NeoPixel Menorah
Created by Ruiz Brothers

https://learn.adafruit.com/neopixel-menorah

Last updated on 2022-03-01 02:00:35 PM EST
# Table of Contents

## Overview
- Light Up The Days 5
- 3D Printed Bulbs 5
- USB-C Powered 6
- QT Py RP2040 6
- Parts 7
- Hardware 9

## 3D Printing
- CAD Parts List 11
- Build Volume 11
- Slicing Bulbs 12
- Slicing Frames 12
- Design Source Files 12

## Circuit Diagram
- Adafruit Library for Fritzing 13
- Wired Connections 13
- Powering 14

## CircuitPython
- CircuitPython Quickstart 14
- Safe Mode 16
- Flash Resetting UF2 18

## Coding Menorah
- Upload the Code and Libraries to the QT Py RP2040 20

## Wiring Push Button
- Cables for Push Button 21
- Solder Wires to Push Button 21
- Wired Push Button 22

## Wiring QT Py
- Cables for QT Py 22
- Solder 3-pin Cable 22
- Solder 2-pin Cable 23
- Wired QT PY 23

## Wiring NeoPixels
- Cables for NeoPixels 23
- Wiring Input 24
- Wiring Output 24
- Wired NeoPixel 24
- NeoPixel with Cables 25
- 6x Wired NeoPixels 25
- First NeoPixel Wiring 25
- Last NeoPixel 26
- Center NeoPixel 26
- 9x Wired NeoPixels 26
- Test Circuit 27
<table>
<thead>
<tr>
<th>Assembly</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Top Cover and Holders</td>
<td>27</td>
</tr>
<tr>
<td>• Install Holders</td>
<td>27</td>
</tr>
<tr>
<td>• First NeoPixel Install</td>
<td>28</td>
</tr>
<tr>
<td>• Installing NeoPixel</td>
<td>28</td>
</tr>
<tr>
<td>• Installed NeoPixel</td>
<td>29</td>
</tr>
<tr>
<td>• Connect NeoPixel Cables</td>
<td>29</td>
</tr>
<tr>
<td>• Install Frame to Top Cover</td>
<td>30</td>
</tr>
<tr>
<td>• Bulb Diffusers</td>
<td>30</td>
</tr>
<tr>
<td>• Install Diffusers</td>
<td>31</td>
</tr>
<tr>
<td>• Secure Stand to Cover</td>
<td>31</td>
</tr>
<tr>
<td>• Secure Stand to Cover (continued)</td>
<td>32</td>
</tr>
<tr>
<td>• Install Push Button</td>
<td>32</td>
</tr>
<tr>
<td>• Install QT Py</td>
<td>33</td>
</tr>
<tr>
<td>• Secured QT Py</td>
<td>33</td>
</tr>
<tr>
<td>• Secure Frame to Base Cover</td>
<td>34</td>
</tr>
<tr>
<td>• Connect Button to QT Py</td>
<td>34</td>
</tr>
<tr>
<td>• Install NeoPixel Cable</td>
<td>35</td>
</tr>
<tr>
<td>• Secure Base to Stand</td>
<td>35</td>
</tr>
<tr>
<td>• Install Frame</td>
<td>36</td>
</tr>
<tr>
<td>• NeoPixel Order</td>
<td>36</td>
</tr>
<tr>
<td>• Connect QT Py to NeoPixel Strand</td>
<td>37</td>
</tr>
<tr>
<td>• Secure Top Cover to Frame</td>
<td>37</td>
</tr>
<tr>
<td>• Install Stand Covers</td>
<td>38</td>
</tr>
<tr>
<td>• Installed Stand Covers</td>
<td>38</td>
</tr>
<tr>
<td>• Final Build</td>
<td>39</td>
</tr>
</tbody>
</table>
Overview

In this guide you'll learn how to build a 3D printed Menorah that lights up with NeoPixel LEDs and CircuitPython. This minimalistic design features parts that snap fit together and print without any supports. The assembly is modular and easy to put together.

Light Up The Days
A push button allows you to light up a NeoPixel LED on each night of Hanukkah. The Shamash (center NeoPixel) stays lit each night. On the first night of Hanukkah, the NeoPixel LED on the far right is lit. Just press the button each night to light up the next one.
3D Printed Bulbs
The bulbs are printed in translucent PLA and the NeoPixels are fitted inside these little holders. The bulbs are hollow to better diffuse the LEDs so they’ll illuminate nicely.

USB-C Powered
The bottom has a cut out so you have access to the USB-C port. It can be powered by a 5V USB supply from either a computer or a wall outlet.

QT Py RP2040
This project is powered by the Adafruit QT Py RP2040 running CircuitPython. The QT Py is secured to the bottom cover and just snaps into the build-in holder.
Parts

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

NeoPixel Mini Button PCB - Pack of 50
These are the smallest NeoPixel breakouts around! Tiny, bright RGB pixels to your project. These little PCBs are only 9.1mm x 9.1mm and have two sets of three pads on the back for...
https://www.adafruit.com/product/4356
16mm Panel Mount Momentary Pushbutton - Burgundy
OK, this item is pretty simple - it's a panel mount pushbutton. It's not that exciting, no LEDs, no bells & whistles. But we really like it anyways – look at that... https://www.adafruit.com/product/1503

9x quantity of the 1.25mm Pitch 3-pin Cable Matching Pair - 40cm long - Molex PicoBlade Compatible are required

1.25mm Pitch 3-pin Cable Matching Pair - 40cm long
When 0.1" is too big, and JST PH's too chunky, these ultra-slim 1.25mm connectors are a reliable alternative. These are only 1.25mm pitch, but have a nice clicky...
https://www.adafruit.com/product/4721

1.25mm Pitch 2-pin Cable Matching Pair - 40cm long
When 0.1" is too big, and JST PH's too chunky, these ultra-slim 1.25mm connectors are a reliable alternative. These are only 1.25mm pitch, but have a nice clicky...
https://www.adafruit.com/product/4720
USB Type C Cable with Data/Charge Switch
Perhaps your smart phone, tablet, or even your Raspberry Pi 4, charges off of USB, but can you really trust that charging station at the airport? Now you can control when to allow...
https://www.adafruit.com/product/4696

Hardware

The following fasteners are required to assemble this build.

- 4x M2.5 x 10mm
- 4x M2.5 hex nuts

Black Nylon Machine Screw and Stand-off Set – M2.5 Thread
Totaling 380 pieces, this M2.5 Screw Set is a must-have for your workstation. You'll have enough screws, nuts, and hex standoffs to fuel your maker...
https://www.adafruit.com/product/3299
3D Printing

CAD Parts List
STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material. Original design source may be downloaded using the links below:

- Base Bottom Cover
- Base Frame
- Base Top Cover
- 9x Flame (Hollow or Solid)
- 1x Holder Long
- 8x Holder Short
- LED Case Bottom
- LED Case Frame
- LED Case Top
- 2x Stand Cover
- Stand Frame

Download STLs.zip
https://adafru.it/X4d

Download CAD source
https://adafru.it/X4e

Download 3MF.zip
https://adafru.it/X4f

Build Volume
The parts require a 3D printer with a minimum build volume.

- 182mm (X) x 182mm (Y) x 90mm (Z)
Slicing Bulbs
For best quality, it's recommend to print bulbs one at a time as opposed to a set of multiple. Optionally print the "solid" version for slicing with vase mode.

Slicing Frames
For best results, a brim is suggested with printing the frames parts of the design.

Design Source Files
The project assembly was designed in Fusion 360. This can be downloaded in different formats like STEP, STL and more. Electronic components like Adafruit's board, displays, connectors and more can be downloaded from the Adafruit CAD parts GitHub Repo (https://adafru.it/AW8).
Circuit Diagram

The diagram below provides a visual reference for wiring of the components. This diagram was created using the software package Fritzing (https://adafruit.it/oEP).

Adafruit Library for Fritzing

Use Adafruit’s Fritzing parts library to create circuit diagrams for your projects. Download the library or just grab individual parts. Get the library and parts from GitHub - Adafruit Fritzing Parts (https://adafruit.it/AYZ).

Wired Connections

The project uses NeoPixel PCB Buttons which are slightly different than the Circuit Diagram but they use the same pin outs.

NeoPixels to QT Py RP2040

- DIN from NeoPixel strip to Pin A0 on QT Py
- 5V from NeoPixel strip to 5V on QT Py
- GND from NeoPixel strip to GND on QT Py
NeoPixel PCB Buttons

- DOUT from NeoPixel to DIN on NeoPixel
- 5V from NeoPixel to 5V on NeoPixel
- GND from NeoPixel to GND on NeoPixel

Push Button

- Pin 1 from button to A1 on QT Py
- Pin 2 from button to GND on QT Py

Powering

The Adafruit board can be powered via USB or JST using a 3.7v lipo battery. In this project, a 5V power supply is used.

CircuitPython

CircuitPython (https://adafrui.it/tB7) is a derivative of MicroPython (https://adafrui.it/BeZ) 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

https://adafrui.it/RLD
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 in CircuitPython 6.x

To enter safe mode when using CircuitPython 6.x, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 700ms. On some boards, the onboard status LED (highlighted in green above) will turn solid yellow during this time. If you press reset during that 700ms, 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.)
Entering Safe Mode in CircuitPython 7.x

To enter safe mode when using CircuitPython 7.x, 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

Once you've entered safe mode successfully in CircuitPython 6.x, the LED will pulse yellow.

If you successfully enter safe mode on CircuitPython 7.x, 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.
Coding Menorah

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 as a zipped folder.

```python
# SPDX-FileCopyrightText: 2021 Liz Clark for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import neopixel
from digitalio import DigitalInOut, Direction, Pull
from adafruit_led_animation.animation.rainbow import Rainbow
from adafruit_led_animation.sequence import AnimationSequence
from adafruit_led_animation.color import AMBER

# button setup
button = DigitalInOut(board.D1)
button.direction = Direction.INPUT
button.pull = Pull.UP

# neopixel setup
pixel_pin = board.D0
pixel_num = 9

pixels = neopixel.NeoPixel(pixel_pin, pixel_num, brightness=0.3, auto_write=False)
```
# variable for number of pixels in the pixelmap helper
num = 1

# pixel map helper
# allows you to light each candle up one by one
# begins with one being lit (num)
candles = helper.PixelMap.horizontal_lines(
    pixels, num, 1, helper.horizontal_strip_gridmap(pixel_num, alternating=False)
)

# rainbow animation
rainbow = Rainbow(candles, speed=0.1, period=5)
animations = AnimationSequence(rainbow)

# turn on center candle
pixels[4] = AMBER
pixels.show()

while True:
    # if only one candle is lit, don't rewrite center neopixel
    if num == 1:
        pass
    # otherwise write data to center neopixel
    else:
        pixels[4] = AMBER
        pixels.show()
        # animation the rainbow animation
        animations.animate()

    # if you press the button...
    if not button.value:
        # if all of the candles are not lit up yet...
        if num < 9:
            # increase value of num by one
            num += 1
        # skip the center candle so that it stays amber
        if num == 4:
            num = 5
        # recreate the pixel helper to increase the size of the horizontal grid
        # this is how the next neopixel is lit up in the sequence
candles = helper.PixelMap.horizontal_lines(
    pixels, num, 1, helper.horizontal_strip_gridmap(pixel_num, alternating=False)
)
    rainbow = Rainbow(candles, speed=0.1, period=5)
    animations = AnimationSequence(rainbow)
    # quick delay so that everything flows well
    time.sleep(0.5)

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

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

---

**Wiring Push Button**

**Cables for Push Button**

Use a 2-pin molex cable to connect the push button to the QT Py. Measure and cut the wires down to size. Use the following lengths to trim the molex cables.

- 2-pin molex – 9cm (3.5in)

**Solder Wires to Push Button**

Solder the two wires to the two pins on the push button. Polarity does not matter.
Wired Push Button
Double check the wires have been properly soldered to the pins on the push button.

Wiring QT Py

Cables for QT Py
Use a 3-pin molex cable to connect the NeoPixels to the QT Py

A 2-pin molex cable is used to connect the push button to the QT Py.

Measure and cut the cables down to size. Use the following lengths to trim the cables.

- 3-pin molex cable - 19cm (7.5in)
- 2-pin molex cable - 9cm (3.5in)

Solder 3-pin Cable
Wire the 3-pin cable to the QT Py. Solder the black wire to ground, red to 5V and yellow to the A0 pin.
Solder 2-pin Cable
Wire the 2-pin cable to the QT Py. Solder the black wire to ground and red to the A1 pin.

Wired QT PY
Double check the solder joints are correctly soldered and have a solid connection.

Wiring NeoPixels

Cables for NeoPixels
Use a new pair of Molex cables and cut up a set for six NeoPixels. Make the cables the following length.

6cm (2.3in)
Wiring Input
Start by wiring up the molex cable plugs to the SMD pads on the back of a NeoPixel button PCB. Wire yellow to data in, black to ground and red to the voltage pad.

Wiring Output
Proceed to wire the molex socket cable to the data out, ground and voltage pads.

Wired NeoPixel
Double check the wires are soldered in the correct orientation with them facing inwards.
NeoPixel with Cables
The two molex cables should be able to straighten out for the assembly.

6x Wired NeoPixels
Measure and cut 6x pairs of molex cables for 6x NeoPixels. Solder the cables to the NeoPixels. Wire the socket cables to the NeoPixel's data out pads and the cable plugs to the data in pads.

First NeoPixel Wiring
The first NeoPixel in the strand will need to have one longer molex cable (plug) wired to the data in pin. The shorter molex cable (socket) is wired to the data out pin.
Last NeoPixel
The last NeoPixel in the strand only needs one molex cable (plug) connected to the data in pad. Use the following length to size the cable.

6cm (2.3in)

Center NeoPixel
The fifth NeoPixel in the strand features longer molex cables. Use the following length to measure and cut the cables.

12cm (4.7in)

9x Wired NeoPixels
Double check each NeoPixel is probably wired up. Line them up in order (from left to right) and reference the photo for correct placement.
Test Circuit
Plug the molex cables together with the first NeoPixel connecting to the QT Py. Use 5V power supply or computer's hub to power the circuit.

Assembly

Top Cover and Holders
Get the holders ready to install into the LED holder top cover. There should be eight short holders and one longer holder.
Install Holders
Start by fastening the longer holder to the hole in the center of the top cover.

Proceed to install all eight shorter holder into the remaining holders in the top cover.

Firmly hand tighten all of the holders to the top cover.

First NeoPixel Install
Get the first NeoPixel ready to install into the holder.
Installing NeoPixel

Carefully fit the molex cables from the first NeoPixel through the short holder that's first in line. The connectors should fit all the way through the holder.

Installed NeoPixel

Proceed to install the second NeoPixel in the strand.

Connect NeoPixel Cables

Grab the socket cable from the first NeoPixel and connect it to the second NeoPixel. The first NeoPixels data out flows into the second NeoPixels data in.
Install Frame to Top Cover
Continue to install NeoPixels into the holders. Connect the cables together after installing each NeoPixel.

Bulb Diffusers
Get the nine diffuser bulbs ready to install into the top of the holders.
Install Diffusers

The bulbs are press fitted into the top of each holder.

The tolerances should have a tight snug fit.

If the bulbs are too tight to fit, you may need to file down the edges or use a deburring tool on the holders.

Secure Stand to Cover

Attach the Stand Frame part to the bottom cover of the LED case using the following hardware.

- 2x M2.5 x 10mm
- 2x M2.5 hex nuts
Secure Stand to Cover (continued)

Attach the stand frame to the bottom base cover using the following hardware.

- 2x M2.5 x 10mm
- 2x M2.5 hex nuts

Install Push Button

The 16mm push button is panel mounted to the base top cover.

Insert the push button into the hole in the cover with the cable going in first.

Use the included hex nut to secure the button to the cover.
Install QT Py
The QT Py is secured to the built-in holder in the base bottom cover with the USB port facing the edge.

Start by fitting the QT Py PCB into the holder at an angle.

Slightly bend the cover to allow the tabs to snap onto the edges of the PCB.

Secured QT Py
Double check the QT Py is fully seated into the built-in holder. The USB-C port should be facing the correctly
Secure Frame to Base Cover
The frame from the base is attached to the base bottom cover.

Orient the frame so the cut out is facing the QT Py's USB-C Port.

Press the parts together to snap fit them together.

Connect Button to QT Py
Grab the cables from the QT Py and push button and connect them together.
Install NeoPixel Cable
Grab the 3-pin cable from the QT Py and thread the connector through the holes in the base cover, stand frame and bottom cover of the LED case.

Secure Base to Stand
Snap fit the base cover to the base frame by firmly pressing the two parts together.

Make sure the wires from the 2-pin cable is not kinking.

Ensure the 3-pin cable is still threaded through the holes and sticking out the top of the LED bottom cover.
Install Frame
Get the framing of the LED case ready to attached to the bottom cover.

Snap fit the bottom cover into the frame.

NeoPixel Order
Check the order of the NeoPixel LEDs are correct with the first NeoPixel being on the far right.

Orient the base so the USB port is facing the back and the push button is facing the front.
Connect QT Py to NeoPixel Strand
Grab the connector from the QT Py and connect it to the first NeoPixel.

Get the top cover ready to install onto the framing of the LED case.

Secure Top Cover to Frame
Press the edges from the top cover so they snap fit into the framing of the LED case.
Install Stand Covers
Grab the two covers for the stand and snap fit them onto the framing of the stand.

Keep the wires from the cable inside the stand and avoid kinking any wires.

Installed Stand Covers
Double check the edges are all flush so the covers are installed properly.
Final Build
Congratulations! Your NeoPixel Menorah is ready for lighting up Hanukkah!