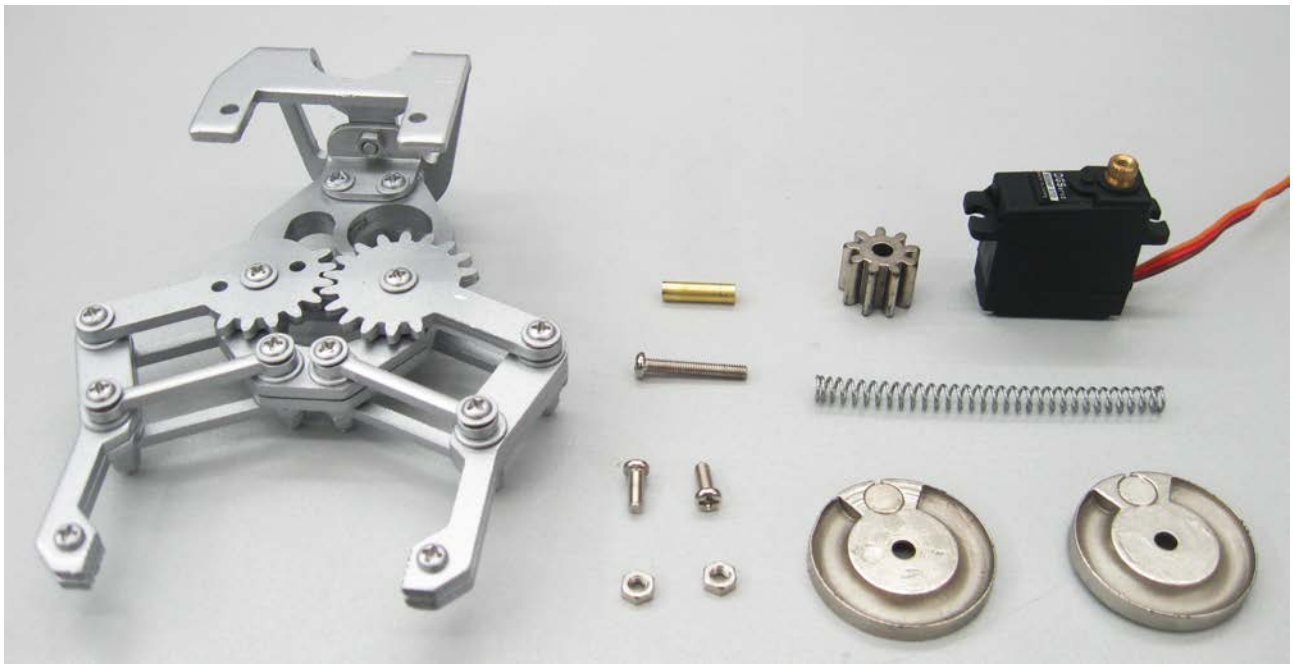
Assembly of Mk II Gripper from

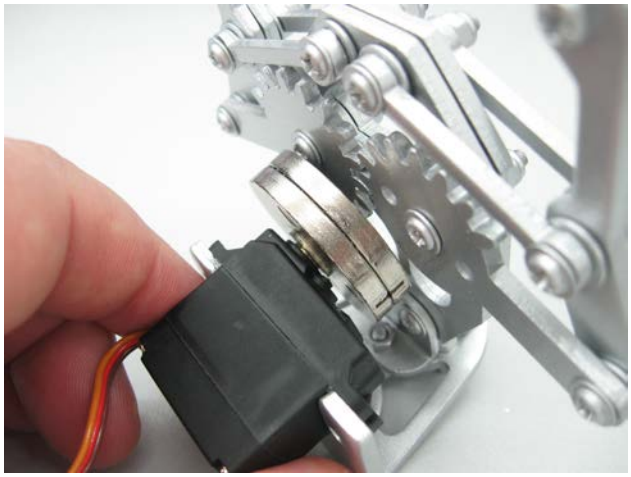
The Mk II robot gripper has many improvements over the original design but like any device controlled by a servo it is important for the servo to be correctly aligned so that the gripper assembly does not limit the servos range of movement.

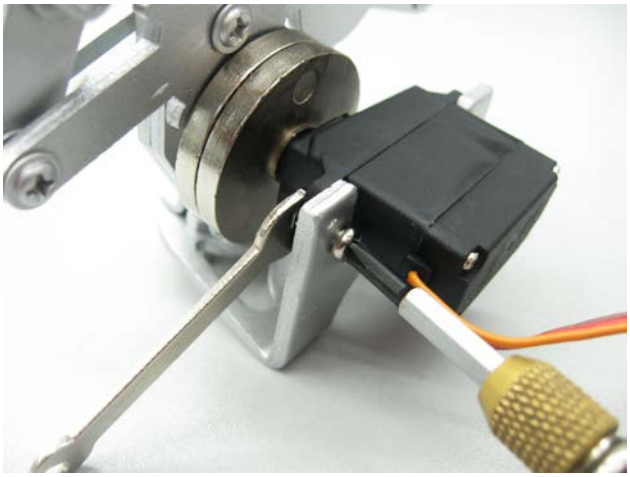


Start by installing the spring in the clutch. Hold the spring in place with one hand while you place the other half on top. When the two halves are joined the magnets will hold the assembly together. The main purpose of the magnets is to lock the clutch when it is not under excessive load.

Now fit the clutch onto the servo and rotate the servo fully clockwise. This will be the servos position when the gripper is fully closed. This step is very important to ensure the servo and mechanism is correctly aligned.

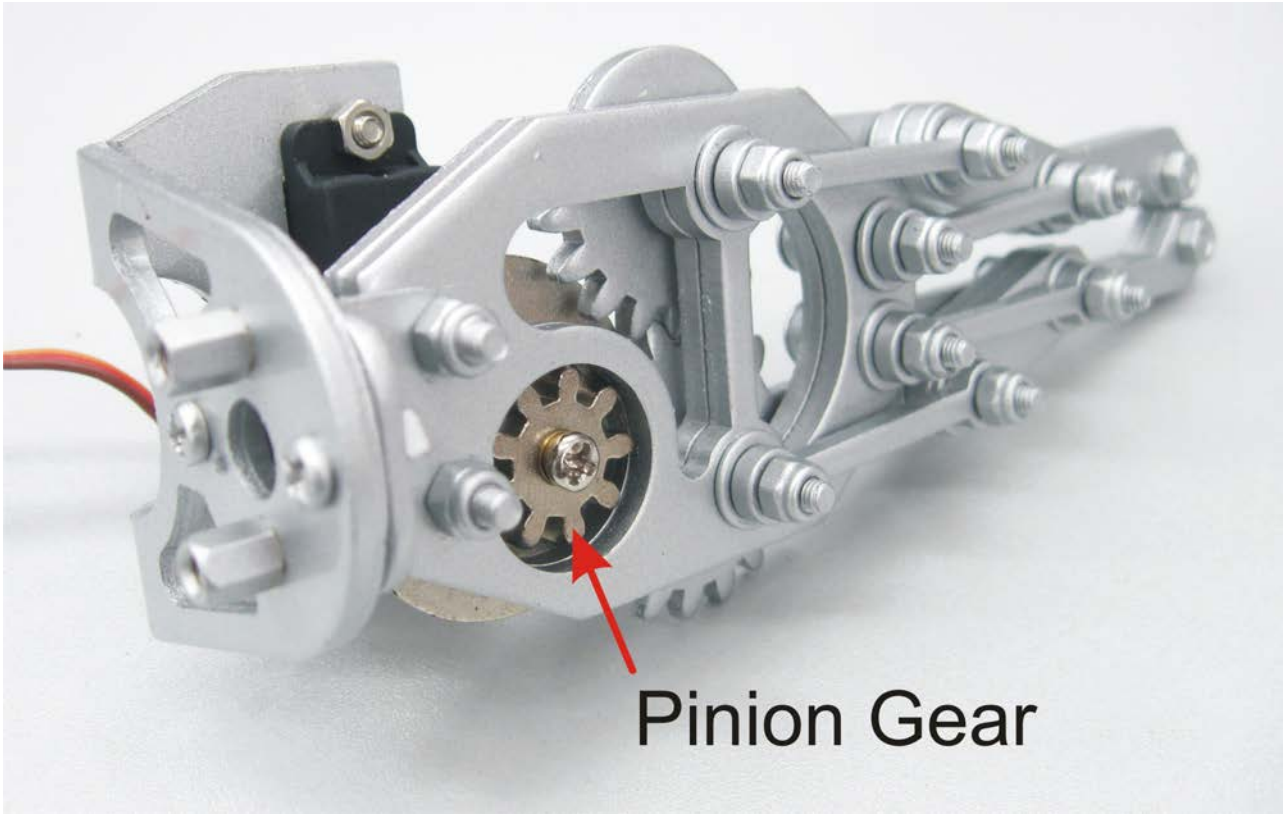
Slide the servo and clutch assembly into the gripper and mount with the screws provided. Do not over-tighten the servo mounting screws.



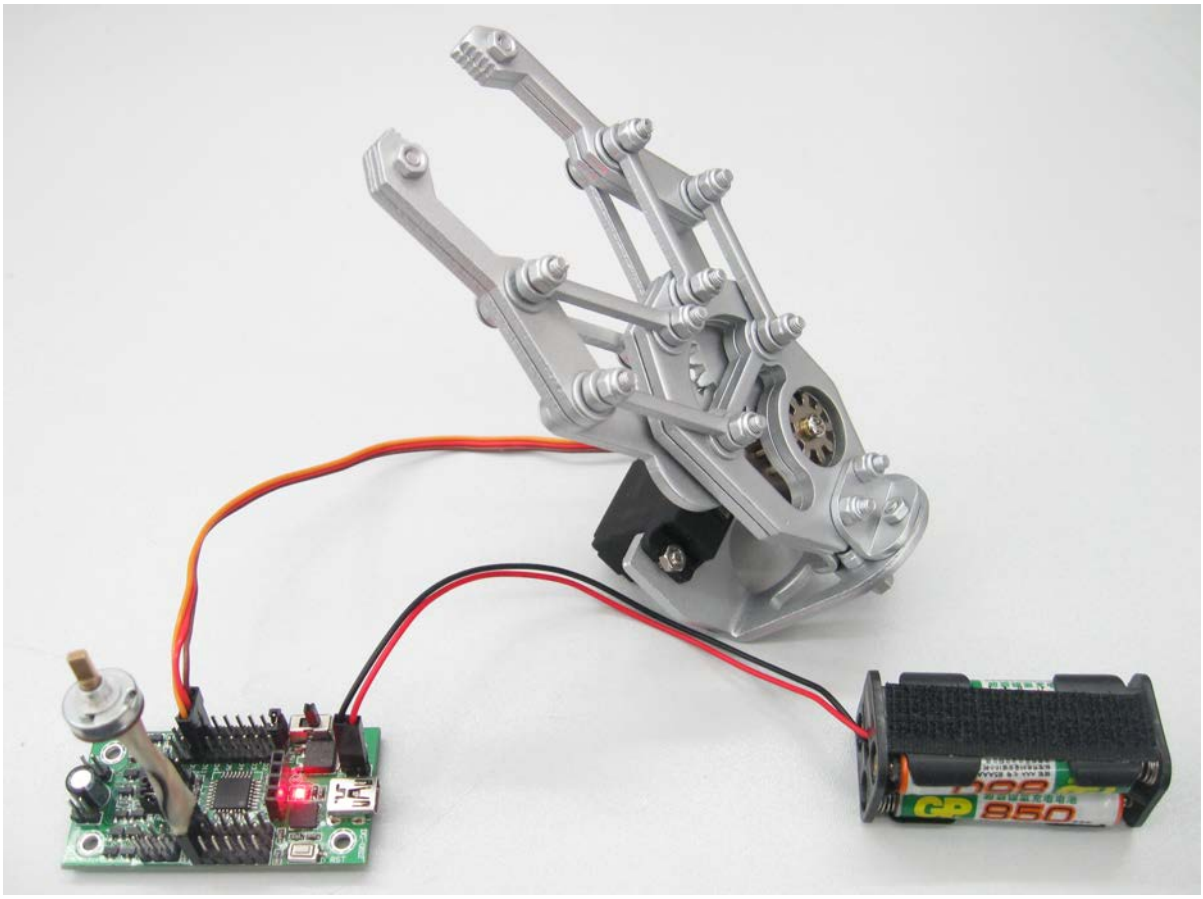


Now move the gripper to the fully closed position. As the servo was also moved to the fully closed position in the previous step they should now be aligned.

If necessary you may need to force the clutch to slip slightly to get the teeth of the gears to align. Install the clutch pinion gear, brass sleeve and mount them with the screw provided.



Finally connect the gripper to your favorite controller and run some test code. In this photo I am using an Arduino compatible Mini Driver from DAGU and running a slightly modified version of the “Knob” sample code that comes with the servo library.



Knob | Arduino 1.0.4

File Edit Sketch Tools Help

Knob

```
// Controlling a servo position using a potentiometer (variable resistor)
// Original code by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
// Modified by Russell Cameron

#include <Servo.h>

Servo myservo;                          // create servo object to control a servo
int potpin = 0;                         // analog pin used to connect the potentiometer
int val;                                // variable to read the value from the analog pin

void setup()
{
  myservo.attach(11,400,2600);          // attaches the servo on pin 11 with minimum and maximum pulse widths specified
  Serial.begin(9600);
}

void loop()
{
  val = analogRead(potpin);             // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 400, 2600);   // scale it to use it with the servo range slightly greater than the servo can respond to
  myservo.writeMicroseconds(val);       // send a servo control pulse val uS in width
  Serial.println(val);                  // display current servo pulse width in uS
  delay(15);                            // waits for the servo to get there
}
```

Done compiling.

```
Binary sketch size: 3,962 bytes (of a 7,168 byte maximum)
```

13 Arduino NG or older w/ ATmega8 on COM49

My modification used the **writeMicroseconds()** command to give me more precise control of the servo. My code displays the position in µS on the serial monitor so that when the servo will not turn any further I can see what the pulse width is.

You may need to experiment a bit to find out what is the full range of your servo. Each servo is slightly different and servo manufacturers can change their design without notice.

When I tested this gripper, the servo responded to pulses from 500uS to 2400uS to get a travel of just over 180°. Beyond this range, the servo could not move.

If your gripper still does not close properly then you may need to take the pinion gear back out and rotate it by one tooth.

Now your gripper is assembled and working correctly, what can you make with it?