Kinetic Origami/Kirigami

For my Kinetic Origami/Kirigami assignment, I combined an origami "mouth" with a servo motor. The servo moves the mouth with bent paper clips that have been wrapped around one of the servo arms on one end and stapled to the top of the mouth on the other end. As a bonus feature, the staples that I used to staple the ends of the paper clips onto the origami mouth somewhat look like eyes. The motor is mounted onto a 3D-printed structure that I drew up in Autodesk Inventor 2012 and printed using one of the Type A 3D printers from Jacobs Hall. The 3D printed structure I printed out has a wedge that the servo motor fits nicely into, and I secured the servo motor even more using packaging tape.

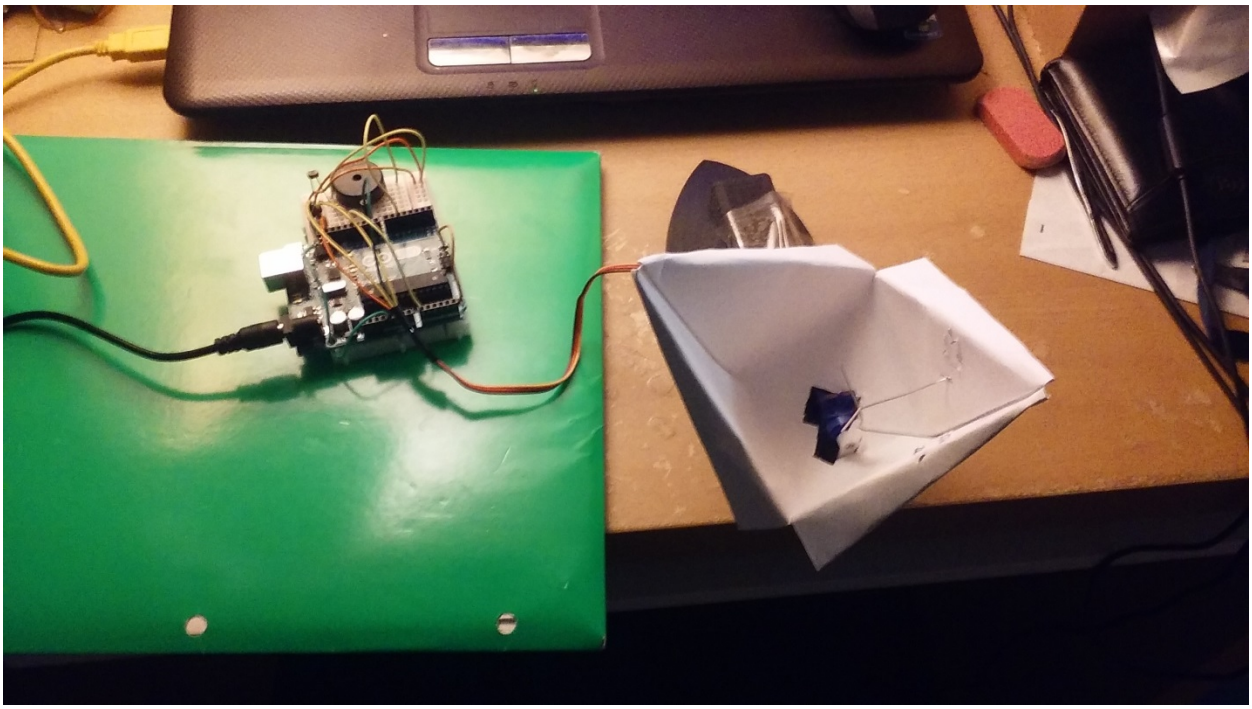
When the motor moves, the mouth opens/closes while twisting. For controlling the motor, I used, of course, an Arduino Uno. Additional components I used included a photoresistor, a 10k ohm resistor, a piezo buzzer, and a few jumper wires. For the Arduino code I wrote, rather than just simply making the servo move using the photoresistor, I did the following:

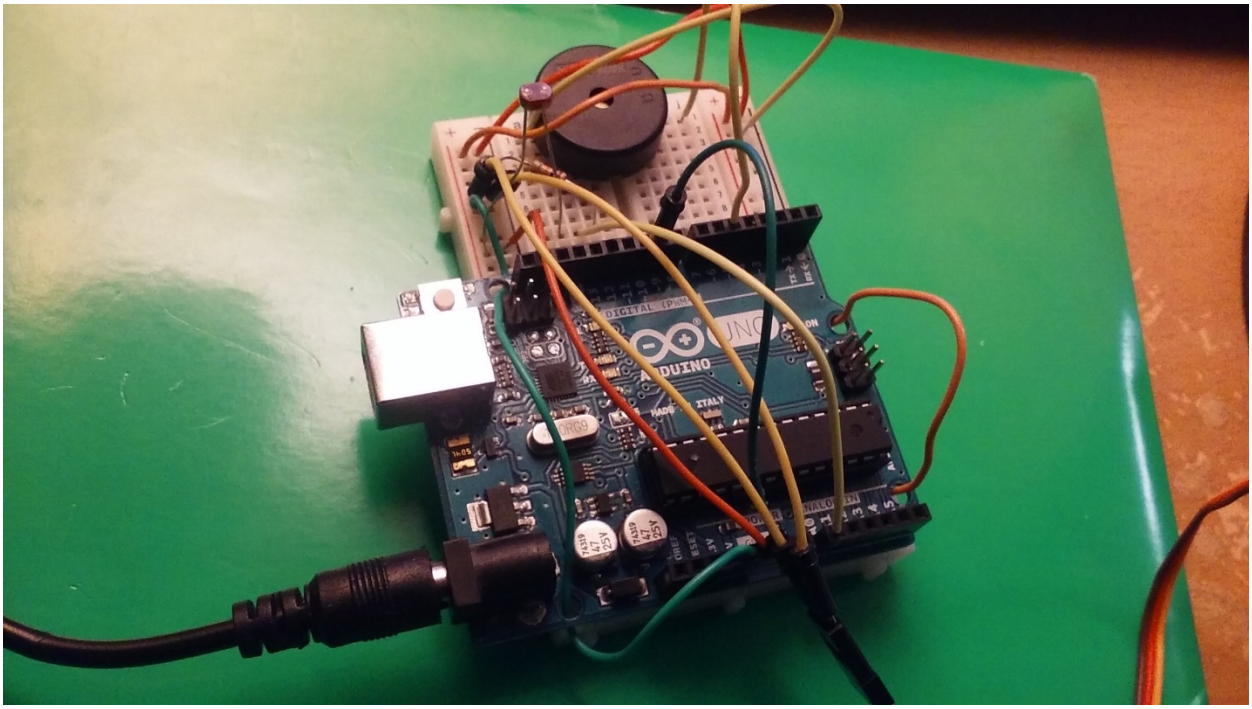
1. Generate a random number from 0 to 4 (not including 4), and if the random number generated is 1, I set a boolean value named **angry** to true. This boolean value the "anger" of my contraption (which I am pretending is a monster).

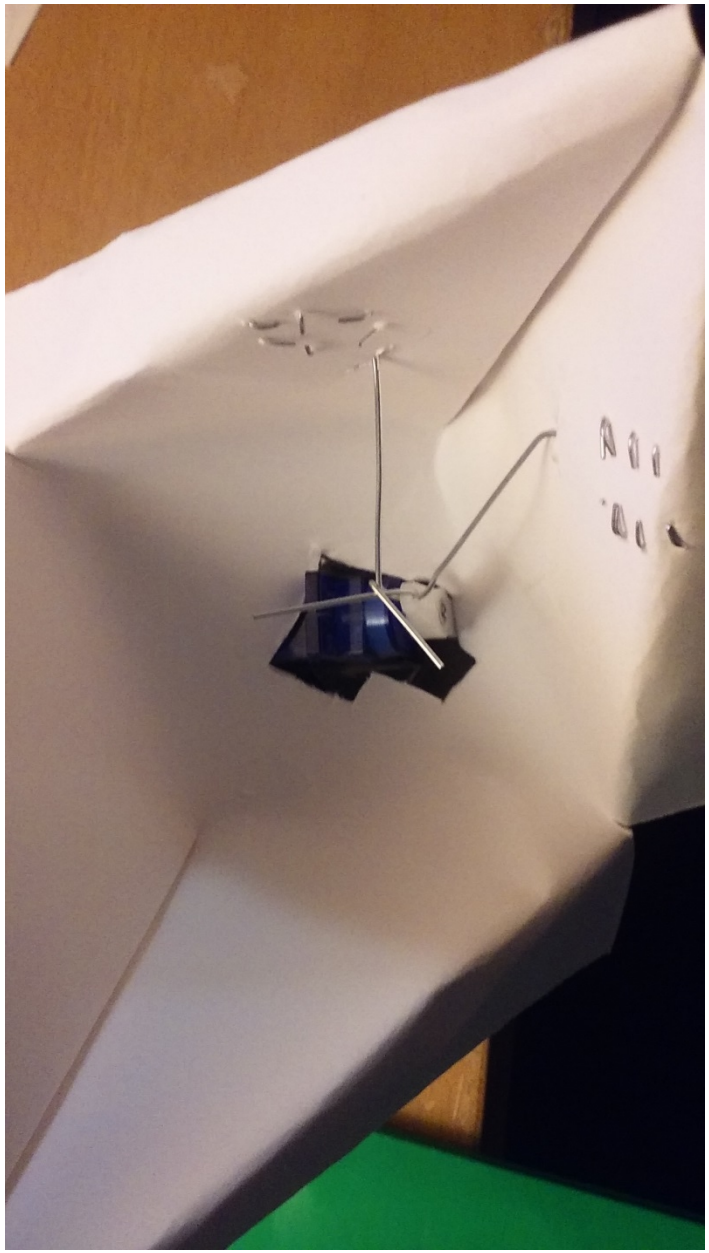
2. As long as the monster is feeling angry, as determined by the value of the **angry** boolean variable, I have the piezo buzzer play random tones (from 500 to 1000 Hz, not including 1000) and the have the motor open and close the mouth.

3. To relieve the monster of its anger, one would just simply need to block enough light going into the photoresistor. Once the photoresistor reads a dark enough value, the angry boolean will be set to false, the random tones and mouth movement will stop, and the monster will wait 5 seconds before attempting to get angry again.

See the next pages for images and the Arduino code.







```
#include <Servo.h>

Servo mouthServo;
const int servoPin = 9;
const int prPin = A0;
const int piezoPin = 5;

bool angry;

void setup()
{
  pinMode(prPin, INPUT);
  pinMode(piezoPin, OUTPUT);
  angry = false;
}

void loop()
{
  if (!angry)
  {
    if (random(0, 4) == 1)
    {
       angry = true;
    }
  }
  if (angry)
  {
   tone(piezoPin, random(500, 1000));
    chomp();
    if (analogRead(prPin) < 700)
    {
      angry = false;
      noTone(piezoPin);
      delay(5000);
    }
  }

}

void chomp()
{
  if (!mouthServo.attached())
  {
   mouthServo.attach(servoPin);
  }
 mouthServo.write(180);
```

```
  delay(1000);
  mouthServo.write(0);
  delay(500);
  mouthServo.detach();
}
```