**Table of Contents**

**Overview**
- Adafruit Parts
- Decorate & Diffuse

**Wiring Diagram**

**CircuitPython**
- CircuitPython Quickstart
- Safe Mode
- Flash Resetting UF2

**Software**
- Upload the Code and Libraries to the QT Py RP2040
- Customizing the Code
- Troubleshooting

**Electronics Assembly**
- Troubleshooting

**Bezel Assembly**
Overview

Complete your look this Valentine's day with NeoPixel novelty cufflinks. Add your own scrolling text messages to send a surprise smile to that special someone's face. Whether it's a romantic declaration of love, a sweet message of affection, or simply a playful joke, the customizable scrolling text will get your point across in a fun and silly way.

This is a fairly easy project. There's a little tight soldering work, but there are only 3 pieces to assemble so you can do this project in an afternoon, and finish it in time to impress your sweetie with your mad maker skills.

This project would also make a wonderful hat pin, brooch or necklace with just a little modification. It packs a punch with 25 pixels in a teeny tiny package.

Adafruit Parts

Adafruit 5x5 NeoPixel Grid BFF Add-On for QT Py and Xiao
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...
https://www.adafruit.com/product/5646
Adafruit QT Py RP2040
What a cutie pie! Or is it... a QT Py? This diminutive dev board comes with one of our new favorite chip, the RP2040. It's been made famous in the new https://www.adafruit.com/product/4900

Switched JST-PH 2-Pin SMT Right Angle Breakout Board
This switched JST connector is the best way to quickly prototype with our LiPoly batteries. We paired a genuine JST connector with a slide switch that can do up to 600mA. Both are... https://www.adafruit.com/product/1863

1 x Battery
Lithium Ion Polymer Battery - 3.7v 100mAh https://www.adafruit.com/product/1570

1 x Battery Charger
Adafruit Micro Lipo - USB LiIon/LiPoly charger - v2 https://www.adafruit.com/product/1304

1 x Solid Core Hook-up Wire
Hook-up Wire Spool Set - 22AWG Solid Core - 6 x 25 ft https://www.adafruit.com/product/1311

1 x Solderless Breadboard
Half-Size Breadboard with Mounting Holes https://www.adafruit.com/product/4539

Decorate & Diffuse

- Decorative bezel with an open back with a 3/4" opening and a 1" diameter
- UV Artist Resin ()
- Mica Powder or Alcohol ink (optional)
We'll solder the QT Py and the NeoPixel BFF add-on together using headers, back-to-back. We'll leave the center pins (MO and A3) open / unused on both boards so we have room for our wire to pass through.

The on/off switch will be connected to the + and - pads on the BFF using solid-core wire.

**CircuitPython**

CircuitPython is a derivative of MicroPython designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.

**CircuitPython Quickstart**

Follow this step-by-step to quickly get CircuitPython running on your board.

Download the latest version of CircuitPython for this board via circuitpython.org
Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.
You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :)

Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.
Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

**Entering Safe Mode**

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

**In Safe Mode**

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.
CircuitPython is in safe mode because you pressed the reset button during boot.
Press again to exit safe mode.
Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

**Flash Resetting UF2**

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep
clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

**Software**

Once you've finished setting up your QT Py RP2040 with CircuitPython, you can access the code, audio files and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download to your computer as a zipped folder.

```python
# SPDX-FileCopyrightText: Copyright (c) 2023 Erin St. Blaine for Adafruit Industries
# SPDX-License-Identifier: MIT
import time
from random import randint
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 getGlyph(self, glyph):
        g = self.base_font.getGlyph(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 getBoundingBox(self):
        return self.base_font.getBoundingBox()

font = PatchedFont(tom_thumb,
                   {32: {'shift_x': 1, 'dx': 0},
                    105: {'dx': 0, 'shift_x': 2}},
```
Upload the Code and Libraries to the QT Py RP2040

After downloading the Project Bundle, plug your QT Py RP2040 into the computer USB port. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the QT Py RP2040's CIRCUITPY drive.

- lib folder
- code.py
- tom-thumb.pcf

Your QT Py RP2040 CIRCUITPY drive should look like this after copying the lib folder and code.py file.
Customizing the Code

This code will show a bitmap of a heart, and then scroll some text chosen randomly from a list of text strings. The messages I've chosen were written by an early AI when it was asked to create sayings for candy valentine hearts. They're pretty wonderful.

To add your own text strings, look for this section of code:

```python
# List of text strings
text_strings = [
    "  love bun   ", "  dear me   ", "  my bear   ",
    "  my my   ", "  you are babe   "]
```

Change them to whatever you'd like, or add more. You can adjust the spaces to change the amount of time between messages.

The heart bitmap is cleverly drawn with 1s and 0s. If you want to tweak it, you can make your own bitmap in ASCII style by turning pixels "on" or "off" in the 5x5 grid. A 1 means the pixel is on, and a 0 means it's off.

```python
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
]
```

Troubleshooting

1. The drive doesn't appear when you plug your QT Py RP2040 into your computer.

This is normal but a little weird because it works differently from the other QT Py boards. You have to hold down theBOOT button while plugging it in, with this board. Start with the board un-powered (i.e. no battery plugged in) and hold down BOOT while you attach the cable. Keep holding it until the drive appears.
2. The code appears to be working but no images or text are appearing

This code assumes we're using pin A1. The board defaults to pin A3 -- we changed this using a solder bridge during assembly. If your assembly method is different, and you soldered headers to the whole board, try changing the code to pin A3 to see if that fixes the problem.

3. Be sure the code you downloaded matches the version of CircuitPython you're running. This is a fast-changing world and new releases are coming out all the time. If you installed CircuitPython 8, be sure you've got the correct code bundle for version 8.

---

Electronics Assembly

Cut 4 pieces of header with 3 pins each. Slot them into a solderless breadboard so that the NeoPixel BFF fits on top, leaving the middle hole open on both sides.

Solder the pins in place with the LEDs facing upwards. If it's your first time soldering headers, check out our How to Solder Headers guide to get started.
Tin the + and - pads on the NeoPixel BFF and solder two long-ish (8” or so) solid core wires. I'm using a red wire on the + pad and a black wire on -.
Bend the wires carefully around over the top of the board and thread them through the holes we left in the headers. Flip the board over and solder the QT Py on. The boards go back-to-back with the USB port facing the top of the NeoPixel BFF (the same end the wires are soldered to).
Twist the wires together over the QT Py, being careful not to squish the Boot button. We'll build the structure of the cufflink out of these wires since they're pretty stiff and easy to work with.

Make a stem of about 1/2"-3/4" and then twist the wires around the JST connector switch to hold it in place. Watch the assembly video on the first page of this guide for more details on how I did the wire wrapping.

Trim the two wires to length and solder them into the JST connector. The red wire goes to SW and the black wire can go to either of the G pins.
The QT Py defaults to sending NeoPixel data on pin A3. However, we can't use A3 since we skipped that pin on the header to make room for our wire to go through. No problem: it's easy to change the pin on the NeoPixel BFF.

Use a sharp knife to cut through the copper trace on the A3 pin pad to disable it. Then, make a solder bridge to connect the two halves of the A1 pin. This will tell the QT Py to send data on pin A1 instead.

Remember to change this in your code! Most sample projects for the BFF will default to pin A3.

If you've already uploaded your code, it's time to test and make sure everything works. Plug a battery into your JST connector and flip the switch to ON. You should see an animated heart and scrolling text. Hooray!

Trim down any pokey header wires with a pair of flush cutter pliers to streamline your cufflink.

Troubleshooting

If you're not getting hearts and messages, here are some things to try:

1. Did you fully cut the trace for pin A3 and fully bridge the solder pads on A1?
2. Does your code call for pin A1? If you're using example code for this board from a different source you'll want to update the pin number.
3. Check to be sure your solder joints are clean and not bridging anywhere
4. Try re-uploading the code, and check the troubleshooting tips on the code page
5. Make sure you soldered to the correct pins on the on/off switch
Bezel Assembly

I'm using a 1" metal bezel I found at the craft store, in the UV Resin area of the jewelry section. Take your BFF shopping with you to find something you like.

I've also got some hardcore clear nail polish from the drug store. This stuff is for gel manicures, and it's nice and thick and chip-resistant. Paint it liberally on the back of the bezel, or else the metal will bridge the pins on the NeoPixel BFF and cause it to short out.
Place the bezel on a piece of clear packing tape. If you want, put a little mica powder or other diffusion material down on the tape.

Squeeze a little resin into the bezel. Pop any bubbles with a push pin or by hovering a flame or lighter just above the resin.

My resin came with its own UV flashlight. Shine the light on the resin for 2-3 minutes and it will cure fully. You can also just leave the bezel out in the sun for 30 minutes.

I found it worked best to do the resin in a couple layers, and kept adding it until I got a nice domed effect.
Glue the bezel to the front of the NeoPixel BFF. Do this while it's turned on, so you can center the graphic inside the bezel - it won't be perfectly centered on the BFF but this bezel will cover the edges adequately anyway.

To use the cufflinks, thread the JST connector piece through the first button hole. Then plug in the battery, which will hide inside the sleeve. Thread the JST connector through the second hole to hold the link in place.