Voice to ASL & Morse Code

Arduino Uno Microcontroller Project - Michelle Davies

Part 1

Overview

01 About the Product

Project Proposal and Purpose

Social Justice is an essential component of Engineering and Technological Development. When developers are creating new innovations, one of the most important questions to ask is “How can I use technology to help meet the world’s needs?”

When I enrolled in MAE 3780: Mechatronics, at Cornell University this past semester for the Robotics minor, we were challenged to create projects to address one of three prompts. Now that the semester is over, I have decided to make another project on my own accord to answer a different prompt, which is:

> The Human–Robot Interaction conference is looking for "interactive" robot designs. This year's competition theme is “A Robot to Support Us Through Lockdown.” Blossom, in Figure 3, designed in Cornell MAE's Human-Robot Collaboration & Companionship Lab, is here to remind you to be kind to yourself. Think of your experience since the beginning of 2020: how do you feel supported, comforted, cheered? How could a robot help you with that?
>
> We all have some common tasks we do around our home/dorm/office. Can you think of any task you can automate?

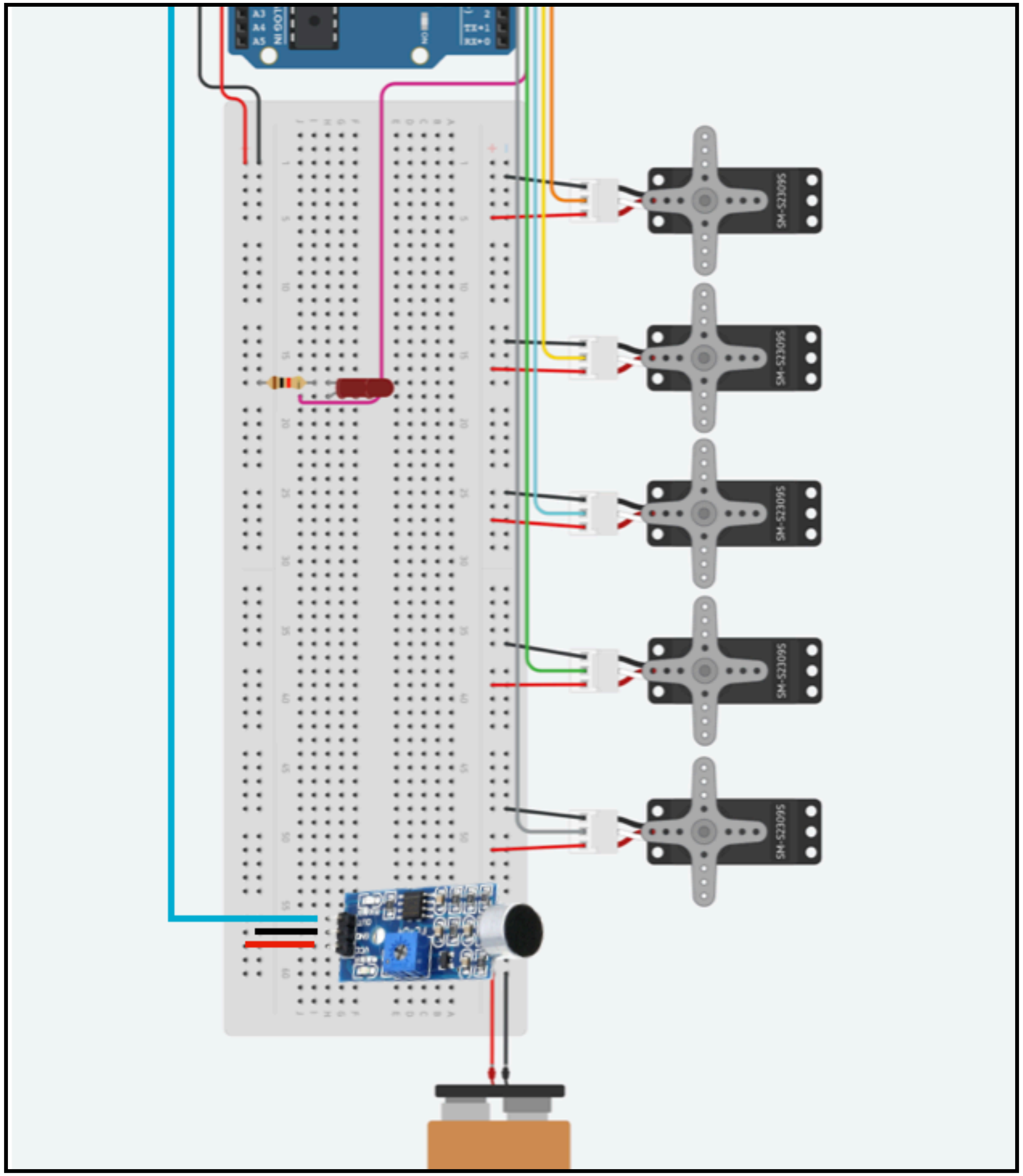
To answer this prompt, I wanted to focus on a home and office automation system that can help a group of people that have not only experienced the impact of COVID-19 on their daily lives as we all have, but now face more difficulty in receiving the services they need to adjust to this new “normal” and sense of daily life: the deaf community.

About 600,000 people in the United States, and 466 million people worldwide, are medically diagnosed as deaf or otherwise hard-of-hearing according to the World Health Organization (WHO). During COVID-19, many deaf/hard-of-hearing students are struggling to have their Individualized Education Plans (IEP) met in the wake of online learning. There are many solutions being developed to make it easier for able-bodied individuals to communicate with deaf individuals. However, few solutions exist to ease the barrier of communication for deaf people. So, I decided to develop a Voice to ASL and Morse Code robotic hand to decode verbal speech to visual signals and representations to bridge this communicative gap on the other end.

02 Circuit Diagram

Planning the Circuit

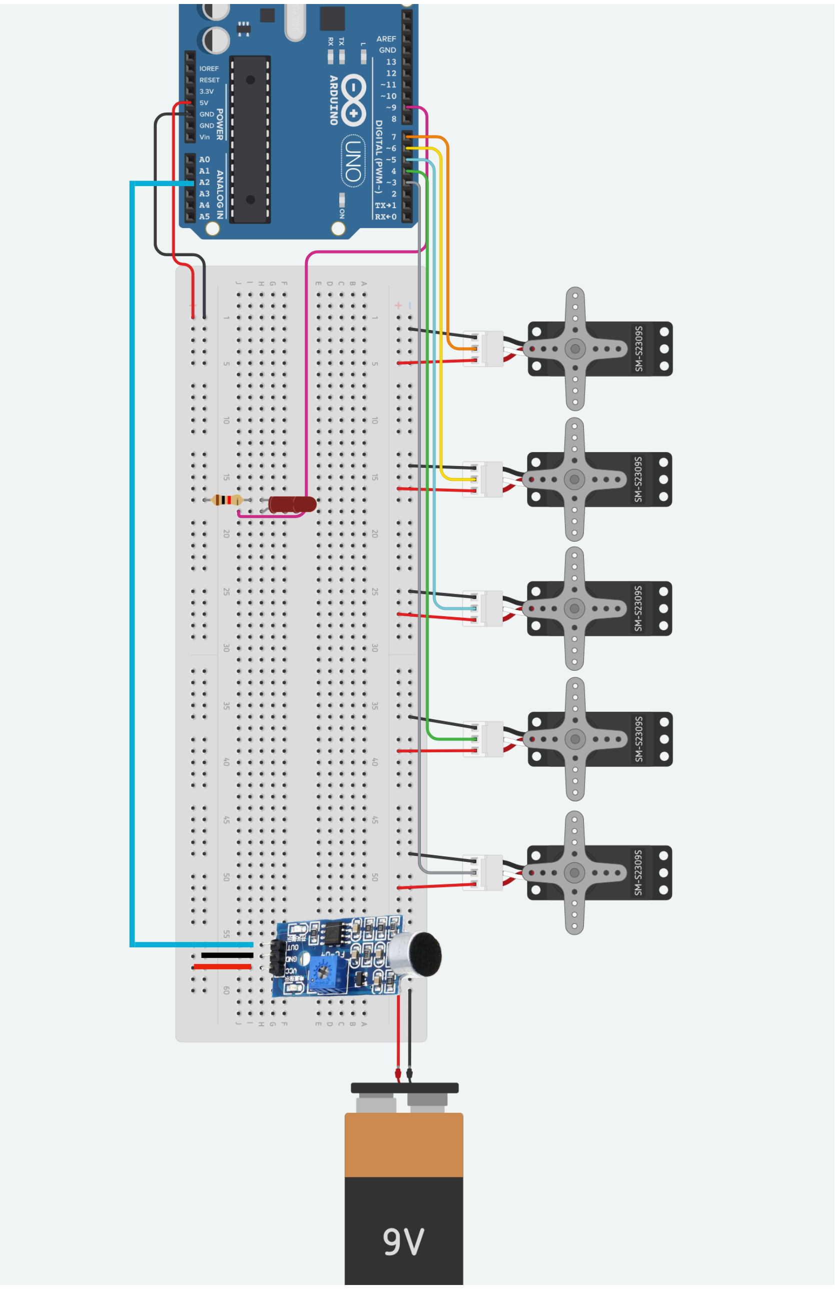
I used TinkerCAD to model the circuit for my device. Aspects of the circuit that are important to note are that I am using 3 LEDs in parallel for the Morse Code output in order to provide accessible replacement, and include backup lights. Originally, I was also going to add an additional Sound Sensor to limit random bias, but ultimately I decided to limit the complexity of the sound sensor to one degree.



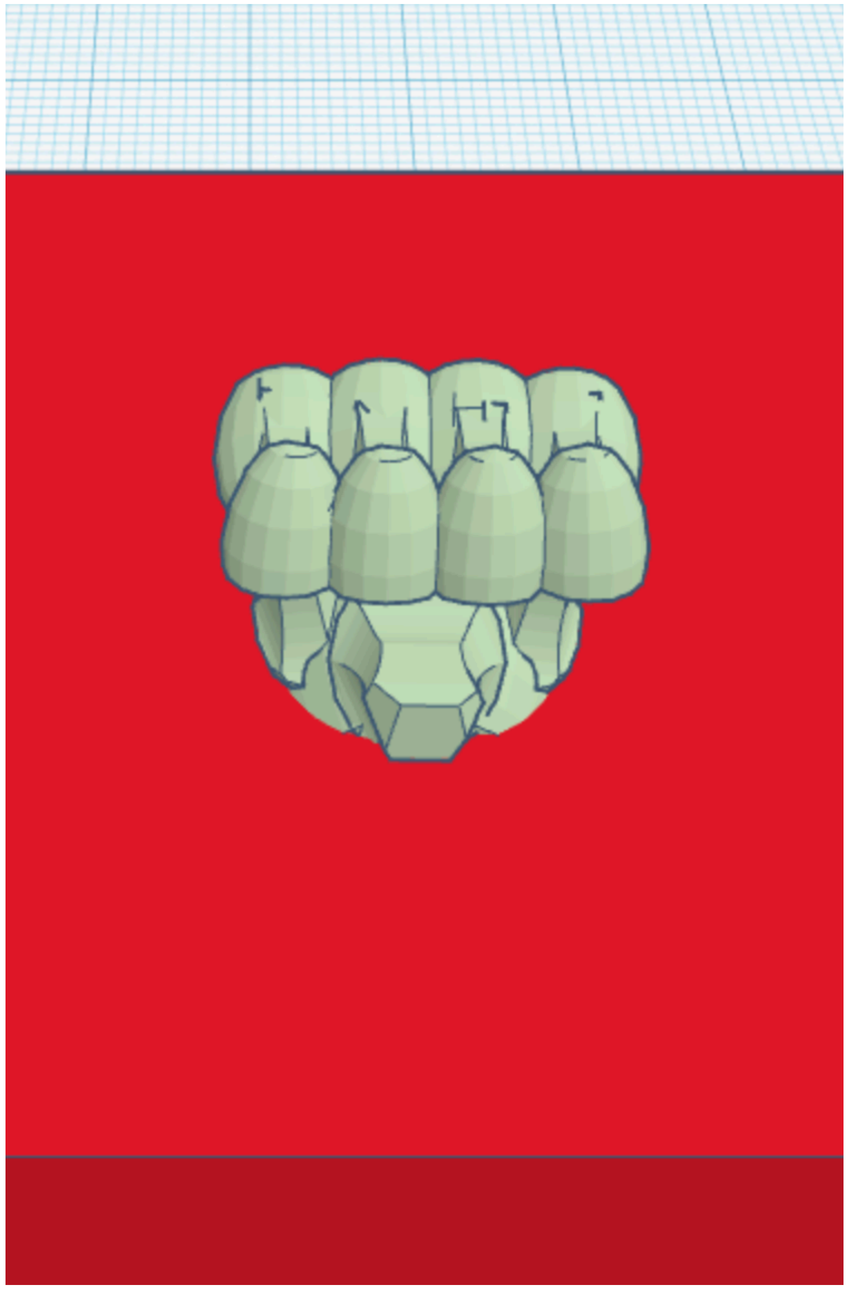
Materials:

- DAOKI Sound Sensor
- 3 LEDs
- 3 1k Ohm Resistor
- Arduino Uno Microcontroller
- 5 Servo Motors + Fan Blades
- 1m String
- Hand Model & Packaging

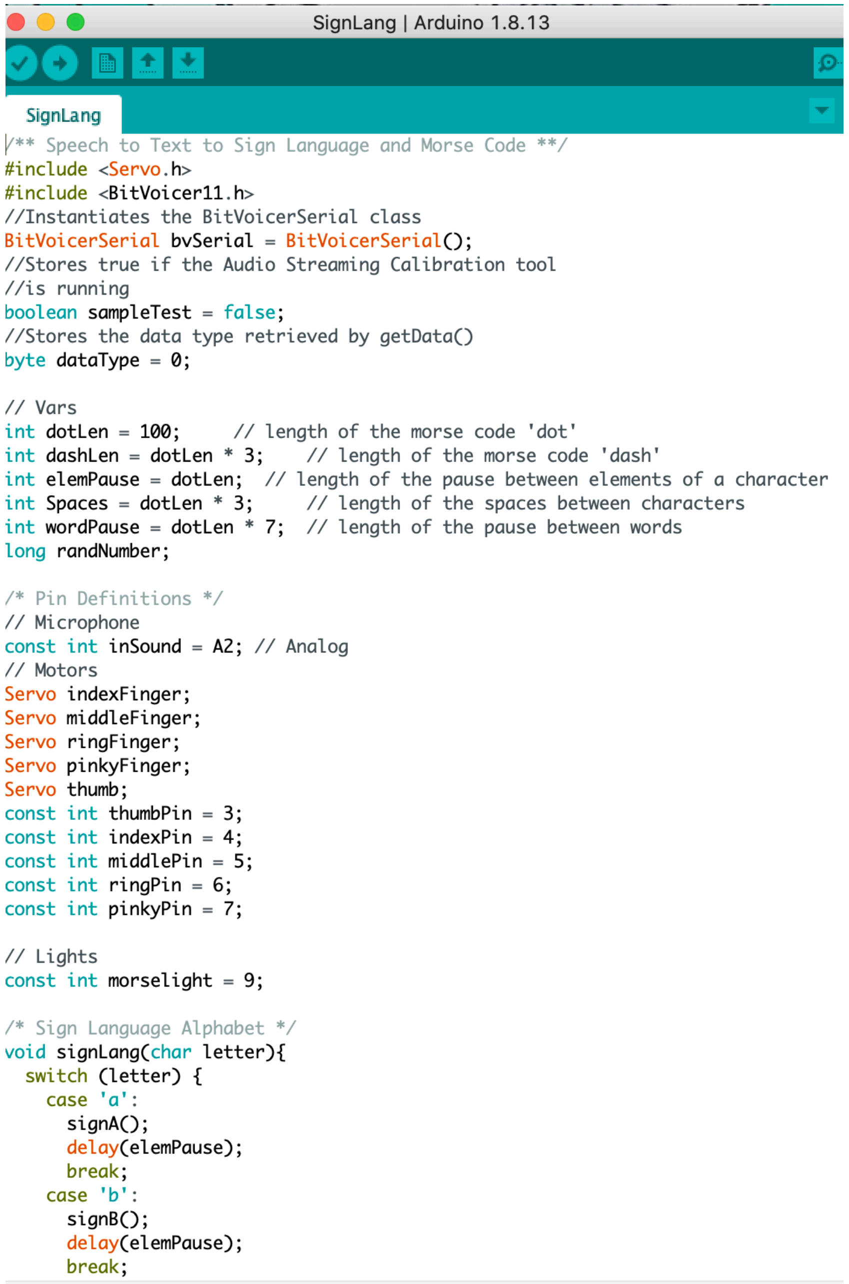
03 Design Schematic



This is the TinkerCAD circuit for this project, with the 5 motors, the sound sensor, and the three LED lights.



This is a 3D model of the physical model of the automated hand. The circuit is encased in the prototype's base.



```
/** Speech to Text to Sign Language and Morse Code **/
#include <Servo.h>
#include <BitVoicer11.h>
//Instantiates the BitVoicerSerial class
BitVoicerSerial bvSerial = BitVoicerSerial();
//Stores true if the Audio Streaming Calibration tool
//is running
boolean sampleTest = false;
//Stores the data type retrieved by getData()
byte dataType = 0;

// Vars
int dotLen = 100;     // length of the morse code 'dot'
int dashLen = dotLen * 3;    // length of the morse code 'dash'
int elemPause = dotLen;  // length of the pause between elements of a character
int Spaces = dotLen * 3;     // length of the spaces between characters
int wordPause = dotLen * 7;  // length of the pause between words
long randNumber;

/* Pin Definitions */
// Microphone
const int inSound = A2; // Analog
// Motors
Servo indexFinger;
Servo middleFinger;
Servo ringFinger;
Servo pinkyFinger;
Servo thumb;
const int thumbPin = 3;
const int indexPin = 4;
const int middlePin = 5;
const int ringPin = 6;
const int pinkyPin = 7;

// Lights
const int morselight = 9;

/* Sign Language Alphabet */
void signLang(char letter){
  switch (letter) {
    case 'a':
      signA();
      delay(elemPause);
      break;
    case 'b':
      signB();
      delay(elemPause);
      break;
```

This is the Arduino code that is uploaded to the microcontroller to control the functionality of the project.

Part 2

Development Criteria

04 Timeline

Development Timeline

In building this project from start to finish, I made a list of the components of the project that I needed to complete in order to construct it. In making this list, I was able to develop a timeline for completing this project. The timeline is largely informed by the availability of certain physical parts of the device. However, the timeline is ultimately a result of the developmental process from code to circuit to design.

Step 1: Writing the code from the sensor to translate audio into text, which can then send output signals to the actuators to display the message.

Step 2: Testing the sensors and actuators.

Step 3: Wiring the circuit for the prototype.

Step 4: Building the physical prototype.

Step 5: Do a run-through.

05 Criteria for Successful Operation

Benchmarks for Project:

- Can the project convert audio into text with 70+% accuracy?
- Can the program output the correct corresponding letters in morse code form?
- Can the program output the correct corresponding letters in sign language form?
- Can the program sync up the I/O system to complete it's output signal for each character within human reaction time?
- Are the output signals clear and discernible?
- Is the product user-friendliness?