

**AUTODESK**  
Instructables

## 3D Printed Hexapod Robot

By [zeroshot](#) in [CircuitsRobots](#)

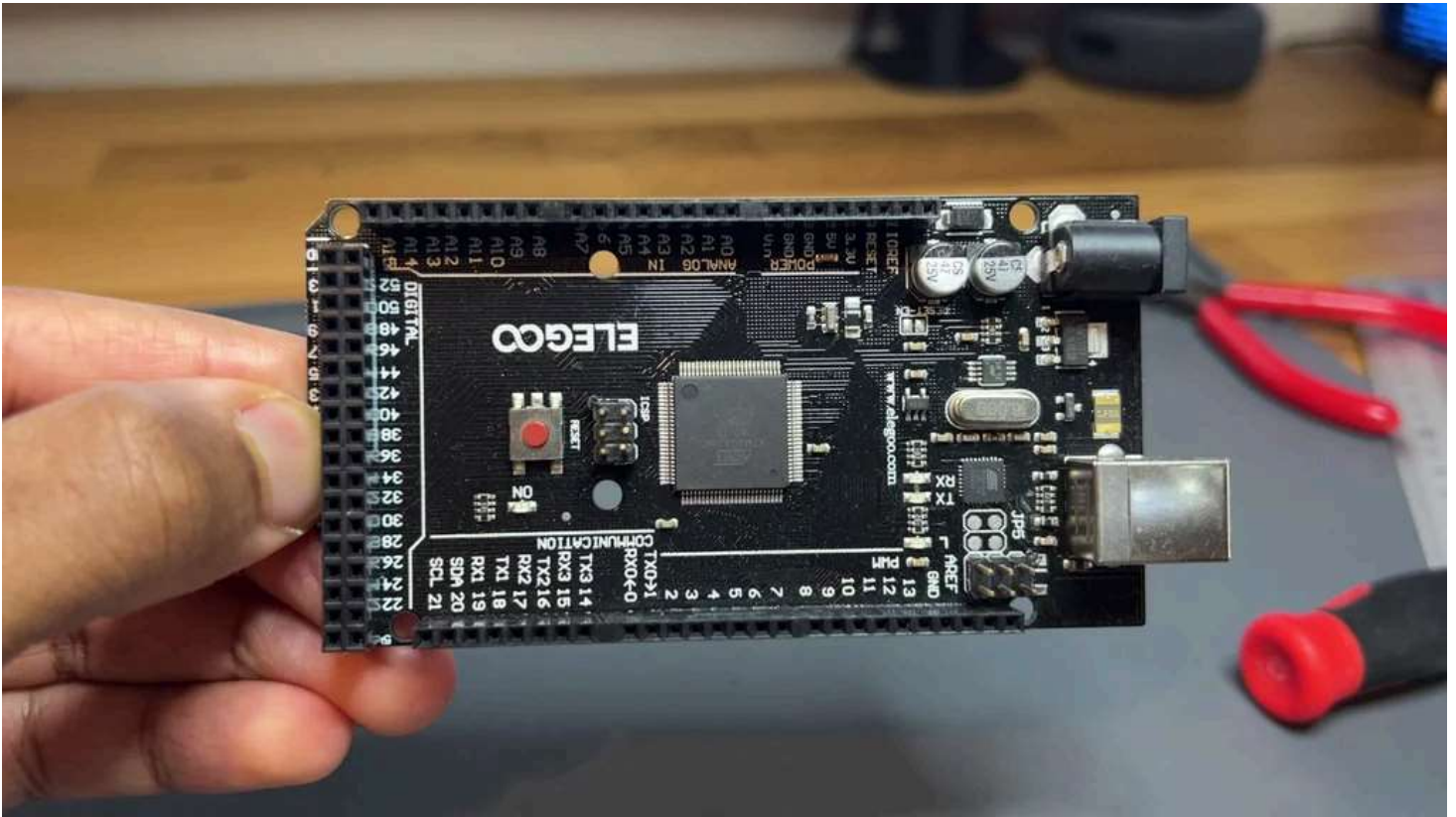


### Introduction: 3D Printed Hexapod Robot



A hexapod is a type of robot with 6 legs that can move independently, which makes them good at moving across different types of terrain. Now a few months ago I came up with the idea of making my own hexapod just for fun. So in this project I show the design and assembly of a fully 3D printed, arduino based hexapod.

## Supplies



*Disclaimer - Supplies list may include affiliate/paid links.*

## Parts

- 18× MG996R Servos [Amazon](#)
- Arduino MEGA [Amazon](#)
- Arduino MEGA Sensor Shield [Amazon](#)
- 3S Lipo Battery 2200mAh [Amazon](#)
- 9V Battery [Amazon](#)
- 9V Battery barrel connector [Amazon](#)
- DC-DC Buck Converter Module [Amazon](#)
- 22 AWG Solid Core wire [Amazon](#)
- XT60 Connectors [Amazon](#)
- M3 screws
- Zip ties

## Tools

- 3D Printer
- Soldering Iron
- Screwdriver
- Multimeter

## Step 1: Watch the Video!

For a better overview of the entire build process and for some footage of the hexapod in action check out the build video.

## Step 2: 3D Printing

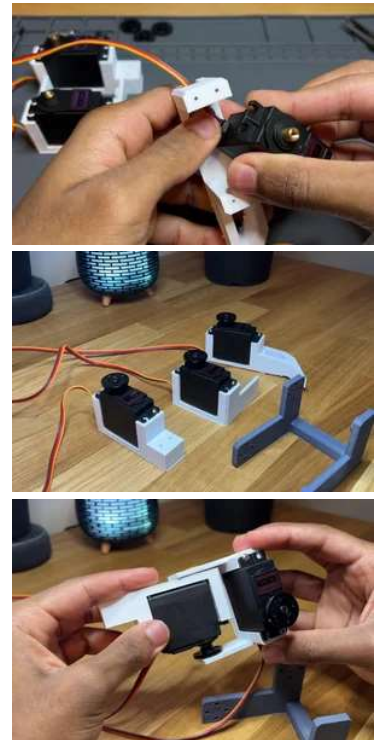
For this project all of the mechanical parts are 3D printed. Most of them can be printed using PLA with standard settings and minimal supports.

A leg is made from these parts: **leg\_mount.stl**, **coxa.stl**, **femur.stl**, **tibia.stl**, which will need to be printed 6 times in total.

The rest of the parts make up the base of the robot and only need to be printed once, with **mounting\_post.stl** being the only exception as two copies of this are needed.

⚠ I would recommend printing a single leg and assembling it first to check that everything fits together correctly.

## Step 3: Leg Assembly



First, insert servo motors into the **leg\_mount**, **coxa** and **tibia** pieces - making sure to carefully pass the servo cables through the wiring holes in each piece. The sizing might be quite tight but everything should fit snugly together.

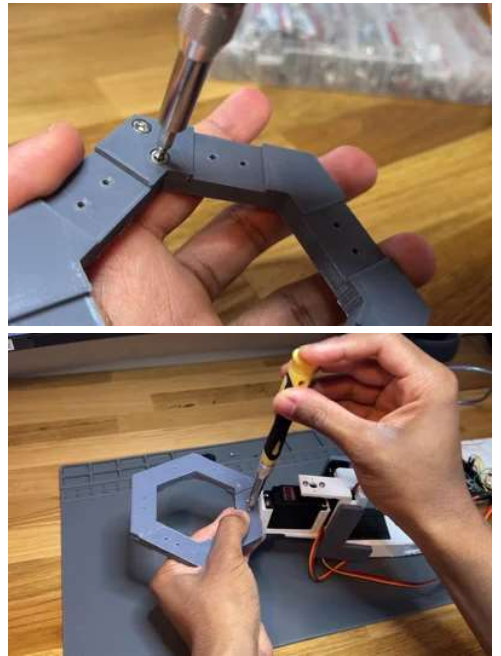
Making sure that each of the servos are set to the **0 degrees** position, attach servo horns to each motor shaft.

Assembling a leg should be fairly straightforward now with all the parts are laid out.

We want the leg to have as wide a range of motion as possible, so I recommend connecting the joints at the same angle as shown in the GIF above (when the leg is pulled down).

⚠ Take note of the angle that each joint is connected at, as we will be using that in the code later.

## Step 4: Leg Code



The next step is to get the leg moving. The 3 pins of each servo need to be connected: red to a 5v power supply or battery, brown to ground, and yellow to a digital pin on the Arduino.

To test the leg joints this code snippet can be uploaded to the Mega:

```
// snippet to move leg joints
#include <Servo.h>

#define SERVO_PIN 23
#define J1_OFFSET 45
#define J2_OFFSET 0
#define J3_OFFSET 90

Servo servo;

void move_joints(double j1, double j2, double j3){
  if (j1>=-15 && j1<=105 && j2>=0 && j2<=90 && j3>=0 && j3<=90){
    servo.write(j1 + J1_OFFSET);
    servo.write(j2 + J2_OFFSET);
    servo.write(j3 + J3_OFFSET);
  }
}

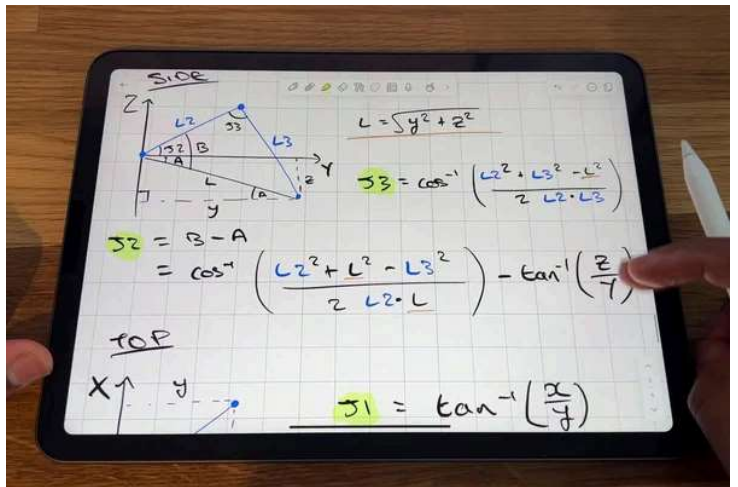
void setup(){
  servo.attach(SERVO_PIN, 600, 2600);
}

void loop(){
  move_joints(...);
}
```

Also, at this stage the 2 part base frame can be printed off and attached to the leg to provide some support.



## Step 5: Inverse Kinematics



Placing the end of the leg at a specific point requires the inverse kinematics of the leg to be calculated. Combined with the **move\_joints()** code from earlier the code below can be used to move to an XYZ position:

```
// inverse kinematics
#define FEMUR_L 60
#define TIBIA_L 95.334
#define TIBIA_ANGLE 17.543
#define Y_OFFSET 50
#define Z_OFFSET -90

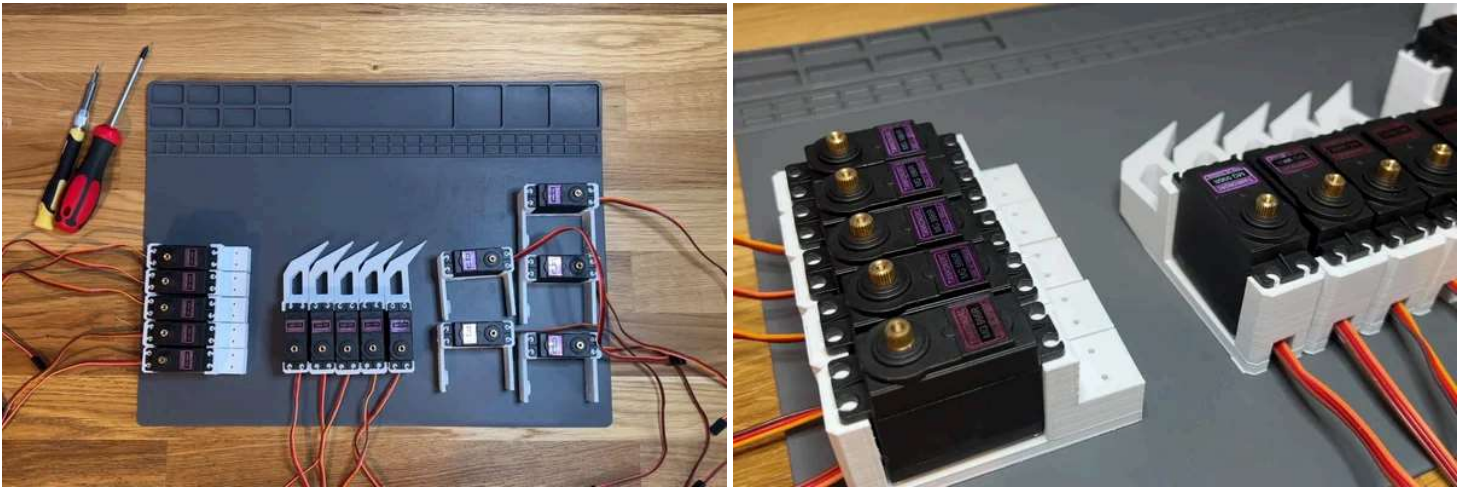
void move_xyz(int leg, double x, double y, double z)
{
    y += Y_OFFSET;
    z += Z_OFFSET;

    double l = sqrt(x*x + y*y + z*z);
    double j1 = atan(x/y);
    double j2 = acos((FEMUR_L*FEMUR_L - TIBIA_L*TIBIA_L + l*l) / (2*FEMUR_L*l)) + atan(z / sqrt(x*x + y*y));
    double j3 = acos((FEMUR_L*FEMUR_L + TIBIA_L*TIBIA_L - l*l) / (2*FEMUR_L*TIBIA_L));

    j1 = 45 + j1*(180/PI);
    j2 = 30 + j2*(180/PI);
    j3 = 90 + 30 - TIBIA_ANGLE - j3*(180/PI);

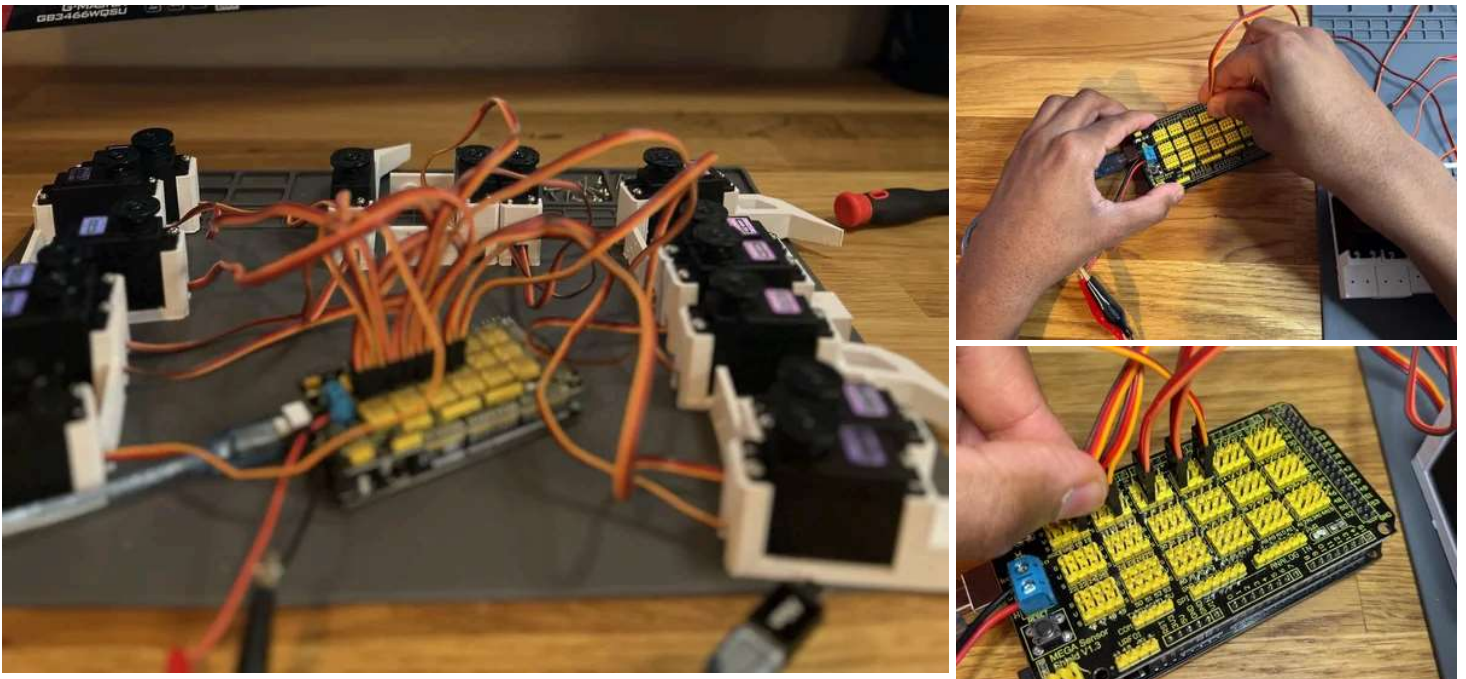
    move_joints(j1, j2, j3);
}
```

## Step 6: Building the Rest of the Legs



The rest of the hexapod legs can be printed and assembled in the exact same way as before. Since there are 6 legs, a total of 18 servos will be required.

## Step 7: Servo Wiring

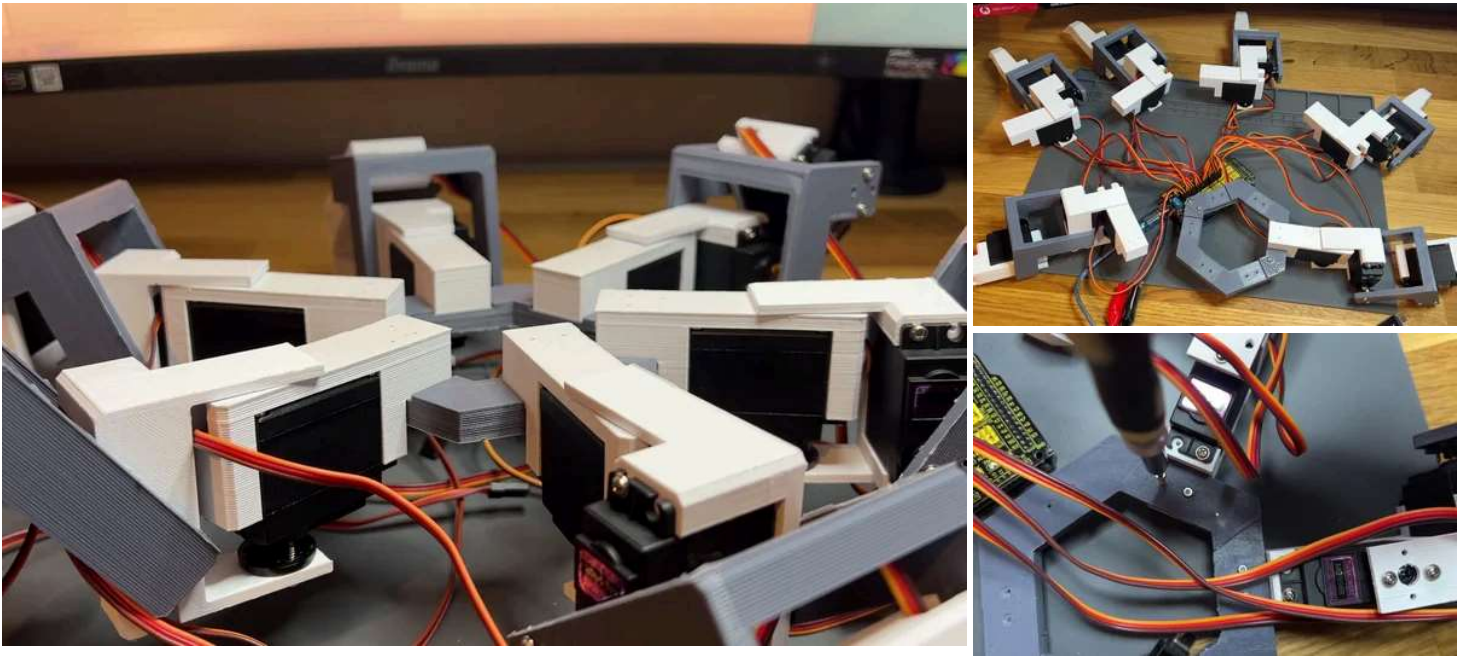


Attach the **sensor shield** to the Arduino MEGA and connect each of the servos. Take a picture or make a note of which servo is connected to which pin on the shield

Connect 5V and GND wires to the terminal block on the sensor shield for later use.

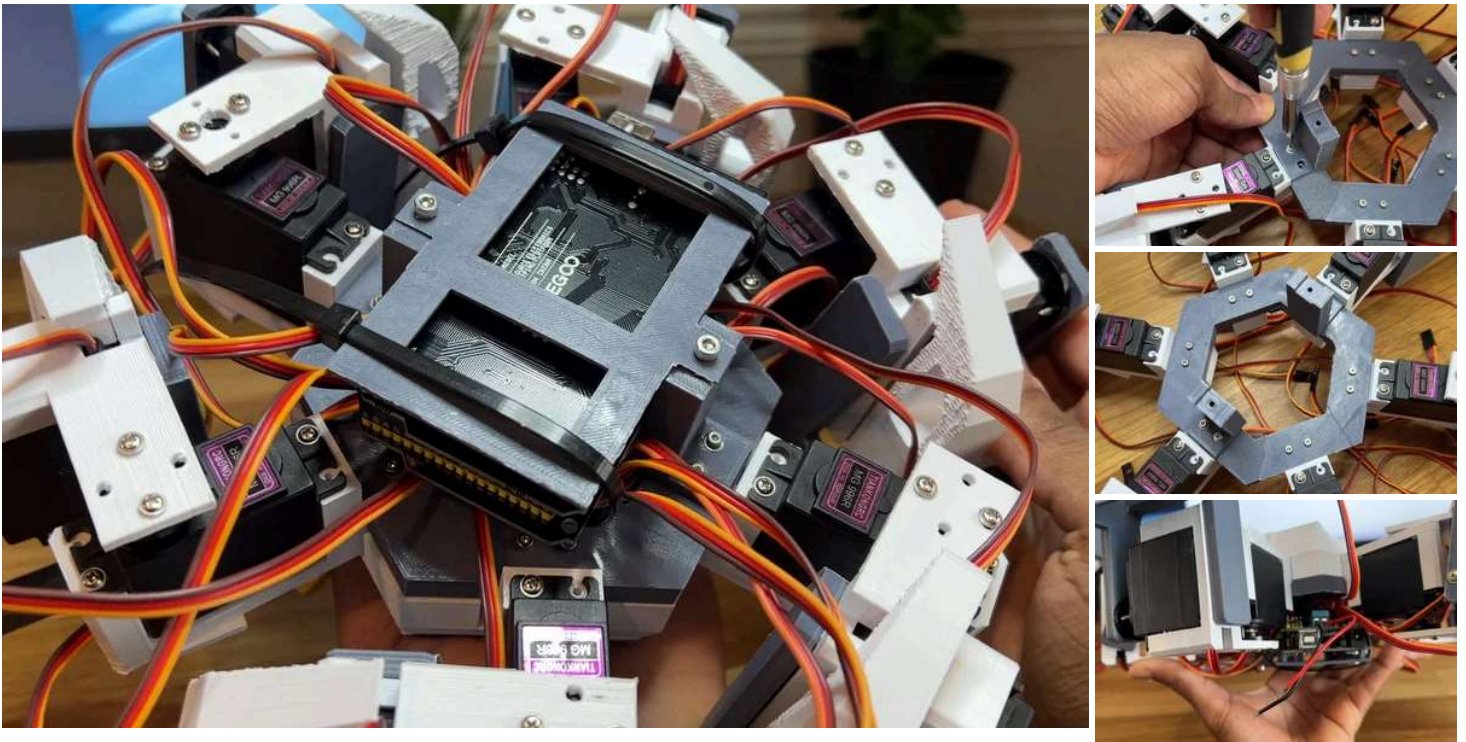


## Step 8: Attaching the Legs



Using M3 screws, attach the remaining legs to the hexapod base frame, leaving two legs that are parallel to each other **unattached** for the next step.

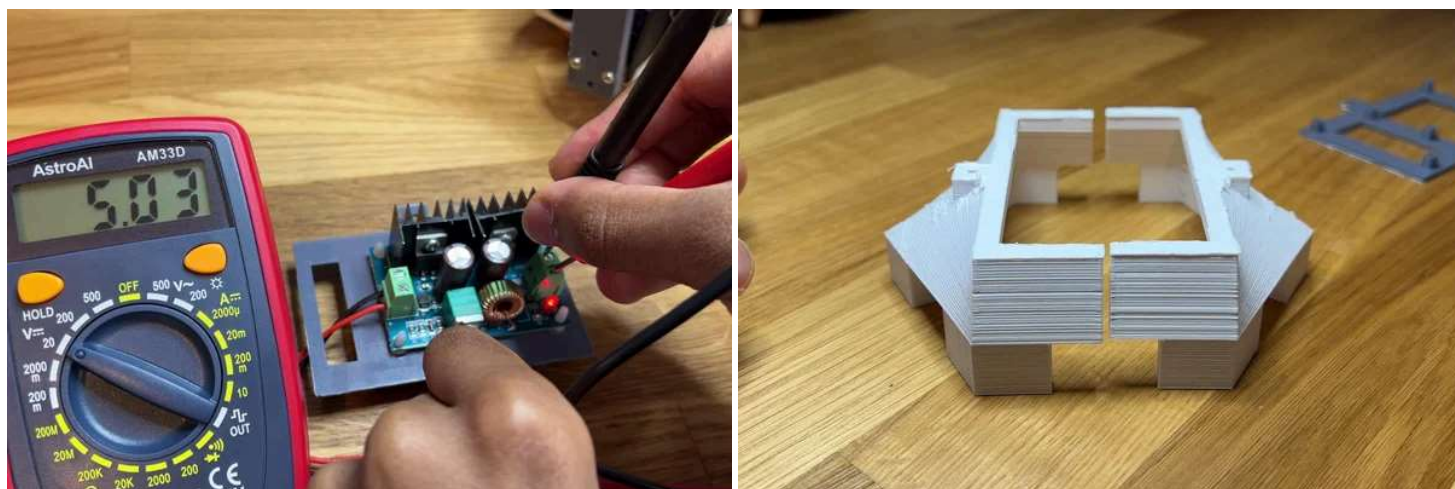
## Step 9: Undercarriage Assembly



With the hexapod flipped upside down, connect the remaining two legs to both the base frame and two **mounting posts** as shown in the image.

Fasten the arduino to the **mega\_mount** using standoffs (or zip ties) and screw into the mounting posts.

## Step 10: Top Cover Electronics



Fit the **DC-DC Buck Converter** module to the **top\_cover** piece. Using a multimeter this should be set to output **5V**.

The two halves **hat\_1** and **hat\_2** can now be attached to the top cover using screws.

Connect the 5V and GND wires from the sensor shield to the buck converter **output terminal block**.

Solder an XT60 battery connector to some wires and screw these into the buck converter **input terminal block**. Remember to check and make sure the polarity is correct for both this connection and the last one.



## Step 11: Hexapod Gait Code

The last step is to upload the code **hexapod.ino** onto the arduino. In the code there are functions that implement a simple ripple walking gait for the hexapod.


### Quick explanation of the code:

- Each leg goes through two phases: the swing phase and the stance phase.
- **Swing Phase:** The leg is lifted and moved forward.
- **Stance Phase:** The leg is on the ground, pushing the body forward.
- The legs move in a ripple pattern to ensure the hexapod remains stable.
- No more than two legs are lifted from the ground at a time

### Movement example

- Imagine the legs are in this order: 1, 2, 3, 4, 5, 6.
- The sequence starts with leg 1 moving into the swing phase (lifted and moving forward).
- As leg 1 begins to land (end of swing phase), leg 3 starts its swing phase.
- Next, as leg 3 is landing, leg 5 starts its swing phase.
- Meanwhile, legs 2, 4, and 6 are in their stance phases, keeping the hexapod stable and moving forward.

More information on hexapod gaits can be found online.

 Before uploading the code make sure to change the offsets and pin assignments to the correct values.

## Step 12: Finished!



The build is now finished!

To use the hexapod:

- Power the arduino through the barrel jack using a 9V battery
- Connect the LiPo through the XT60 connector.
- The code is designed to be easily extendable to make the hexapod turn, move at different speeds and walk with different gaits.

If you would like to see more footage of the hexapod and other robotics projects check me out [here](#).

Thank you for reading.