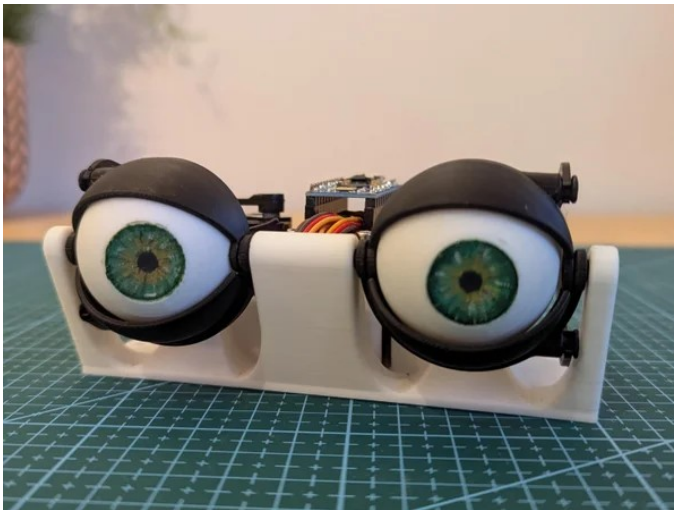
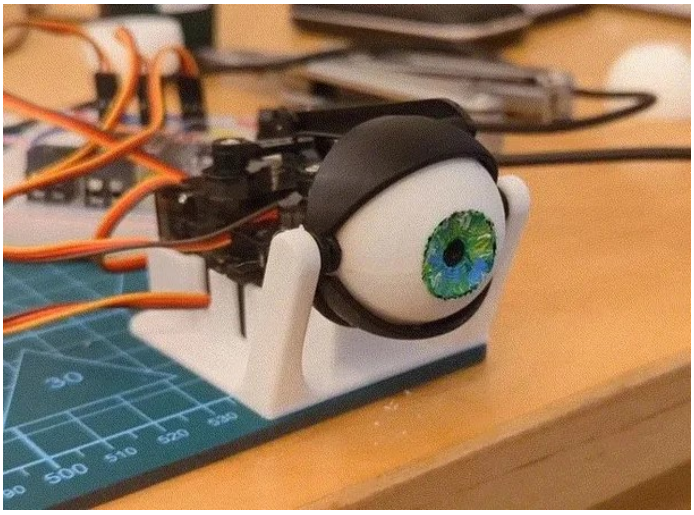
AUTODESK
Instructables

Animatronic Eyes - (Double and Single, Fully 3D Printed, Compact, With Arduino)

By MorganManly in Workshop3D Printing
Unpublished

Introduction: Animatronic Eyes - (Double and Single, Fully 3D Printed, Compact, With Arduino)





Here are the brief (hopefully) instructions on how to print, assemble and program this neat little animatronic eye mechanism I've designed over the last few months, in either single or double format. This project is **suitable for beginners** who already have a little experience in 3D printing and arduino programming. If you're new to arduino, and you want to learn about driving servo motors, this is also a great project to get you started.

This project can be built as either a **single eye** or a **double eye** configuration. The instructions to build both are nearly identical.

Supplies

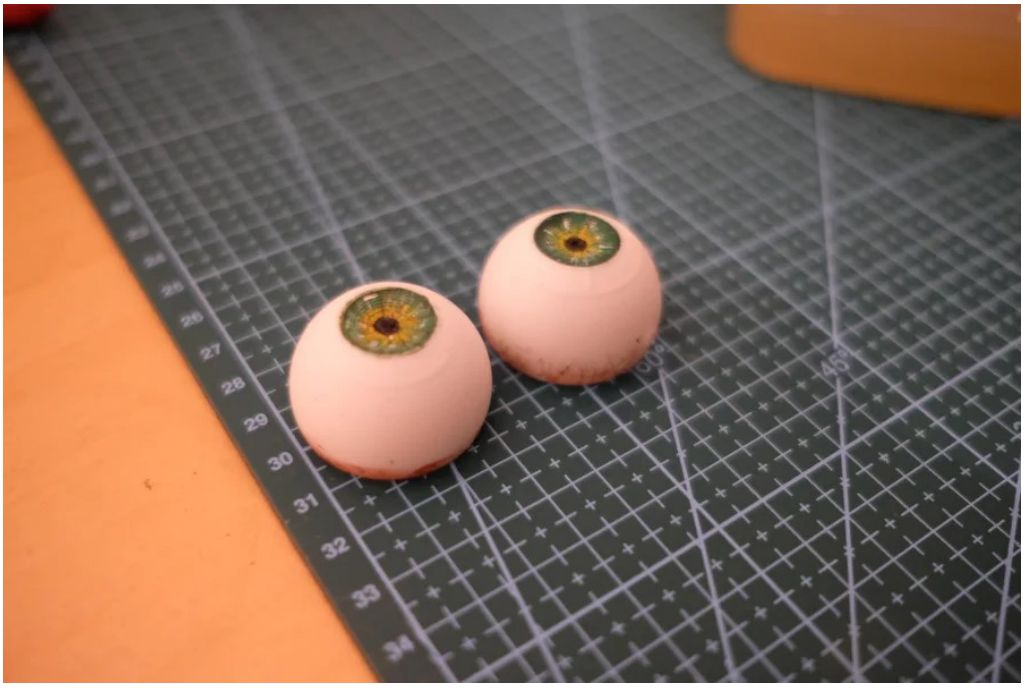
1. **Servo Motors x 3 per Eye**
2. Choose either MG90S or SG90, the SG90s are cheaper and will work but you get what you pay for. I recommend MG90S for this project as they will give you far fewer headaches.
3. **Arduino/Microcontroller x 1**
4. Use any microcontroller that has at least 3 pwm pins per eye. Common boards such as Arduino Uno, Nano, Pro Micro, Mega, ESP32 will all work.
5. For mounting the arduino onto the double eye, I have used an arduino pro micro, due to its footprint.
6. **Power Supply**
7. A **5V** power supply that can put out at least **3A** is recommended, such as this
8. **Wire for Eye Linkages**
9. I have used 1mm thickness steel wire that I found at a craft shop
10. **Breadboard and Jumpers**
11. If you have none of these, a small breadboard kit like this will be useful for this and future projects
12. **M3 Machine Screws x 1 per Eye**
13. Countersunk, 8mm-16mm length will work
14. **M3 Heat-In Brass Threaded Inserts**
15. Such as these
16. **Power Supply Connector**
17. Something to plug the connector into, I have made a circular hole on the base plate for you to mount this

Step 1: Print Print Print

1. For **Single Eye** - Find the Print Files Here
2. For **Double Eye** - Find the Print Files Here

I printed everything in PLA, none of the files need any support material. The trickiest part to print is the eyeball. Cleaning the build plate and slowing down the first layer will improve bed adhesion.

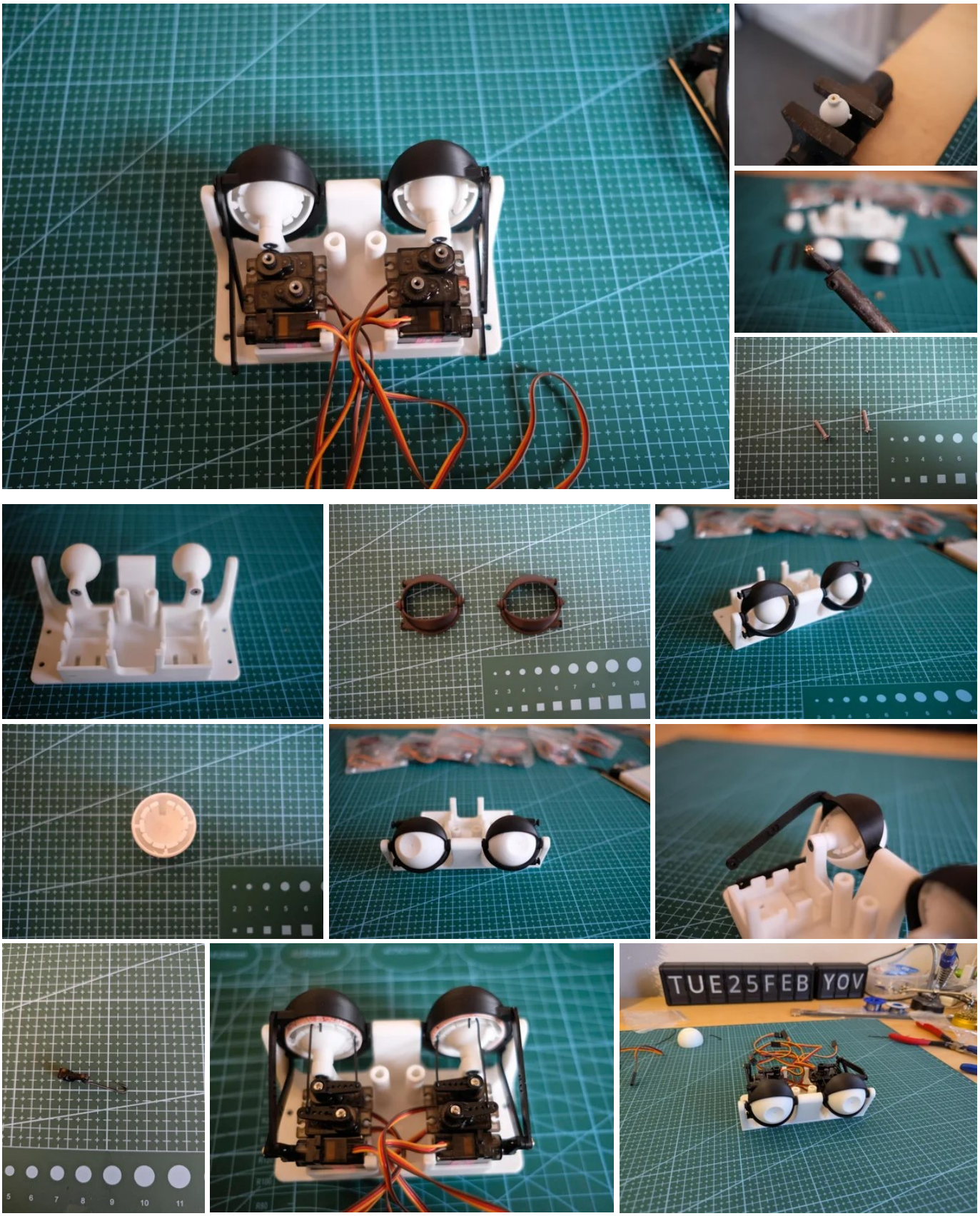
Step 2: Optional - Paint Eyeball



Painting the eyeball is a nice step to make the finished product look much more life-like. I didn't follow any tutorial, but there are some out there which will probably give you a better result than mine.

After painting, I covered the pupil in wood glue, which then dried clear, to give the eye a realistic sheen.

Step 3: Assembly

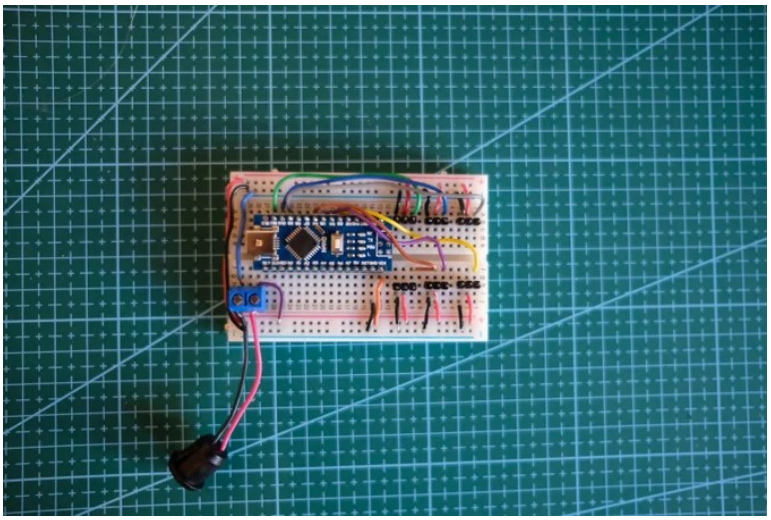


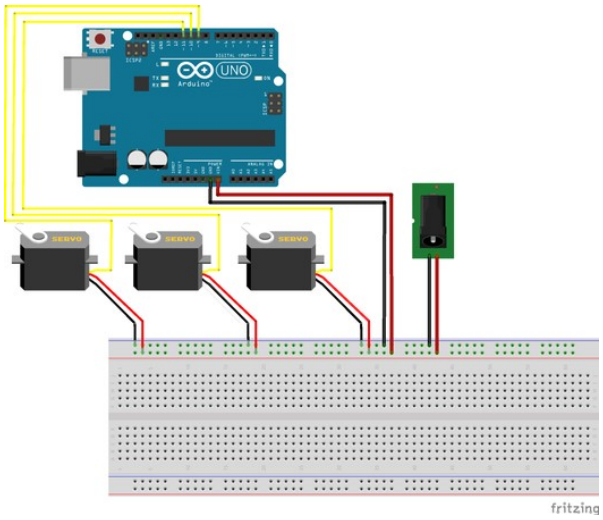
Once all the parts are printed, assembly is straightforward.

1. Use a soldering iron to melt the inserts into the eyeball ball joint, then they can be bolted onto the base plate.
2. Snap fit the eyelids onto the baseplate, the thicker eyelid with the lip is the bottom one
3. Snap the eyelid linkages onto the eyelids, they are labelled so you know where they go

4. Press fit the eyeball onto the eyeball ball joint, there is a correct orientation, if you look at the slow on the eyeball
5. Mount the servos as in the photos, there should be enough friction to hold them in place but you may want to use glue if they are moving around a lot
6. Cut, bend and mount the wire linkages between the motor arms and the eyeball, you may need to remove the eyeball to get the mounting correct. It is important that the eye is pretty much looking straight when the motor arms are at 90 degrees. Use the photo for reference. Do not screw the motor arms in yet because we need to set the motor positions first.

Step 4: Wiring 1





Wiring the servos to the controller is simple because there are very few connections.

1. Each servo requires 5V and GND from the external power supply
2. Each servo has one signal wire which goes to any PWM pin on your controller
3. The controller is powered using the external power supply, which is connected through the VIN pin
4. The GND of the controller is connected to the GND of the power supply.

If you are looking for more resourses on connecting servos, this tutorial may help

Step 5: Initial Setup

Step 1: Set all motors to position of 90 degrees

Before securing the motor arms and running any code, the motor positions must be set to 90 degrees. This ensures that there is enough range of motion in the motor for the eye to move fully.

The code can be found here

1. Install Arduino IDE if you haven't already
2. In the library manager tab on the left hand side, find and install the ServoEasing library
3. Open the calibration.ino file found in the 03_calibration folder of the repository
4. Inside calibration.ino, you will be able to select a pin to move by changing the int servoPin line, make sure this corresponds with the pin of the motor you plan to move
5. **Important! Make sure the eyeball linkage is disconnected from the motor when you upload the code**
6. Upload the code to the controller, then use the serial monitor to write a value of 90 to the motor (type 90 and press enter)
7. Repeat for all the motors in your system
8. You can then secure the motor arms to the motors using the screws provided.

Step 2: Use calibration.ino to record the centre, upper and lower position for each motor

Since all builds will be slightly different, this code lets you find the minimum and maximum value you can safely write to the servo without breaking the eye. Start at values close to 90 degrees, and slowly increase/decrease until the eye mechanism hits it's limit. Record the numbers somewhere for later. Also record the servo command that will completely center the eye in each direction. This process also applies to the eyelids, where you will want to record the servo positions that fully close/open the eye to where you want.

Step 6: Run the Code to Make the Eye Move

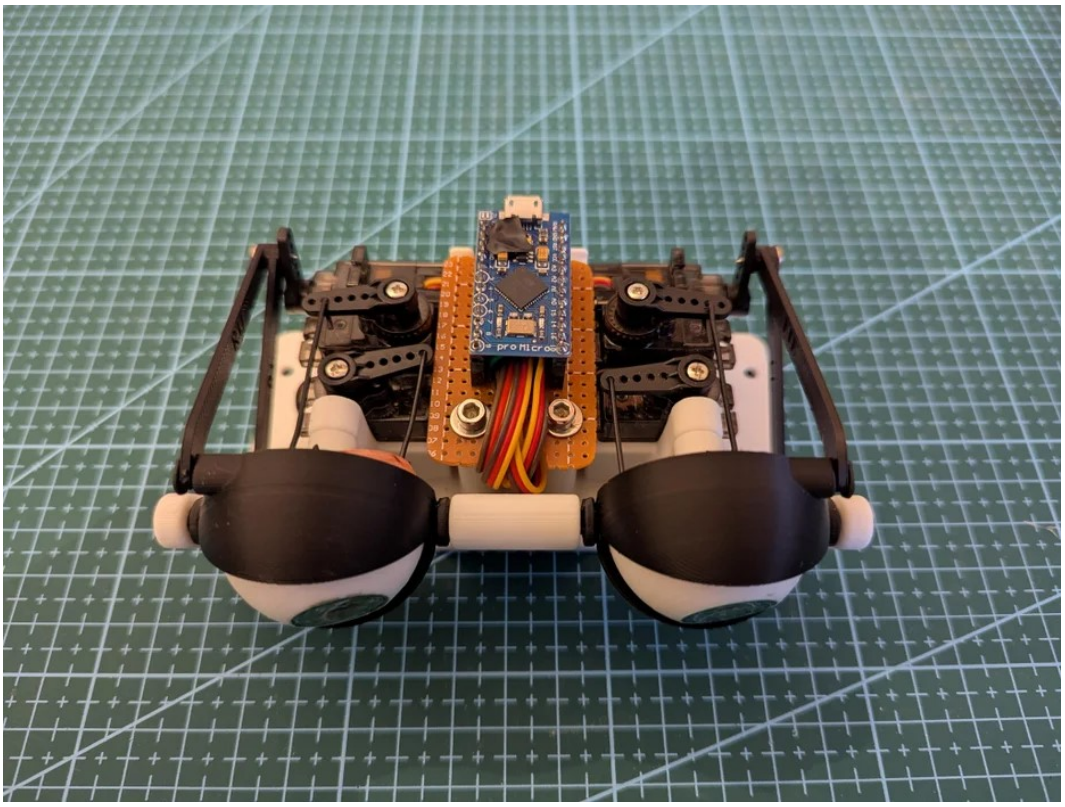
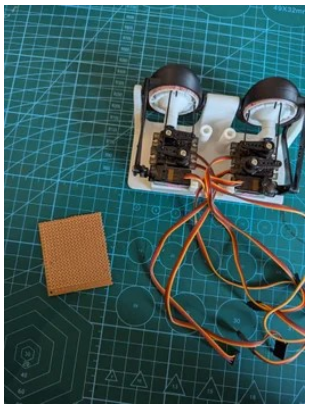
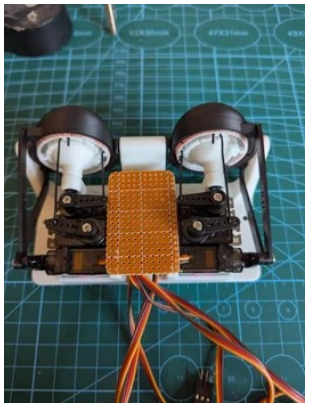
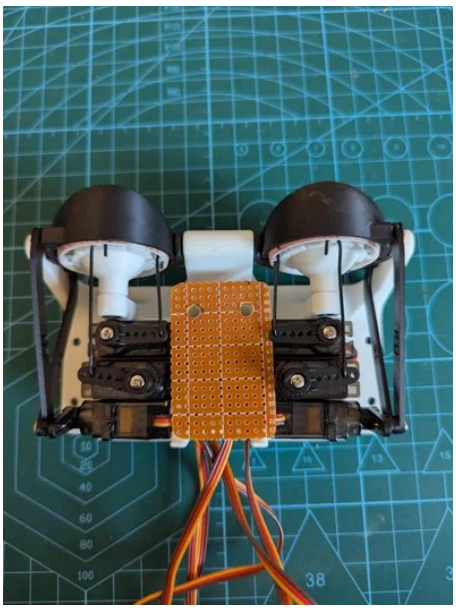
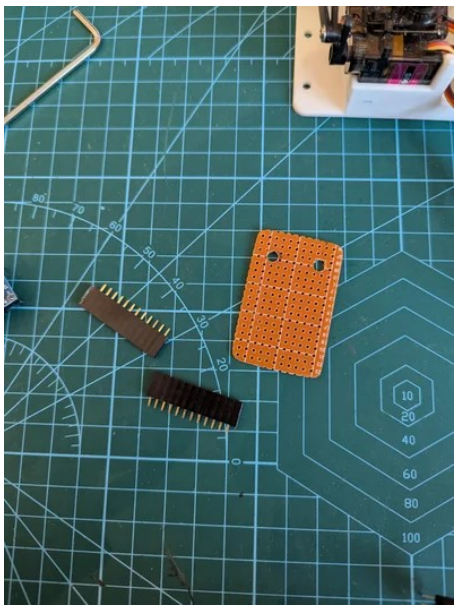
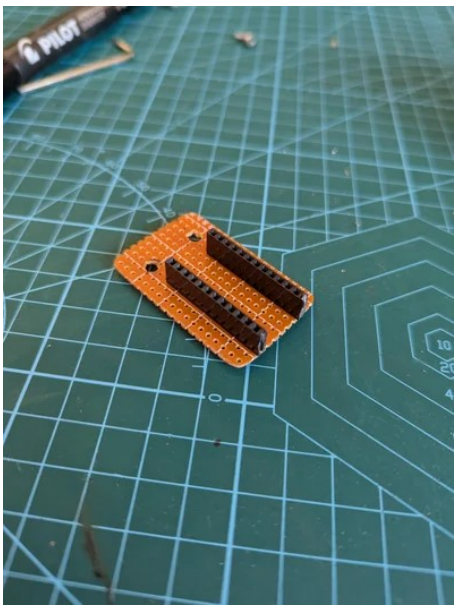
```
#include "Arduino.h"
#include "double_eye.h"

//Servo Connection Pins, Left Eye, then Right Eye (Left Right, Up Down, Open Close)
// The order of left and right is important, and it it taken from the robots perspective (use the linkage labels for reference)
Eyes Eyes(10,9,3,
          14,6,5);

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

  // ENTER YOUR VALUES FOR UPPER/LOWER/CENTRE HERE, USE CALIBRATION.INO To Find the Values
  Eyes.rightEye.setLeftRightLowerUpperCentre(60,115,90);
  Eyes.rightEye.setUpDownLowerUpperCentre(45,120,85);
  Eyes.rightEye.setEyeLidOpenClose(100,35);

  Eyes.leftEye.setLeftRightLowerUpperCentre(60,120,85);
  Eyes.leftEye.setUpDownLowerUpperCentre(35,110,75);
  Eyes.leftEye.setEyeLidOpenClose(75,145);

  //Setup Eyes
  Eyes.init();

  // Centre Eyes
  Eyes.home();
}

void loop() {
  Eyes.eyeMotion(random(200,500));
  someFunction();
}

void someFunction() {
  //Put whatever you want here (non blocking)
}
```

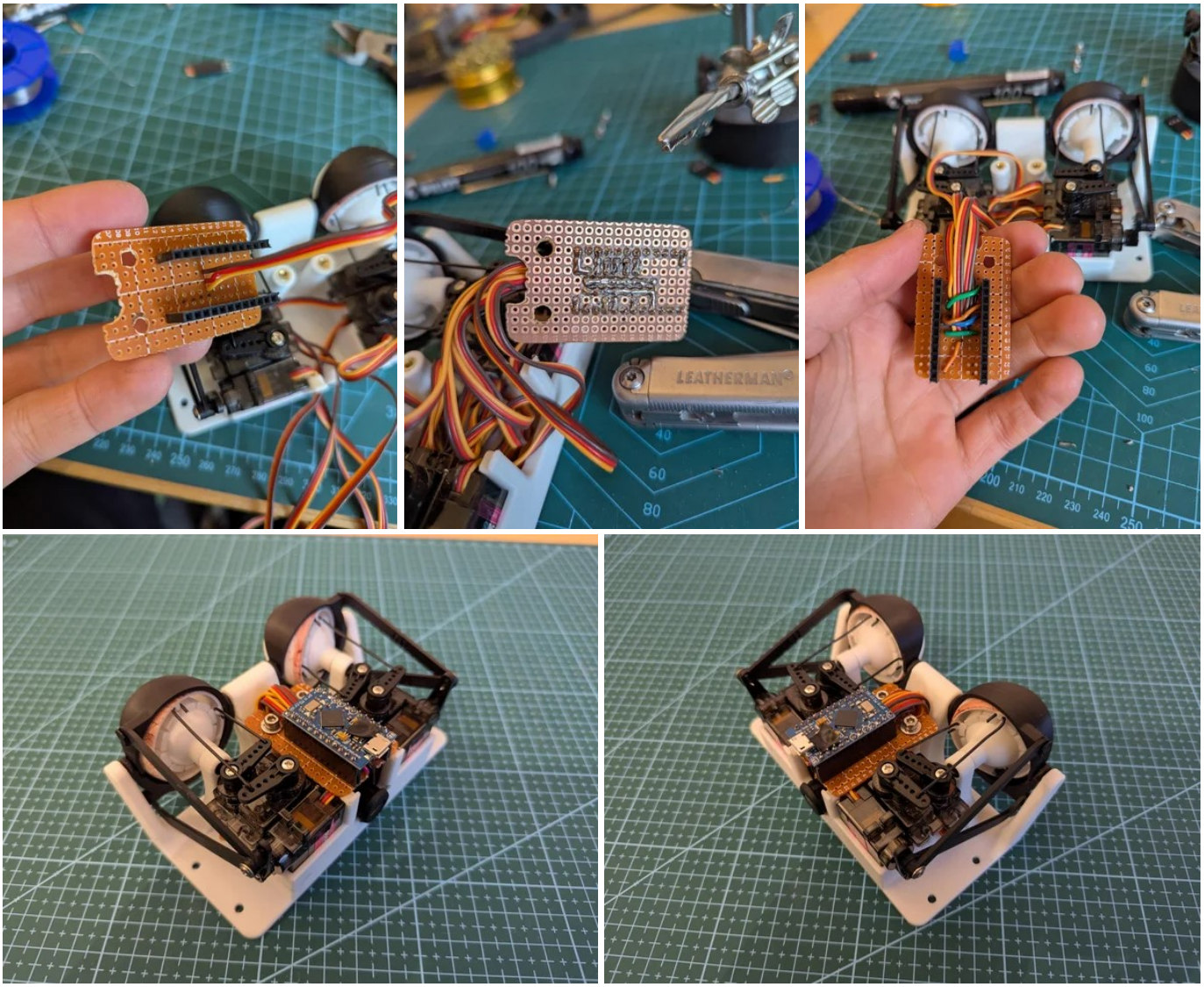

Now that we know the range of motion of the system, we can run the real code to make the eye(s) move

1. Open single_eye_motion.ino or double_eye_motion.ino, depending on your build
2. In the ino tab, change the Eye or Eyes declaration to have the correct pins as in your build
3. Change the values in the set functions to match your lower, upper and centre values from your calibration

From here, you should be able to run the code and watch the magic happen. I would be rather cautious on the first run, just to make sure nothing breaks.

Step 7: Optional - Package Into Enclosure

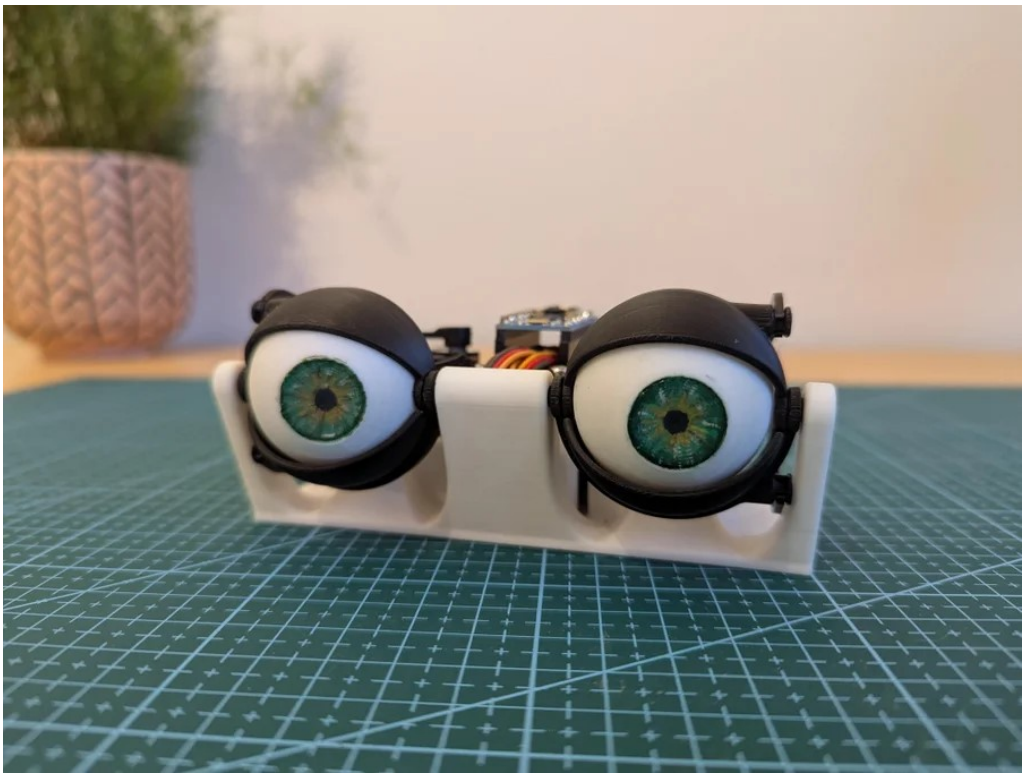
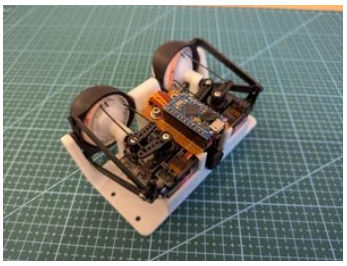







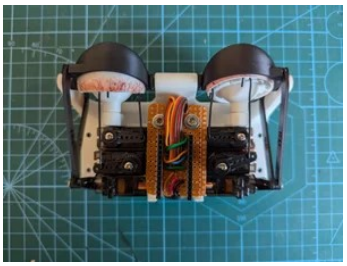


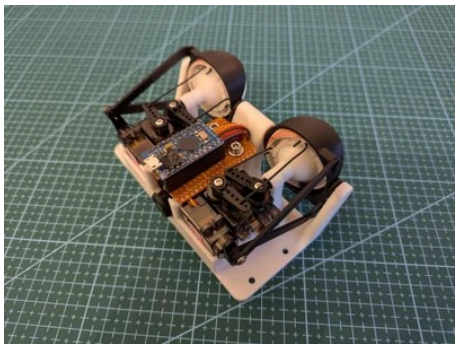
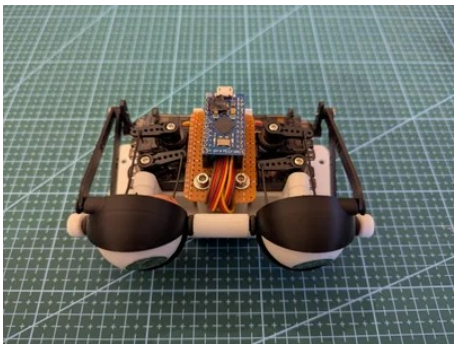
If you want to go all the way and package the arduino into the baseplate, I have added features to mount a piece of protoboard. You can place heat-in inserts into the two pillars to mount the board. An arduino pro micro was the perfect footprint for this as I can solder the servo connections between the header pins.

There is also a hole on the back for mounting a panel mount connector for the power supply

Step 8: Enjoy



I hope you enjoyed this project, thanks for taking the time to read it :)