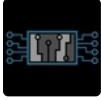




## Arduino Controlled Robotic Biped



by Technovation

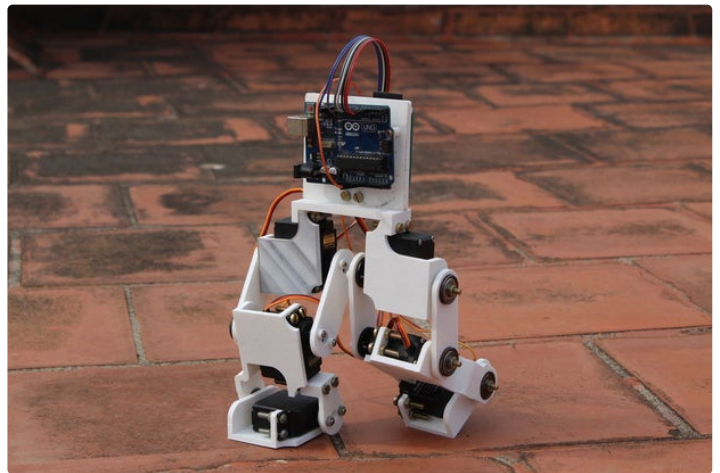
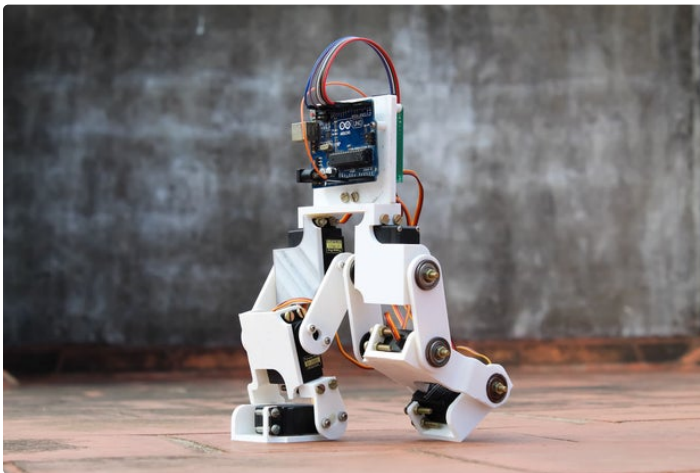
I have always been intrigued by robots, especially the kind that attempts to mimic human actions. This interest led me to try to design and develop a robotic biped that could imitate human walking and running. In this Instructable, I will show you the design and assembly of the robotic biped.

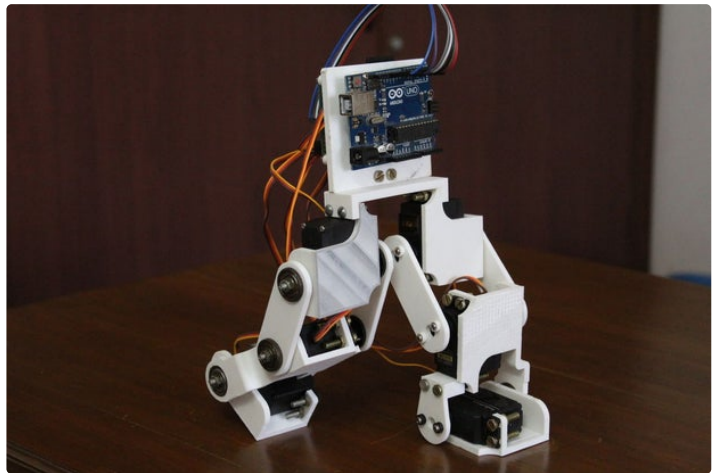
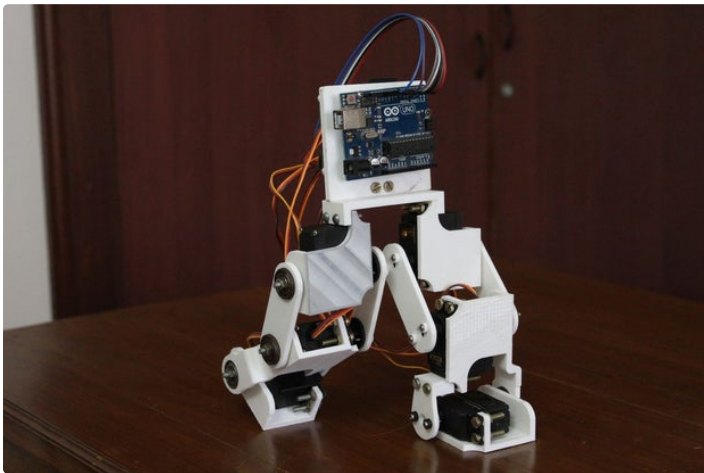
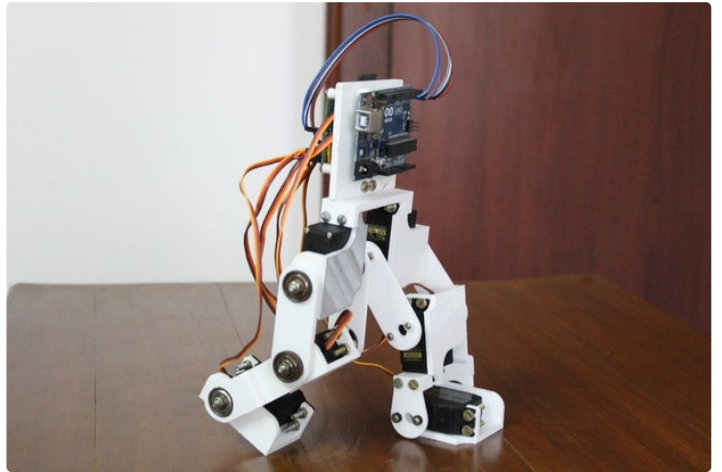
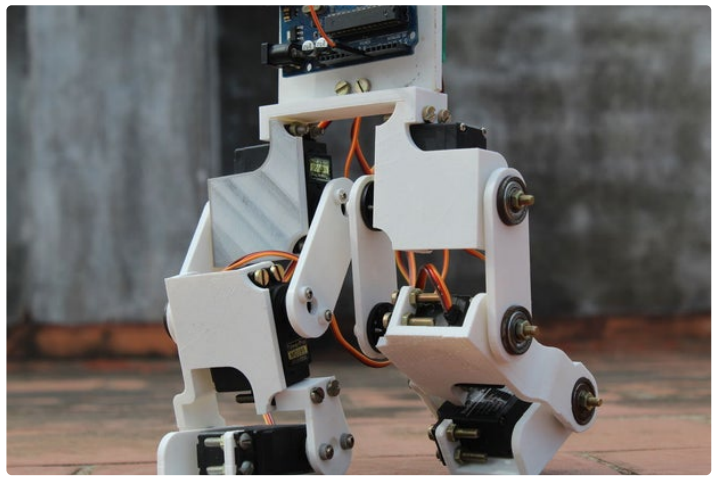
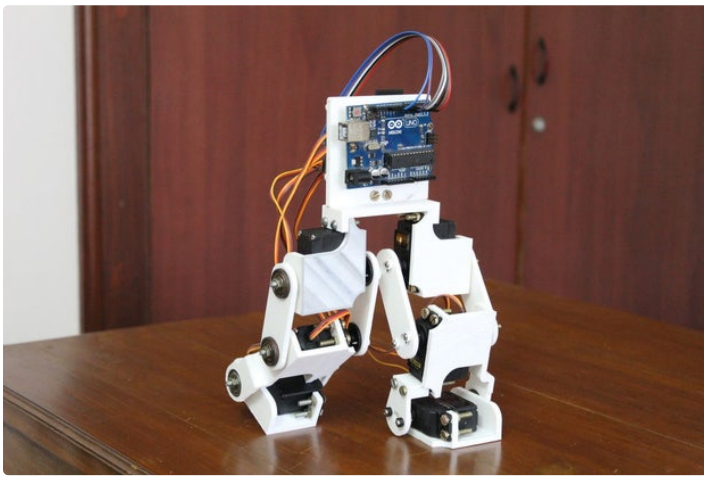
The primary goal while building this project was to make the system as robust as possible such that while experimenting with various walking and running gaits, I wouldn't have to constantly worry about the hardware failing. This allowed me to push the

hardware to its limit. A secondary goal was to make the biped relatively low-cost using readily available hobby parts and 3D printing leaving room for further upgrades and expansions. These two goals combined provide a robust foundation to perform various experiments, letting one develop the biped to more specific requirements.

Follow on to create your own **Arduino controlled Robotic Biped** and do drop a vote in the "**Arduino Contest**" if you liked the project.

<https://youtu.be/CxociTjzR4Q>





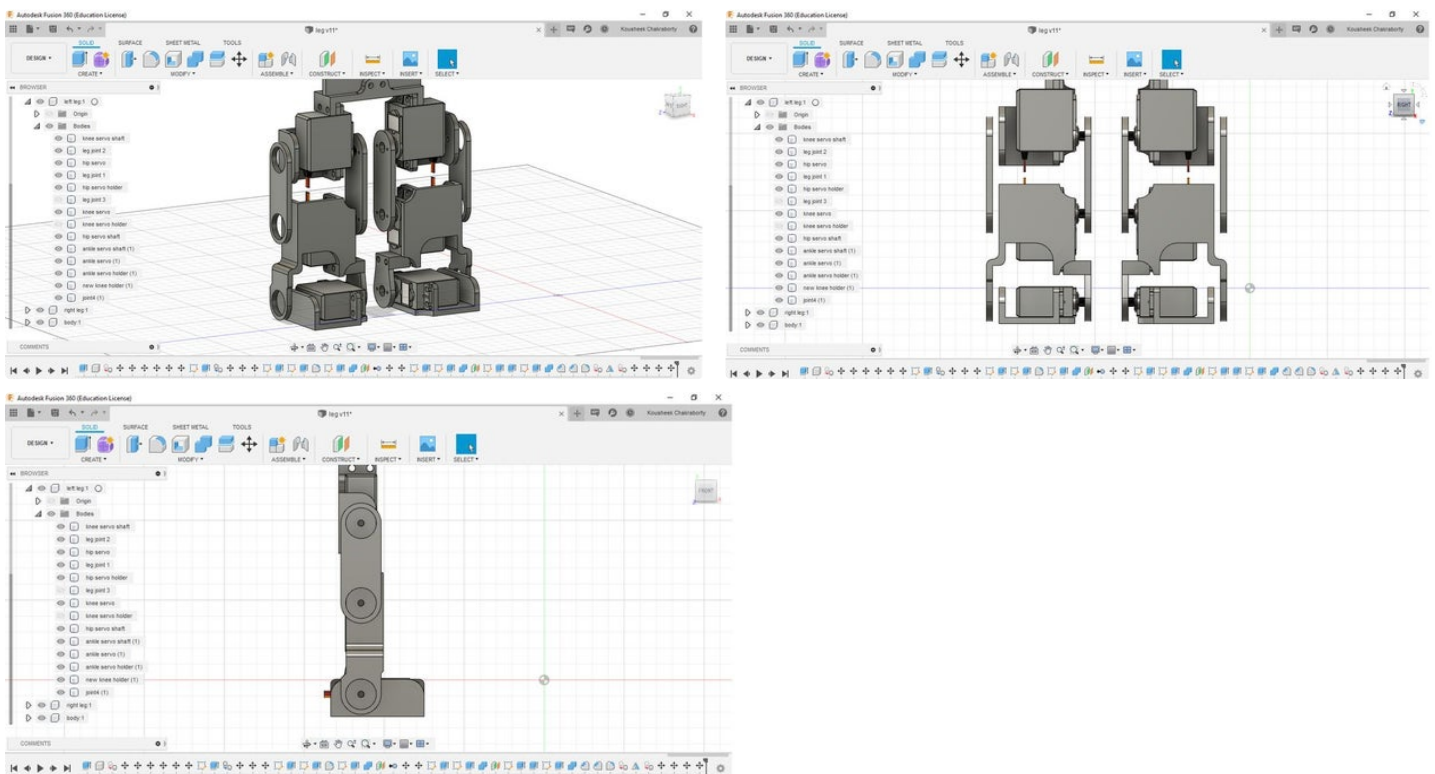
---

## Step 1: Design Process

The humanoid legs were designed in Autodesk's free to use [Fusion 360](#) 3d modelling software. I began by importing the servo motors into the design and built the legs around them. I designed brackets for the servo motor which provides a second pivot point diametrically opposite to the servo motor's shaft. Having dual shafts on either end of the motor gives structural stability to the design and eliminates any skewing that may occur when the legs are made to take some load. The links were designed to hold a bearing while the brackets used a bolt for the shaft. Once the links were mounted to the shafts using a nut, the bearing would provide a smooth and robust pivot point on the opposite side of the servo motor shaft.

Another goal while designing the biped was to keep the model as compact as possible to make maximum use of the torque provided by the servo motors. The dimensions of the links were made to achieve a large range of motion while minimizing the overall length. Making them too short would make the brackets collide, reducing the range of motion, and making it too long would exert unnecessary torque on the actuators. Finally, I designed the body of the robot onto which the Arduino and other electronic components would mount.

**Note:** The parts are included in one of the following steps.



## Step 2: The Role of the Arduino



An Arduino Uno was used in this project. The Arduino was responsible to compute the motion paths of the various gaits that were tested and instructed the actuators to move to precise angles at precise speeds to create a smooth walking motion. An Arduino is a great choice for developing projects because of its versatility. It provides a bunch of IO pins and also provides interfaces such as serial, I2C, and SPI to communicate with other microcontrollers and sensors. The Arduino also provides a great platform for rapid prototyping and testing and also gives developers room for improvements and expandability.



In this project, further versions will include an Inertial Measurement Unit for motion processing such as fall detection and dynamic locomotion in uneven terrain and a distance measuring sensor to avoid obstacles.

The Arduino IDE was used for this project. (Arduino also provides a web-based IDE)

**Note:** The programs for the robot can be downloaded from one of the following steps.

---

### Step 3: Materials Needed

Here is the list of all the components and parts required to make your very own Arduino powered Bipedal robot. All parts should be commonly available and easy to find.

#### **ELECTRONICS:**

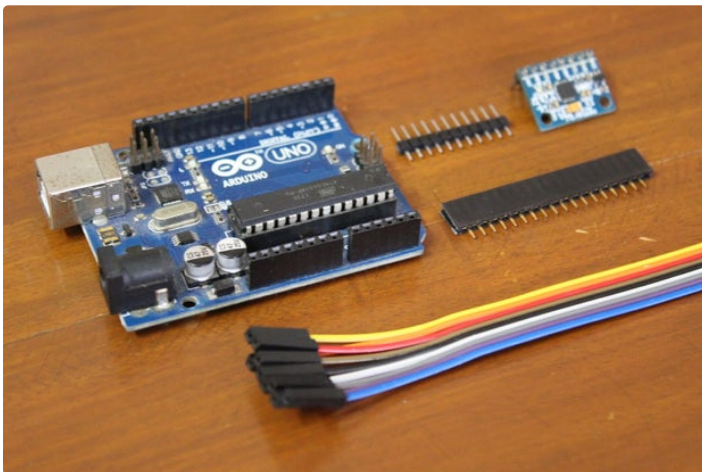
- **Arduino Uno** x 1
- **Towerpro MG995 servo motor** x 6
- **Perfboard** (similar size to the Arduino)
- **Male and Female header pins** (about 20 of each)
- **Jumper Wires** (10 pieces)
- **MPU6050 IMU** (optional)

- **Ultrasonic sensor** (optional)

## **HARDWARE:**

- **Skateboard Bearing** (8x19x7mm)
- **M4 nuts and bolts**
- **3D printer filament** (in case you don't own a 3D printer, there should be a 3D printer in a local workspace or the prints can be done online for quite cheap)

Excluding the Arduino and 3D printer the total cost of this project is 20\$.



---

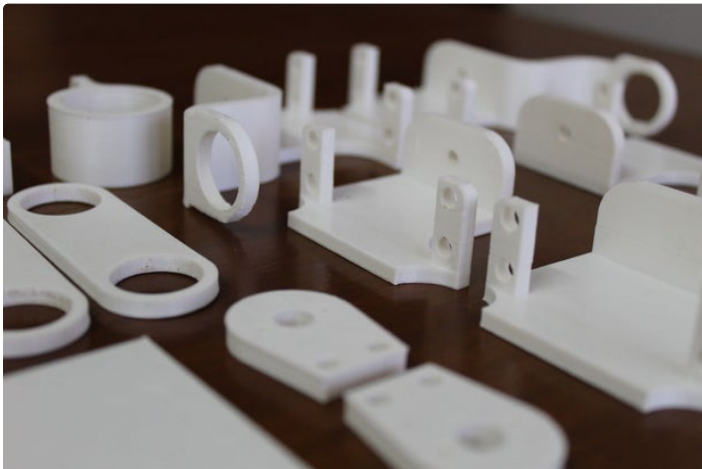
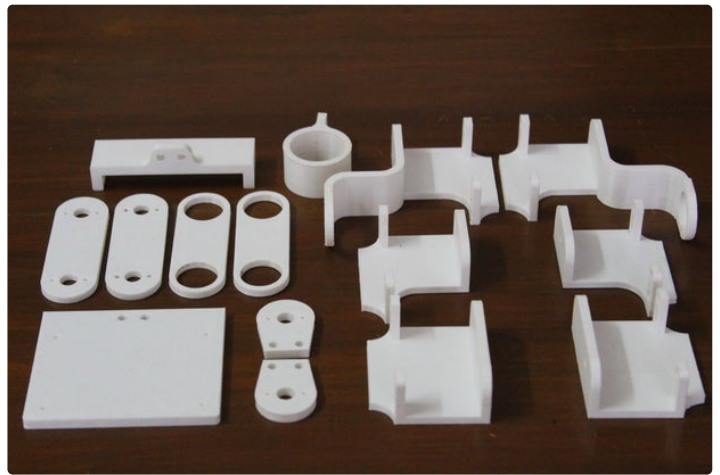
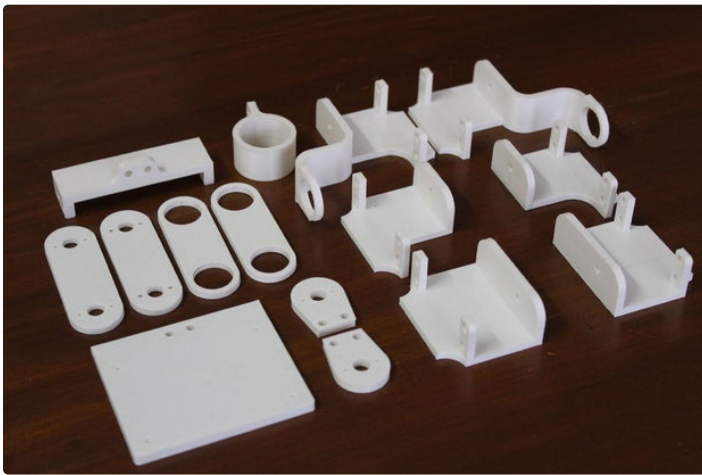
## **Step 4: 3D Printed Parts**

The parts required for this project had to be custom designed therefore a 3D printer was used to print them out. The prints were made at 40% infill, 2 perimeters, 0.4mm nozzle, and a layer height of 0.1mm with PLA, color of your choice. Below you can find the complete list of parts and the STLs to print your own version.

**Note:** From here on the parts will be referred to using the names in the list.

- **foot servo holder** x 1
- **foot servo holder mirror** x 1
- **knee servo holder** x 1
- **knee servo holder mirror** x 1
- **foot servo holder** x 1
- **foot servo holder mirror** x 1
- **bearing link** x 2
- **servo horn link** x 2
- **foot link** x 2
- **bridge** x 1
- **electronics mount** x 1
- **electronics spacer** x 8 (optional)
- **servo horn space** x 12 (optional)

In total, excluding the spacers, there are 14 parts. The total printing time is about 20 hours.



<https://www.instructabl...>

Download

## Step 5: Preparing the Servo Brackets

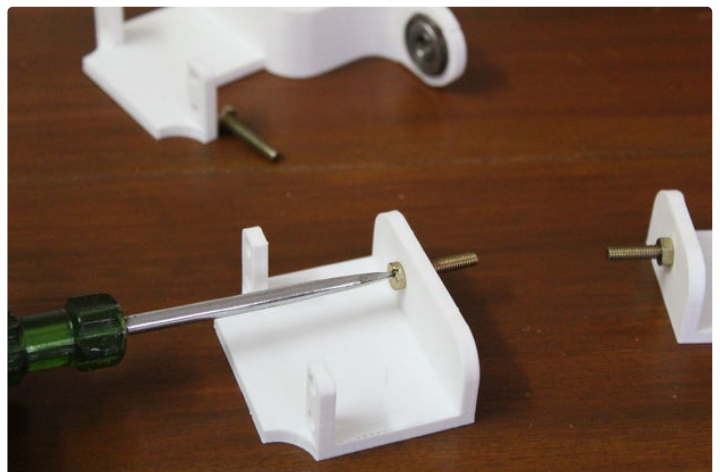
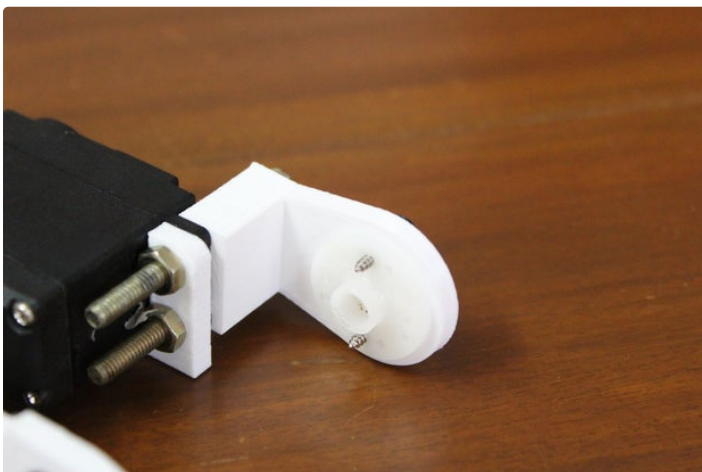
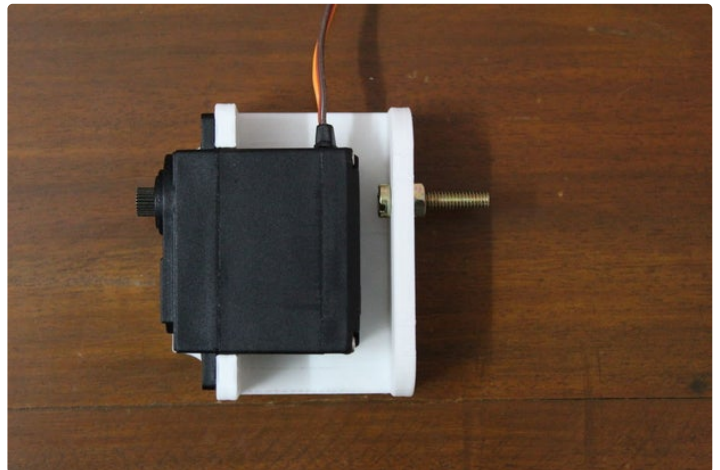
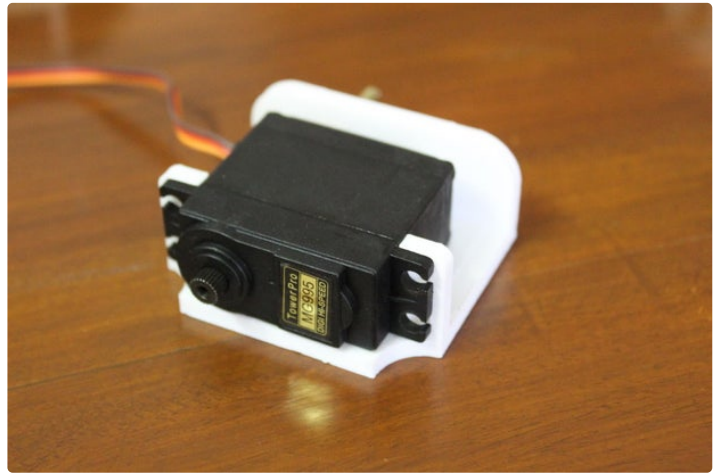
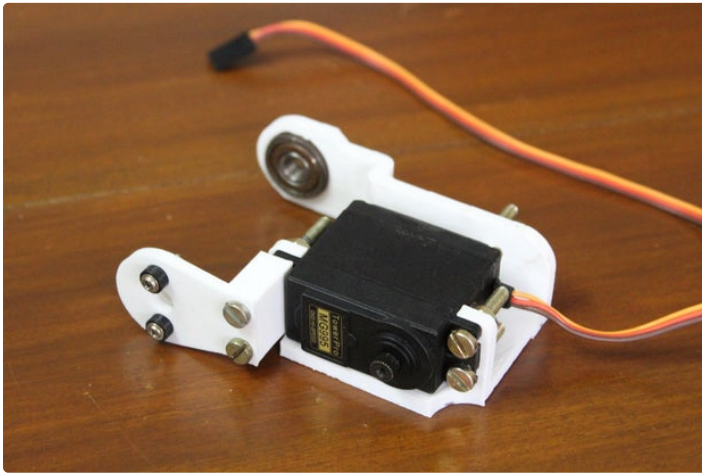


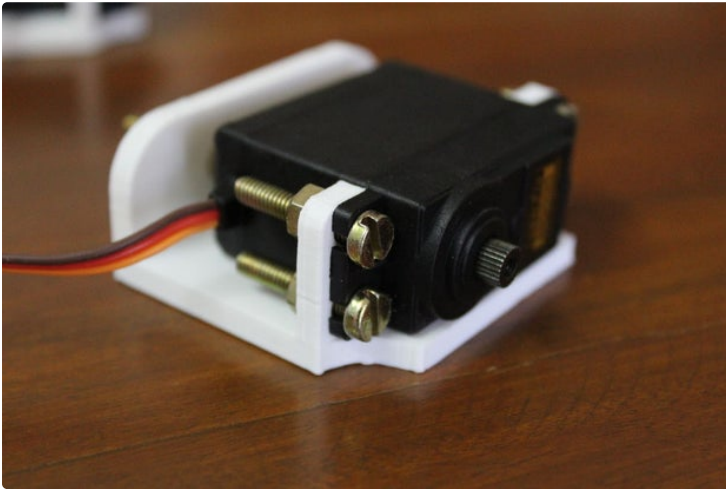
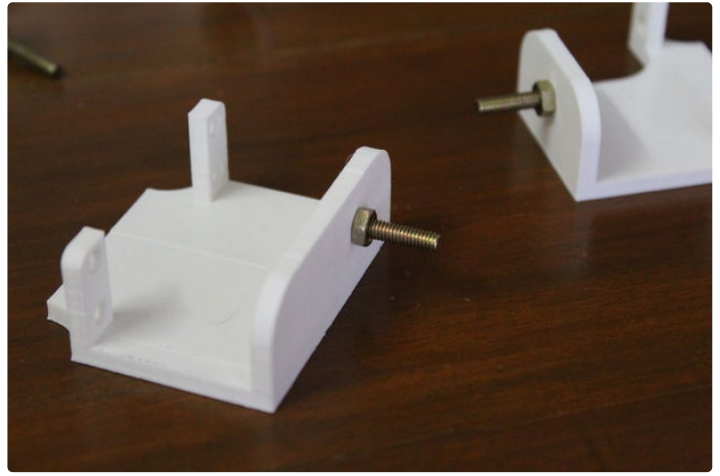
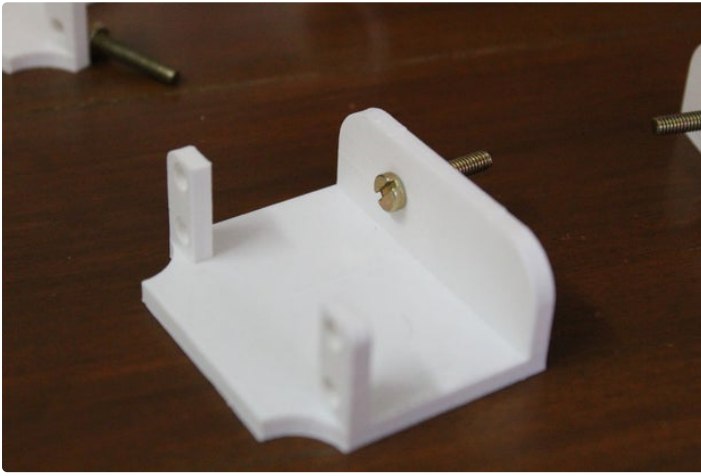
Once all of the parts are printed you can begin by setting up the servos and the servo brackets. First push in a bearing into the knee servo holder. The fit should be snug but I would recommend sanding the inner surface of the hole a bit instead of forcing the bearing which may risk breaking the part. Then pass an M4 bolt through the hole and tighten it using a nut. Next, grab the foot link and attach a circular servo horn to it using the supplied screws. Attach the foot link to the knee servo holder using the screws you will use to also attach the servo motor. Make sure

to align the motor so that the shaft is on the same side of the bolt you had attached earlier. Finally secure the servo with the rest of the nuts and bolts.

Do the same with the hip servo holder and foot servo holder. With this, you should have three servo motor and their corresponding brackets.

**Note:** I am providing instructions for building one leg, the other is simply mirrored.



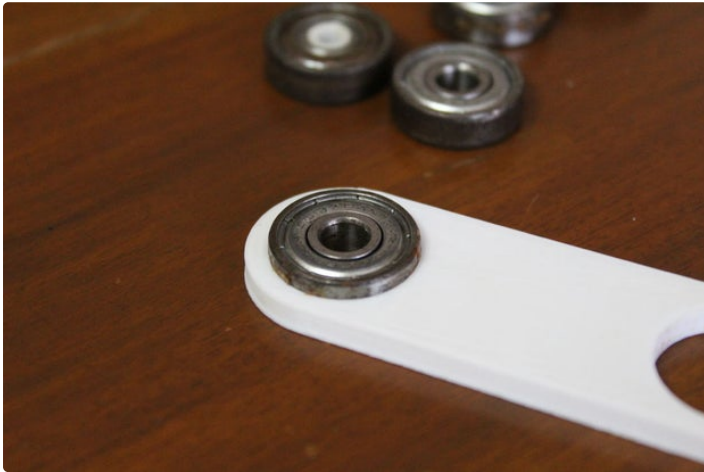
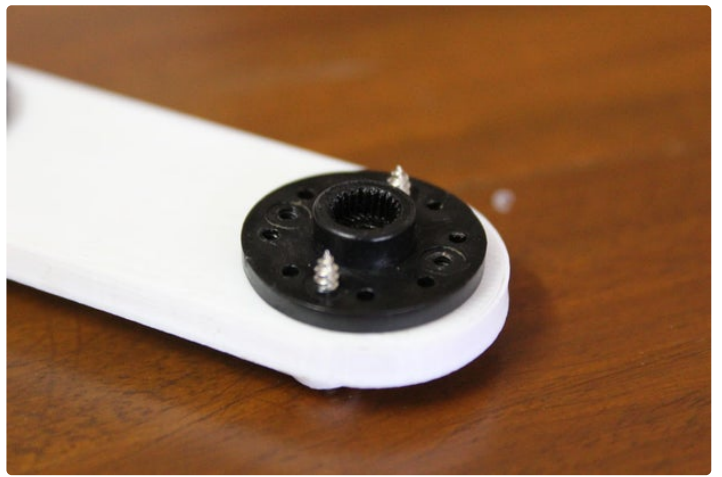
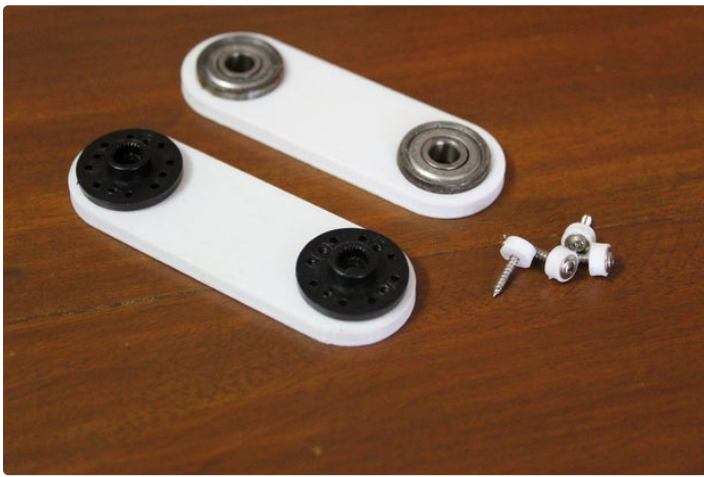


## Step 6: Making the Link Pieces

Once the brackets are assembled, start making the links. To make the bearing link, once again lightly sand the inner surface of the holes for the bearing then push the bearing into the hole on both sides. Make sure to push the bearing in till one side is flush. To build the servo horn link, grab two circular servo horns and the supplied screws. Place the horns on the

3D print and line up the holes, next screw the horn onto the 3D print by attaching the screw from the 3D print side. I recommend using a 3D printed servo horn spacer for these screws. Once the links are built you can begin the assemble the leg.





---

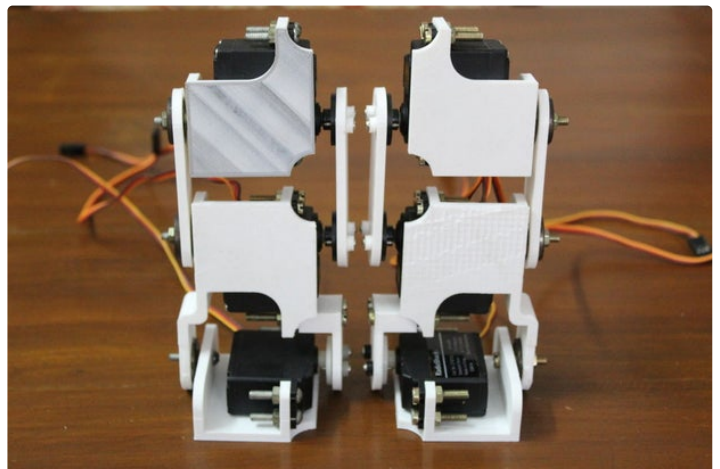
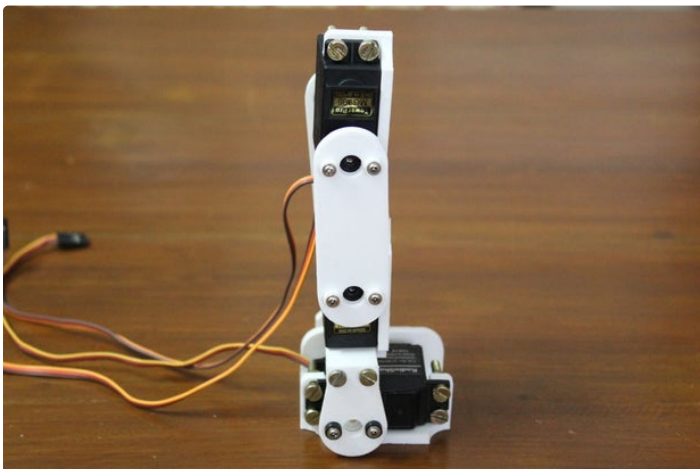
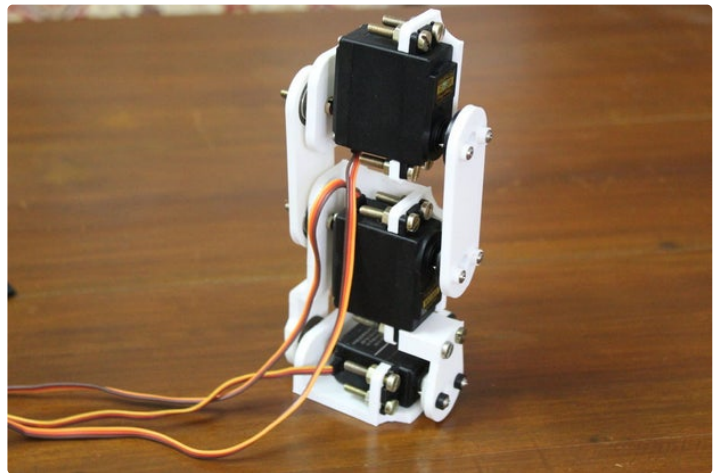
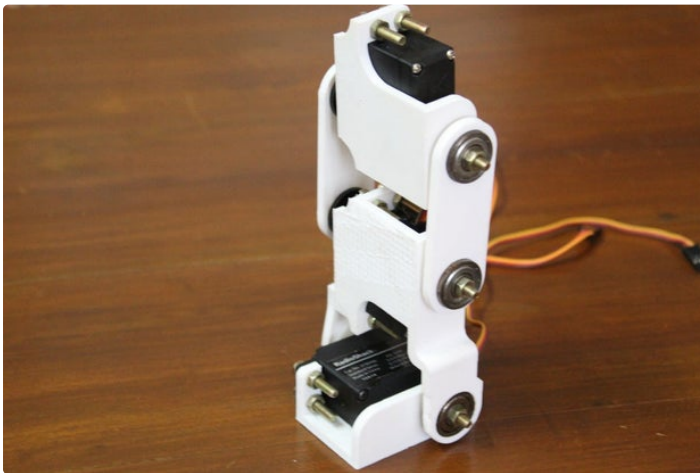
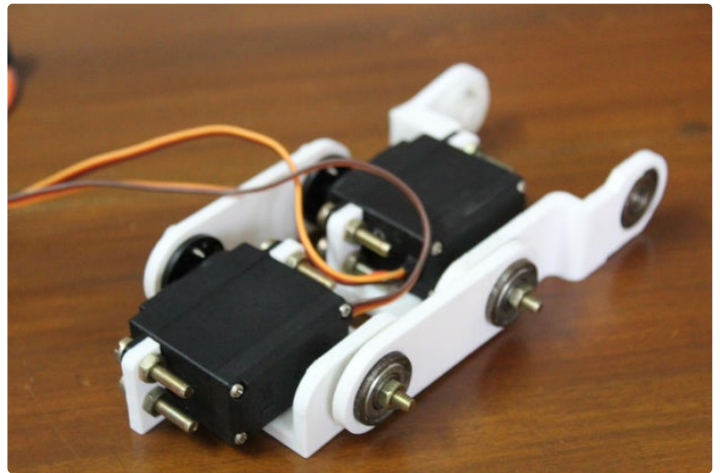
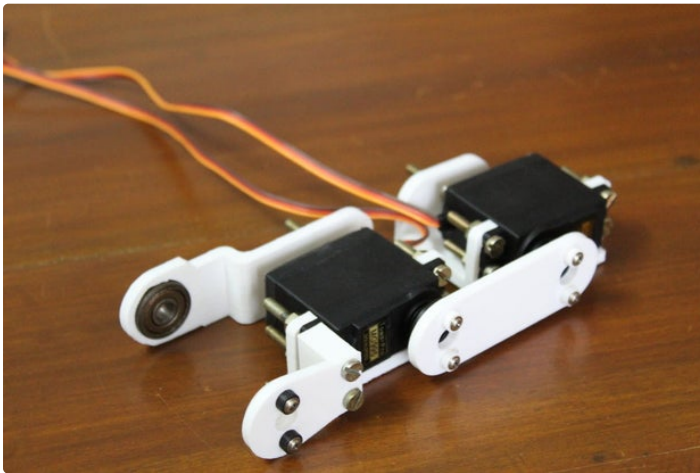
## Step 7: Assembling the Legs

Once the links and brackets are assembled, you can combine them to build the leg of the robot. First, use the servo horn link to attach the hip servo bracket and knee servo bracket together. **Note:** Don't screw the horn to the servo just yet as there is a setup stage in the following stage and it'll be an inconvenience if the horn were screwed onto the servo motor.

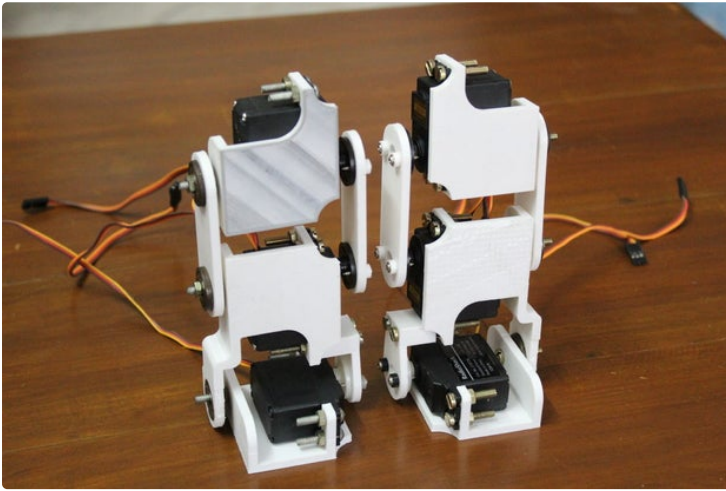
On the opposite side mount the bearing link onto the protruding bolts using nuts. Lastly, attach the foot

servo bracket by inserting the protruding bolt through the bearing on the knee servo holder. And fix the servo shaft to the servo horn connected to the knee servo holder on the other side. This may be a tricky task and I would recommend a second pair of hands for this.

Repeat the steps for the other leg. Use the pictures attached to each step as a reference.







---

## Step 8: Custom PCB and Wiring

This is an optional step. To make the wiring neater I decided to make a custom PCB using perf board and header pins. The PCB contains ports to directly connect the servo motor wires. In addition, I also left extra ports in case I wanted to expand and add other sensors such as Inertial Measurement Units or ultrasonic distance sensors. It also contains a port for the external power source required to power the servo motors. A jumper connection is used to switch between USB and external power for the Arduino. Mount the Arduino and PCB to either side of the electronics mount using screws and the 3D printed spacers.

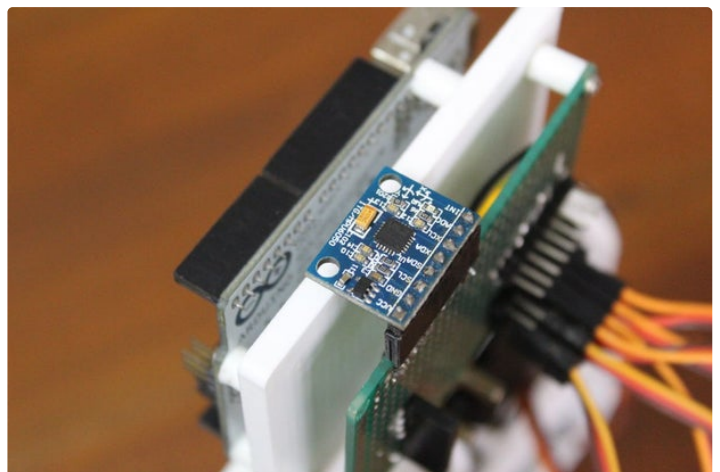
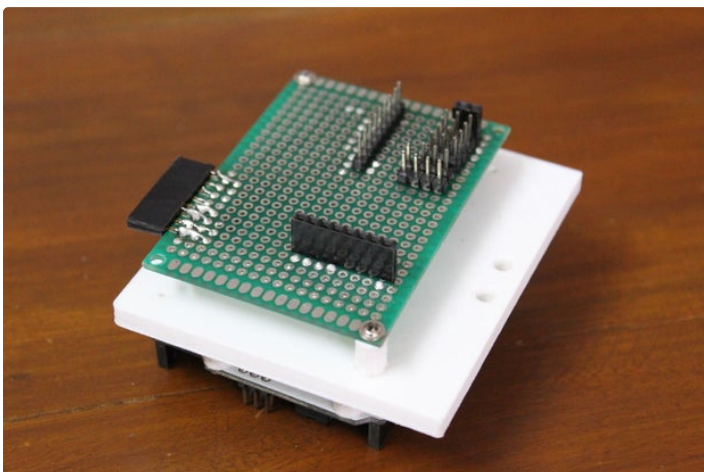
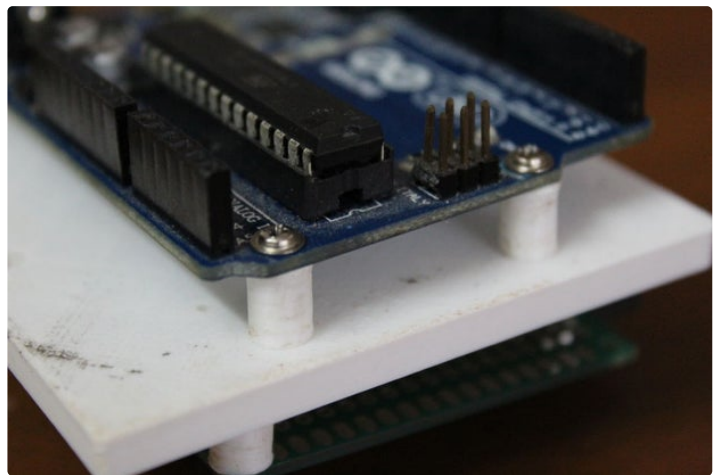
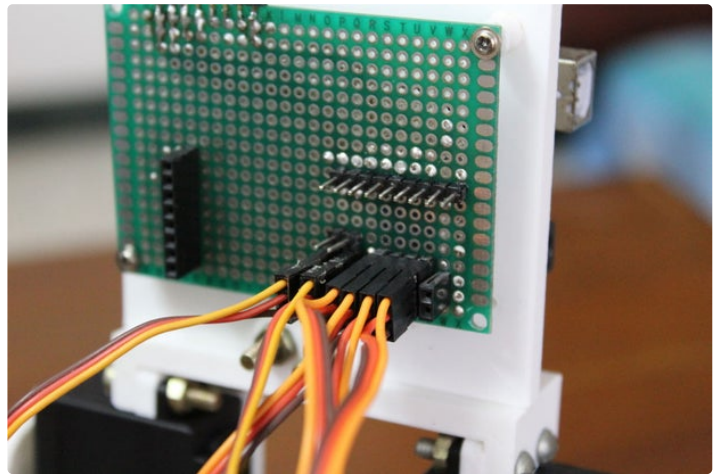
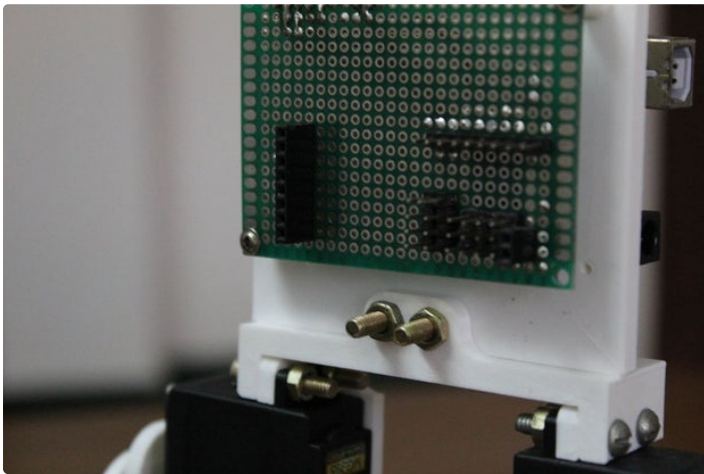
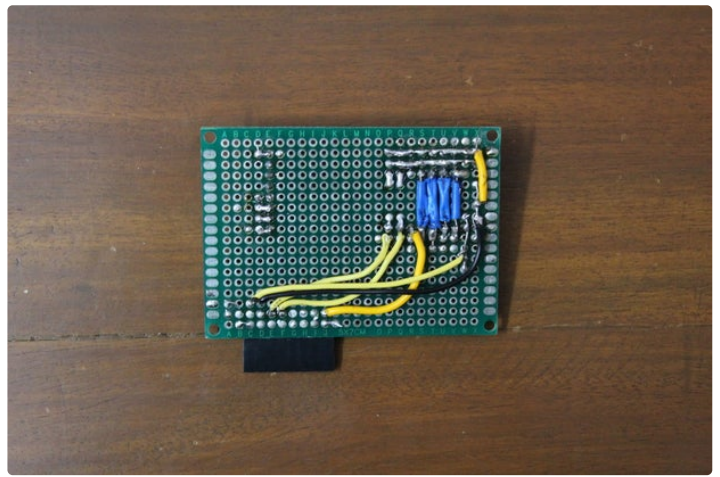
**Note:** Make sure to unplug the jumper before connecting the Arduino to your computer through USB. Not doing this may result in damaging the Arduino.

If you decide to not use the PCB and instead use a breadboard here are the servo connections:

- Left Hip >> pin 9
- Right Hip >> pin 8
- Left Knee >> pin 7
- Right Knee >> pin 6
- Left Foot >> pin 5
- Right Foot >> pin 4

If you do decide to make the PCB follow the same order as above by using the ports on the PCB from right to left with the IMU port facing up. And use regular male to female jumper wires to connect the PCB to the Arduino using the above pin numbers. Make sure to also connect the ground pin and create the same ground potential and Vin pin for when you decide to run it without USB power.





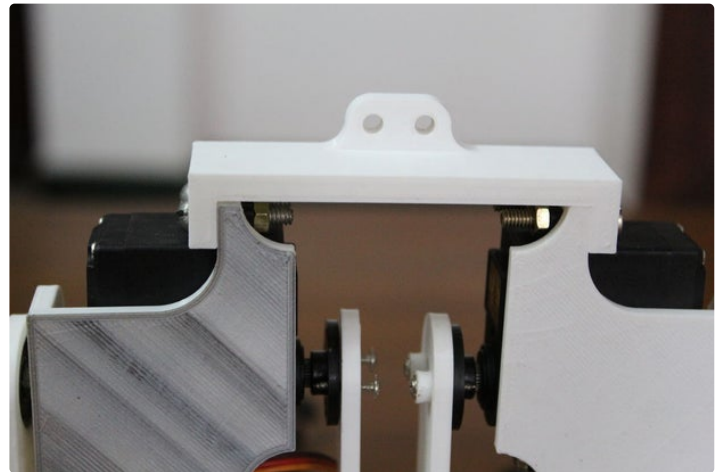
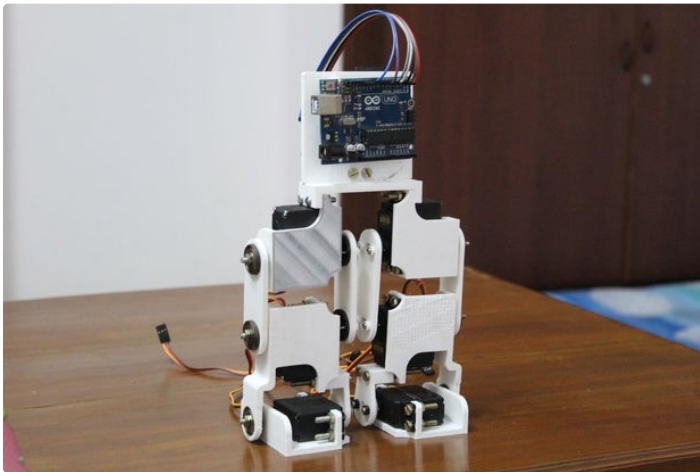
---

## Step 9: Assembling the Body

Once the two legs and the electronics are assembled, combine them together to build the robot body. Use the bridge piece to link the two legs together. Use the same mounting holes on the hip servo holder and nuts and bolts that hold the servo motor. Finally, connect the electronics mount to the bridge. Line up the holes on the bridge and electronics mount and

use M4 nuts and bolts to make the joint.

Refer to the attached images for help. With this, you have completed the hardware build of the robot. Next, let's jump into the software and bring the robot to life.



---

## Step 10: Initial Setup

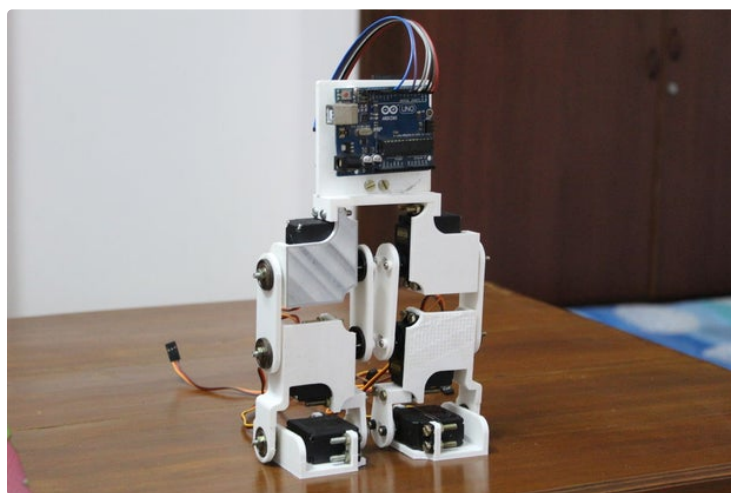
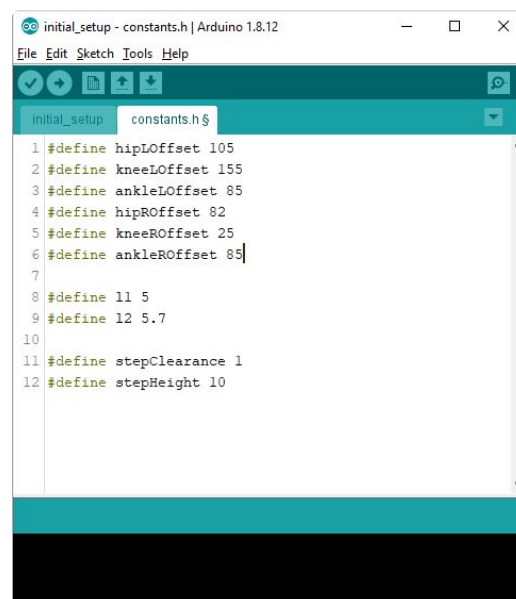
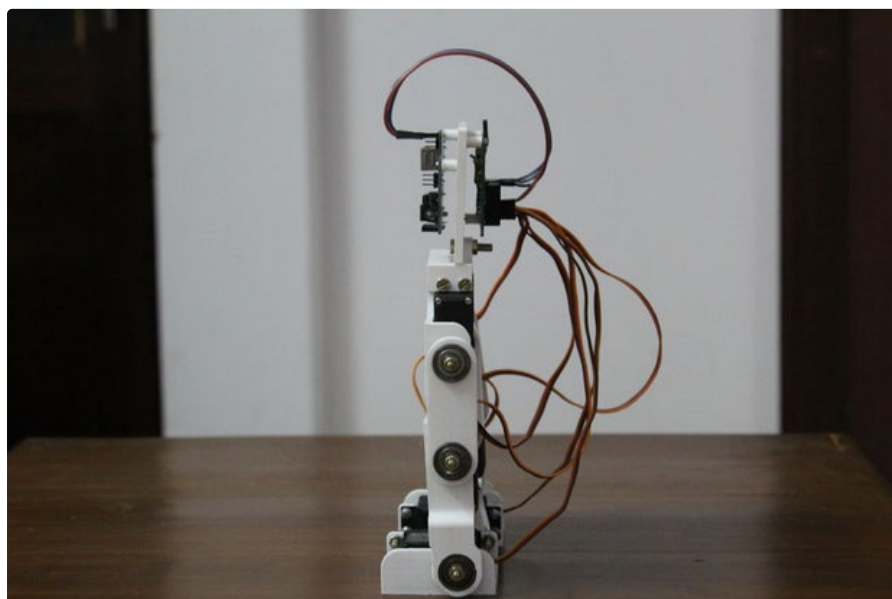


What I've noticed while building this project is that the servo motors and the horns need not align perfectly to stay relatively parallel. This is why the "central position" of each servo motor has to be manually adjusted to align with the legs. To achieve this remove the servo horns from each servo and run the **initial\_setup.ino** sketch. Once the motors have settled in their central position reattach the horns such that the legs are perfectly straight and the foot is perfectly parallel to the ground. If this is the case you are in luck. If not open the **constants.h** file

found on the adjacent tab and modify the servo offset values (lines 1-6) till the legs are perfectly aligned and the foot is flat. Play around with the values and you will get an idea about what is necessary in your case.

Once the constants have been set, note these values as they will be needed later on.

Refer to the pictures for help.



<https://www.instructabl...>

Download

## Step 11: A Bit About the Kinematics

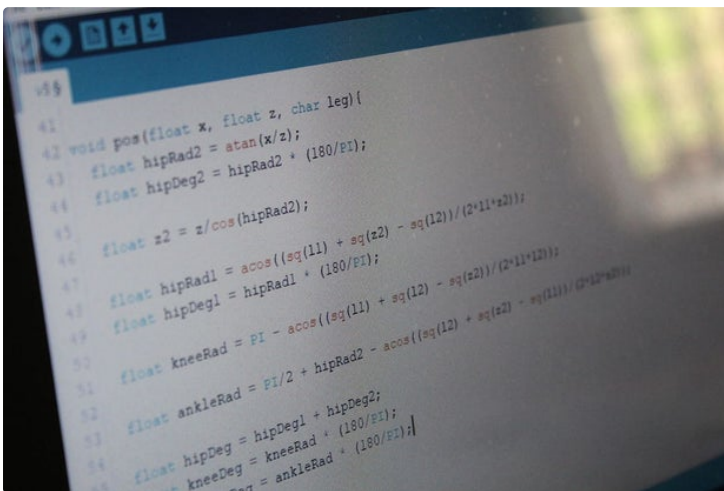
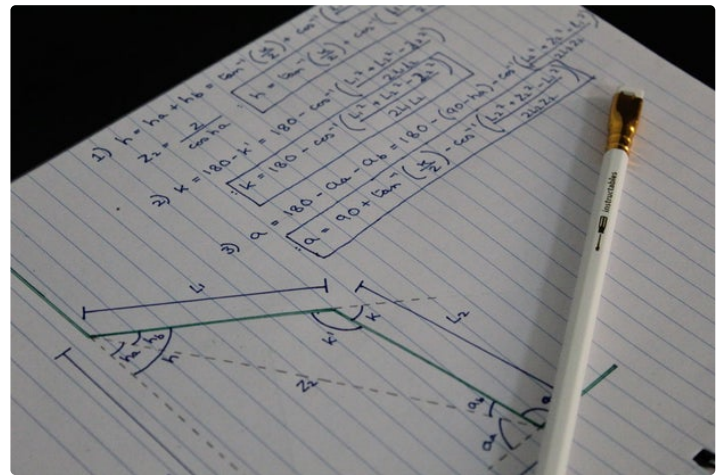
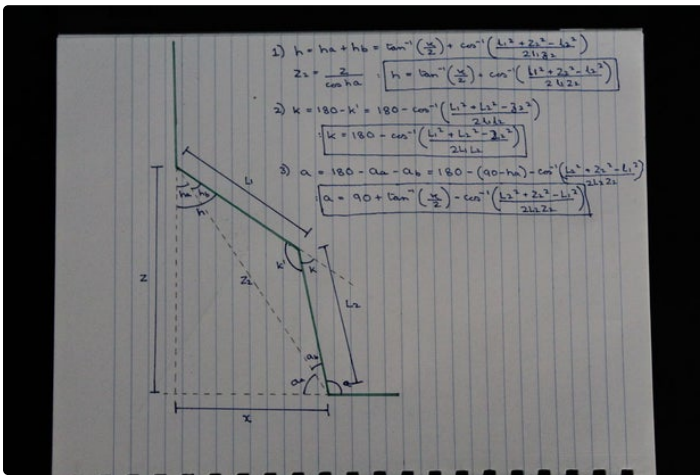
To make the biped perform useful actions such as running and walking the various gaits need to be programmed in the form of motion paths. Motion paths are paths along which the end effector (the feet in this case) travel along.



There are two ways of achieving this:

1. One approach would be to feed the joint angles of the various motors in a **brute force** manner. This approach can be time-consuming, tedious, and also filled with errors since the judgment is purely visual. Instead, there is a smarter way of achieving the desired results.
2. The second approach revolves around feeding the coordinates of the end effector instead of all the joint angles. This is what is known as **Inverse Kinematics**. The user inputs coordinates and the joint angles adjust to position the end effector at the specified coordinates. This method can be considered as a black box that takes as inputs a coordinate and outputs the joint angles. For those who are interested in how the trigonometric equations of this black box were developed can look at the diagram above. For those who are not interested, the equations are already programmed and can be used using the **pos** function which takes as input x, z and outputs three angles corresponding to the motors.

The program containing these functions can be found on the next step.



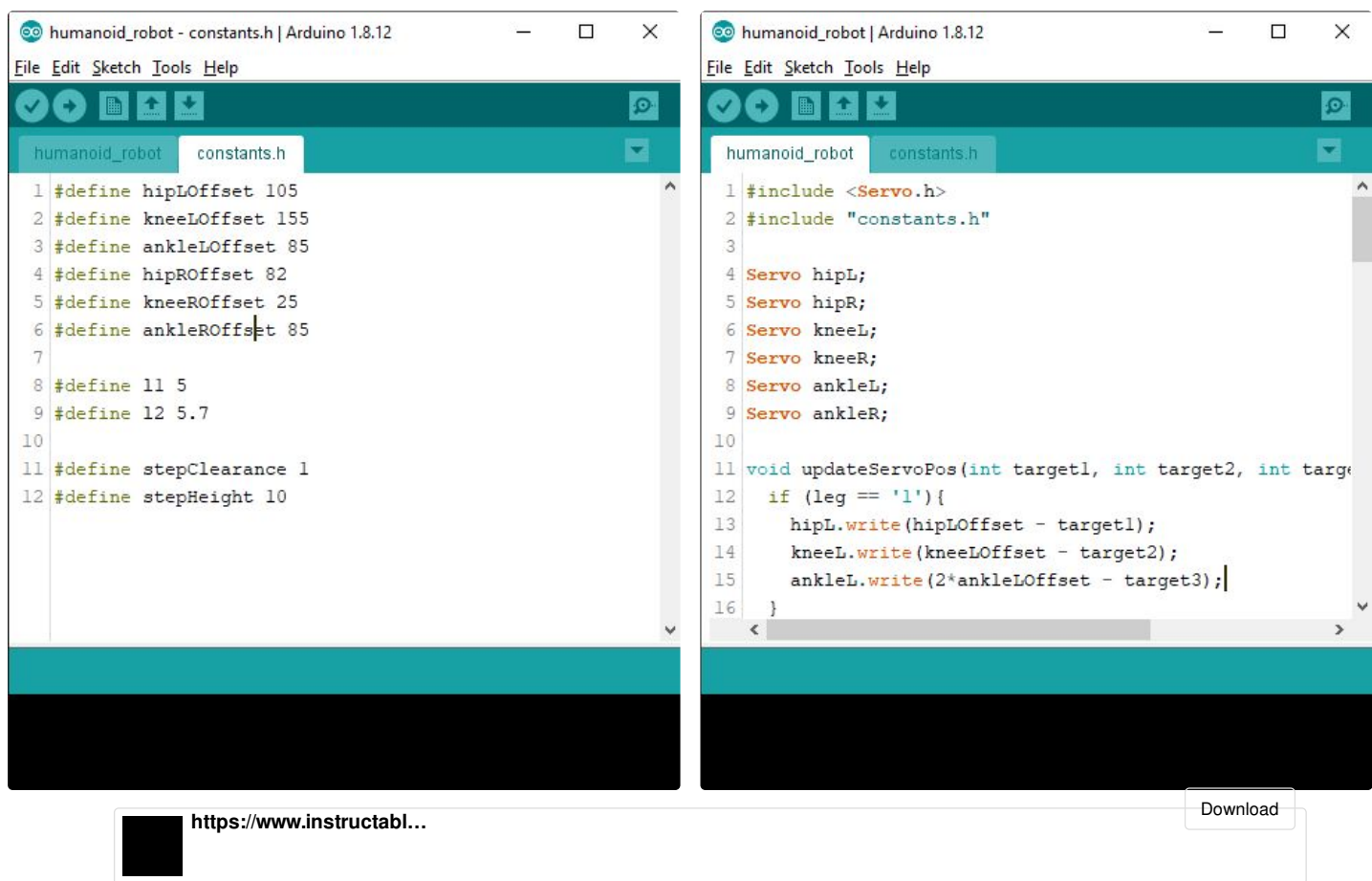
## Step 12: Programming the Arduino

Before programming the Arduino, slight modifications need to be made to the file. Remember the constants I had asked you to take a note off? Modify the same constants to the values you set in the **constants.h** file.

**Note:** If you have used the designs provided in this Instructable, you have nothing to change. In case there are some of you who have made their own designs you will have to change a few more values along with the offsets. The constant **I1** measures the distance between the hip pivot and knee pivot. The constant **I2** measures the distance between the knee pivot and ankle pivot. So if you designed your own model, measure these lengths, and modify the

constants. The final two constants are used for the gaits. The **stepClearance** constant measures how high the foot will lift while coming forward after a step and the **stepHeight** constant measures the height from the ground to the hip while its taking steps.

Once all the constants are modified according to your need, you can upload the main program. The main program simply initializes the robot into a walking stance and starts taking steps forward. The functions can be modified according to your need to explore the various gaits, speeds and step lengths to see what works best.



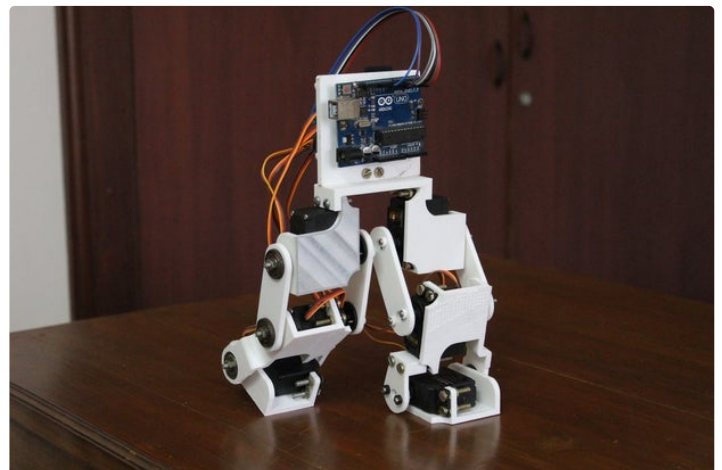
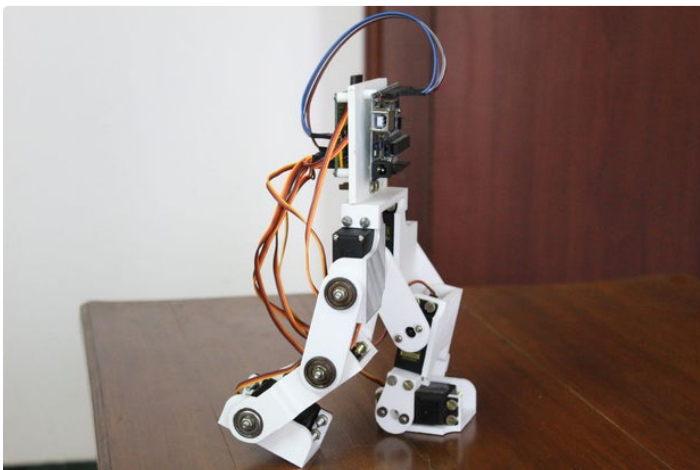
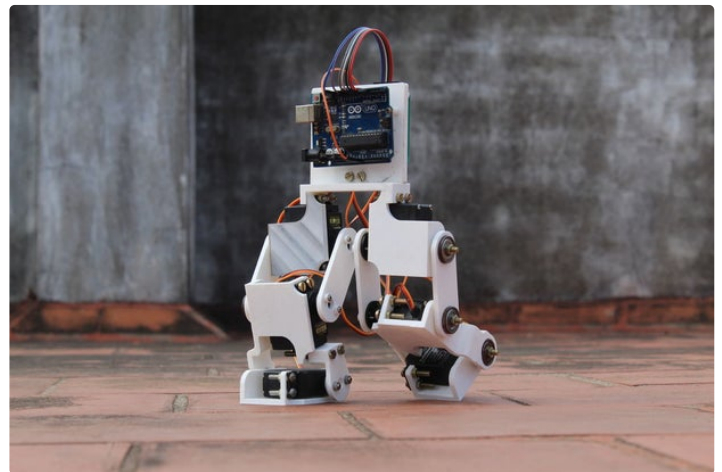
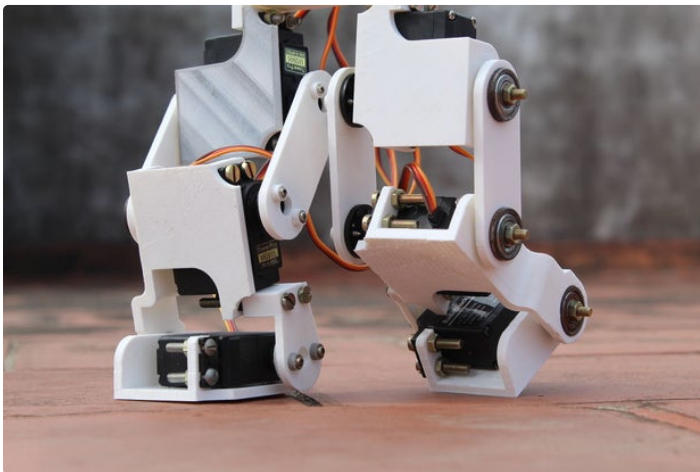
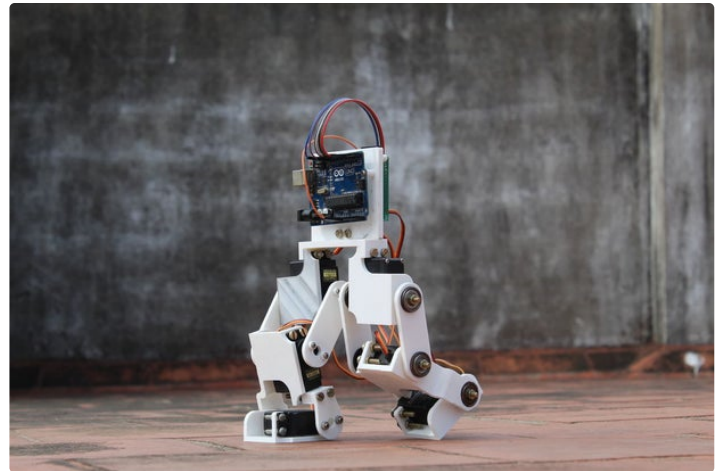
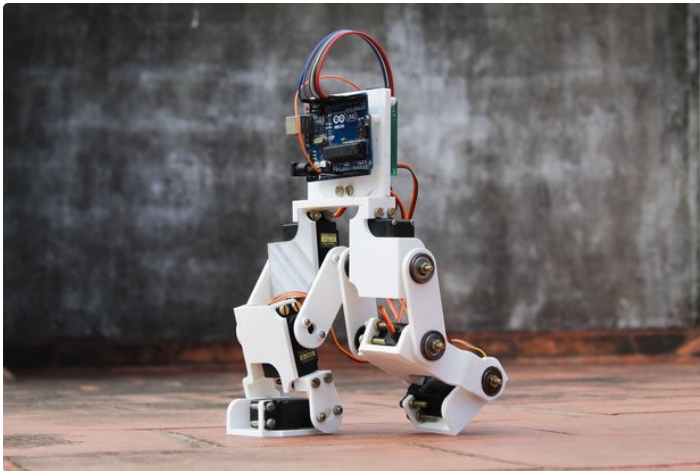
## Step 13: Final Results: Time to Experiment

The biped can take steps that vary from 10 to 2 cms long without tipping over. The speed too can be varied while keeping the gait balanced. This biped combined with the power of the Arduino provides a robust platform to experiment with various other gaits and other objectives such as jumping or balancing while kicking a ball. I would recommend you to try to change the motion paths of the legs to create your own gaits and discover how various gaits affects the performance of the robot. Sensors such as an IMU and distance sensor can be added to the system to increase its functionalities while force

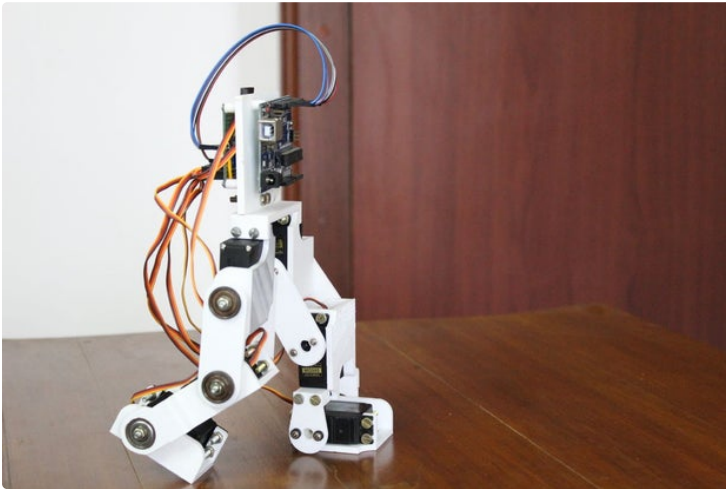
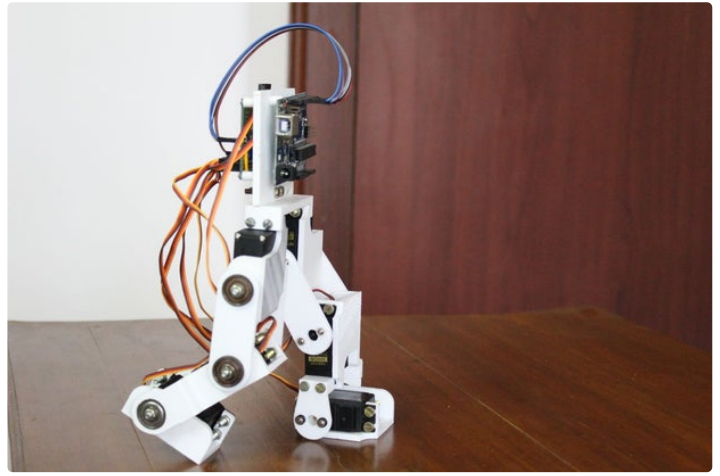
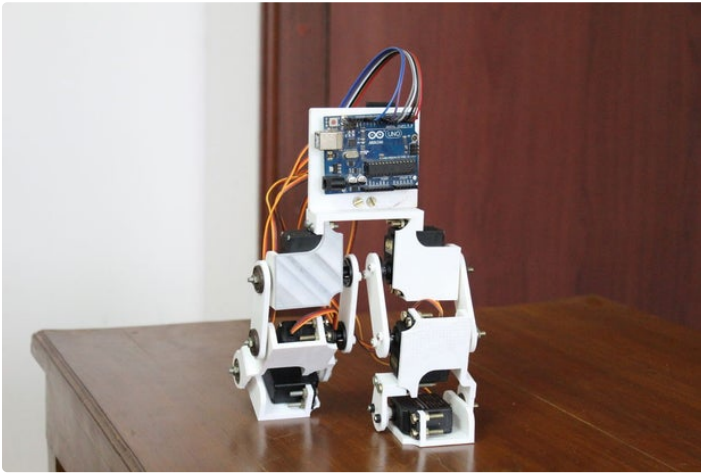
sensors can be added to the legs to experiment with dynamic locomotion on uneven surfaces.

Hope you enjoyed this Instructable and is enough of an inspiration to build your own. If you liked the project do support it by dropping a vote in the **"Arduino Contest"**.

Happy Making!







Update: Bearings in parts list changed from 22mm diameter to 19mm.



My robot is ready today! Behind 3 weeks of waiting for components from Aliexpress. A very interesting and exciting project for my 60 years old :)

Tomorrow I will start teaching him to walk, and today I just downloaded the installation program and adjusted the constants. I upgraded the movable units and made additional inserts for 4x10x4 bearings. This is a more rational solution than sawing holes up to 22mm. Also, I used a standard Arduino UNO expansion board to connect the servos.

It should be noted that the names of the legs are not indicated as in humans. The left leg in the program is the one on the left when you look at the robot's "face" in the photo. In my opinion, this is a gross mistake. I'd love to know how to use the MPU6050 IMU. Maybe the author will share this information?

In general, I find the author's work very useful. I express my deepest gratitude and respect to him.



My robot is ready today! Behind 3 weeks of waiting for components from Aliexpress. A very interesting and exciting project for my 60 years old :)

Tomorrow I will start teaching him to walk, and today I just downloaded the installation program and adjusted the constants. I upgraded the movable units and made additional inserts for 4x10x4 bearings. This is a more rational solution than sawing holes up to 22mm. Also, I used a standard Arduino UNO expansion board to connect the servos.

It should be noted that the names of the legs are not indicated as in humans. The left leg in the program is the one on the left when you look at the robot's "face" in the photo. In my opinion, this is

a gross mistake. I'd love to know how to use the MPU6050 IMU. Maybe the author will share this information?

In general, I find the author's work very useful. I express my deepest gratitude and respect to him.



where can i found the code?

FINALLY A HUMAN HAS DESIGNED LEGS FOR ME!!!

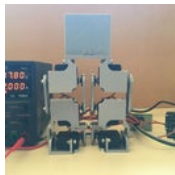
Haha! Next the brains...

Oh yes!

Sir can we make it without the perfboard

Mine's done! Need to mount the PWBs on the head and figure out why the stride isn't as smooth as in your video, but that's why I like it . . . very stimulating and great learning experience! Thanks for posting and referencing Fusion 360 . . . I'm going to give it a try also.

BTW, I think there's an error in the parts list . . . bearings should be 8x19x6 not 8x22x7 . . . the 19mm ones fit much better in the 19mm hole :-)



I was disappointed in the amount of material I have to remove to get the bearing to fit.

To make this easier and to keep the hole circular, I used an adjustable hand reamer. Be careful when enlarging the bearing hole in the knee servo holders. Not much meat there at all, very easy to break:(

In the picture for step #4, next to the bridge, is what looks like a cylinder with a tab. I don't see this piece in any of the .stl files. What is this piece for?

I wondered also, but if you watch the video it's to suspend the robot while experimenting with leg movements. I will be building one also. ( my last robot went down the basement stairs by accident.lol)

Yes the mount is used during the testing phase, safety measure if anything goes wrong!

Please provide a link to the .stl file for this test fixture. It was not included in the zipped file (stls.rar)

making it, thanks for this!

Great, good luck!

Great project and great job explaining the kinetics. Very informative.

Thanks for the positive feedback.

They are looking phenomenal. Excellent photoshoot.Thanks for sharing.

Thanks!

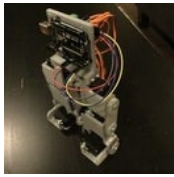


Great work!



Awesome project For my 8 yr old! Mine so far, working on the programming, should be walking soon!

i looked online for a while for something like this. This has to be the best project online right now. Thanks!



Hi, I am in the process of building your robot legs but am stumped with the connections on the perf board and the connections from the perf board to the UNO.

Would you be able to supply a connections drawing or schematic of the perf board and pin to pin between the perf board and the UNO?



You have provided a very good explanation of your robot and your build steps including lots of pictures. You also demonstrated a degree of professionalism that we talk about with our students. Students will also see the importance of being able to communicate well with technical writing. I believe Fusion 360 is free to students and it is one of the preferred cad/cam programs for today (search for fusion 360 free). There are tons of tutorials on YouTube for learning how to design with it but nothing helps more than providing a finished design like the one you created here (perhaps you might consider creating a fusion tutorial!) Also, 3D printers are pretty cheap now (see sainsmart, etc) and they make a good learning and prototyping tool to use at home. I am certain there are some FIRST robotics teams that might enjoy the challenge of creating this design during their 2020 season cancellation. It's small enough and uses inexpensive parts making it the perfect project. Again, thank you for submitting a great project and for sharing it with all of us!



Absolutely! This project is fun and enjoyable to work on and one can learn about various fields through it such as design, 3D printing, electronics, kinematics etc. It is perfect for beginners as well as for people with some experience. Thank you so much for your feedback!



I will be trying this soon once i have the 3D printer and accessories



Absolutely, go for it, its an enjoyable process!



It's an awesome project! I just want to replicate it!

But I have a question: Can it turn or it only can go forward and backwards?

And an idea: You can add hands to build a full humanoid robot!



Thank you! Yes, it can turn by taking bigger steps with one leg and smaller with the other. Adding hands would also be a great addition



very very very nice ...

Have you already thought about whether there is a technique to rotate left or right?



Thank you, to rotate the robot could take bigger steps on one side and smaller on the other.



nice and cool work



Thanks for the feedback!



Would love to do this with my robotics class, any chance I can get a .zip file of the 3D printed parts and the Arduino code?



It's already been added in the steps. If there is any troubling downloading it, I could mail them





I'm going to build one with my grandchild, she is really into robots!



Thanks, would love to see it!



Great article! May I know what kind of power supply are you using? I'm currently working on 6-axis DOF robot arm with 6 pieces MG996r servos. I'm kinda newbie so i'm having a hard time to power them up. Here's the image of what i am working..



One reason you may be having a problem with power supply is a "start up spike". When the robot powers up, all 6 servos are most likely way out of position. It is not uncommon for a servo to draw a peak of 2 amps when making a large move, so that would be 12A for your robot, most likely overwhelming the supply, and causing Arduino to crash and reboot, repeating the whole sequence forever...

The way to solve this is to put in the software a start up routine that starts the PWM one servo at a time, spaced 1 second apart, in sequence. This staggers the start-up spikes so does not overwhelm the supply. Also in the software, it is wise to put a slew-rate limiting routine which prevents sudden large movements of a servo (especially when debugging the movement calculation routines). I have flowchart for such a routine if you are interested.



Thank you, for testing purposes a variable lab bench power supply was used



I see. Thanks for the reply



.rar files require a paid program to expand. zip is more universal and free.



7 Zip is free and can unpack rar files. Also WinZip works.



Use free 7-Zip .. <https://www.7-zip.org/> .. it opens most compressed files inc zip, rar etc..



can you tell me about the hole for the bearings, why is the hole 3mm smaller then the bearing6?



It is so that the bearing can friction fit properly into the pocket.



Exceptionally well done. It's not clear to me how it's powered.



Thank you. At the moment, it's powered with a variable power supply.



Voted for you, this is great work! Great to see Fusion 360 being used