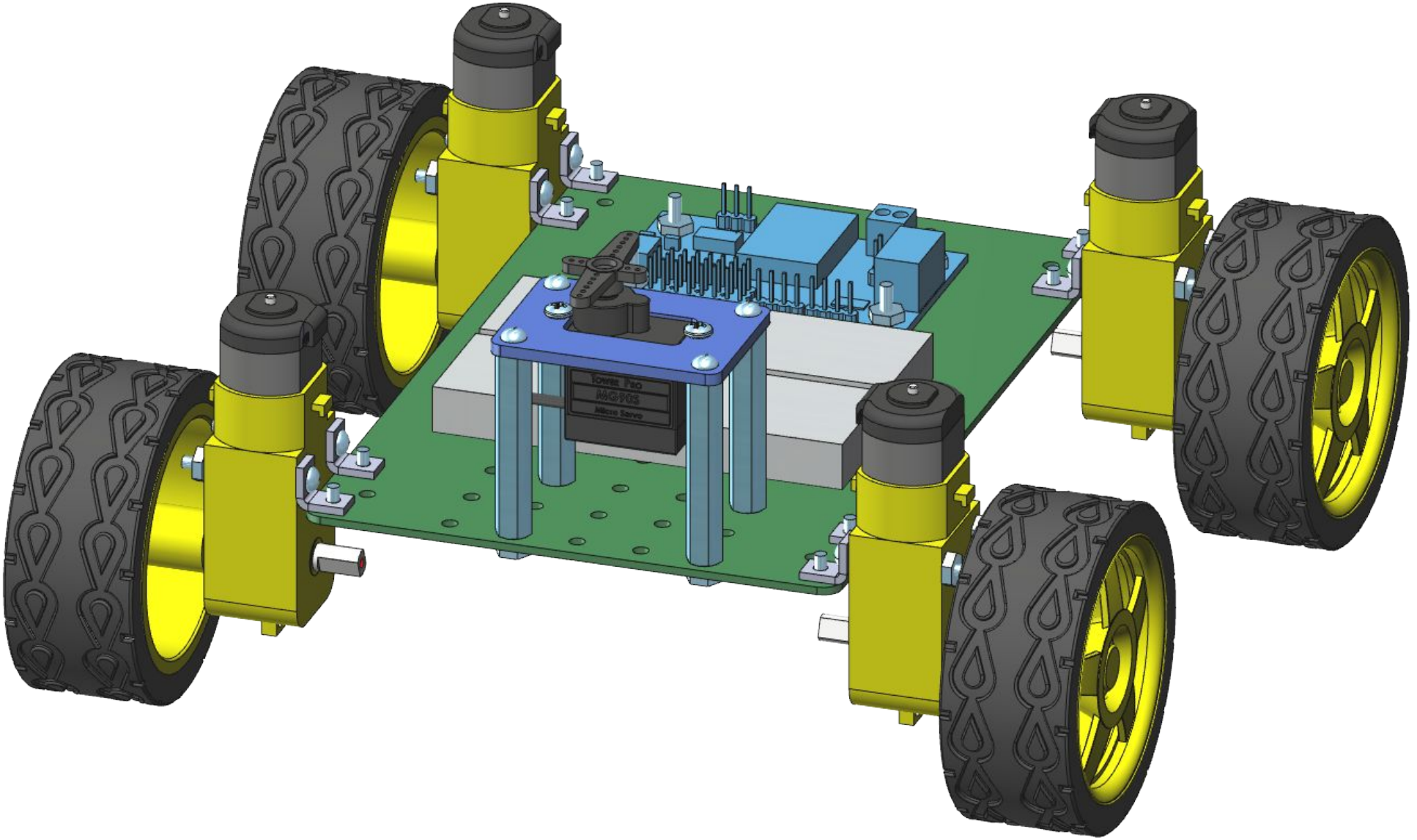
*Sten**BOT*** Robot Kit

Servo Interface



Legal Stuff

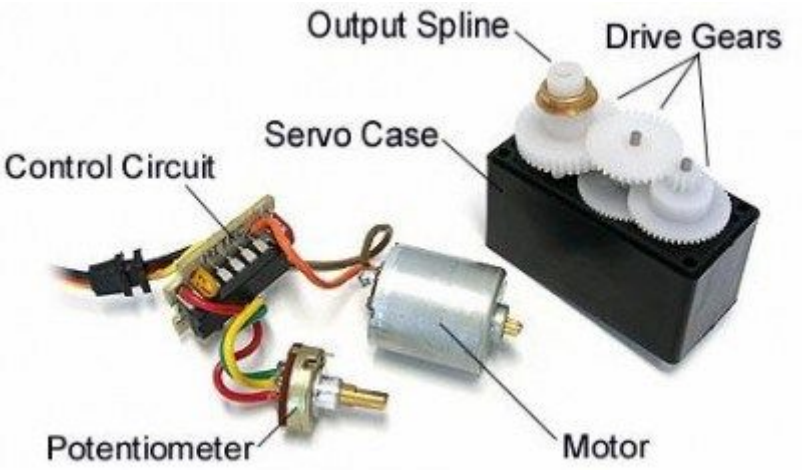
- Stensat Group LLC assumes no responsibility and/or liability for the use of the kit and documentation.
- There is a 90 day warranty for the Quad-Bot kit against component defects. Damage caused by the user or owner is not covered.
    - Warranty does not cover such things as over tightening nuts on standoffs to the point of breaking off the standoff threads, breaking wires off the motors, causing shorts to damage components, powering the motor driver backwards, plugging the power input into an AC outlet, applying more than 9 volts to the power input, dropping the kit, kicking the kit, throwing the kit in fits of rage, unforeseen damage caused by the user/owner or any other method of destruction.
- If you do cause damage, we can sell you replacement parts or you can get most replacement parts from online hardware distributors.
- This document can be copied and printed and used by individuals who bought the kit, classroom use, summer camp use, and anywhere the kit is used. Stealing and using this document for profit is not allowed.
- If you need to contact us, go to www.stensat.org and click on contact us.

References

- www.arduino.cc
- http://esp8266.github.io/Arduino/versions/2.1.0/doc/reference.html

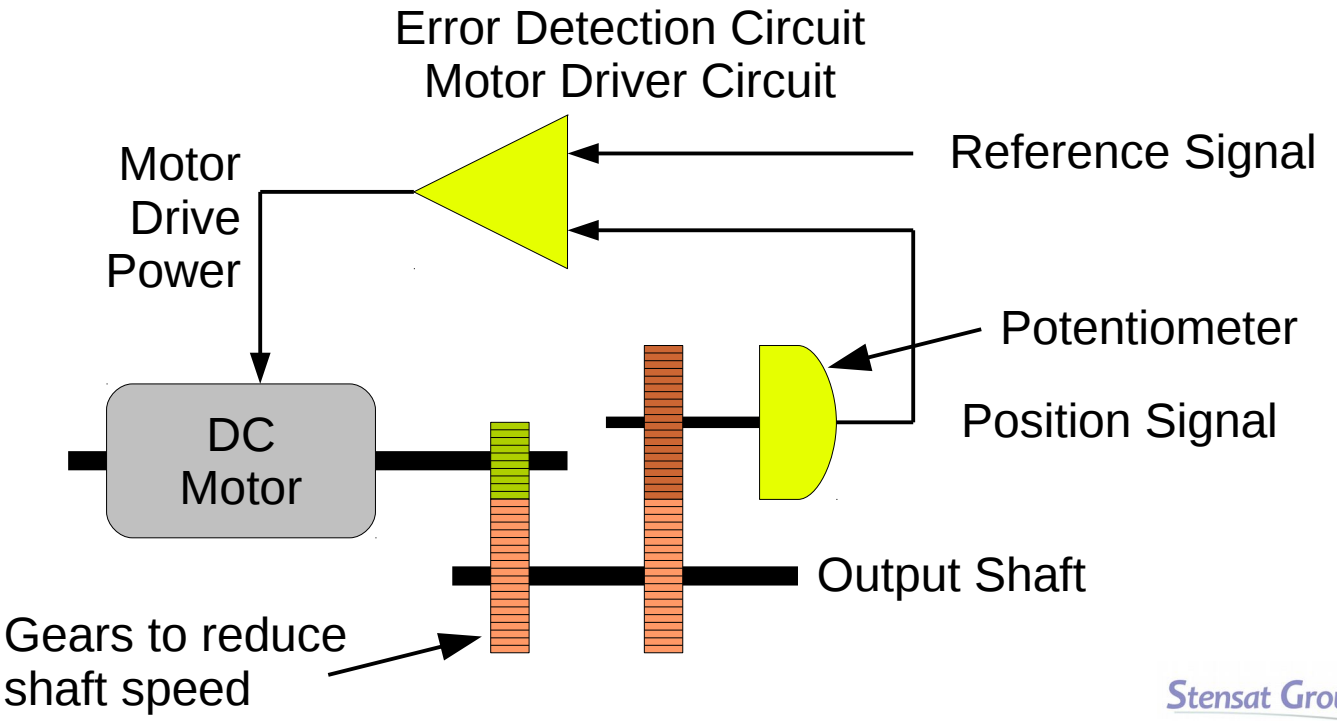
What is a Servo

- A servo is a geared motor with feedback used to control the position of the shaft of the motor.
- The servo consists of a motor that drives a bunch of gears to reduce the speed of the output spline or shaft. A potentiometer or variable resistor is connected to the output shaft and turns with the shaft. As it turns clockwise or counter clockwise, the resistance of the potentiometer changes. The resistance value indicates the angle of the shaft.



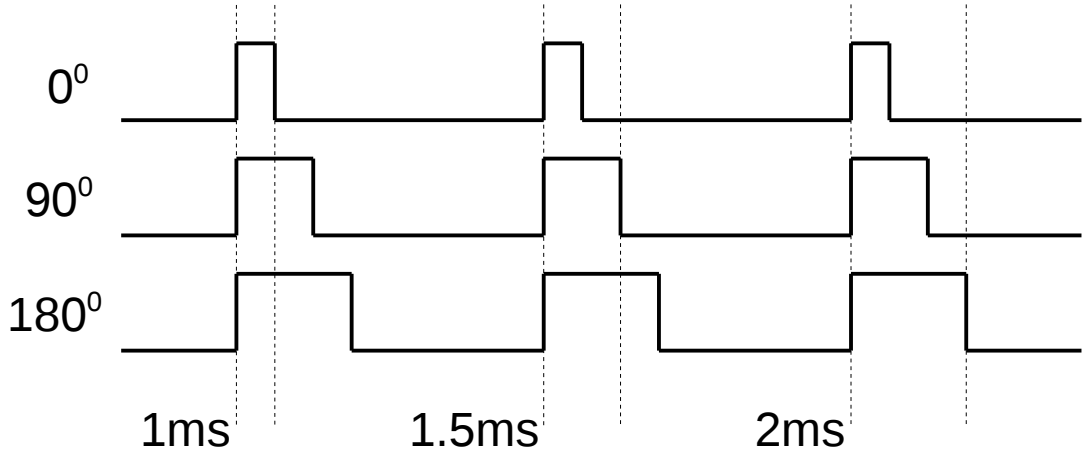
What is a Servo

- The potentiometer feeds a voltage signal based on the position of the shaft. A reference signal feeds a voltage signal for the desired position.
- The error detection circuit compares the two voltages and generates a voltage to power the DC motor in the desired direction until the position signal equals the reference signal.
- When the position signals equals the reference signal, the DC motor stops turning and the shaft is at the right angle.



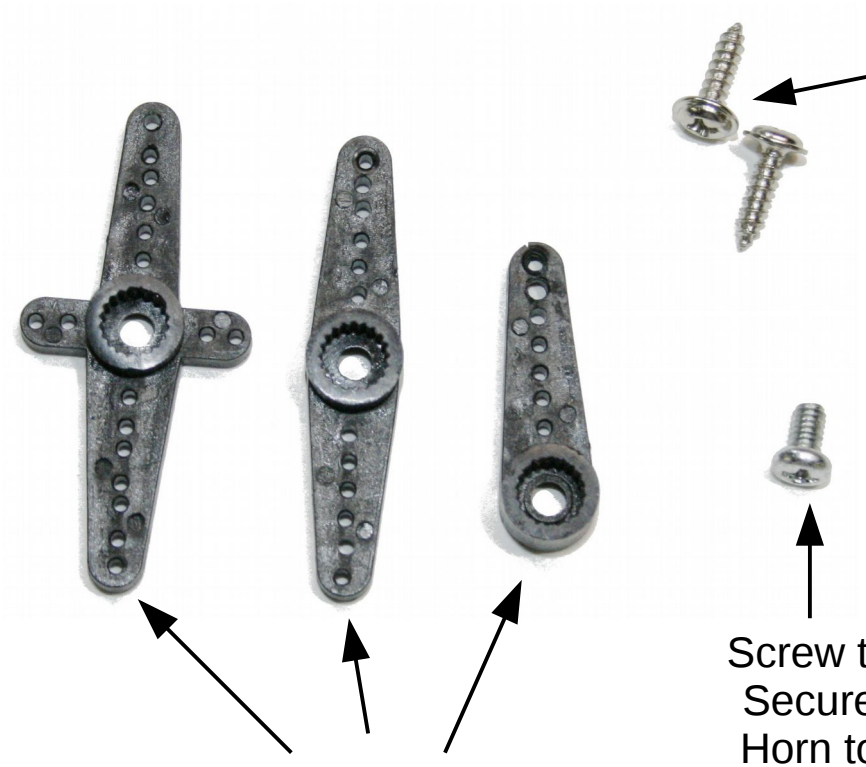
What is a Servo

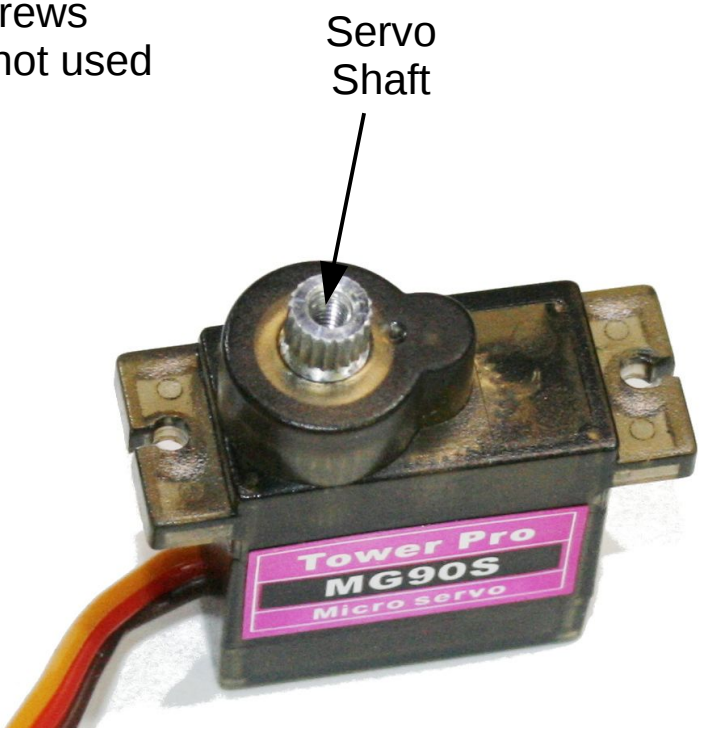
- The processor board uses pulses to control the position of the servo. The servo has an electronic circuit convert the pulse width to a position voltage.
- The processor board sends a pulse 50 to 60 times a second. The width of the pulse determines the position of the shaft which can range from 0 to 180 degrees.
- Neutral position is 90 degrees. The pulse width is 1.5 milliseconds (ms).
- 0 degree position is specified with a pulse width of 1 ms.
- 180 degree position is specified with a pulse width of 2 ms.
- The wave form below show what the signal looks like.



Servo Parts

- A servo is a geared motor that is used to rotate to specific angles. It is used in model airplanes to control the rudder, flaps and aeriolons.
- The servos come with parts. Most will be used.

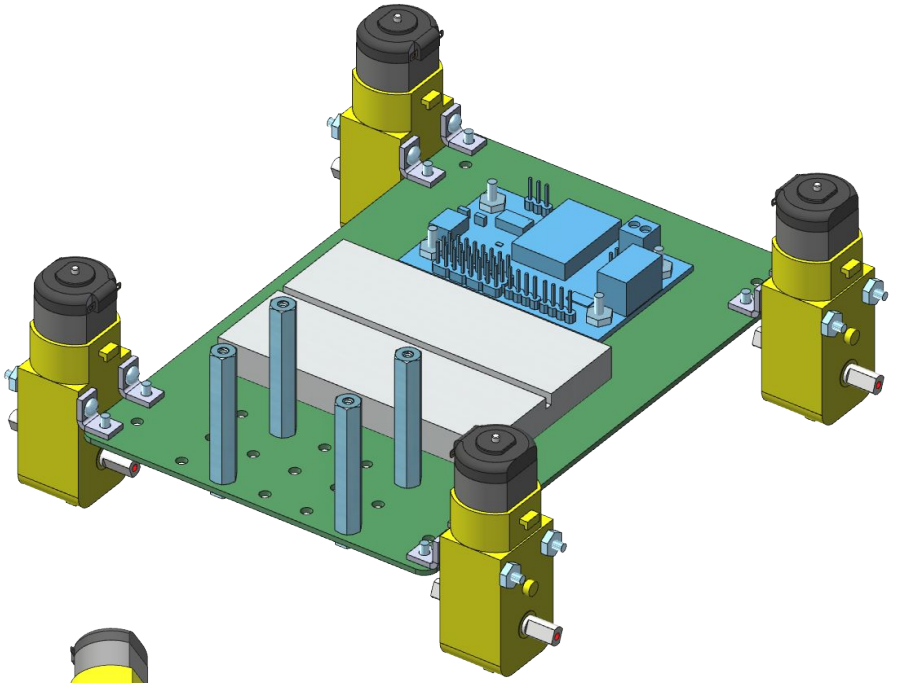


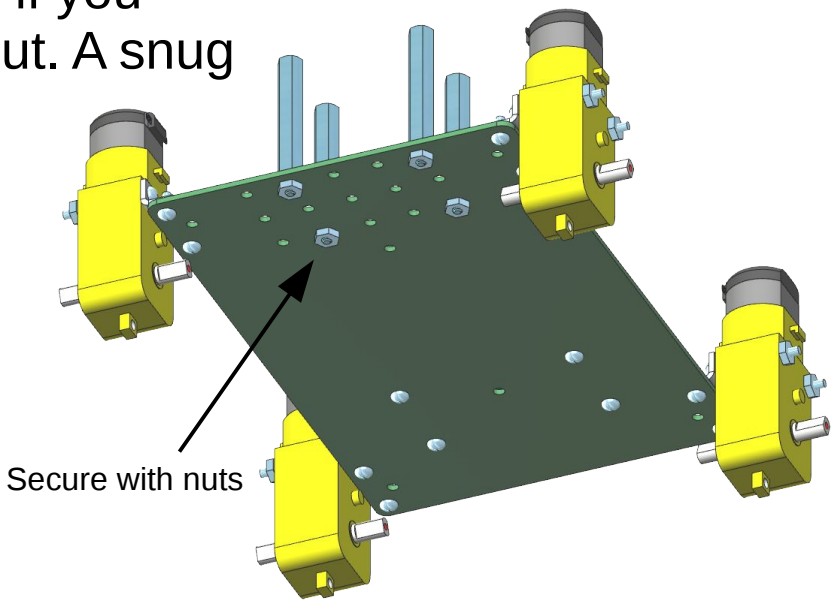


Servo

Robotic Arm Assembly

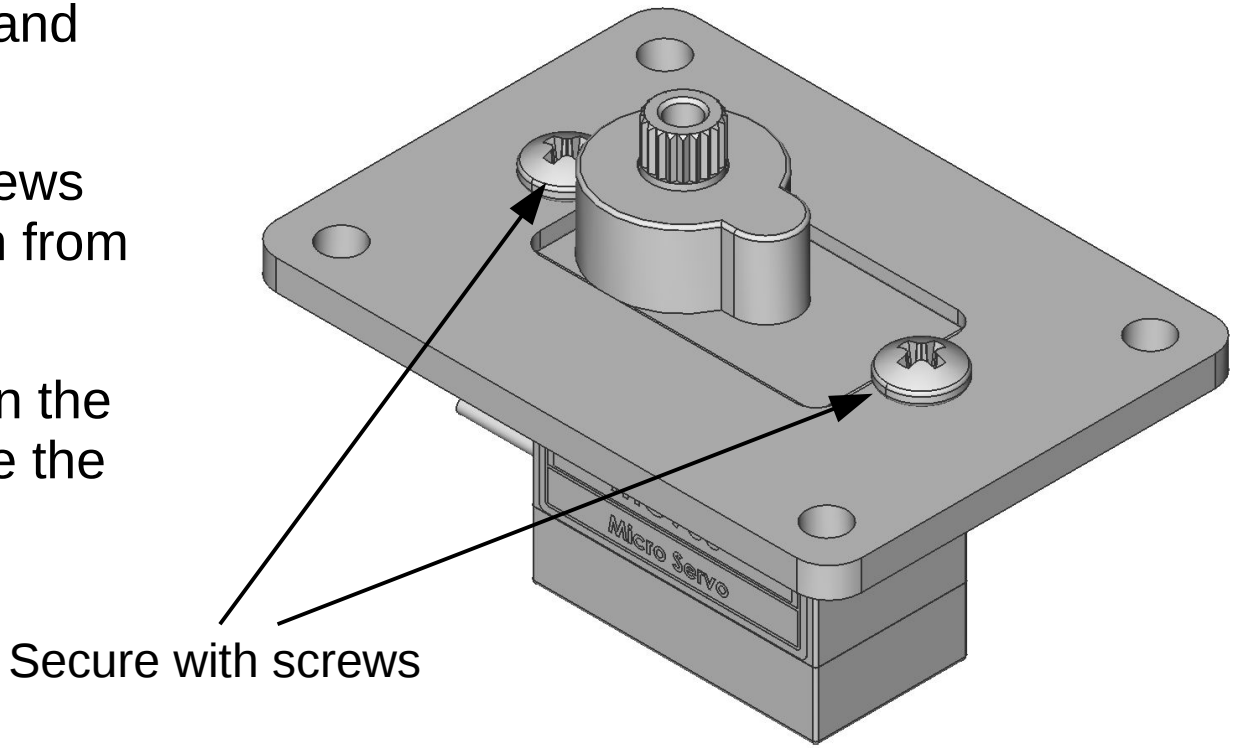
- Insert the standoffs in the positions shown. Secure each standoff with a nut from the bottom.
- Do not over tighten the nuts on the standoffs. The standoffs are made of aluminum and the threading can be broken off if you over tighten the nut. A snug fit is sufficient.

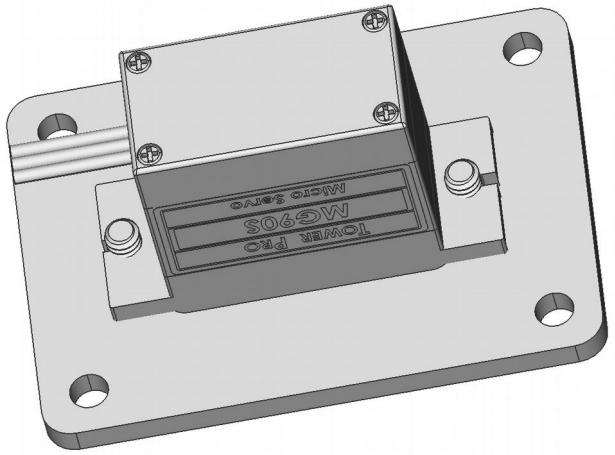




Azimuth Assembly

- Two metric screws are packaged with the servo and servo mounting plate.
- Use two of the 2.56M screws and insert them as shown from the top into the servo.
- You need to push and turn the screw clockwise to secure the servo.

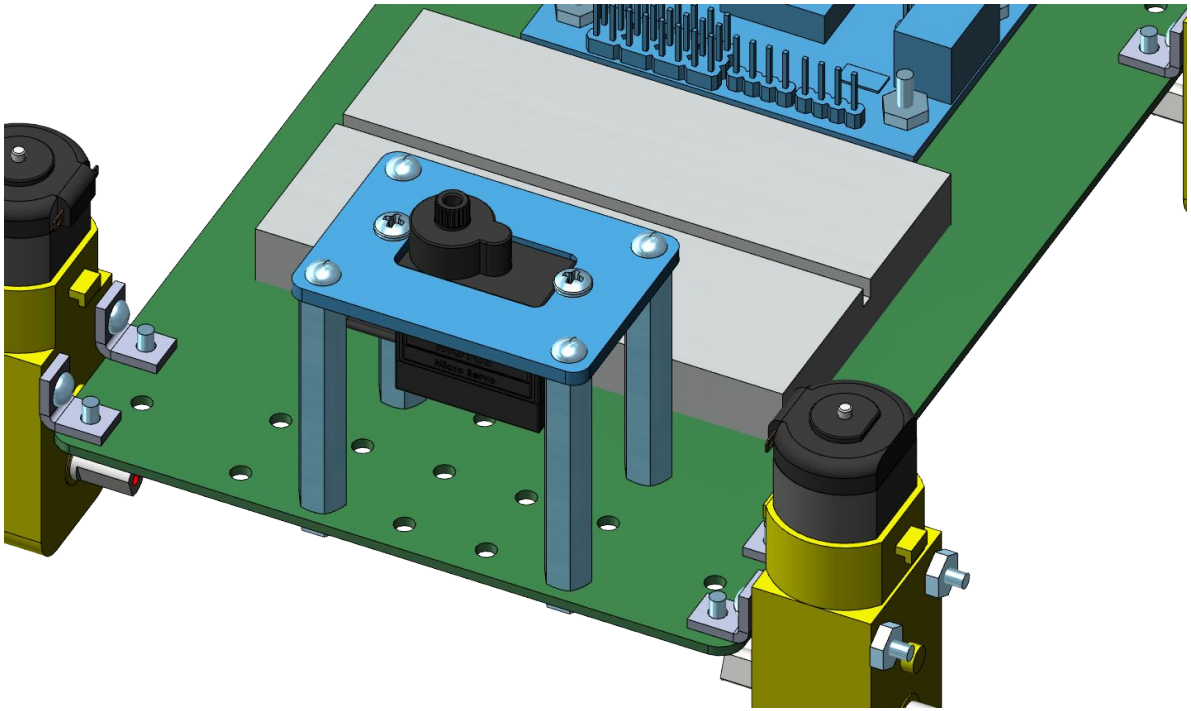




Bottom View

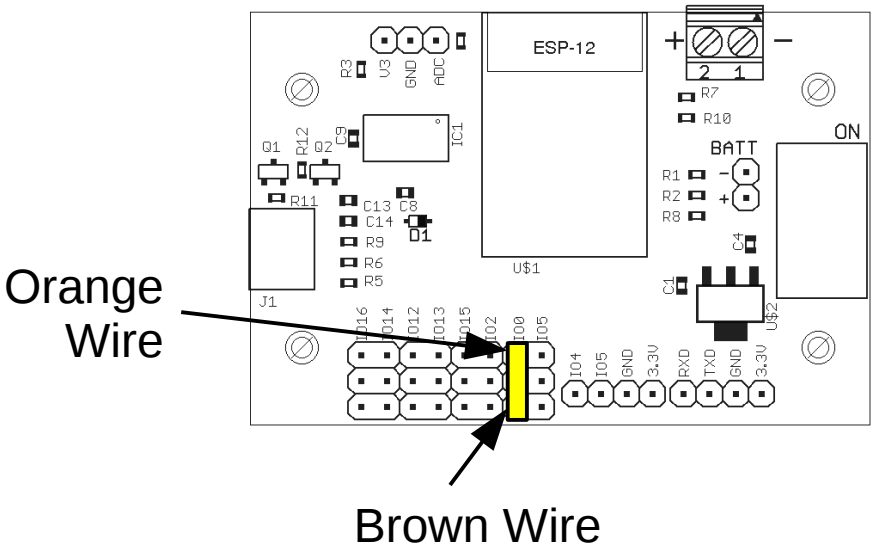
Robotic Arm Assembly

- Install the base servo plate onto the standoffs as shown.
- Secure with four 4-40 ¼ inch screws.



Connecting The Servo

- Connect the servo to digital pin 0. The servo has a 3 pin socket connector.
- Orientation is critical. Installing it backwards can damage the servo.
- Insert the connector onto the pins with the brown wire closest to the edge of the processor board. The orange wire should be closest to the Pin 0 mark.



Controlling the Servo

- To control the servo, the servo library needs to be installed.
- In the arduino software, click on the **Sketch** menu and select **Include Library**. Locate **Servo** and select it.
- At the beginning of the program will be a statement #include <Servo.h>
- This tells the compiler to include functions for controlling servos.

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

Controlling the Servo

- Next, create a servo object. It will be called **base**. **base** is an instance of the Servo object. Multiple instances can be created.
- In the **setup()** function, **base** is attached to digital pin 0. The **attach()** function connects the specified pin to the base servo. All operations will occur on that pin until another **attach()** function is called with a different digital pin selected.
- After base is attached, the servo position can be set with the **write()** function. The parameter is the angle in degrees.

```
#include <Servo.h>

Servo base;

void setup()
{
     base.attach(0);
     base.write(90);
}

void loop()
{
}
```

Operating The Servo

- To operate the servo, batteries need to be installed. The servo can only be operated on battery power.
- Turn on the processor board and upload the code from the previous page.
- When the upload is complete, the servo should make a sound and the shaft should rotate.

Servo Operation

- The servo does take some time to move to new positions. If you perform multiple moves in sequence, a delay is needed between new positions to give the servo time to move.
- The code to the right shows the simple method of moving between two positions.
- You can reduce the delays to determine how little of a delay is needed to move from 10 degrees to 170 degrees.
- These servos can move about 180 degrees.

```
#include <servo.h>

Servo base;

void setup() {
  base.attach(0);
}

void loop() {
  base.write(10);
  delay(1000);
  base.write(170);
  delay(1000);
}
```

End

- This completes the assembly of the robotic arm.