

Bots Breaking Barriers

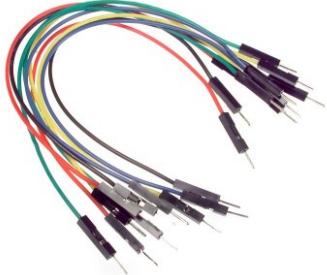
Team 3: Art Bot

By: Christine Liceralde, Liam Bell, Namrata Rana

Art Bot Description:

This DIY art robot is capable of being programmed to draw anything one's heart desires. It is currently adapted to use a standard marker (i.e. Crayola) but can be adjusted for use with other mediums in varying size diameters. Program it to draw shapes, write words, draw scenes, colour, and if you're up for the challenge, take a spin at programming the art bot to create spirographs!

Parts, Materials, Equipment:

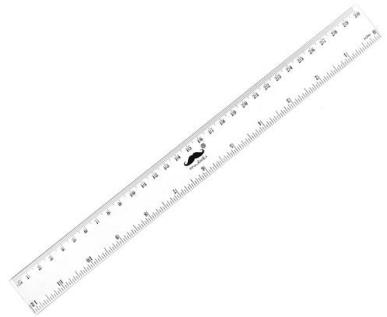
		
Arduino UNO	Jumper wires x	Motor/Wheel Combo x4
		
Breadboard	9V Battery	Tape



Marker

Foam Core Sheet

Exacto Knife/ Box Cutter



Glue

Pencil

30cm Ruler

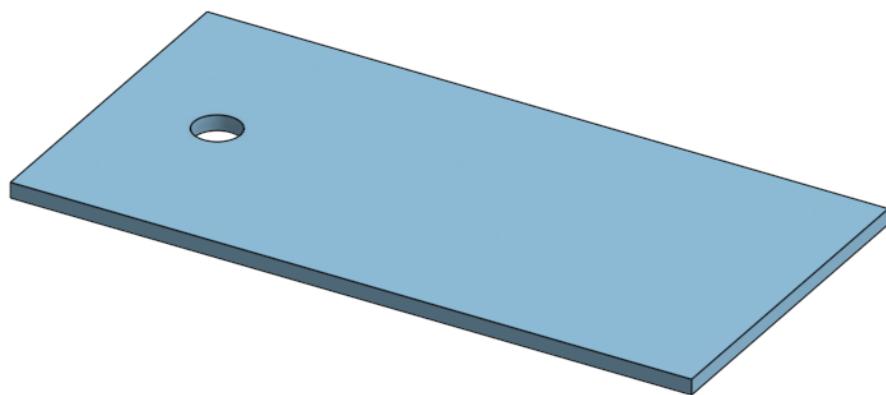
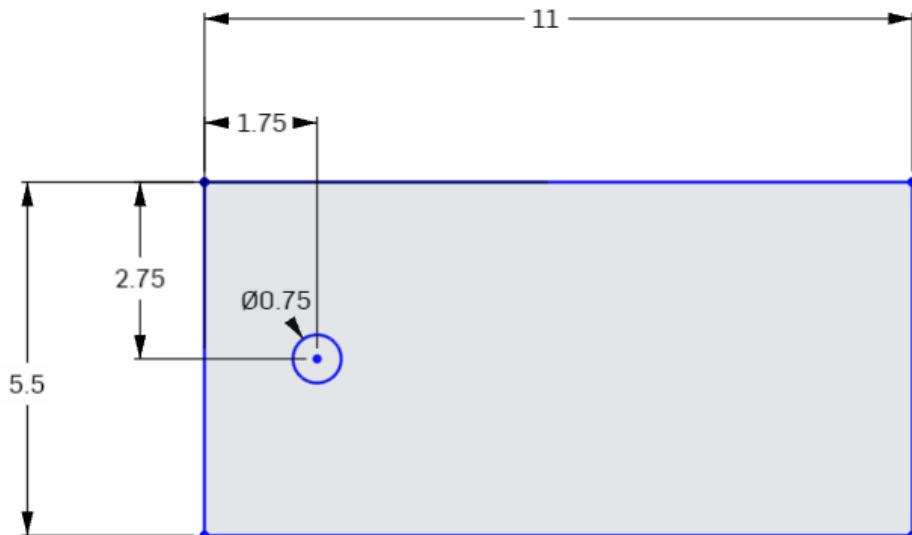
INSTRUCTIONS:

Foam Core Structural Pieces:

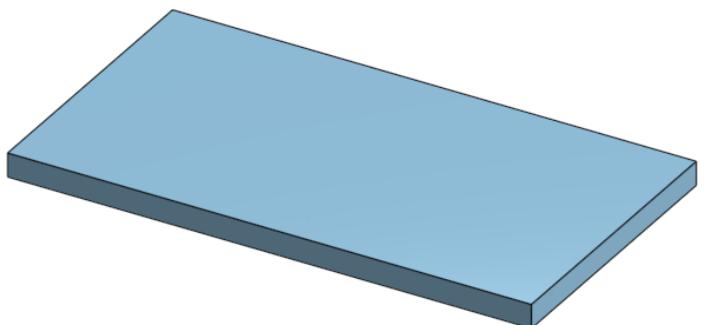
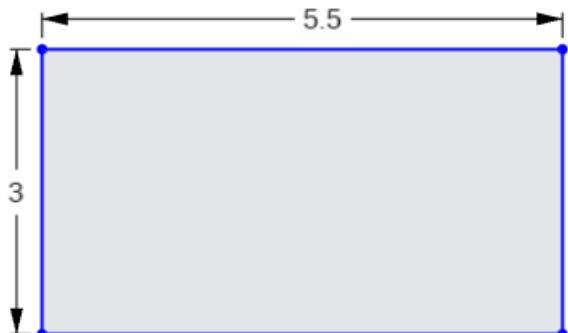
Use the pencil and ruler to sketch the following structural pieces of the robot on the foam core sheet. Then, using the exacto knife/box cutter, cut out the pieces.

Main Base/Cover (A) x 2:

Sketch and cut two of this piece. One will act as the main foundational base for the robot on which the breadboard will be placed and wheels will be attached. The second piece will serve as part of a top cover. The marker will be inserted through the holes, as shown later on.

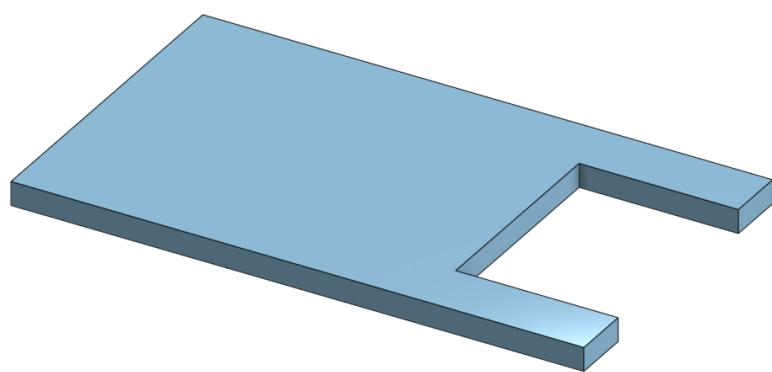
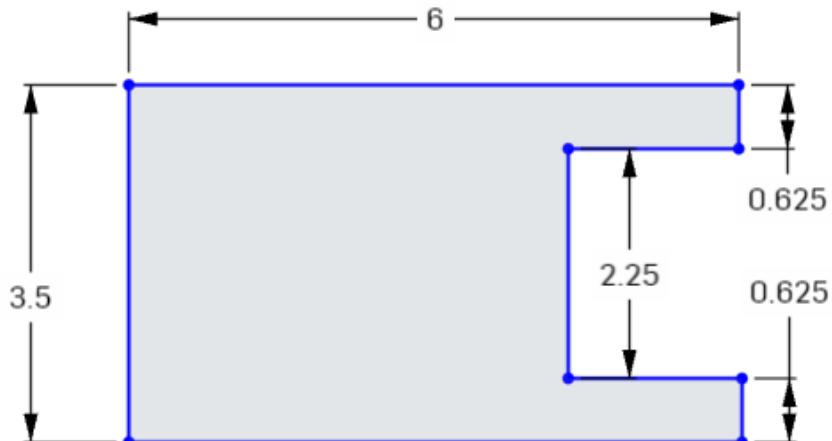


Main Cover – Front/Back (B) x 2:

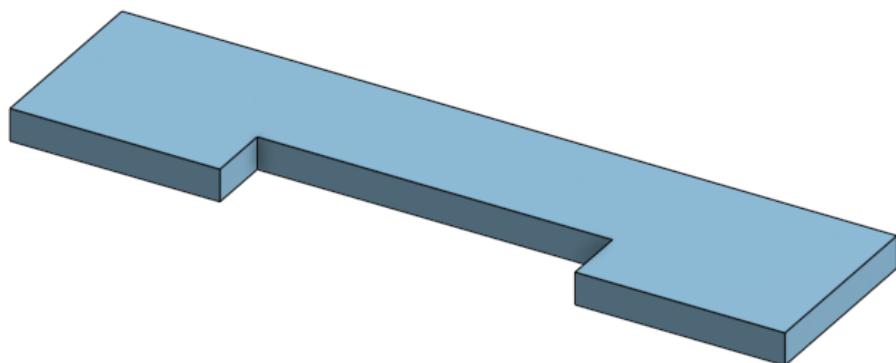
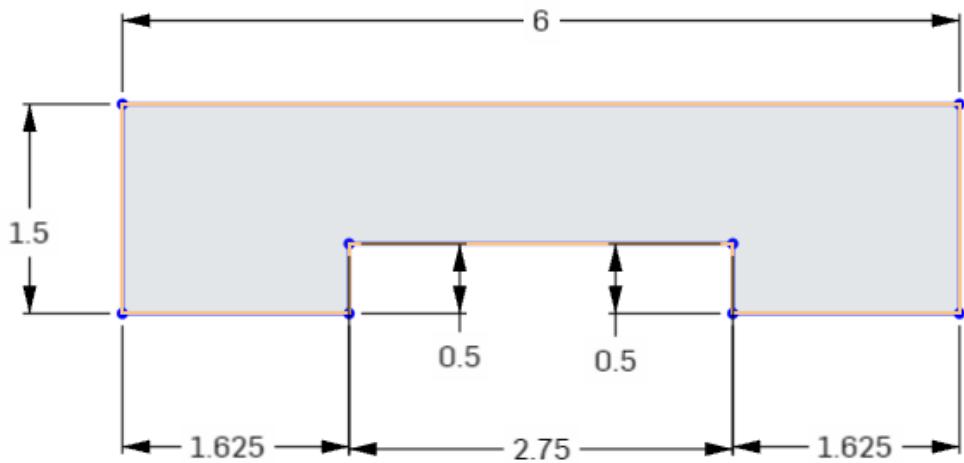


Arduino Foundation:

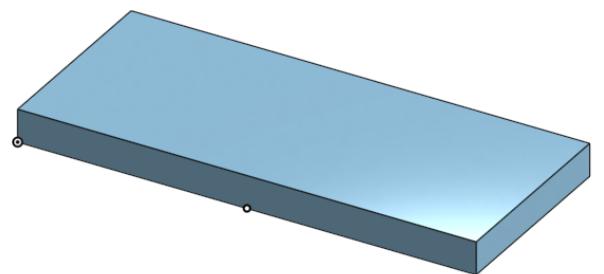
Top (C) x 1:



Side (D) x 2:

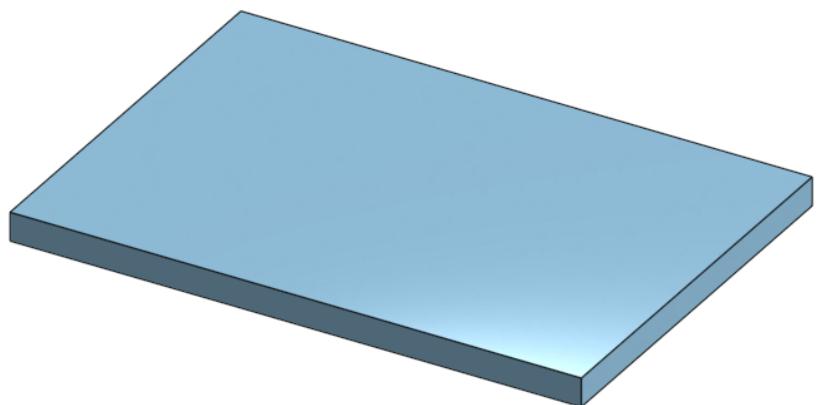
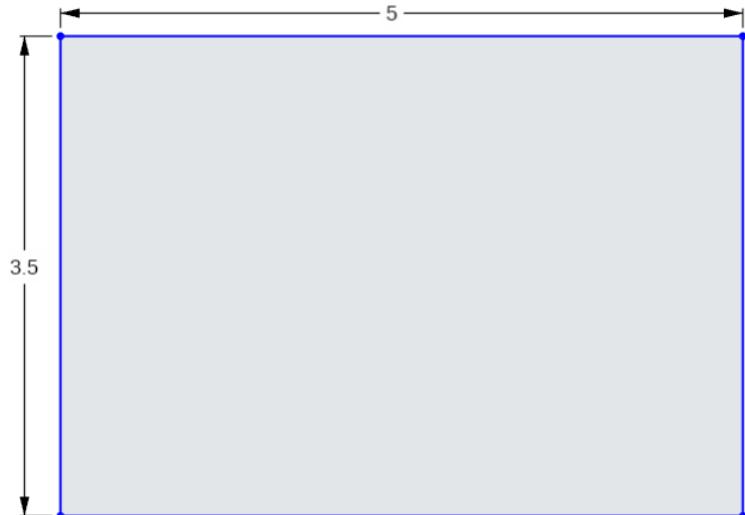


Front (E) x1:

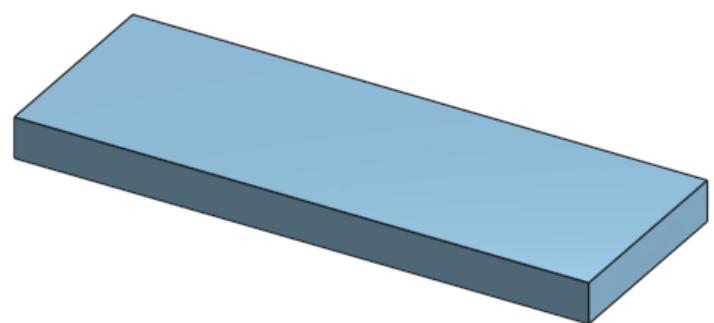
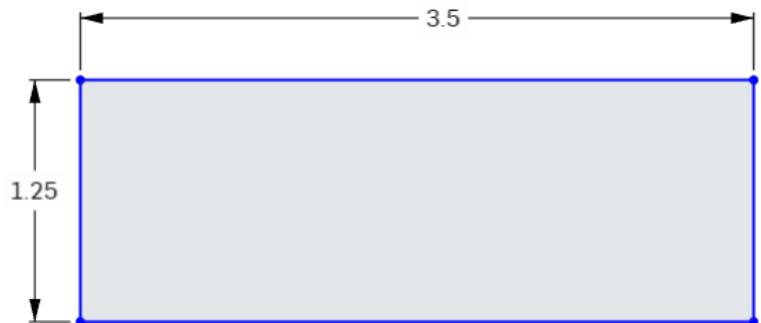


Arduino Top Cover:

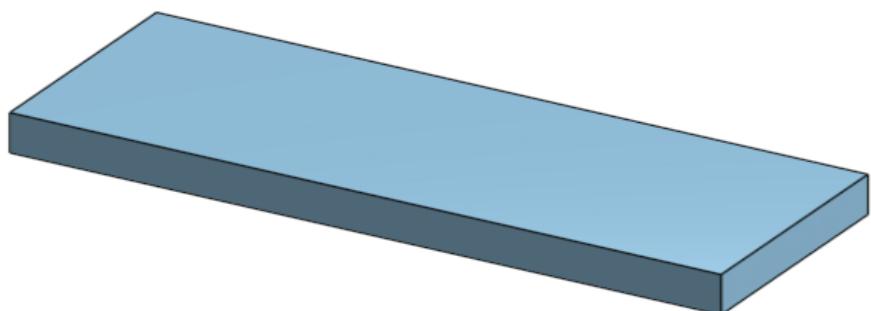
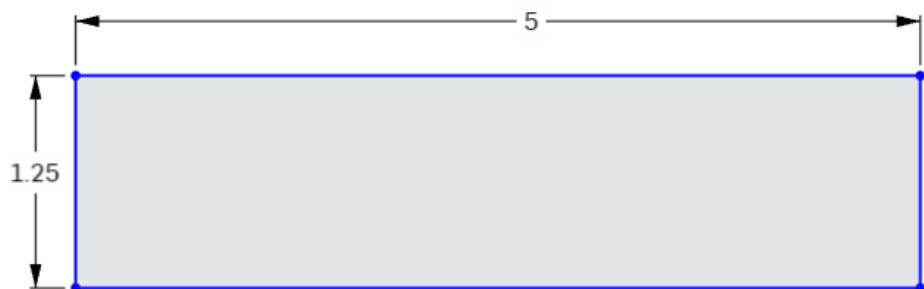
Top (F) x 1:



Front/Back (G) x 2:



Side (H) x 2:

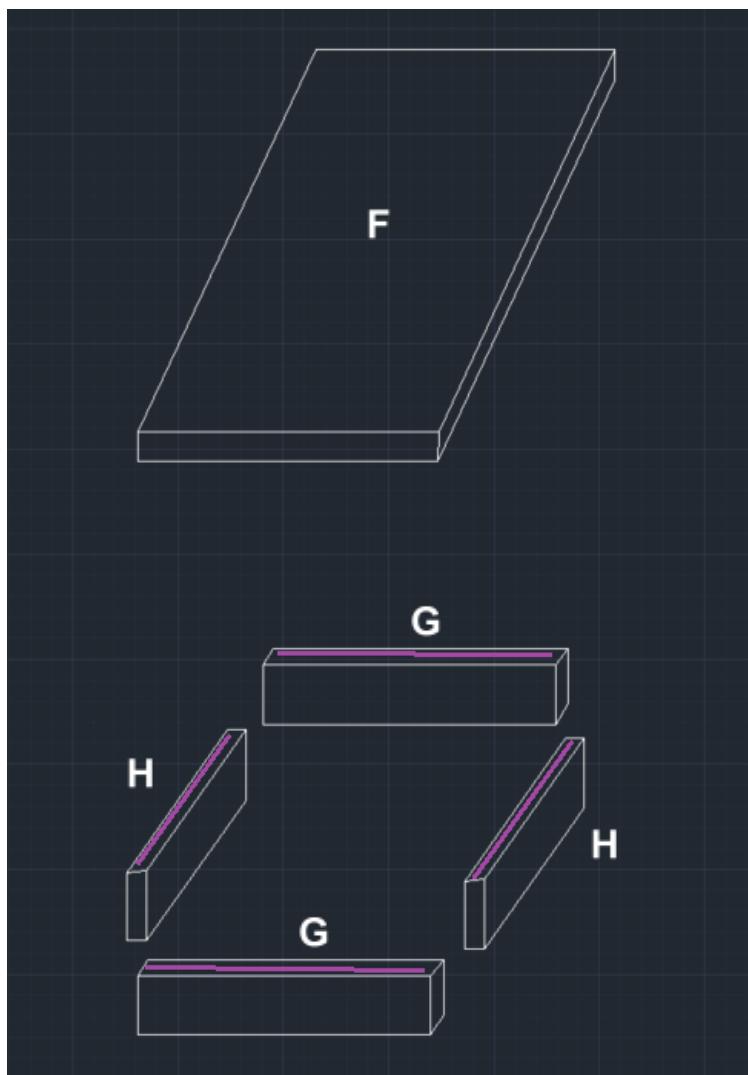


Structural Assembly

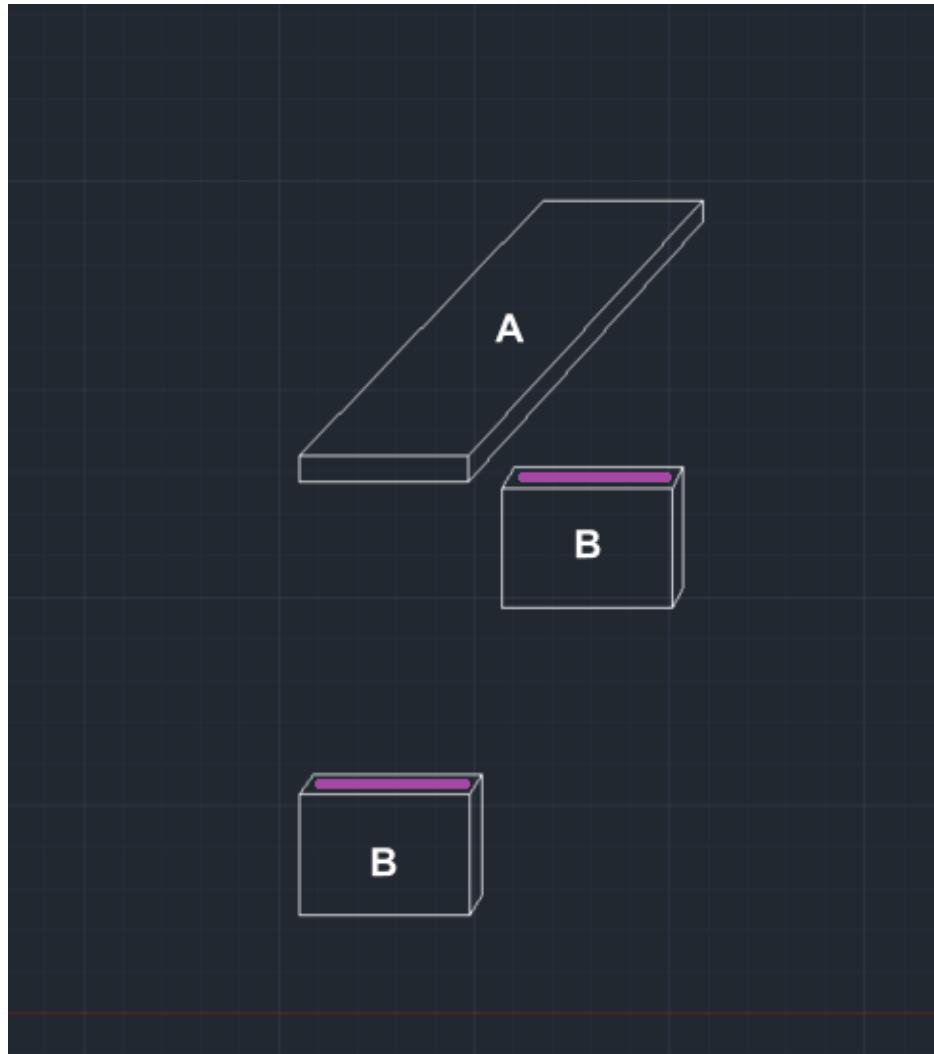
Now that all of the robot's structural pieces are cut out of the foam core, we can assemble them using glue and tape.

1. Arduino Cover:

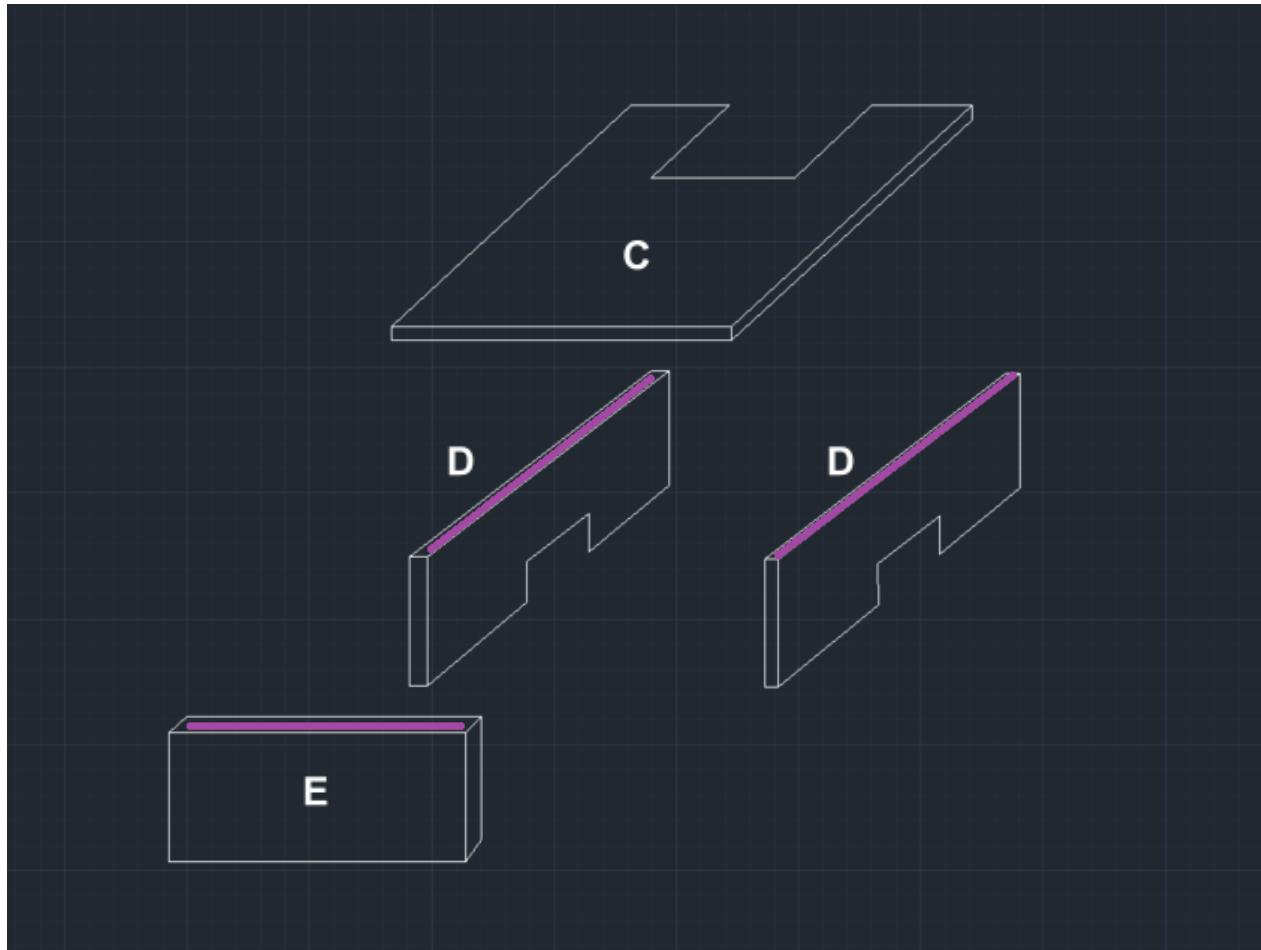
Use glue, to attach the front and sides to the top piece to form a rectangular prism.



2. Robot Top Cover:

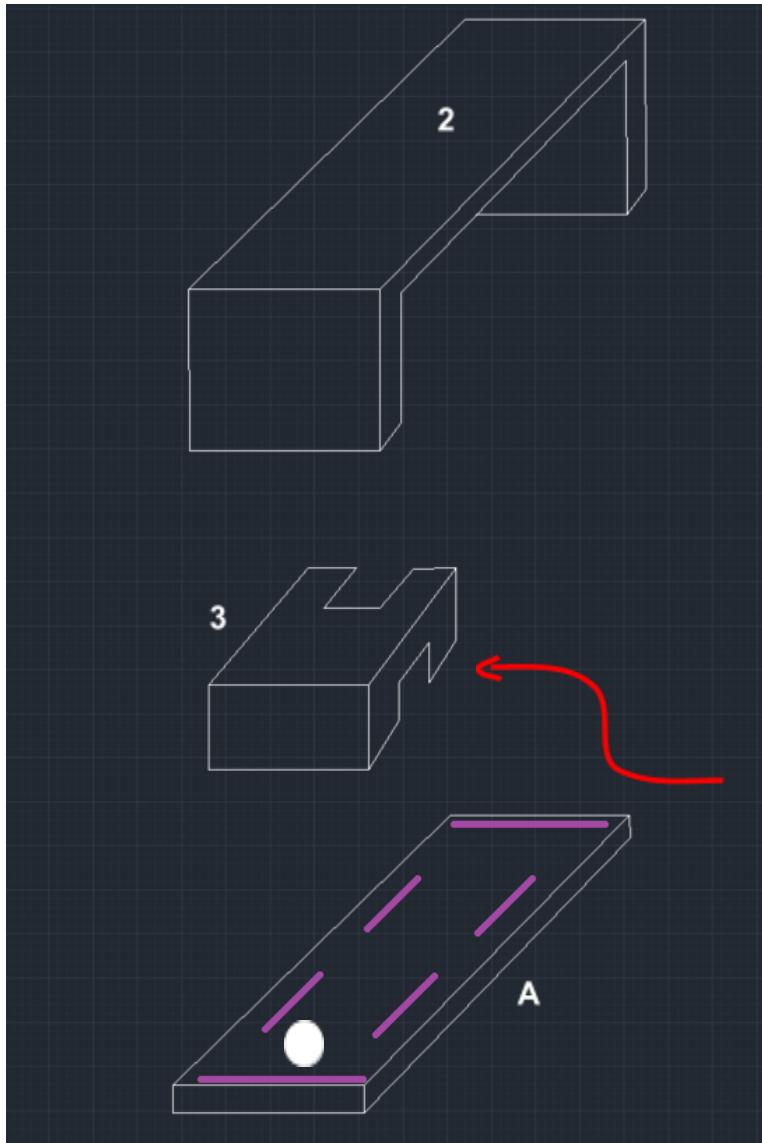


3. Arduino Base:



Robot Assembly

Now that all of the robot's structural sub-components are assembled, the sub-components can be assembled together to form the robot, along with the electronic equipment thereafter.

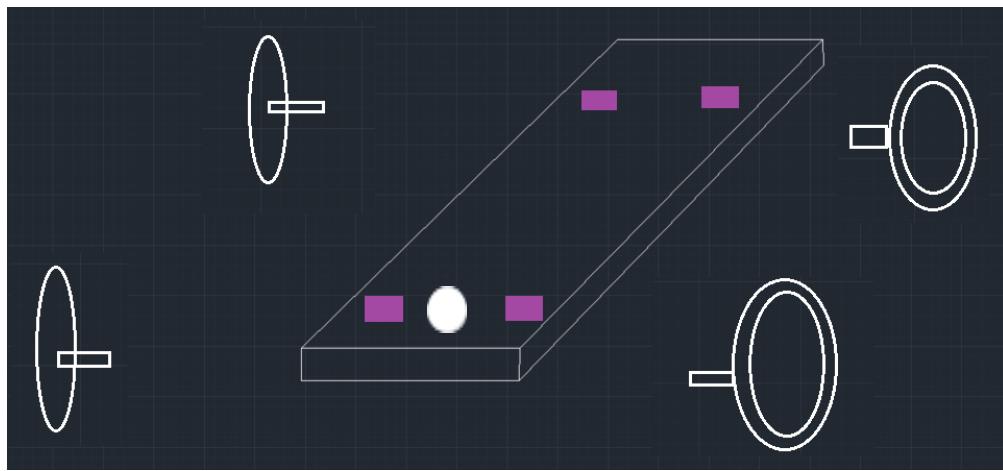


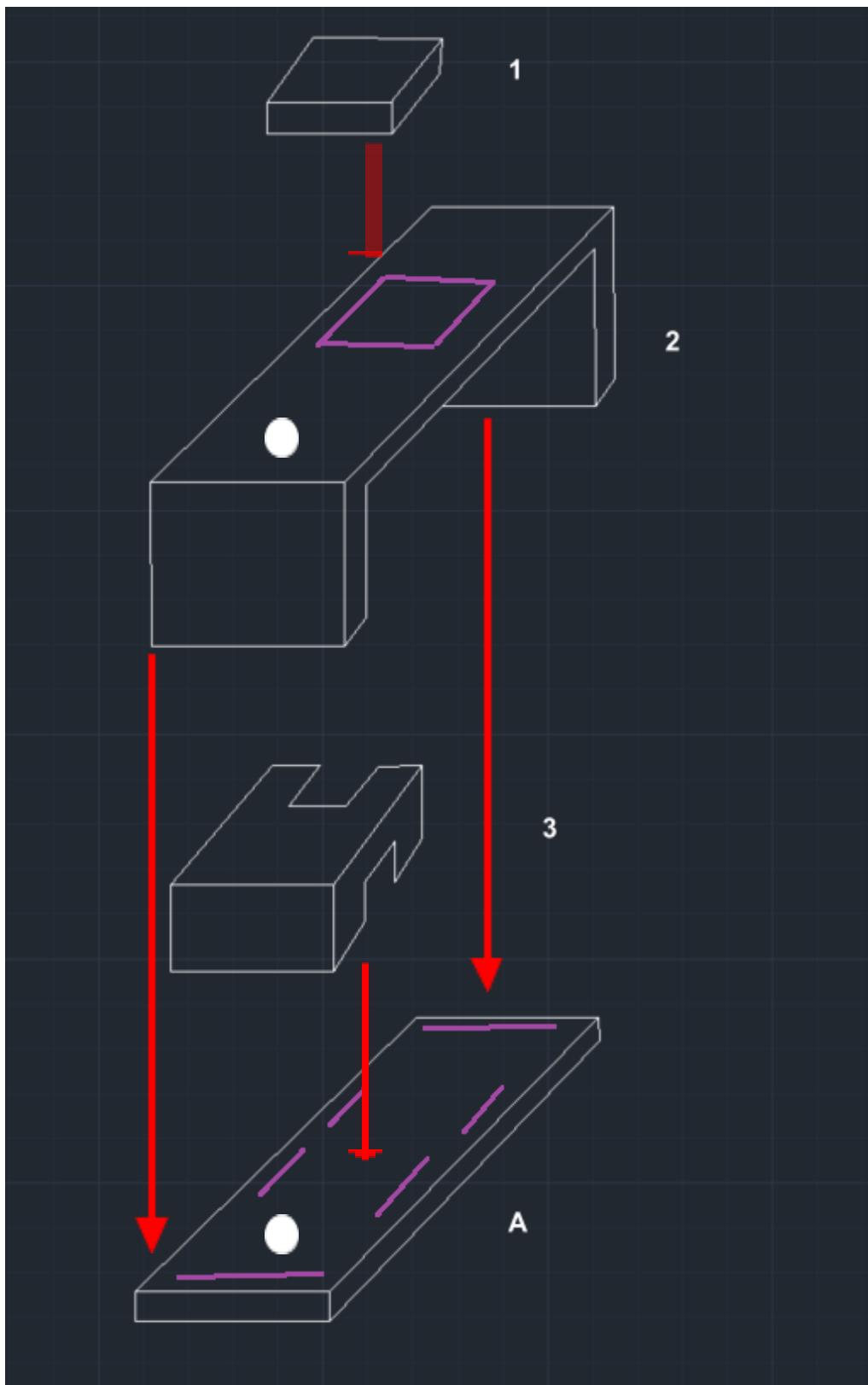
Use **tape** to attach the Top Cover(2) to the Main Base(A) for ease of removal when accessing internal robot components (i.e. the Arduino).

Use **glue** to attach the Arduino Foundation (2) to the main base. As only the front end of the Arduino Foundation (2) is closed off, the breadboard can slide in through the back.

Main Base with Wheel Assembly/Attachment:

Use glue to attach the wheels to the four sides of the base as shown in the picture below.

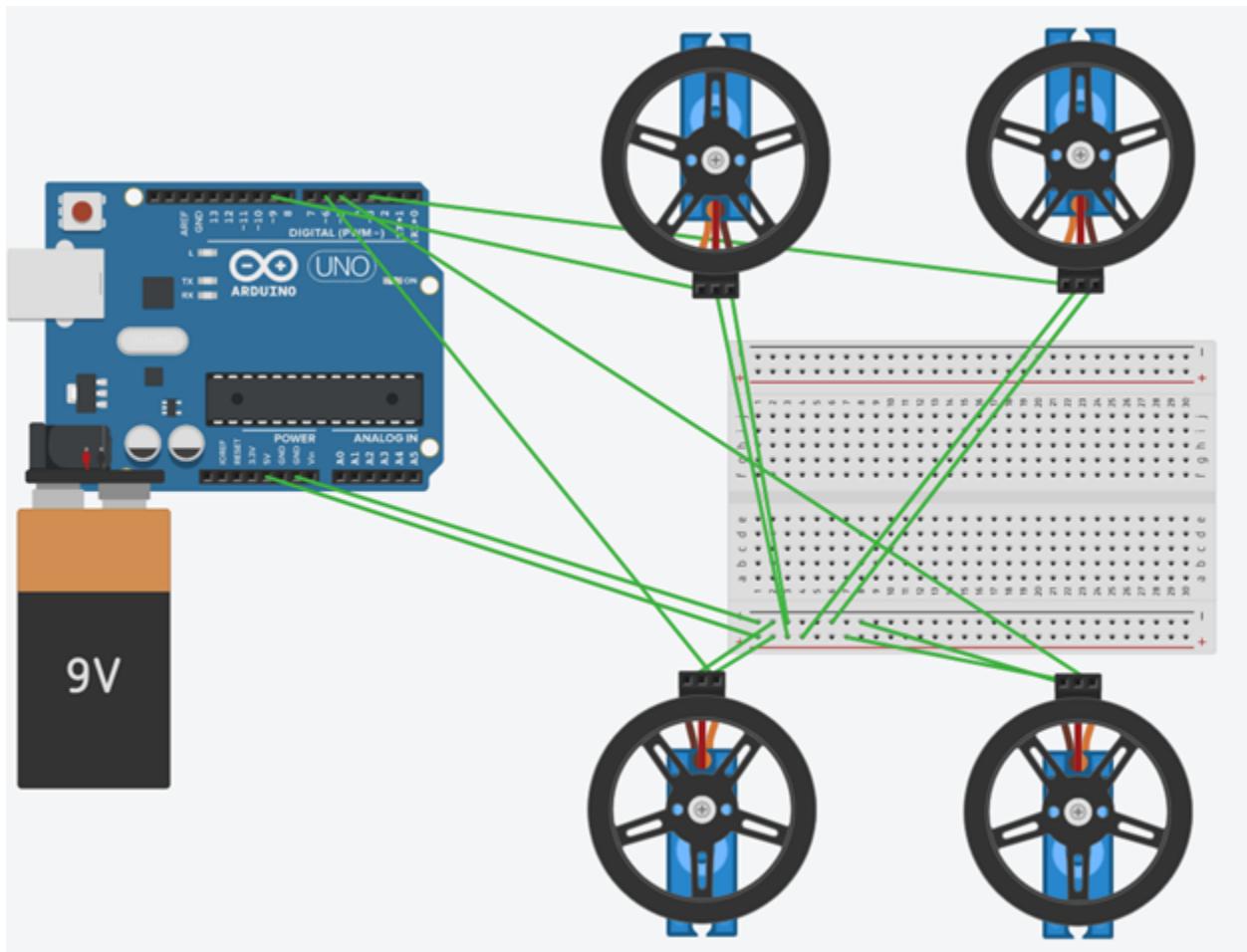




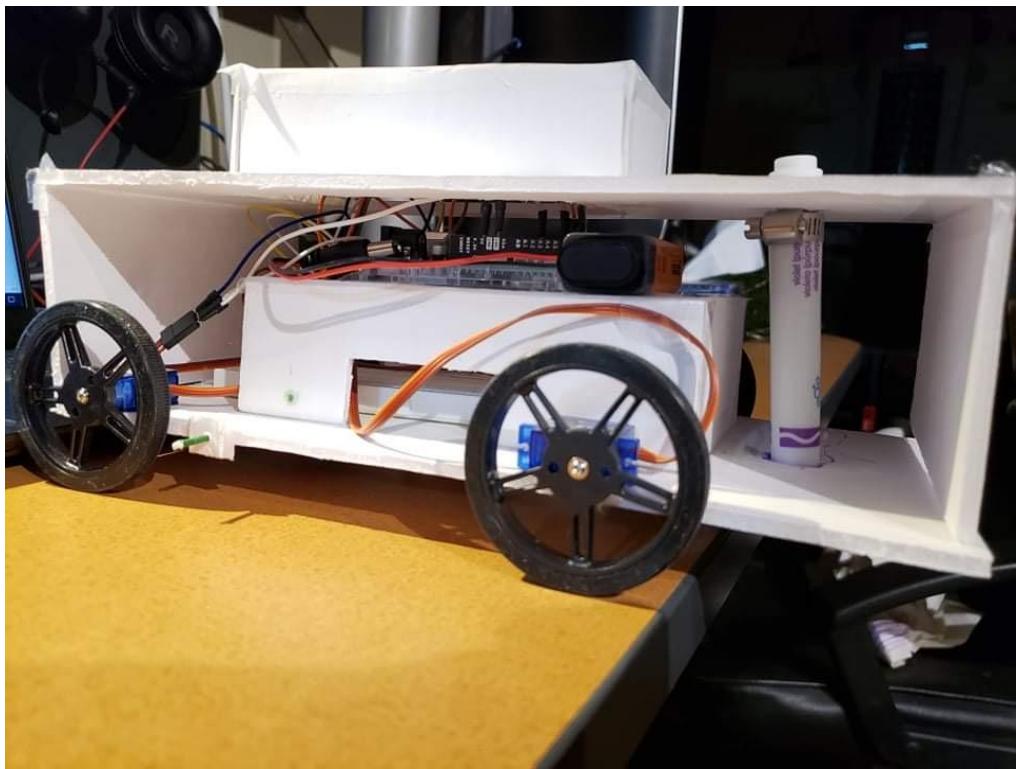
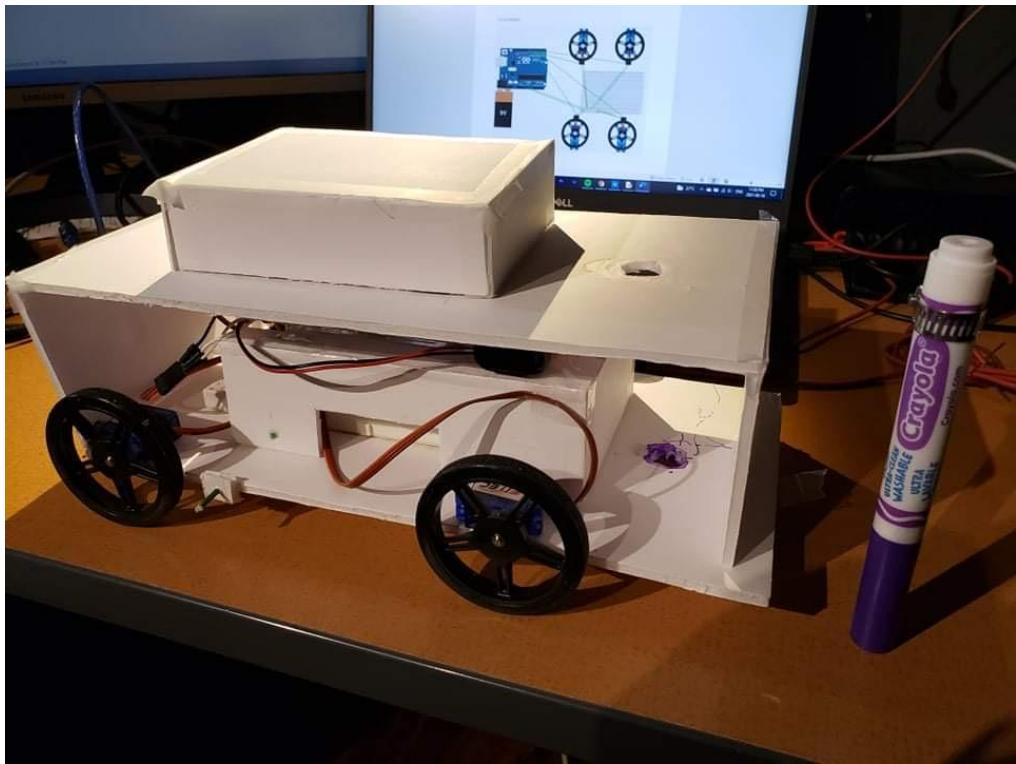
Circuit Assembly

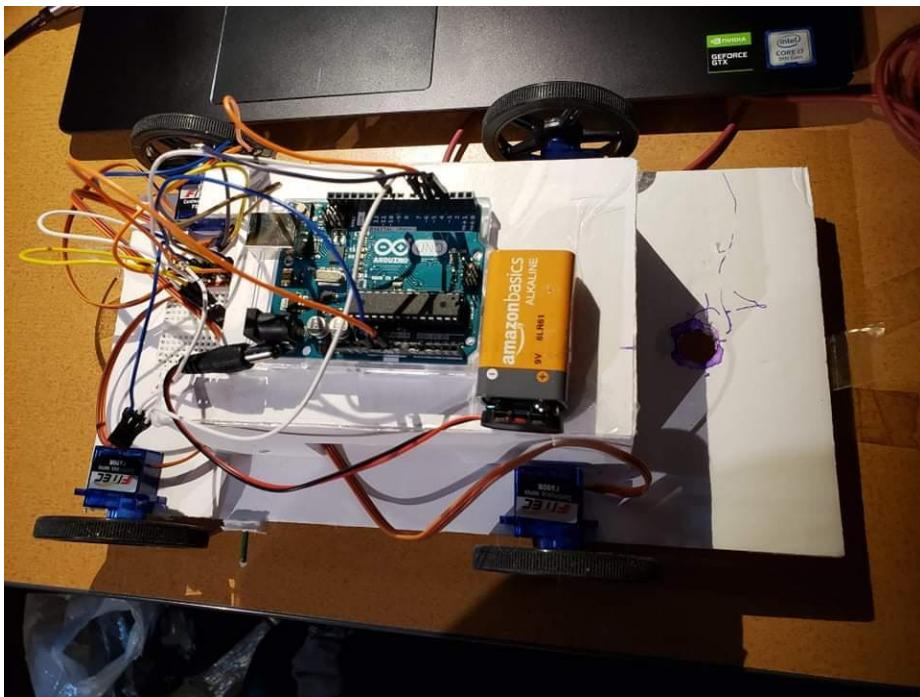
Each servo motor has 3 female ports to connect male jumper wires to. The brown wires will connect to your ground rail on the breadboard, the red wires will connect to your 5V power rail on the breadboard, and the orange wires will connect to a pin # on the Arduino board.

**Keep your battery disconnected until the code is uploaded to the Arduino, shown in the next section.



Final Assembled Robot





Arduino Code

The following code enables the robot to draw 1 square. Feel free to get creative with what you can make it draw!

Some tips:

E.g. Servo1.write(x)

x = 0 (forward (or backwards) depending on your wire orientation)

x = 90 (stop)

x = 180 (backwards (or forward) depending on your wire orientation)

To go forward or backwards make sure your wheels are all spinning in the same direction. To make a turn, your 2 sets of wheels need to be turning in opposite directions.

```
// Include the Servo library
```

```
#include <Servo.h>
```

```
// Create a servo object
```

```
Servo servo1;
```

```
Servo servo2;
```

```
Servo servo3;
```

```
Servo servo4;
```

```
void setup() {
```

```
    // We need to attach the servo to the used pin number
```

```
    servo1.attach(9);
```

```
    servo1.write(90);
```

```
    servo2.attach(6);
```

```
    servo2.write(90);
```

```
servo3.attach(3);  
servo3.write(90);  
servo4.attach(5);  
servo4.write(90);
```

//forward

```
servo1.write(0);  
servo2.write(180);  
servo3.write(0);  
servo4.write(180);  
delay(2000);
```

//stop

```
servo1.write(90);  
servo2.write(90);  
servo3.write(90);  
servo4.write(90);  
delay(1500);
```

//Right

```
servo1.write(180);  
servo2.write(180);  
servo3.write(180);  
servo4.write(180);  
delay(2400);
```

//stop

```
servo1.write(90);  
servo2.write(90);  
servo3.write(90);  
servo4.write(90);  
delay(1500);
```

//forward

```
servo1.write(0);  
servo2.write(180);  
servo3.write(0);  
servo4.write(180);  
delay(2000);
```

//stop

```
servo1.write(90);  
servo2.write(90);  
servo3.write(90);  
servo4.write(90);  
delay(1500);
```

//Right

```
servo1.write(180);  
servo2.write(180);  
servo3.write(180);  
servo4.write(180);  
delay(2400);
```

```
//stop  
  
servo1.write(90);  
  
servo2.write(90);  
  
servo3.write(90);  
  
servo4.write(90);  
  
delay(1500);
```

```
//forward  
  
servo1.write(0);  
  
servo2.write(180);  
  
servo3.write(0);  
  
servo4.write(180);  
  
delay(2000);  
  
//stop  
  
servo1.write(90);  
  
servo2.write(90);  
  
servo3.write(90);  
  
servo4.write(90);  
  
delay(1500);  
  
//Right  
  
servo1.write(180);  
  
servo2.write(180);  
  
servo3.write(180);  
  
servo4.write(180);
```

```
delay(2400);

//stop

servo1.write(90);

servo2.write(90);

servo3.write(90);

servo4.write(90);

delay(1500);
```

```
//forward

servo1.write(0);

servo2.write(180);

servo3.write(0);

servo4.write(180);

delay(2000);
```

```
//stop

servo1.write(90);

servo2.write(90);

servo3.write(90);

servo4.write(90);

delay(1500);
```

```
//stop

servo1.write(90);

servo2.write(90);

servo3.write(90);
```

```
servo4.write(90);  
delay(1500);  
}
```

```
void loop(){  
}
```