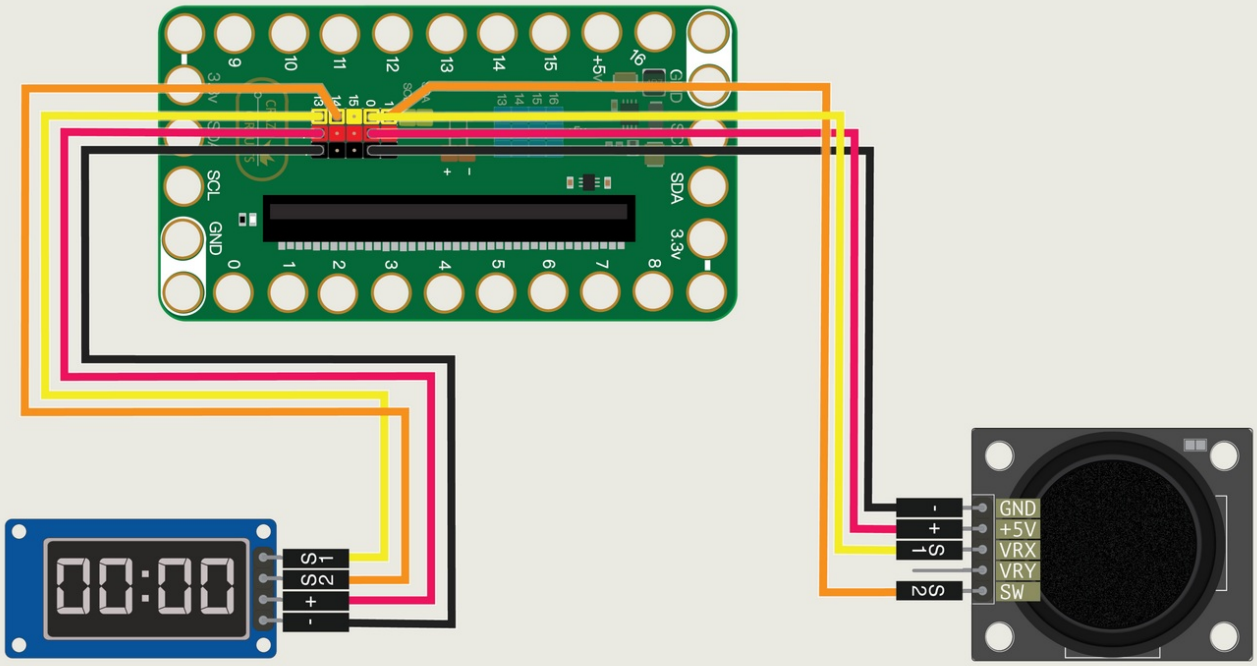
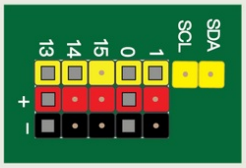
CH43-D - Thumbstick 7 Segment Binary

Connect a Thumbstick to a Bit Board and view the output using a 7 Segment Display.

Written By: Pete Prodoehl



TSTK	Bit Board
GND	-> - (Negative) Pin
+5V	-> + (Positive) Pin
VRX	-> PIN 0 (S1)
VRY	-> (not used)
SW	-> PIN 1 (S2)



7SEG	Bit Board
CLK	-> Pin 13 (S1)
DIO	-> Pin 14 (S2)
VCC	-> + (Positive) Pin
GND	-> - (Negative) Pin

INTRODUCTION

Connect a Thumbstick to a Bit Board and view the output using a 7 Segment Display.

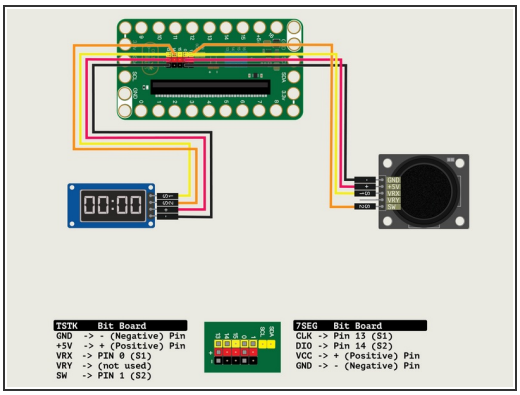
TOOLS:

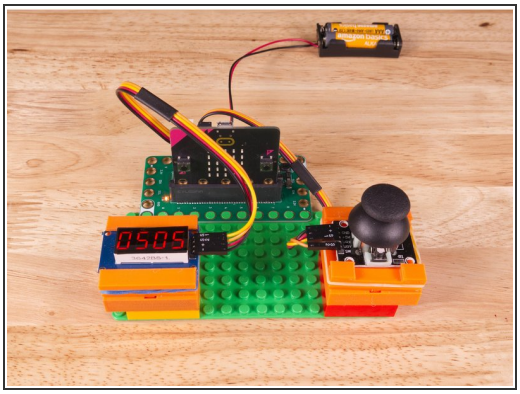
- Computer (1)

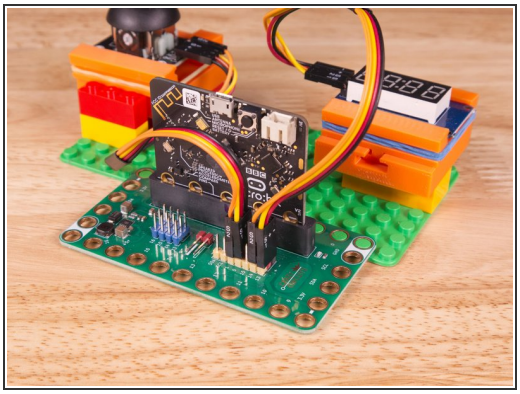
PARTS:

- Crazy Circuits Bit Board (1)
- micro:bit (1)
- Crazy Circuits Ribbon Cable (2)
- Thumbstick (1)
- 7 Segment Display (1)

Step 1 — Connect the Thumbstick



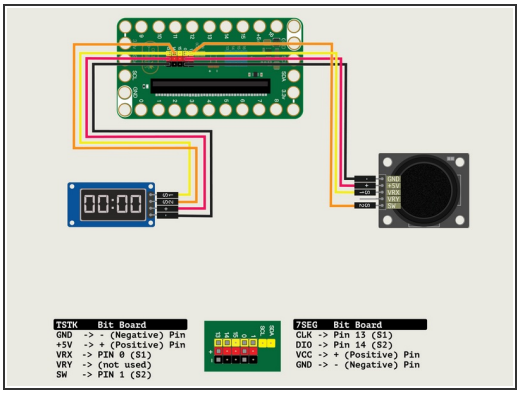


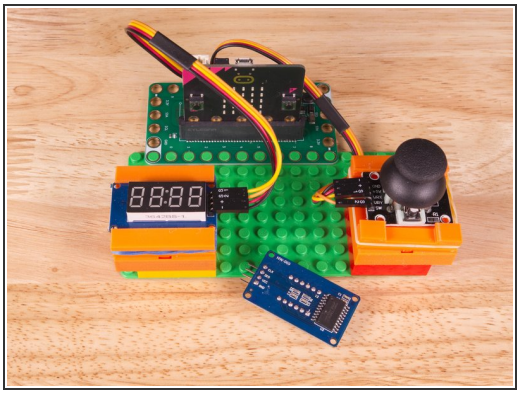


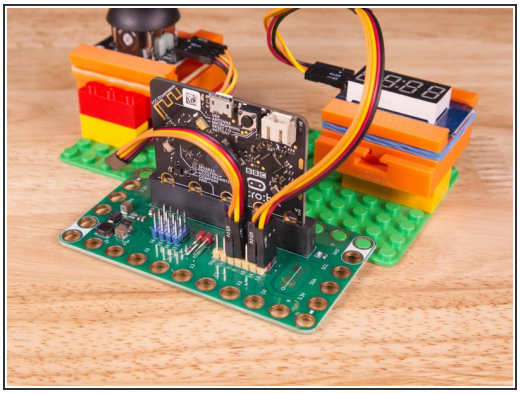
- We'll connect the **Thumbstick** with a **Crazy Circuits Ribbon Cable** using all four wires.
- Plug the three-wire connector of the **Crazy Circuits Ribbon Cable** into the row with **Pin 0** on the Bit Board. The wire colors should match the pin header colors, with **S1** (yellow) going to the yellow pin.
- Next, plug **S2** (orange) into **Pin 1**. *(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 — Connect the 7 Segment Display



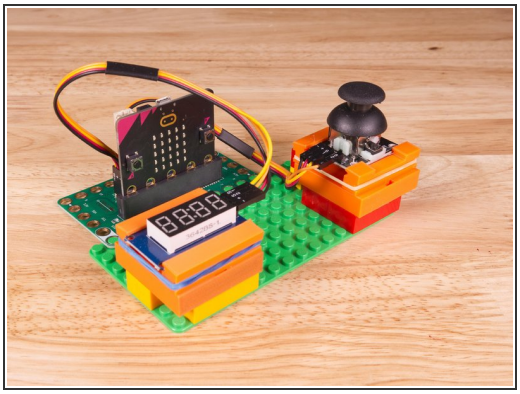
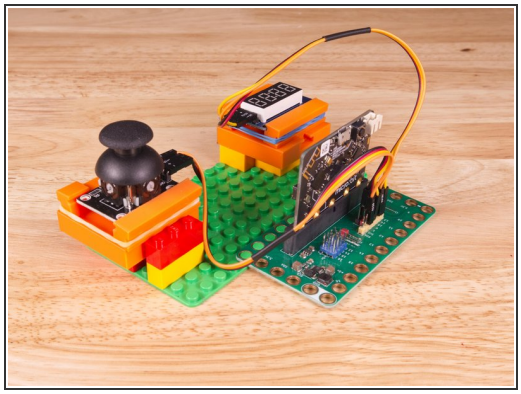
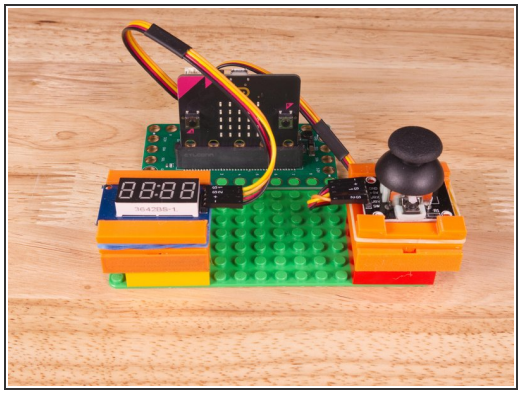




- We'll connect the **7 Segment Display** with a **Crazy Circuits Ribbon Cable** using all four wires.
- Start by plugging the connector with the three wires in it (Black, Red, Yellow) into the **Pin 13** row of the Bit Board. (Make sure the Black wire is in **-** and the Yellow wire labeled **S1** is closest to the number **13** on the Bit Board.)
- We can then plug the Orange wire labeled **S2** into **Pin 14** on the Bit Board.
- Now we'll connect the other end of the Ribbon Cable to the **7 Segment Display**.
- The pin labels are on the back of the 7 Segment Display PCB. From top to bottom it's **CLK**, **DIO**, **VCC**, **GND**.
  - **CLK** stands for **Clock** and **DIO** stands for **Data Input/Output**. **VCC** is the **Positive** (+) and **GND** is the **Ground** (-) or **Negative**.
- **S1** (Yellow) goes to **CLK**, **S2** (Orange) goes to **DIO**, and then **+** (Positive) goes to **VCC** and **-** (Ground or Negative) goes to **GND**.

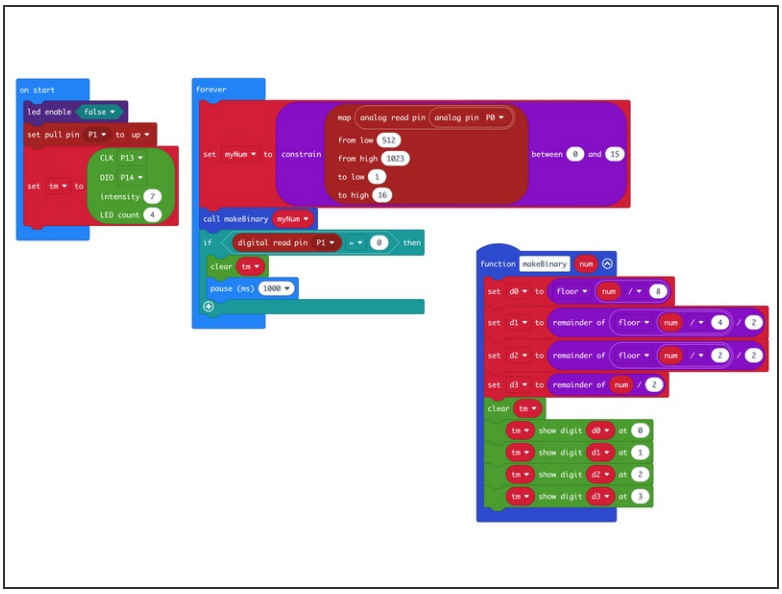
Make sure every wire is connected in the correct place, or our display will not work!

Step 3 — Mounting the Thumbstick and 7 Segment Display

- We used some 3D printed parts we designed to hold the Thumbstick and 7 Segment Display in place on a LEGO baseplate.
- We attached two 2x4 LEGO Bricks to the bottom of the 3D printed holders 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/tape to hold the sensors in place.
- You can always 3D print your own Thumbstick Sensor Holder and 7 Segment Display Holder if you need them.

Step 4 — Load the Code



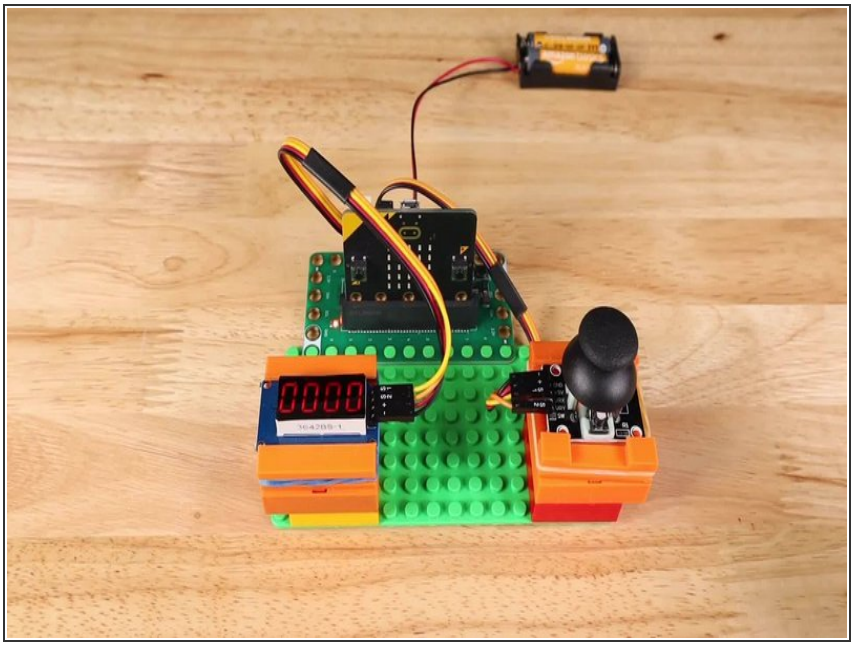
Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

- 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 7 Segment Binary** program: https://makecode.microbit.org/_g3dRC8077...
- In the on start block we'll disable the LED matrix and set **Pin 1** to **pull up**, as we'll attach this to the button on the Thumbstick.
- We'll also set up our 7 Segment Display by setting **CLK** to use **Pin 13** and **DIO** to use **Pin 14**. (We assign the display to a variable named **tm**.)
- In the forever loop we'll first clear the display, and then we'll read **Pin 0**, map the value from the input (**0 to 1023**) to an output ranging from **0000 to 1111**... Yes, it's Binary!
- Our function **makeBinary** does a bit of math to convert the decimal into a 0 or 1 and then sets the specific position on the display. The floor function is basically a "round down" operation and the remainder of function (normally called Modulo) is used to determine if a number is odd or even.

When you see a binary string like **1010** you need to assign a value, from left to right it's **8,4,2,1**. So **1010** would be **10** in decimal (8 plus 2). **0111** would be **7** in decimal. With four different values this would be considered **4 bit**.

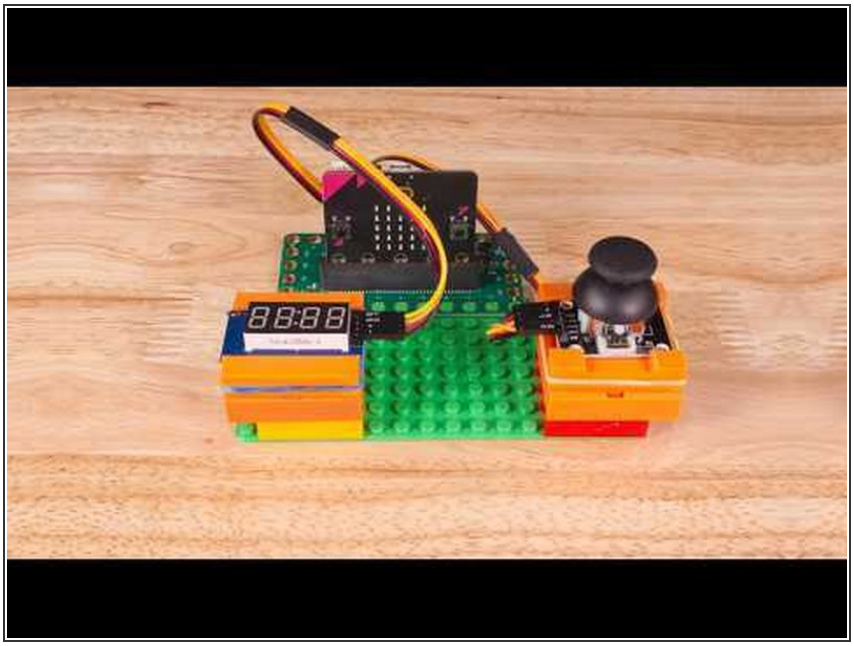
Learn more about Binary, Decimal and Hexadecimal Numbers!

Step 5 — 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 from the center to the right and you should see the value shown on the 7 Segment Display change. (Displayed in **Binary** this time!)
- Press the Thumbstick (button) down and the display will clear for 1 second.

Step 6 — 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/8NoqL68PDVw

© 2026