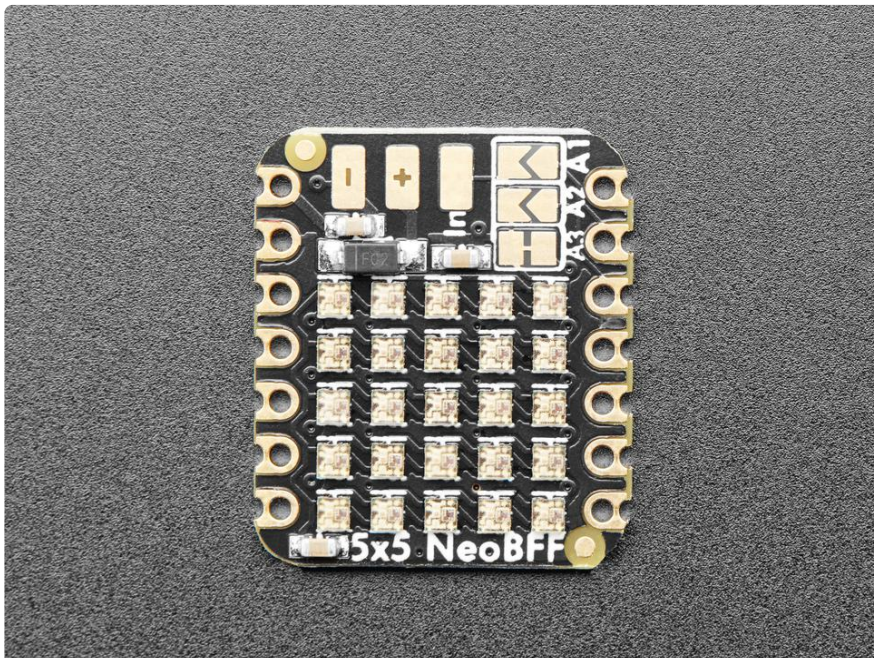
Table of Contents

Overview	3
Pinouts	5
<ul style="list-style-type: none">• External Input Pads• A3 Jumper• Pin Select Solder-jumpers	
CircuitPython	6
<ul style="list-style-type: none">• CircuitPython Microcontroller Wiring• CircuitPython Usage• Basic Scrolling Text Example• Fancy Scrolling Text Example	
Python Docs	10
Arduino	10
<ul style="list-style-type: none">• Wiring• Library Installation• Example Code	
Adafruit NeoPixel Library Docs	15
Adafruit NeoMatrix Docs	15
Adafruit GFX Library Docs	15
Downloads	15
<ul style="list-style-type: none">• Files• Schematic and Fab Print	

Overview



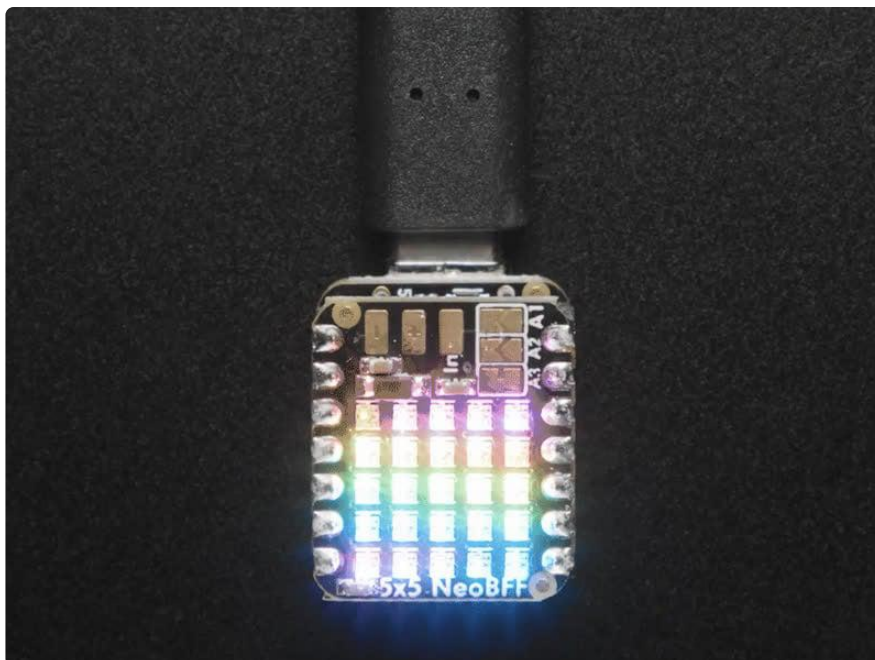
Our QT Py boards are a great way to make very small microcontroller projects that pack a ton of power - and now we have a way for you to quickly add a glittering grid of 25 RGB addressable micro NeoPixels. It's an excellent way to make tiny wearable, cosplay, or IoT projects with dazzling LEDs.



We call this the Adafruit 5x5 NeoPixel Grid BFF - a "Best Friend Forever". When you were a kid you may have learned about the "buddy" system, well this product is kinda like that! A board that will watch your QT Py's back and act like a technicolor raincoat.



This PCB is designed to fit onto the back of any QT Py or Xiao board, it can be soldered into place or use pin and socket headers to make it removable. Onboard is a grid of 25 1.5x1.5mm NeoPixel LEDs (SK6805 to be exact), connected to pin A3 by default. You can cut/solder some jumpers to select A1, or A2 instead if you like. There's also solder pads for 5V, GND and IN data, in case you want to power externally. Advanced solderers can use a thin wire to connect from the IN pad to other GPIO pins if desired.

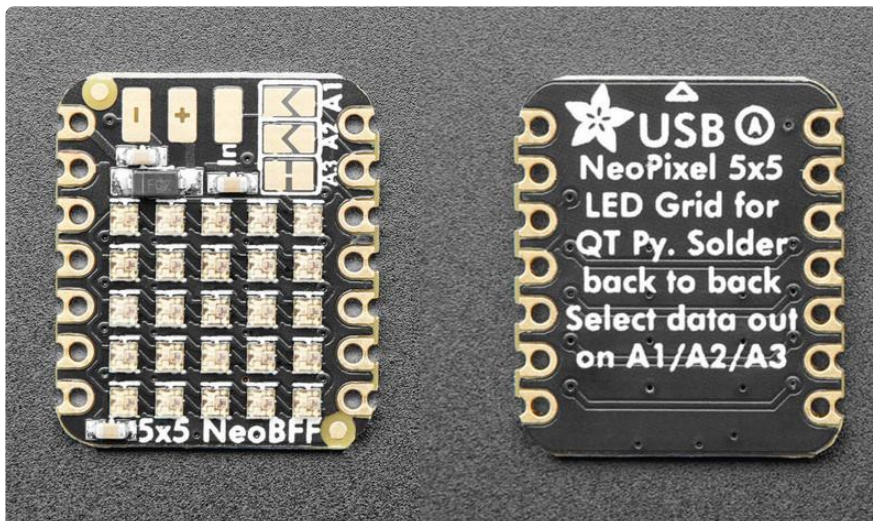


We include some header that you can solder to your QT Py. [You can also pick up an Itsy Bitsy short female header kit to make it removable but compact \(\)](#), you'll just need to trim down the headers to 7 pins long.

- Comes as an assembled and tested PCB
- For any QT Py or Xiao boards
- Uses the 5V input via USB Type-C connector on QT Py to power 25 NeoPixels
- Default connected to pin A3 for NeoPixel data line. Not level shifted but the SK6805's are happy to run at 5V power with 3V input logic level.

QT Py is not included.

Pinouts



The default input pin is A3.

External Input Pads

These pads are for powering and/or controlling the board externally. They are located at the top of the board going horizontally across.

- - - common ground for power and logic.
- + - 5V input power pad.
- In - Input data pin. Connect to a data output pin if controlling externally.

A3 Jumper

- A3 jumper - This jumper is located on the front of the board, towards the right and is labeled A3. If cut, the In pin is no longer connected to A3 and you can solder one of the solder-jumpers closed to change the output pin.

Pin Select Solder-jumpers

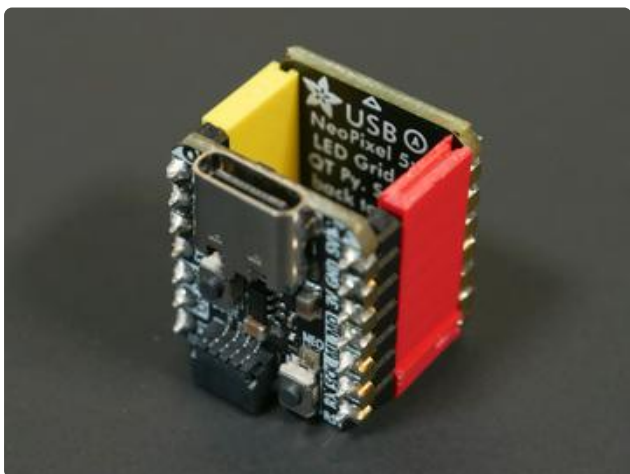
- Two solder-jumpers are available to change the input pin for the NeoPixel BFF. They are placed in the top right-hand corner on the front of the board and are located next to their corresponding pin names. You can choose to solder one of the solder-jumpers for one of the following pins:
 - A1
 - A2

CircuitPython

It's easy to use the 5x5 NeoPixel Grid BFF with CircuitPython and the [Adafruit_CircuitPython_NeoPixel \(\)](#) module. This module allows you to easily write Python code that lets you control NeoPixels.

CircuitPython Microcontroller Wiring

Plug a 5x5 NeoPixel Grid BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py RP2040 to the BFF.



Connect the QT Py RP2040 with plug headers into the 5x5 NeoPixel Grid BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

For more information on soldering socket headers, [check out this Learn Guide \(\)](#).

[How to Solder Headers Learn Guide](#)

CircuitPython Usage

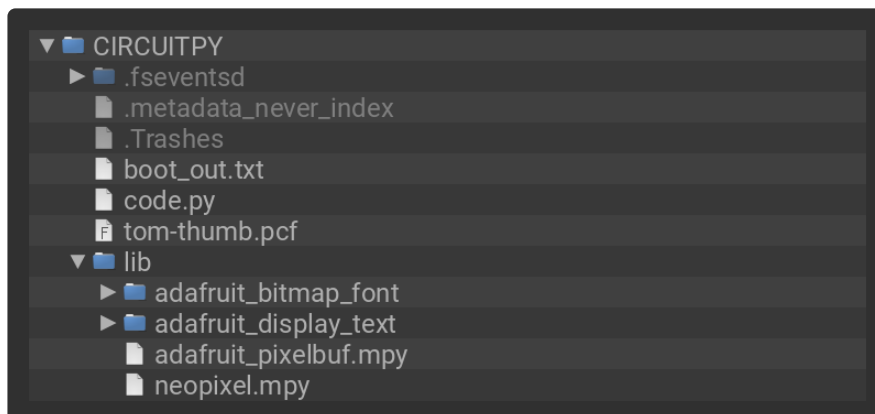
To use with CircuitPython, you need to first install the NeoPixel library, and its dependencies, into the lib folder on your CIRCUITPY drive. Then you need to update code.py with the example script.

Thankfully, we can do this in one go. In the example below, click the Download Project Bundle button below to download the necessary libraries and the code.py file in a zip file. Extract the contents of the zip file, and copy the entire lib folder, the tom-thumb.pcf font file, and the code.py file to your CIRCUITPY drive.

Your CIRCUITPY/lib folder should contain the following folders and files:

- /adafruit_bitmap_font
- /adafruit_display_text
- adafruit_pixelbuf.mpy
- neopixel.mpy

Additionally, a font file, tom-thumb.pcf, is included. This font will be utilized in both examples.



Basic Scrolling Text Example

```
# SPDX-FileCopyrightText: Copyright (c) 2022 Jeff Epler for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
import time
import board
import neopixel
from adafruit_display_text.bitmap_label import Label
from adafruit_bitmap_font import bitmap_font
from displayio import Bitmap
from rainbowio import colorwheel
```

```

font = bitmap_font.load_font("tom-thumb.pcf", Bitmap)
label = Label(text="Hello World!! Adafruit QT Py RP2040 + NeoPixel BFF ",
font=font)
bitmap = label.bitmap

pixels = neopixel.NeoPixel(board.A3, 5*5, brightness=.07, auto_write=False)
pixels.fill(0)
pixels.show()
colors = [0, 0]
hue = 0
while True:
    for i in range(bitmap.width):
        # Use a rainbow of colors, shifting each column of pixels
        hue = hue + 7
        if hue >= 256:
            hue = hue - 256

        colors[1] = colorwheel(hue)
        # Scoot the old text left by 1 pixel
        pixels[0:20] = pixels[5:25]

        # Draw in the next line of text
        for y in range(5):
            # Select black or color depending on the bitmap pixel
            pixels[20+y] = colors[bitmap[i,y]]
        pixels.show()
        time.sleep(.1)

```

Once everything is saved to the CIRCUITPY drive, you'll see your 5x5 NeoPixel grid scroll the text "Hello World!! Adafruit QT Py RP2040 + NeoPixel BFF" in rainbow colors.

You can edit the scrolling text by updating the `text` property of `label`.

```

label = Label(text="Hello World!! Adafruit QT Py RP2040 + NeoPixel BFF ",
font=font)

```

The code begins by instantiating the `label` text object and the `pixels` NeoPixel object. In the loop, the text scrolls continuously across the grid. The text is scrolled by moving the `y` coordinate of the text by `1`.

Fancy Scrolling Text Example

```

# SPDX-FileCopyrightText: Copyright (c) 2022 Jeff Epler for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
import time
import board
import neopixel
import fontio
from adafruit_display_text.bitmap_label import Label
from adafruit_bitmap_font import bitmap_font
from displayio import Bitmap
from rainbowio import colorwheel

tom_thumb = bitmap_font.load_font("tom-thumb.pcf", Bitmap)

```



```

_ glyph_keys = ['bitmap', 'tile_index', 'width', 'height', 'dx', 'dy', 'shift_x',
'shift_y']
def patch_glyph(base, **kw):
    d = {}
    for k in _glyph_keys:
        d[k] = kw.get(k, getattr(base, k))
    return fontio.Glyph(**d)

class PatchedFont:
    def __init__(self, base_font, patches):
        self.base_font = base_font
        self.patches = patches

    def get_glyph(self, glyph):
        g = self.base_font.get_glyph(glyph)
        patch = self.patches.get(glyph)
        if patch is not None:
            print("patching", repr(chr(glyph)), g)
            g = patch_glyph(g, **patch)
            print("patched", g)
        return g

    def get_bounding_box(self):
        return self.base_font.get_bounding_box()

font = PatchedFont(tom_thumb,
{
    32: {'shift_x': 1, 'dx': 0},
    105: {'dx': 0, 'shift_x': 2},
    33: {'dx': 0, 'shift_x': 2},
})

label = Label(text=" adafruit! ", font=font)
bitmap = label.bitmap

heart_bitmap = [
    0,1,1,0,0,
    1,1,1,1,0,
    0,1,1,1,1,
    1,1,1,1,0,
    0,1,1,0,0,
]

pixels = neopixel.NeoPixel(board.A3, 5*5, brightness=.06, auto_write=False)

while True:
    for hue in range(0, 255, 3):
        color = colorwheel(hue)
        pixels[:] = [pixel * color for pixel in heart_bitmap]
        pixels.show()
        time.sleep(.01)

    hue = 0
    for i in range(bitmap.width):
        # Use a rainbow of colors, shifting each column of pixels
        hue = hue + 7
        if hue >= 256:
            hue = hue - 256

        color = colorwheel(hue)

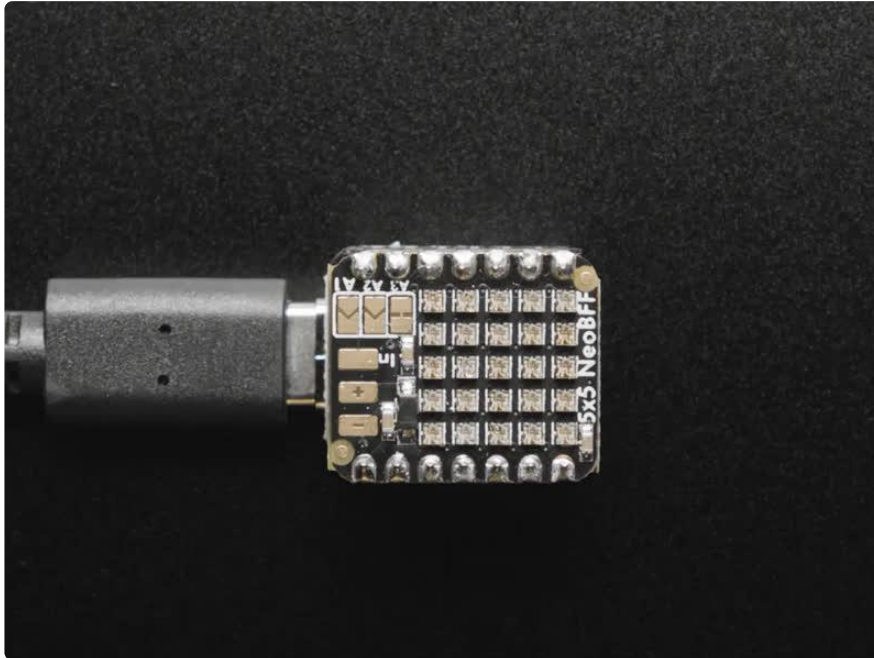
        # Scoot the old text left by 1 pixel
        pixels[:20] = pixels[5:]

        # Draw in the next line of text
        for y in range(5):
            # Select black or color depending on the bitmap pixel
            pixels[20+y] = color * bitmap[i,y]

```

```
pixels.show()  
time.sleep(.1)
```

Once everything is saved to the CIRCUITPY drive, you'll see your 5x5 NeoPixel grid scroll a heart shape followed by the text "adafruit!".



The code begins by adjusting the `tom-thumb` font file with `PatchedFont`. The "i" and "!" characters and space are made narrower so it looks a little better.

Then, the text and heart sprite shape are instantiated, followed by the `pixels` NeoPixel object.

In the loop, the heart sprite cycles through a `colorwheel` rainbow color cycle and then the text scrolls on the grid. The text is scrolled by moving the `y` coordinate of the text by `1`.

Python Docs

[Python Docs \(\)](#)

Arduino

Using the 5x5 NeoPixel Grid BFF with Arduino involves plugging the breakout into your Arduino-compatible QT Py or Xiao form factor board, installing the [Adafruit-GFX-Library \(\)](#), [Adafruit_NeoPixel \(\)](#) and [Adafruit_NeoMatrix \(\)](#) libraries, and running the provided example code.

Wiring

Plug a 5x5 NeoPixel Grid BFF into your QT Py or Xiao form factor board exactly as shown below. Here's an example of connecting a QT Py RP2040 to the BFF.



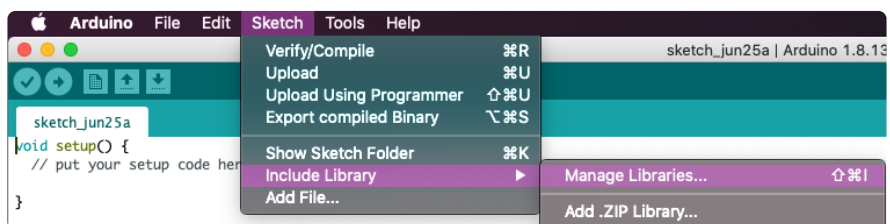
Connect the QT Py RP2040 with plug headers into the 5x5 NeoPixel Grid BFF with socket headers. They should be plugged in with the backs of the boards facing each other.

For more information on soldering socket headers, [check out this Learn Guide \(\)](#).

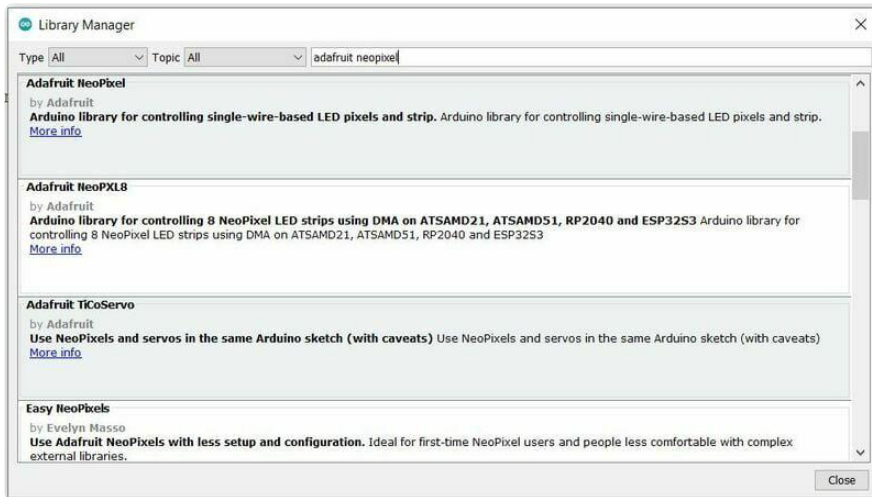
[How to Solder Headers Learn Guide](#)

Library Installation

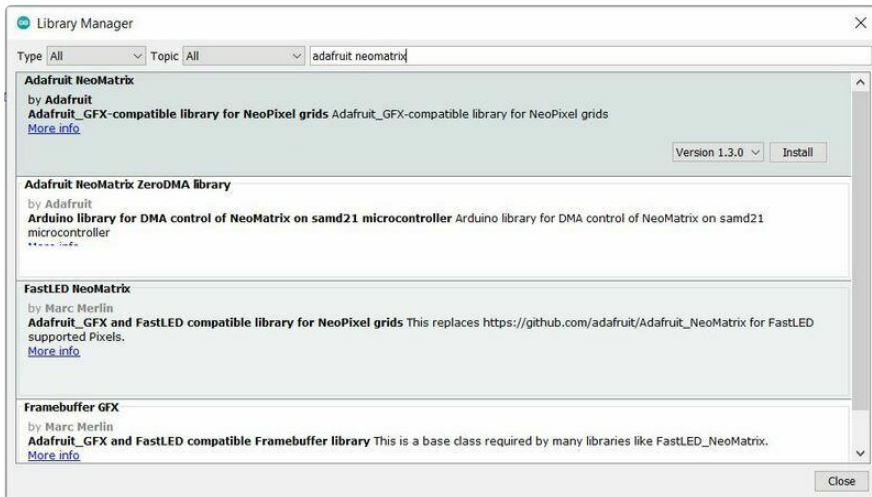
You can install the Adafruit NeoPixel, Adafruit NeoMatrix and Adafruit GFX libraries for Arduino using the Library Manager in the Arduino IDE.



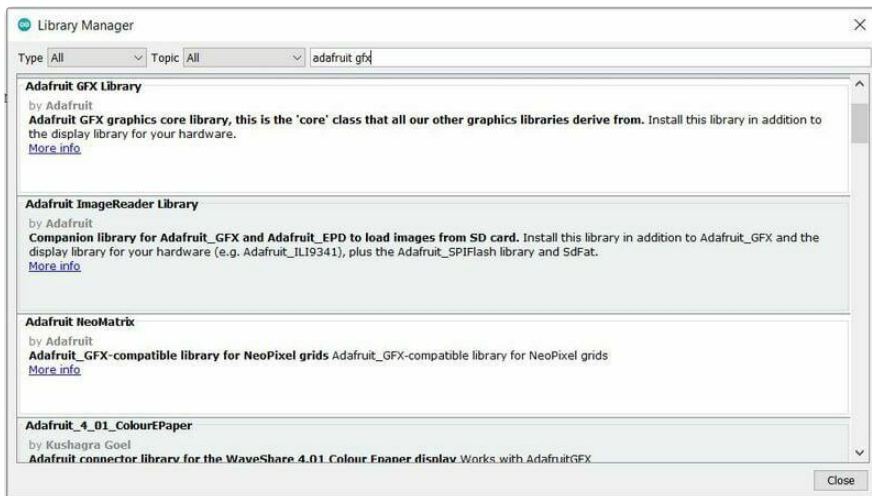
Click the Manage Libraries ... menu item, search for Adafruit NeoPixel and select the Adafruit NeoPixel library:



Then, search for Adafruit NeoMatrix and select the Adafruit NeoMatrix library:



Finally, search for Adafruit GFX, and select the Adafruit GFX library:



If asked about dependencies for any of the libraries, click "Install all".



If the "Dependencies" window does not come up, then you already have the dependencies installed.

If the dependencies are already installed, you must make sure you update them through the Arduino Library Manager before loading the example!

Example Code

```
// SPDX-FileCopyrightText: 2022 Phil B. for Adafruit Industries
//
// SPDX-License-Identifier: MIT

// Example for 5x5 NeoBFF - scrolls a message across the LED matrix.
// Requires Adafruit_GFX, Adafruit_NeoPixel and Adafruit_NeoMatrix libraries.

#include <Adafruit_GFX.h>           // Graphics library
#include <Adafruit_NeoPixel.h>     // NeoPixel library
#include <Adafruit_NeoMatrix.h>    // Bridges GFX and NeoPixel
#include <Fonts/TomThumb.h>       // A tiny 3x5 font incl. w/GFX

#define PIN A3

// NeoMatrix declaration for BFF with the power and
// Neo pins at the top (same edge as QT Py USB port):
Adafruit_NeoMatrix matrix(5, 5, PIN,
  NEO_MATRIX_TOP + NEO_MATRIX_RIGHT +
  NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE,
  NEO_GRB + NEO_KHZ800);

// Message to display, and a set of colors to cycle through. Because
// the matrix is only 5 pixels tall, characters with descenders (e.g.
// lowercase p or y) are best avoided. There are even smaller fonts
// but these get progressively less legible. ALL CAPS helps!
const char message[] = "HELLO BFF";
const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };
uint16_t message_width; // Computed in setup() below

void setup() {
  matrix.begin();
  matrix.setBrightness(40);           // Turn down brightness to about 15%
  matrix.setFont(&TomThumb);         // Use custom font
  matrix.setTextWrap(false);          // Allows text to scroll off edges
  matrix.setTextColor(colors[0]);    // Start with first color in list
  // To determine when the message has fully scrolled off the left side,
  // get the bounding rectangle of the text. As we only need the width
  // value, a couple of throwaway variables are passed to the bounds
  // function for the other values:
```

```

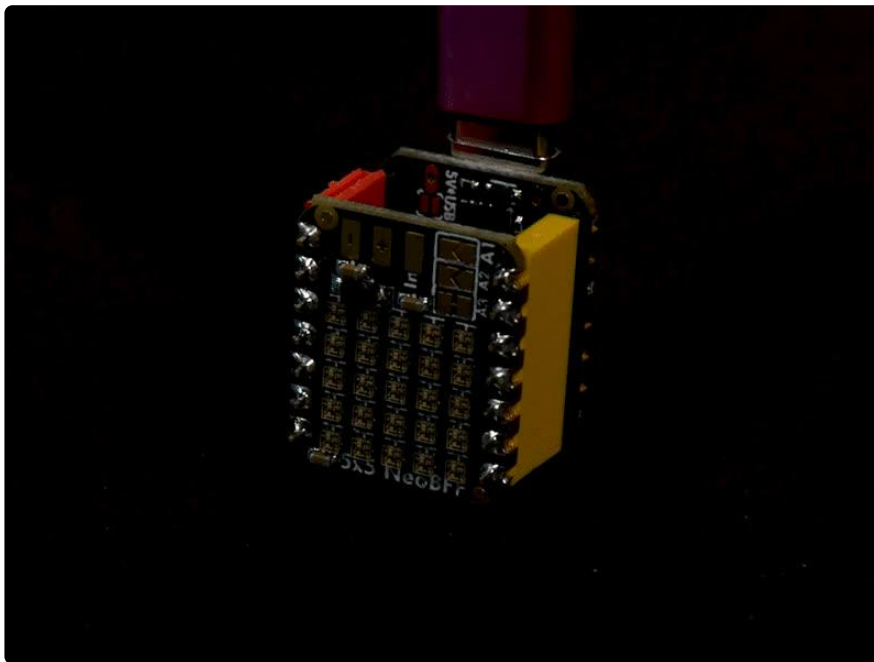
int16_t d1;
uint16_t d2;
matrix.getTextBounds(message, 0, 0, &d1, &d1, &message_width, &d2);
}

int x = matrix.width(); // Start with message off right edge
int y = matrix.height(); // With custom fonts, y is the baseline, not top
int pass = 0; // Counts through the colors[] array

void loop() {
  matrix.fillScreen(0); // Erase message in old position.
  matrix.setCursor(x, y); // Set new cursor position,
  matrix.print(message); // draw the message
  matrix.show(); // and update the matrix.
  if(--x < -message_width) { // Move 1 pixel left. Then, if scrolled off left...
    x = matrix.width(); // reset position off right edge and
    if(++pass >= 3) pass = 0; // increment color in list, rolling over if needed.
    matrix.setTextColor(colors[pass]);
  }
  delay(100); // 1/10 sec pause
}

```

Upload the sketch to your board. You'll see the 5x5 NeoPixel grid scroll the text "HELLO BFF". The text will loop through being red, green and blue.



If you'd like to change the text being scrolled, you can edit the `message[]` constant at the top of the code.

```

// Message to display, and a set of colors to cycle through. Because
// the matrix is only 5 pixels tall, characters with descenders (e.g.
// lowercase p or y) are best avoided. There are even smaller fonts
// but these get progressively less legible. ALL CAPS helps!
const char message[] = "HELLO BFF";

```

Adafruit NeoPixel Library Docs

[Adafruit NeoPixel Library Docs \(\)](#)

Adafruit NeoMatrix Docs

[Adafruit NeoMatrix Docs \(\)](#)

Adafruit GFX Library Docs

[Adafruit GFX Library Docs \(\)](#)

Downloads

Files

- [SK6805 NeoPixel datasheet \(\)](#)
- [EagleCAD PCB files on GitHub \(\)](#)
- [3D models on GitHub \(\)](#)
- [Fritzing object in the Adafruit Fritzing Library \(\)](#)

Schematic and Fab Print

