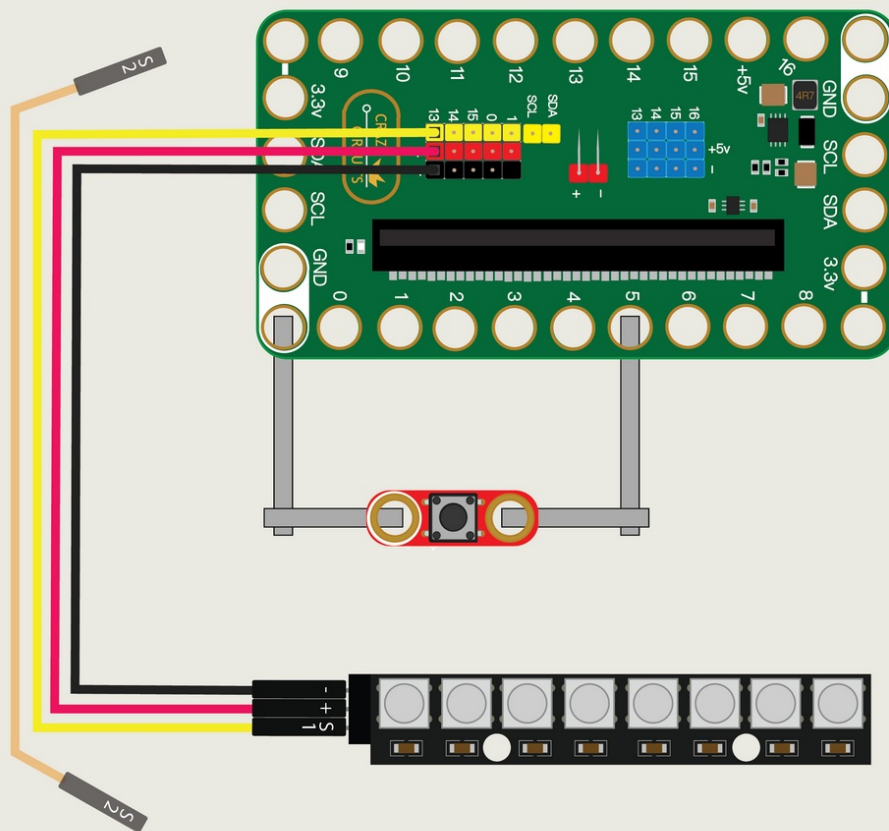




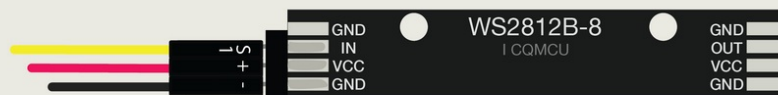
CH22-C - NeoPixel Rotate Speed

Connect a NeoPixel Stick and a Pushbutton to a Bit Board and control them with code.

Written By: Pete Prodoehl



Back of NeoPixel Strip



INTRODUCTION

Connect a NeoPixel Stick and a Pushbutton to a Bit Board and control them with code.

We'll explore how to control an Addressable RGB LED Strip (also called "NeoPixels").



TOOLS:

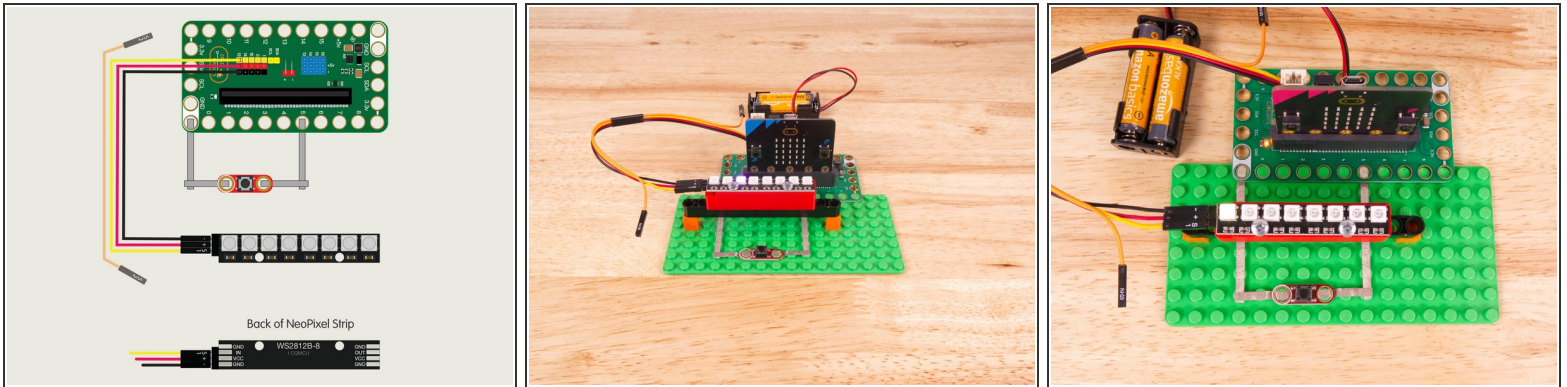
- [Computer](#) (1)
- [Scissors](#) (1)



PARTS:

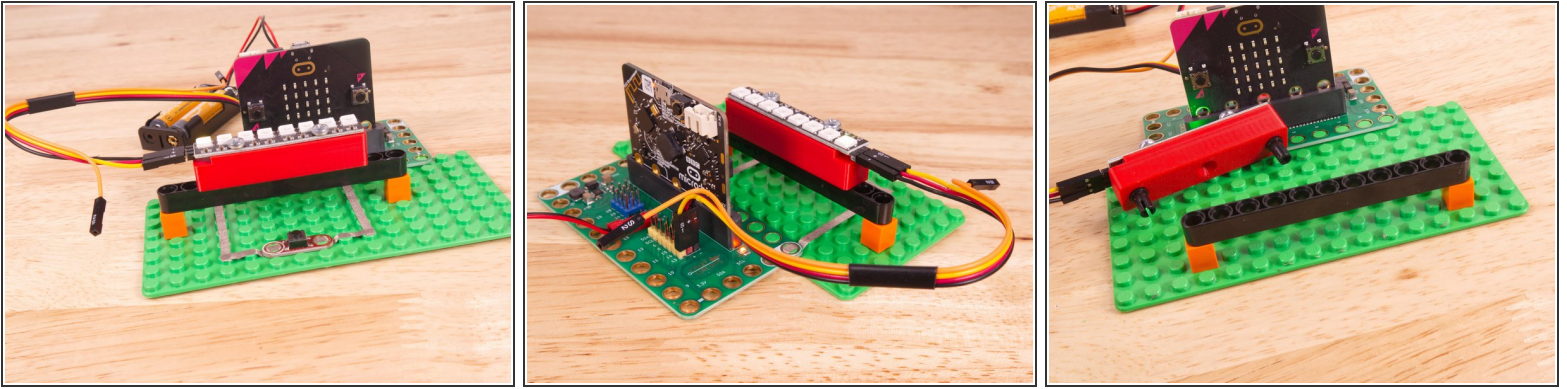
- [Crazy Circuits Bit Board](#) (1)
 - [micro:bit](#) (1)
 - [Addressable RGB LED Strip](#) (1)
 - [Standard Pushbutton Chip](#) (1)
 - [Crazy Circuits Ribbon Cable](#) (1)
 - [Maker Tape](#) (1)
- 1/8" Wide

Step 1 — Connect the NeoPixel Stick and Button



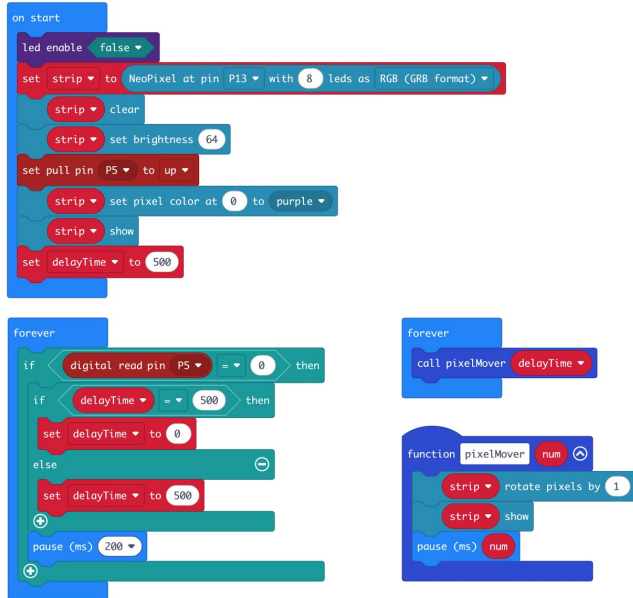
- The magical thing about these “Individually Addressable” LEDs is that you don’t have to connect each LED to an individual pin on your micro:bit, you just need **one signal wire** along with power (**Positive** and **Negative**) to light things up.
- So let's get connected! If you turn the **NeoPixel Strip** over you'll see labels for the three pins. You should see **IN**, **VCC**, and **GND**.
- Use the [Crazy Circuits Ribbon Cable](#) to connect the **NeoPixel Strip** to the **Bit Board**.
- The end with **S1**, **+**, and **-** can plug directly into the **Pin 13** row (make sure S1 goes into 13) and the other end should plug into the **NeoPixel Strip** as shown.
- ☑ We only need three wires for the **NeoPixel Strip** so you can leave **S2** (the Orange wire) unconnected.
- We'll also connect a button using Maker Tape. Connect it to **Pin 5** and **GND**. (Polarity does not matter with these buttons.)

Step 2 — Mount the NeoPixel Stick



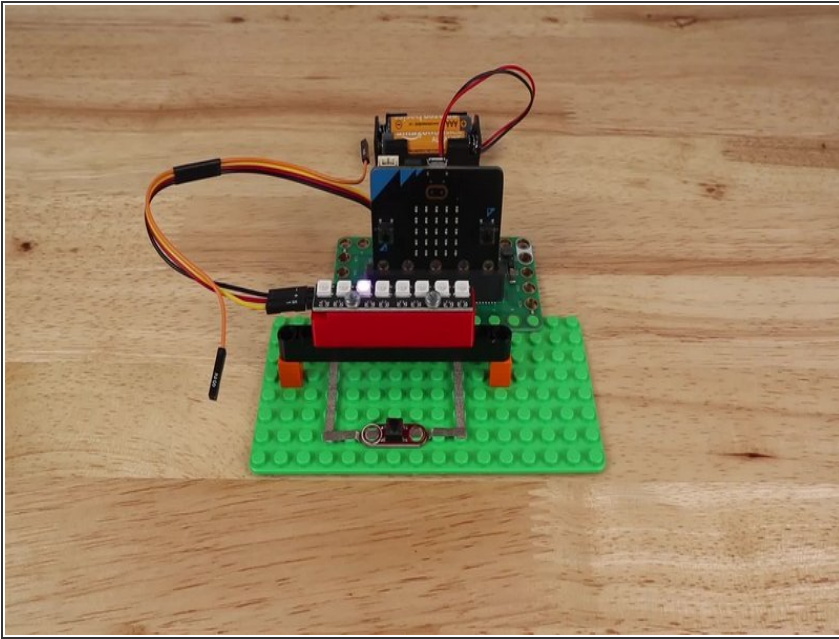
- To mount the NeoPixel Stick in place we used our [NeoPixel Beam Holder](#) with a long Technic Beam and two 1x1 LEGO bricks.
- You can always use rubber bands, tape, or some other method.
- The important thing is to make sure the connector is on the left hand side so the number index for the NeoPixel strip makes sense.

Step 3 — Load the Code



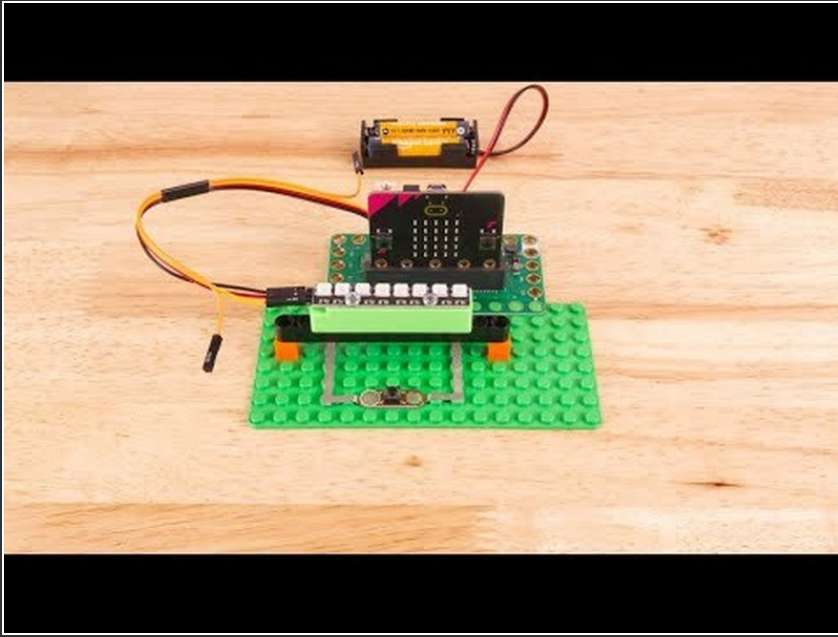
- If you've never used a micro:bit before you'll want to check out this guide: [Bit Board V2 Setup and Use](#)
- We're going to load the following code for our **NeoPixel Rotate Speed** program:
https://makecode.microbit.org/_ebzKmh61i...
- We will once again check for a button press and if we see one we will toggle between two different **delayTime** values. (We'll also pause after as a primitive **debounce** mechanism.)
- A second forever loop call our pixelMover function, passing in the **delayTime**.
- Finally, our pixelMover function runs and a pixel is *rotated* with the delay we've set between each **move/rotation**.

Step 4 — Test it Out!



- Once the code is loaded it should start running immediately.
- ⓘ You can power the micro:bit via the USB cable you used to load the code or you can use a battery pack plugged into the Bit Board.
- You'll see the pixel move from left to right at a slow speed.
- Press the button down and the pixel should speed up! Press it again to slow it down.
- ☑ Things may or may not work as expected depending on how long the button is held down. We covered some more advanced button behavior in [Chapter 4 - Single Button](#).
- ☑ See [CH4-D - Button Toggle Better](#) for a good "toggle" example.

Step 5 — Follow Along



- Follow along with our recorded Live Stream!
- ① You can watch the full video of us walking through this project, along with explaining and exploring the code:
<https://youtube.com/live/8EJiYAk8qk>