XYC-WB-DC Microwave Radar Motion sensor module

Technical Manual Rev 1r0

The module has a wider range from 6 to 9 meters, 360 detection angle. Other applications for such modules include security,body sensors toys, industrial automation and control, auto-sensing electrical equipment, and battery-powered automatic control. Compatible in all gizDuino and Arduino boards.

Features:

- Wide range of motion detection
- The microwave module will also work through walls but the signal can be blocked by metal
- The module will output high signal (3.3V) when motion is detected.

General Specifications:

Operating Frequency: 5.8GHz
Detection angle: 360 degrees
Detection range: 6 to 9 meters
Working voltage: 3.3 to 20V DC
Standby current: <3mA
Transmit power: <2mW
Operating temperature range: -20 to +80 C
Dimensions: 32mm x 23mm

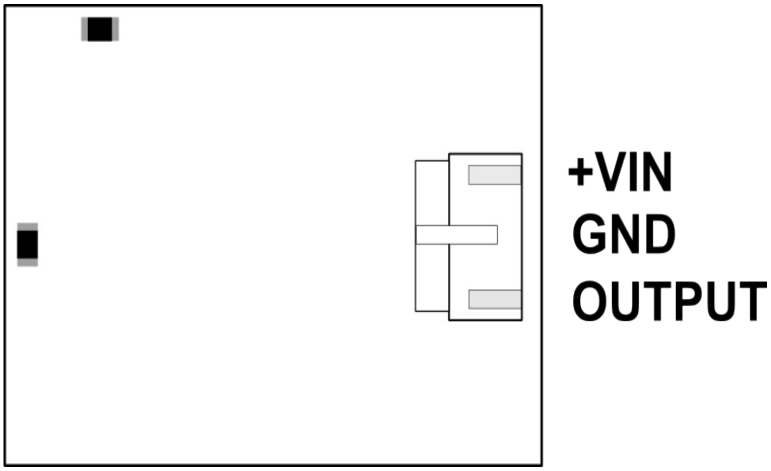


Figure 1. Pinouts Presentation

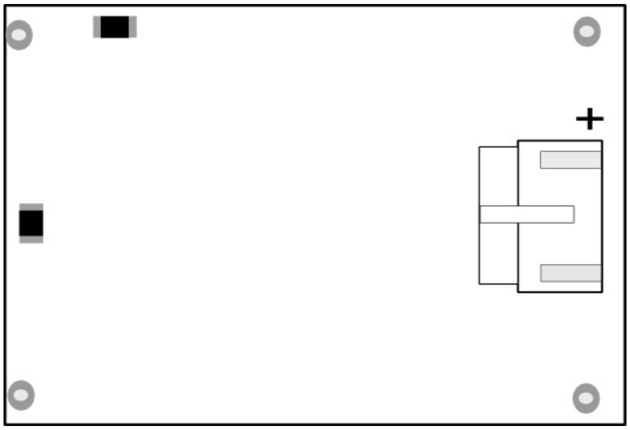


Figure 2. PCB Bottom Layer Guide

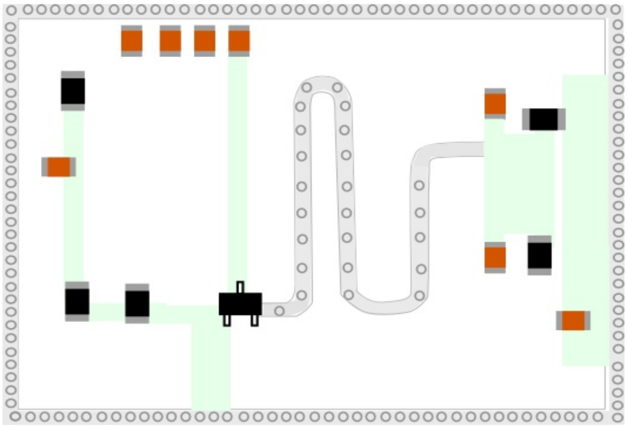
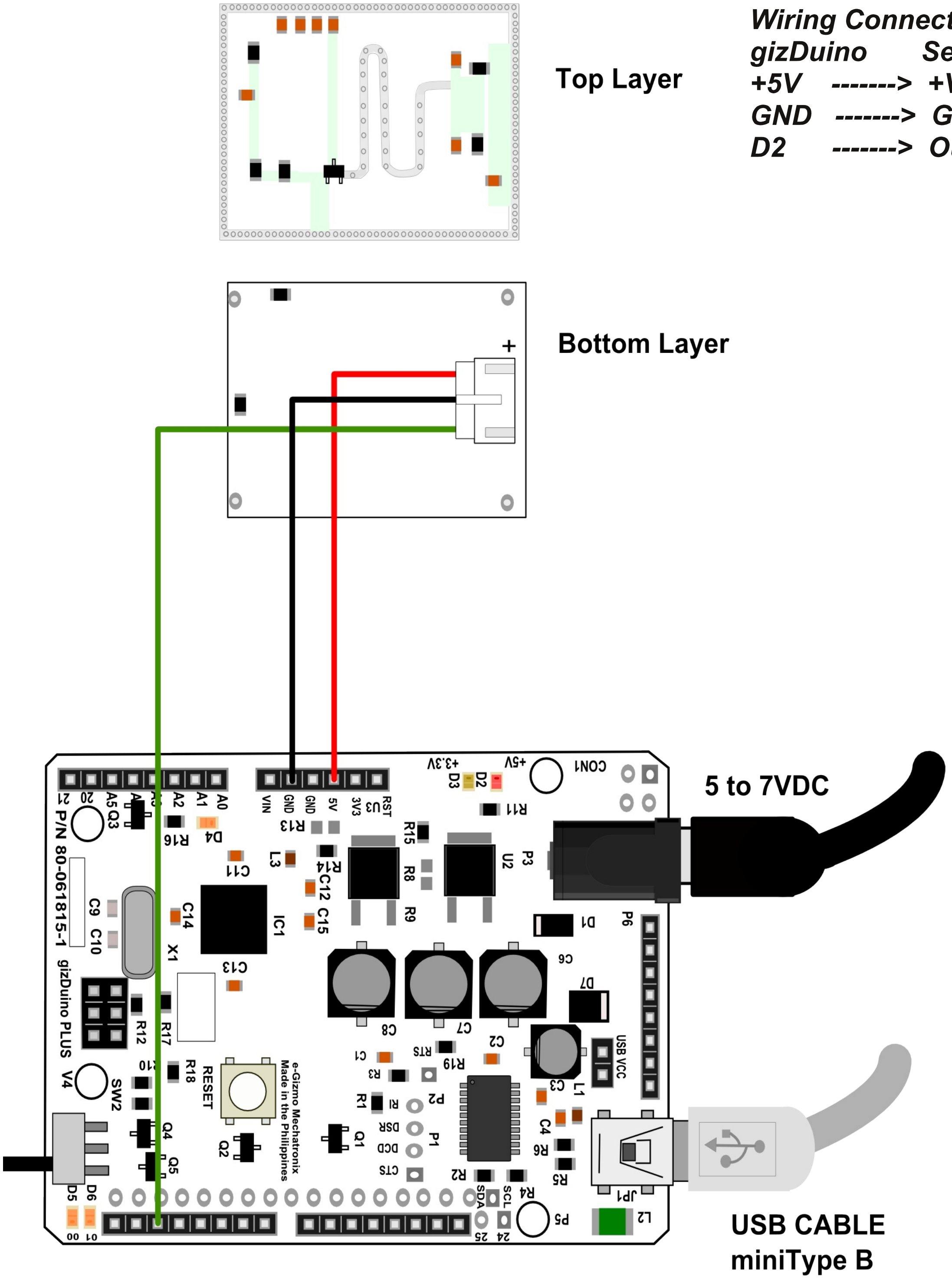


Figure 3. PCB Top Layer Guide

Wiring Connections:

gizDuino		***Sensor***
+5V	***------->***	***+VIN (3.3 to 20V)***
GND	***------->***	***GND***
D2	***------->***	***Output***

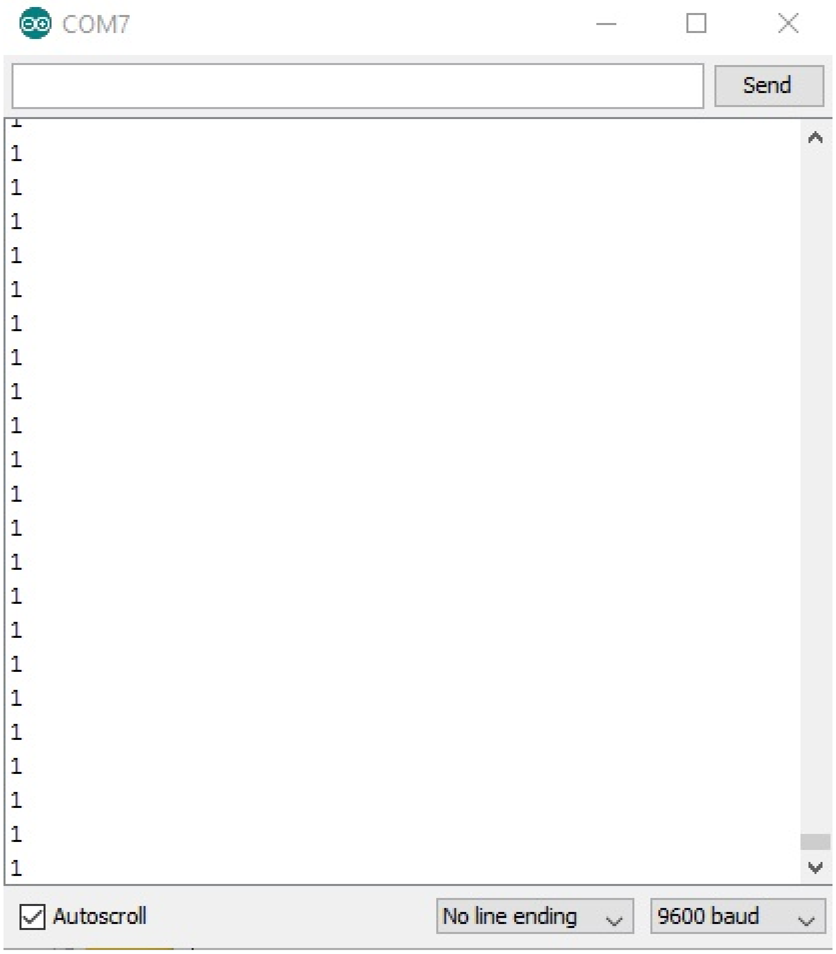


gizDuino PLUS w/ ATmega644P

Sample Code

```
/*
  XYC-WB-DC Microwave Radar
  Motion sensor module

  This sample sketch is for reading the
  Output and display it to the Serial Monitor.
  No detection = 0; detected motion = 1.

  Wiring connections:
  ==================
  gizDuino   to   Microwave Radar Motion sensor
  +5V     -------> +VIN (3.3V to 20V DC)
  GND     -------> GND
  D2      -------> OUTPUT

  Codes by:
  e-Gizmo Mechatronix Central
  http://www.e-gizmo.com
  June 19,2017

 */

//pin assignment for input
int Output = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(Output, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int SensorVal = digitalRead(Output);
  // print out the state of sensor:
  Serial.println(SensorVal);
  delay(1);        // delay in between reads for stability
}
```

Serial Monitor



```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

☑ Autoscroll

No line ending

9600 baud

Figure 4. Serial Monitor