I am demonstrating a fully working system

My assigned Clock Speed = 30 Mhz

My onboard LED is PN0

Questions to Answer during mid project demo:

1. How do you specify the angle the motor has rotated through?

So i have a loop which goes from 0 to 512 in my code and whenever the given condition is met, we know that the motor has rotated 45 degrees and then the LED blinks.

1. How do you modify the speed of the stepper motor?

By changing the value of the variable “delay” which is passed to the function “rotate”.

1. How do you change the direction of the stepper motor?

By reversing the port e data register values, just like we did for lab 6.

1. Explain the concept of a state machine. How does it relate to this design?

(what are the states? What are the inputs? How do we transition from one state to another?)

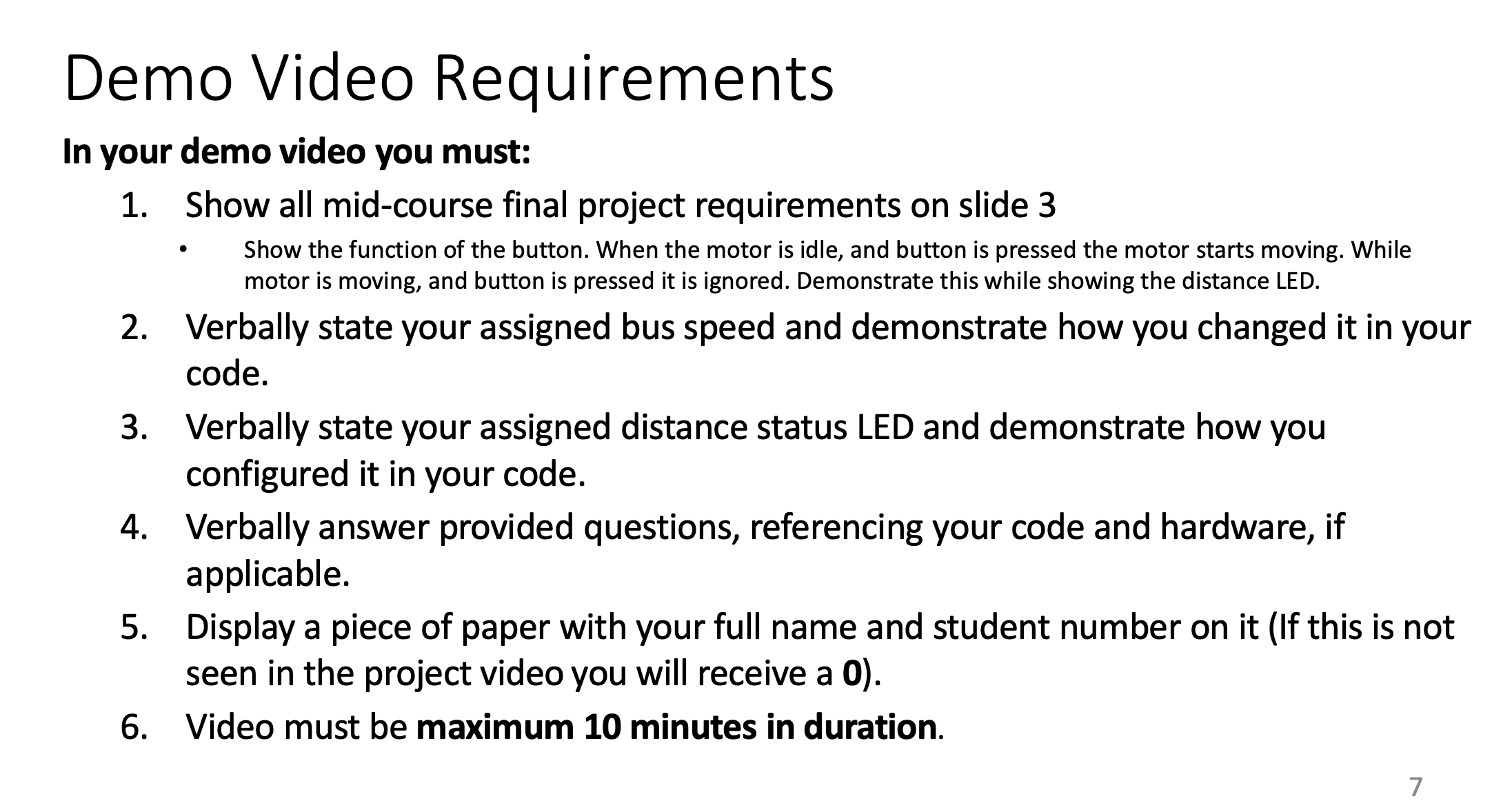
A state machine is a behavioural model for any computational system. It consists of a finite number of states and is therefore also called finite-state machine.

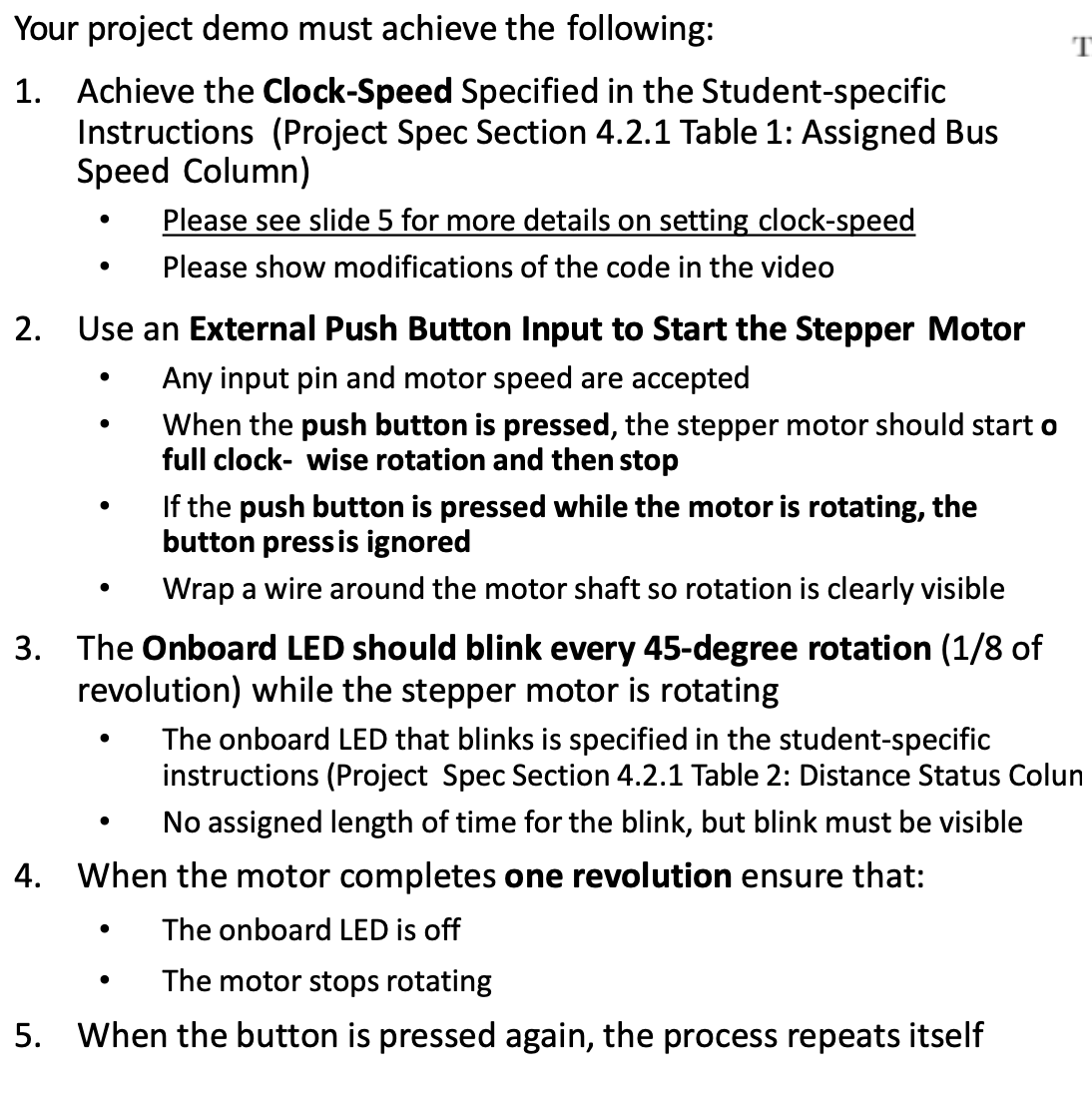
For our design, we essentially have two states, we can maybe call them ON and OFF.

We have a single bit input coming from the push button. We transition from OFF to ON by the press of this button and automatically go from ON to OFF after one complete rotation.

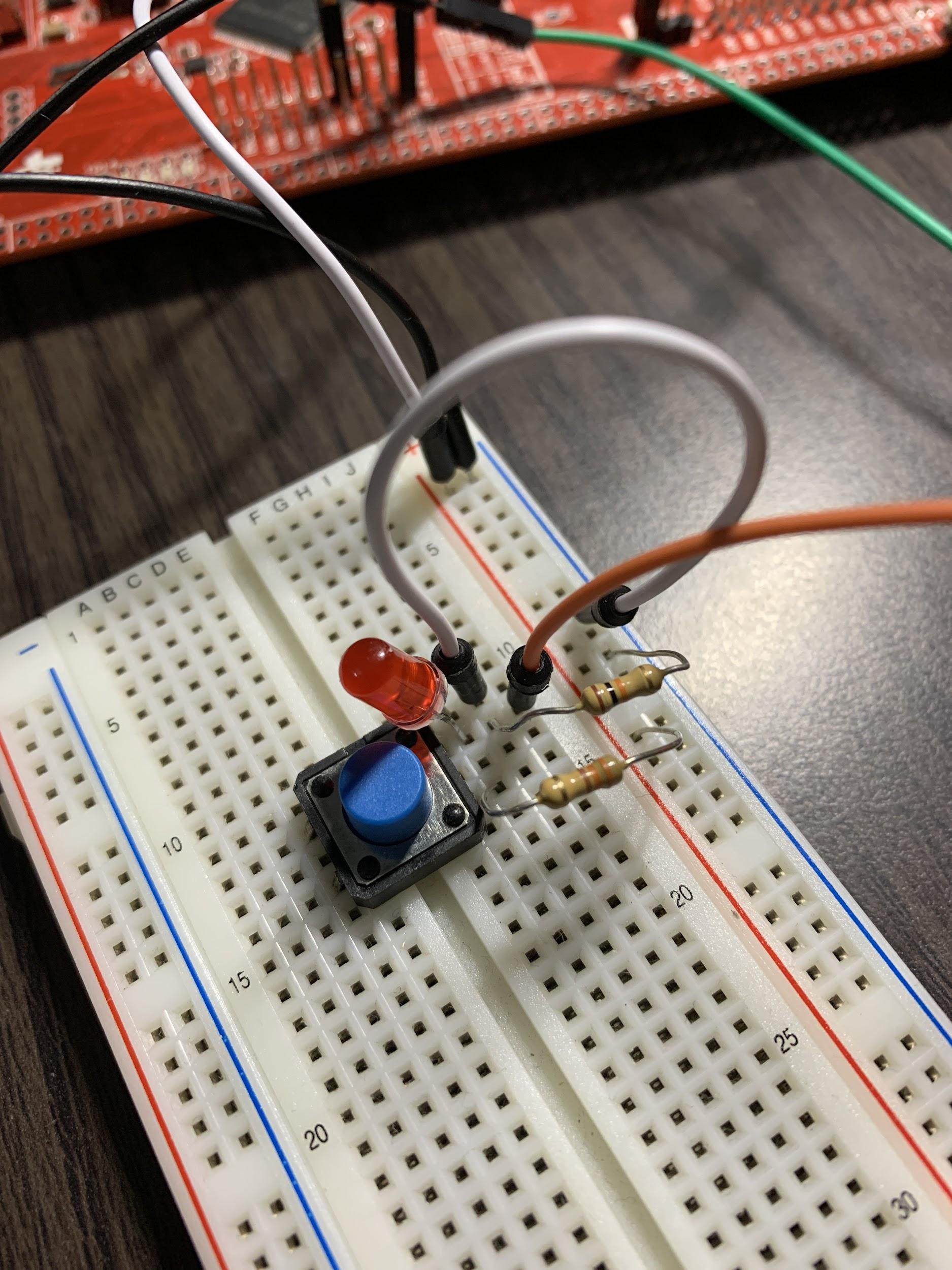
So that is all I wanted to demo today. Thank you for your time, if i missed anything or if i can fix something to make it better, please let me know in the feedback.

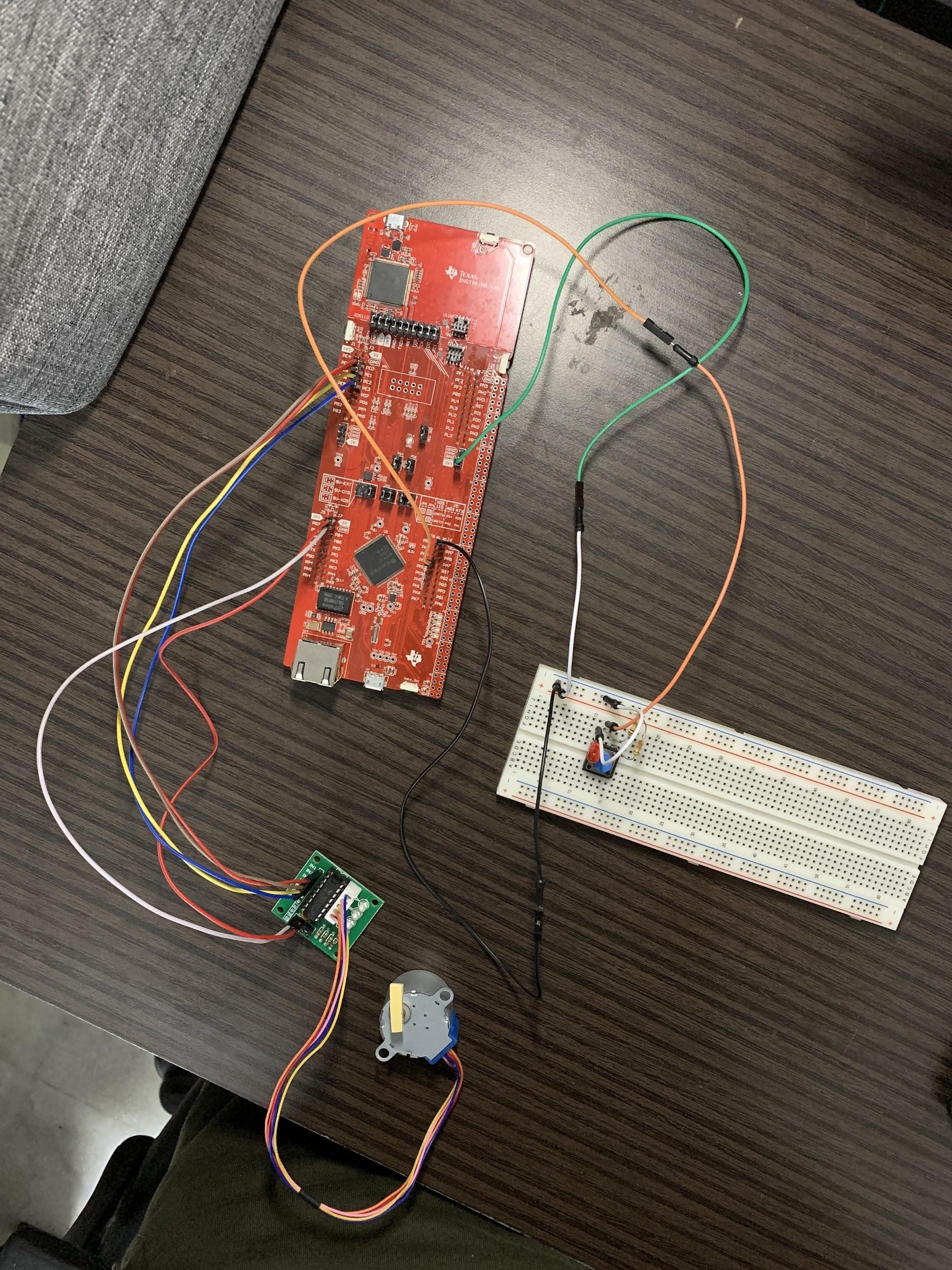
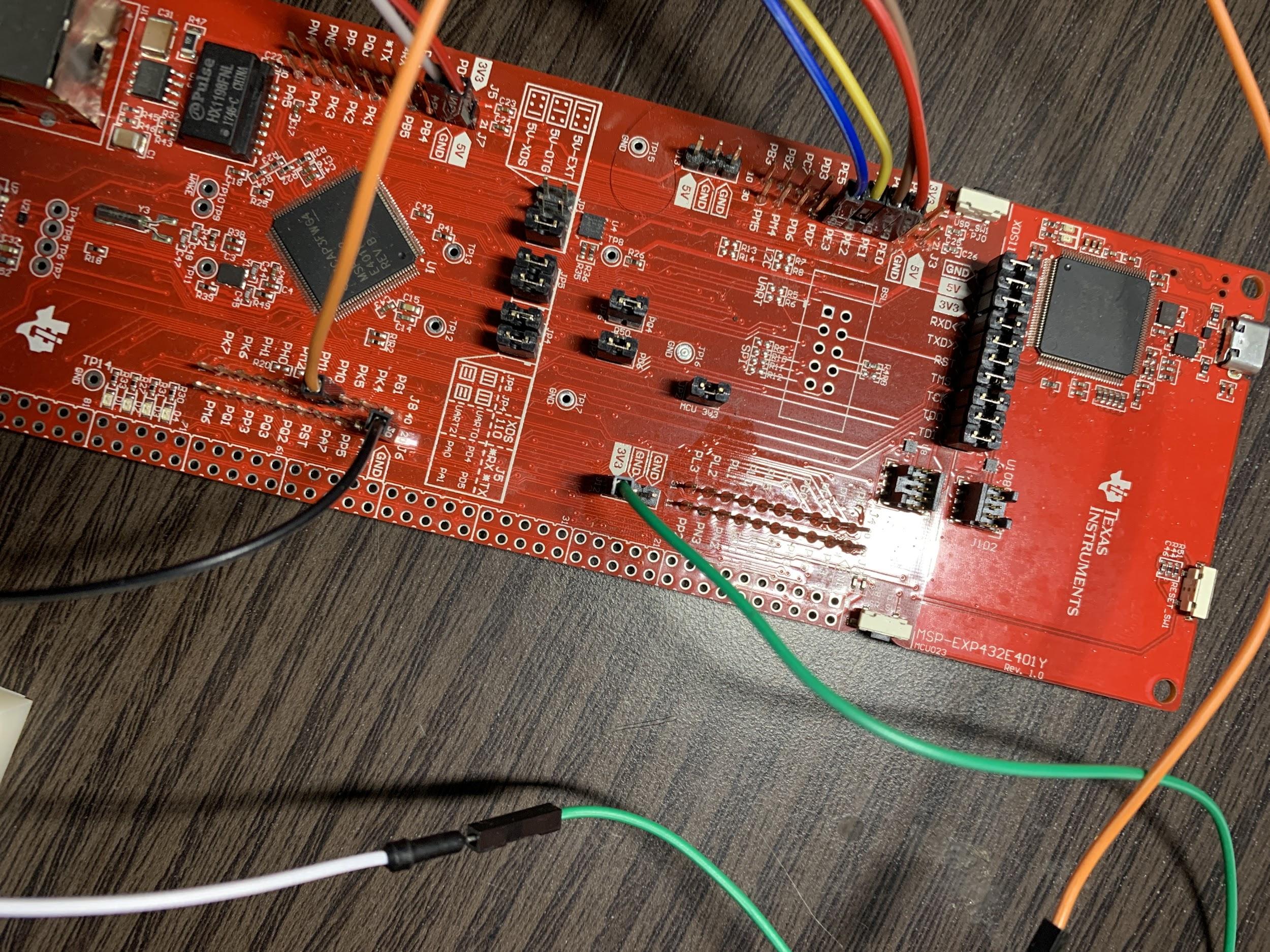
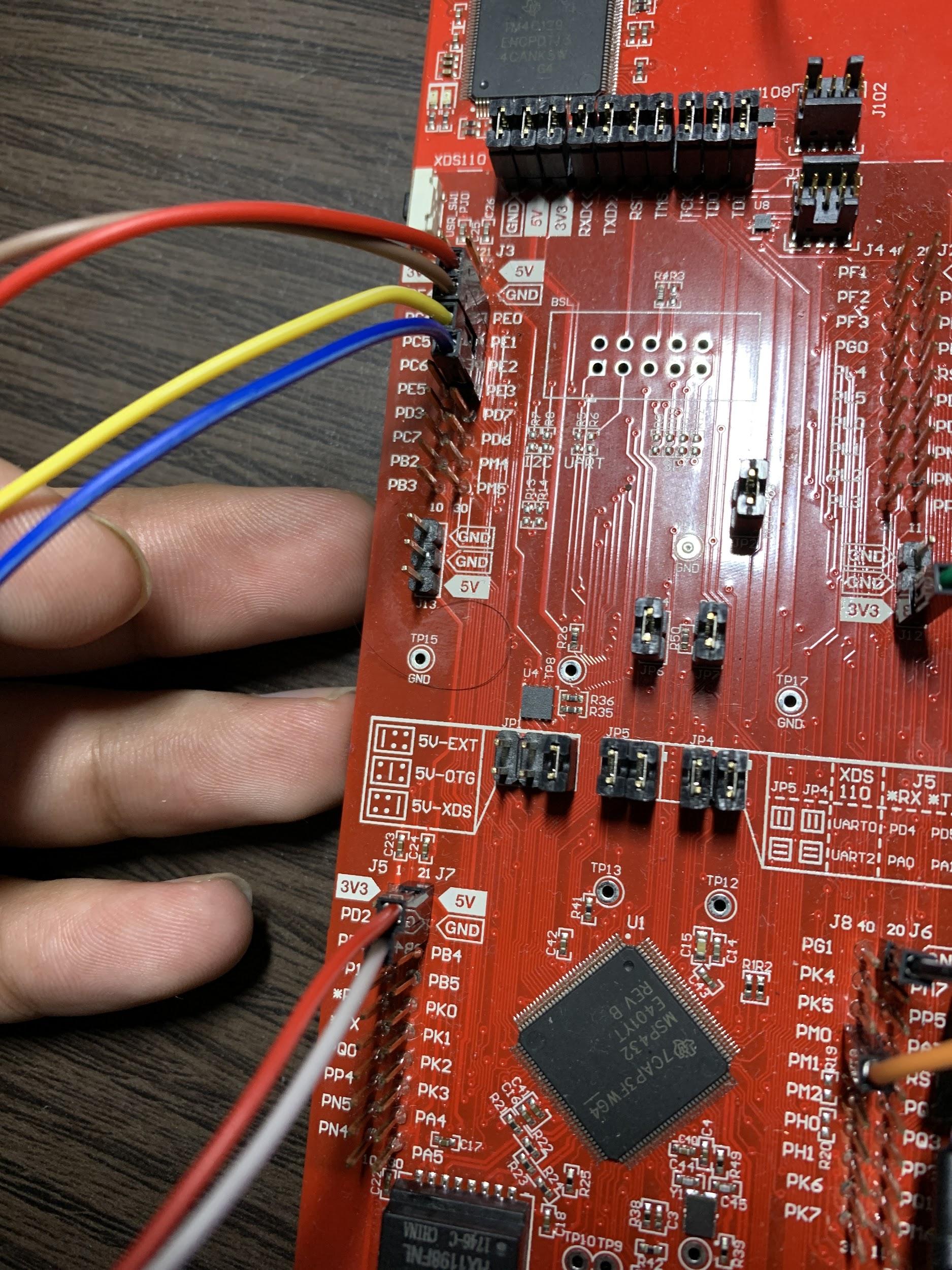
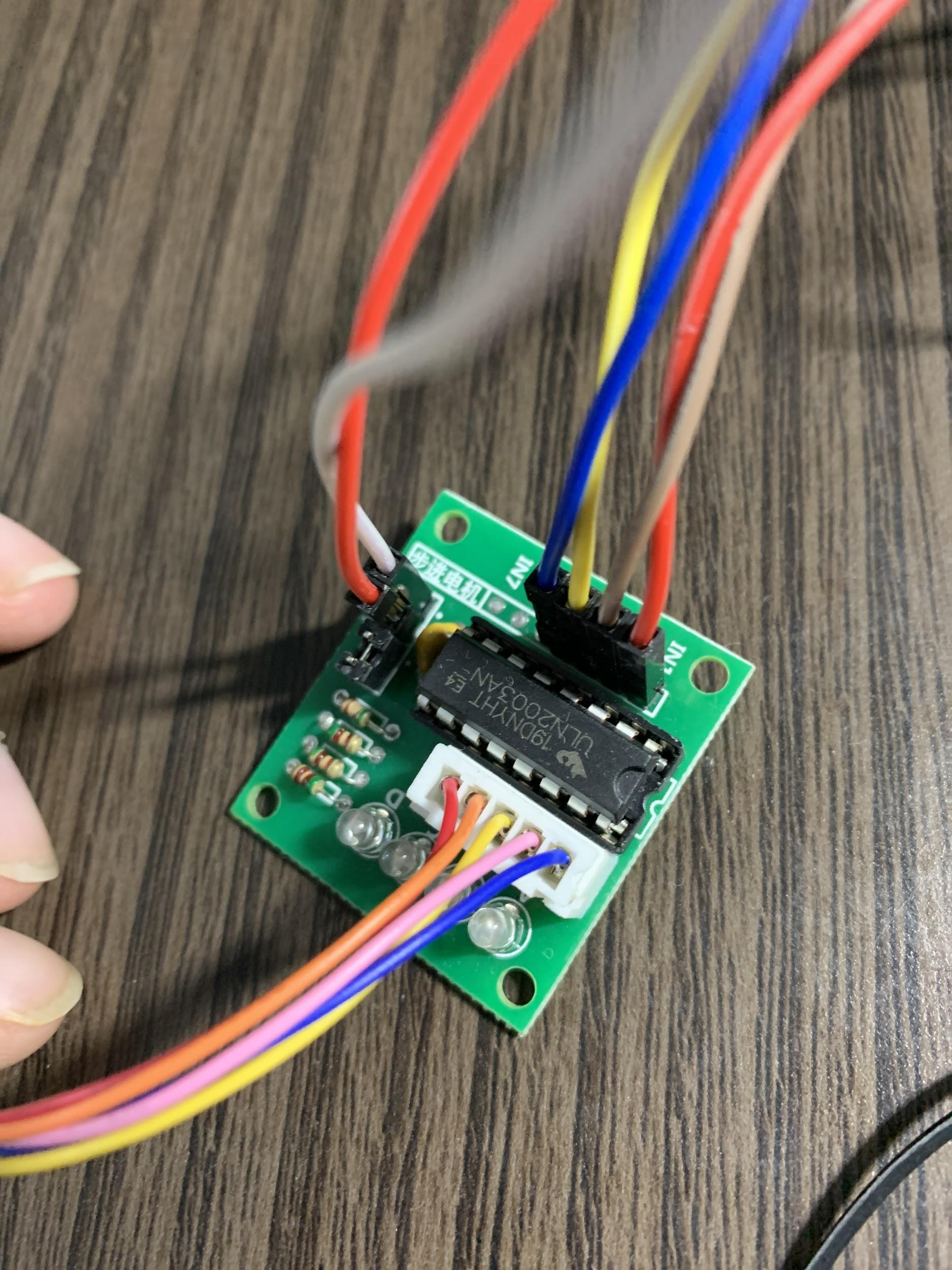
Have a nice day.





Stuff for my own reference:





Questions to be answered during the final project video:

1. How did you communicate the raw sensor data from your uC to your computer?

* Show student card
* So the first thing is, the microcontroller is connected to the computer using a USB cable. I used the microcontroller’s UART functionality and also the I2C protocol to enable communication. In my Keil project code, I added the UART.c file that was made available to us in the studios. That was used to send arrays of characters, in this case that data was distance measurements obtained by the time of flight sensor. On my PC, I used python to facilitate this data communication. The studio from Week 10 was followed to obtain the available ports, which in my case was COM4. I set the baud rate 115.2 kbps. I wrote the code in python to replicate the functionality of realterm. I used the pyserial library. This is how I was able to send raw distance data from my microcontroller to my PC.

1. What data processing did you perform on the raw data?

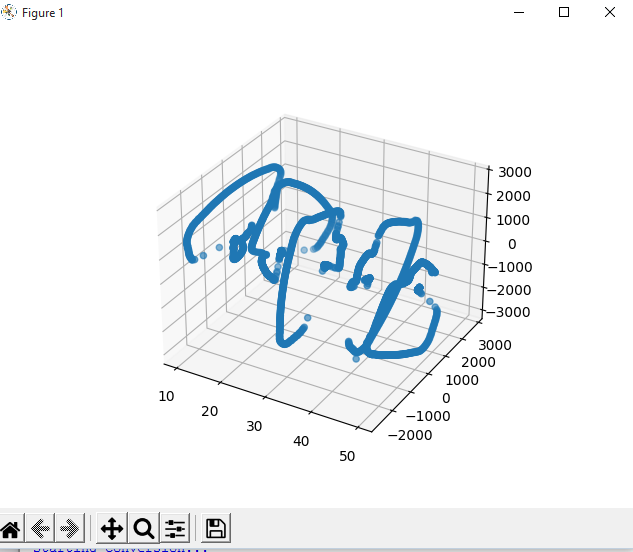
* The 3D rendering software I used for the project was MatPlotLib which is built into python. The raw data collected was basically the distance that the Time Of Flight sensor was measuring during each measurement. The data sent from the uC to the computer was the x displacement value, the tof distance value and the angle of the stepper motor. This needed to be converted into X, Y and Z coordinates for visualization. I performed this conversion in the python code that i wrote. So basically, X remains the same. And Y was the Y data multiplied by the cosine of the angle and similarly Z was the Z data multiplied by the sine of the angle. The angle that I am talking about was first converted into radians from degrees.

I used 3 separate arrays to hold these values.

1. How did you visualize the processed data using 3D rendering?

* So I used matplotlib to visualize the data. As mentioned in the previous answers i already had the data converted into xyz coordinates. I repeated the process of collecting these points a number of times. Then I imported all of those points as a point cloud. I used a process similar to what was taught to us in the last lecture except I did not use Open3D. So I continued on my python code, imported various other libraries like the Axes3D library and all this code I already have submitted in the zip file. Then further I used the plt.figure() function and then the ax.scatter() function to generate the plot similar to the one shown in my project demo.

5 measurements output:



10 scans

