

Grove Beginner Kit for Arduino Projects

Project 1: Intrusion Alarm



Overview

This wiki introduces how to make a intrusion alarm.

Feature

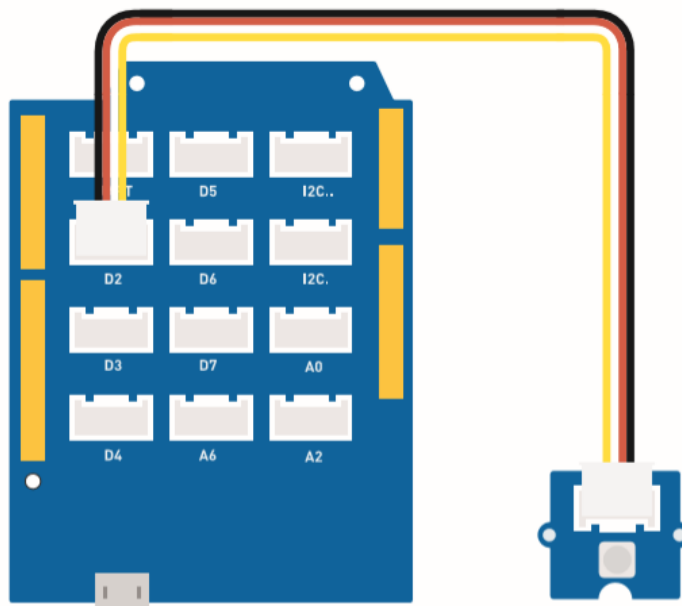
- The PIR motion sensor can detect people if in the area, then alarm triggered.

Component required

- **Grove Beginner Kit** [<https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>]
- **MINI PIR motion sensor** [<https://www.seeedstudio.com/Grove-mini-PIR-motion-sensor-p-2930.html>]

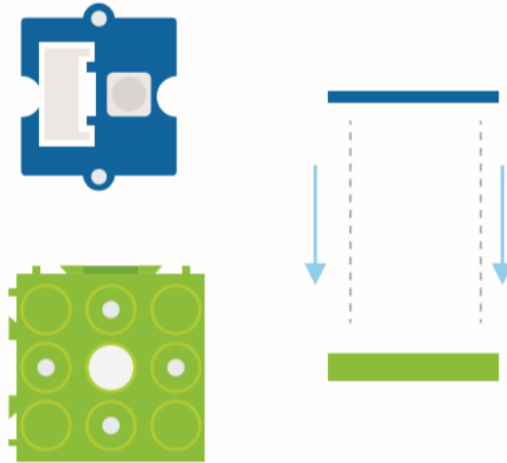
Hardware Connection

Please follow the same color line to connect each sensor on the board, put the PIR motion sensor grove cable to the D2.



Assembly instructions

Assembly Instructions



Note

The buzzer (D5) and LED (D4) are embedded in the board.

Arduino Instructions

Step 1. Download the **Aruidno IDE**

[<https://www.arduino.cc/en/Main/software>]

Step 2. Follow the connection picture connect all the sensor on the board.

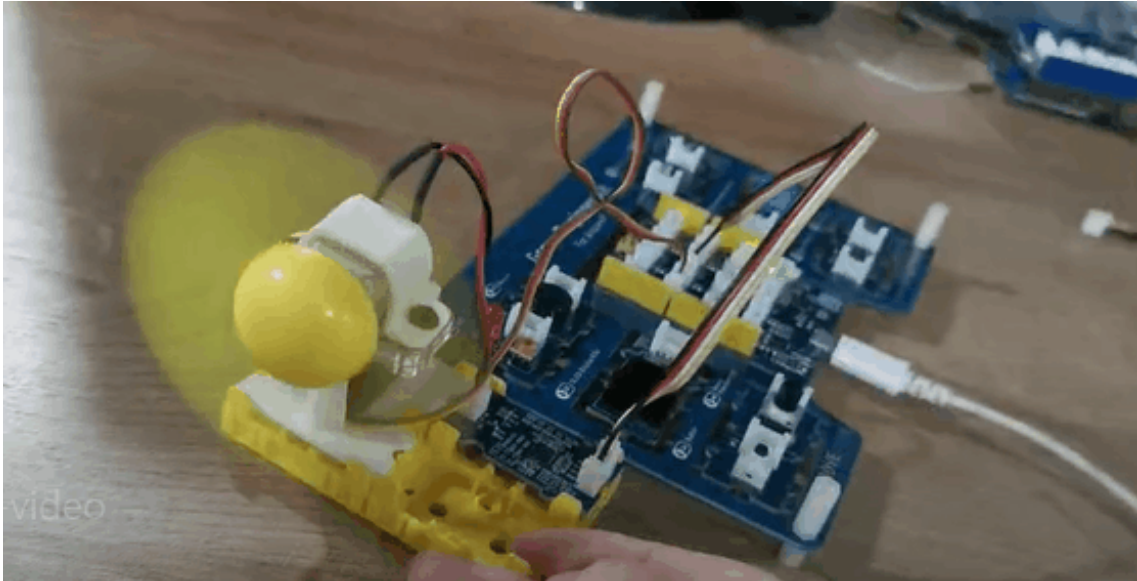
Step 3. Place the Mini PIR motion sensor in the location of the detection.

Step 4. Copy the code stick on the Aruino IDE then upload it.

Code

```
1  #define PIR_MOTION_SENSOR 2 //Use pin 2 to receive the signal
2
3  int BuzzerPin = 5; // set D5 as buzzer
4  int LED_RAD = 4; // set D4 as LED
5
6  void setup() {
7    Serial.begin(9600);
8    pinMode(PIR_MOTION_SENSOR, INPUT);
9    pinMode(BuzzerPin, OUTPUT);
10   pinMode(LED_RAD, OUTPUT);
11 }
12
13 void loop() {
14
15   if (digitalRead(PIR_MOTION_SENSOR)) {
16     analogWrite(BuzzerPin, 100);
17     digitalWrite(LED_RAD, HIGH);
18     delay(3000);
19     analogWrite(BuzzerPin, 0);
20     digitalWrite(LED_RAD, LOW);
21     delay(4000);
22   }
23
24 }
```

project 2: Oscillating fan



Overview

This wiki introduce how to make a Mini fan to plase on your room keep cool.

Feature

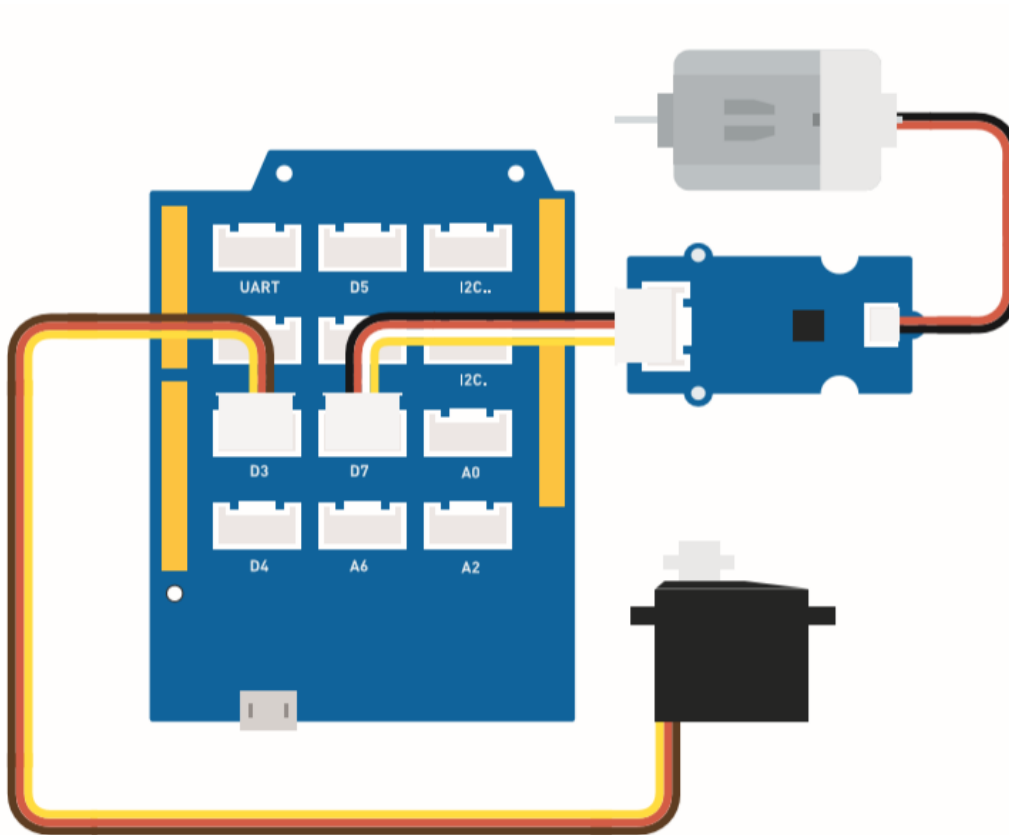
- Automatic swing fan

Component required

- **Grove Beginner Kit** [<https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>]
- **Grove mini fan** [<https://www.seeedstudio.com/Grove-Mini-Fan-v1-1.html>]
- **Grove-Servo** [<https://www.seeedstudio.com/Grove-Servo.html>]

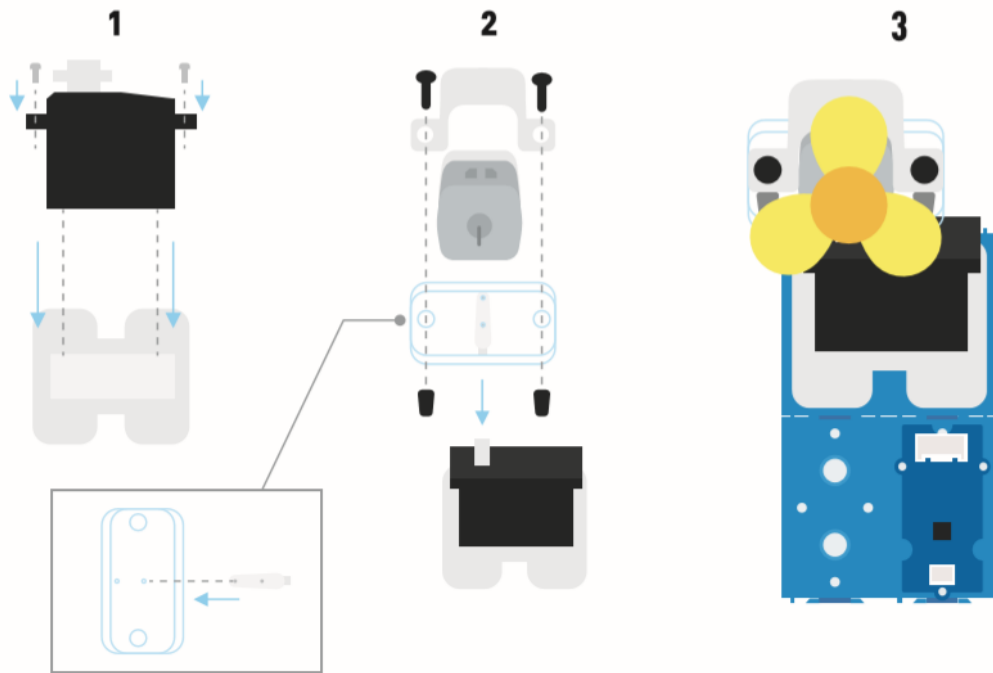
Hardware Connection

Please connect the fan grove cable to D7, Servo grove cable to D3.



Assembly instruction

Assembly Instructions



Arduino Instructions

Step 1. Download the **Aruidno IDE**

[<https://www.arduino.cc/en/Main/software>]

Step 2. Copy the Code and stick on the Arduino IDE

Step 3. Select the current port: Tools -> Port -> COM (number)

Step 4. Upload the code



Note

Please set the fan in the safety position.

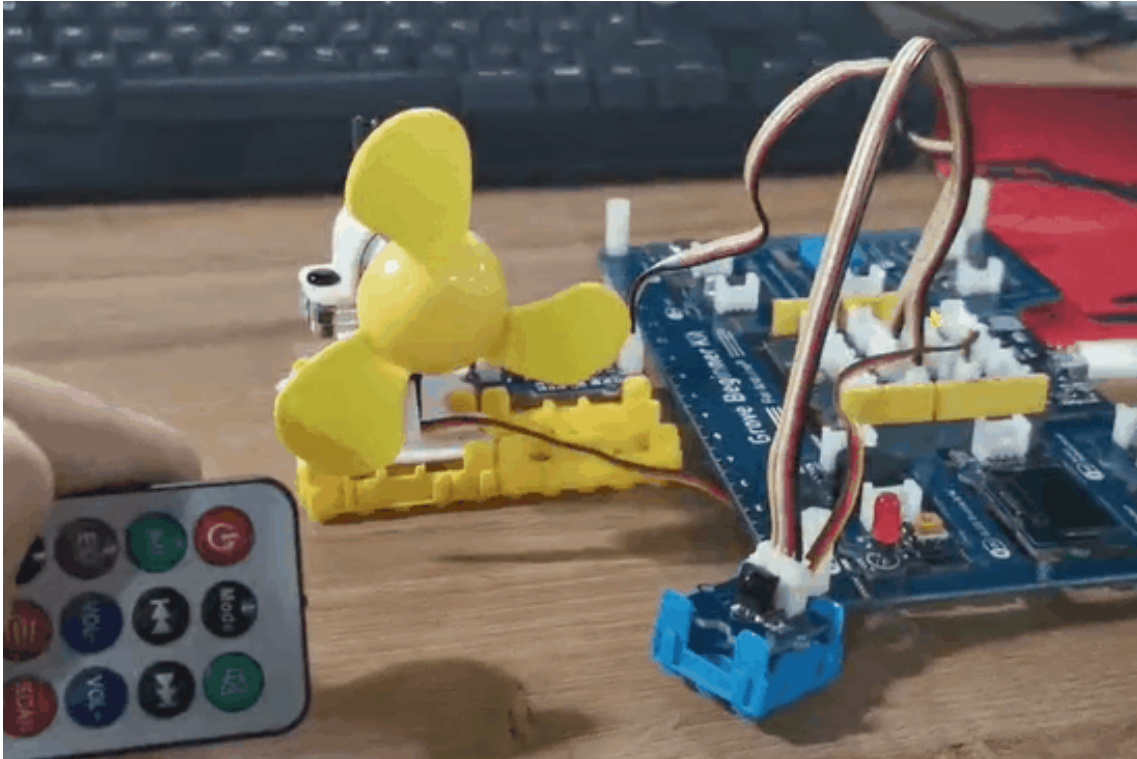
Code

```
1  #include <Servo.h>
2
```



```
3 Servo myservo; // create servo object to control a servo
4
5 int pos = 0; // variable to store the servo position
6 int fanPin = 7; // set D6 as control switch
7 int fanState = LOW;
8
9 void setup() {
10     Serial.begin(9600);
11     myservo.attach(3); // attaches the servo on pin 2 to the servo
12     pinMode(fanPin, OUTPUT);
13 }
14
15 void loop() {
16
17     fanState = HIGH;
18     digitalWrite(fanPin, fanState);
19
20     for (pos = 0; pos <= 100; pos += 1) { // goes from 0 to 100
21         // in steps of 1 degree
22         myservo.write(pos); // tell servo to go to position
23         delay(40); // waits 15ms for the motor to get to the position
24     }
25     for (pos = 100; pos >= 0; pos -= 1) { // goes from 100 to 0
26         myservo.write(pos); // tell servo to go to position
27         delay(40); // waits 15ms for the motor to get to the position
28     }
29 }
```

Project 3: Remote Control Oscillating Fan



Overview

This wiki introduces how to make a remote control oscillating fan.

Feature

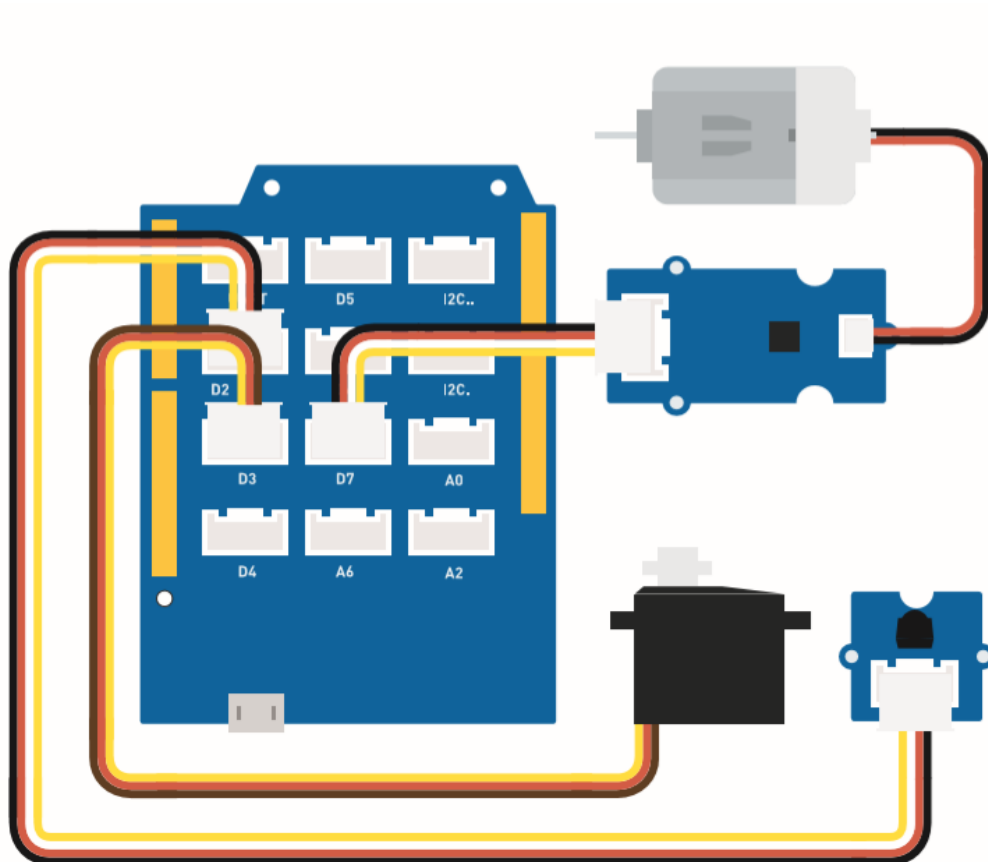
- The fan power controlled by the controller.
- The fan swing way is able to use remote control.

Component required

- **Grove Beginner Kit** [<https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>]
- **Grove mini fan** [<https://www.seeedstudio.com/Grove-Mini-Fan-v1-1.html>]
- **Grove-Servo** [<https://www.seeedstudio.com/Grove-Servo.html>]
- **Grove - IR (Infrared) Receiver** [<https://www.seeedstudio.com/Grove-Infrared-Receiver.html>]

Hardware Connection

Please follow the same color line to connect each sensor on the board. Please connect the fan grove cable to D7, servo grove cable to D3, IR grove cable to D2.

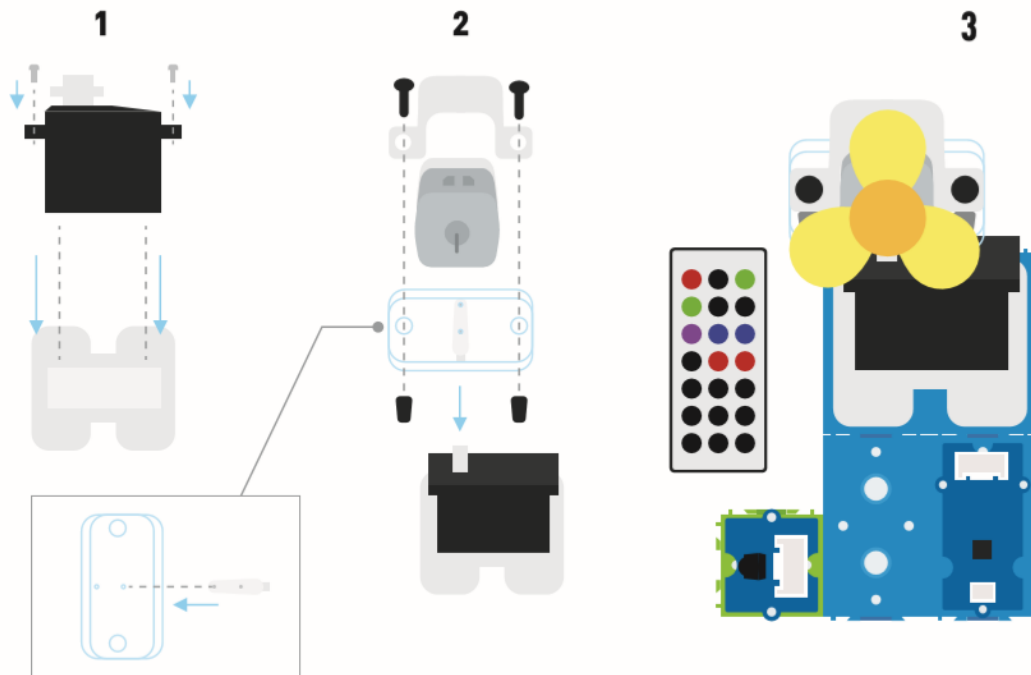


This is controller bottom function.



Assembly instruction

Assembly Instructions



Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2. Download the **Aruidno IDE**

[<https://www.arduino.cc/en/Main/software>]

Step 3. Navigate to **Sketch -> Include Library -> Manage Libraries**, search **IRremote** then install it.

Step 4. Copy the code stick on the Aruino IDE then upload it.

Step 5. Place the Fan in the safety position, try to press the button make sure it can work safely.



Note

Refer How to [install library](#)

[https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.

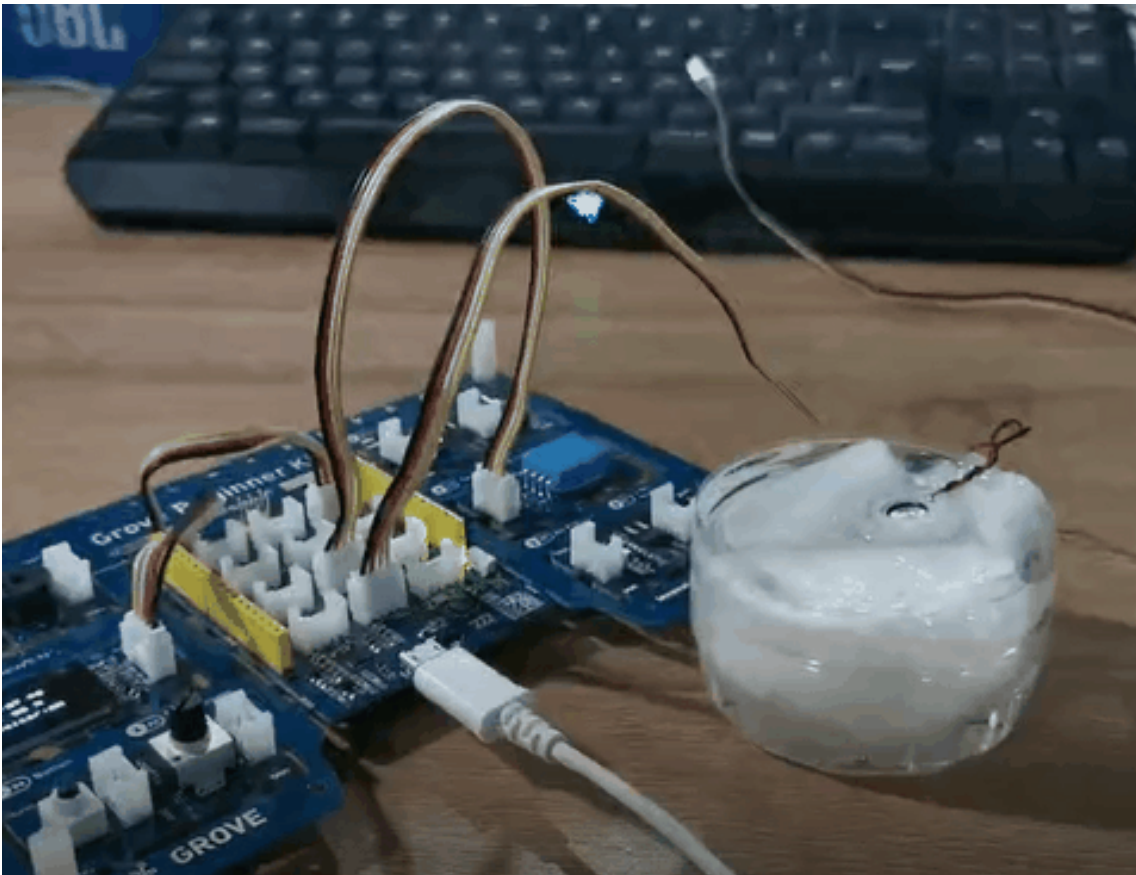
Code

```
1  #include <IRremote.h>
2  #include <Servo.h>
3
4  Servo myservo; // create servo object to control a servo
5  int RECV_PIN = 2; // set pin 2 as IR control
6
7  IRrecv irrecv(RECV_PIN);
8
9  decode_results results;
10
11 int pos = 90; // variable to store the servo position
12 int fanPin = 7; // set D6 as control switch
13 int fanState = LOW;
14 int IO = 0;
15
16 void setup()
17 {
18   Serial.begin(9600);
19   Serial.println("Enabling IRin"); // remind enabling IR
20   irrecv.enableIRIn(); // Start the receiver
21   Serial.println("Enabled IRin");
22   myservo.attach(3); // attaches the servo on pin 2 to
23   pinMode(fanPin, OUTPUT);
24
25 }
26
27 // power_encode 2155829415      left 2155870215 right
28
29 void loop() {
30   if (irrecv.decode(&results)) { //checking IR signal
31     if (results.value == 2155829415) { // Power off/on
32       IO++;
33       if (IO % 2 == 0) {
```

```
34     fanState = HIGH;
35     digitalWrite(fanPin, fanState);
36     delay(100);
37 }
38 else {
39     fanState = LOW;
40     digitalWrite(fanPin, fanState);
41     delay(100);
42 }
43 }
44
45 if (results.value == 2155821255 ) { // fan swing
46     for (pos; pos <= 89; pos += 1) { // goes from 0 deg
47         // in steps of 1 degree
48         myservo.write(pos); // tell servo to
49
50         delay(40); // waits 15ms f
51         if (irrecv.decode(&results)) {
52             irrecv.resume();
53             if (results.value == 2155870215)
54                 break;
55         }
56     }
57 }
58
59 if (results.value == 2155870215 ) { // fan swing
60     for (pos; pos >= 1; pos -= 1) { // goes from 90 deg
61         myservo.write(pos); // tell servo to
62         delay(40); // waits 15ms f
63
64         if (irrecv.decode(&results)) {
65             irrecv.resume();
66             if (results.value == 2155821255)
67                 break;
68         }
69     }
70 }
71 Serial.println(pos);
72 Serial.println(results.value, HEX);
73 Serial.println(results.value);
74 irrecv.resume(); //receive next in
```

```
75  
76   }  
77   delay(100);  
78   }
```

Project 4: Smart Humidifier



Overview

This wiki introduces how to make water atomization to keep indoor humidity normal.

Feature

- Automatically use the water atomization when the humidity is low.

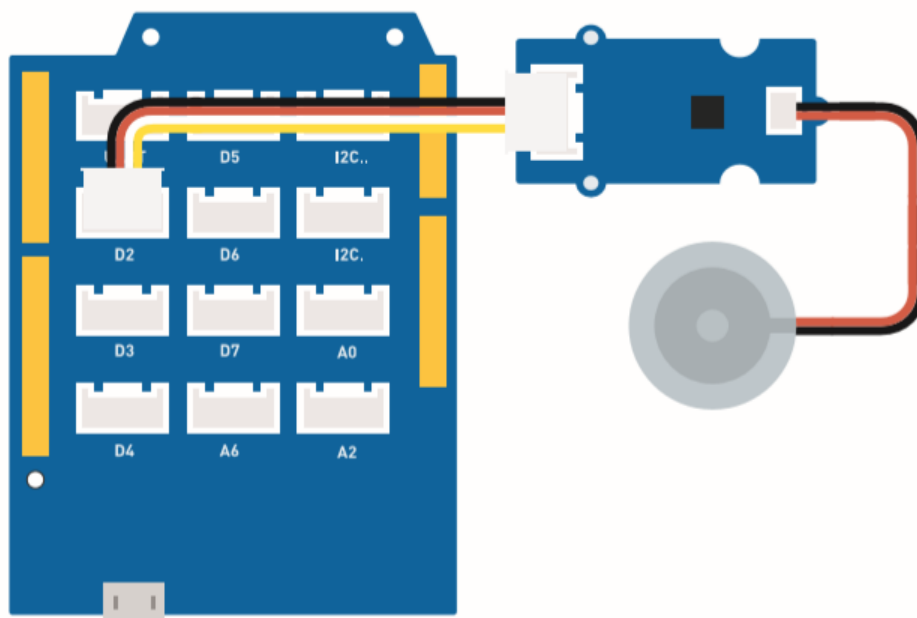
- Display the temperature and humidity in real-time.

Component required

- **Grove Beginner Kit** [<https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>]
- **Grove water atomization** [<https://www.seeedstudio.com/Grove-Water-Atomization-v1-0.html>]

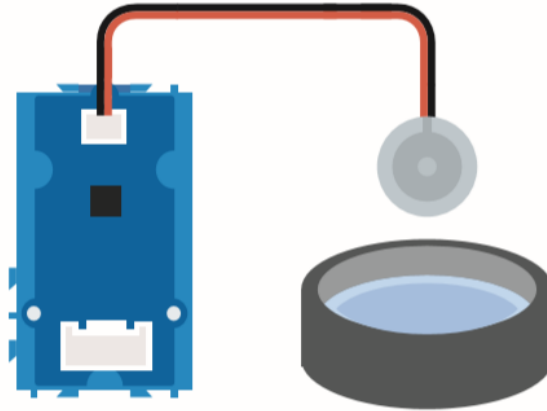
Hardware Connection

Please follow the same color line to connect each sensor on the board. Connect the Grove water atomization cable to D2.



Assembly instructions

Assembly Instructions



Arduino Instructions

Step 1. Follow the connection picture connect all the sensor on the board.

Step 2 Download the **Aruidno IDE**
[<https://www.arduino.cc/en/Main/software>]

Step 3 Navigate to **Sketch -> Include Library -> Manage Libraries**, search **U8g2** then install it.

Step 4 Download the **Grove_Temperature_And_Humidity_Sensor library** [https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor] and install it

Step 5. Copy the code stick on the Aruino IDE then upload it.

Step 6. Prepare a contain with full water then put the water atomization on the water.

**Note**

Refer How to [install library](#)

[https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.

**Note**

Prepare some tissue put on the water, let the water atomization keep afloat. The function of tissue is lead water to the transducer and keep upper side of transducer above water.

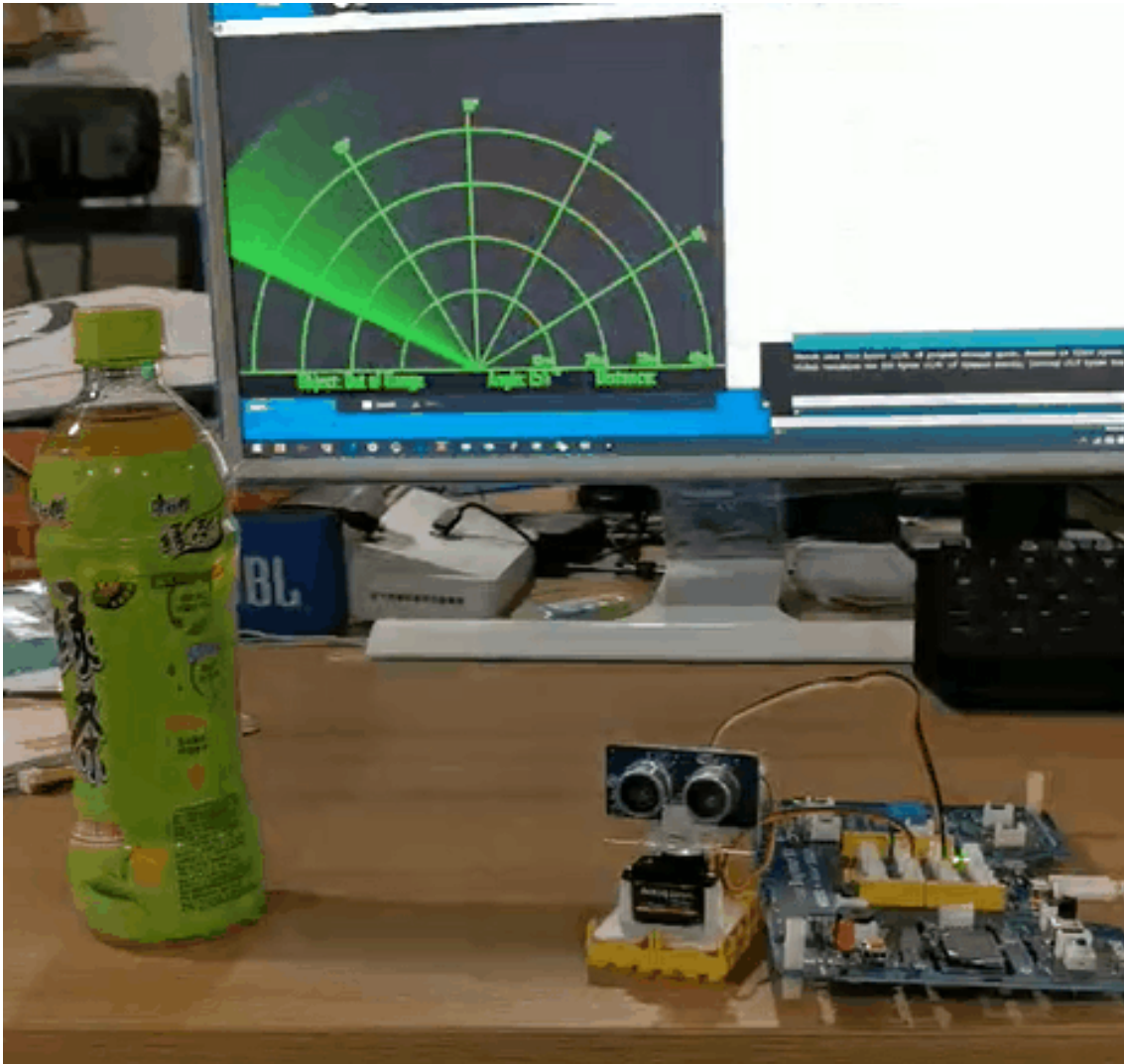
Code

```
1  #include <Arduino.h>
2  #include <U8x8lib.h>
3
4  #include "DHT.h"
5  #define DHTTYPE DHT11 // DHT 11
6  #define DHTPIN 3 // what pin we're connected to
7  DHT dht(DHTPIN, DHTTYPE);
8
9  #include <Wire.h>
10
11 U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_I
12
13 void setup(void) {
14     Serial.begin(115200);
15     u8x8.begin();
16     u8x8.setFlipMode(1);
17     Wire.begin();
18     dht.begin();
19     pinMode(2, INPUT);
20 }
21
22 void loop(void) {
```



```
23 float temp_hum_val[2] = {0};
24 int b;
25 int c;
26 if (!dht.readTempAndHumidity(temp_hum_val)) {
27     b = temp_hum_val[0];
28     c = temp_hum_val[1];
29 }
30 else{
31     Serial.println("Failed to get temprature and humidit
32     }
33     u8x8.setFont(u8x8_font_chroma48medium8_r); // choose
34     u8x8.setCursor(0, 0);
35     u8x8.print("Temp: ");
36     u8x8.setCursor(5, 0);
37     u8x8.print(c);
38     u8x8.setCursor(8, 0);
39     u8x8.print("*C");
40     u8x8.setCursor(0, 10);
41     u8x8.print("Hum: ");
42     u8x8.setCursor(5, 10);
43     u8x8.print(b);
44     u8x8.setCursor(8, 10);
45     u8x8.print("%");
46     u8x8.setCursor(0, 20);
47     u8x8.print("atomizer: ");
48
49     if(b > 70){
50         u8x8.setCursor(9, 20);
51         u8x8.print("OFF");
52         digitalWrite(2, LOW); // atomization stopped
53     }
54     if(b <= 70) {
55         u8x8.setCursor(9, 20);
56         u8x8.print("ON ");
57         digitalWrite(2, HIGH); // atomize
58     }
59     delay(1000);
60
61 }
```

Project 5: Ultrasonic Radar



Overview

This wiki introduce how to make a Ultrasonic Radar to detect the object and distance.

Feature

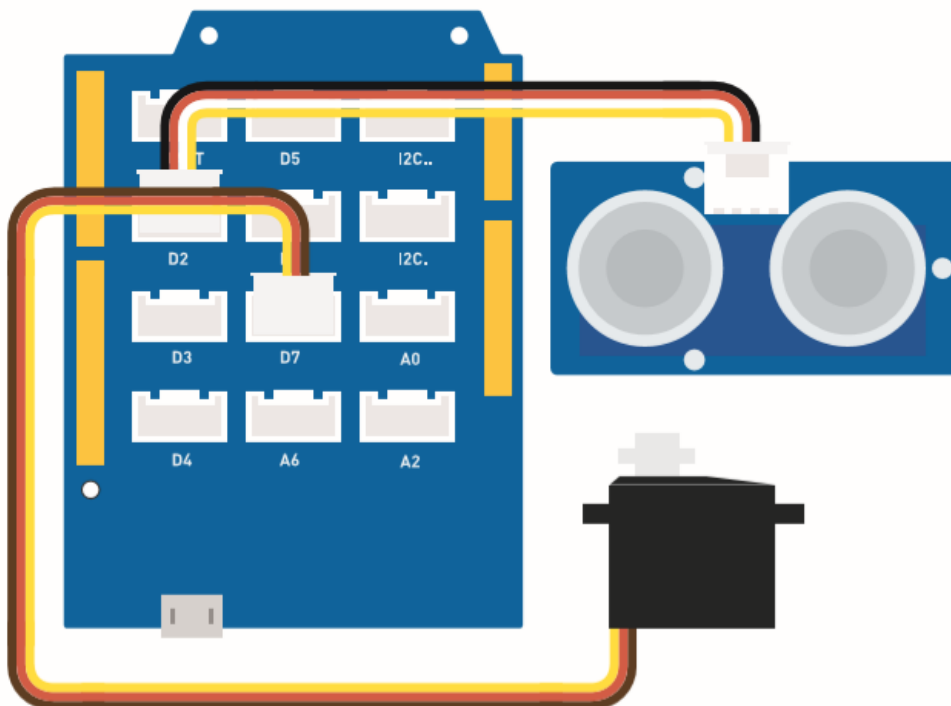
- Detect the object distance
- Scan if exist the object in around

Component required

- **Grove Beginner Kit** [<https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>]
- **Grove-Servo** [<https://www.seeedstudio.com/Grove-Servo.html>]
- **Grove-Ultrasonic distance sensor** [<https://www.seeedstudio.com/Grove-Ultrasonic-Distance-Sensor.html>]

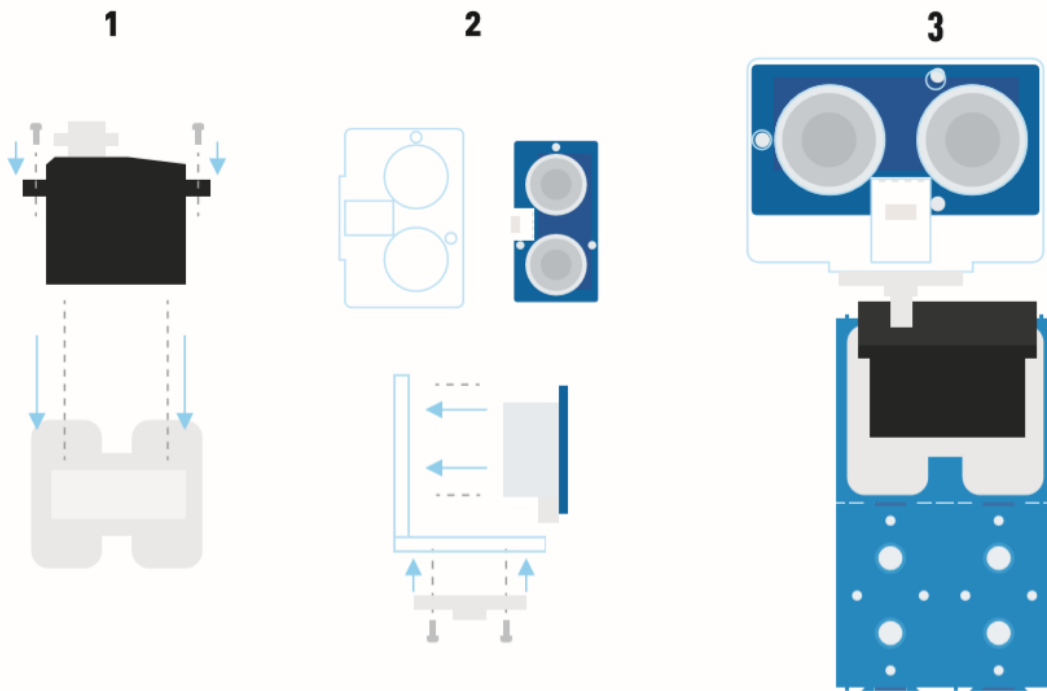
Hardware Connection

Please follow the picture, connect the ultrasonic sensor Grove cable to the D2, connect the servo to the D7.



Assembly instruction

Assembly Instructions



Arduino Instructions

Step 1. Follow the Connection to plug the cable on the port.

Step 2. Download the **Aruidno IDE**
[<https://www.arduino.cc/en/Main/software>]

Step 3. Download the **Processing**
[<https://processing.org/download/>]

Step 4. Download the **UltrasonicRanger** [https://github.com/Seeed-Studio/Seeed_Arduino_UltrasonicRanger/archive/master.zip]
Library from Github.

Step 5. Copy the Radar code and stick on the Arduino IDE then upload it.

Step 6. Download the **Processing**

[<https://processing.org/download/>] for display the Radar scan map.

Step 7. Copy the Radar-Processing Code and stick on the Processing.

Step 8. After the Servo start swing, click play on the Processing software.

**Note**

If you do not know how to install the library, Click [here](https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/#software) [https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/#software].

Radar code

```
1  #include <Servo.h>
2  #include "Ultrasonic.h"
3
4  int distance;
5  Servo myServo;
6
7  Ultrasonic ultrasonic(2);
8
9  void setup() {
10     Serial.begin(9600);
11     myServo.attach(7);
12 }
13 void loop() {
14
15     for(int pos = 15; pos <= 165; pos += 1){
16         myServo.write(pos);
17         delay(30);
18         distance = ultrasonic.MeasureInCentimeters();
19         Serial.print(pos);
20         Serial.print(",");
21         Serial.print(distance);
22         Serial.print(".");
```



```

23     }
24
25     for(int pos = 165; pos >= 15; pos -= 1){
26         myServo.write(pos);
27         delay(30);
28         distance = ultrasonic.MeasureInCentimeters();
29         Serial.print(pos);
30         Serial.print(",");
31         Serial.print(distance);
32         Serial.print(".");
33     }
34 }

```

Radar-Processing Code

```

1  import processing.serial.*; // imports library for serial
2  import java.awt.event.KeyEvent; // imports library for key events
3  import java.io.IOException;
4  Serial myPort; // defines Object Serial
5  // defubes variables
6  String angle="";
7  String distance="";
8  String data="";
9  String noObject;
10 float pixsDistance;
11 int iAngle, iDistance;
12 int index1=0;
13 int index2=0;
14 PFont orcFont;
15 void setup() {
16
17     size (1000, 720); // ***CHANGE THIS TO YOUR SCREEN RES
18     smooth();
19     myPort = new Serial(this,"COM14", 9600); // starts the
20     myPort.bufferUntil('.'); // reads the data from the se
21     orcFont = loadFont("AgencyFB-Bold-48.vlw");
22 }
23 void draw() {
24

```



```

25   fill(98,245,31);
26   textFont(orcFont);
27   // simulating motion blur and slow fade of the moving
28   noStroke();
29   fill(0,4);
30   rect(0, 0, width, height-height*0.065);
31
32   fill(98,245,31); // green color
33   // calls the functions for drawing the radar
34   drawRadar();
35   drawLine();
36   drawObject();
37   drawText();
38 }
39 void serialEvent (Serial myPort) { // starts reading da
40 // reads the data from the Serial Port up to the char
41 data = myPort.readStringUntil('.');
42 data = data.substring(0,data.length()-1);
43
44 index1 = data.indexOf(","); // find the character ','
45 angle= data.substring(0, index1); // read the data fr
46 distance= data.substring(index1+1, data.length()); //
47
48 // converts the String variables into Integer
49 iAngle = int(angle);
50 iDistance = int(distance);
51 }
52 void drawRadar() {
53   pushMatrix();
54   translate(width/2,height-height*0.074); // moves the
55   noFill();
56   strokeWeight(2);
57   stroke(98,245,31);
58   // draws the arc lines
59   arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,
60   arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_
61   arc(0,0,(width-width*0.479),(width-width*0.479),PI,TW
62   arc(0,0,(width-width*0.687),(width-width*0.687),PI,TW
63   // draws the angle lines
64   line(-width/2,0,width/2,0);
65   line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(r

```

```
66   line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(r
67   line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(r
68   line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(
69   line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(
70   line((-width/2)*cos(radians(30)),0,width/2,0);
71   popMatrix();
72 }
73 void drawObject() {
74   pushMatrix();
75   translate(width/2,height-height*0.074); // moves the
76   strokeWeight(9);
77   stroke(255,10,10); // red color
78   pixsDistance = iDistance*((height-height*0.1666)*0.02
79   // limiting the range to 40 cms
80   if(iDistance<40){
81     // draws the object according to the angle and the
82     line(pixsDistance*cos(radians(iAngle)),-pixsDistance*
83     }
84     popMatrix();
85   }
86   void drawLine() {
87     pushMatrix();
88     strokeWeight(9);
89     stroke(30,250,60);
90     translate(width/2,height-height*0.074); // moves the
91     line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(
92     popMatrix();
93   }
94   void drawText() { // draws the texts on the screen
95
96     pushMatrix();
97     if(iDistance>40) {
98       noObject = "Out of Range";
99     }
100    else {
101      noObject = "In Range";
102    }
103    fill(0,0,0);
104    noStroke();
105    rect(0, height-height*0.0648, width, height);
106    fill(98,245,31);
```

```
107   textSize(25);
108
109   text("10cm",width-width*0.3854,height-height*0.0833);
110   text("20cm",width-width*0.281,height-height*0.0833);
111   text("30cm",width-width*0.177,height-height*0.0833);
112   text("40cm",width-width*0.0729,height-height*0.0833);
113   textSize(40);
114   text("Object: " + noObject, width-width*0.875, height
115   text("Angle: " + iAngle + " Â°", width-width*0.48, hei
116   text("Distance: ", width-width*0.26, height-height*0.
117   if(iDistance<40) {
118   text("      " + iDistance + " cm", width-width*0.225
119   }
120   textSize(25);
121   fill(98,245,60);
122   translate((width-width*0.4994)+width/2*cos(radians(30
123   rotate(-radians(-60));
124   text("30Â°",0,0);
125   resetMatrix();
126   translate((width-width*0.503)+width/2*cos(radians(60
127   rotate(-radians(-30));
128   text("60Â°",0,0);
129   resetMatrix();
130   translate((width-width*0.507)+width/2*cos(radians(90
131   rotate(radians(0));
132   text("90Â°",0,0);
133   resetMatrix();
134   translate(width-width*0.513+width/2*cos(radians(120))
135   rotate(radians(-30));
136   text("120Â°",0,0);
137   resetMatrix();
138   translate((width-width*0.5104)+width/2*cos(radians(15
139   rotate(radians(-60));
140   text("150Â°",0,0);
141   popMatrix();
142 }
```