Building Instructions for APQuad 3DoF models

Alex Pylypenko (macaquedev)

NOTE:

These will work for ANY model of APQuad with ANY servo (other than 2DOF versions). The parts may look slightly different depending on which model you have, but please rest assured that the assembly WILL be correct no matter which servo is used. However, the Electronics part is slightly different for 9g servos.

DISCLAIMERS:

I hold NO responsibility whatsoever for any damage done or injuries caused by carelessness or incorrect assembly.

This is presented ‘as is’ with ABSOLUTELY NO WARRANTY.

Mechanical Assembly:

3D printed parts required (can be commercially printed):

1x Base Plate.stl

4x Foot.stl

4x Leg.stl

2x Hip.stl

2x Mirrored Hip.stl

These files are located in the ‘To 3D print folder’

Optional:

1x Dome.stl

3x Mount for dome.stl

Mechanical parts required:

12x Servo motors (please use the servo you chose the designs for)

12x M8 washers

15x 10mm M3 screws (12 if not using dome)

Tools required:

Screwdriver (type dependent on the screws you are using)

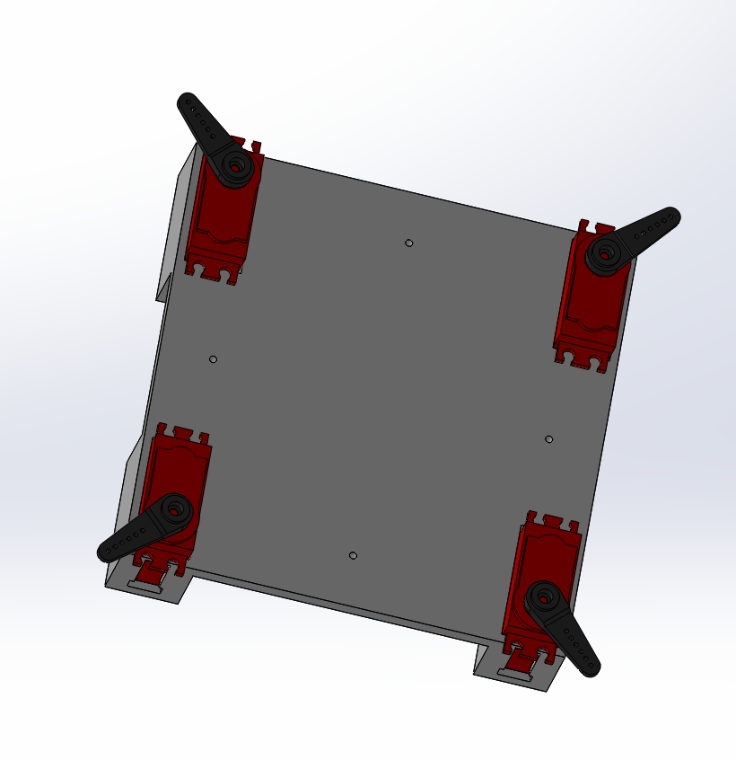
3D printer

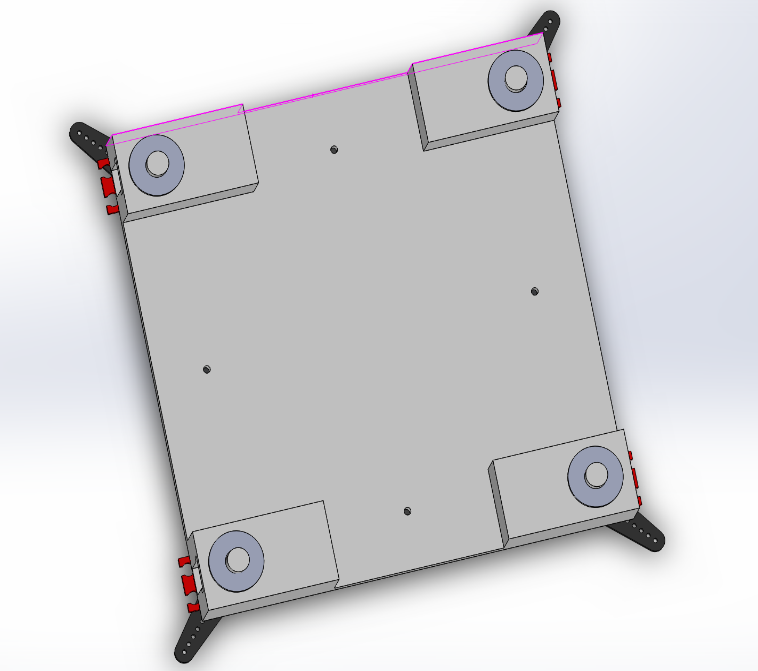
Prusaslicer or equivalent slicer (unless printed commercially)

Note before assembly: if any of this is unclear, please go into:

Building Instructions/3D Models and then see 3D models.

Instructions:

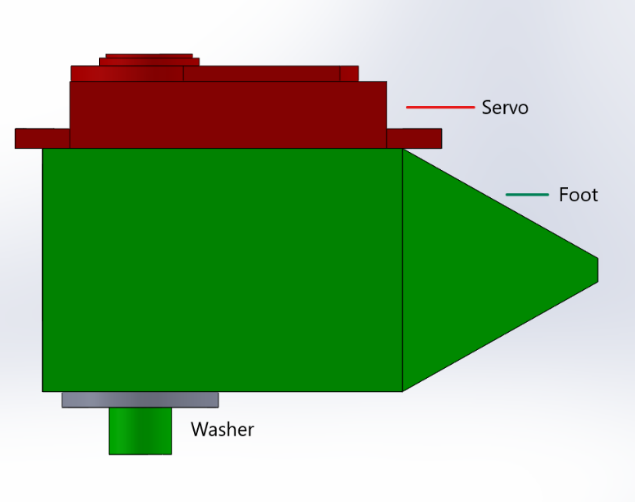
1. Assemble the base plate as per the images below



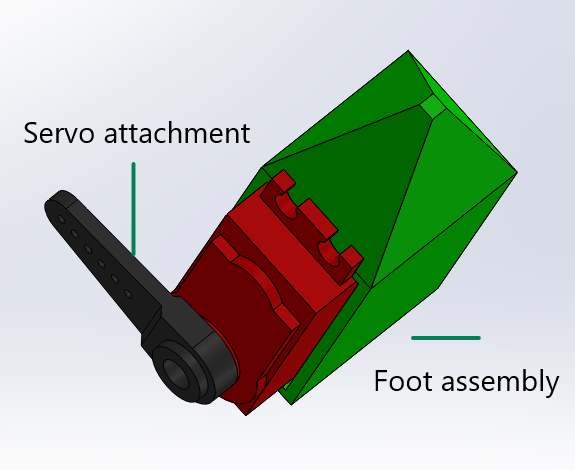
This step requires 4 servos, 4 washers, 4 attachments and a Base Plate.

This step is rather self-explanatory. However, before attaching the servo attachments completely, please attach them so that the angles shown on the image are the midpoints of the servo’s full turn angle so that the Arduino command *moveServo(blh, 90);*  moves the back left hip to the position illustrated in the diagram above (top left).

1. Assemble all the feet as per the image below.

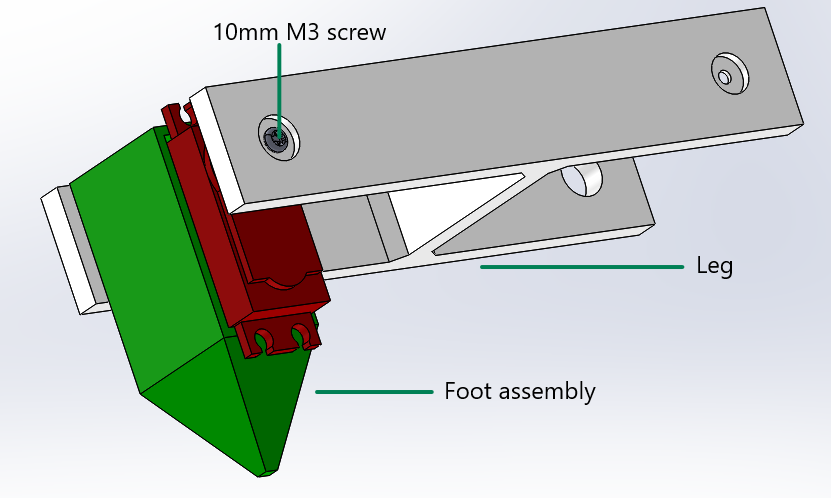


1. Add the attachments onto the servos. Please follow the instructions in step 1 as to what to do with the attachments (regarding midpoints etc.), however, please make 2 feet as in the diagram below and 2 feet with the attachment 180 degrees from where it is in the diagram below. Please keep the ones that are as per the diagram away from the mirrored ones and please remember which ones aren’t mirrored and which ones are.

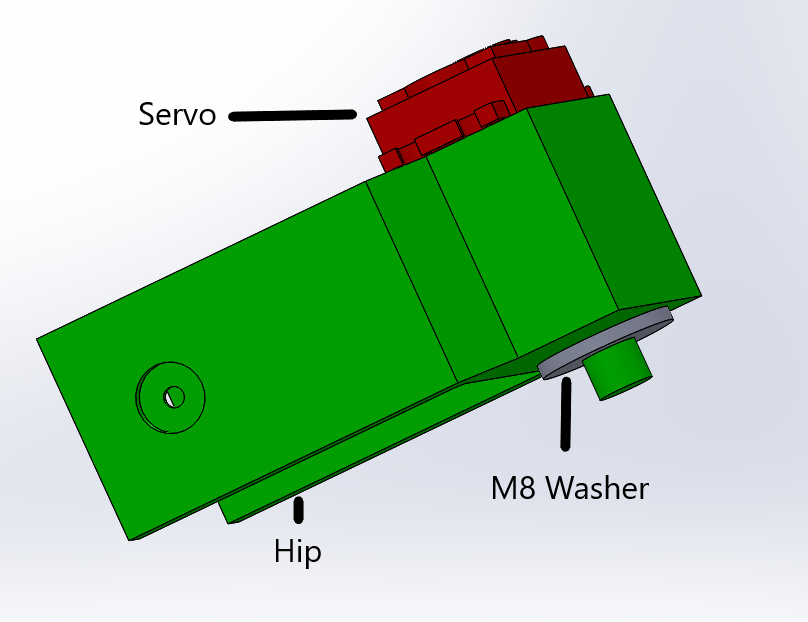
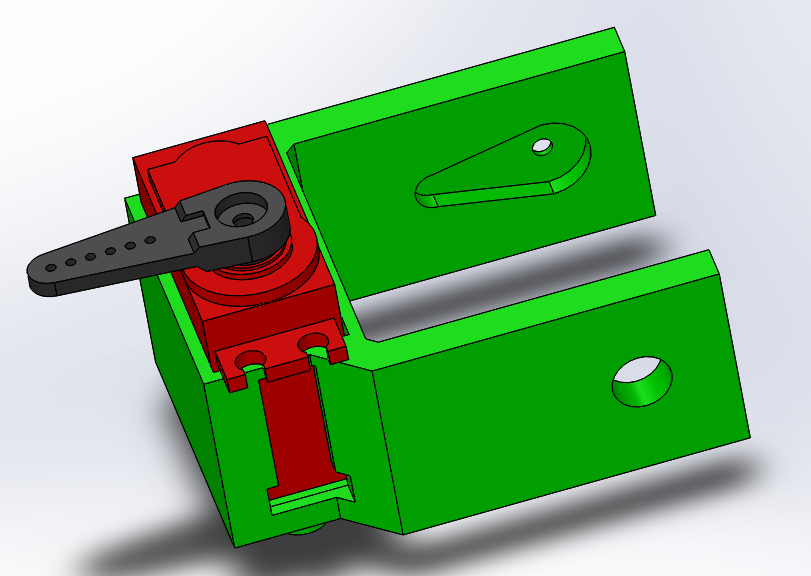


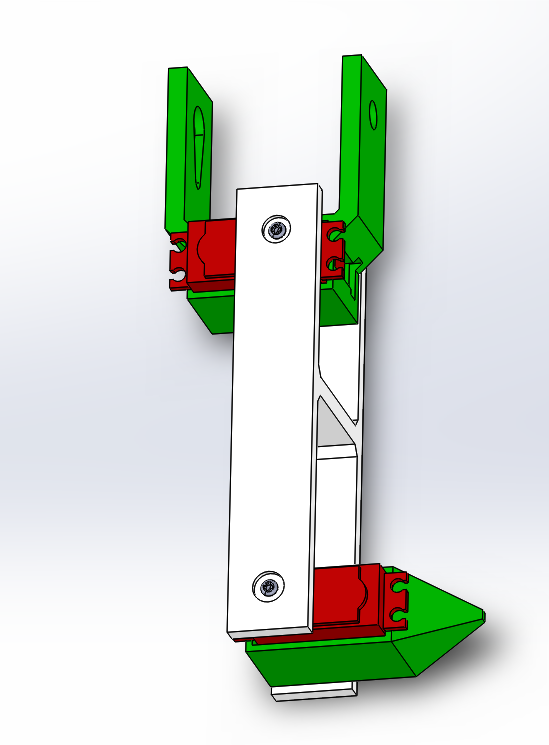
1. Add the legs to the foot assemblies. The mirrored ones and the non-mirrored ones will look slightly different, but that is totally correct.

Please make sure that the servo attachment goes into its designated hole. Please put it into the LONGER end of the leg.



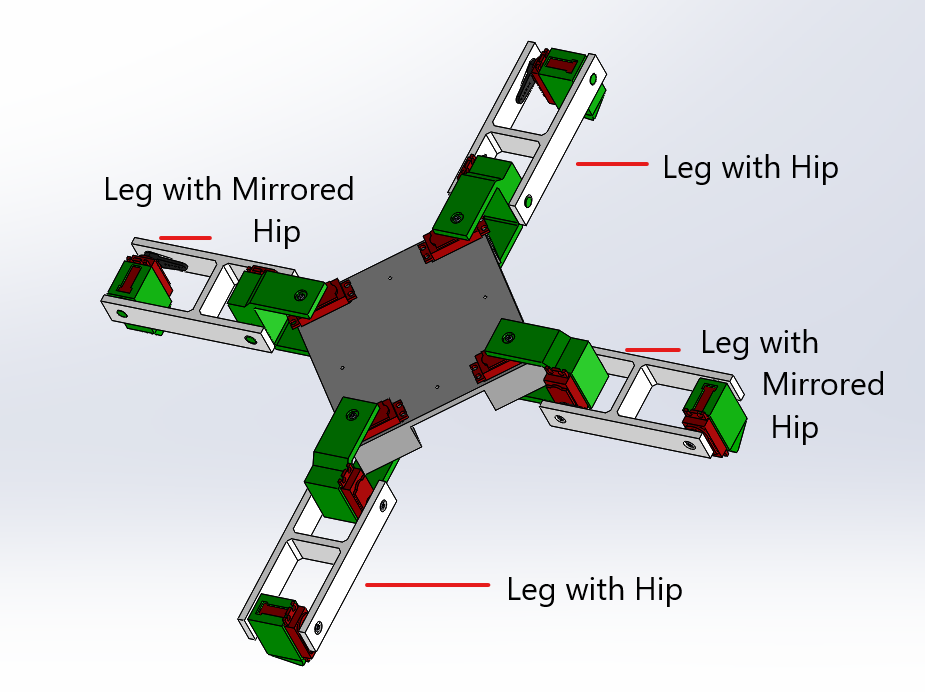
1. Assemble the two hips and the two mirrored hips as shown in the diagrams below. Again, the angles shown in the diagram should be the midpoint of the servo’s angle of rotation.



1. Insert the hips into the non-mirrored legs and the mirrored hips into the mirrored legs into the other ends of the leg. Please see diagram on how this should look. If it doesn’t look right, you have probably used a non-mirrored leg with a mirrored hip.
2. Before finishing the mechanical assembly of the APQuad, make sure that you have the following:
   1. 2 Legs with normal Hips
   2. 2 Legs with Mirrored Hips
   3. A fully assembled Base Plate as shown in Step 1

If not, please reread the manual. As an ABSOLUTE LAST RESORT, please email [macaquedev@gmail.com](mailto:macaquedev@gmail.com) with DETAILED information about your issue. Please include photos if possible. I will be very happy to answer any questions but please consider the fact that I am very busy. I will try to respond to any emails within 24 hours but if this isn’t the case, please do not spam me with emails.

1. If you have completed the checklist in question 7, please finish assembling your APQuad as shown below. The front of the APQuad is on between the top 2 legs on the image.



This is the end of the mechanical assembly. Please carry on with the electronic assembly.

Electronics Assembly (for non-9g servos):

Parts required:

Arduino Nano

PCA9685 servo multiplexer

A 3s LiPo battery

A load of jumper wires of all types

Thick wire (for high amperages)

Thinner wire (for low amperages)

R1966ablkblkff switch

10A 150W DC DC stepdown converter

LM2596S stepdown converter

A piece of PCB

Tools required:

Glue Gun

Soldering iron

Multimeter

NOTE: THIS PART IS QUITE FIDDLY AND IT IS VERY EASY TO DO SOMETHING WRONG. LITHIUM BATTERIES ARE VERY FLAMMABLE AND CAN CAUSE A MAJOR FIRE IF NOT CONNECTED PROPERLY.

Connect both converters in parallel to the battery through the switch. Check that output voltage of both converters is exactly 5V when supplying battery’s output voltage. Connect the Arduino to the 2A converter via VIN and the servo board to the 10A converter via the terminal block.

Solder 3 rows of roughly 10 female pin headers. Connect them together underneath using a soldering iron. Connect 1 row to Ground, 1 row to 3.3v and 1 row to 5v

Electronic Assembly for 9g servos:

Parts required:

Arduino Nano (use a clone to save money)

PCA9685 servo multiplexer

A 2s LiPo battery (I salvaged one from an old laptop but any will do)

A load of jumper wires of all types

Thick wire (for high amperages)

Thinner wire (for low amperages)

R1966ablkblkff switch

2x LM2596S stepdown converter

A piece of PCB

Tools required:

Glue Gun

Soldering iron

Multimeter

NOTE: THIS PART IS QUITE FIDDLY AND IT IS VERY EASY TO DO SOMETHING WRONG. LITHIUM BATTERIES ARE VERY FLAMMABLE AND CAN CAUSE A MAJOR FIRE IF NOT CONNECTED PROPERLY.

Connect both converters in parallel to the battery through the switch. Check that output voltage of both converters is exactly 5v when supplying battery’s output voltage. Connect the two converters’ outputs together. Connect both the Arduino (via VIN) and the servo board (via terminal holes) to the converters’ outputs.

Solder 3 rows of roughly 10 female pin headers. Connect them together underneath using a soldering iron. Connect 1 row to Ground, 1 row to 3.3v and 1 row to 5v

Electronic Assembly continued for ALL servos:

Connect the servo board to the Arduino:

|  |  |
| --- | --- |
| Servo Board | Arduino (or power rail if applicable) |
| GND | GND |
| VCC | 3.3v |
| SCL | A5 |
| SDA | A4 |

Connect the servos to the Servo Board:

B=back, F=front, L=left, R=right, A=ankle, K=knee, H=hip

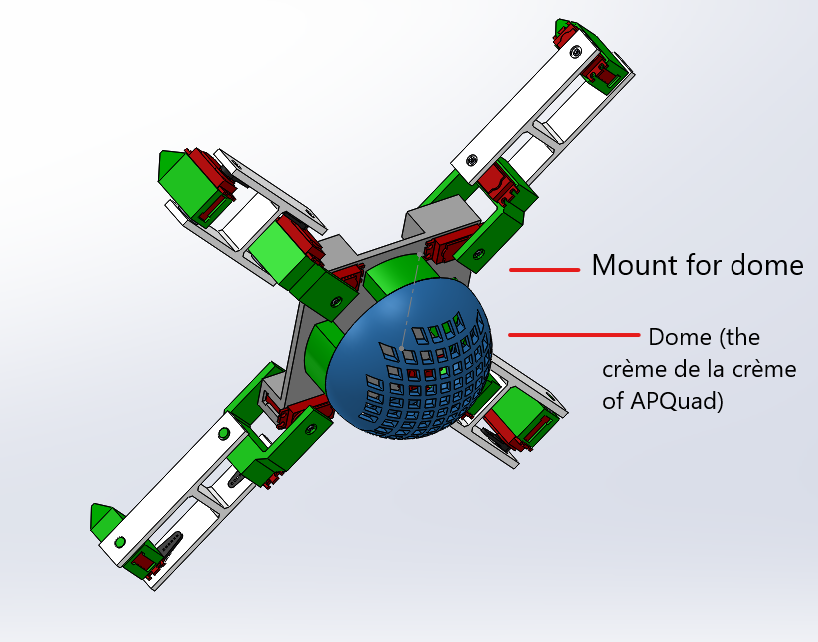
|  |  |
| --- | --- |
| Servo | Servo |
| FLA | 0 |
| FLK | 1 |
| FLH | 2 |
| FRA | 3 |
| FRK | 4 |
| FRH | 5 |
| BLA | 6 |
| BLK | 7 |
| BLH | 8 |
| BRA | 9 |
| BRK | 10 |
| BRH | 11 |

This is it for the Electronic Assembly.

NOTE: It is recommended for the Arduino Nano’s USB to be facing the back of the robot and to be easy to access from outside.

Adding the Dome (optional):

This step is extremely self-explanatory. Note – the part missing a Mount should be where the Arduino Nano’s USB port is facing. This is done so that the USB port is easily accessible.



Uploading the Firmware:

Please install the latest version of the Arduino IDE (do not use the web editor).

Go to Sketch 🡪 Include Library 🡪 Add .ZIP library and then browse for APQuad.zip.

Then, when you need to use APQuad commands, use the ones below.

Also, examples are provided. Go to File 🡪 Examples and then scroll down to custom libraries. You should see APQuad. Go to demo. It is a good representation of what your APQuad can do. The possibilities are endless!

It is really that simple!

For more experienced devs: I challenge you to create another APQuad function and send it to me so I can include it in the APQuad.h file with your name attributed to it!

Criteria:

1. To move a servo, please use my moveServo function
2. Please look at the APQuad.h file in the library and see how it is laid out.
3. Please don’t pass more than 4 arguments to a function – it simply overcomplicates the design.
4. Please don’t litter the global namespace using #defines or global variables!
5. NOTE: THIS IS NOT REQUIRED: If possible, please send me a video to [macaquedev@gmail.com](mailto:macaquedev@gmail.com) of your APQuad doing that manoeuvre.

Commands that you can use to move the APQuad are on the next page:

APQuad q(ServoType, I2C\_ADDRESS); // replace q with anything you want to call your APQuad object. For ServoType, use the namespace ServoTypes with the servo you are using. For example, ServoTypes::MG996R or ServoTypes::SG90. I2C\_ADDRESS is the I2C\_ADDRESS of your board. Please leave blank if unsure as it will default to 0x40 and work 99 percent of the time. Please just use “APQuad q;” if you are using SG90 servos want to be safe. For the rest of the commands, replace the initial q with your APQuad object’s name. Or just stick with q in the whole program to make life easier!

q.initializeQuad(); // This is mandatory in void setup() otherwise the quad won’t walk.

q.unfold(speed\_lower\_is\_faster); // Unfolds the robot, folded() returns false

q.fold(speed\_lower\_is\_faster); // Folds the robot, folded() returns true

q.folded(); // returns true at beginning of program as it assumes that the robot was folded while turned off. returns

q.stepForwardsCreepGait(steps, speed\_lower\_is\_faster);

q.stepBackwardsCreepGait(steps, speed\_lower\_is\_faster);

q.stepLeftCreepGait(steps, speed\_lower\_is\_faster);

q.stepRightCreepGait(steps, speed\_lower\_is\_faster);

q.stepForwardsTrotGait(steps, speed\_lower\_is\_faster);

q.stepBackwardsTrotGait(steps, speed\_lower\_is\_faster);

q.stepLeftTrotGait(steps, speed\_lower\_is\_faster);

q.stepRightTrotGait(steps, speed\_lower\_is\_faster);

q.turnRight(steps, speed\_lower\_is\_faster);

q.turnLeft(steps, speed\_lower\_is\_faster);

q.raiseBody(degrees, speed\_lower\_is\_faster); // negative degrees lower body

q.pressUp(x); // Does x press-ups

q.squat(x); // Does x squats

q.moveServo(servo, pos); moves servo to position pos.

Servo can be from 0 to 15 on the PCA9685 board.