A 4-finger robotic hand on Arduino platform

Alexis Bustos id: 1359210

Veronika Danello id : 1828966

Overview

Our project is to design a robotic hand with 4 active fingers to imitate the movement of human fingers. Each finger will be able to move independently controlled by the position data taken from flex sensors. There will be 4 servo motors which will be connected to each finger by a braided wire. We will also use flex sensors, the robotic hand, and the Arduino software in order to make the hand act as human ringers. In this class we have learned various ways in which computers can help humanity, and our project is a perfect example. Our project can get developed to be a prosthetic hand and bring robotic functionality to someone.

Due to the complications with COVID 19 we had to adjust our project in order to accomodate not being able to put the flex sensors and the rest of the hardware together. Instead of using the flex sensors to control our robotic hand, we adjusted to use...

Requires Hardware

- Arduino board
- Braided wire
- 4 Servo motors
- 2 mini breadboards
- Jumper wires
- 4 Flex sensors

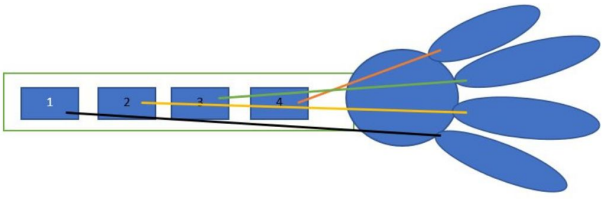
Intermediate Stages

- Stage 1: Build robotic hand with servo motors.
- Stage 2: Build glove with flex sensors.
- Stage 3: Put the 2 pieces together and write the source code.

Stage 1: Ordering and Planning

A majority of our parts had been ordered from China. Unfortunately some of the orders got cancelled which delayed the build process.However, we had received some of the parts to begin building.

We planned to use a plastic grabbing hand as the base for our project. We disassembled the hand in order to make it usable for our purpose. Each finger will be connected to a servo motor by a braided wire. We have included below how we planned to assemble the hand.



We have planned to position 2 servo motors at position 0 degrees and the other 2 at 90 degrees. The first and second servo motors will rotate from 0 to 90 degrees while the third and fourth will rotate from 90 to 0 degrees.

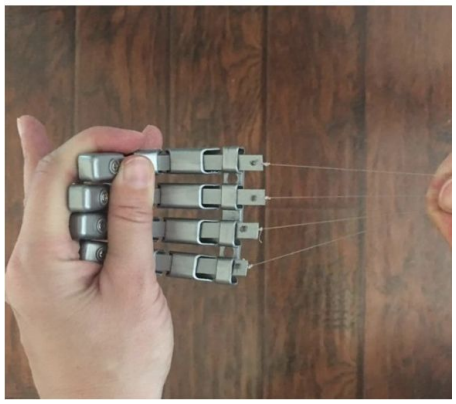
At this stage we also wanted to connect servo motors as follows:

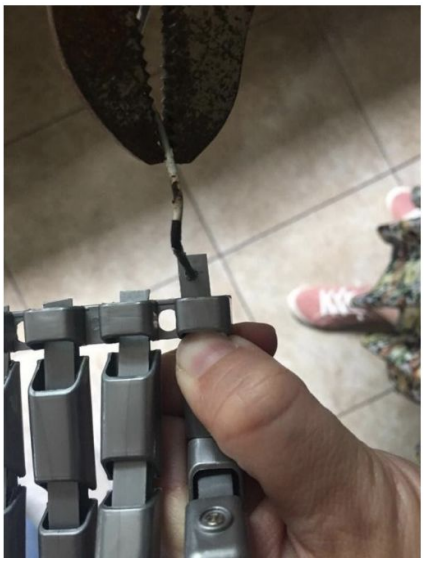
- 1 to digital 3
- 2 to digital 5
- 3 to digital 6
- 4 to digital 9

Finally the flex sensors had arrived. We planned to use these in order to move the plastic arm around. We planned to use these in our program, but ended up not using this idea.

Stage 2: Building

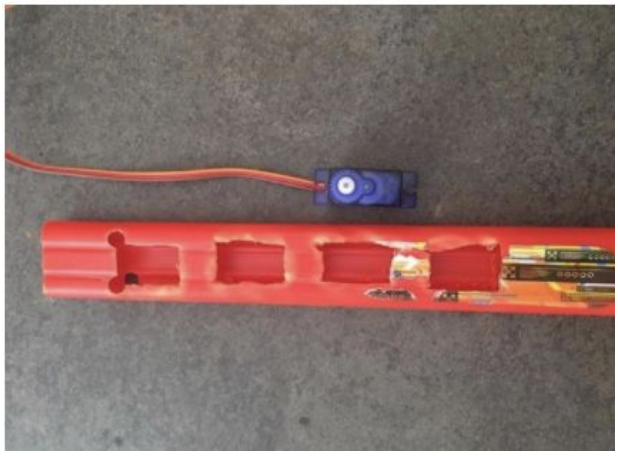
Due to the circumstances we ended up dividing the work into two parts. Veronika was able to assemble the robotic hand. The plastic arm was disassembled into parts. She needed to burn a hole in each finger plate (4 total plates) in order to attach the line.



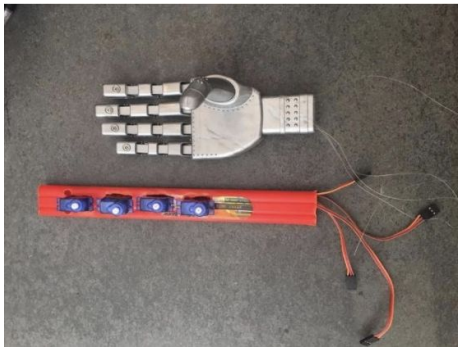


Then the parts needed to get put back together keeping intact the wires.

In the plastic tube, 4 holes needed to get cut out in order to install the motos.

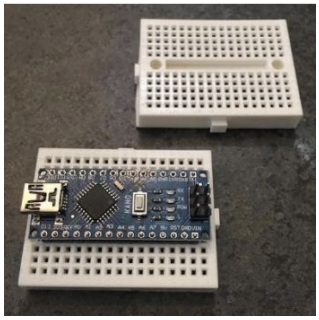


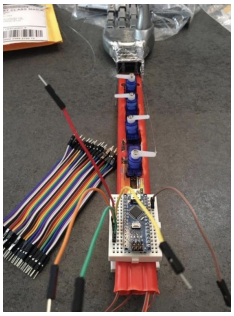
Each motor has an attached cabal that was pushed into the tube.



Next, the connection between the hand and the red tube needed to be made. In order to connect we used duct tape. Each finger was used to connect to each servo motor as we described in phase 1.

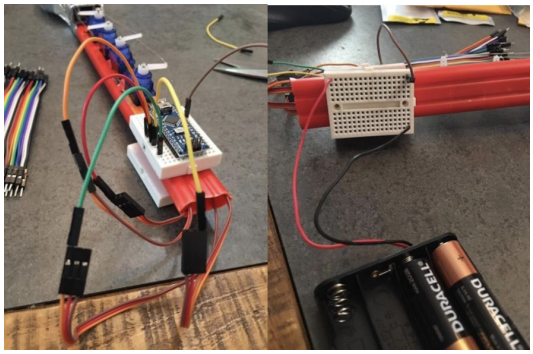
Using the Mini USB Nano V3.0 5V 16M ATmega328P CH340G is Micro-controller board For Arduino is installed on a Mini Solderless Prototype Breadboard for Arduino Shield. Then both parts of the mini breadboard are glued at the end of the plastic tube.





With all major parts assembled, the cables need to get plugged between motors, mini breadboards.The connections again are:

- Servo 1 to digital 3
- Servo 2 to digital 5
- Servo 3 to digital 6
- Servo 4 to digital 9



More about the SG 90 servo motors:

These will be used anytime we need to position something that we do not need to go a full 360 degrees an analog servo motor is a great idea.

Servo motor is a motor whose shaft position can be precisely controlled. It uses an internal servo mechanism and some gearing in order to accomplish this. The servo mechanism provides positional feedbacks, so the motor always knows what position its shaft is currently in. Servos are used in industrial and hobbyist application. Most of hobby servo motors travel range of 180 degrees. Hobbyists servo motors have a 3-pin connector. Pin one of the servo is a ground connection (brown), pin 2 is a power connection(red), pin 3 is a control line, the line that the pulse width modulation signal is sent to the servo motor on(orange).

The speed of the servo motor is specified as the time required to move the shaft by sixty degrees. For example, 0.25 seconds per 60 degrees indicates that the shaft will move 60 degrees a quarter of the second.

Torque is defined as the amount of the force a servo motor Cn apply to a level. It is measured in ounce inches or kilogram centimeters and the two measurements can be converted. As an example, if a servo motor rated at five kilograms centimeters means that one centimeter from the shaft position can support a weight of five kilograms. Two centimeters from the shaft position can support half the weight-2.5 kilograms.

There are a few torque specifications associated with a servo motor. The rated torque is the maximum torque at the servos rated speed. The stall torque is the amount of torque that will cause the motor to stall. The peal torque is the maximum torque for very short time periods. By knowing the speed and torque requirements for the application the correct servo motor can be selected.

As of the flex sensors, they needed to get attached to the glove in order to move the prostetic hand. Using construction paper we glued to sensors on to it and then glued the paper with the sensor to the glove.

Later we decided to cut the flex sensors out and find a way to use the Arduino software in order to program the hand using code.

Stage 3: Programming

Through some research we ended up adopting the Arduino library in order to get our sensors working. Since the motion flexors are not available at this time, we will use the Arduino library to control the motion of the servo motors. We will use example sketches sweep from Arduino IDE. The sweep sketch takes a servo motor and sweeps to shaft from 0 to 180 degrees and back. It is used for loops, so the rotation happens again and again. The program includes the Arduino servo library that's already been included with IDE. Next we are going to create an object myservo (4 fingers = 4 myservo objects). Every servo is associated with a pin that's capable of PWM. The integer pause is defined as 0 - it takes the servo position. The servo motors are going to sweep from 0 to 180 degrees and back. Each object is assigned to the pin on The Arduino. Each object will go into the two for loops the 1st one is going from 0 to 180 with 1 degree as an increment for a step. For every step it writes that value (+=1) over to the servo motors with delays for 15 milliseconds. The second for loop does exactly the opposite. It starts from 180 and drops to 0 with a step of one-degree minus.

Sweep §

```
Servo myservoOne;  // create servo object to control a servo
// twelve servo objects can be created on most boards
Servo myservoTwo;
Servo myservoThree;
Servo myservoFour;


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

void setup() {
  myservoOne .attach(9);  // attaches the servo on pin 9 to the servo object
  myservoTwo.attach(10);
  myservoThree.attach(11);
  myservoFour.attach(12);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservoOne.write(pos);              // tell servo to go to position in variable 'pos'
    myservoTwo.write(pos);
    myservoThree.write(pos);
    myservoFour.write(pos);
    delay(20);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservoOne.write(pos);              // tell servo to go to position in variable 'pos'
    myservoTwo.write(pos);
    myservoThree.write(pos);
    myservoFour.write(pos);
    delay(20);                       // waits 15ms for the servo to reach the position
  }
}
```

```
Done uploading.

Sketch uses 2300 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 61 bytes (2%) of dynamic memory, leaving 1987 bytes for local variables. Maximum is 2048 bytes.
```

Final Product:

Through some research we were able to move forward on how to implement our program to get the robotic hand to move like a real hand. Using a pre-built function in the Arduino software we are able to make the robotic hand functional. You can see in the video provided the final project and the fingers moving as we run the program. Each finger arised to about 20 degrees even though our motors did rotate to 180 degrees.

Improvements for next time:

1) Since we hooked the servo motor up to the 5-volt power supply, this is how much servo motor uses, from the Arduino Uno, the motor tends to put noise on the power supply lines and that affected the digital circuitry. They can also have sudden demands which can ultimately cause a spike which will reset our Arduino or cause it to act erratically. For the future we would use a separate power supply for the servo motor.

2) The movement of the farthest point (fingertip) from servo motor on average 20 degrees with the rotation of servo motor of 180. The longest finger plane is 11.5 cm, the shortest is 8 cm. The thickness of the plane is 1.5 mm. However, the plastic for the finger plane is not really flexible. Since the servo motors pull the bottom of the finger plane, and the fingertip is moved as a result of the plastic plane to be bended, the correct calculation is not available at this time. Potential change would be adjusting the wires we use for better flexibility.

3) Another issue involved the inability of the fingers to return to its initial resting position. The plastic bands around the finger plane are too inelastic and prevent the servos from pulling the fingers in toward the palm with a full force.