

## User and Product Manual Instructions

This document is a template of a user and product manual. The client may wish to make improvements on the prototype or need to fix it if something goes wrong or another group of students may work to make a more rugged prototype. The document needs to be clear for someone else who is not an engineer **to use, maintain or reproduce the project**. Include as many images and diagrams as possible for a better understanding. Keep it plain, simple, visual and logical.

In general, if you are not sure exactly what to include, imagine that this document was the only thing that you had. Imagine also that your job was to add a new feature or recreate the project that is described in your document. What would you need to know?

Only include details relating to your final prototype.

Template conventions:

- Remove all **red text**, it is only there to guide you
- Remove this page (instructions)
- Replace all instances of <xxx> with the appropriate information for your group, for example you could replace <System Name (Acronym)> by The Amazing Product (TAP)
- Save this document as 'User and Product Manual\_group number' instead of Deliverable X so that others know what it represents when they see it in MakeRepo

GNG2101  
**Design Project User and Product Manual**

**Snack Arm**

Submitted by:

Snack Swivel Solutions, Team B33

Aidan Charette, 300305703

Omar Negm, 300309396

Phillip Gauthier, 300299738

Sebastian Mazhari, 300296152

2023-12-10

University of Ottawa

# Table of Contents

---

- Table of Contents ..... ii
- List of Figures ..... iv
- List of Tables ..... v
- List of Acronyms and Glossary ..... vi
- 1 Introduction ..... 1
- 2 Overview ..... 3
  - 2.1 Conventions ..... 7
  - 2.2 Cautions & Warnings ..... 7
- 3 Getting started ..... 8
  - 3.1 Set-up Considerations ..... 8
  - 3.2 User Access Considerations ..... 8
  - 3.3 Accessing the System ..... 9
  - 3.4 System Organization & Navigation ..... 9
  - 3.5 Exiting the System ..... 9
- 4 Using the System ..... 10
  - 4.1 <Given Function/Feature> ..... 4
    - 4.1.1 <Given Sub-Function/Sub-Feature> ..... 4
- 5 Troubleshooting & Support ..... 12
  - 5.1 Error Messages or Behaviors ..... 12
  - 5.2 Special Considerations ..... 12
  - 5.3 Maintenance ..... 12
  - 5.4 Support ..... 12

6	Product Documentation .....	13
6.1	<Subsystem 1 of prototype> .....	13
6.1.1	BOM (Bill of Materials) .....	13
6.1.2	Equipment list .....	13
6.1.3	Instructions.....	13
6.2	Testing & Validation.....	17
7	Conclusions and Recommendations for Future Work .....	18
8	Bibliography .....	9
	APPENDICES .....	19
	APPENDIX I: Design Files .....	19
	APPENDIX II: Other Appendices .....	11

## List of Figures

---

Figure 1. Picture 1 of the Final Prototype of the Snack Arm .....	4
Figure 2. Picture 2 of the Final Prototype of the Snack Arm .....	4
Figure 3. Block Diagram of the Major Functions of the Product .....	5
Figure 4. Mechanical Drawing and Bill of Materials .....	6

List of Tables

Table 1. Acronyms .....	vi
Table 2. Glossary .....	vi
Table 3. Assumption Regarding the Product .....	1
Table 4. Structure of the Document Starting from the Introduction .....	2
Table 5. Security and Safety Considerations .....	2
Table 6. Key Features and Major Functions of the Product .....	4
Table 7. Main Components	
Table 8. Referenced Documents .....	10

# List of Acronyms and Glossary

---

Provide a list of acronyms and associated literal translations used within the document. List the acronyms in alphabetical order using a tabular format as depicted below.

**Table 1. Acronyms**

Acronym	Definition
BHC	Bethany’s Health Center

Provide clear and concise definitions for terms used in this document that may be unfamiliar to readers of the document. Terms are to be listed in alphabetical order.

**Table 2. Glossary**

Term	Acronym	Definition
Integrated Development Environment	IDE	Software application that helps in developing software code.

# 1 Introduction

Explain the basic context for your work and any assumptions that you have made for your work. Give an overview of the structure of your document (i.e. explain how it is organized) and summarize the purpose of the document, the scope of activities and the intended audience for the document. Also describe any security/safety or privacy considerations associated with the use of the User and Product Manual.

This User and Product Manual (UPM) provides the information necessary for customers to effectively use the Snack Arm and for prototype documentation.

This product was made in collaboration with Bethany's Health Center. Cathryn from BHC communicated with the University of Ottawa to have students work on a robotic arm system for the health center. We were honored to have the opportunity to work with BHC on this project.

The purpose of the product is to lift snacks and drinks from a lower to a higher surface with minimal input. The Bethany's Health Center was unable to find such a product on the market. It is crucial for the robotic arm system to function with minimal input since BHC wishes to have its patients control the Snack Arm.

The health center serves patients with significant motor deficits. Their objective is to increase the level of independence of the patients. The health center serves snacks and drinks at a pick-up window and would like for patients to help with serving the snacks and drinks. This product was made to serve snacks and drinks with minimal input.

Assumptions made regarding the product include (but are not limited to) those in Table 3.

**Table 3. Assumption Regarding the Product**

Assumptions
Lifespan of two years
Free of water or corrosive substances
Assume circuit will not fry
Assume regulators will not be defective
Correct positioning when installed

The structure of the document is described in the table of contents. The structure of the document starting from the introduction is described below (see Table 4).

**Table 4. Structure of the Document Starting from the Introduction**

<b>Structure</b>
Overview
Getting Started
Using the System
Troubleshooting
Product Documentation
Conclusions and Recommendations for Future Work
Bibliography
Appendices

The purpose of this document is to provide the necessary information for customers to effectively use the Snack Arm. This document also provides documentation on the product. This User and Product Manual is intended for use by the customers.

Security/safety considerations associated with the use of the product include (but are not limited to) those inserted in Table 5.

**Table 5. Security and Safety Considerations**

<b>Security/safety considerations</b>
Keep hands away from all moving parts
Read the material safety data sheet for WD40
Keep hands away from electrical components
Beware of motor overheating after extended use

## 2 Overview

Explain the problem and why it is important.

Explain the fundamental needs of the user.

Explain what differentiates your product from others or the key aspects that make your product better.

[add pictures of your final prototype]

Explain the key features or major functions of the product without going into much detail.

Explain the architecture/construction of the system in non-technical terms (metal frame, microcontroller, web-based, etc.), the user access mode (unfold frame, GUI, button, etc.) and any special conditions.

A block diagram is a useful thing to include here too.

This product was made with made in collaboration with Bethany's Health Center. The product serves snacks and drinks at a pick-up window. No product currently on the market suits the needs of BHC. It is crucial for the robotic arm system to function with minimal input so that patients at the BHC may control the Snack Arm.

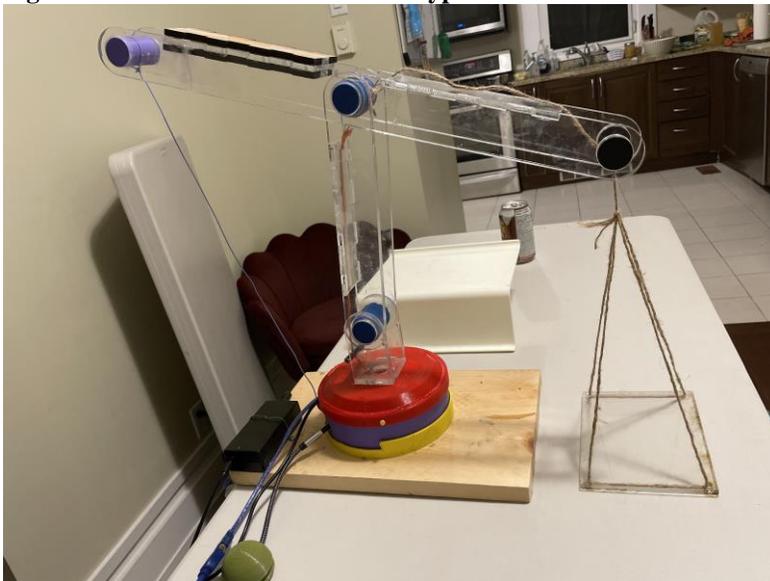
The Bethany's Health Center serves patients with physical impairments. The objective of the health center is to maximize every child's potential and increase their level of independence. Bethany's Health Center serves snacks at a pick-up window and hopes that the robotic arm system can provide patients the opportunity to help with serving the snacks and drinks.

Here are pictures of our final prototype.

**Figure 1. Picture 1 of the Final Prototype of the Snack Arm**



**Figure 2. Picture 2 of the Final Prototype of the Snack Arm**



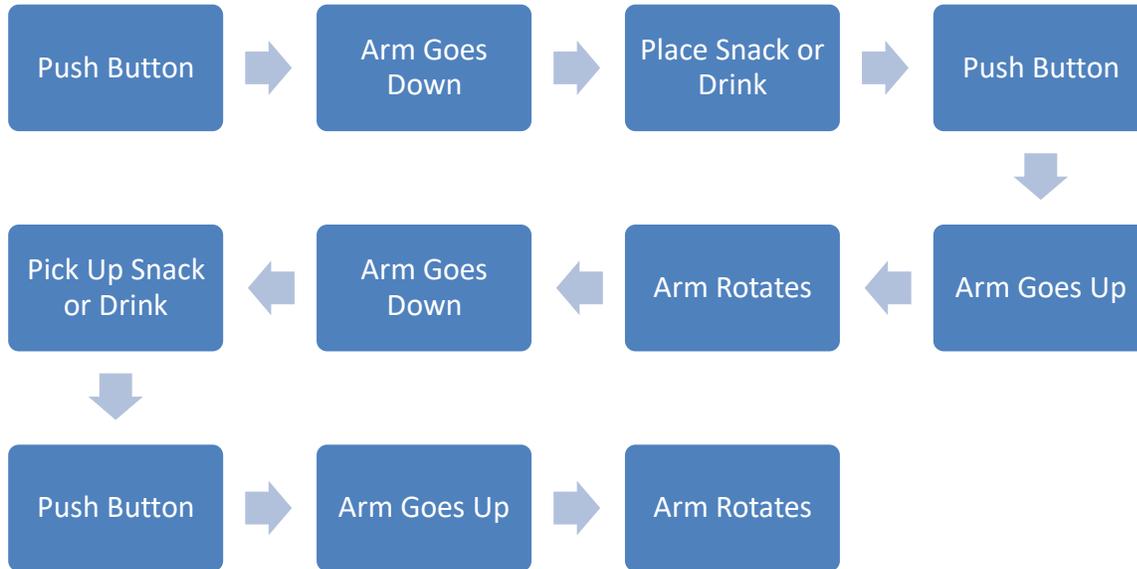
Key features and major functions of the product are summarized in Table 6.

**Table 6: Key Features and Major Functions of the Product**

Key Features	Major Functions
Control Button	Height Adjustment
Serving Platform	Swivel Motion
Electrical Cord	
Data Transmission Cable	

The major functions of the product are also described in this block diagram.

**Figure 3. Block Diagram of the Major Functions of the Product**



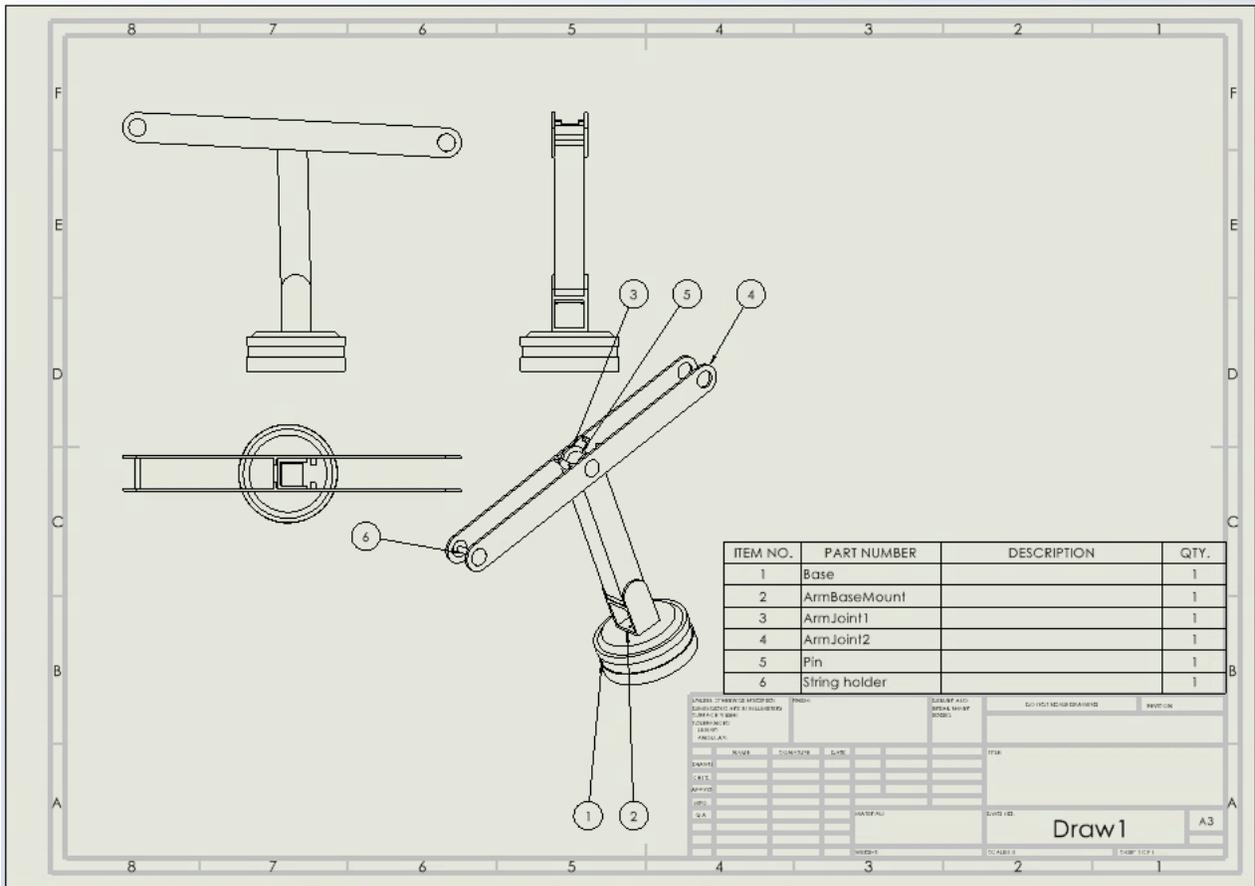
The product was conceptualized to resemble the arm of a human. The main components making the arm are include in Table 7.

**Table 7. Main Components of Snack Arm**

Main Components
Stepper Motor Drivers
Arduino Uno
Arm Base Mount
Arm Joint 1
Arm Joint 2
Bottom of Base
CNC Shield
Data Transmission Cable
Gear
Height Ajustment Cable
Middle of Base
Platform
Power Cable
Stepper Motors
String Holder
Top of Base

Please see Figure 4 below for a visual description of the assembly.

**Figure 4. Mechanical Drawing and Bill of Materials**



The arm joints and the base is made of acrylic. The pin, string holder, and arm base mount is 3D printed. The parts are fastened using epoxy, screws, and with other fastening methods.

## 2.1 Conventions

No stylistic and command syntax conventions were used within this document. When an action is required on the part of the reader, it is indicated as a verb.

## 2.2 Cautions & Warnings

Cautions and warnings are included in the introduction in Table 5. Please consider reading the material safety data sheet for WD40 which is not included in this user manual, but can be easily accessed on the internet or by other means.

### **3 Getting started**

In case the product requires troubleshooting, please consider asking for help from someone knowledgeable in coding software. The product is pre-assembled when purchased. Users must follow simple instructions when setting up the product.

1. Position the product on a flat surface.
2. Verify that nothing will stop the product from moving. Please remember that the product rotates counterclockwise.
3. Plug the data transmission cable to your computer.
4. Press the control button.

The product should be functioning. Please refer to the troubleshooting instructions in case the product does not function. The user can start operating the product. Instruction on operating the product are represented visually in overview (see Figure 3).

1. Push the control button
2. Arm moves downwards
3. Position the snack
4. Push the control button
5. Arm moves upwards
6. Arm rotates counterclockwise
7. Arm moves downwards
8. Remove the snack
9. Push the control button
10. Arm moves upwards
11. Arm rotates clockwise

#### **3.1 Configuration Considerations**

There are no specific configuration considerations regarding the robotic arm system besides those expressed in Getting Started (see page 9).

#### **3.2 User Access Considerations**

This product was made for individuals with significant motor deficits. Please note that someone must assist an individual with significant motor deficits in using this product. The robotic arm system functions when the control button is pressed. An individual using this product should have the ability to press on the control button.

### 3.3 Accessing/setting up the System

Please note that instructions on setting up the system were previously elaborated. The product is purchased already pre-assembled. Simple instructions must be followed when setting up the product. Users must follow simple instructions when setting up the product.

1. Position the product on a flat surface.
2. Verify that nothing will stop the product from moving. Please remember that the product rotates counterclockwise.
3. Plug the data transmission cable to your computer.
4. Press the control button.

The product should be functioning. If the product does not function, please refer to the troubleshooting instructions.

The software code for the robotic arm system is fully adjustable. Users can adjust any feature of the code for personal use. The easiest features that can be adjusted are the number of steps per revolution and the number of milliseconds between steps of the stepper motors.

1. Open the software code for the robotic arm system. Instructions on accessing the software code are provided in troubleshooting and support.
2. Modify the integers for the steps per revolution and the milliseconds between steps (`const int stepsPerRev = 500;`, `int millisBtwnSteps = 10000;`).

### 3.4 System Organization & Navigation

The main components of the robotic arm system are listed in Overview (see Table 7). Please see Figure 4 in Overview for a visual representation of the main features. The main components include (but are not limited to) stepper motor drivers, Arduino Uno, arm base mount, arm joint 1, arm joint 2, bottom of base, CNC Shield, data transmission cable, gear, height adjustment cable, middle of base, platform, power cable, stepper motors, string holder, top of base.

The materials used for the fabrication of the product include acrylic and 3D printing plastic. The arm joints and the base support are made of acrylic, while the pin, the string holder, and the arm base mount are 3D printed. The parts are fastened using various fastening methods, including epoxy, and screws.

### 3.5 Exiting the System

To properly exit the system, unplug the data transmission cable from your computer, and unplug the electrical cable from the electrical outlet. Position the robotic arm so that it does not fall.

## 4 Using the System

### Step 1: Initial Setup

**Placement:** Position the robotic snack arm in its designated operating place, ensuring it has ample space to move freely.

**Winch Adjustment:** Manually crank the winch to its highest height before the use of the robot. This ensures the arm starts from the optimal position for safe and efficient operation.

### Step 2: Power On and Interface Setup

**Power On:** Activate the power source for the robotic snack arm, ensuring all connections are secure.

**Interface Configuration:** If applicable, set up the control interface. Connect the mono auxiliary jack for swappable buttons if the client prefers their own controls.

### Step 3: Loading Snacks

**Initiate Loading:** Press the designated button to initiate the loading sequence. Observe as the robotic arm gracefully descends.

**Load Tray:** While the arm is in the lowered position, carefully load the tray with the desired snacks.

**Secure Snacks:** Ensure the snacks are securely placed on the tray, and the tray is positioned correctly.

#### Step 4: Snack Delivery

**Initiate Delivery:** Press the button for snack delivery. Observe as the arm moves with precision to the pick-up window.

**Stable Snack Pickup:** The robotic arm ensures stability during snack pickup, accommodating the diverse motor deficits of the children.

#### Step 5: Return to Starting Position

**Complete Delivery:** Once the snacks are retrieved, press the button again to initiate the return sequence.

**Raise Arm:** Watch as the robotic arm gracefully ascends back to its starting position, ready for the next interaction.

#### Step 6: Power Off and Secure

**Power Off:** When not in use, power off the robotic snack arm.

**Secure Components:** If applicable, secure any loose components, ensuring the arm is in a safe and stable condition for future use.

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the <System Name and/or Acronym>.

## **5 Troubleshooting & Support**

### **5.1 Error Messages or Behaviors**

In the event of any issues or irregularities, refer to the serial monitor for detailed information. If you encounter problems, please contact our support team with the provided information for prompt assistance. By actively utilizing the serial monitor for real-time feedback, users can ensure that each step of the process is executed correctly and promptly identify and address any potential issues through contact with the support team.

### **5.2 Special Considerations**

Because of the strength of the stepper motor, the arm is prone to skipping steps, the code is programmed to mildly overshoot its target to compensate. Feel free to manually adjust the code or the physical body of the arm when needed.

### **5.3 Maintenance**

Take care to avoid the overheating of the motors by not subjecting the arm too more than 2 hours of continuous use. Also unplug the motors when they are not in use. It is also encouraged to semi frequently apply oil to the gear system.

### **5.4 Support**

For support feel free to contact Sebastian Mazhari ([smazh057@uottawa.ca](mailto:smazh057@uottawa.ca)) or Aidan Charette ([achar270@uottawa.ca](mailto:achar270@uottawa.ca)) in case of any help with modification or potential repairs on the design or if any of the files do not work or provide access.

## 6 Product Documentation

### 6.1 <Arm Subsystem>

#### 6.1.1 BOM (Bill of Materials)

Item	Part	Part N(s) <sup>o</sup>	Price (\$)
Acrylic Sheet	Arm Middle (x2)	1	14.00
	Arm Side (x2)	2	
	Joint Middle	3	
	Joint Side (x2)	4	
	Hinge Side (x2)	5	
	Hinge Bottom	6	
Epoxy Cement	N/A	N/A	11.25
3D Printed Pieces	Pin (x2)	7	0.00
	String Holder (x2)	8	

#### 6.1.2 Equipment list

- 3D Printer
- Laser cutter

#### 6.1.3 Instructions

1. Secure the 2 joint side pieces with 2 pin pieces on either end.

2. Add the joint middle piece to the joint side pieces they should fit in the grooves at a 90-degree angle, further secure the joints together with epoxy cement.
3. Attach the 2 arm side pieces to both sides of the pin on one end of the joint
4. Epoxy cement the 2 arm sides with the 2 arm middles on either side of the arm
5. Attach the hinge sides to the remaining pin and epoxy the hinge bottom to those pieces.
6. Add the string holder pieces to both ends of the arm.

## 6.2 <Base Subsystem>

### 6.2.1 BOM (Bill of Materials)

Item	Part(s)	Part N(s) <sup>o</sup>	Price (\$)
3D Printed Pieces	Base Part 1	9	0.00
	Base Part 2	10	
	Base Part 3	11	
	Spur Gear	12	
	Base Pin	13	
Ball Bearings	Balls	14	9.80

## 6.2.2 Equipment List

- 3D printer
- Oil of choice

## 6.2.3 Instructions

1. There should be a circular groove with no gear teeth in the bottom base plate, it is for the ball bearings to be inserted, you can put oil in this groove or around the gear to help the arm move.
2. Make sure the “x” motor is through the middle plate and the spur gear is at the end of the “x” motor's shaft.
3. Place the middle plate on the bottom base plate
4. Cover with the top plate
5. To put the arm on the top plate it is recommended to use pins, screws or bolts, we used string and glue because of time and budget concerns.

## 6.3 <Electronic Subsystem>

### 6.3.1 BOM (Bill of Materials)

Item	Part	Part N(s) <sup>o</sup>	Price (\$)
Control Board	Arduino Uno	15	15.25
Jumper cables	(x2)	16	0.20
3D Printed Pieces	String Pulley	17	0.00

	Spur Gear	12	
	Pulley axle	18	
Motors	Nema 17 (x2)	19	21.99
CNC Shield	CNC Shield Addon	20	17.87
CNC Shield addons	Actuator circuit (x4)	21	Comes with item above
String	N/A	22	0.80
IDE	Arduino IDE	N/A	0.00
Power Supply	20V laptop Charger	23	N/A

### 6.3.2 Equipment List

- 3D Printer
- Wire Cutters
- Soldering iron
- Lighter

### 6.3.3 Instructions

1. Connect the Arduino Uno to your computer and upload the code.
2. Connect the CNC shield to the Arduino and the actuators to the CNC shield.
3. Using the cords provided with the motors, connect the motors to the actuators
4. To supply power, find a 20V AC to DC adapter such as a laptop charger and cut off the part that would go into your computer before any bulges in the wire is applicable.

5. Pull out the red and black wires and clamp them to the CNC shield red to positive and black to negative.
6. Attach a button circuit to the Arduino via the CNC shield via pin \_\_\_\_ and pin \_\_\_\_.  
Some soldering, may be needed.
7. Put the electronic components in the base
8. Wrap and tie the string around the winching motor
9. Put a spur gear at the end of the winch motor
10. Insert the pulley to the pulley axle and “weld” the axle to the base so that the pulley can support the string coming out from the hole in the base.
11. “Weld” using plastic filament and a lighter

## **6.4 Testing & Validation**

Some recommended tests to do would be to test the motors before they are put in the assembly to make sure the x and y motors are not mixed up. Another good test is to put some weight on the assembled arm pieces to make sure they can withstand the weight and not fall apart.

## 7 Conclusions and Recommendations for Future Work

### 7.1 Conclusion

Overall, the arm works but can be finicky and needs a little more development that we were not able to fully complete due to time and budget constraints. We recommend trying what works for you as the design and idea behind it is relatively simple.

### 7.2 Recommendations

Some suggested improvements on the initial design:

Hole in top of base for cooling of motors and/or stronger motors which would need a larger case

Consider using different bodies depending on the application:

X	MDF	ACRYLIC	ALUMINUM
Density (mass)	Least Dense	Medium Dense	Most Dense
Ductility	Medium Ductile	Least Ductile	Most Ductile
Price	Least Expensive	Medium Expensive	Most Expensive
Material Required	Most Material	Medium Material	Least Material
Strength	Least Strength	Medium Strength	Most Strength

# APPENDICES

## 8 APPENDIX I: Design Files

Makerepo: <https://makerepo.com/AidanCharette/1695.gng2101-b33-g3ntds-snack-arm>

Code:

```
const int enPin = 8;

const int stepXPin = 2; // X.STEP
const int dirXPin = 5; // X.DIR
const int stepYPin = 3; // Y.STEP
const int dirYPin = 6; // Y.DIR
const int buttonPin = 9;

bool firstMoveCompleted = false;
bool secondMoveCompleted = false;

const int stepsPerRev = 500;
int pulseWidthMicros = 4000; // microseconds
int millisBtwnSteps = 10000;

enum State
{
    IDLE,
    MOVE_DOWN,
```

```

    MOVE_UP,
    CLOCKWISE,
    COUNTER_CLOCKWISE
};

State currentState = IDLE;
bool buttonPressed = false;

void setup()
{
    Serial.begin(9600);
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(enPin, OUTPUT);
    digitalWrite(enPin, LOW);
    pinMode(stepYPin, OUTPUT);
    pinMode(dirYPin, OUTPUT);
    pinMode(stepXPin, OUTPUT);
    pinMode(dirXPin, OUTPUT);
    Serial.println("CNC Shield Initialized");
}

void loop()
{
    int buttonState = digitalRead(buttonPin);

```

```

if (buttonState == LOW && !buttonPressed)
{
    buttonPressed = true; // Button is pressed

    switch (currentState)
    {
    case IDLE:
        currentState = MOVE_DOWN;
        break;

    case MOVE_DOWN:
        Serial.println(F("Running Winch Down"));
        digitalWrite(dirXPin, LOW);
        for (int i = 0; i < 1500; i++) //change number up and down to adjust distance
        {
            digitalWrite(stepXPin, HIGH);
            delayMicroseconds(1000);
            digitalWrite(stepXPin, LOW);
            delayMicroseconds(2000);
        }
        firstMoveCompleted = true;
        currentState = MOVE_UP;
        break;

```

```

case MOVE_UP:
    Serial.println(F("Running Winch Up"));
    digitalWrite(dirXPin, HIGH);
    for (int i = 0; i < 1500; i++) //change number up and down to adjust distance
    {
        digitalWrite(stepXPin, HIGH);
        delayMicroseconds(pulseWidthMicros);
        digitalWrite(stepXPin, LOW);
        delayMicroseconds(millisBtwnSteps);
    }
    delay(1000);
    Serial.println(F("Running clockwise"));
    digitalWrite(dirYPin, LOW);
    for (int i = 0; i < 250; i++) //change number up and down to adjust distance
    {
        digitalWrite(stepYPin, HIGH);
        delayMicroseconds(pulseWidthMicros);
        digitalWrite(stepYPin, LOW);
        delayMicroseconds(millisBtwnSteps);
    }
    delay(1000);
    Serial.println(F("Running Winch Down"));
    digitalWrite(dirXPin, LOW);

```

```

for (int i = 0; i < 500; i++) //change number up and down to adjust distance
{
    digitalWrite(stepXPin, HIGH);
    delayMicroseconds(pulseWidthMicros);
    digitalWrite(stepXPin, LOW);
    delayMicroseconds(millisBtwnSteps);
}

delay(1000);

secondMoveCompleted = true;

currentState = COUNTER_CLOCKWISE;

break;

case COUNTER_CLOCKWISE:

    Serial.println(F("Running Winch Up"));

    digitalWrite(dirXPin, HIGH);

    for (int i = 0; i < 500; i++) //change number up and down to adjust distance
    {
        digitalWrite(stepXPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(stepXPin, LOW);
        delayMicroseconds(2000);
    }

    delay(1000);

    Serial.println(F("Running counter-clockwise"));

```

```
digitalWrite(dirYPin, HIGH);
for (int i = 0; i < 250; i++) //change number up and down to adjust distance
{
    digitalWrite(stepYPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepYPin, LOW);
    delayMicroseconds(6000);
}
delay(1000);
currentState = IDLE;
break;
}
}
else if (buttonState == HIGH)
{
    buttonPressed = false; // Button is released
}
}
```