

Chopduino

Noah Jaussi

STEM Capstone Project

Abstract

The goal of this experiment is to design and build a device that can electronically control a pair of chopsticks with the right precision to be used by people, who have disabilities that prevents them from using chopsticks to feed themselves. This project went through several design iterations before the final design was decided upon. Using an arduino pro micro, 18650 battery cell, an analog switch, the arduino IDE, windows 3d builder, a soldering iron, and a 3d printer to manufacture and assemble the necessary components, this project was able to make use of Science, Technology, Engineering, and Math (STEM). The process of designing and troubleshooting would not have been possible if it wasn't for the research done beforehand, especially in the field of arduino wiring and programming. The process of prototyping this device followed the process of testing all of the electronics on a breadboard, soldering them together, programming the arduino to function properly, modeling the base structure, 3d printing the necessary components, and assembling the frame and the electronics into their final form to be trialed for functionality improvements. There were many challenges encountered that arose in each technical division of this project and it continues to have room for improvement in future designs, should this project be continued, in the areas of size, comfort, sound deadening, and aesthetic appeal however it did fulfill all of the engineering goals it was originally set out to accomplish.

Introduction

In order to create the object of this project there are several techniques, programs, and skills that need to be acquired to ensure success. First it is necessary to understand the hardware required and the essential properties/interactive capabilities of said hardware for optimal performance. This includes the physical parameters for the servo, connections of the mechanical switch and button, and the pinout/specs for the microcontroller. These devices will be required to interact with the program set to be installed on the microcontroller so as to perform the functions it is intended to complete. In order to accomplish isolated and untethered operation, a rechargeable 18650 battery cell will be included, as it's nature allows both high current power draw for the servo and consistent power output for the raw inputs on the microcontroller. Aside from the electronics, a housing will need to be constructed from 3d printed plastic which has to be confirmed to be safe for human contact. This is especially vital in this project, due to the nature and proximity the 3d printed plastics will be to objects being consumed by users. The plastic will need to be strong enough to endure the nature of its intended use while also being flexible enough to provide smooth movements for precise control.

The most essential part of this project is the microcontroller, which will require proper

implementation of its pinout abilities and programming. The Arduino Pro Micro includes 14 digital in/out (also I/O) leads and 4 analog I/O leads. This project will require one digital lead to communicate with the servo and one analog lead to interpret the incoming information from the user. The user will interact with the project via a button switch on the side. After properly connecting the switch and servo the Arduino, the micro controller chip will need to be programmed using an Integrated Development Environment (IDE), such as the one provided at arduino.cc for the specific ATmega chip on the Arduino Pro Micro. There are many options for IDE's, however, the arduino.cc IDE was chosen for its preferential level of documentation and development status. This IDE also includes firmware that is needed to ensure that the ATmega chip in the Arduino is programmed without any issues and will utilize proper functionality. Detailed information and more on these subjects were found at Arduino.cc and the *Make: Getting Started with Arduino* book (Source 1).

Programming this Arduino microcontroller will require not only hardware documentation but an understanding of simple C++ programming knowledge, as is dictated by the IDE. This is necessary as the complex nature of the task, in terms of electronics, is not something that can be easily performed without a microcontroller. The arduino will need to communicate user input from the switch to the servo in a way that translates on/off into a degree or number of rotations that the servo needs (Source 3). These translations of electrical signals will be interpreted and sent out through the program given to the arduino microcontroller.

In order to power the Arduino and its necessary accessories, an 18650 lithium ion battery cell will be included with a voltage and current regulator to ensure proper power distribution is delivered to the electronics. This amplification and control will occur on the arduino board. It will also limit the amperage in order to protect the sensitive electronics from the high power draw of the servo motor. The 18650 battery cells are lithium ion, which means that the battery in this project will need to have overcharge/discharge protection circuits. This is necessary, not only to ensure battery life, but also to prevent the battery from becoming a fire hazard. According to previous research and the properties of power distribution in arduinos, these concerns will not manifest should the project be assembled exactly according to its design guidelines (Source 2).

To make the prototyping and development of this project faster, a 3d printer will be used to make the frame and housing for the electronics as well as other necessary items. Having plastics that are toxic to humans and not rigid enough to support the required components are important design concerns to be considered for this project. Fortunately, these concerns has been tested before and it has been found that consumer grade Acrylonitrile butadiene styrene (ABS) and Thermoplastic Polyurethane (TPU) plastic filament are both not toxic to humans (unless burned and fumes are inhaled, see source 5) and provide the mixture of strength and flexibility required for this projects success (source 4). There are many other plastics that could be used but considering this project's proximity to food that will be consumed, ABS and TPU plastics were decided upon as the final developing material for this project.

Materials and Procedures

Materials:

- Computer to run Windows 3d builder
- 3d printer (creality ender 3)
- ABS + TPU 3d printer filament

- 18650 battery
- Arduino pro micro
- Analog switch
- Breadboard
- Wire clips
- Chopsticks
- Soldering iron + solder
- Wires
- Small tension spring
- 9g servo

Procedure:

1. Prototype electronics circuit on breadboard
2. Revise until all connections are functional
3. Design code for arduino
4. Upload and test code in arduino
5. Revise code until proper functionality
6. Open 3d builder and design first 3d printable prototype
7. Save 3d model in cura printing software
8. Upload model to 3d printer
9. Solder electronics together while model is 3d printing
10. Assemble 3d model and observe potential changes
11. Return to 3d builder to add changes to the 3d model
12. Begin print of revised model
13. Repeat assemble, observation, edit, and 3d print until fully functional design is achieved

Data Analysis

Figure A: Prototype v.1

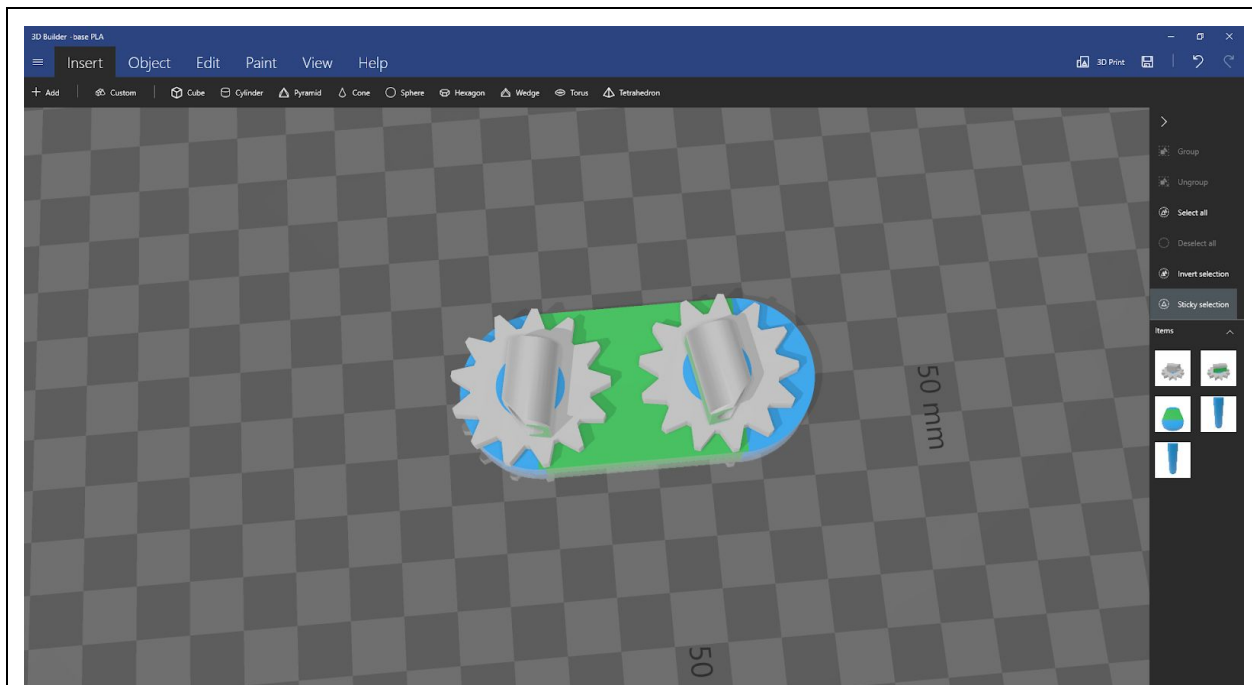


Figure B: Prototype v.1.2

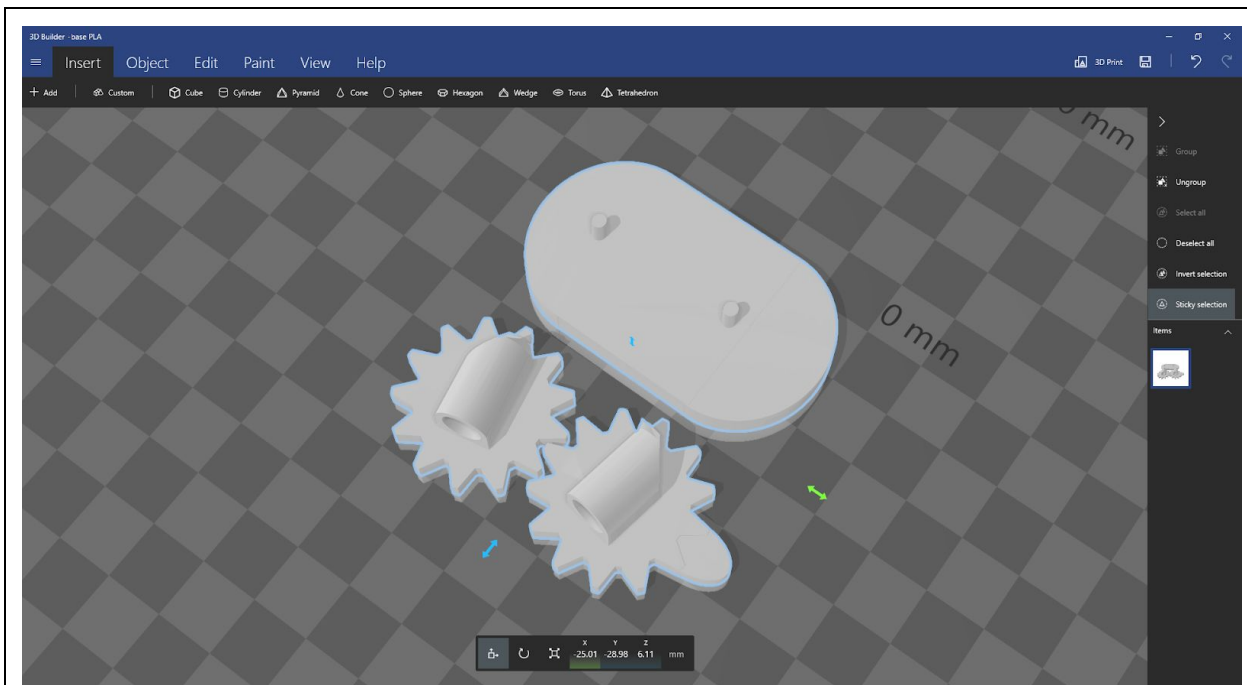
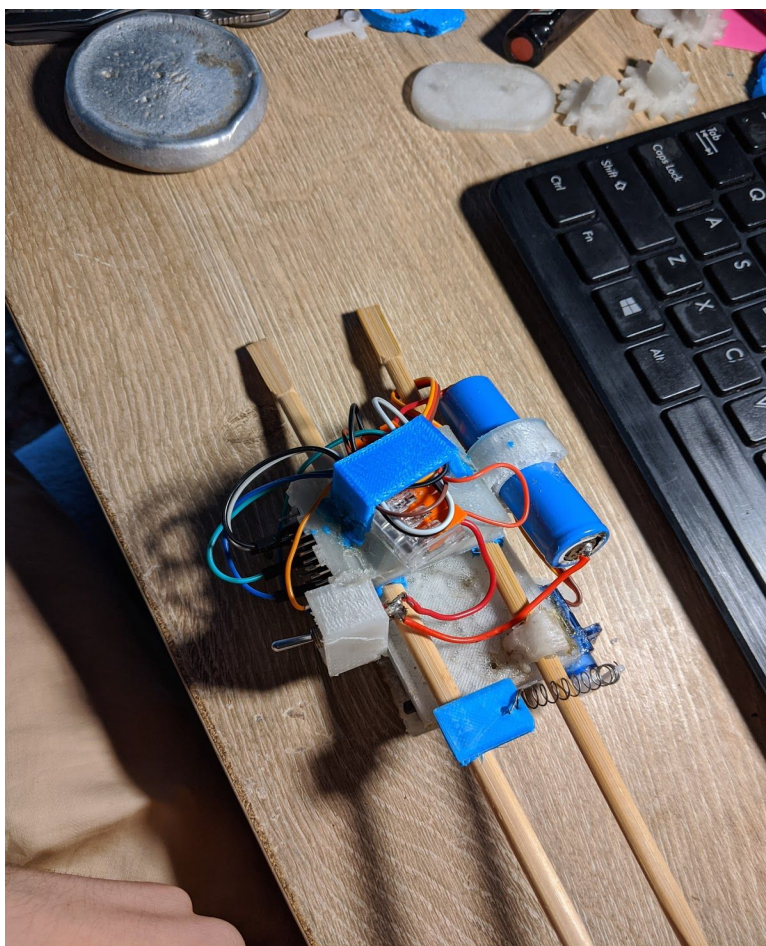
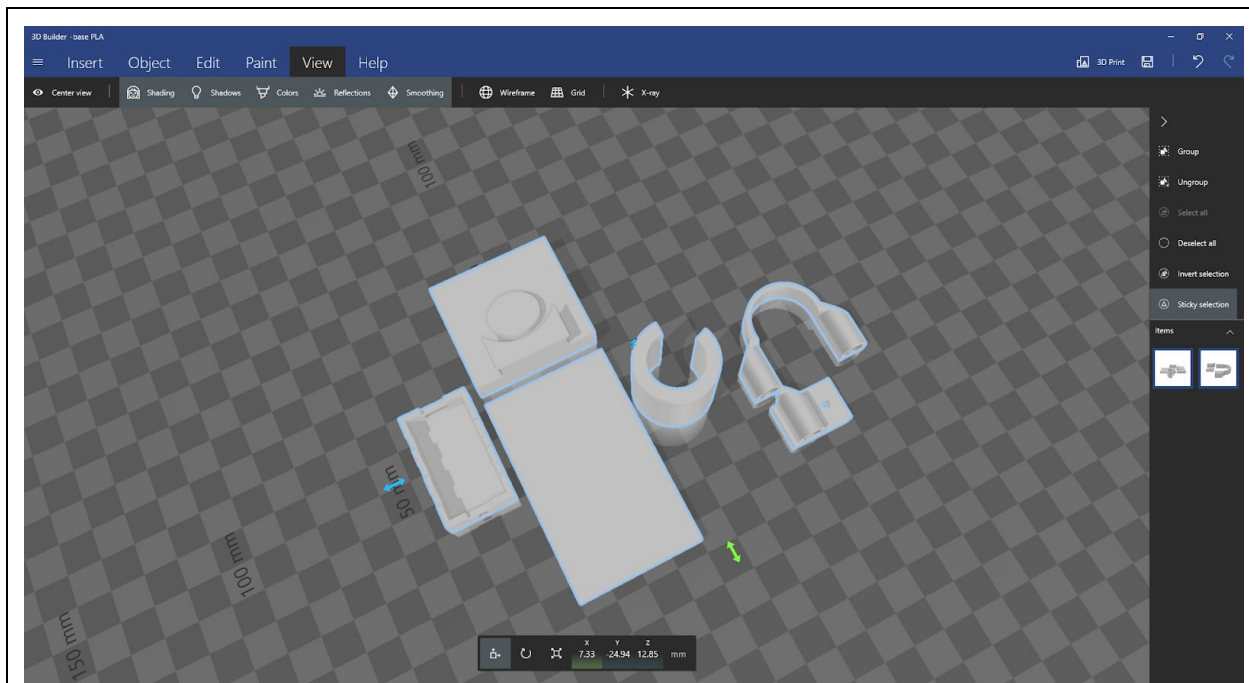
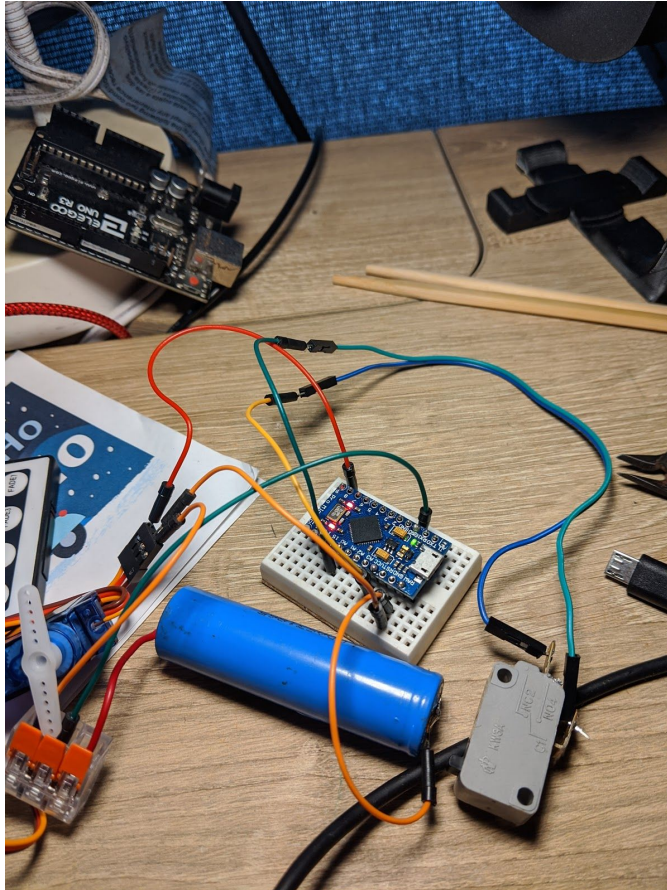


Figure C: Prototype v.2



Electronics v.1



Code v.1

```
#include <Servo.h>
Servo serv;

int switchA = A0;
int val = 0;

void setup() {
  // put setup code here
  serv.attach(9);
  serv.write(0);

  delay(100);
  val = analogRead(switchA);

  //pinMode(15, INPUT);
}

void loop() {
  // put your main code here
```

```

if(val > 1000){
    //serv.attach(9);
    while(val > 1000){
        serv.write(90);
    }
    } else {
        //serv.attach(9);
        serv.write(0);
        //serv.detach();

    delay(300);
    }
}

```

Code v.2

```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo
```

```
int pos = 0; // variable to store the servo position
```

```
int button = 4; // The button will be on Pin 4
```

```
int val = 0;
```

```
int switchA = A0;
```

```
void setup(){
```

```
    myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
    pinMode(pos, OUTPUT);
```

```
    pinMode(button, INPUT);
```

```
    digitalWrite (button, LOW);
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop(){
```

```
    Serial.println(val);
```

```
    val = analogRead(switchA);
```

```
    if (val > 1000)
```

```
        for(pos = 0; pos < 90; pos += 90){ // goes from 0 degrees to 90 degrees
```

```
        myservo.write(pos); // tell servo to go to position in variable 'pos'
```

```
        delay(30); // waits for the servo to reach the position
```

```
    }
```

```
    if (val < 1000)
```

```
        for(pos = 90; pos>=90; pos-=90){ // goes from 90 degrees to 0 degrees
```

```
        myservo.write(pos); // tell servo to go to position in variable 'pos'
```

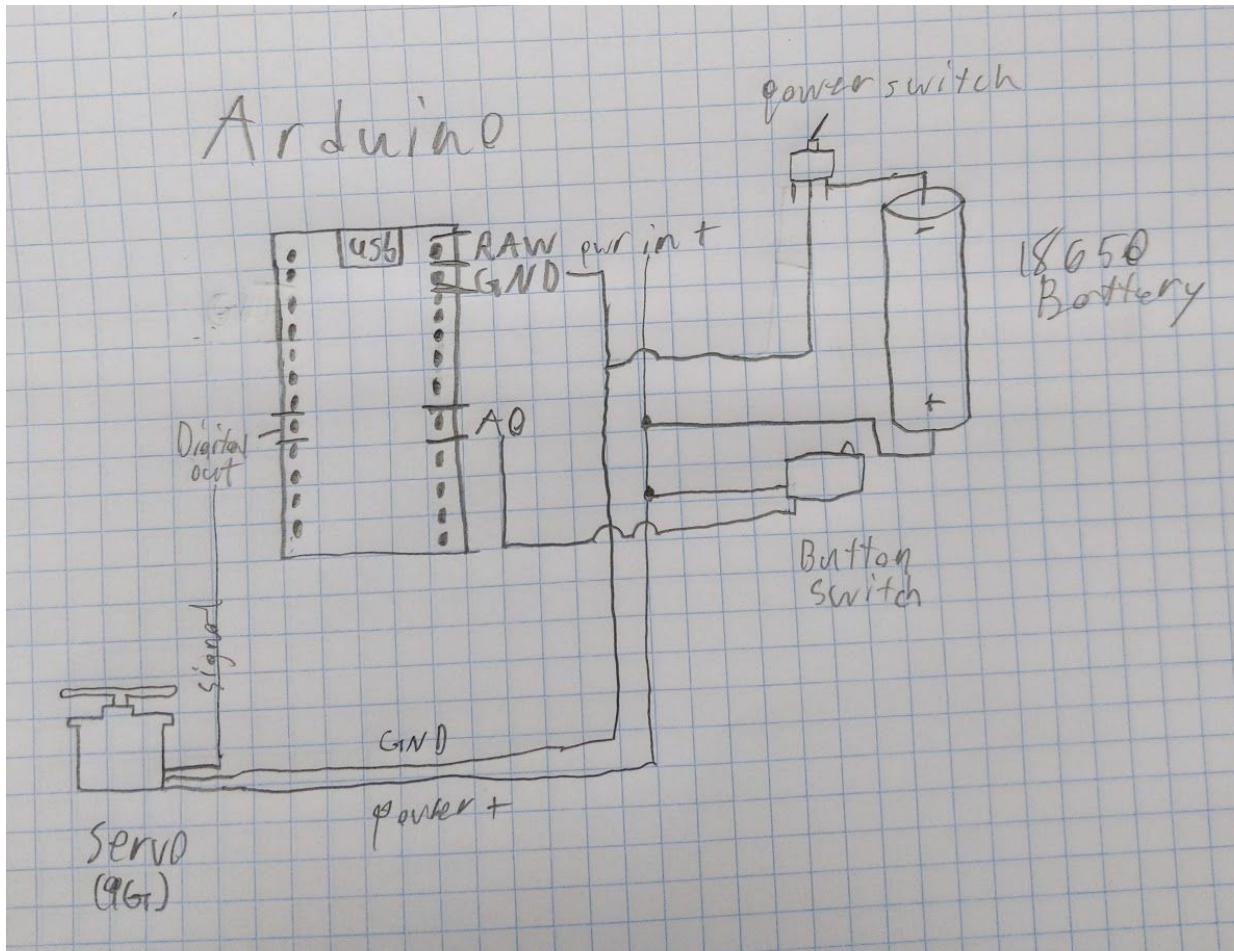
```
        delay(100); // waits for the servo to reach the position
```

```
        analogWrite(val, 0);
```



```
}  
}
```

Electronics Diagram



Engineering Challenges	Observations
1	<ul style="list-style-type: none">Initially the electronics were assembled and tested which proved trials as the analog pin for the switch was giving off unusual pulses making the servo uncontrollable. This was fixed with an <code>analogWrite(0);</code> statement to reset the analog signal on pin A0 to zero after every run of the program.
2	<ul style="list-style-type: none">While testing the electronics the first servo broke and would spin out of

	control so several more servos were tested until a 180* servo worked with full control.
3	<ul style="list-style-type: none"> When assembling the 3d printed frame the plastic had to be welded together using a soldering iron which was a challenge and required the use of multiple prints to get successful and strong bonds.
4	<ul style="list-style-type: none"> The servo placement on the printed frame required multiple trials before it could provide the proper tension on the spring to pull the pivoting chopstick into place and push it back when finished with its cycle. To solve this the servo was placed on the bottom of the overall device.
5	<ul style="list-style-type: none"> The final challenge before the working product was chopstick alignment. The moving chopstick had the challenge of shifting the still chopstick far enough out of place that it proved not functional. This was resolved by welding a plastic brace over the still chopstick.
6	<ul style="list-style-type: none"> There was difficulty getting an ir distance sensor to ship to Hawaii so a model function change was altered to use a button for activation of the software devices in place of the distance sensor as originally planned.

Conclusion

This project is based on the idea that STEM can create a device that provides functionality to those in need of assistive eating devices. The result of this idea was an electronically controlled, self-contained device that can close, hold closed, and open a pair of chopsticks under the control of a single button. This device encountered many challenges as prototypes were developed that had to be overcome in many unique ways that incorporated all fields of STEM.

In chronological order of development, the first engineering challenge encountered was found in the connections between the switch/button and servo to the ATmega chip controlling the Arduino pro micro. The pins were either being shared with other logic data lines on the Arduino or there was interference that would cause the electrical values of the pins to vary from 0 to 1000 (state of all off to all on). This complication would cause the button to not be recognized while being pushed and the servo to twitch and not respond at times. The troubleshooting process for this problem required analysis into the electrical engineering of the Arduino pro micro to find the problem and then was resolved in the technological field through extra lines to the code. By adding analog write functions to reset the button pin value every time the program is run, the Arduino verified the switch position value. This switch problem was resolved by using the digital write function. The servo was able to reassign the pin value to the servo rotational status in a way to keep it in the proper orientation and clear the interference that the connection was experiencing. During the process of troubleshooting these issues, the first servo burnt out and had to be replaced. The next issue was found in the code, which would not reset the servo position at times and not react to user input through the button on other occasions. This was found, in the troubleshooting process, to be the fault of using an if-else

statement to run the servo. To remedy this issue, more technical aspects were applied through a double “if” statement, which was used to run individual loops for the function of opening and closing the servo direction. Following this, the 3d model was designed and printed for the housing of the project, which went through several designs before being functional. The first design had an error in which the gear sockets were printed too far apart to prevent the gears from interacting. This was fixed by bringing the gears 5mm closer to each other. After fixing that, it was found that the design did not have enough room for the electronics needed and more importantly could not provide the servo with a position to get enough leverage to move the chopsticks into open and closed positions. Eventually, the model was scrapped and a new model was designed based on the flexibility of the 3d printed plastic rather than a gear system that solved the electronic housing and servo positioning issues but required more plastic welding to be assembled.

Having built a functioning project that completes the engineering requirements set at the beginning of this project, there are several things that could be done differently or better upon repeating this project. The project would greatly benefit from precise measurements with better software to require less post-processing in the assembly. Additionally, adding rails to guide the chopsticks and downsizing the flexible joint for the chopsticks would improve performance. More time could be spent making a better grip for the model would improve the comfort of the device. As far as electronics are concerned, a custom circuit board with power and grounding rails, rather than the current clips, would help with downsizing the project. In the future a single ATTiny chip could also be used to decrease size and power consumption alongside the custom PCB. The solder joints could be improved with more flux in the joints that could increase their potential longevity and efficiency. Alongside that, adding a circuit to reset the servo position before cutting the power, after hitting the power switch, would reduce manual maintenance. Silicon coating on the electronics would make them waterproof to alleviate the concern when using this device for eating near liquids. The code for this project isn't as efficient as it could be and could use cleaning. This could also be solved by using a different chip, such as the ATTiny, which would not require a solution to fix interference. The code could also use some changes to make the servo quieter. The battery would benefit from the use of a step-up circuit to provide power at a higher voltage to the servo and decrease the motor noise.

This experiment provides a solution for individuals with disabilities, who struggle to feed themselves without assistance, by automating processes that are not within their capabilities and necessary to be independent. This device automates the coordination and strength required to precisely close and hold onto an object using chopsticks to feed themselves. By simplifying these actions to the control of a button press, people are able to focus on carrying food to their mouth without requiring the assistance that would be necessary to perform this action otherwise.

References

Source 1

Banzy, M., & Shiloh, M. (2014). *Make: Getting Started with Arduino*. Maker Media, Inc.

Source 2

Ecker, M., Nieto, N., Käbitz, S., Schmalstieg, J., Blanke, H., Warnecke, A., & Sauer, D. U. (2013, October 11). Calendar and cycle life study of Li(NiMnCo)O₂-based 18650 lithium-ion batteries. Retrieved from

<https://www.sciencedirect.com/science/article/abs/pii/S0378775313016510>.

Source 3

Minns, P. D. (2013). *C programming for the Pc the Mac and the Arduino microcontroller system*. Bloomington, IN: AuthorHouse.

Source 4

SajTom Light Future. (n.d.). Is 3D printing safe? Analysis of the thermal treatment of thermoplastics: ABS, PLA, PET, and nylon. Retrieved from <https://www.tandfonline.com/doi/abs/10.1080/15459624.2017.1285489>.

Source 5

nnn946@gmail.com. "TPU Filament 3D Printing Material – The Complete Guide (Mar. 2020)." *AllThat3D*, AllThat3D, 25 Oct. 2019, www.allthat3d.com/tpu-filament/.

Source 6

Warren, J.-D., Adams, J., & Molle, H. (1970, January 1). Arduino for Robotics. Retrieved from https://link.springer.com/chapter/10.1007/978-1-4302-3184-4_2.

Source 7

My Continuous Rotation Servo Won't Stop Spinning?, 27 Feb. 2015, forum.arduino.cc/index.php?topic=304146.0.

Source 8

"How to Make My Servo Stop Twitching." *How to Make a Servo Stop Twitching*, 26 June 2018, forum.arduino.cc/index.php?topic=555150.0.

Source 9

"Language Reference." *Arduino Reference*, www.arduino.cc/reference/en/.