NeoTrellis Light Painting

Created by Anne Barela

https://learn.adafruit.com/neotrellis-light-painting

Last updated on 2021-11-15 07:26:09 PM EST

Table of Contents

Overview

The NeoTrellis M4 is "some awesome hardware" as Charlotte the Spider would say. There are some pretty great demos (https://adafru.it/Dm1) on the Adafruit Learning System for using it.

All work and no play, well let's say the NeoTrellis isn't going to heavy lifting 24/7. Say you or your young one want a simple game in between gigs?

This tutorial turns your NeoTrellis into a simple paint program in seconds. Fun for young and old. You can change the keys to any of 12 colors (plus off/black).

Parts

Adafruit NeoTrellis M4 with Enclosure and Buttons Kit Pack

So you've got a cool/witty name for your band, a Soundcloud account, a 3D-printed Daft Punk...

https://www.adafruit.com/product/4020

USB cable - USB A to Micro-B

This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...

https://www.adafruit.com/product/592

Optional portable power

[USB Battery Pack - 2200 mAh Capacity - 5V 1A Output](https://www.adafruit.com/product/1959)
A smaller-sized rechargeable battery pack for your Raspberry Pi or Raspberry...
https://www.adafruit.com/product/1959

Code and Play

Plug your NeoTrellis into a USB port on your computer. In your file explorer/finder, you should see a new flash drive named CIRCUITPY pop up.

If you see a new drive named TRELM4BOOT, you will need to load CircuitPython onto your NeoTrellis first. See this page to do so (https://adafru.it/C-O). Your board should reboot when you put the code on the board and a CIRCUITPY drive should now be available.

Library

You will need one library file in the /lib directory on your NeoTrellis. Check for the file adafruit_trellism4.mpy. If you do not have this file, you can download the CircuitPython 4.x library bundle available at this link (https://adafru.it/zB-). Go into the zip file, get the adafruit_trellism4.mpy file and put it on your CIRCUITPY drive as /lib/adafruit_trellism4.mpy (create the /lib directory if it isn't there).

Code

Copy the code below over to the CIRCUITPY drive as code.py. The program will start as soon as the code is in the drive. The NeoTrellis will clear all the buttons on startup.

```
# Simple paint program for Trellis M4 Express
# Press any button it will cycle through a palette of colors!
#
# Anne Barela for Adafruit Industries   November, 2018
#
import time
import adafruit_trellism4

trellis = adafruit_trellism4.TrellisM4Express()
```

```
# See https://www.w3schools.com/colors/colors_picker.asp
RED = 0xFF0000
ORANGE = 0xB34700
YELLOW = 0xFFFF00
OLIVE = 0x66DD00
GREEN = 0x008000
AQUA = 0x00FF66
TEAL = 0x00BFFF
BLUE = 0x0080FF
NAVY = 0x000080
MAROON = 0x800000
PURPLE = 0x800080
PINK = 0xFF66B3
WHITE = 0xFFFFFF
BLACK = 0x000000

color_cycle = [BLACK, RED, ORANGE, YELLOW, OLIVE, GREEN, AQUA,
               TEAL, BLUE, NAVY, MAROON, PURPLE, PINK, WHITE]

colors = 13  # Number of colors in color_cycle

key_state = [0 for _ in range(32)]  # All keys are color 0 (BLACK)

trellis.pixels.fill(BLACK)  # Turn off all pixels

current_press = set()
while True:
    pressed = set(trellis.pressed_keys)
    for press in pressed - current_press:
        if press:
            print("Pressed:", press)
            x, y = press
            pixel = (press[1] * 8) + press[0]
            if key_state[pixel] == colors:  # If we're at white
                key_state[pixel] = 0        #  Set back to black
            else:
                key_state[pixel] += 1       # Use next color
            # Change the pushed pixel to the next color
            trellis.pixels[x, y] = color_cycle[key_state[pixel]]

    time.sleep(0.08)
    current_press = pressed
```

Play!

Press the buttons to get colors. Use multiple presses to advance through the rainbow available. You can change the color of a button any time just by pushing it and cycling through the color palette.

Customization

The colors are defined in the code as RED, ORANGE, YELLOW, OLIVE, GREEN, AQUA, TEAL, BLUE, NAVY, MAROON, PURPLE, PINK, WHITE, and BLACK. You can change the value of any of these colors by changing the hexadecimal code next to the name.

Use any text editor to change the code. We suggest the Mu editor but you can use any editor that saves plain text. See this guide on installing Mu (https://adafru.it/ANO).

What values are good? I like to use the color picker at w3schools.com (https://adafru.it/BPY) to make choices. Get the 6 digit alphanumeric (hexadecimal) code and put it after the # (which tells CircuitPython that the number following it is hexadecimal).

If you are adventurous, you can add or delete colors from `color_cycle`, just keep `BLACK` & `WHITE` where they are and change the variable `colors` to the number of colors in `color_cycle` including `BLACK` & `WHITE`.

The code looks for key pressed events and selects the next color in the palette when it detects a key has been pressed.

There is no save/load in the code but you can look to modify the code as you'd like.