Robotic Arm Assembly Instructions

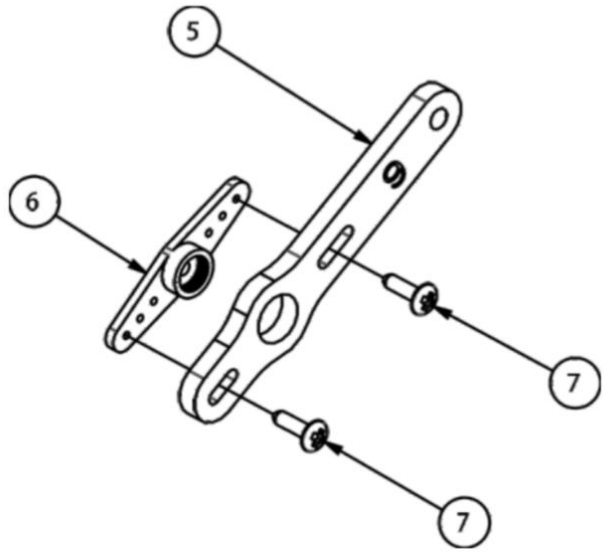
Last Revised: 11 January 2017

Part A:

First follow the instructions: http://www.robotshop.com/media/files/zip2/rb-mea-02_-_documentation_1.zip

While assembling the servos:

- Put the black plastic gear on the servo as well as the transparent plastic.
- Before assembly the servos, attach the transparent part and black servo gear together with the screws.



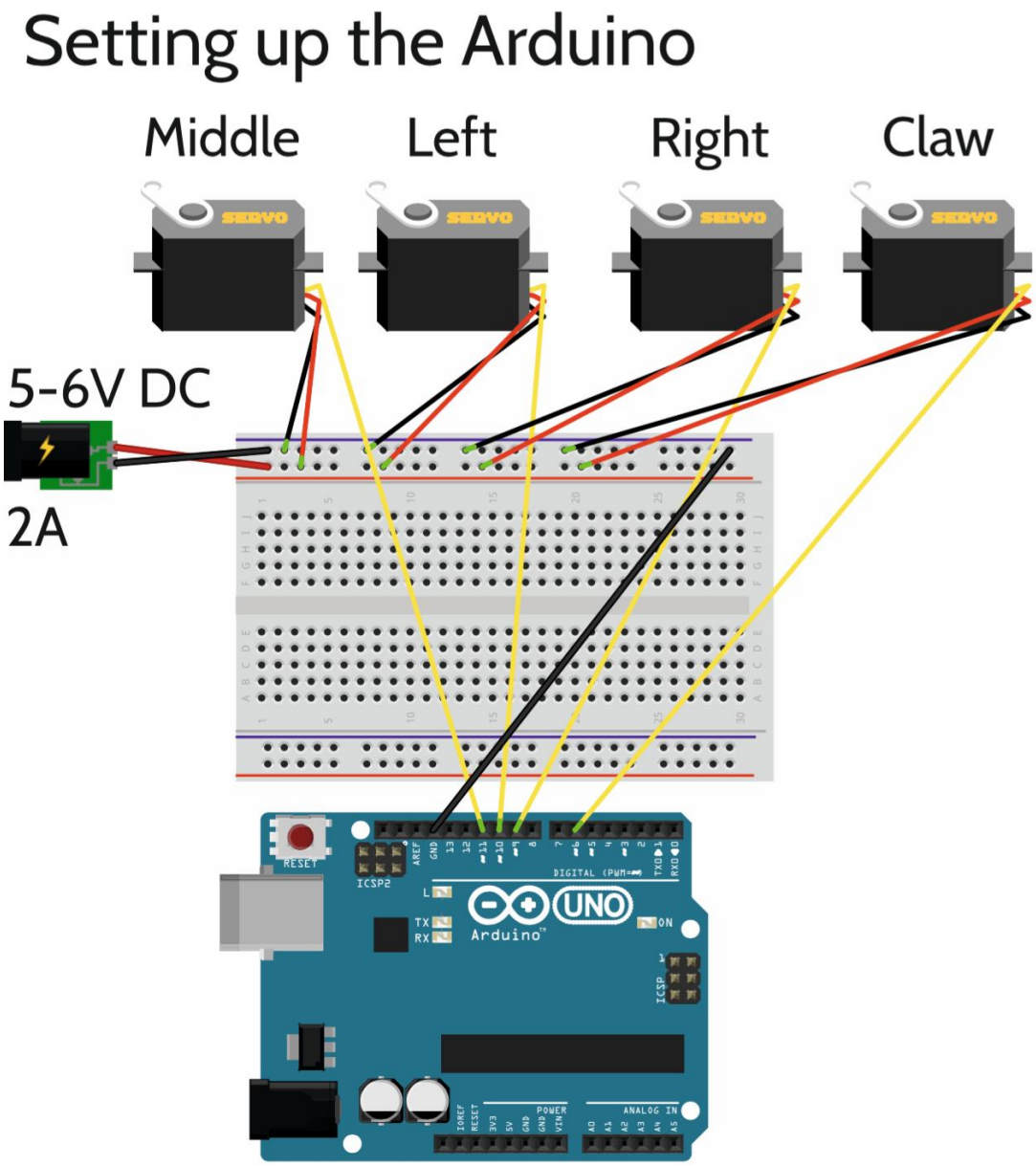
Step 12. Attach the Servo Arm to the Long Servo Arm Extension using the sharp screws in the servo pack. These will self tap with a little pressure.

,

Then, put part 6 (see above picture) on the servo, run the test program (from 1 to 180, to adjust the servo position. **Take note of maximum angle the servo can rotate.**

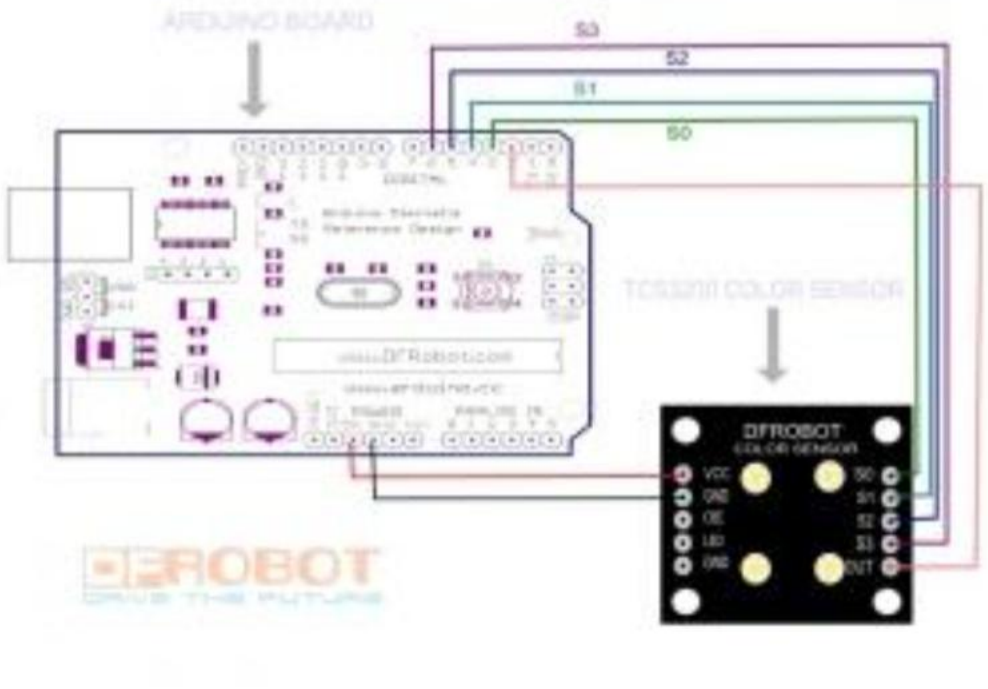
Repeat above step until all four servos are properly set up.

For wiring the servos:



Part B:

Color sensor: see the wiring document from:
https://www.dfrobot.com/wiki/index.php/File:G3771.png



Part C:

Range sensor:

The out is output, and should be connected to A0.
The vin should be connected to 5V,
The GND should be connected to GND.

Appendix:

Servo unit test code: please connect servo input (yellow) to pin 11

```
/* Sweep
 by BARRAGAN <http://barraganstudio.com>
 This example code is in the public domain.

 modified 8 Nov 2013
 by Scott Fitzgerald
 http://www.arduino.cc/en/Tutorial/Sweep
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(11);  // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
  pinMode(11,OUTPUT);
  pinMode(13,OUTPUT);
}

void loop() {

  int c = Serial.parseInt();

  if(c!=0){

    myservo.write(c);
    Serial.println(c);
  }
    delay(500);
}
```

Color sensor sample code:

```
/*     Arduino Color Sensing Tutorial
 *
 *  by Dejan Nedelkovski, www.HowToMechatronics.com
 *
 */

#define S0 3
#define S1 4
#define S2 5
#define S3 6
#define sensorOut 2

int frequency = 0;

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);

  // Setting frequency-scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);

  Serial.begin(9600);
}

void loop() {
  // Setting red filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("R= ");//printing name
```

```
  Serial.print(frequency);//printing RED color frequency
  Serial.print("  ");
  delay(100);

  // Setting Green filtered photodiodes to be read
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("G= ");//printing name
  Serial.print(frequency);//printing RED color frequency
  Serial.print("  ");
  delay(100);

  // Setting Blue filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  frequency = pulseIn(sensorOut, LOW);
  // Printing the value on the serial monitor
  Serial.print("B= ");//printing name
  Serial.print(frequency);//printing RED color frequency
  Serial.println("  ");
  delay(100);
}
```

Range sensor sample code:

```
/* Ping))) Sensor

   This sketch reads a PING))) ultrasonic rangefinder and returns the
   distance to the closest object in range. To do this, it sends a pulse
   to the sensor to initiate a reading, then listens for a pulse
   to return.  The length of the returning pulse is proportional to
   the distance of the object from the sensor.

   The circuit:
        * +V connection of the PING))) attached to +5V
        * GND connection of the PING))) attached to ground
        * SIG connection of the PING))) attached to digital pin 7

   http://www.arduino.cc/en/Tutorial/Ping

   created 3 Nov 2008
   by David A. Mellis
   modified 30 Aug 2011
   by Tom Igoe

   This example code is in the public domain.

 */


// this constant won't change.  It's the pin number
// of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
```

```
void loop() {
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
}
```

```
long microsecondsToInches(long microseconds) {
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```