

<http://electronicshobbyists.com>



LATEST POSTS → [Raspberry pi PWM Tutorial | Control Brightness of LED and Servo Motor](http://electronicshobbyists.com/category/latest-posts/) ([http://electronic...](http://electronicshobbyists.com/category/latest-posts/)) [Google+](#)

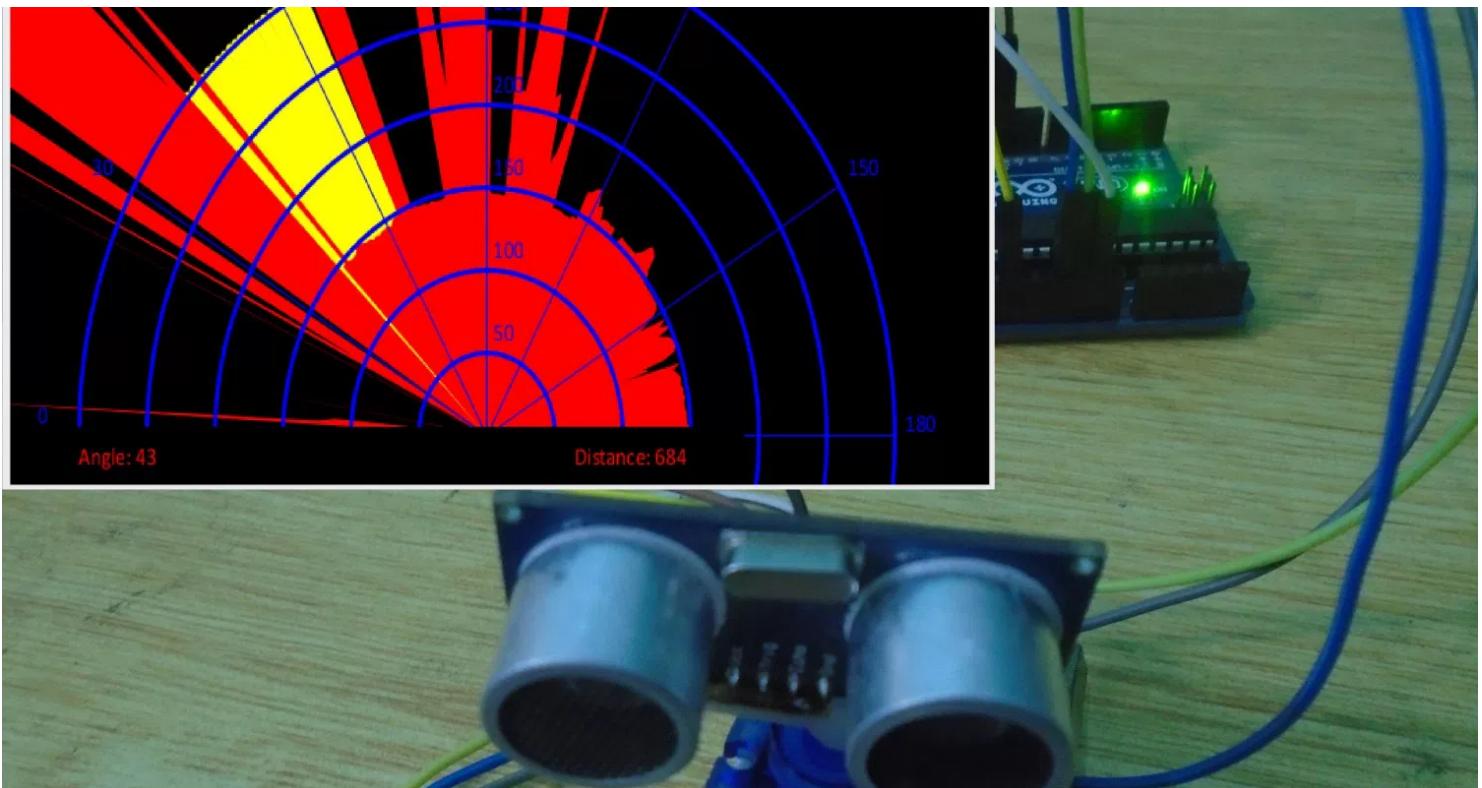
[\(https://plus.google.com/110431300036945931834\)](https://plus.google.com/110431300036945931834)

[\(https://twitter.com/ElectronicsHoby\)](https://twitter.com/ElectronicsHoby)

[Facebook \(https://www.facebook.com/ElectronicsHobbyists-565164177019817/\)](https://www.facebook.com/ElectronicsHobbyists-565164177019817/)



[Home \(http://electronicshobbyists.com\)](http://electronicshobbyists.com) > [Arduino \(http://electronicshobbyists.com/category/arduino/\)](http://electronicshobbyists.com/category/arduino/)



Arduino Sonar System using Ultrasonic/Sonar Sensor

⌚ March 5, 2017

📁 [Arduino \(http://electronicshobbyists.com/category/arduino/\)](http://electronicshobbyists.com/category/arduino/), [Arduino Processing \(http://electronicshobbyists.com/category/arduino-processing/\)](http://electronicshobbyists.com/category/arduino-processing/), [DIY \(http://electronicshobbyists.com/category/diy/\)](http://electronicshobbyists.com/category/diy/)

In this project, we are going to make a Arduino sonar system which will detect the nearby project using the sonar sensor and will show us on the radar. For making the radar, we will use Processing IDE software and for the sending the values of the sensors to the processing, we will use the Arduino IDE software.

If you haven't used Arduino and processing together before, then follow this Tutorial | [Arduino](#)

[Processing Tutorial \(http://electronicshobbyists.com/arduino-processing-tutorial-connect-arduino-to-processing/\)](http://electronicshobbyists.com/arduino-processing-tutorial-connect-arduino-to-processing/)

Required Components Arduino Sonar System

The components required for the Arduino sonar system are as follows

Click on the part to buy from amazon.

- Arduino Uno
- Servo Motor (sg90)
- Sonar Sensor

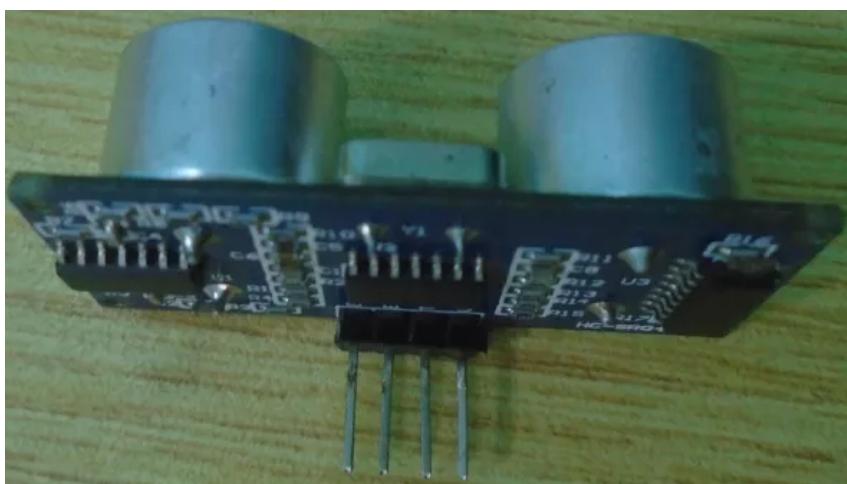
Circuit Diagram and Hardware Explanation

When you will receive the Arduino sonar sensor, the pins of the sonar sensor will be towards down as you can see in the below figure.



(<https://i2.wp.com/electronicshobbyists.com/wp-content/uploads/2017/03/Ultrasonic-sensor.png>)

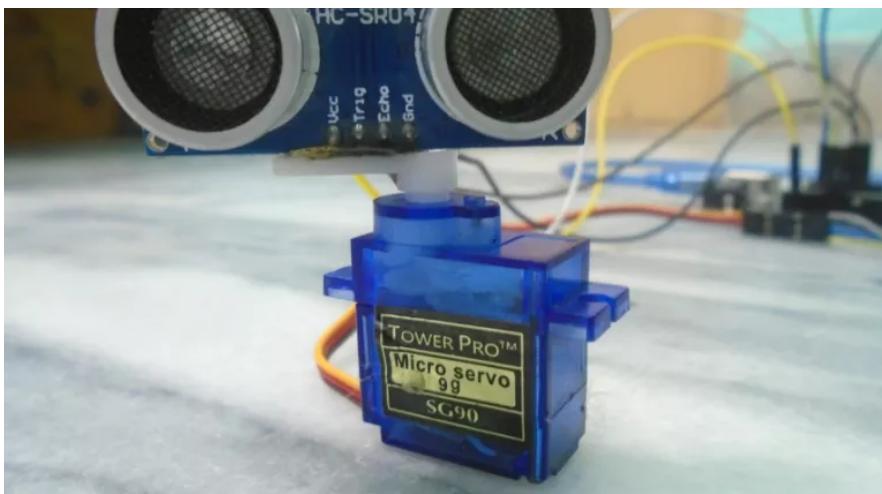
To place it on the servo, you will have to straighten these pins, so use some tool to make it straighten as shown in the below figure.



(<https://i0.wp.com/electronicshobbyists.com/wp-content/uploads/2017/03/Ulraonic-sensor-1.png>)

After that, use the glue gun and place the sonar sensor on the top of the servo motor so that the the sonar sensor moves with the movement of the servo.





(<https://i2.wp.com/electronicshobbyists.com/wp-content/uploads/2017/03/Servo-and-ultrasonic.png>)

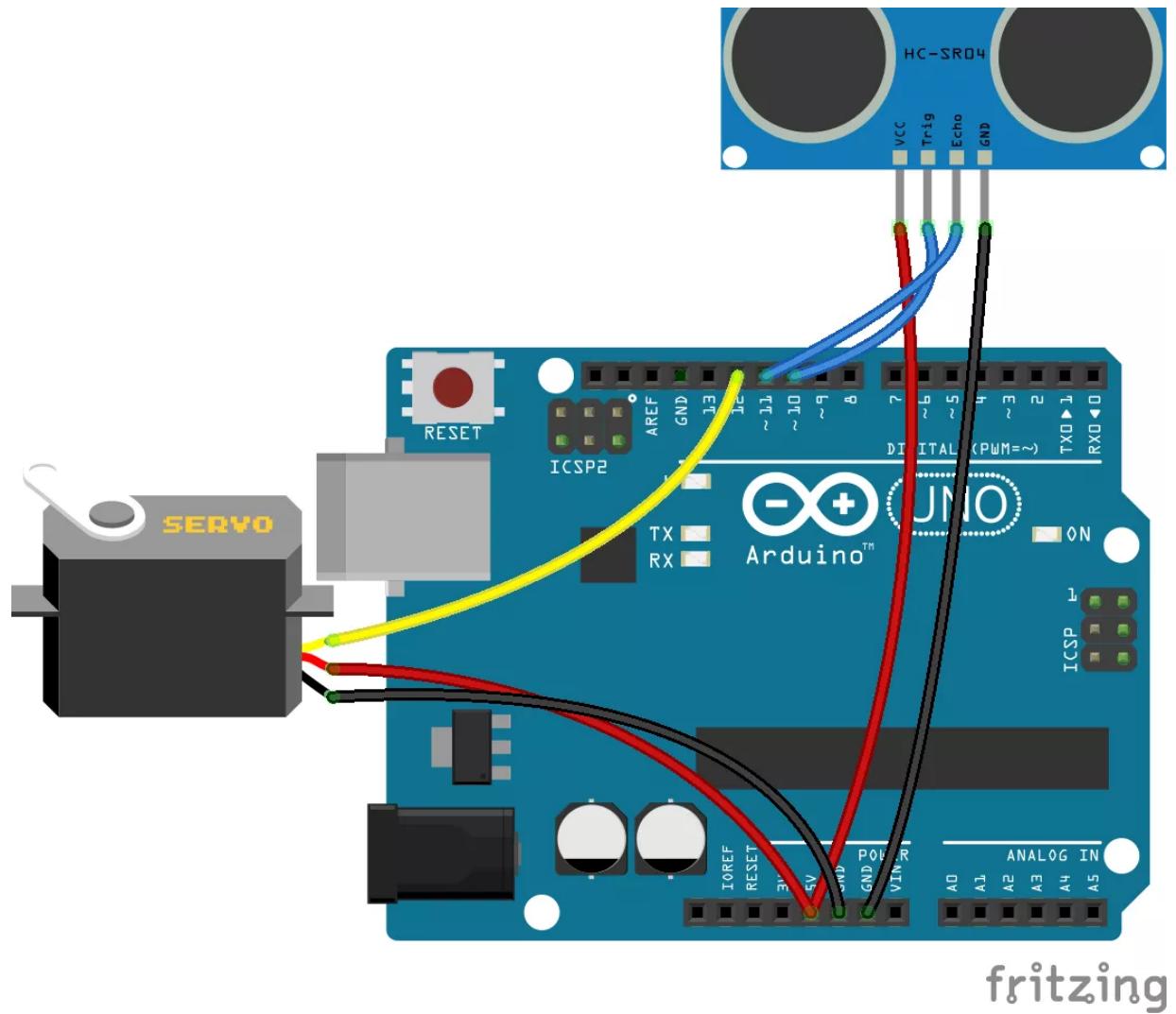
After that, make the connections as described below

- o Sonar sensor VCC to Arduino 5V
- o Sonar sensor GND to Arduino GND
- o Sonar sensor Trig to Arduino 10
- o Sonar sensor Echo to Arduino 11
- o Servo motor Black to Arduino GND
- o Servo motor yellow to Arduino 12
- o Servo motor red to Arduino VCC

Also Read

- o [Ultrasonic/Sonar Sensor HC-SR04 Arduino Tutorial](http://electronicshobbyists.com/ultrasonic-distance-sensor-hc-sr04-arduino-tutorial-proximity-sensor/) (<http://electronicshobbyists.com/ultrasonic-distance-sensor-hc-sr04-arduino-tutorial-proximity-sensor/>)
- o [Servo Motor interfacing with Arduino – Arduino servo control](http://electronicshobbyists.com/servo-motor-interfacing-arduino-arduino-servo-control/) (<http://electronicshobbyists.com/servo-motor-interfacing-arduino-arduino-servo-control/>)





fritzing

(<https://i2.wp.com/electronicshobbyists.com/wp-content/uploads/2017/03/Arduino-Sonar-system.png>)

Arduino Code

Before uploading the code, [download the servo library from here](#) (<http://www.arduino.cc/playground/uploads/ComponentLib/SoftwareServo.zip>).

```

1. #include <Servo.h>           //including the servo library
2. Servo sg90;                 //Variable for servo
3. const int echo_pin = 11;     // Initializing echo pin for distance sensor
4. const int trig_pin = 10;     // initializing trigger pin for distance sensor
5. unsigned long distance = 0; //Variable to store the distance value
6. unsigned long distance_cm = 0; //variable to store the distance in cm
7. int servo_pin = 12; //initializing servo pin
8. int i = 0;
9. int j = 0;
10. int total = 0;
11. int average = 0;
12.
13. void setup() {
14.   sg90.attach(servo_pin);    //Arduino will take the servo value from this pin
15.   pinMode(trig_pin, OUTPUT); //declaring trig pin as output pin
16.   pinMode(echo_pin, INPUT); //declaring echo pin as input pin
17.   Serial.begin(9600);       //Setting the baudrate at 9600 for serial communication
18. }
19.
20. void loop() {
21.   for(i = 0; i < 180; i++) { //Moving the servo from left to right

```

```

22.     sg90.write(i);
23.     distanceVal();
24. }
25.
26. for(i = 180; i > 0; i--) { //Moving the servo from right to left
27.     sg90.write(i);
28.     distanceVal();
29. }
30. }
31.
32. //Function to calculate the distance
33. void distanceVal()
34. {
35.     for (j = 0; j<=10;j++) {
36.         digitalWrite(trig_pin, LOW);
37.         delayMicroseconds(50);
38.         digitalWrite(trig_pin, HIGH); //setting the trig pin to generate a wave
39.         delayMicroseconds(50);
40.         digitalWrite(trig_pin, LOW);
41.         distance = pulseIn(echo_pin, HIGH); //setting the echo pin high to receive the wave
42.         distance_cm = distance/58; //converitng the distance into cm
43.         total = total + distance_cm;
44.         delay(10);
45.     }
46.     average = total/10;
47.     if (j >= 10) {
48.         j = 0;
49.         total = 0;
50.     }
51.     Serial.print("X");
52.     Serial.print(i);
53.     Serial.print("V");
54.     Serial.println(average);
55. }

```

Processing Code

```

1. import processing.serial.*; //importing the serial library for serial communication between
2. arduino and processing
3. Serial port; //declaring a variable for serial communication
4. int[] sensor_value = new int[181]; //Variable to store the value of the sensor
5. int[] previous_value = new int[181];//Variable to store the previous value of sensor
6. PFont text; //Variable for setting the font size and type
7. int radarDist = 0;
8. int radius = 350; //declaring a variable for setting the radius of objects
9. int widthh = 300; //declaring a variable for setting the width of objects
10. int position = 0; //Declaring a variable for storing position state (left or right)
11. int motion = 0;
12. float x_coordinate; //declaring a Variable for storing the x co-ordinate value
13. float y_coordinate; //declaring a Variable for storing the y co-ordinate value
14. int value;
15. String data ; //String to recieve value from the serial port
16. String servo_value; //String for storing the servo position
17. String sensor_val; //String for storing the sensor value
18.
19. void setup(){
20.     size(720, 380); //Setting the size of the output window
21.     port = new Serial(this, "COM3", 9600); /*Setting the COM port and baudrate for serial
communication. Set the baudrate here according to Arduino IDE's baudrate */

```

```
22.     text = createFont("calibri", 15); //Setting the text size and type
23.     background (0); //Setting the colour of background. '0' for black background
24.     textFont(text);
25. }
26.
27. //Below function will make all the shapes for output window
28. void draw(){
29.     fill(0); //black background for the following shape
Shares
2     noStroke(); //No outline for the below shape
30.     ellipse(radius, radius, 1080, 720); //Drawing a ellipse
31.     rectMode(CENTER); //drawing a rectangle at the center of ellipse
32.     rect(350,402,1080,720); //size of rectangle
1
33.     drawradar();
34.     drawshape();
35.     drawvalues();
36.     drawgridlines();
37.     drawtext();
38. }
39.
40.
41. void serialEvent (Serial port) {
42.     data = port.readStringUntil('\n'); //This will wait until the data is received
43.     if (data != null) {
44.         data = trim(data); //trim the empty space
45.         servo_value = data.substring(1, data.indexOf("V")); //Storing the servo value
46.         sensor_val = data.substring(data.indexOf("V")+1, data.length()); //Storing the sensor value
47.         position = Integer.parseInt(servo_value); //storing values in position variable
48.         value = Integer.parseInt(sensor_val);
49.         previous_value[position] = sensor_value[position]; //storing values in array
50.         sensor_value[position] = value;
51.     }
52. }
53.
54. void drawradar()
55. {
56.     if ( position >= 179) { //animation will move from right to left
57.         motion = 1;
58.     }
59.     if (position <= 1) { //animation will move from left to right
60.         motion = 0;
61.     }
62.     strokeWeight(6); //Seting the thickness of lines
63.     if (motion == 0) { //Move left to right
64.         for (int i = 0; i <= 20; i++) { //drawing 20 lines
65.             stroke(255, 255, 0); //setting the colour of lines
66.             line(radius, radius, radius + cos(radians(position+(180+i)))*widthh, radius +
67.                 sin(radians(position+(180+i)))*widthh); //Start and end point of the line
68.         }
69.     }
70.     else { //move right to left
71.         for (int i = 20; i >= 0; i--) { //draw 20 lines
72.             stroke(255,255, 0); //Setting the colour of lines
73.             line(radius, radius, radius + cos(radians(position+(180+i)))*widthh, radius +
74.                 sin(radians(position+(180+i)))*widthh); //Start and end point of the line
75.         }
76.     }
77.     //Drawing the shapes of the sensor values
78.     void drawshape()
79. {
```

```

80. //First round
81. noStroke(); //No outline for these values
82. fill(255,0,0); //set the colour
83. beginShape(); //make the shape
84. for (int i = 0; i < 180; i++) {
85. x_coordinate = radius + cos(radians((180+i)))*((previous_value[i])); //make x co-ordinate
86. y_coordinate = radius + sin(radians((180+i)))*((previous_value[i])); //make y co-ordinate
87. vertex(x_coordinate, y_coordinate);
88. }
89. endShape();
90.
91. //Second Round
92. fill(255,0,0);
93. beginShape();
94. for (int i = 0; i < 180; i++) {
95. x_coordinate = radius + cos(radians((180+i)))*(sensor_value[i]);
96. y_coordinate = radius + sin(radians((180+i)))*(sensor_value[i]);
97. vertex(x_coordinate, y_coordinate);
98. }
99. endShape();
100. }
101.
102. void drawvalues()
103. {
104. for (int i = 0; i <=6; i++){ //Loop for making rings
105. noFill();
106. strokeWeight(3); //Thickness of rings
107. stroke(0, 0, 255); //color of rings
108. ellipse(radius, radius, (100*i), (100*i)); //drawing the rings
109. fill(0, 0, 255);
110. noStroke();
111. text(Integer.toString(radarDist+50), 380, (305-radarDist), 50, 50); //Settings the values for
each ribg
112. radarDist+=50;
113. }
114. radarDist = 0;
115. }
116.
117. void drawgridlines()
118. {
119. for (int i = 0; i <= 6; i++) { //loop for drawing the grid lines
120. strokeWeight(1); //thickness of lines
121. stroke(0, 0, 255); // color of lines
122. line(radius, radius, radius + cos(radians(180+(30*i)))*widthh, radius + sin(radians(180+
(30*i)))*widthh); //values at which these will be drawn
123. fill(0, 0, 255);
124. noStroke();
125. if (180+(30*i) >= 300) {
126. text(Integer.toString(0+(30*i)), (radius+10) + cos(radians(180+(30*i)))*(widthh+10),
(radius+10) + sin(radians(180+(30*i)))*(widthh+10), 25,50); //giving value for each line
127. } else {
128. text(Integer.toString(0+(30*i)), radius + cos(radians(180+(30*i)))*widthh, radius +
sin(radians(180+(30*i)))*widthh, 60,40); //giving value for each line
129. }
130. }
131. }
132.
133. //Below function will show the Angle and distance values below the radar
134. void drawtext()
135. {

```

```
136. noStroke();
137. fill(0);
138. rect(290,370,500,50);
139. fill(255, 0, 0);
140. text("Angle: "+Integer.toString(position), 100, 380, 100, 50);
141. text("Distance: "+Integer.toString(value), 540, 380, 250, 50);
142. }
```

Video



If you have any questions, feel free to ask us in the comment section.

JOIN OUR NEWSLETTER

Enter Your Name

Enter Your Email

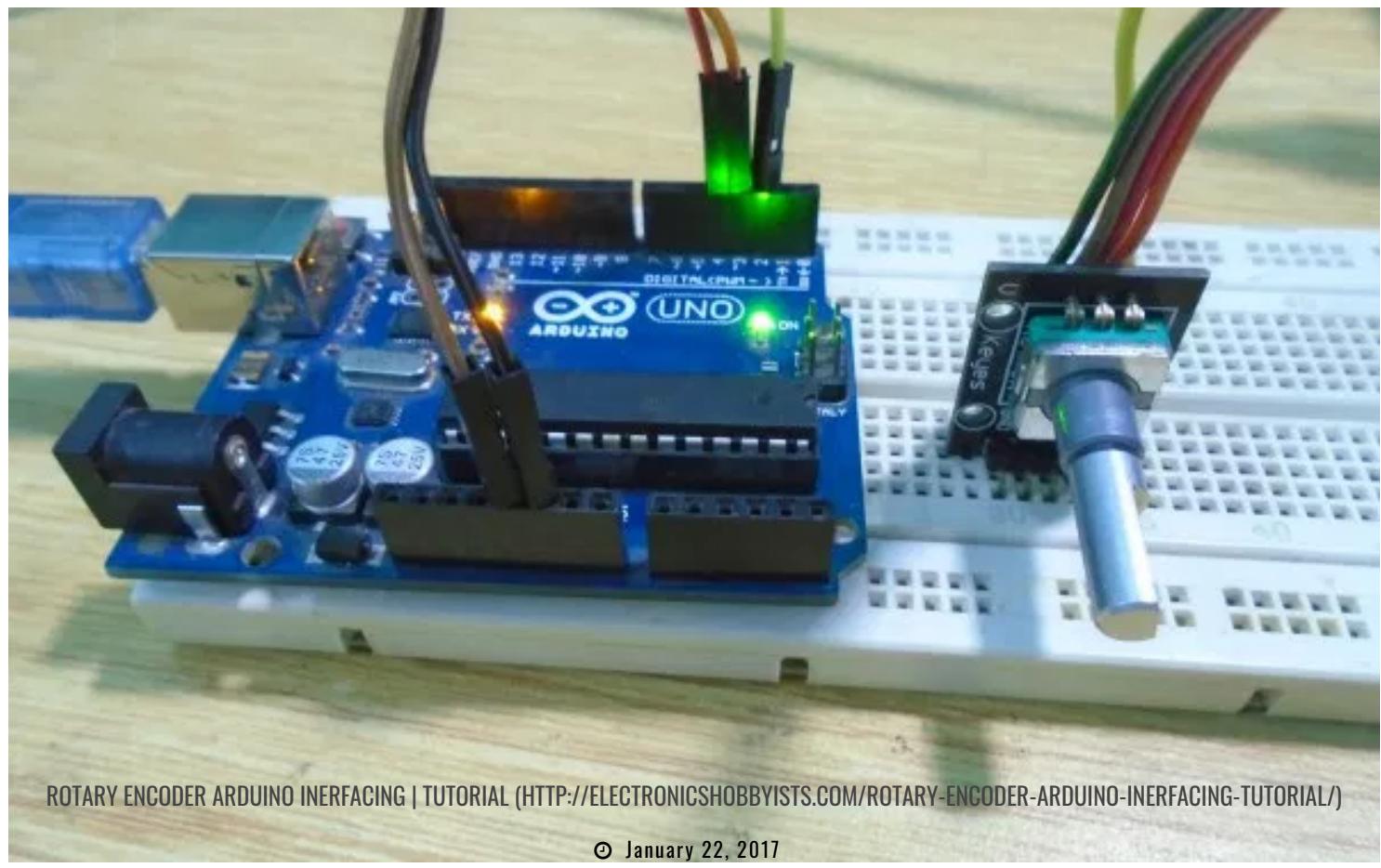
SUBSCRIBE Now

We help electronics hobbyists to create projects related to Micro-controllers. Join our newsletter to get access to new projects first.

We hate spam. Your email address will not be sold or shared with anyone else.

RELATED POSTS

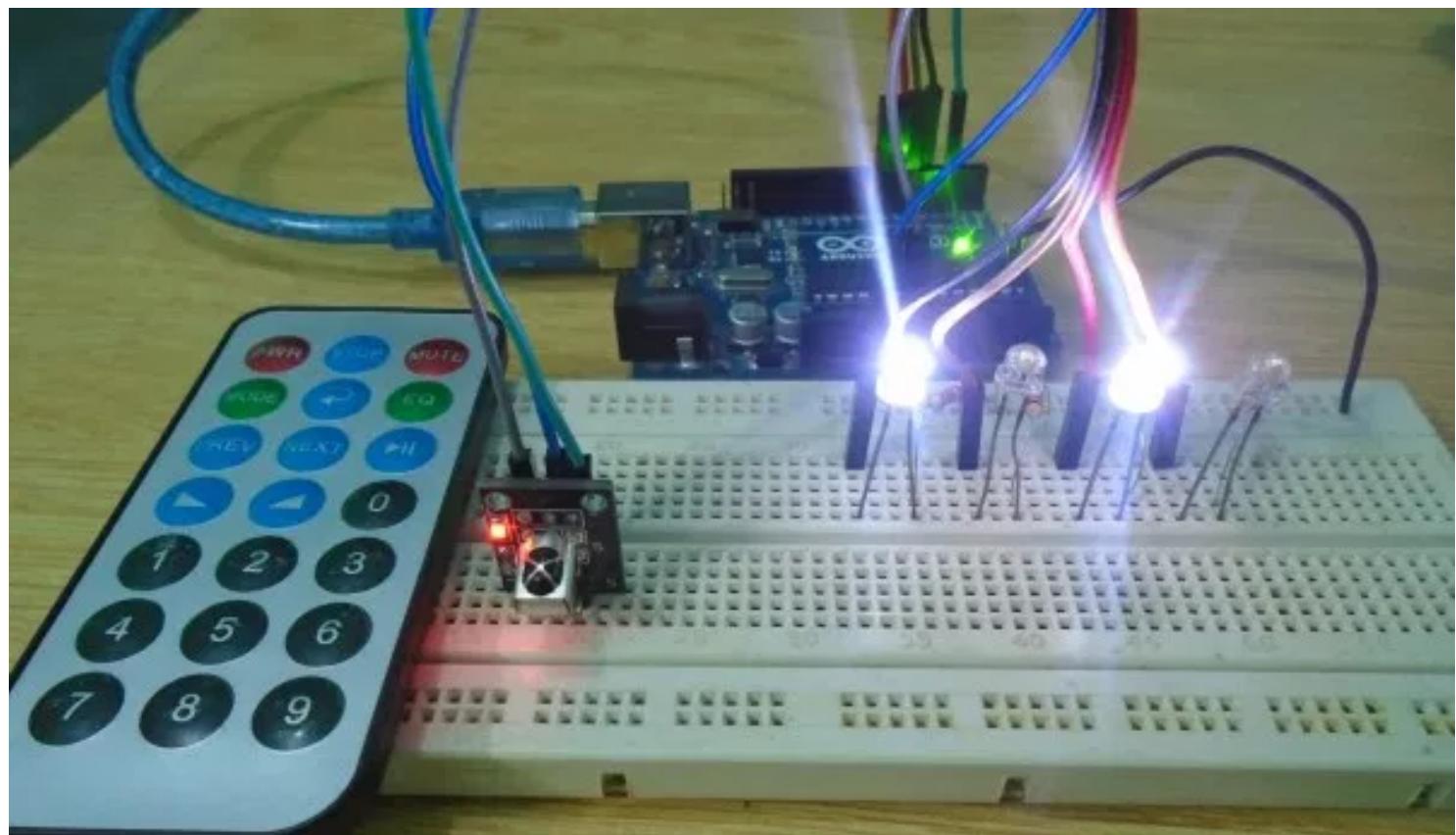




ROTARY ENCODER ARDUINO INERFACING | TUTORIAL ([HTTP://ELECTRONICSHOBBYISTS.COM/ROTARY-ENCODER-ARDUINO-AINERFACING-TUTORIAL/](http://electronicshobbyists.com/rotary-encoder-arduino-inerfacing-tutorial/))

⌚ January 22, 2017

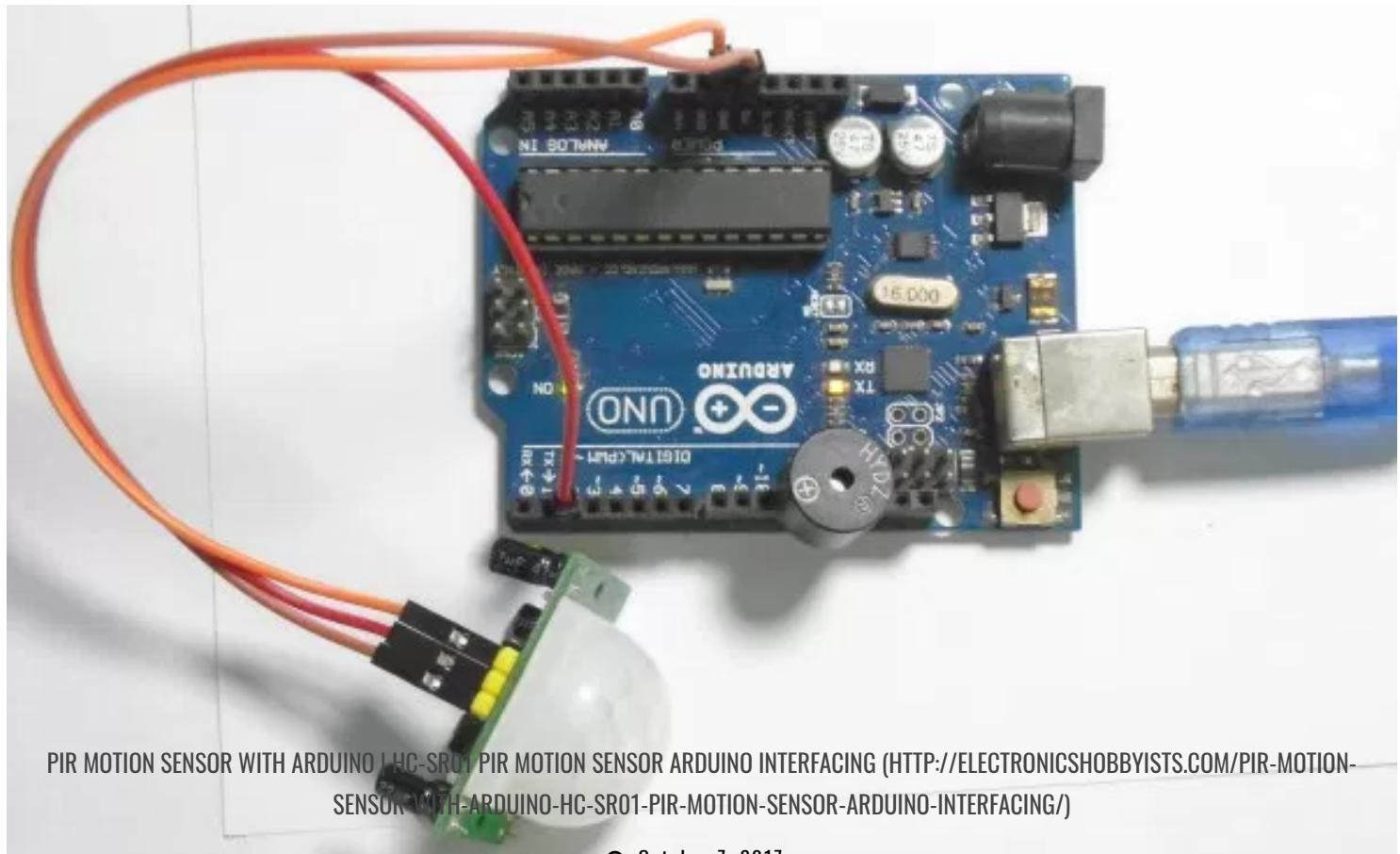
[\(http://electronicshobbyists.com/rotary-encoder-arduino-inerfacing-tutorial/\)](http://electronicshobbyists.com/rotary-encoder-arduino-inerfacing-tutorial/)



ARDUINO REMOTE CONTROL COMMUNICATION – ARDUINO IR SENSOR TUTORIAL ([HTTP://ELECTRONICSHOBBYISTS.COM/ARDUINO-REMOTE-CONTROL-COMMUNICATION-ARDUINO-IR-SENSOR/](http://electronicshobbyists.com/arduino-remote-control-communication-arduino-ir-sensor/))

⌚ January 11, 2017

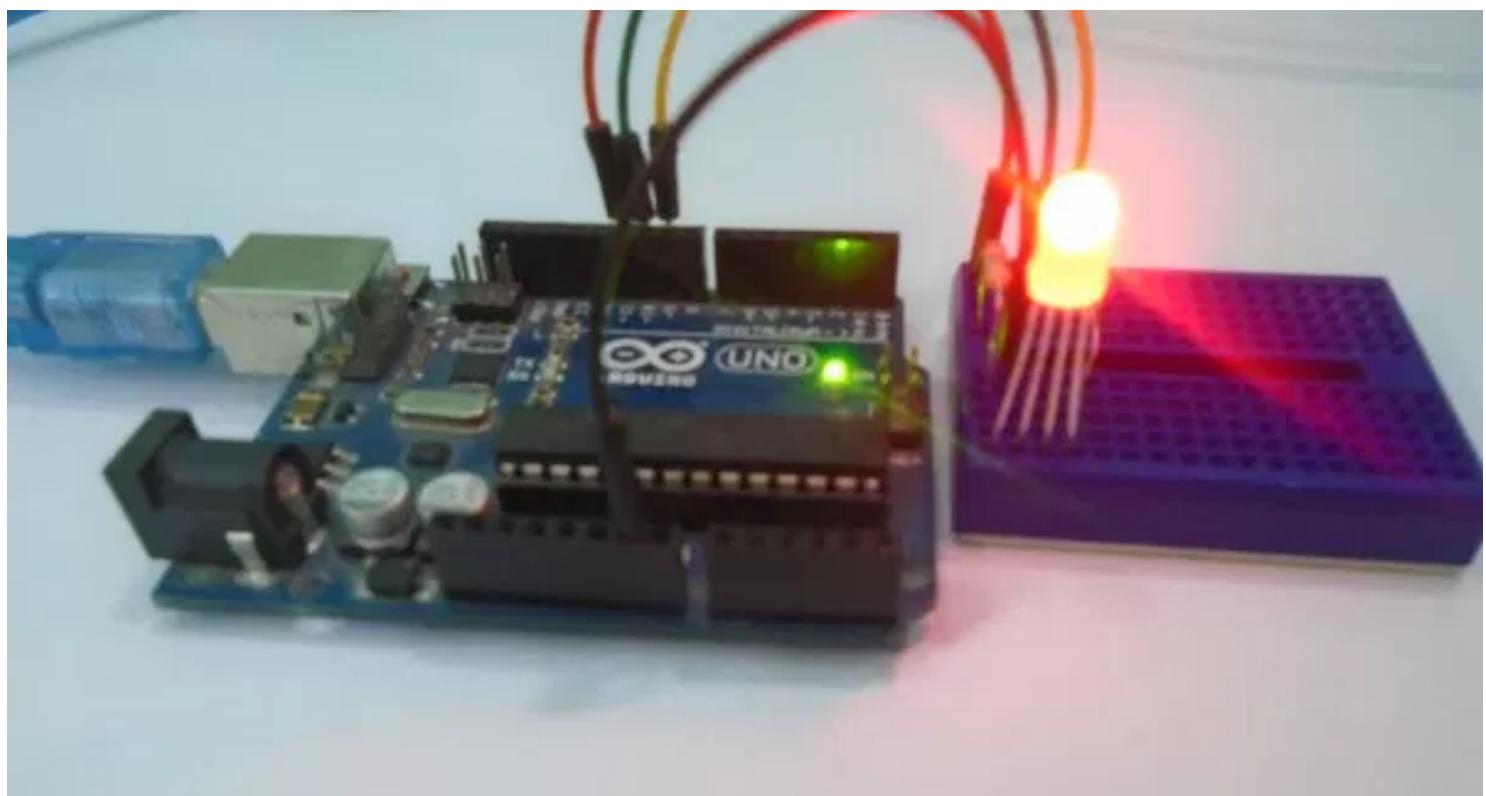
[\(http://electronicshobbyists.com/arduino-remote-control-communication-arduino-ir-sensor/\)](http://electronicshobbyists.com/arduino-remote-control-communication-arduino-ir-sensor/)



PIR MOTION SENSOR WITH ARDUINO | HC-SR01 PIR MOTION SENSOR ARDUINO INTERFACING ([HTTP://ELECTRONICSHOBBYISTS.COM/PIR-MOTION-SENSOR-WITH-ARDUINO-HC-SR01-PIR-MOTION-SENSOR-ARDUINO-INTERFACING/](http://electronicshobbyists.com/pir-motion-sensor-with-arduino-hc-sr01-pir-motion-sensor-arduino-interfacing/))

⌚ October 7, 2017

(<http://electronicshobbyists.com/pir-motion-sensor-with-arduino-hc-sr01-pir-motion-sensor-arduino-interfacing/>)



ARDUINO RGB LED TUTORIAL ([HTTP://ELECTRONICSHOBBYISTS.COM/ARDUINO-RGB-LED-TUTORIAL/](http://electronicshobbyists.com/arduino-rgb-led-tutorial/))

⌚ April 19, 2017

(<http://electronicshobbyists.com/arduino-rgb-led-tutorial/>)

LEAVE A REPLY

Your email address will not be published.

Comment

Name*

Email*

Website

SUBMIT

Notify me of follow-up comments by email.

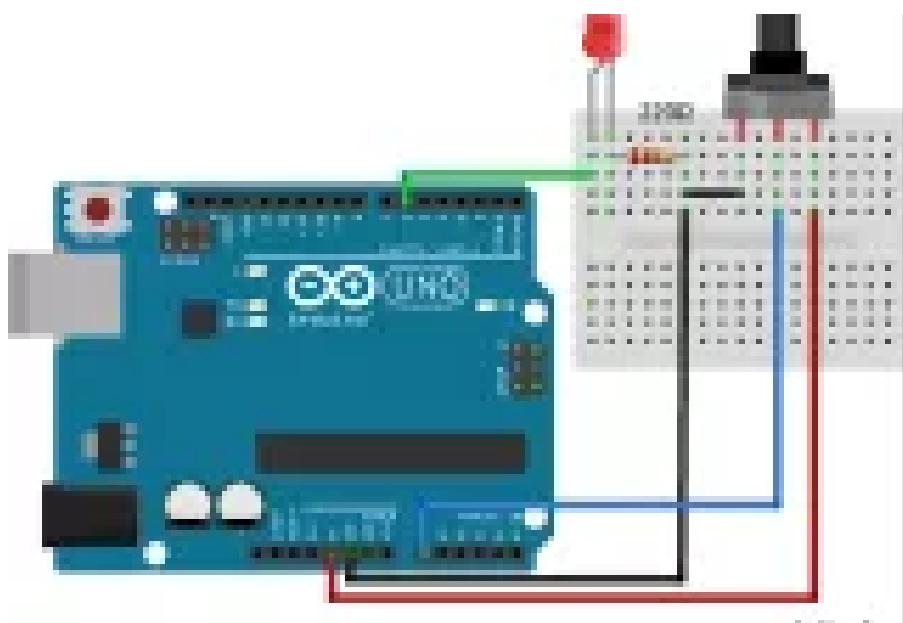
Notify me of new posts by email.

RECENT POSTS

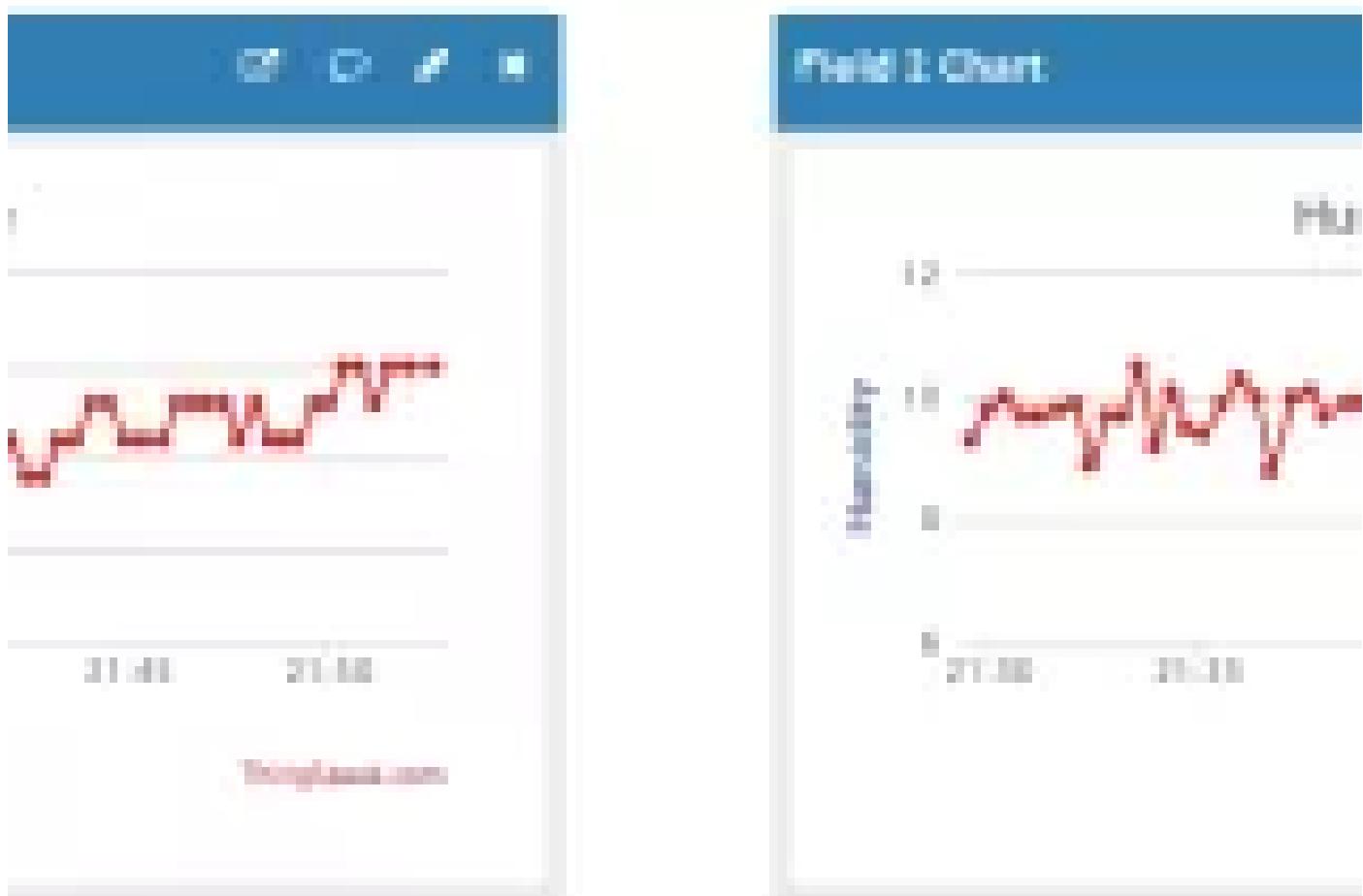
Arduino PWM Tutorial (<http://electronicshobbyists.com/arduino-pwm-tutorial/>)

⌚ January 9, 2018





(<http://electronicshobbyists.com/arduino-pwm-tutorial/>)



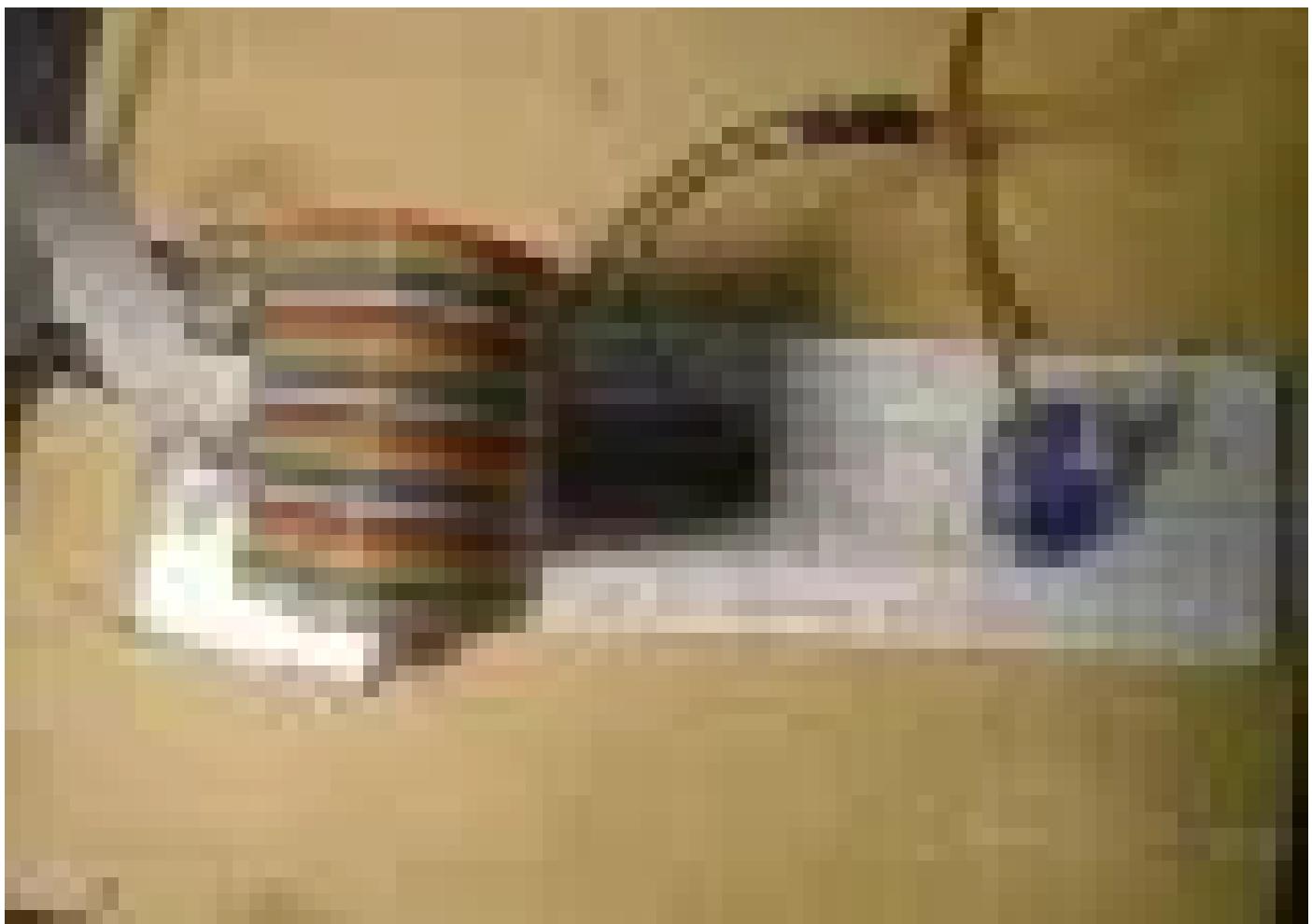
(<http://electronicshobbyists.com/raspberry-pi-sending-data-to-thingspeak-simplest-raspberry-pi-iot-project/>)



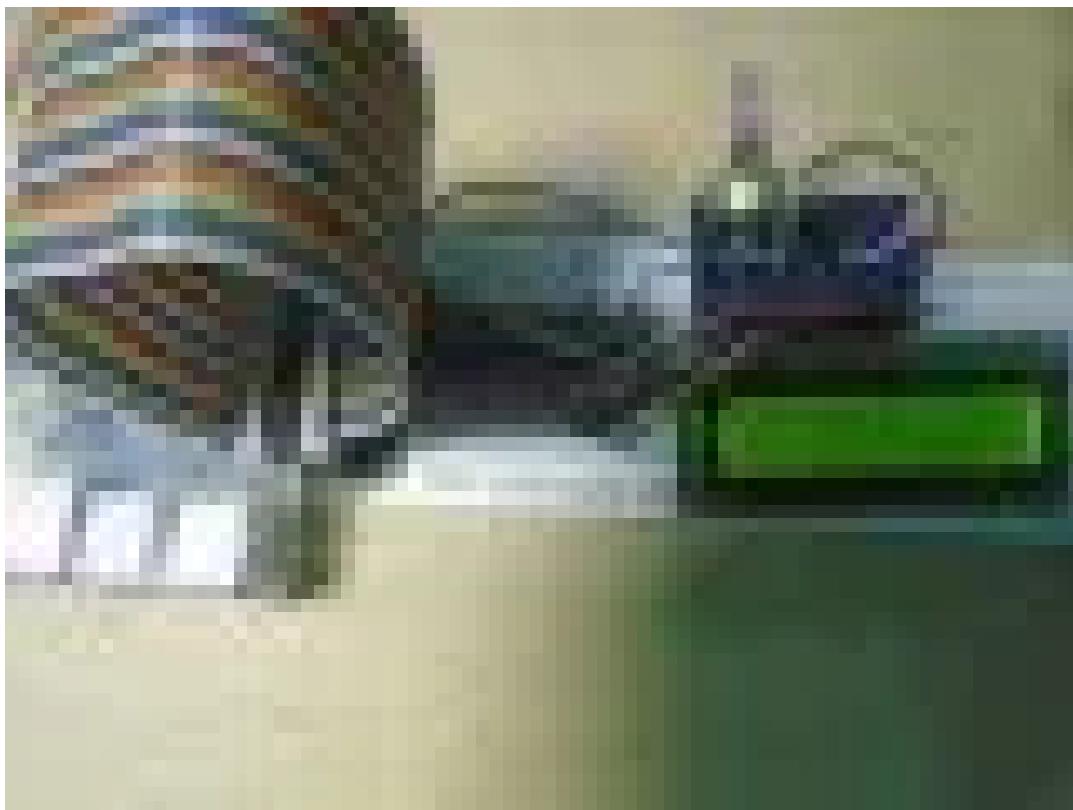
(<http://electronicshobbyists.com/raspberry-pi-pwm-tutorial-control-brightness-of-led-and-servo-motor/>)

Servo Motor Interfacing with Raspberry pi | Raspberry pi Servo Control (<http://electronicshobbyists.com/servo-motor-interfacing-with-raspberry-pi-raspberry-pi-servo-control/>)

© November 18, 2017



[\(http://electronicshobbyists.com/servo-motor-interfacing-with-raspberry-pi-raspberry-pi-servo-control/\)](http://electronicshobbyists.com/servo-motor-interfacing-with-raspberry-pi-raspberry-pi-servo-control/)



[\(http://electronicshobbyists.com/raspberry-pi-lcd-display-interfacing/\)](http://electronicshobbyists.com/raspberry-pi-lcd-display-interfacing/)

**Connect LCD with Raspberry pi |
Raspberry pi LCD Display
Interfacing**
[\(http://electronicshobbyists.com/pi-lcd-display-interfacing/\)](http://electronicshobbyists.com/pi-lcd-display-interfacing/)

© November 12, 2017

POPULAR POSTS



[\(http://electronicshobbyists.com/relay-module-interfacing-with-arduino-arduino-relay-module/\)](http://electronicshobbyists.com/relay-module-interfacing-with-arduino-arduino-relay-module/)

Arduino Real time clock (RTC) and Temperature Monitor using DS3231 module (<http://electronicshobbyists.com/arduino-real-time-clock-rtc-and-temperature-monitor-using-ds3231-module/>)

© June 7, 2017



(<http://electronicshobbyists.com/arduino-real-time-clock-rtc-and-temperature-monitor-using-ds3231-module/>)

Temperature and Humidity Sensor DHT22 Arduino Interfacing | Tutorial (<http://electronicshobbyists.com/temperature-and-humidity-sensor-dht22-arduino-interfacing-tutorial/>)

© January 4, 2017



[\(http://electronicshobbyists.com/temperature-and-humidity-sensor-dht22-arduino-interfacing-tutorial/\)](http://electronicshobbyists.com/temperature-and-humidity-sensor-dht22-arduino-interfacing-tutorial/)

Logging data to Database using Arduino Ethernet shield (<http://electronicshobbyists.com/logging-data-to-database-using-arduino-ethernet-shield/>)

⌚ June 14, 2017



[\(http://electronicshobbyists.com/logging-data-to-database-using-arduino-ethernet-shield/\)](http://electronicshobbyists.com/logging-data-to-database-using-arduino-ethernet-shield/)

Interfacing Nokia 5110 LCD with Arduino | Nokia 5110 Arduino Tutorial (<http://electronicshobbyists.com/interfacing-nokia-5110-lcd-with-arduino-nokia-5110-arduino-tutorial/>)

© March 22, 2017



[\(http://electronicshobbyists.com/interfacing-nokia-5110-lcd-with-arduino-nokia-5110-arduino-tutorial/\)](http://electronicshobbyists.com/interfacing-nokia-5110-lcd-with-arduino-nokia-5110-arduino-tutorial/)

