Salisbury University

Gripper Final Report

Ethan Gray

December 15th, 2021

I Abstract

On a global scale, the worlds ecosystem is suffering from the excessive amount of plastic waste. With tons being produce yearly, the clean up process becomes very grueling. The goal of the Gripper project is to demonstrate a tool to aid in this issue. By attaching it to a drone, it can be used to pick up such wastes remotely. The project is built around the usage of an Arduino Uno controlling a servo motor and Bluetooth module. The Gripper is comprised of three main parts, the box to hold the hardware, the attached leg with a perpendicular bar, and the unattached leg that is drawn across the bar to act as the gripper mechanism. Here in this paper, the design process and final build are showcased.

II Introduction

As years pass by, our population exponentially grows with waste production following the same pattern. For several years now, the global production of plastic wastes have been about 300 million tonnes per year [2]. This amount has lead to ecosystems worldwide to become polluted and bringing harm to them. The intention of the Gripper project is to aid in the clean up process within such areas. The concept of the project is to attach it to the bottom of a drone and then via Bluetooth, remotely grab an item. This can help aid in clean up by begin able to utilize it in hard to reach places, such as in inaccessible waters or higher up areas. The Gripper can also be intending for other various reasons.

III Processes and Methods

Coming into the Fall of 2021 semester, the original plan was to continue work on a previous project. But after the first two weeks, it was decided to move onto another project due to the complexity of it. After some discussion and brainstorming with my mentor Dr. Giulia Franchi, we came up with the design of the Gripper. In *figure 1* is the original concept drawing of the Gripper to it's final concept drawing.

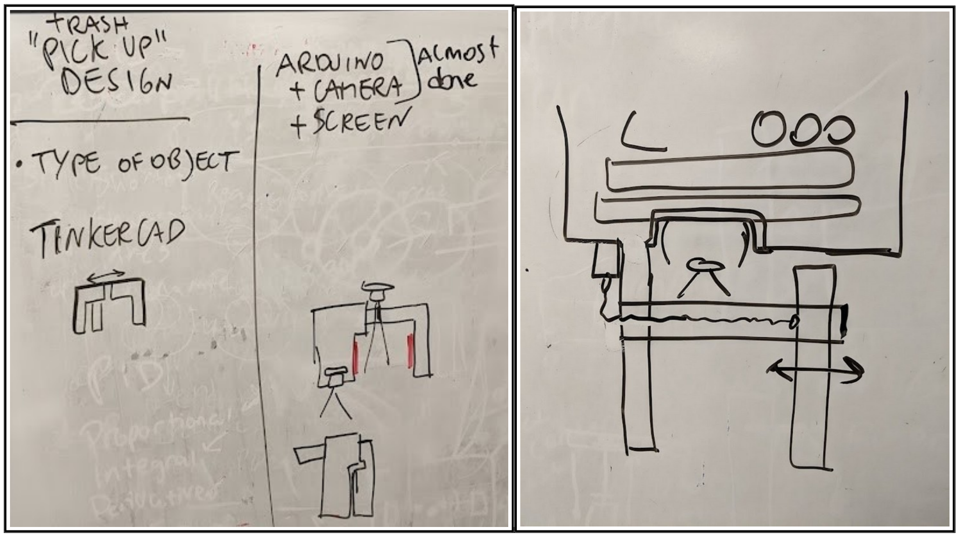


Figure 1: Concept drawings of the Gripper

The final drawing was the idea taken into the build of the project. The plan for the Gripper was to design a box to hold the electronical components, that pulls an unattached leg across a bar that is then connected to another leg which is attached to the box. Inside the box, an Arduino Uno will control a continuous servo motor that pulls the string that is connected to the unattached leg of the Gripper. The Arduino acts accordingly through the use of a Bluetooth module that receive signals from a phone application. The application is designed with three signals in mind, one for pulling, stopping, and reversing the string. The circuit design can be found in *figure 2*.

The following list is all the hardware and software that are used in the Gripper:

- *Arduino Uno Board (An open source microcontroller board)*

- *HC-05 Wireless Bluetooth RF Transceiver*
- *Continuous Rotation Micro Servo – FS90R*
- *9 Volt Battery with power clip*
- *MIT App Inventor*
- *Arduino IDE (Uploads C/C++ code to Arduino boards)*
- *Tinkercad (Free online 3D Modeling)*
- *Makerbot (3D Printer)*
- *DJI Phantom 2 Drone*
- *Rubber Bands and Hot Glue*

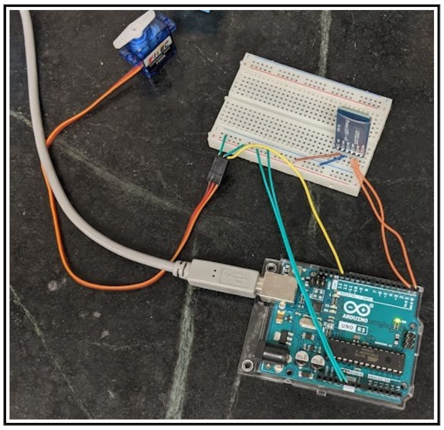


Figure 2: Circuit build of Gripper

The steps following the circuit design were to construct and 3D print the actual Gripper itself. Using Tinkercad, the dimensions of the Arduino board were taken into account when creating the box, while make room for the wiring. There was also holes placed in the box, one for the servo to pop out of behind the connected leg, one for the battery clip to run out, and a last one for plugging the battery into the board. For the legs, one was designed with a bar connected perpendicular to it. While an unattached leg had a hole cut out to fit the bar through. Finally, a lid to the box was made in order to attach the Gripper to the drone. In *figure 3* are images of the Tinkercad model of the Gripper.

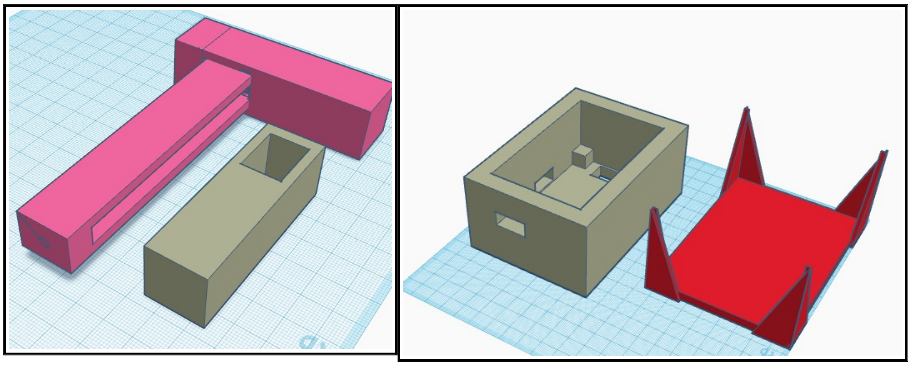
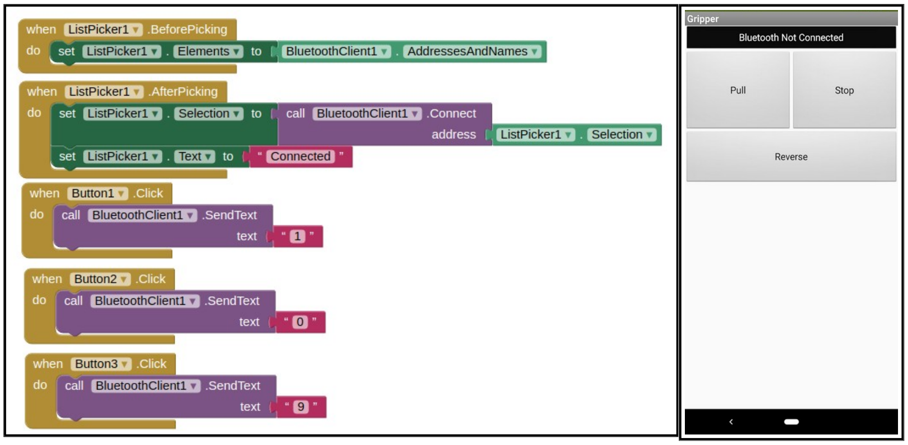


Figure 3: Tinkercad 3D design of Gripper

To create the design, the files were sent to Salisbury University's Makerlab and printed on a Makerbot. There were a few faulty prints with measurements being slightly off until the final print. During this print process, as a single print could take up to 8 hrs, the code and phone application was made and tested. The code that was uploaded to the Arduino Uno can be found in the *Appendix* of this report. The concept of the code is that it runs in a continuous loop of waiting for a signal from the Bluetooth module. Once the signal for pulling is received, the code is prompted into another loop of pulling until it receives the stop signal. The same process occurs for the reverse option. The signals are sent from a phone application created by using the *MIT App Inventor* [1]. The images in *figure 4* are of the created application, alongside it's block coding.



```
when ListPicker1 .BeforePicking
do   set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do   set ListPicker1 . Selection to call BluetoothClient1 .Connect
                                                    address ListPicker1 . Selection
     set ListPicker1 . Text to " Connected "

when Button1 .Click
do   call BluetoothClient1 .SendText
                            text " 1 "

when Button2 .Click
do   call BluetoothClient1 .SendText
                            text " 0 "

when Button3 .Click
do   call BluetoothClient1 .SendText
                            text " 9 "
```

Figure 4: MIT App Inventor – Gripper

Within the created app, the user must press “Bluetooth Not Connected” to select from a list of previously connected devices. The HC-05 module must have been connected beforehand in order to pair. Once the device is paired, the user can now press the buttons to control the servo motor accordingly. Once the testing of the software and the final prints finished, the mounting process could then begin.

The build process began with drilling the necessary holes in order to connect the pieces of the attached leg to the box and the lid to both the drone and the box itself. In order to fit the hardware inside the box, the Arduino board was attached to the under part of the lid and had the power for the components go into a small solderless breadboard. This means some soldering was needed in order to shorten the wires so that they could fit inside the box. For the servo motor, it was drilled into the hole made for it on the bottom of the box. From there, a string was tied to the motors head piece and ran through a hole of the attached leg and tied to a screw on the unattached leg. Fi*gure 5* is the final and current build of the Gripper project, along with a link of a video of it in action. A few extra things included were the rubber bands around the legs to give them extra grip and a screw near the servo to act as a casting pulley to help wrap the string around the servo. The following link is a video of the Gripper in action: https://photos.app.goo.gl/GeZiLVezvn4FudHZ7

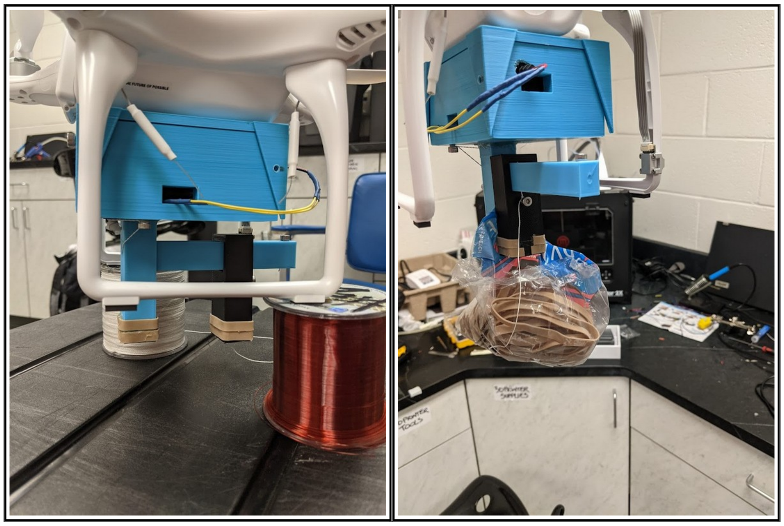


Figure 5: Gripper Project

IV Future Works

For future works and improvements of the Gripper, it would be best to include a camera detection system. That way, the need for manual control is no longer needed. Another inclusion would be to have another servo pull the unattached leg in reverse to open the Gripper back up. Alongside that, other methods of grabbing items could be incorporated, like using some sort of claw mechanism. The overall purpose of the project was to demonstrate a methodology that can be included into aiding in the cleanup crisis of plastic wastes world wide and of course, any other various needs.

References

[1] MIT. (n.d.). Ai2.appinventor.mit.edu. Retrieved December 15, 2021, from

http://ai2.appinventor.mit.edu/

[2] Our planet is drowning in plastic pollution. This World Environment Day, it's time for a change. #BeatPlasticPollution This World Environment Day. (2018). Retrieved December 15, 2021, from https://www.unep.org/interactive/beat-plastic-pollution/

Appendix

```
 #include <Servo.h>

char app_value = 0; // Variable for apps response of button clicks

Servo servo1; // Servo Object
int servo1_pin = 13;  // Servos pin number

void setup() {
  Serial.begin(9600);
}

void loop() {
  if(Serial.available() > 0){
    app_value = Serial.read();

    // On Button
    if(app_value == '1'){
      // Loop for pulling the arm across
      // Breaks loop once stop is selected
      servo1.attach(servo1_pin);  // Starts servo
      while(app_value != '0'){
        app_value = Serial.read();
        servo1.write(180);
      }
      servo1.detach(); // Stops servo
    }

    // Reverse
    else if(app_value = '9'){
      // Breaks loop once stop is selected
      servo1.attach(servo1_pin);  // Starts servo
      while(app_value != '0'){
        app_value = Serial.read();
        servo1.write(-180);
      }
      servo1.detach(); // Stops servo
    }
  }
}
```