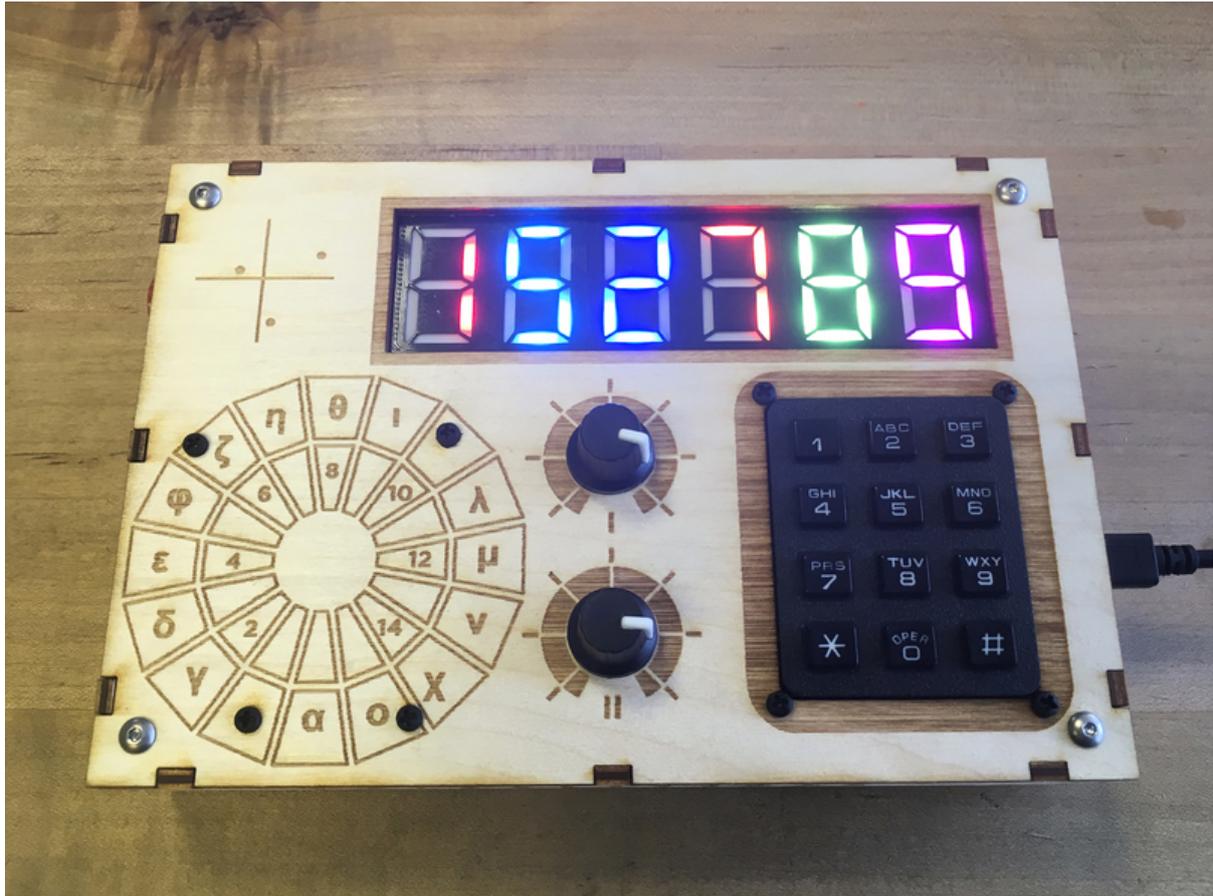




Mystery Box: NeoMatrix Mk I

Created by John Park



<https://learn.adafruit.com/mystery-box-neomatrix-mk-i>

Last updated on 2024-03-08 02:52:54 PM EST

Table of Contents

Overview	3
Make the Panels	6
Build the NeoMatrix	8
<ul style="list-style-type: none">• Keypad• Display Modules• Metro Mounting• Potentiometer Mounting• Piezo Buzzer• Side Panels• Pushbutton• USB Jack• Wiring Everything Together• Low Profile Interconnects• NeoSegment Wiring• Piezo Connection• Potentiometer Wiring• Pushbutton Wiring• USB Port• Closing the Case	
Code and Use	36
<ul style="list-style-type: none">• Board Manager• Keypad• NeoSegment• Coding	

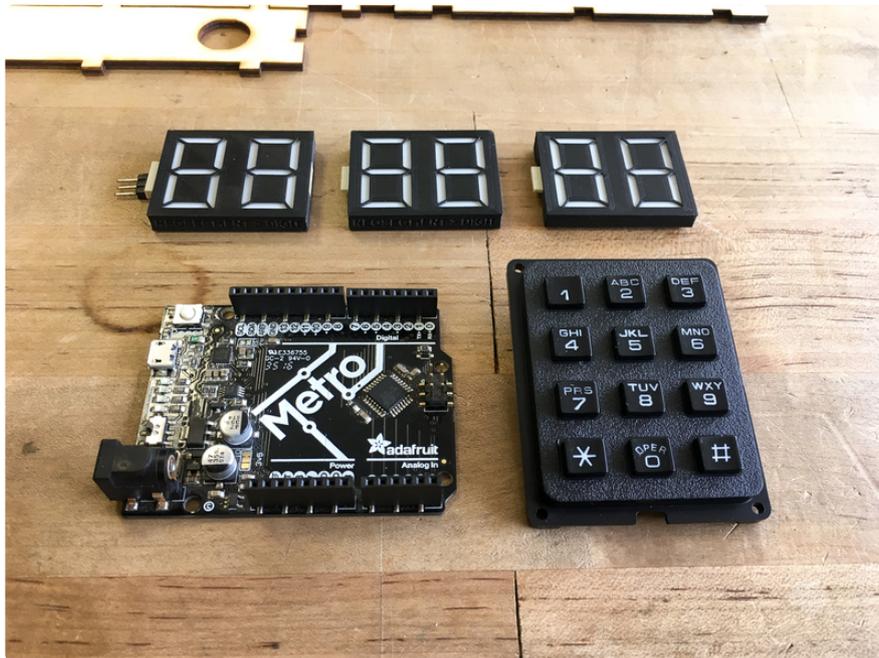
Overview

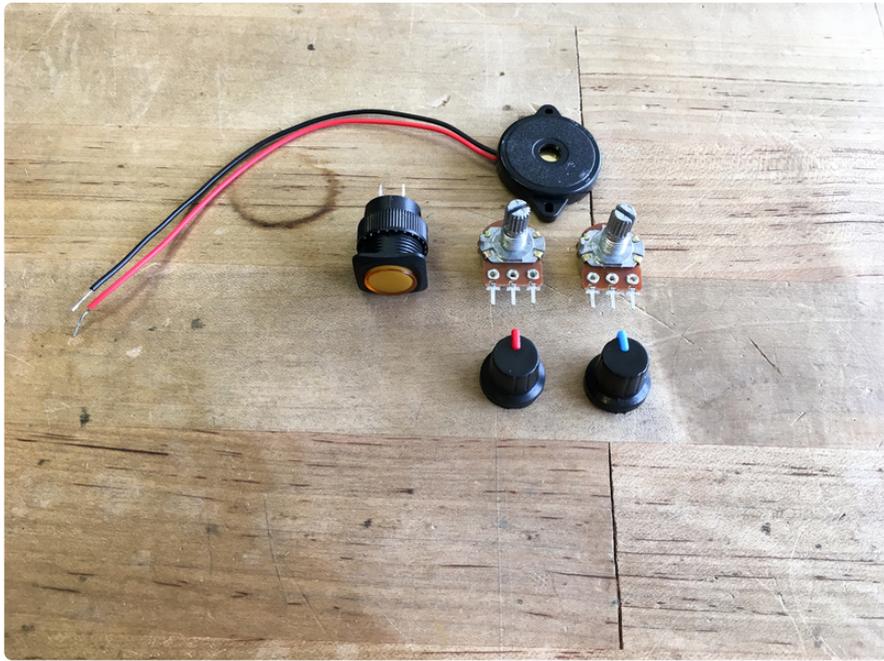
While the NeoSegments are not currently available, this guide remains here for archival purposes, or can be adapted for other display types.

Imagine all of the things you can do with a NeoMatrix Mk I, the Interface of Infinite Possibilities! From puzzle box to step sequencer, from NeoPixel programmer to crypto generator, from TV prop to phone phreaking device, the choices are endless. Think of it as a platform for experimentation and interface design.

At the core of the NeoMatrix Mk I is a [NeoSegment RGB LED character display \(https://adafru.it/-Eo\)](https://adafru.it/-Eo), driven by an Adafruit Metro with a phone pad 3x4 button matrix and two knobs as inputs. It also has a built in piezo buzzer for sound indicators.

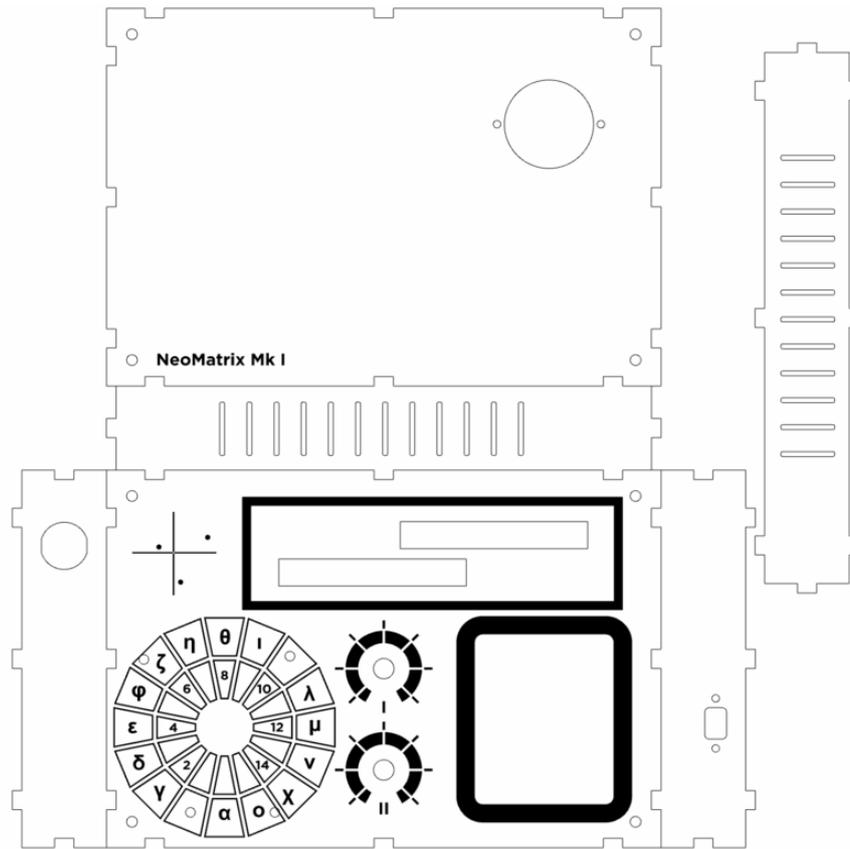
The NeoMatrix Mk I has a lighted pushbutton that can be used for any purpose you like, and an externally accessible USB port for data and power. You could also convert it to run from battery power with a USB battery pack or a LiPo battery and PowerBoost.







Make the Panels



You can house your NeoMatrix Mk I circuit, display, and inputs in any sort of enclosure you like -- 3D printed, cut from cardboard, fashioned from wood, it's up to you. If you'd like to cut one on a laser cutter, you can use the plans here. They are designed to be cut from 1/8" material -- either baltic birch plywood or acrylic will work very nicely.

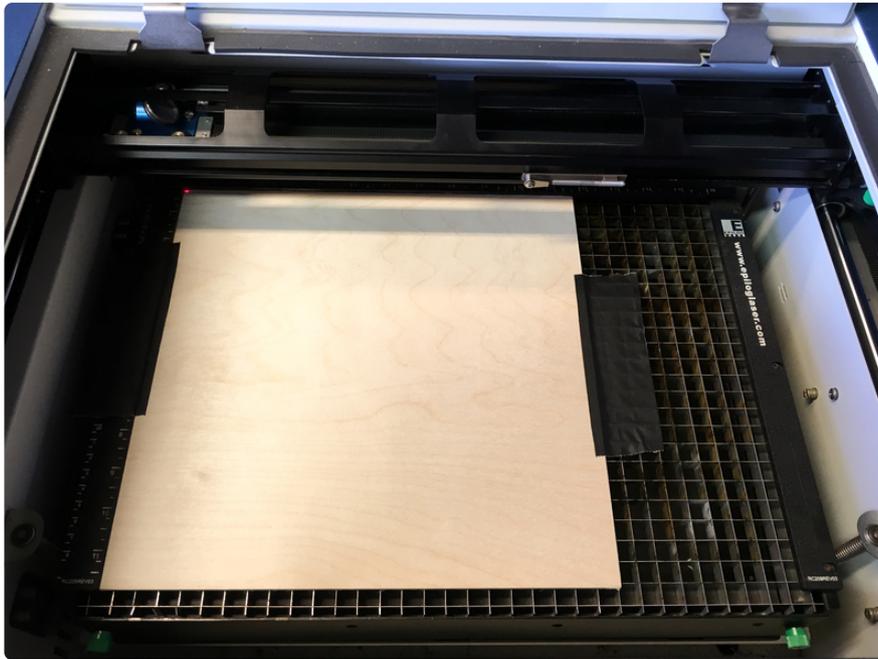
Download the two .svg files below and prepare them for etching and cutting on the laser cutter software of your choice. I used an Epilog Zing 40 Watt laser cutter with the print driver from within Rhino.

The neoMatrixBox_raster.svg file is for engraving the graphics. It's helpful to engrave the material first, so that if any parts shift after cutting the graphics are already registered properly. I used these etching settings:

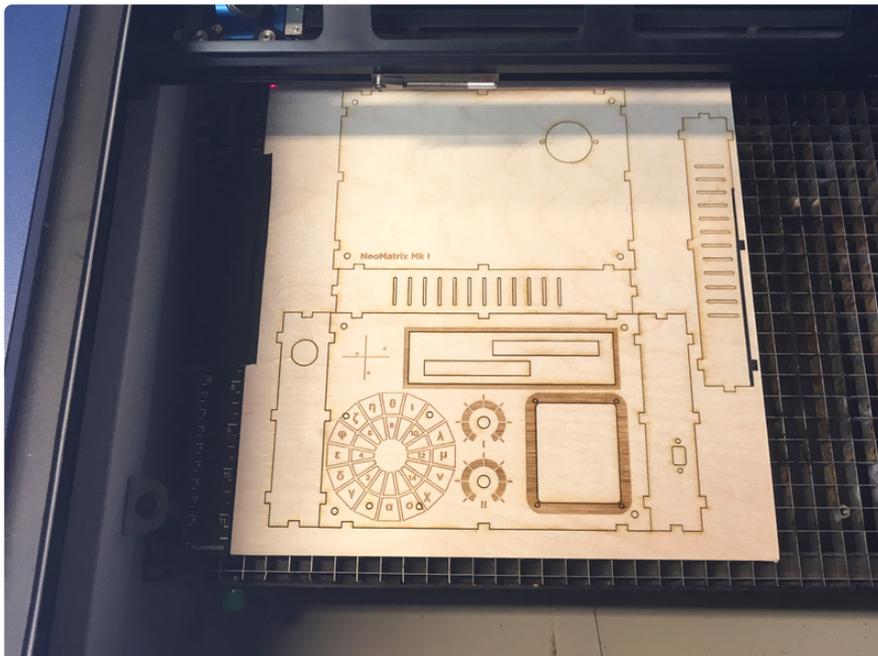
- Speed: 70%
- Power: 70%
- DPI: 500

neoMatrixBox_raster.svg

<https://adafru.it/CmP>



Everything fits on a 12" x 12" piece of material. Here you can see that I'm using a bit of tape at the edges to fight the slight warping of the wood, which can throw off the laser focus.



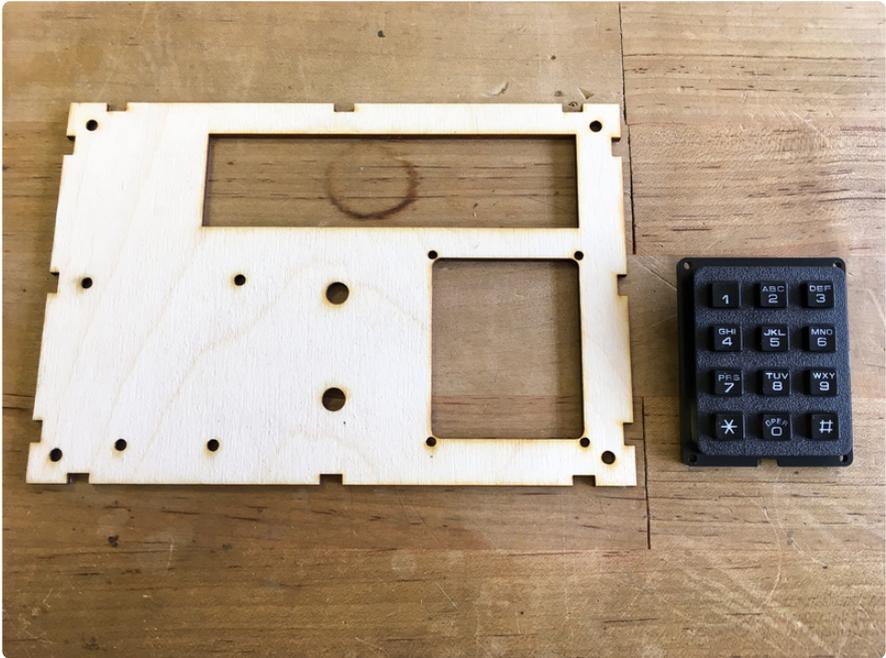
Etching graphics is an optional step. You'll notice in some of the photographs the first iteration of the NeoMatrix Mk I was made without graphics.

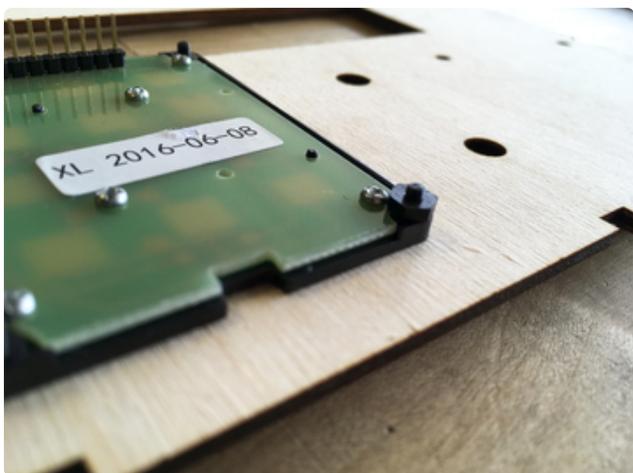
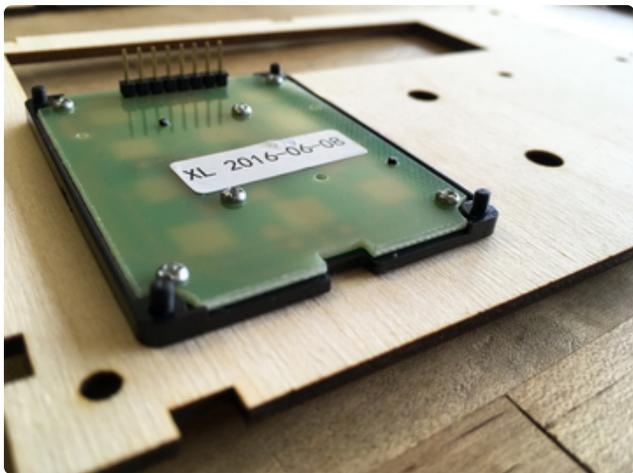
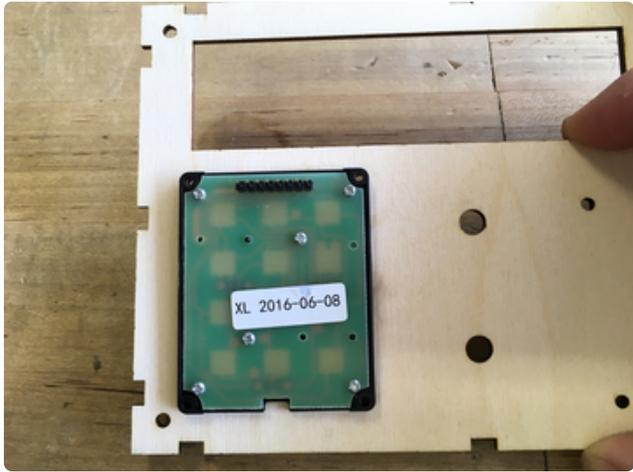
The neoMatrixBox.svg is the cutting file. I used the following settings:

- Speed: 60%
- Power: 55%

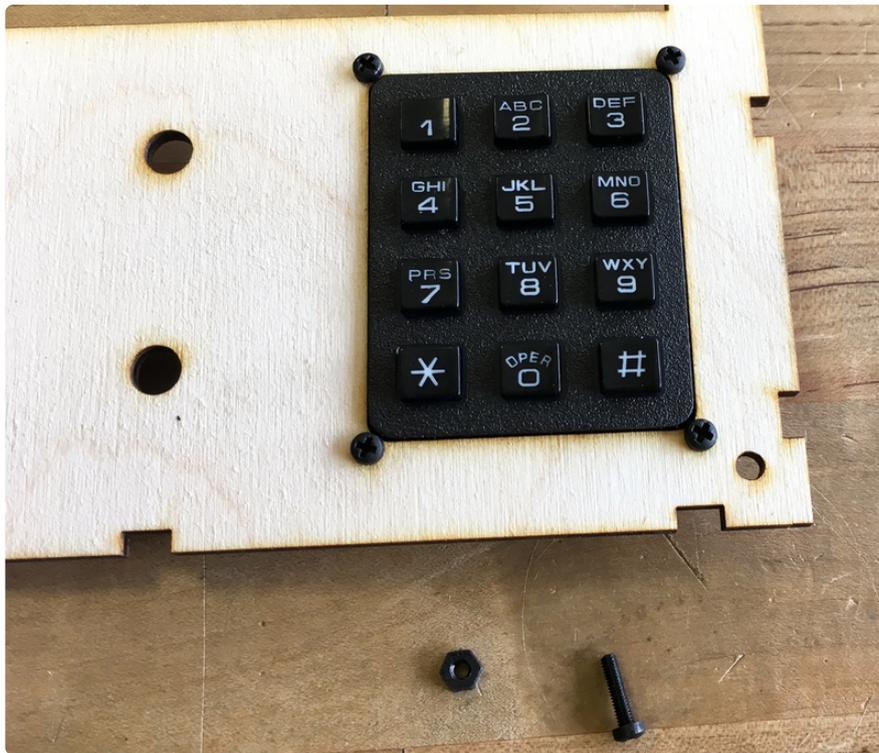
Keypad

Begin assembly by affixing components to the inside of the front panel, starting with the keypad.



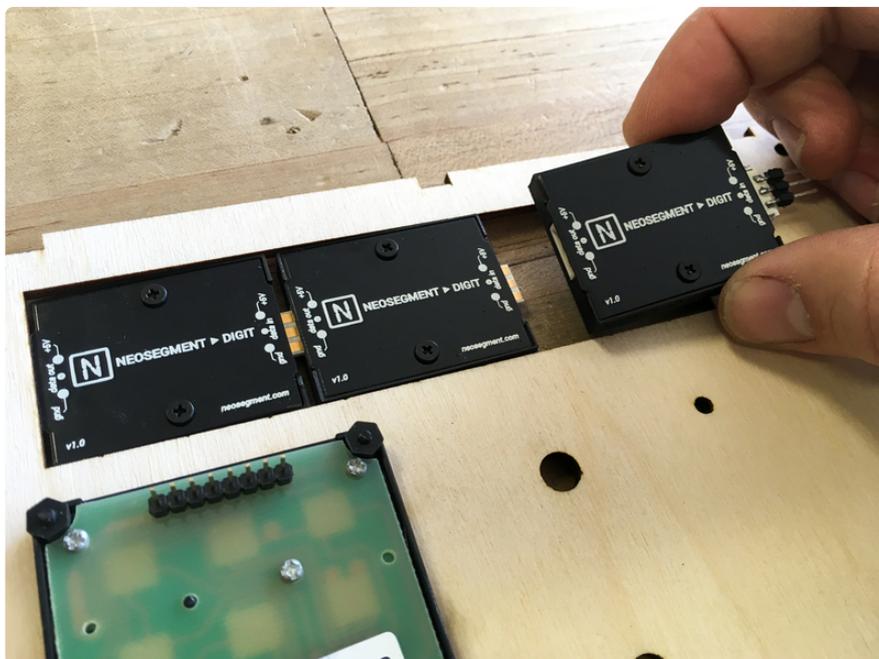


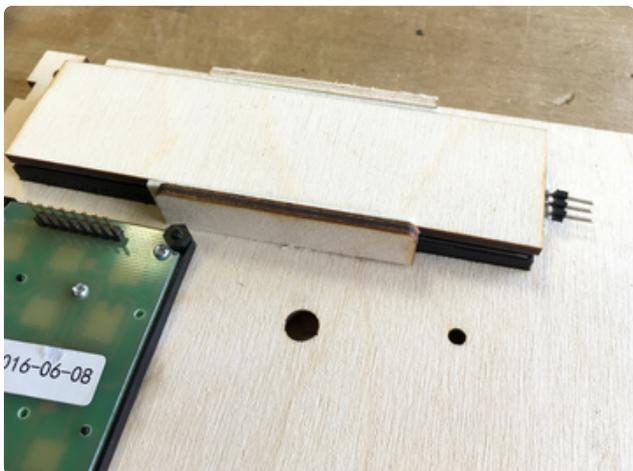
The keypad fits in place from the inside and you can then secure it with four of the 9.5mm long M2.5 nylon screws and nuts.



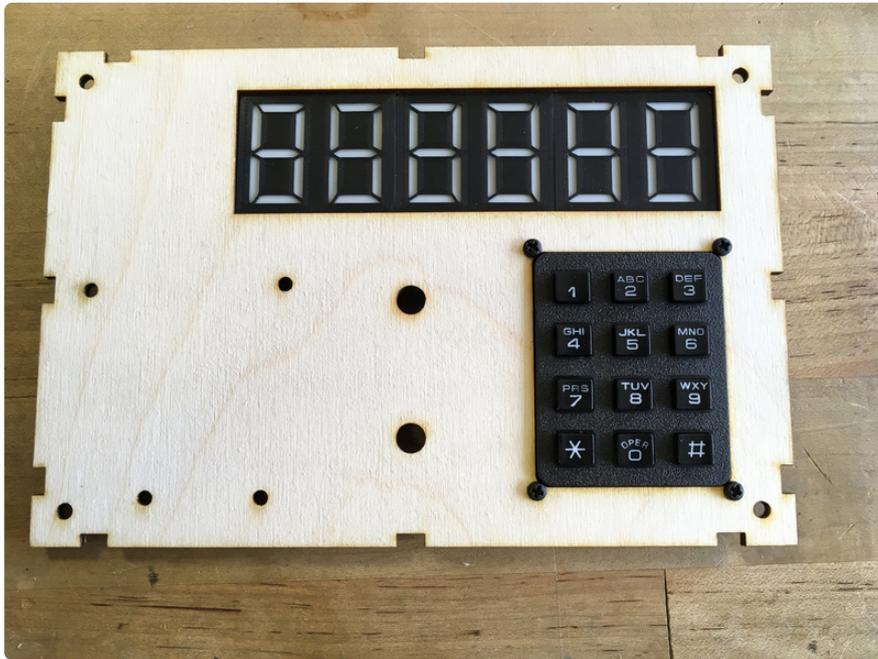
Display Modules

The NeoSegment modules will also be assembled from the inside of the front panel, however, they require a bit of preparation first.



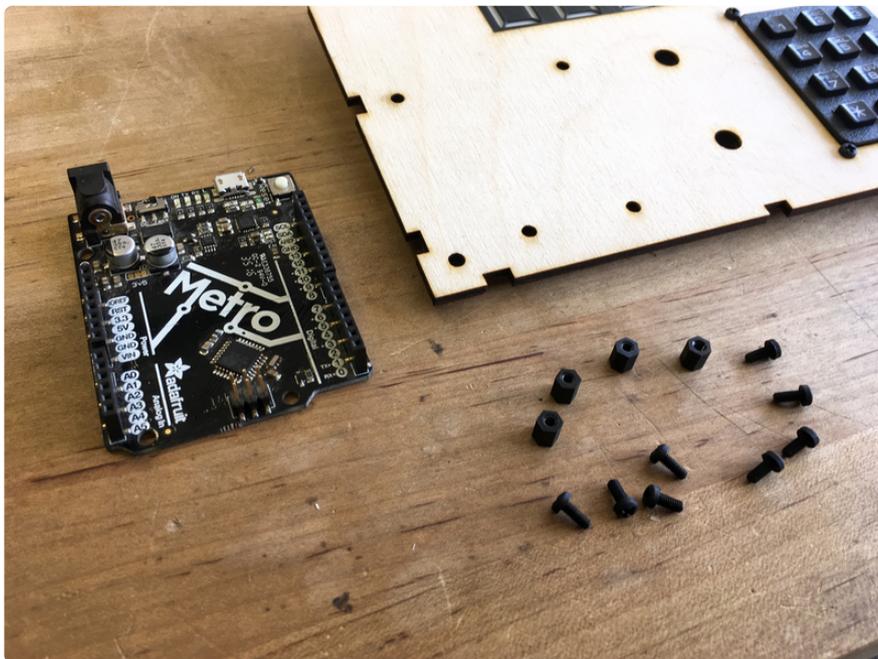


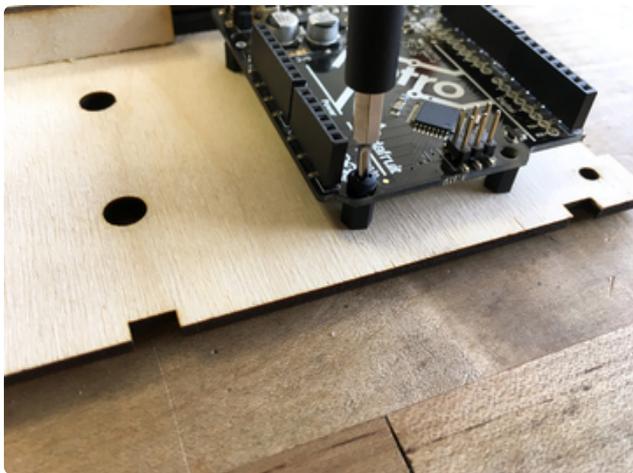
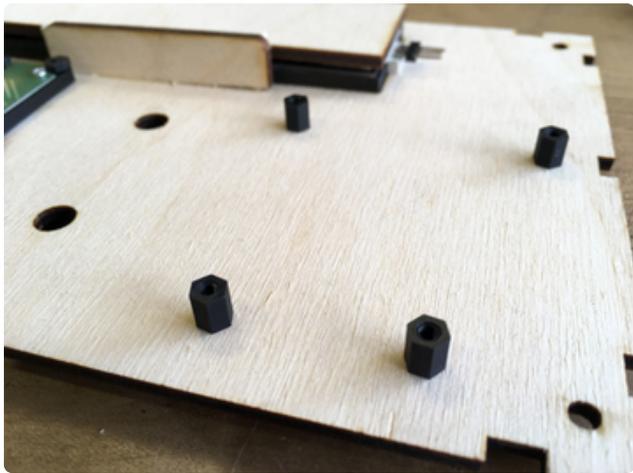
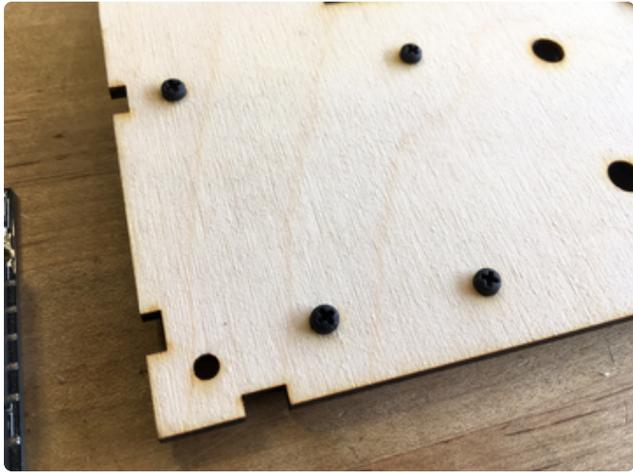
Connect the three NeoSegment modules with their built-in connectors. Then, take the large piece of material that was cut from the panel window and affix it to the back of the NeoSegments with a length of double-stick foam tape. Use the two small pieces of wood that were cut from the window as sides to box in the panel. (Note: in the final laser cutting file these scrap pieces will be cut for you from the window panel.) Affix these to the display as shown with foam tape, and use a bit of wood glue to attach to the inside of the panel.



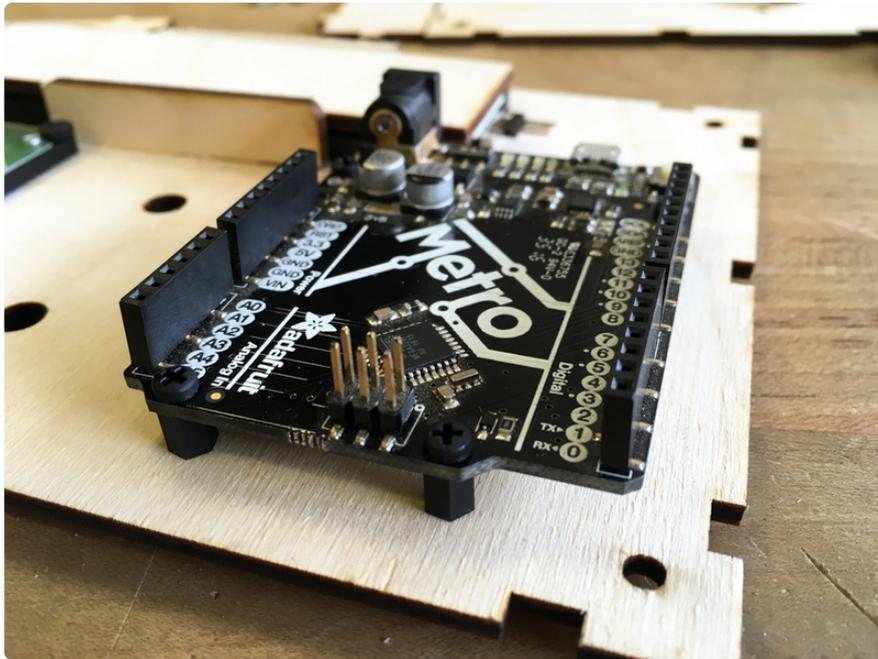
Metro Mounting

We'll use nylon hex standoffs to mount the Adafruit Metro board to the inside of the front panel.





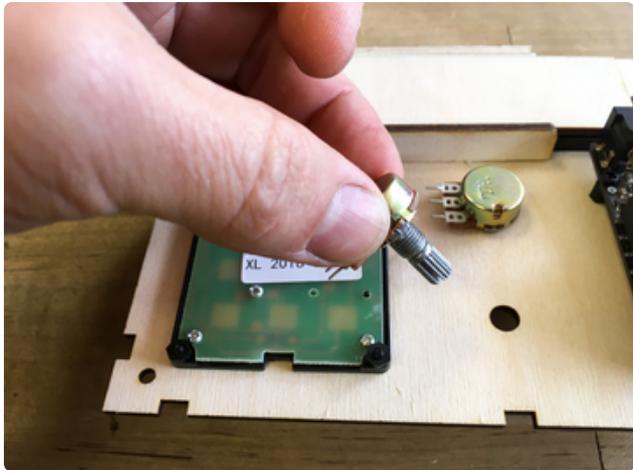
Place four medium nylon M2.5 screws into the front panel from the outside/front facing in
Screw the four stand offs to the screws from the inside of the front panel
Use the four short M2.5 screws to screw the Metro in place on the hex stand offs



Potentiometer Mounting

The two potentiometers will be mounted from the inside of the front panel and screwed into place with their included hex nuts.

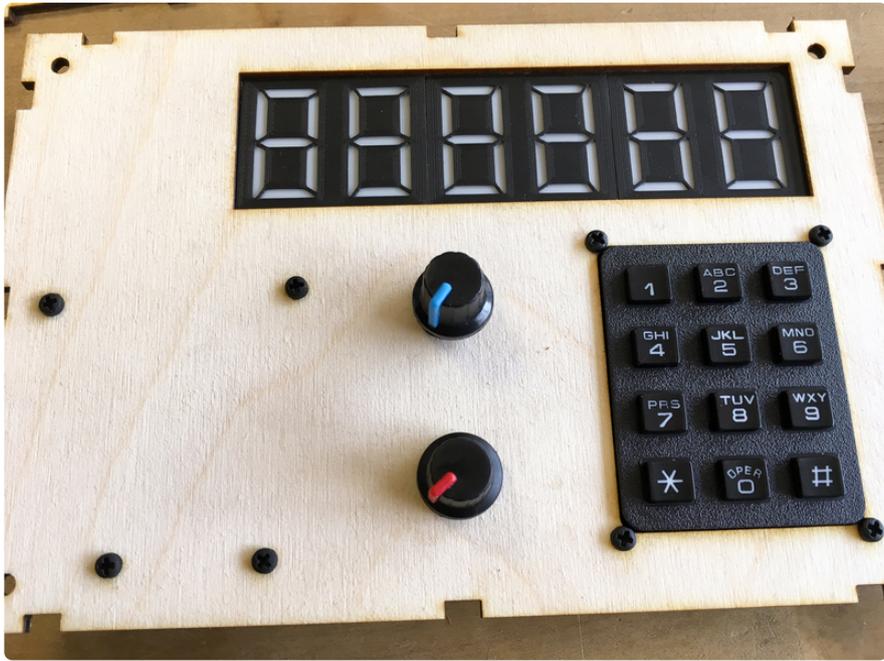




Remove the small metal keying tab from each potentiometer with a pair of pliers, so that they can be mounted flush to the panel without adding a keying registration hole. One reason this works is that the soft wood allows the part to still bite in from the compression of the nut alone. If you use acrylic, you may want to also cut a keying tab registration hole to prevent the potentiometers from spinning, since acrylic does not compress

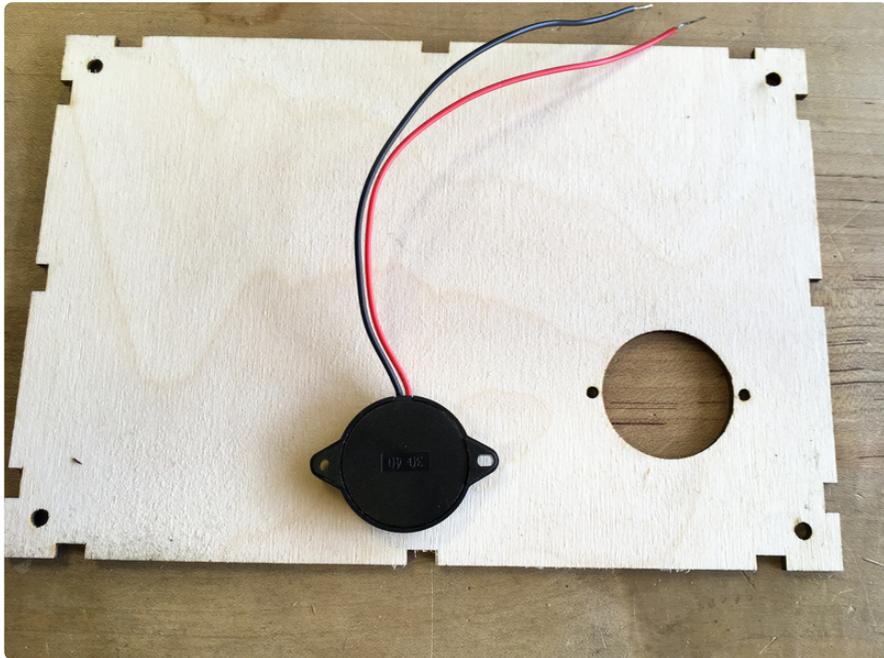
Mount the potentiometers from the back side of the front panel as shown
Screw the washers and nuts into place from the top

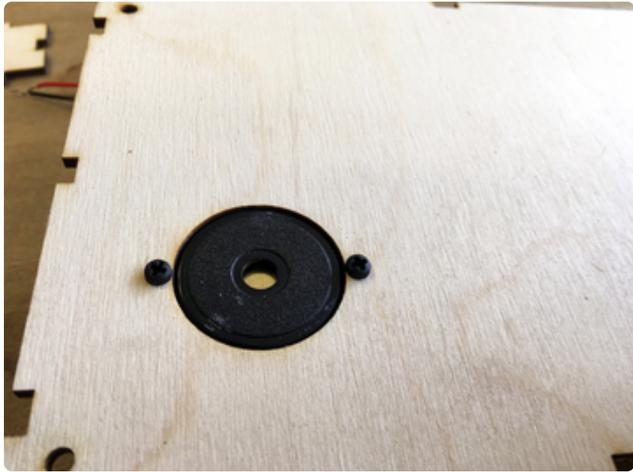
Turn the potentiometer shafts to the far left and then place the knobs on them with the indicator at the lower left position. Check that this positioning is symmetrical when the knob is turned to the far right



Piezo Buzzer

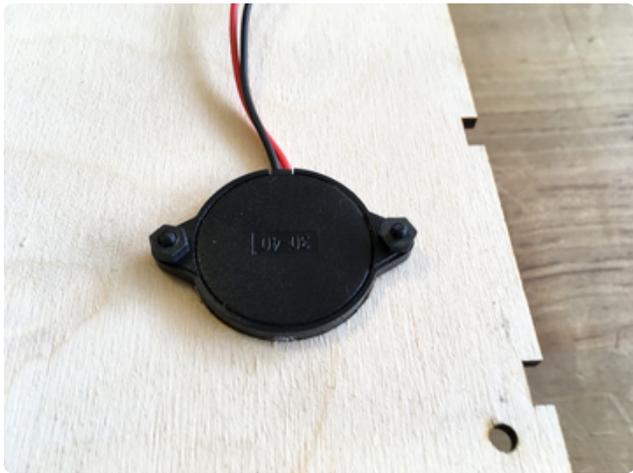
You'll attach the piezo buzzer to the back panel next.





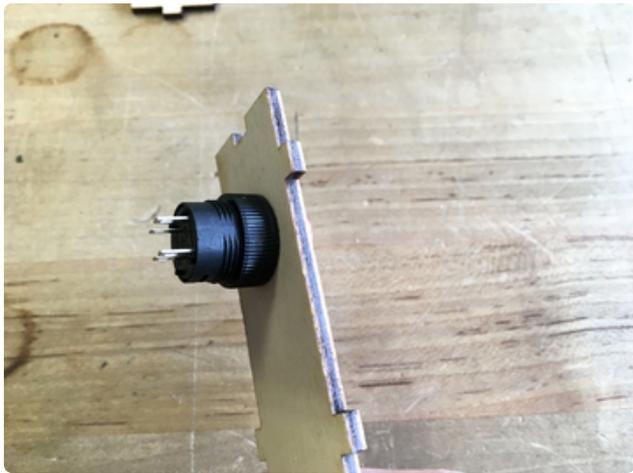
Put the piezo into the pre-cut hole from the inside of the back panel

Insert the two long M2.5 nylon screws from the outside, then screw on the nuts from the inside



Side Panels

Next, you'll connect the pushbutton and the USB jack to the two side panels.



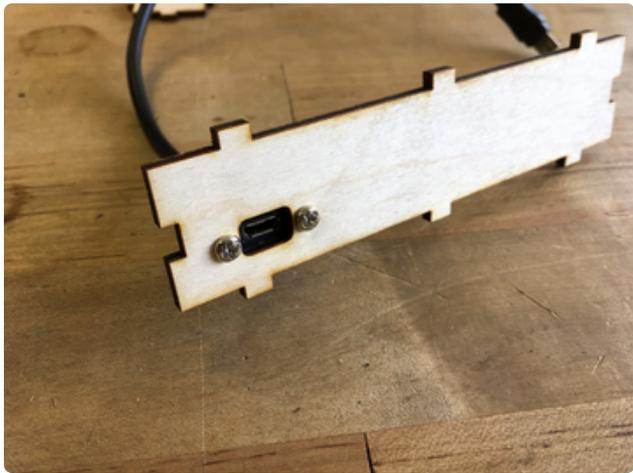
Pushbutton

Unscrew the retention collar, then fit the button into it's hole in the left side panel, then screw the collar back on.



USB Jack

Align the USB panel mount jack with the holes from the inside of the right side panel. Then, screw in the two provided screws.



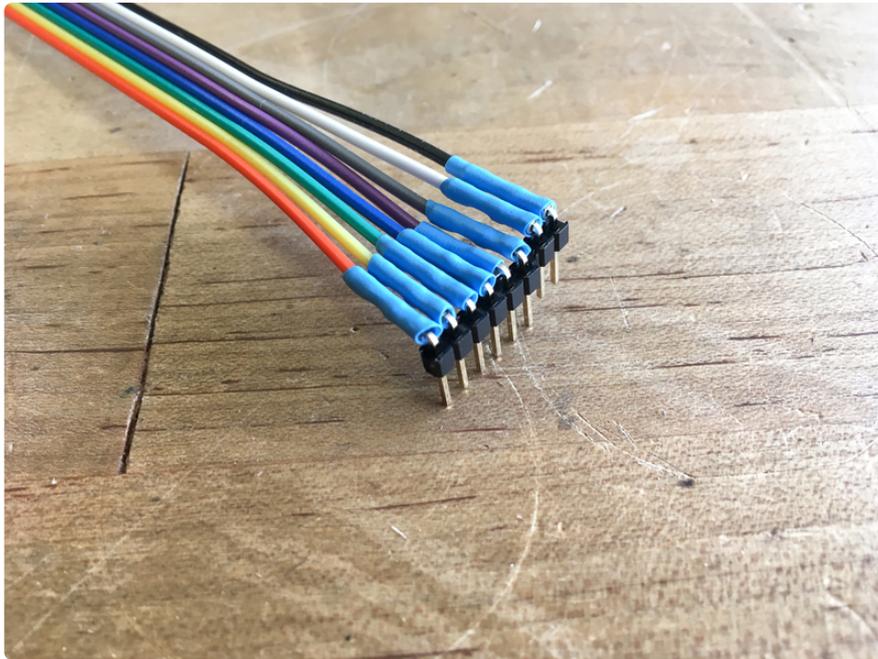
Wiring Everything Together

Now that everything is assembled, we need to wire all of the parts together. We'll start with the keypad.

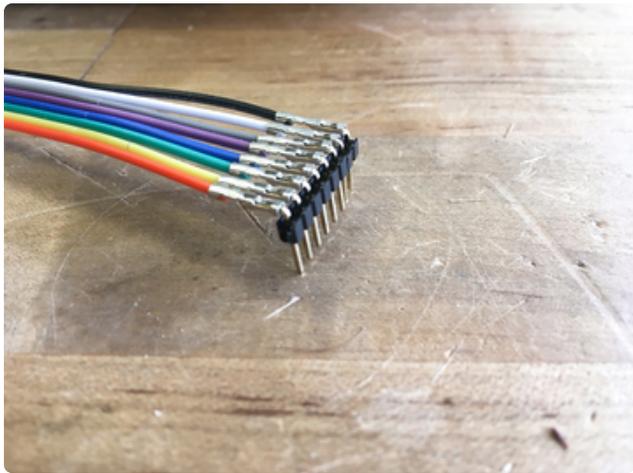
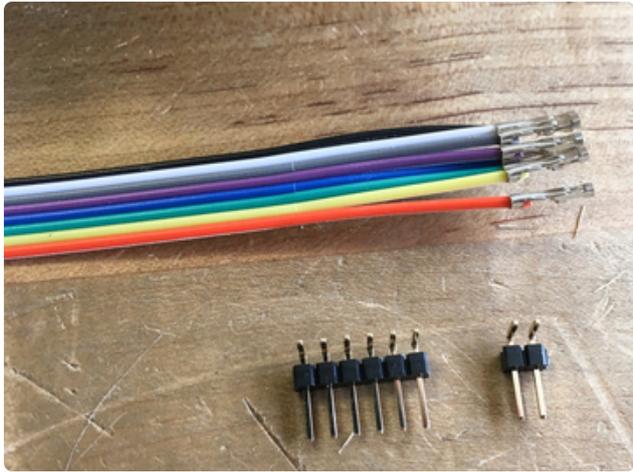
The keypad matrix uses eight wires to send data to the Metro, which we'll set up in software for pins 2-9. We'll create a custom cable to make this set of connections.

Low Profile Interconnects

Connecting the electronics is fairly straightforward, however, in order to keep a low profile in a small enclosure, we will need to make 90 degree angle connector cables.

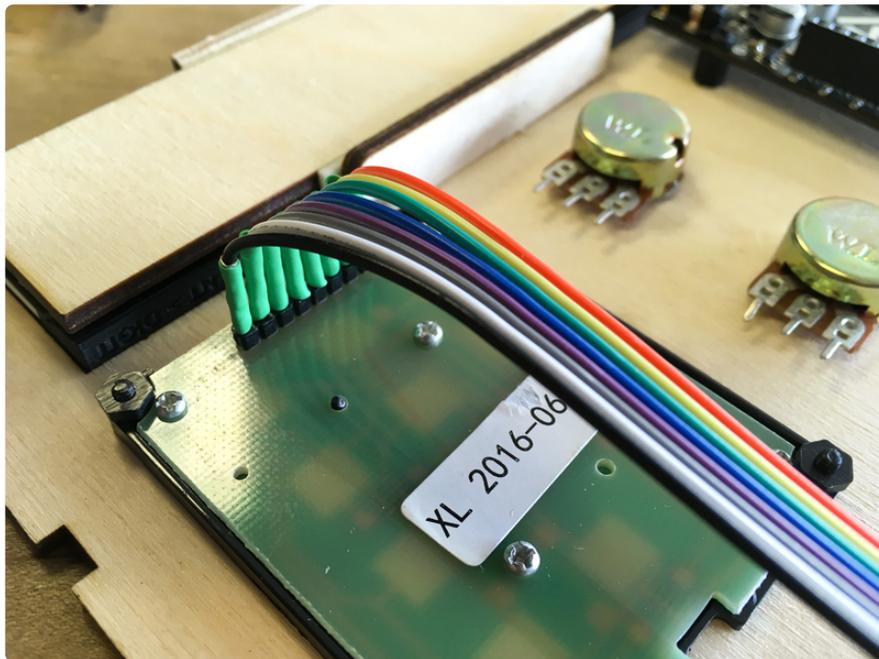


A nice, more permanent alternative is to solder the wires directly to the Metro. If you choose this path, start of with the Metro that comes with no headers.

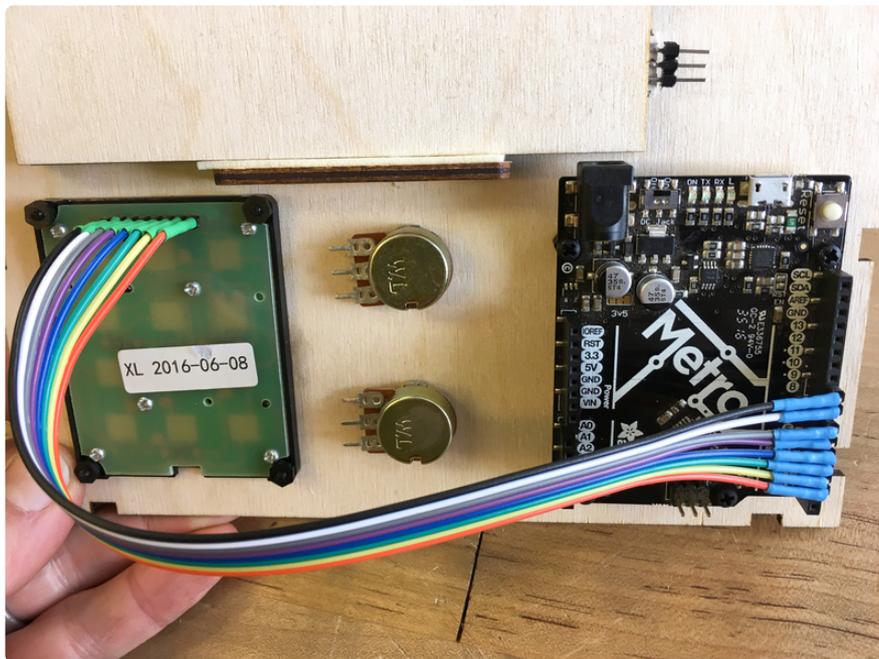


Split off an eight conductor section of the jumper wires with the shown colors
Use diagonal cutters to cut off a six- and two-conductor set of 90 degree angle jumper pins.
Fit the wires' onto the jumper pins as shown
Optionally, hit each connection with a small spot of solder to prevent them from wiggling loose later
Slide on small sections of heat-shrink tubing, and then heat them up
Voila, instant custom interconnect!

The other end of the wires can be pressed directly onto the phone keypad pins as shown, with a piece of heat shrink tubing over each one to prevent any shorts.

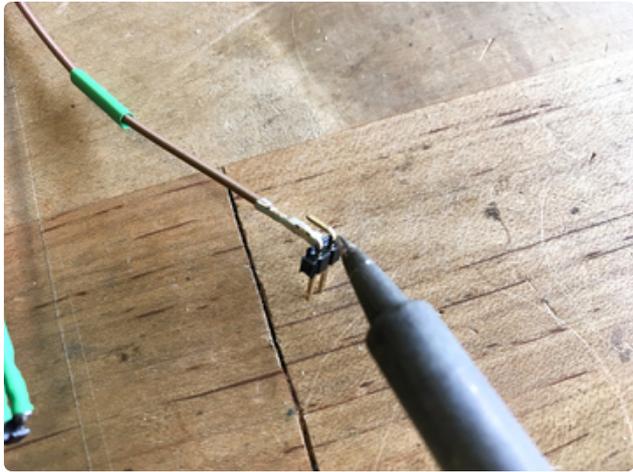


Connect the cable to the Metro as shown.

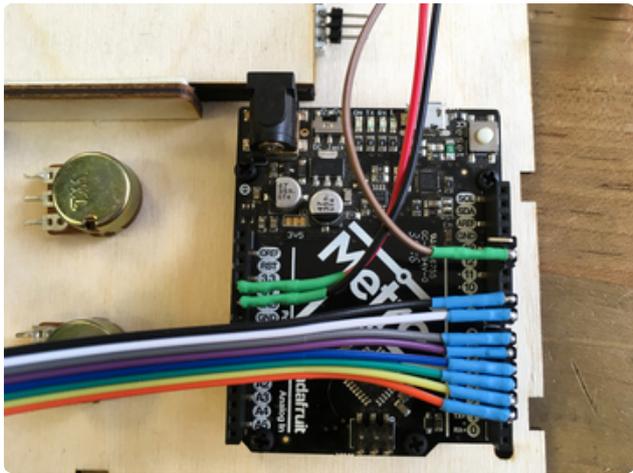


NeoSegment Wiring

Just like any NeoPixel-based project, we'll need to connect power, ground, and data lines from the Metro to the display.

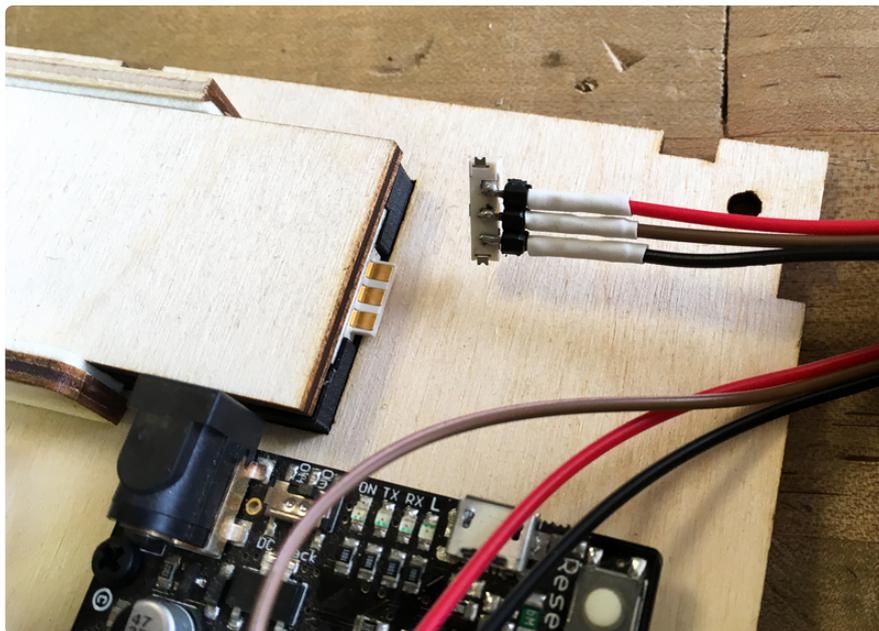


Snip off two two-pin section of 90 degree jumper pins
Strip off a brown jumper wire from the pack and fit and solder one end to the "left side" of the short lead of a 90 degree jumper pin as shown
Cover the connection with heat shrink tubing

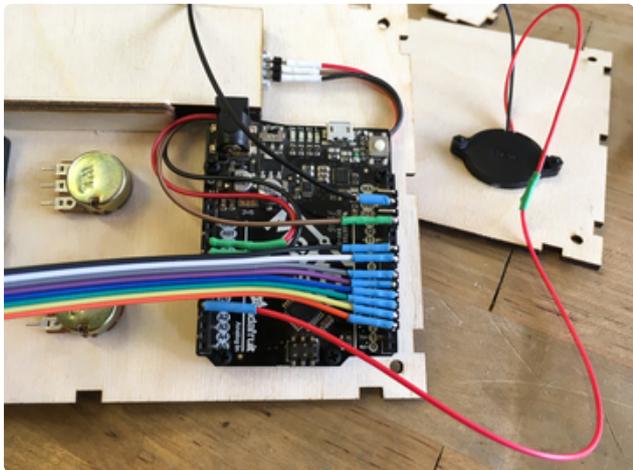
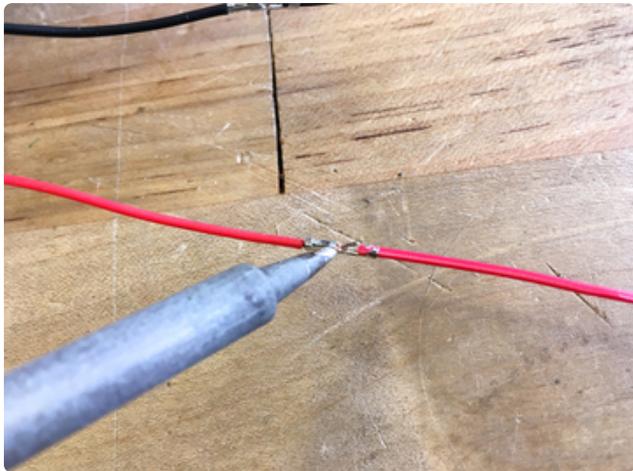
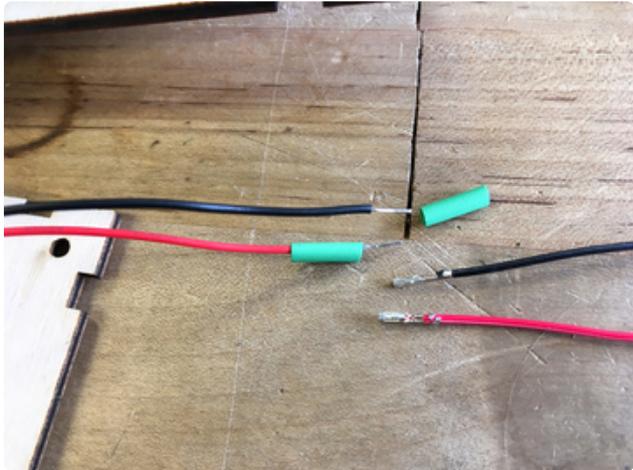


Strip off a red and black wire from the pack and fit and solder one end of each to the other two-pin 90 degree jumper, with red on the "left" and black on the "right" as shown in the second image here
Insulate the connections with heat shrink tubing
Connect the brown jumper pair to pins 12 and 13 on the Metro. Pin 12 will be the data line for the Metro, while we'll use pin 13 later for the pushbutton's internal LED
Connect the red/black jumper pair to 5V and GND on the Metro

Slide and heat some heat shrink tubing over the the other ends of these three wires. Then connect them to the NeoSegment's interconnect as shown here. Be sure to follow the image closely for wiring order.



Piezo Connection



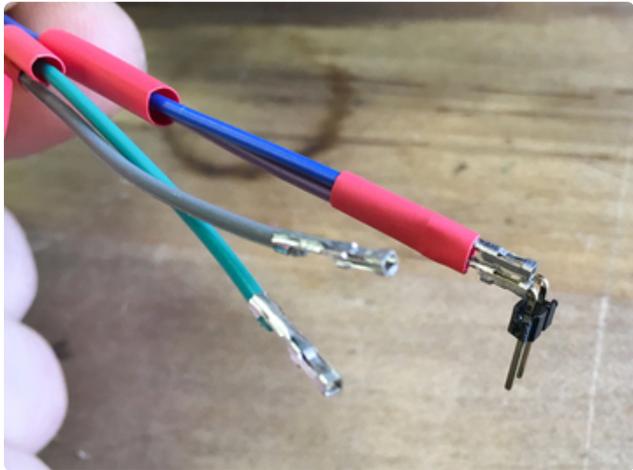
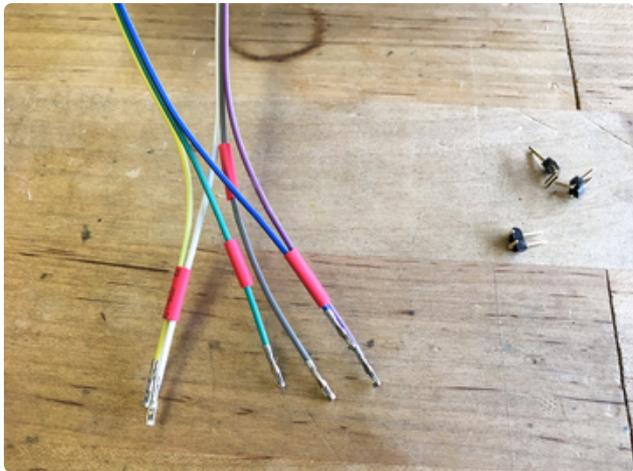
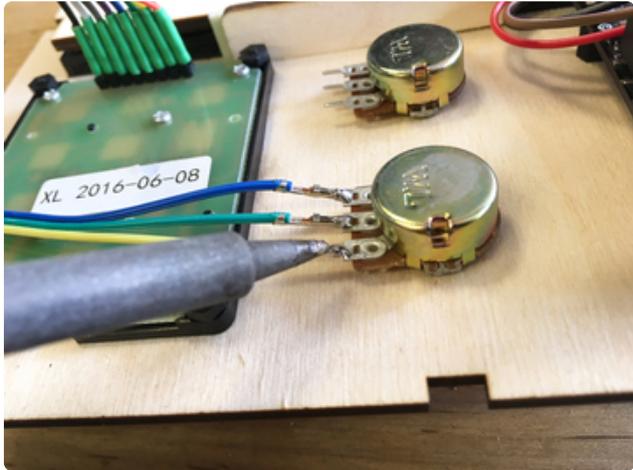
Connect the piezo's red and black wires to two lengths of jumper wires by sliding the piezo wire ends into the connectors and soldering them

Cover the connections with heat shrink tubing

Slide another piece of heat shrink tubing over each wire, then connect, solder and insulate each wire to a 90 degree header pin as we have before

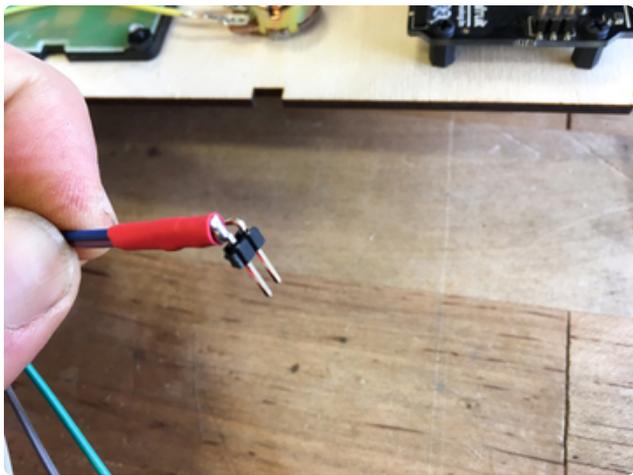
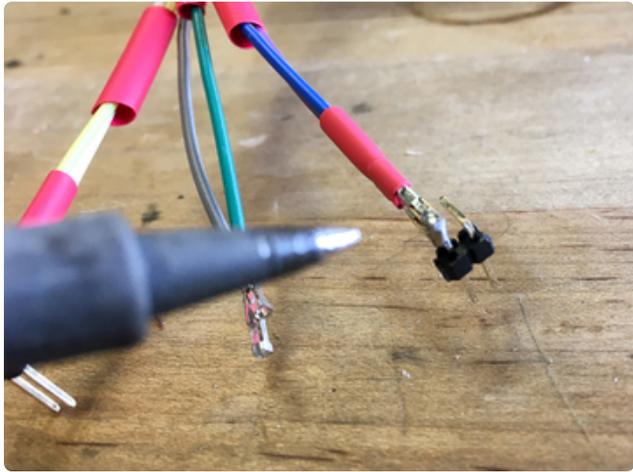
Plug the black wire into GND on the Metro, and plug the red wire into A1

Potentiometer Wiring

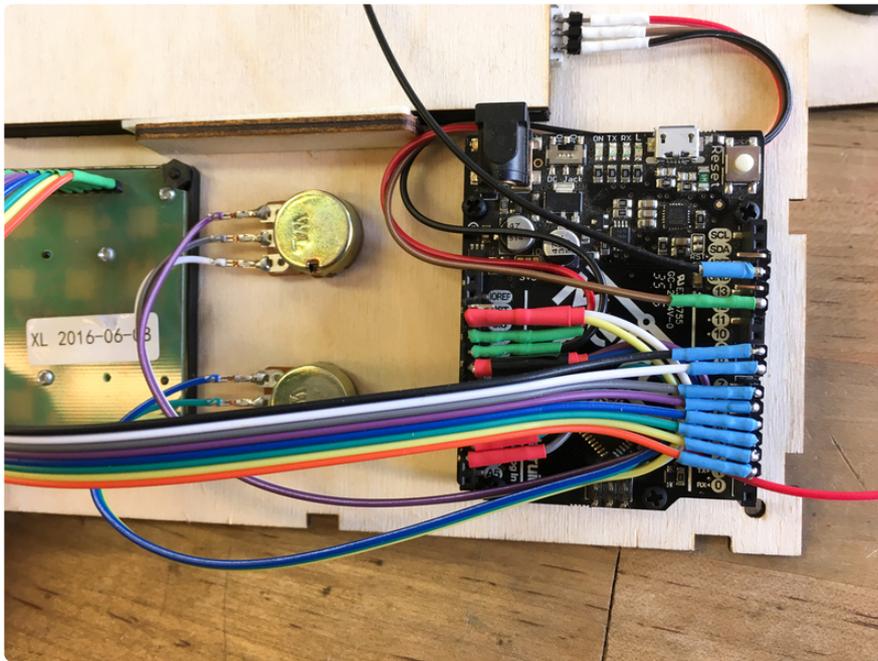
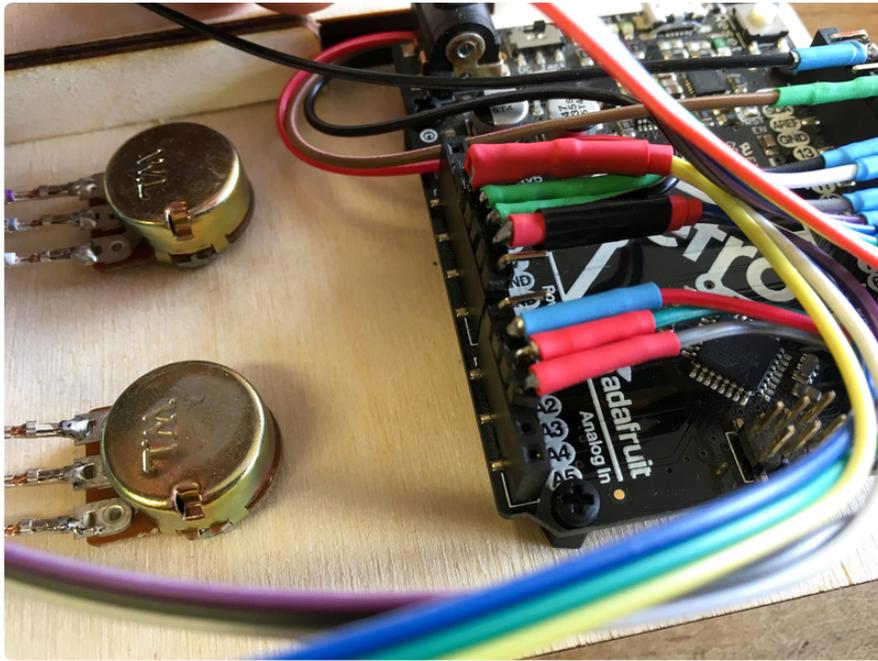


Follow the wiring diagram above and the images here for color coding and connection of the wires for the two potentiometers.

In order to efficiently utilize the available pins on the Metro, we'll share the one pin each for the voltage and ground on the two pots. You can follow these images for one way to do so.

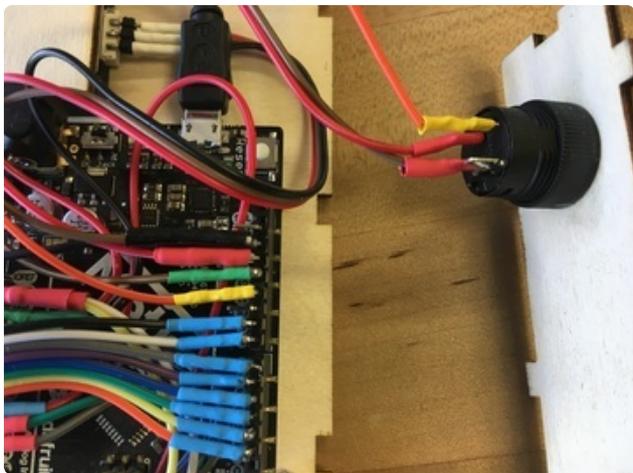
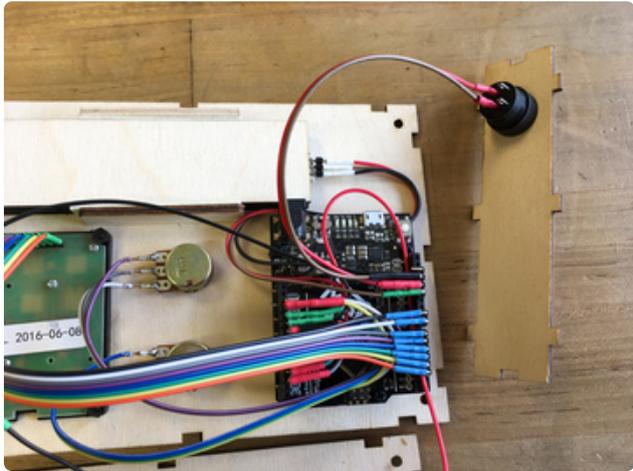
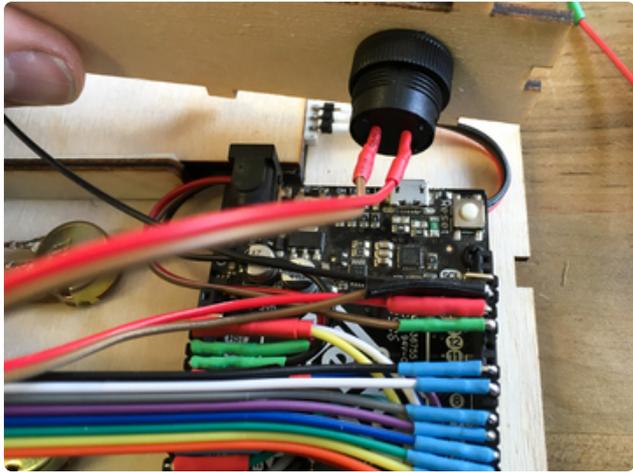


Now, plug the ground wires into **GND** on the Metro, the power wires into **3.3V**, and then the two center wiper wires into their respective pins, **A2** and **A3**.



Pushbutton Wiring

The pushbutton has four contacts -- two for the switch and two for the LED. Connect the switch contacts to GND and pin 11, and the LED contacts to GND and pin 13, as shown in the wiring diagram.

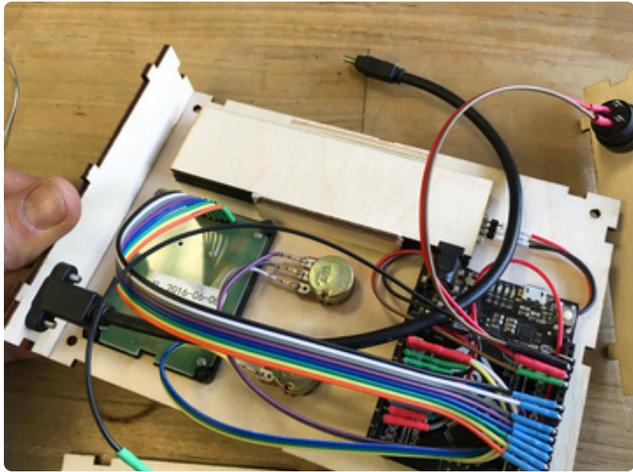


The pushbutton has four contacts -- two for the switch and two for the LED. Connect the switch contacts to GND and pin 11, and the LED contacts to **GND** and pin 13, as shown in the wiring diagram and images here.

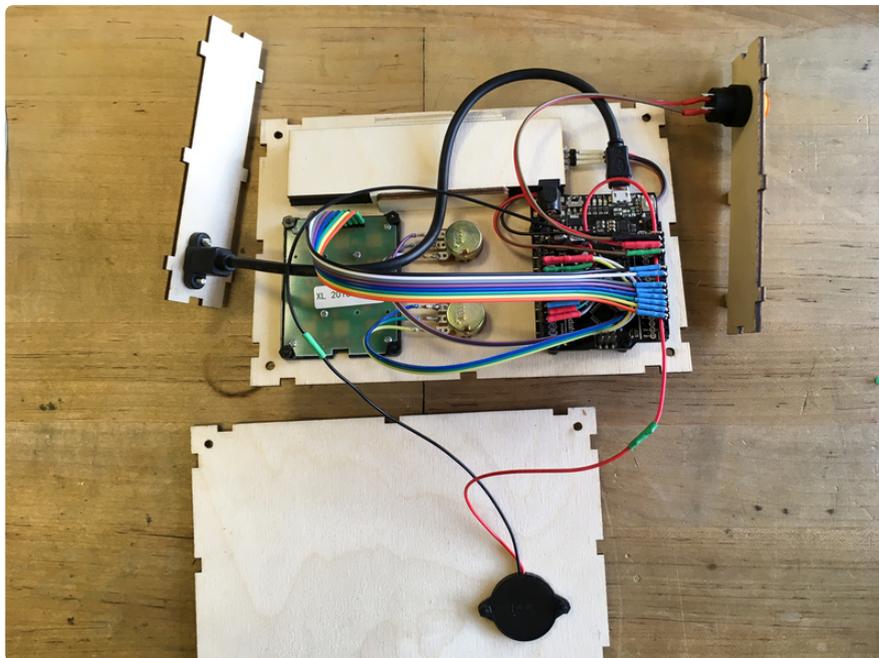
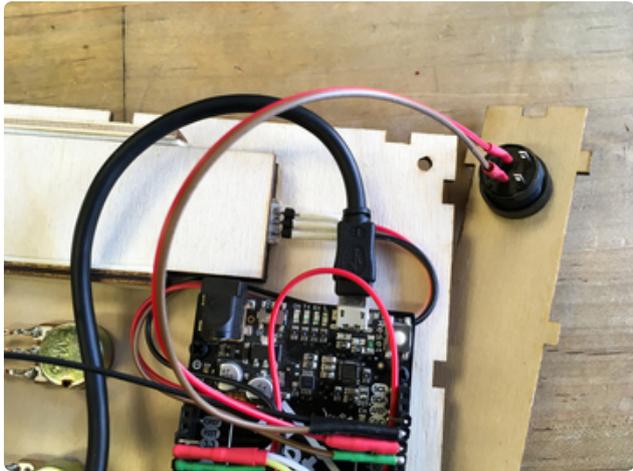
Note: the two ground contacts on the switch have been jumpered with some wire and solder so they will share one connection on the Metro.

USB Port

The final connection to wire is the USB port, which is the easiest!

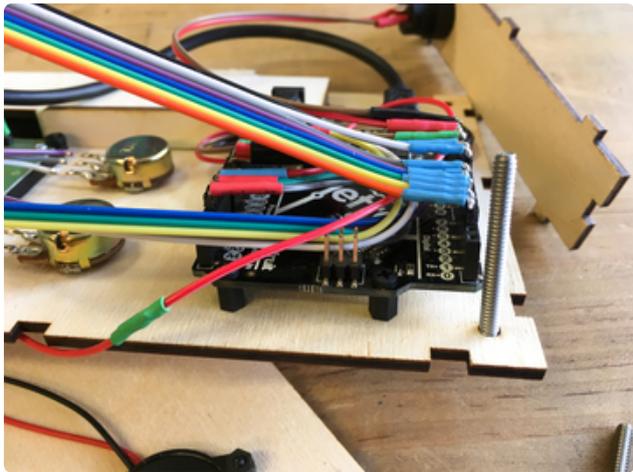
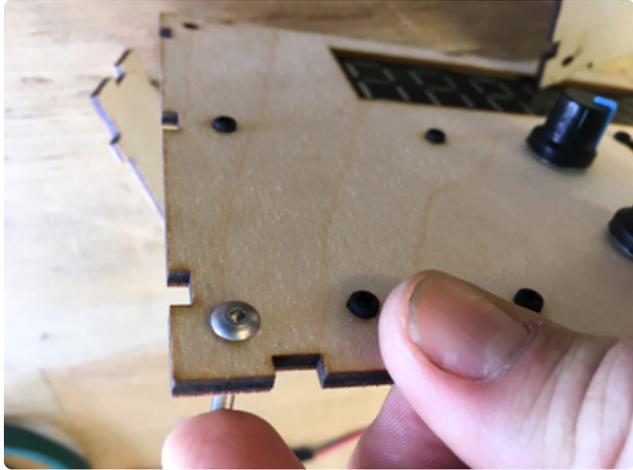


Simply plug it into the USB jack on the Metro, being careful to round the wire neatly so we can close it all up next.

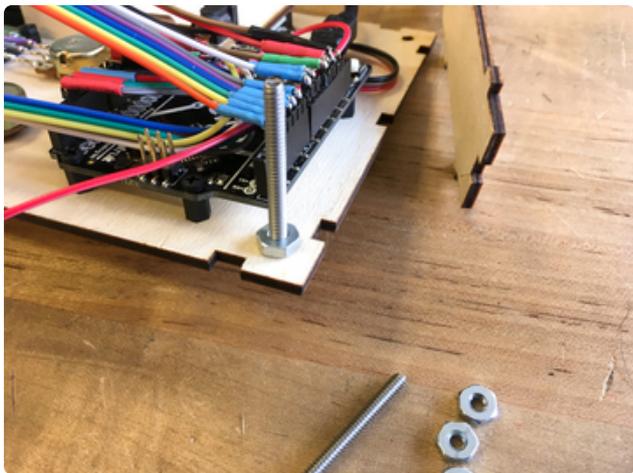


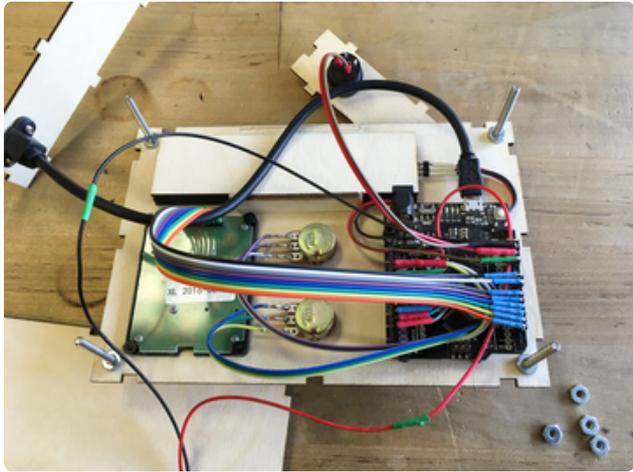
Closing the Case

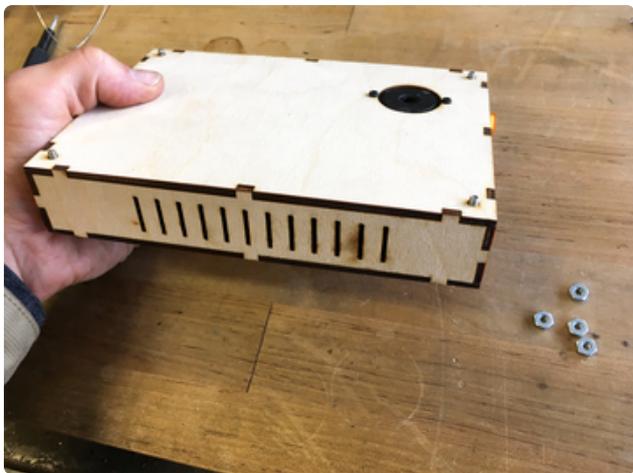
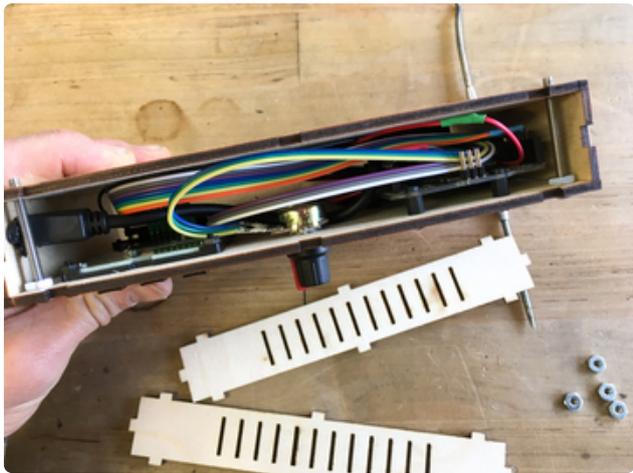
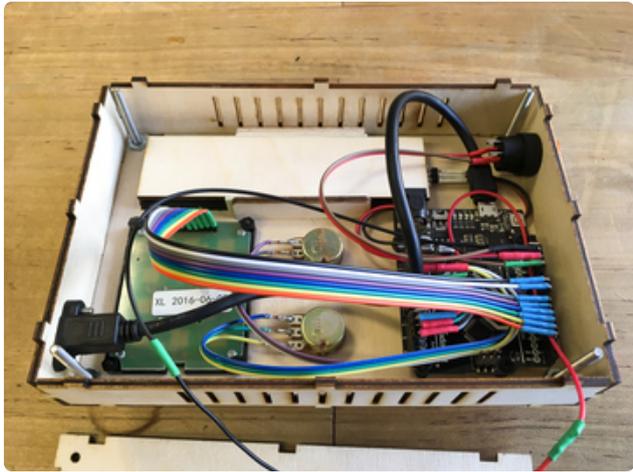
You can add glue to the side and bottom connections to the faceplate if you like, but just the tight fit and compression force of the four screws is enough to hold the case together nicely.



Push the four #6 screws through the front panel, then secure each with a nut

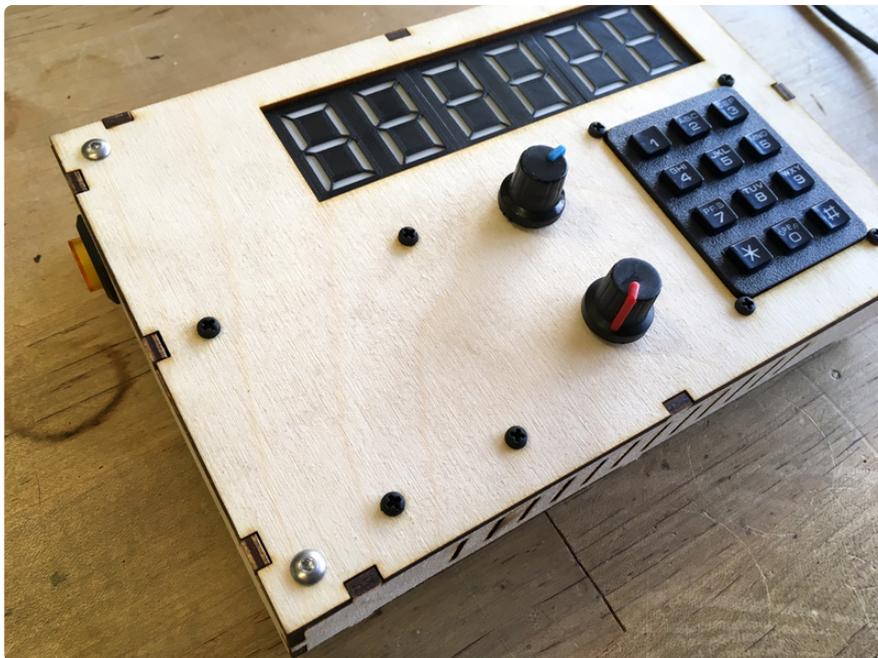
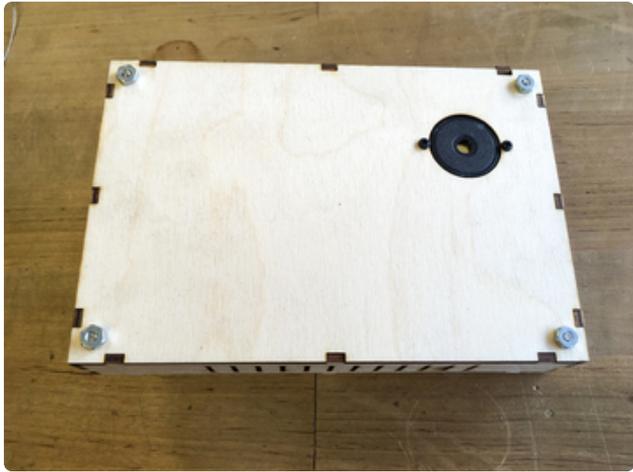






Fit the sides, top, and bottom to the front panel, being careful to keep the wiring neatly tucked inside.

You can then press the back panel into place over the four screws -- you may need to remove one side or top panel to do this -- then screw on the nuts to secure



Time to add the software!

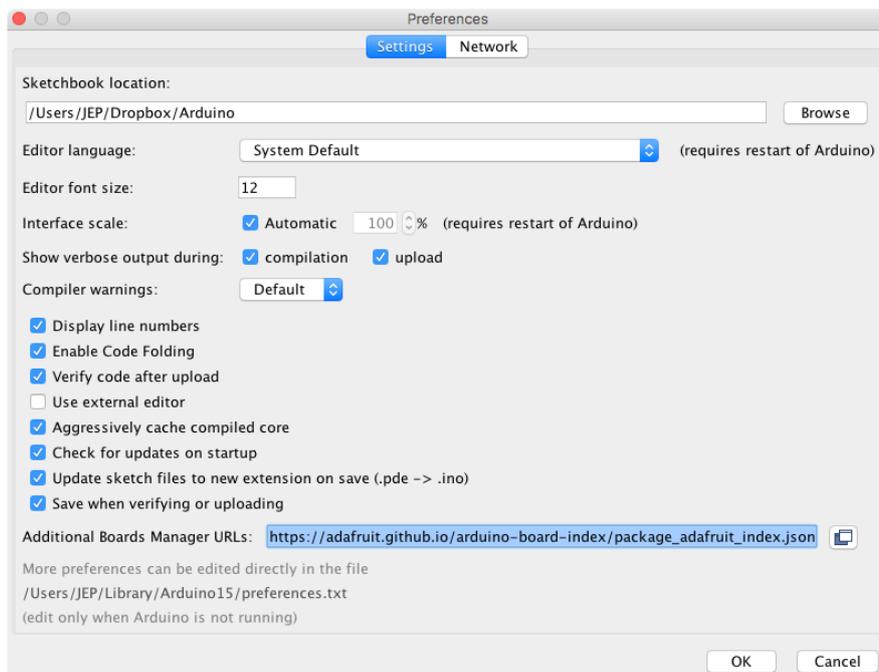
Code and Use

Plug in your NeoMatrix Mk I to your computer over USB.

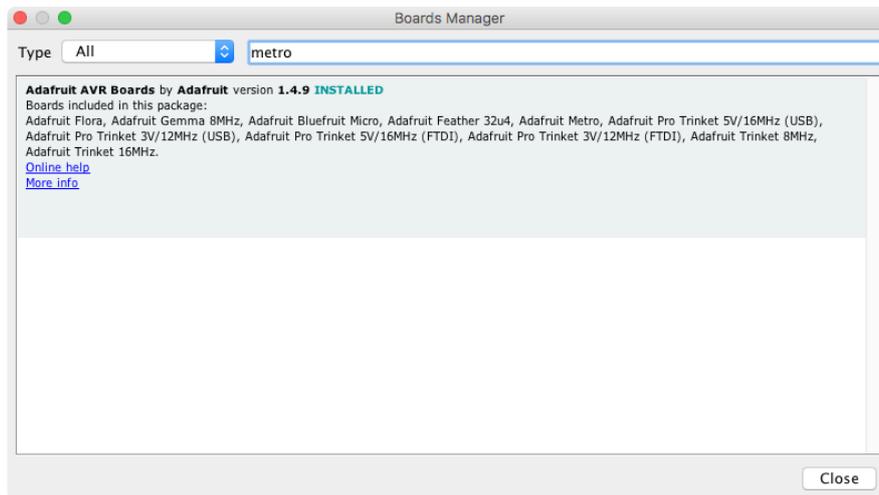
Before going any further, make sure you have a basic understanding of how to program and use an Arduino. Thankfully, we have a lot of great tutorials on how this whole thing works. [Click here to get started with Arduino \(https://adafru.it/pcg\)](https://adafru.it/pcg), and then come back to this guide to continue.

Board Manager

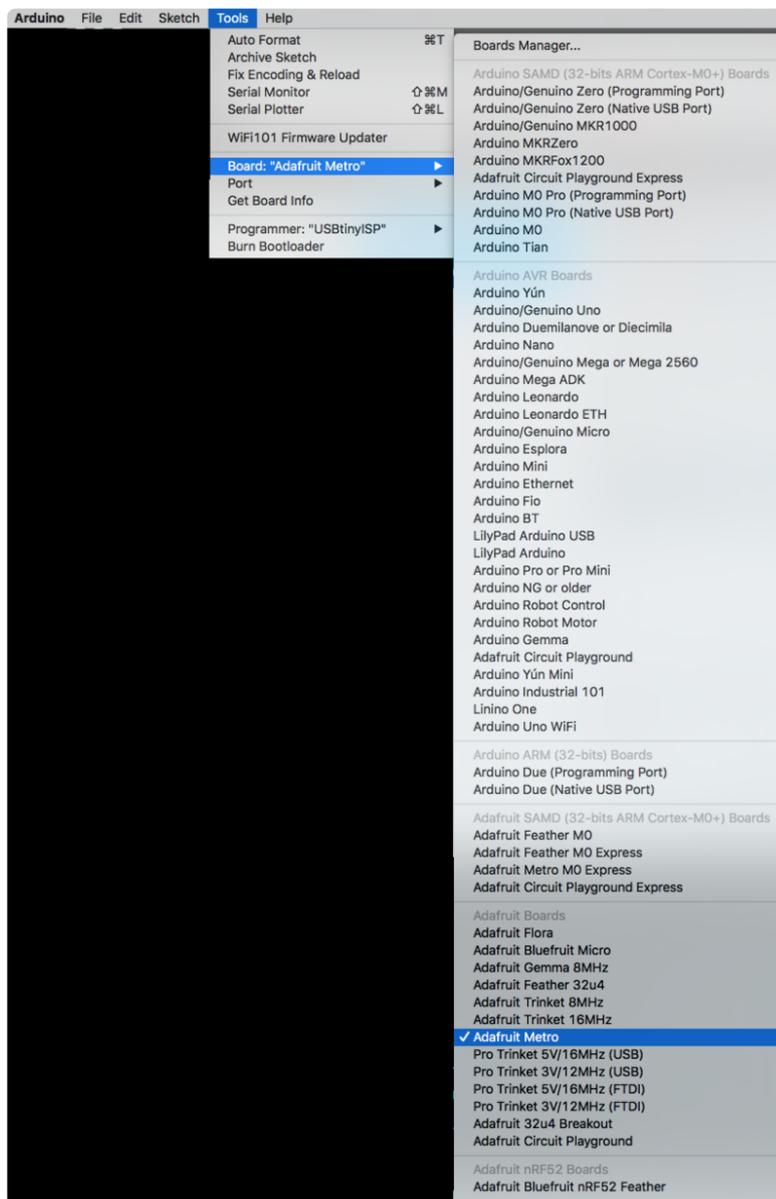
To use the Metro board in Arduino, check the Arduino IDE **Tools > Board** list and select **Adafruit Metro**. Note: if you don't see the Metro on that list, add this URL to your Arduino > Preferences > Additional Boards Manager URLs: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json



Then, go to the **Arduino > Tools > Board > Boards Manager...** and type 'metro' in the search, then install the Adafruit AVR Boards package.



Now, pick the Adafruit Metro board from the **Arduino > Tools > Board** list.




```

    0 |__| 2
      1
  */
  int buttonPin = 11; //pushbutton
  int ledPin = 13;    // select the pin for the LED
  int knobUpperPin = A3;    // input pin for a potentiometer
  int knobLowerPin = A2;    // input pin for a potentiometer
  int knobUpper = 0; // variable to store the value coming from the sensor
  int knobLower = 0; // variable to store the value coming from the sensor

  int buttonState;
  int lastButtonState = LOW;
  long lastDebounceTime = 0;
  long debounceDelay = 50;

  //initialize the neosegment object
  Neosegment neosegment(nDigits, NEOSEGPIN, LEDbrightness);
  uint16_t i, j;

  //set up the keypad
  const byte ROWS = 4; //four rows
  const byte COLS = 3; //three columns
  char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
  };
  byte rowPins[ROWS] = {5, 6, 7, 8}; //connect to the row pinouts of the keypad
  byte colPins[COLS] = {2, 3, 4}; //connect to the column pinouts of the keypad

  //initialize the keypad object
  Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

  //use to set digit cursor position
  int neoCounter = 0;

  //gamma correction table
  const uint8_t PROGMEM gamma8[] = {
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2,
    2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5,
    5, 6, 6, 6, 6, 7, 7, 7, 7, 8, 8, 8, 9, 9, 9, 10,
    10, 10, 11, 11, 11, 12, 12, 13, 13, 13, 14, 14, 15, 15, 16, 16,
    17, 17, 18, 18, 19, 19, 20, 20, 21, 21, 22, 22, 23, 24, 24, 25,
    25, 26, 27, 27, 28, 29, 29, 30, 31, 32, 32, 33, 34, 35, 35, 36,
    37, 38, 39, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 50,
    51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 67, 68,
    69, 70, 72, 73, 74, 75, 77, 78, 79, 81, 82, 83, 85, 86, 87, 89,
    90, 92, 93, 95, 96, 98, 99, 101, 102, 104, 105, 107, 109, 110, 112, 114,
    115, 117, 119, 120, 122, 124, 126, 127, 129, 131, 133, 135, 137, 138, 140, 142,
    144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 167, 169, 171, 173, 175,
    177, 180, 182, 184, 186, 189, 191, 193, 196, 198, 200, 203, 205, 208, 210, 213,
    215, 218, 220, 223, 225, 228, 231, 233, 236, 239, 241, 244, 247, 249, 252, 255 };

  void setup(){
    Serial.begin(SERIAL_BAUD);
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT_PULLUP);
    digitalWrite(ledPin, HIGH); //turn on the LED

    neosegment.begin();
    neosegment.clearAll();
    for(int i = 0; i < 6; i++){ //turn on green lines, top row, L to R
      neosegment.setSegment(i, 5, 0, 40, 0); //the '3' segment is middle dash
      tone(A1, 330, 100);
    }
  }

```

```

    delay(50);
}
neosegment.setSegment(5, 6, 0, 40, 0);
tone(A1, 330, 100);
delay(50);
neosegment.setSegment(5, 2, 0, 40, 0);
tone(A1, 330, 100);
delay(50);
neosegment.clearAll();

for(int i = 5; i > -1; i--){ //turn on green lines, bottom row, R to L
    neosegment.setSegment(i, 1, 0, 40, 0); //the '3' segment is middle dash
    tone(A1, 330, 100);
    delay(50);
}
neosegment.setSegment(0, 0, 0, 40, 0);
tone(A1, 330, 100);
delay(50);
neosegment.clearAll();

for(int i = 0; i < 6; i++){ //turn on green lines, middle row, L to R
    neosegment.setSegment(i, 3, 0, 40, 0); //the '3' segment is middle dash
    tone(A1, 440, 100);
    delay(100);
}
}

void loop(){
    //read button
    int buttonReading = digitalRead(buttonPin);
    if (buttonReading != lastButtonState) {
        lastDebounceTime = millis();
    }
    if ((millis() - lastDebounceTime) > debounceDelay){
        buttonState = buttonReading;
    }
    Serial.println(buttonState);
    if(buttonState){
        digitalWrite(ledPin, LOW);
    }
    else{
        digitalWrite(ledPin, HIGH);
    }
    lastButtonState = buttonReading;

    //read knobs
    knobUpper = analogRead(knobUpperPin); //used for hue
    knobLower = analogRead(knobLowerPin); //used for value
    //map knob range
    int knobUpperMapped = map(knobUpper, 0, 700, 360, 0);
    int knobLowerMapped = map(knobLower, 0, 700, 700, 0);
    //constrain knob range
    knobUpperMapped = constrain(knobUpperMapped, 0, 360);
    knobLowerMapped = constrain(knobLowerMapped, 0, 700);
    //map to hue range
    //int knobHue = map(knobUpperMapped, 0, 255, 0, 360);
    //knobHue = constrain(knobHue, 0, 360);
    float knobHue = ((float)knobUpperMapped);
    //map to value range
    float knobValue = ((float)knobLowerMapped / 700);

    float r, g, b;
    float h = knobHue;
    float s = 1.0;
    float v = knobValue;

    HSVtoRGB(&r, &g, &b, h, s, v); //convert HSV to RGB
    uint8_t r_byte = (int)(r * 255);
    uint8_t g_byte = (int)(g * 255);

```

```

uint8_t b_byte = (int)(b * 255);

//apply gamma correction table values
r_byte = pgm_read_byte(&gamma8[r_byte]);
g_byte = pgm_read_byte(&gamma8[g_byte]);
b_byte = pgm_read_byte(&gamma8[b_byte]);

char key = keypad.getKey();

if (key != NO_KEY){ // a key has been pressed
  Serial.print("key: ");
  Serial.println(key);
  int neoKey = key - '0'; //terminate w zero so don't get ASCII code

  //enter with '#'
  if(key == '#'){
    neosegment.clearAll();
    delay(35);
    neosegment.setSegment(5, 2, 40, 40, 40);
    neosegment.setSegment(5, 6, 40, 40, 40);
    delay(35);
    tone(A1, 660, 100);
    for(int i = 5; i > -1; i--){ //from right to left
      neosegment.setSegment(i, 1, 40, 40, 40);
      neosegment.setSegment(i, 5, 40, 40, 40);
      delay(35);
      tone(A1, 660, 100);
    }
    neosegment.setSegment(0, 0, 40, 40, 40);
    neosegment.setSegment(0, 4, 40, 40, 40);
    tone(A1, 660, 100);
    neoCounter = 0;

    //letters that appear in upper: A, E, F, H, I, J, L, P, S, U
    //letters that appear in lower: b, c, d, g, n, o, q, r, t
    delay(1000);
    neosegment.clearAll();
    delay(1000);
    neosegment.setDigit(5, 'a', 23, 0, 12);
    tone(A1, 220, 100);
    delay(500);
    neosegment.setDigit(4, 'a', 23, 0, 12);
    neosegment.setDigit(5, 'c', 23, 0, 12);
    tone(A1, 220, 100);
    delay(500);
    neosegment.setDigit(3, 'a', 23, 0, 12);
    neosegment.setDigit(4, 'c', 23, 0, 12);
    neosegment.setDigit(5, 'o', 23, 0, 12);
    tone(A1, 220, 100);
    delay(500);
    neosegment.setDigit(2, 'a', 23, 0, 12);
    neosegment.setDigit(3, 'c', 23, 0, 12);
    neosegment.setDigit(4, 'o', 23, 0, 12);
    neosegment.setDigit(5, 'r', 23, 0, 12);
    tone(A1, 220, 100);
    delay(500);
    neosegment.setDigit(1, 'a', 23, 0, 12);
    neosegment.setDigit(2, 'c', 23, 0, 12);
    neosegment.setDigit(3, 'o', 23, 0, 12);
    neosegment.setDigit(4, 'r', 23, 0, 12);
    neosegment.setDigit(5, 'n', 23, 0, 12);
    tone(A1, 220, 100);
    delay(2000);

    for(int x=0; x<50; x++){
      neosegment.setDigit(2, 'c', 0, x, 0);
      delay(15);
    }
    for(int x=0; x<50; x++){

```

```

        neosegment.setDigit(4, 'r', x, x, 0);
        delay(15);
    }
    for(int x=0; x<50; x++){
        neosegment.setDigit(3, 'o', 0, x, x);
        delay(15);
    }
    for(int x=0; x<50; x++){
        neosegment.setDigit(5, 'n', x, 0, 0);
        delay(15);
    }
    for(int x=0; x<50; x++){
        neosegment.setDigit(1, 'A', x/2, x/2, x/2);
        delay(15);
    }
    delay(500);
    neosegment.setDigit(1, 'a', 23, 0, 12);
    neosegment.setDigit(2, 'c', 23, 0, 12);
    neosegment.setDigit(3, 'o', 23, 0, 12);
    neosegment.setDigit(4, 'r', 23, 0, 12);
    neosegment.setDigit(5, 'n', 23, 0, 12);
    tone(A1, 220, 100);
    delay(1700);
    neosegment.clearAll();
    delay(500);
    for(int i = 0; i < 6; i++){ //turn on green lines, middle row, L to R
        neosegment.setSegment(i, 3, 0, 40, 0); //the '3' segment is middle dash
        //tone(A1, 440, 100);
        delay(100);
    }
}

//special function with '*'
else if(key == '*'){ //show HUE value
    neosegment.clearAll();

    neosegment.setDigit(0, 'h', 0, 30, 0);
    neosegment.setDigit(1, 'u', 0, 30, 0);
    neosegment.setDigit(2, 'e', 0, 30, 0);

    int ones = (((int)knobHue) % 10);
    int tens = (((int)knobHue)/10) % 10);
    int hundreds = (((int)knobHue)/100) % 10);

    neosegment.setDigit(5, ones, r_byte, g_byte, b_byte);
    neosegment.setDigit(4, tens, r_byte, g_byte, b_byte);
    neosegment.setDigit(3, hundreds, r_byte, g_byte, b_byte);
    Serial.print("red byte: ");
    Serial.println(r_byte);

    tone(A1, ((knobHue+31)), 200);
    neoCounter = 0;
}

//number keys
else{ // display numbers
    if(neoCounter%6 == 0){ //clear when screen gets full
        neosegment.clearAll();
    }
    int neoPosition = (neoCounter % 6); //use modulo operation to loop
    //through the positions
    int BLUE = knobUpperMapped;
    neosegment.setDigit(neoPosition, neoKey, r_byte, g_byte, b_byte);
    neoCounter++;
    tone(A1, ((knobUpperMapped * neoKey) + 31), 170); //+31 deals with
    //the 0 key
}
}
}
}

```

```

//function to convert Hue, Saturation, Value to Red, Green, Blue
void HSVtoRGB(float *r, float *g, float *b, float h, float s, float v){
//HSV is in HUE: degrees, SATURATION: 0 to 1.0, VALUE: 0 to 1.0
  int i;
  float f, p, q, t;
  if( s == 0 ) {
    // achromatic (grey)
    *r = *g = *b = v;
    return;
  }
  h /= 60;      // sector 0 to 5
  i = floor( h );
  f = h - i;    // factorial part of h
  p = v * ( 1 - s );
  q = v * ( 1 - s * f );
  t = v * ( 1 - s * ( 1 - f ) );
  switch( i ) {
    case 0:
      *r = v;
      *g = t;
      *b = p;
      break;
    case 1:
      *r = q;
      *g = v;
      *b = p;
      break;
    case 2:
      *r = p;
      *g = v;
      *b = t;
      break;
    case 3:
      *r = p;
      *g = q;
      *b = v;
      break;
    case 4:
      *r = t;
      *g = p;
      *b = v;
      break;
    default:    // case 5:
      *r = v;
      *g = p;
      *b = q;
      break;
  }
}

```

You can see from the code that there are a few different functions, depending on which buttons you press.

- The number keys will display numbers and play beeps, which differ in pitch depending on the digit. Also, turn the upper knob to change which hue will be used the next time you press a number. The bottom knob adjusts value/brightness
- Press the '*' key to see the hue value expressed as a number from 0-360 on a color wheel

- Press the '#' key to "enter" your code. This is just a demo mode, so all answers are correct and lead to a short reveal of a new code word. Look at how this was written to adjust for your own needs

