Sign Glove Final Project

ECE167 Final project 2021-22

Kyle Atlas and Jacob Sickafoose

Commit ID:
6d159e60eefe349397e10d4ffe1c92bc9c9da2db

Introduction:

A wearable sign language translation glove will be made using the uC32 and many of the sensors used so far in the class. More specifically, it will use 5 flex sensors- one on each finger of the glove-to convert gestures into words printed on the Oled of the uC32. This involves translating the feedback from the flex sensors into its corresponding letters in the sign language alphabet. Full coverage of the sign alphabet would involve attaching touch capacitors and the BN055 to capture all possible hand movements. This will be considered a stretch goal given the amount of time for the project.

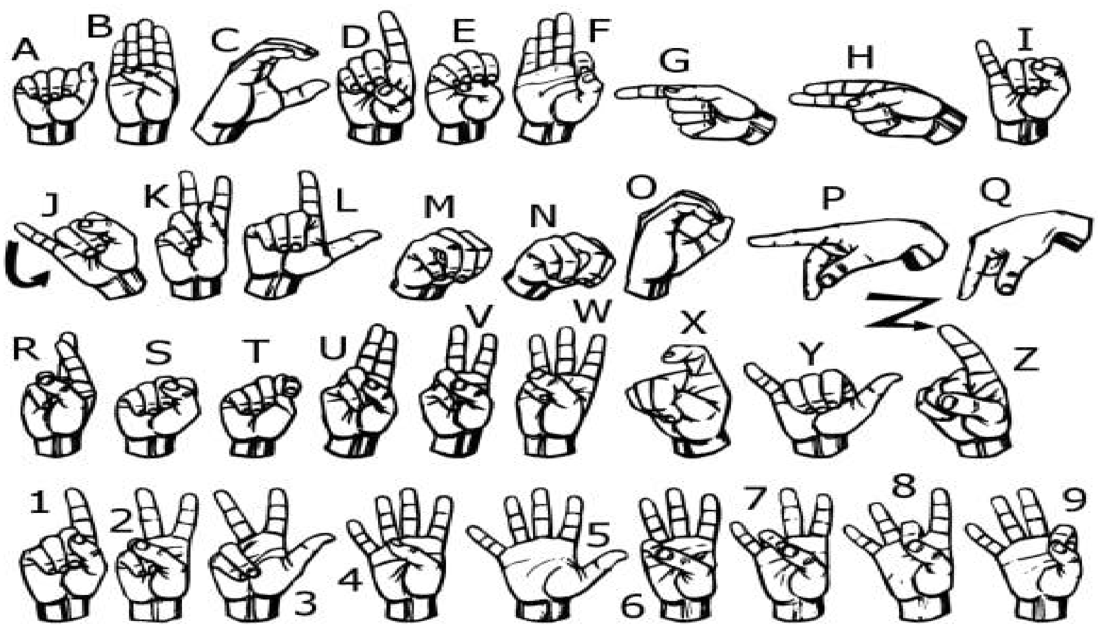


Figure 1: Full sign language alphabet

Above figure shows the goal of the project, incorporating the entire sign alphabet with numbers. Many complex gestures may not be able to be implemented as time is a limiting factor and we would need a strong background in machine learning. However simpler gestures such as “1” and “C” will definitely be incorporated into the project. A full list of implemented signs will be provided in the implementation section.

We will use the C programming language along with MPLABx and the uC32 to accomplish this project. The common libraries used in this class such as BOARD.h, Oled.h, etc. will also be used. For the hardware aspect, the gloves and flex sensors will be purchased separately along with any wires or jumpers needed.

Background:

For centuries, people who were hard of hearing or deaf have relied on communicating with others through visual cues. As deaf communities grew, people began to standardize signs, building a rich vocabulary and grammar that exists independently of any other language. This project aims to create a glove from translating these gestures into text anyone can understand. The text will be displayed on the Oled of the UNO32, but could easily be printed onto serial also.

Implementation:

Parts used:

5 flex sensors
10 jumper wires
8 wires
Hot glue gun
1 gardening glove
1 breadboard
3D printed mount

Project overview:

The following diagram represents the overview of the project: feedback from five flex sensors attached to the top of the glove will send feedback to the microcontroller, which will process that information using a control script and print the appropriate sign gesture on the Oled of the board.

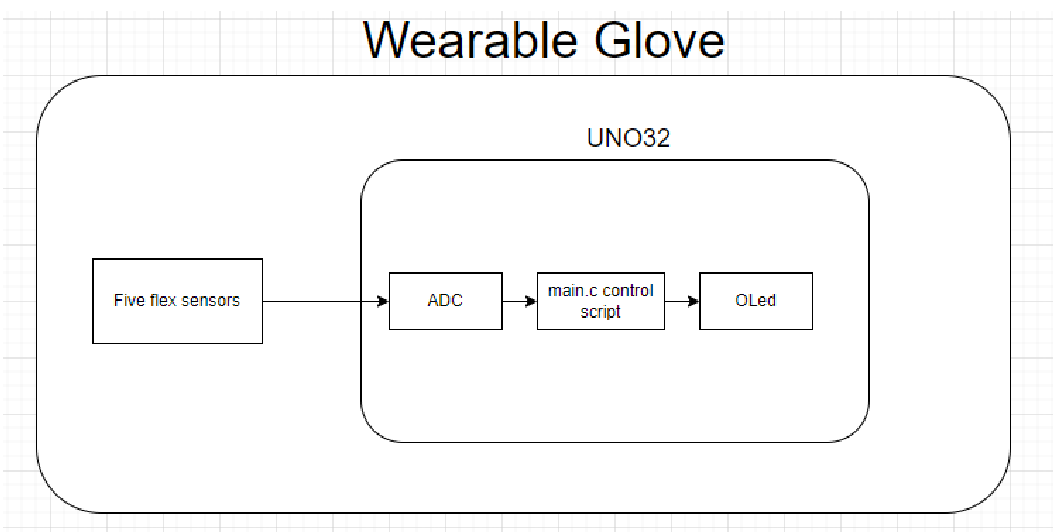
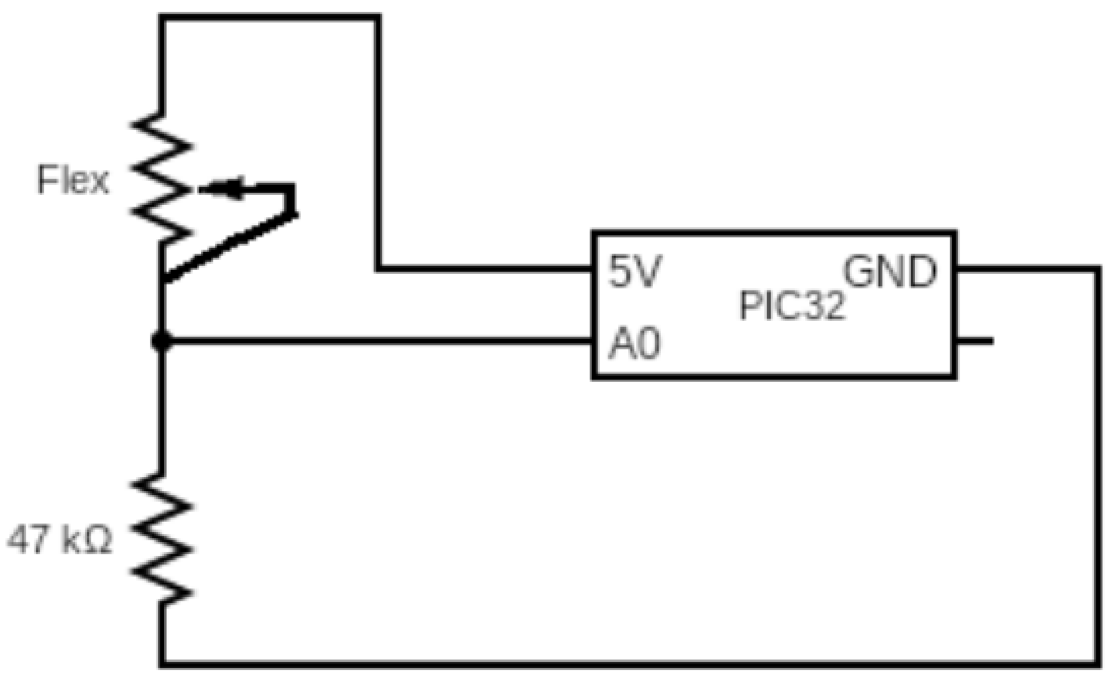


Figure 2: Overall implementation of sign glove

Hardware implementation:

The hardware implementation for the project began with sourcing the glove. We chose a gardening glove because it was sturdy, universal fitting, and could handle repeated movement without wear and tear.

We then 3D printed a mount for the glove, which has a curve on the bottom to contour the hand. Then, we hot glued a small breadboard to the 3D printed mount and created 5 voltage divider circuits with 47kOhm resistors.



Finally, we attached the male to male jumpers to the flex sensors and hot glued everything to the glove.

Final glove design:

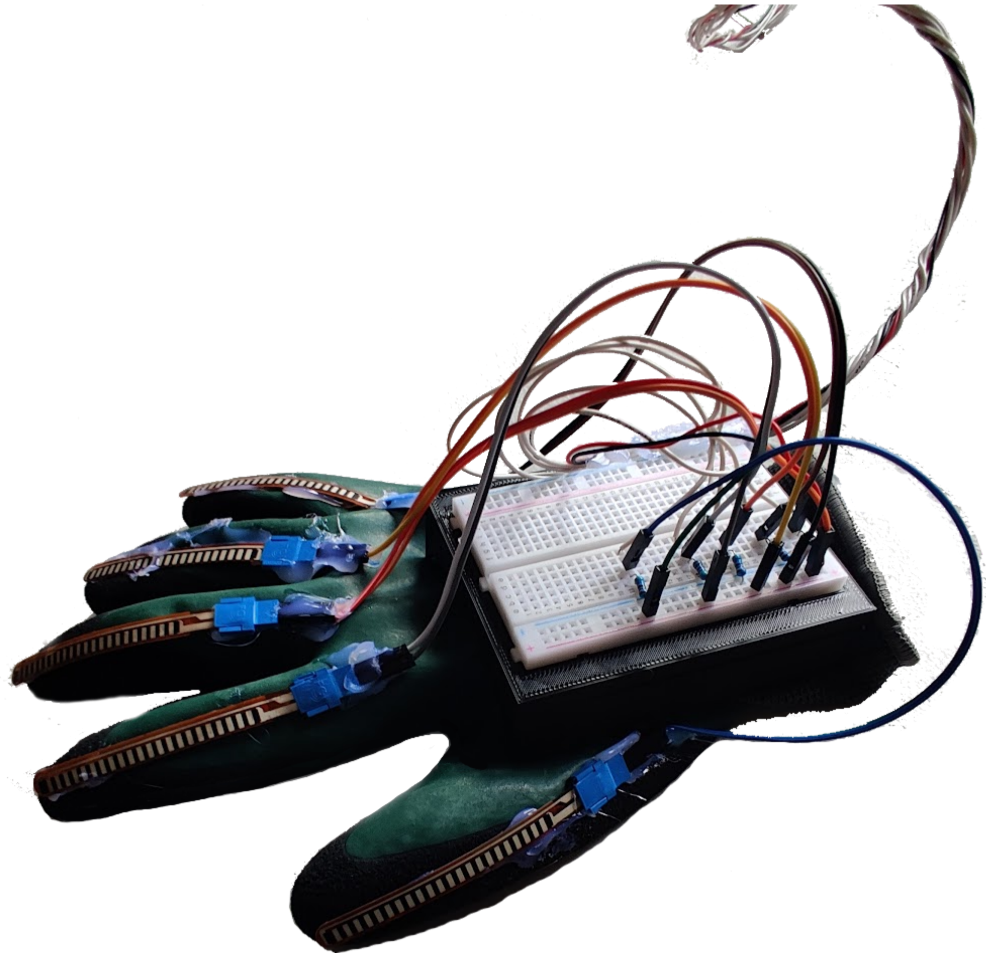


Figure 3: Final sign glove design

In order to keep track of which ADC pins correspond to which flex sensor, we color labeled the wires with labels. The figure below maps the index of that particular flex sensor in the code to the ADC pin used.

Finger	Pin Number	Labeled color	Index
Thumb	A0	None	0
Index	A1	Purple	1
Middle	A2	Orange	2
Ring	A3	Blue	3
Pink	A4	Green	4

Figure 4: Hardware pin to color label table

Software implementation:
The goal of the software portion of this project is to implement real time classification of sign language that the observer can read via text on the Oled. We set out to accomplish this by creating a state machine that prints certain ascii characters to the Oled based on the feedback received from the five flex sensors- one on each finger of the glove. Knowing that the resistance of the flex sensor changes as it is flexed- from approximately 10k to 35k- we can map the change in voltage-and the resulting ADC reading- to predict the gesture made by the user.

Note: We ended up using many of the libraries provided to the class: "BOARD.h", "AD.h" "Oled.h","timers.h", and "serial.h"

The first point to recall is that this won't give us the full 0-1023 analog value range, as we found in Lab 1. Since each ADC pin gives different readings, we had to experimentally determine thresholds to check for to determine flexion and extension of the finger. The following table shows experimentally determined threshold ADC values (count) for 3 possible finger states: straight, curled, and half curled.

	Finger 0(Thumb)	Finger 1(Index)	Finger 2(Middle)	Finger 3(Ring)	Finger 4(Pinky)
Straight	45	1000	1005	850	1020
Half curled	35	930	995	750	1010
Full curled	28	840	980	555	1005

Figure 5: ADC readings for each finger

Specific gestures:

Extension and flexion: Covered by flex sensors as explained in the previous section.

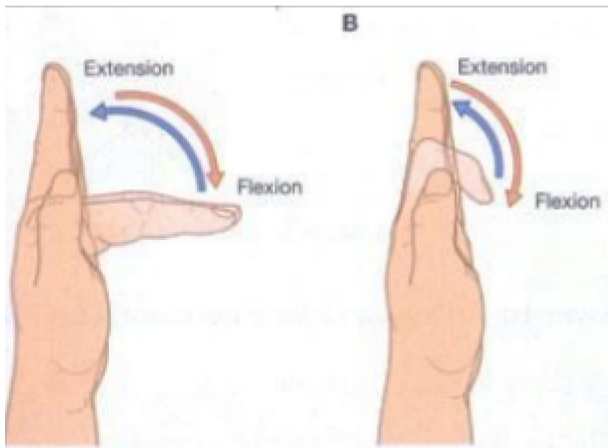


Figure 6: Extension versus Flexion

Using just flex sensors, we were able to implement the base 10 number range, 1-9, and the letters “A”, “C”, “I”, “X”, and “Y”. The rest of the alphabet was deemed too similar to each other to implement in the given time.

Example: The challenge of differentiating between the “A” gesture versus “S” gesture. This would require supervised learning such as Support Vector Machine to train the script to classify these two very similar hand gestures.

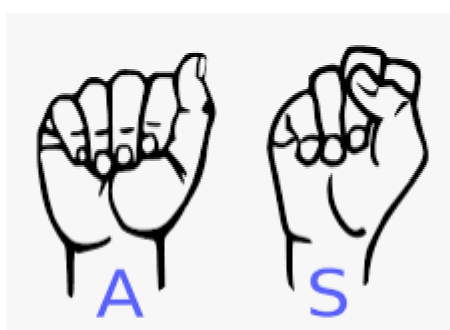


Figure 8: The challenge between determining “A” gesture versus “S” gesture

Abduction and Adduction: Stretch goal, covered by strategically placed touch capacitors on the sides of each finger. We were not able to implement this in time.

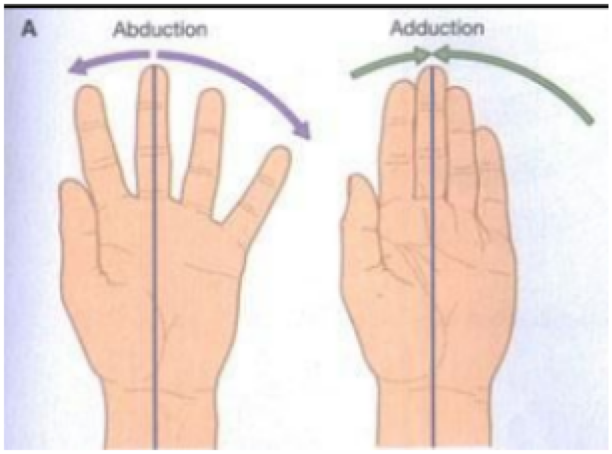


Figure 7: Abduction versus Adduction

Roll, Pitch, and Yaw of the hand: Stretch goal, covered by incorporating BN055 9dof IMU to the palm of the glove. We were not able to implement this in time, as gestures using yaw, pitch, and roll of the hand would require machine learning as stated above to differentiate between similar gestures.

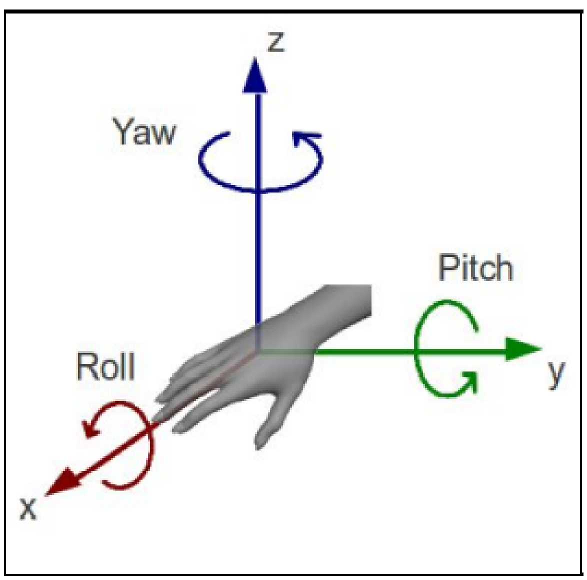


Figure 9: Yaw, Pitch, Roll of the Hand

Evaluation:

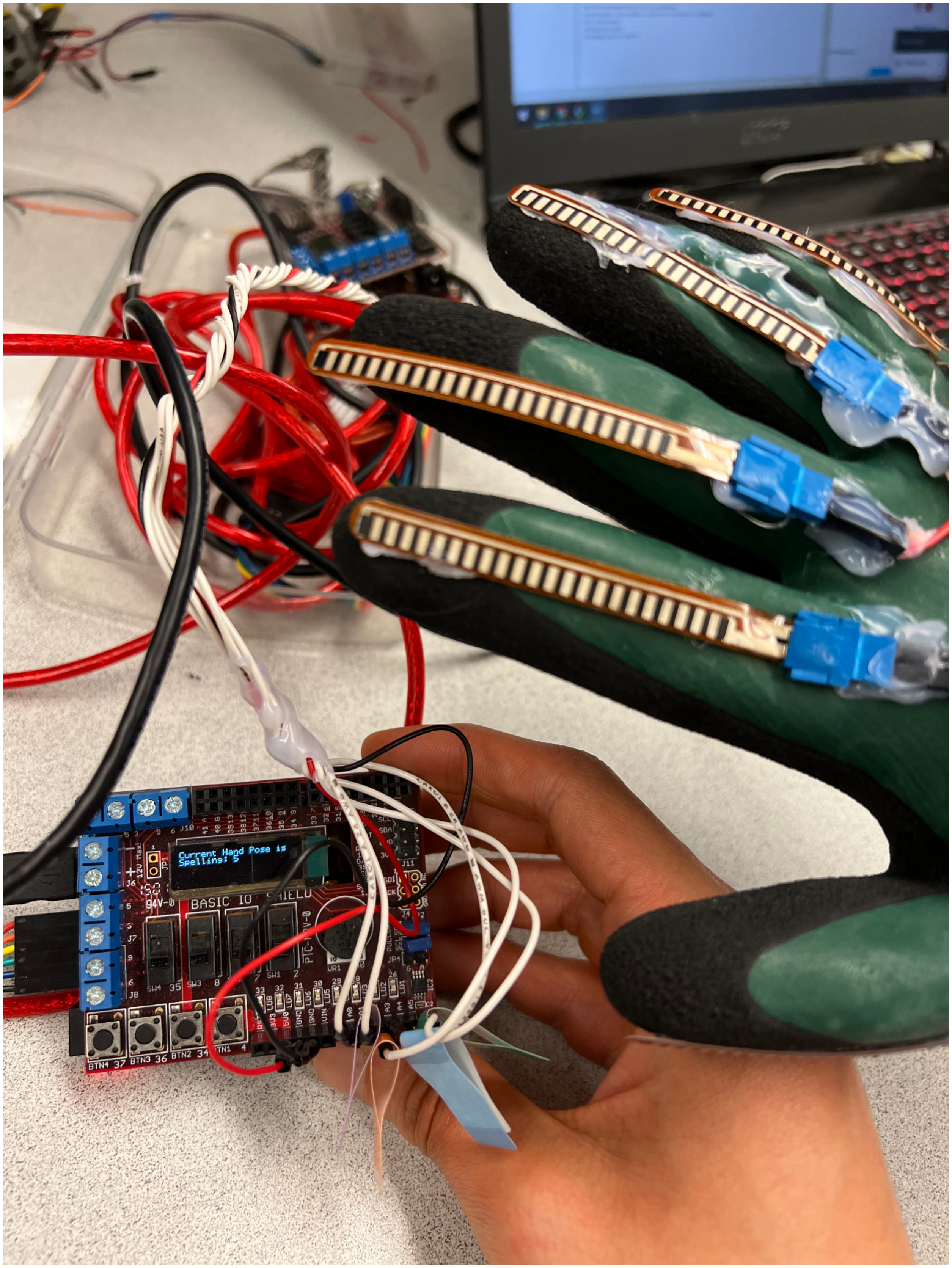
Testing and validation of this project started with the hardware. This involved building the glove with the 5 flex sensor array on a breadboard, and confirming it can be flexed and worn without breaking. More specifically, making sure jumpers and flex sensors do not slip out of place when under strain. We ended up using hot glue as it was much more effective than tape to mount flex sensors to a glove. Also, a breadboard provided a quick way to change around parts, rather than a perfboard circuit.

Testing software first began with using a test harness and printing the AD readings on the Oled from flexing the 5 flex sensors. Then, we moved to testing the main script with the flex sensor glove, and once again printing the resulting sign gesture in ASCII on the Oled.

Issues we ran into included sensor noise and poor AD range values. Some AD ranges such as for the pink finger barely had a usable range to read from (980-1025). Coupled with the noise from the flex sensors and possible difference in the way each person makes sign gestures, this would require machine learning to train the algorithm to detect the correct gesture. This would involve having a training time between each different user of the glove where the glove adjusts to each person's individual way of making sign gestures, and adjusting as necessary.

Simple gestures such as printing the number range 1 through 9 went smoothly. Also, letters such as "A" and "C" were implemented without trouble. As stated above, differentiating between "A" and "S" was not possible without supervised learning.

Picture belows shows converting the "5" sign into text on the OLed:


Current Hand Pose is
Spelling: 5
BASIC IO SHIELD

Discussion and Conclusion:

Our goals outlined in the proposal was to implement sign to text translation for the simple gestures of the sign language alphabet. More specifically, at least the 1-9 number range and partial coverage of the alphabet. I believe we were successful in this as we implemented the number range and 5 letters of the alphabet. This is a bit less of the alphabet than we first expected, but as we moved on in the project we realized more and more sign gestures were closer to each other than we first thought.

If we were to change our goals, we would have 4 members instead of 2 to allow us time to implement touch capacitors to the sides of each finger to add more letters of the alphabet. Shortcomings include not having the machine learning knowledge to train the algorithm to detect between similar gestures. Also, not having enough team members to justify implementing the BN055 IMU and touch capacitors in the given time. These added sensors would require even more time to bug test and program and are trickier to use than flex sensors.

Overall, this was an enjoyable final project and we learned more about flex sensors and the steps involved in planning and executing a project design. If we were to do it again, we would find a better mounting solution for the flex sensors instead of hot glue, as this rendered the sensors unable to be adjusted. Also, we would perfboard the flex sensor control board to make it look cleaner if time permitted instead of having a breadboard mounted on the glove.