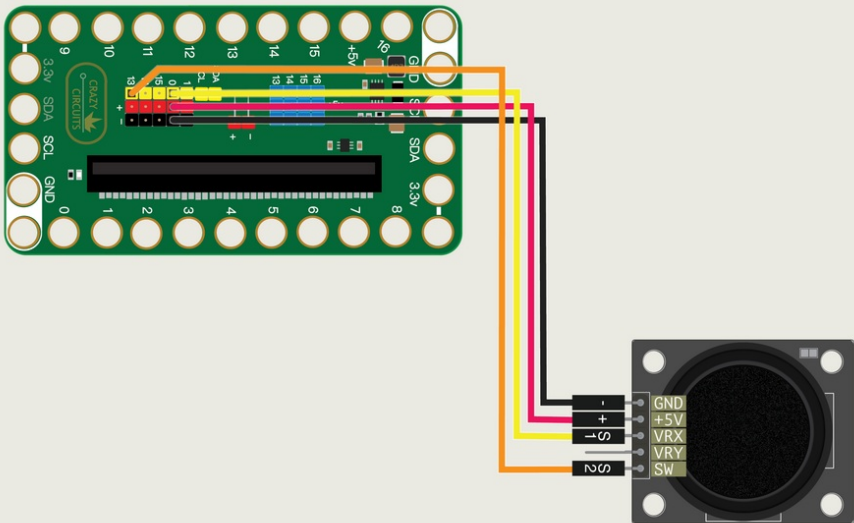




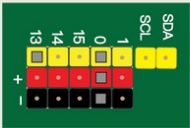
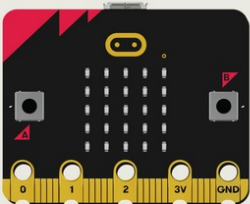
CH40-D - Thumbstick LED Matrix Map

Connect a Thumbstick to a Bit Board and control the built-in LED Matrix on the micro:bit.

Written By: Pete Prodoehl



TSTK	Bit Board
GND	-> - (Negative) Pin
+5V	-> + (Positive) Pin
VRX	-> PIN 0 (S1)
VRY	-> (not used)
SW	-> PIN 13 (S2)



INTRODUCTION

Connect a Thumbstick to a Bit Board and control the built-in LED Matrix on the micro:bit.

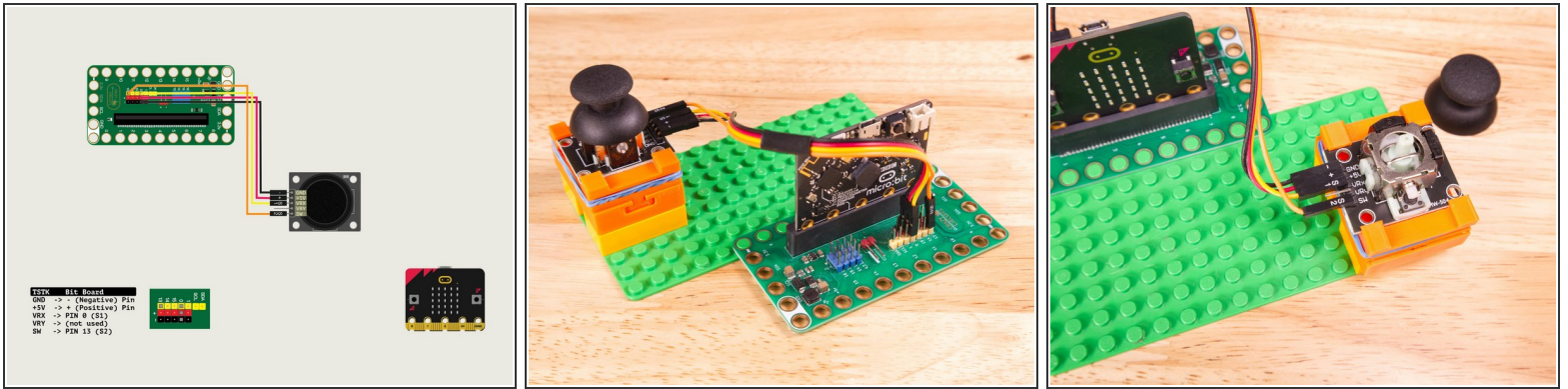
 TOOLS:

- [Computer](#) (1)

 PARTS:

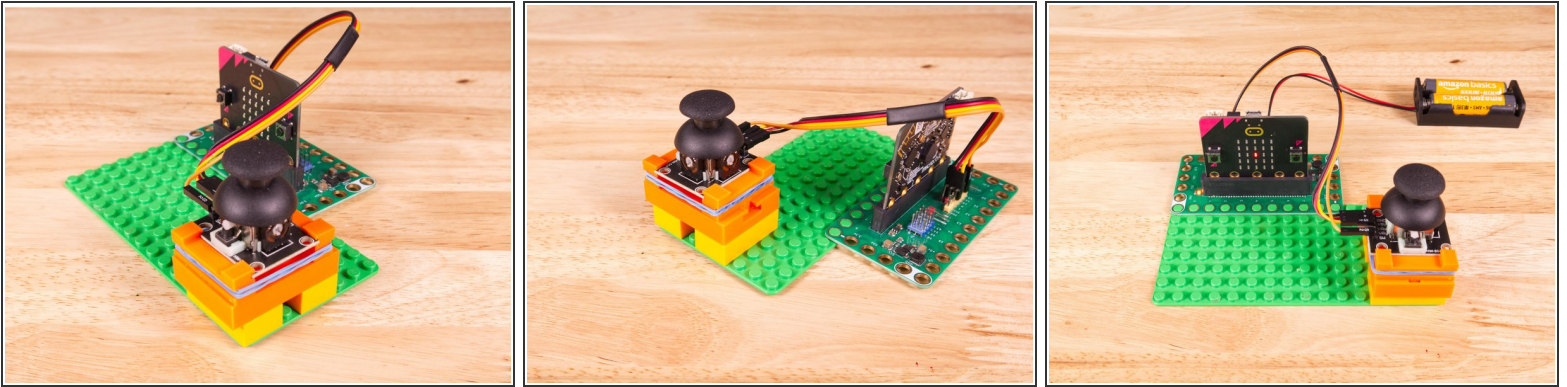
- [Crazy Circuits Bit Board](#) (1)
- [micro:bit](#) (1)
- [Crazy Circuits Ribbon Cable](#) (1)
- [Thumbstick](#) (1)

Step 1 — Connect the Thumbstick



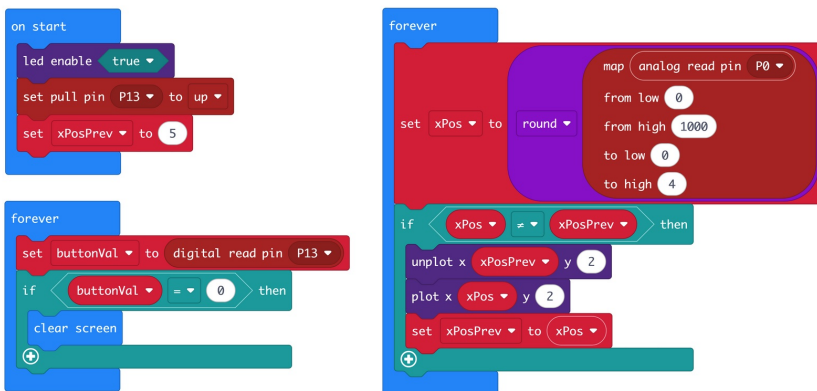
- We'll connect the **Thumbstick** with a **Crazy Circuits Ribbon Cable** using all four wires.
 - Plug the three-wire connector of the **Crazy Circuits Ribbon Cable** into the row with **Pin 0** on the Bit Board. The wire colors should match the pin header colors, with **S1** (yellow) going to the yellow pin.
 - Next, plug **S2** (orange) into **Pin 13**. *(S2 is separated from the three-wire connector to allow you to plug it into another row.)*
 - To the Thumbstick plug - (black) into **GND**, + (red) into **+5V**; **S1** (yellow) into **VRx**; **S2** (orange) into **SW**.
- ☐ We won't plug anything into **VRy** pin for this example.

Step 2 — Mounting the Thumbstick



- We used a 3D printed part we designed to hold the Thumbstick in place on a LEGO baseplate.
- We attached two 2x4 LEGO Bricks to the bottom of the 3D printed holder to allow it to snap onto the LEGO Baseplate a bit easier.
- Alternately you can try to use some LEGO bricks and some rubber bands to hold the sensor in place, or just hold it in your hand.
- 3D print your own [Thumbstick Sensor Holder](#).

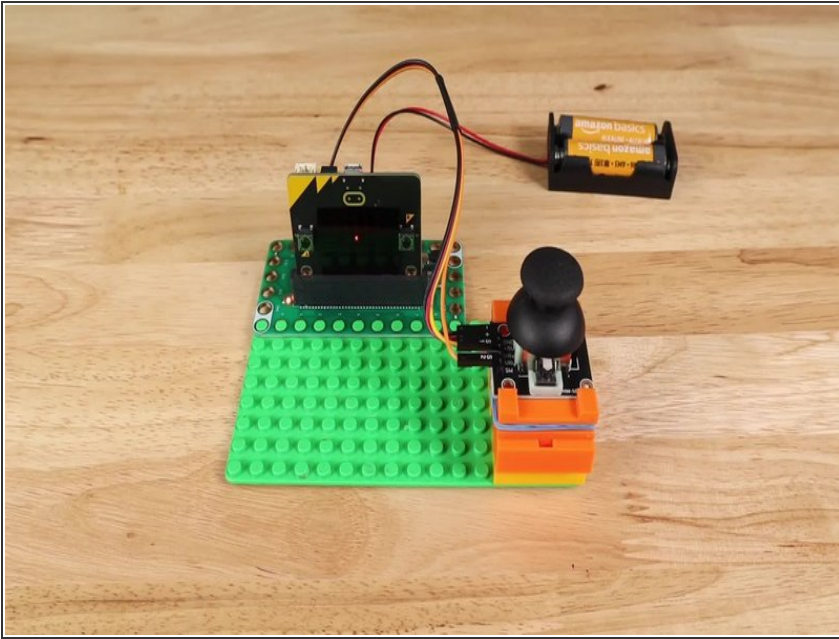
Step 3 — Load the Code



- 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 **Thumbstick LED Matrix Map** program: https://makecode.microbit.org/_fAJLew4y2...
- In the on start block we'll enable the LED matrix and set **Pin 13** to **pull up**.

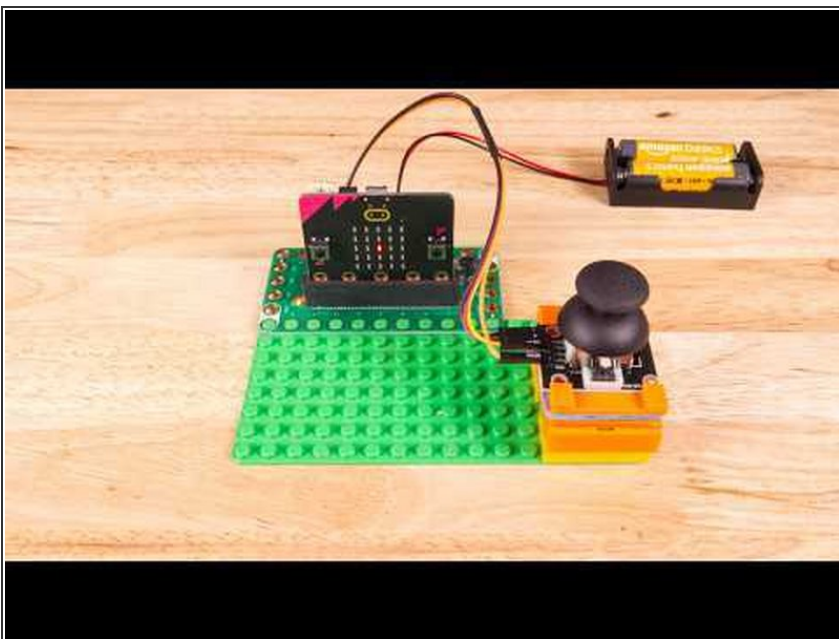
- In one forever loop we'll set **buttonVal** to read **Pin 13** (which is connected to the button/switch on the Thumbstick) and if it's pressed we'll clear the screen.
- In the other forever loop we'll set **xVal** using the map function. We also use round to ensure we get an **integer** (whole number) rather than a **floating point** (decimal) value.
- We then check if **xPos** is *not* equal to **xPosPrev** before we unplot the old value and then plot the new value.
- Finally, we store the value of **xPos** in the variable **xPosPrev** for the next time around.
- ☒ This code functions just like the previous example but with *much less code*. Typically less code is easier to troubleshoot and debug. (As long as it's not overly complicated!)

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.
- Move the Thumbstick left and right and you should see different pixels light up. Press the button to clear the screen.
- We get the same final results as the previous version but with much less code.

Step 5 — Follow Along



- Follow along with our recorded Live Stream!
- ❗ You can watch the full video of us walking through this project, along with explaining and exploring the code:
<https://www.youtube.com/live/itzcDaRW9Q>