I started with the raspberry pi and breadboard plugging the pi in to the either side of the breadboard.

Referencing nerd cave, I learned how to code Led's with the pi. Using the pins 13,12,16 of the raspberry pi then i plugged the resistors to pin 24 25 30 31 36 37 then I took the cathode of the Led's connected it to the side of the resistor then I connected the anode to the ground side of the resistor and i grounded out the whole breadboard to the raspberry pi Pico. Then I started coding Led's. The code for this action makes all the lights light up in a certain pattern. I did this by putting two Leds for each light variable. Then in my main while loop i assigned the value 1 to activate that variable then I make that variable stay on using utime.sleep(time). Then i use the variable .value() with zero in the parenthesis to turn off the light.

Using the motor driver insert (citation) we connected it to pin 14 15 on the data in port so that our code could tell the Pico what the motor driver will do. I was unable to get OUT 3 and OUT 4 to work with 2 different motors, so I connected both motors to OUT 1 and OUT 2. Only after coding variables naruto9 and sasuke9 did I notice the motors spin in opposite directions which I liked. Take in mind, my original battery was only 3.7V and the motor driver needs 5V. Inside my main while loop I used another while loop and the same “.value()” for making the motors spin in both directions and stop.

For the sound action, I used a one-inch speaker and attached the it to pin 10 into bread board line c-62 . Then i connected the cathode of the speaker on the same line a-62 of the breadboard. Completing the circuit by connecting the anode to the ground. the code for the sound circuit needed a function with 2 inputs(frequency and time). In this function I define two more variables (period and cycles). The period is 1 divided by the frequency and the cycles is time times the frequency. I used a for loop to go through a pattern of turning the speaker on and off at the input frequency for the time.

The button was tricky because it required 2 while loops and 2 if statements. I had to set the button variable to false meaning it would not start the show until i pushed it making the variable equal to true. This also meant i needed a variable that would show whether the display was on or not. In the first if statement I made it say if this is equal to 1 while making sure the button had not been pressed. Then the new value of the button variable would be true. Then the second if making sure the display is already not running then the display variable has a new value of true. Then my main while loop starts.