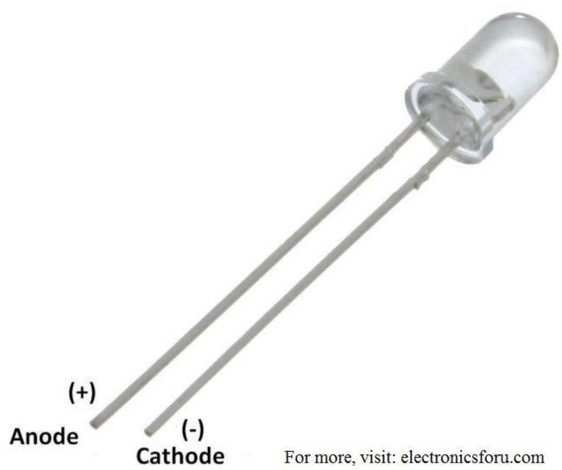
IR Sensor Interfacing

An Infrared light-emitting diode (IR LED) is a special-purpose LED that emits infrared rays ranging from 700 nm to 1 mm wavelength. Different IR LEDs may produce infrared light of differing wavelengths, just like other LEDs produce light of different colors.
IR sensor is a device that uses infrared technology to detect objects or changes in the environment. IR sensors can detect a wide range of physical properties such as temperature, motion, and proximity.
IR LEDs are usually made of gallium arsenide or aluminum gallium arsenide. In complement with IR receivers, these are commonly used as sensors.
The appearance of an IR LED is the same as a common LED. Since the human eye cannot see infrared radiation, it is not possible for a person to identify if an IR LED is working. A camera on a cell phone camera solves this problem. The IR rays from the IR LED in the circuit are shown in the camera.

Pin Diagram of IR LED



IR LED Pins

An IR LED is a type of diode or simple semiconductor. Electric current is allowed to flow in only one direction in diodes. As the current flows, electrons fall from one part of the diode into holes on another part.
In order to fall into these holes, the electrons must shed energy in the form of photons, which produce light.
It is necessary to modulate the emission from the IR diode to use it in the electronic application to prevent spurious triggering. Modulation makes the signal from IR LED stand out above the noise.
Infrared diodes have a package that is opaque to visible light but transparent to infrared. The massive use of IR LEDs in remote controls and safety alarm systems has drastically reduced the pricing of IR diodes in the market.

IR Sensor

An IR sensor is an electronic device that detects IR radiation falling on it. Proximity sensors (used in touchscreen phones and edge-avoiding robots), contrast sensors (used in line following robots), and obstruction counters/sensors (used for counting goods and in burglar alarms) are some applications involving IR sensors.

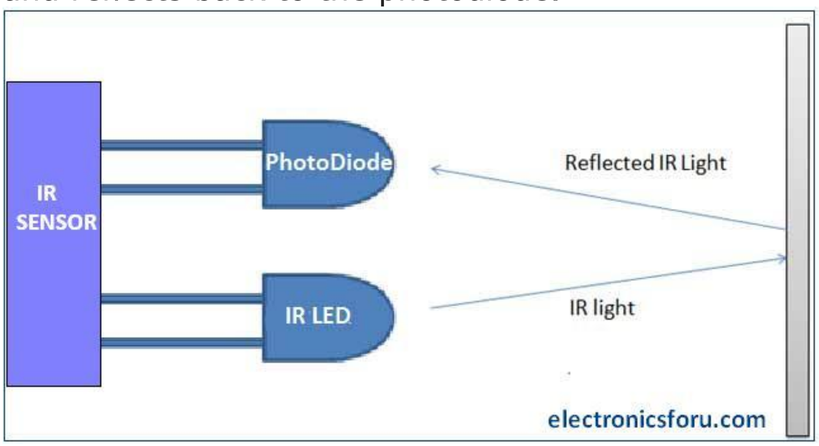
IR Sensor Working Principle

An IR sensor consists of two parts, the emitter circuit, and the receiver circuit. This is collectively known as a photo-coupler or an optocoupler.

The emitter is an IR LED and the detector is an IR photodiode. The IR photodiode is sensitive to the IR light emitted by an IR LED. The photodiode's resistance and output voltage change in proportion to the IR light received. This is the underlying working principle of the IR sensor.

The type of incidence can be direct incidence or indirect incidence. In direct incidence, the IR LED is placed in front of a photodiode with no obstacle.

In indirect incidence, both the diodes are placed side by side with an opaque object in front of the sensor. The light from the IR LED hits the opaque surface and reflects back to the photodiode.



IR Sensor Working Principle

IR Receiver LED and IR Transmitter LED

An IR receiver LED and an IR transmitter LED are both types of light-emitting diodes (LEDs) that are used in infrared (IR) communication.

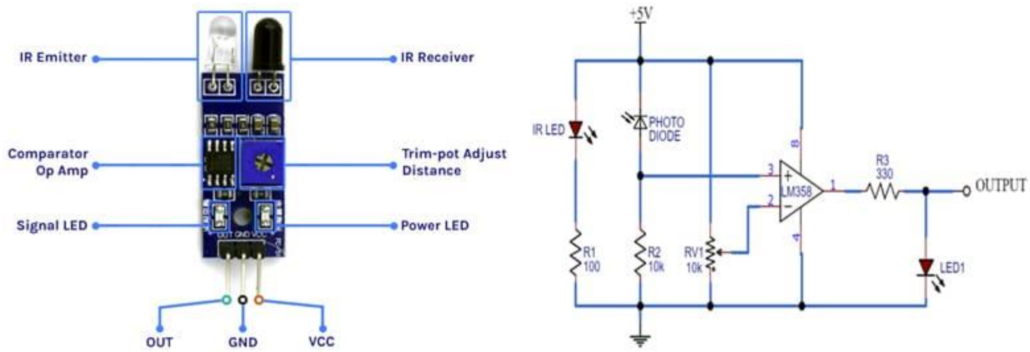
An IR receiver LED is a device that detects infrared signals from remote controls and other IR sources. It is typically a small, clear, or translucent device that is sensitive to IR light in a specific frequency range.

When an IR signal is detected, the IR receiver LED will emit a small amount of visible light, which can be used to confirm that a signal has been received.

An IR transmitter LED, on the other hand, is a device that emits infrared light in order to send signals to other devices. It is typically a small, clear, or translucent device that emits IR light in a specific frequency range. IR transmitter LEDs are commonly used in remote controls and other IR signaling devices.

In summary, the IR receiver LED detects the IR signal from the remote, and the IR transmitter LED emits the IR signal from the remote.

IR Sensor Module



IR Sensor Module with Circuit

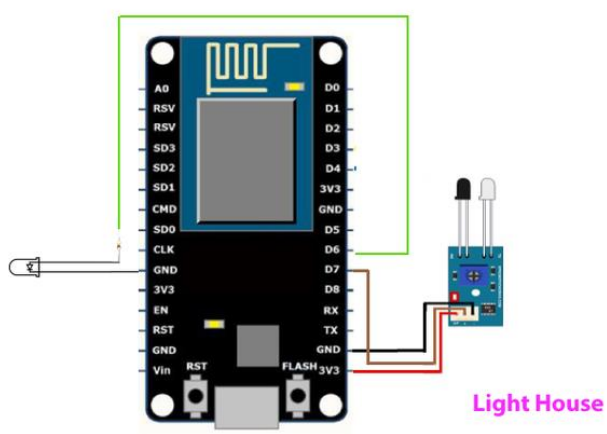
An IR sensor module is a device that contains an IR receiver LED and other components that are used to detect and process IR signals. It typically includes an IR receiver LED, a signal amplifier, and a demodulator circuit.

The IR receiver LED is used to detect IR signals, while the signal amplifier and demodulator circuit are used to amplify and process the received signal, respectively.

IR sensor modules are widely used in various electronic applications such as remote control, motion detection, proximity sensing, and more. They are commonly used in consumer electronics, robotics, and automation systems.

IR sensor modules come in various forms such as simple IR receiver modules and complex IR sensor modules with additional features such as signal processing and signal filtering. Some IR sensor modules also provide an output in a digital format that can be read by a microcontroller or microprocessor.

IR Sensor Interfacing with NodeMCU



Power is usually **3-5v** DC input. Every IR modules have a **3-pin** connection.

The **circuit connections** are made as follows:

Vcc pin of the IR module is connected to **+3v** of the NodeMCU.

Output pin of the IR module is connected to Digital pin **D7** of the NodeMCU.

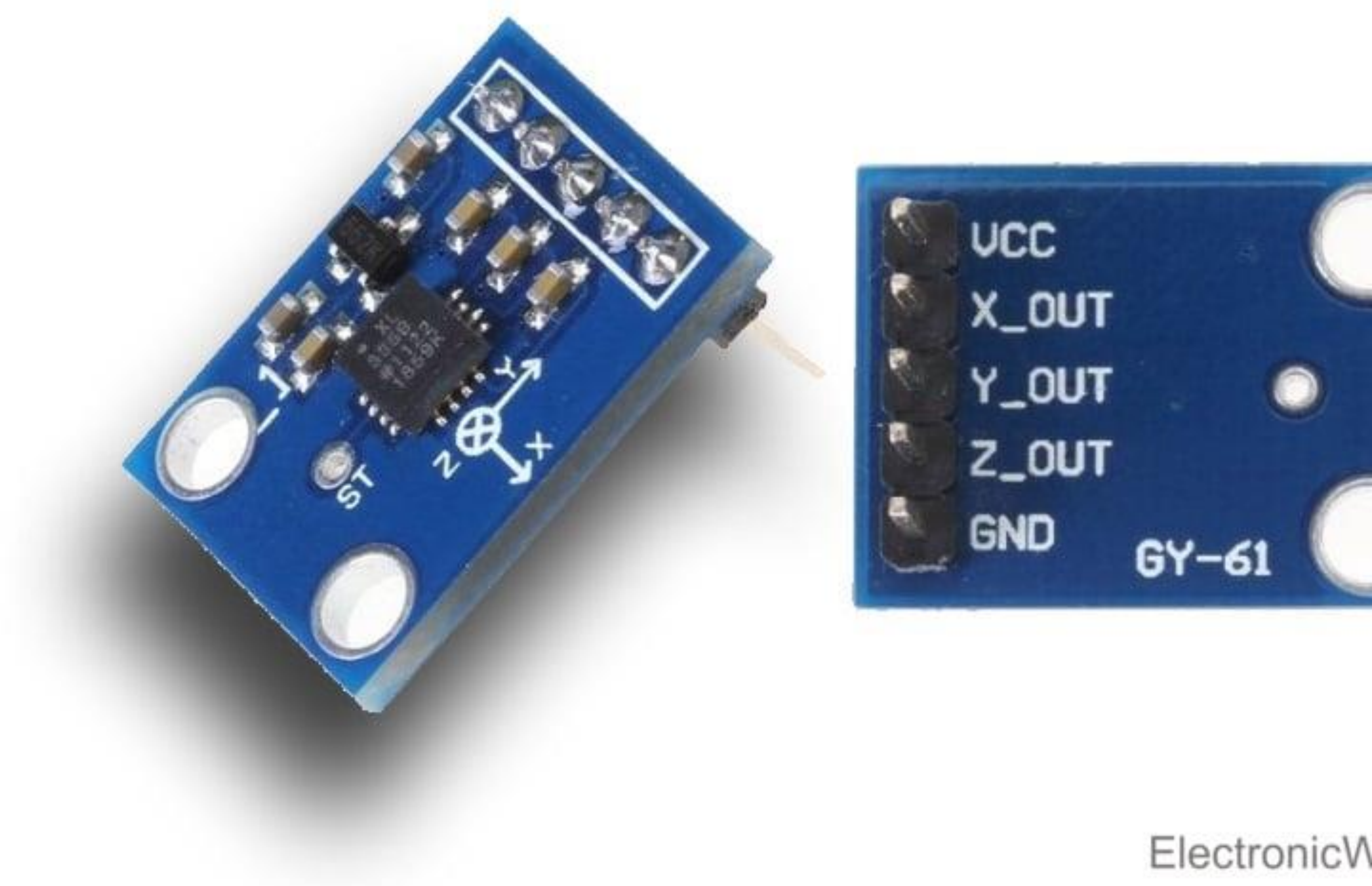
GND pin of the IR module is connected to Ground pin (**GND**) of the NodeMCU.
Anode pin of the LED is connected to Digital pin **D6** and **Cathode** pin is connected to **GND** pin of NodeMCU.

Code:

```
int ledPin = 12; // choose pin for the LED
int inputPin = 13; // choose input pin (for Infrared sensor)
int val = 0; // variable for reading the pin status
void setup()
{
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare Infrared sensor as input
}
void loop()
{
  val = digitalRead(inputPin); // read input value
  if (val == HIGH)
  { // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  }
  else
  {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

ADXL335 Acclerometer

An accelerometer is an electromechanical device that will measure acceleration force. It shows acceleration, only due to cause of gravity i.e. g force. It measures acceleration in g unit.



ADLX335 Accelerometer

On the earth, 1g means acceleration of 9.8 m/s2 is present. On moon, it is 1/6th of earth and on mars it is 1/3rd of earth.
Accelerometer can be used for tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

Specification of ADXL335

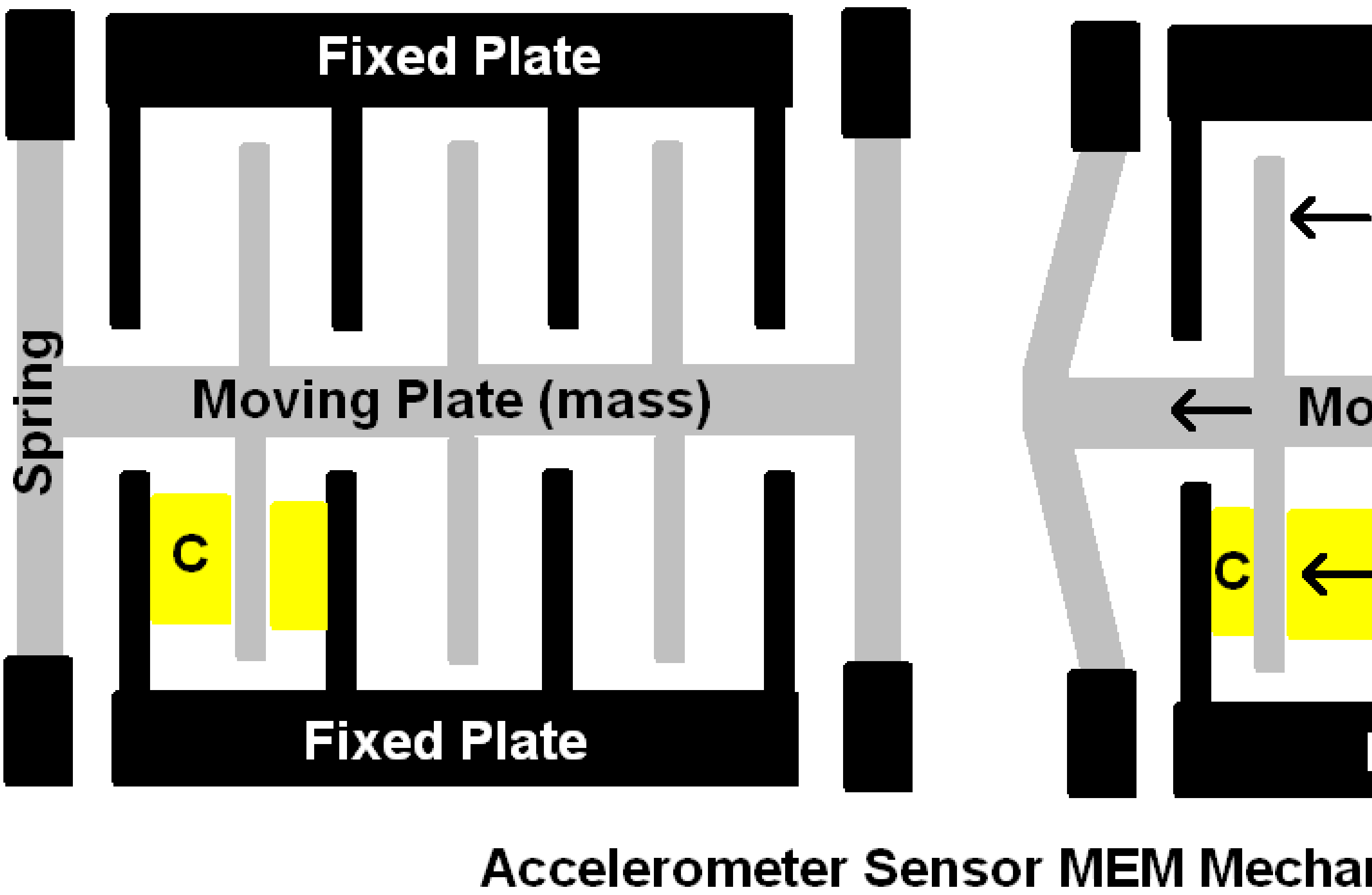
1. Supply Voltage: 2.8V to 3.6V
2. Current Consumption: 320uA
3. Sensitivity: 300mV/g
4. Bandwidth: 3Hz to 5kHz
5. Dynamic Range: ±3g
6. Operating Temperature: -40°C to +85°C
7. Package Type: Surface Mount Plastic Package (LFCSP)
8. Pin Configuration: 5 Pin, 1.27mm Pitch
9. Output Type: Voltage Output
10. Interface: SPI/I2C

11. Output Range: 0V to Vcc
12. Storage Temperature: -65°C to +150°C

ADXL335 module

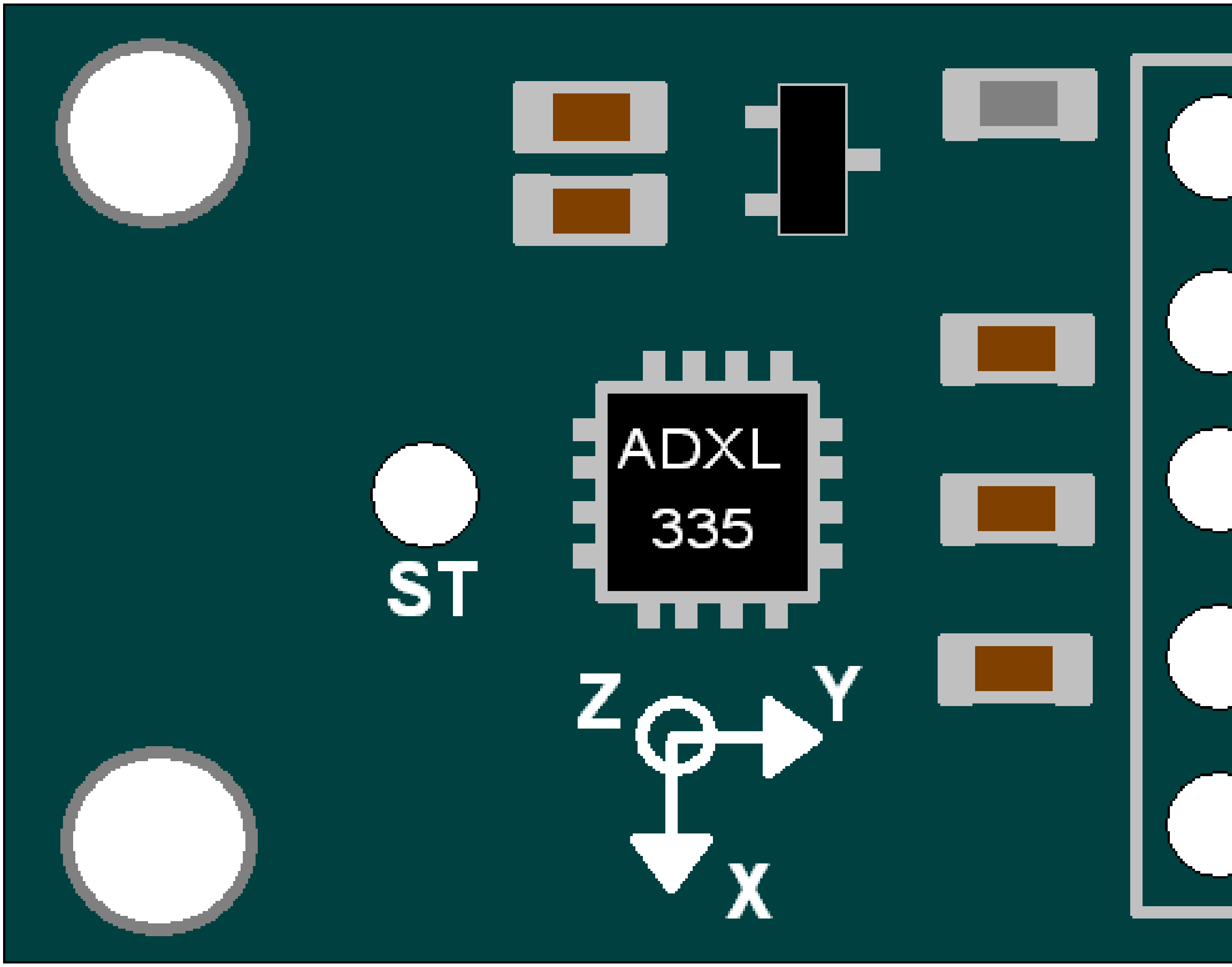
- The ADXL335 gives complete 3-axis acceleration measurement.
- This module measures acceleration within range ±3 g in the x, y and z axis.
- The output signals of this module are analog voltages that are proportional to the acceleration.
- It contains a polysilicon surface-micro machined sensor and signal conditioning circuitry.

How Accelerometer Works



- As we can see from the above figure, basic structure of accelerometer consists fixed plates and moving plates (mass).
- Acceleration deflects the moving mass and unbalances the differential capacitor which results in a sensor output voltage amplitude which is proportional to the acceleration.
- Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

ADXL335 Accelerometer Module Pinout



ADXL335 Sensor Module Pinout

ADXL335 Accelerometer Module Pin Description

VCC: Power supply pin i.e. connect 5V here.
X_OUT: X axis analog output.
Y_OUT: Y axis analog output.
Z_OUT: Z axis analog output.
GND: Ground pin i.e. connect ground here.
ADXL335 accelerometer provides analog voltage at the output X, Y, Z pins; which is proportional to the acceleration in respective directions i.e. X, Y, Z.

Angles using ADXL335

We can calculate the angle of inclination or tilt by using X, Y, and Z values. Also, we can calculate Roll, Pitch, and Yaw angles with respect to X, Y, and Z axis. So first we need to convert 10-bit ADC values into g unit.
As per ADXL335 datasheet maximum voltage level at 0g is 1.65V and sensitivity scale factor of 330mV/g.

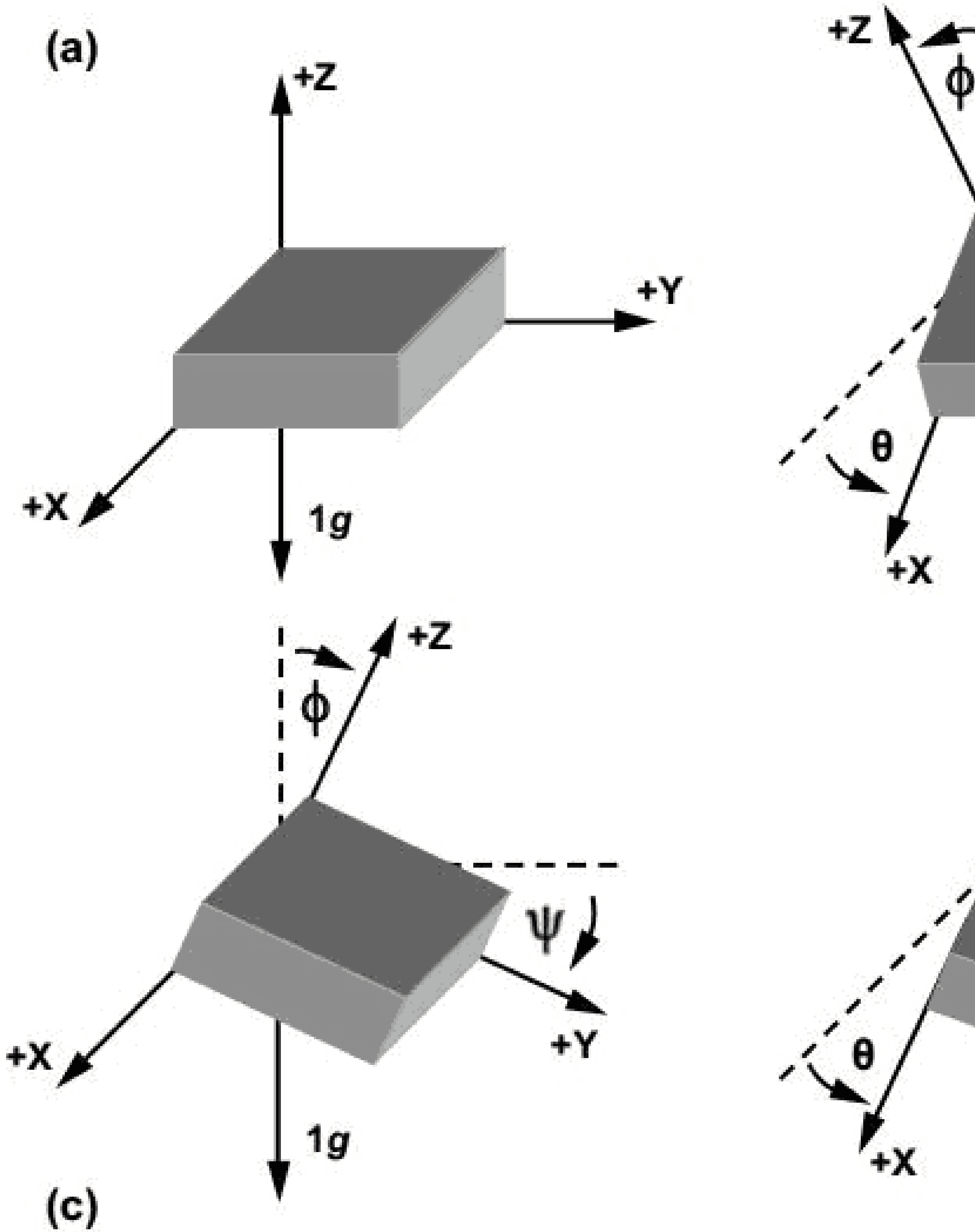
Above formula gives us acceleration values in g unit for X, Y, and Z axis as,

```
Axout = (((X axis ADC value * Vref) / 1024) – 1.65) / 0.330
Ayout = (((Y axis ADC value * Vref) / 1024) – 1.65) / 0.330
Azout = (((Z axis ADC value * Vref) / 1024) – 1.65) / 0.330
```

Note that, practically we get slightly different voltage at 0g. So, put the practical value of the voltage at 0g.

Angle of Inclination

- Angle of inclination means how much angle the device is tilted from its plane of surface.
- Angle of inclination are shown below figure.
- To calculate angle of inclination of X, Y, and Z axis from its reference, we need to use the below formulas.


(a)
+Z
+Y
+X
1g
+Z
ϕ
θ
+X
+Z
ϕ
ψ
+X
+Y
1g
(c)
θ
+X

Angle of inclination can be calculated as,

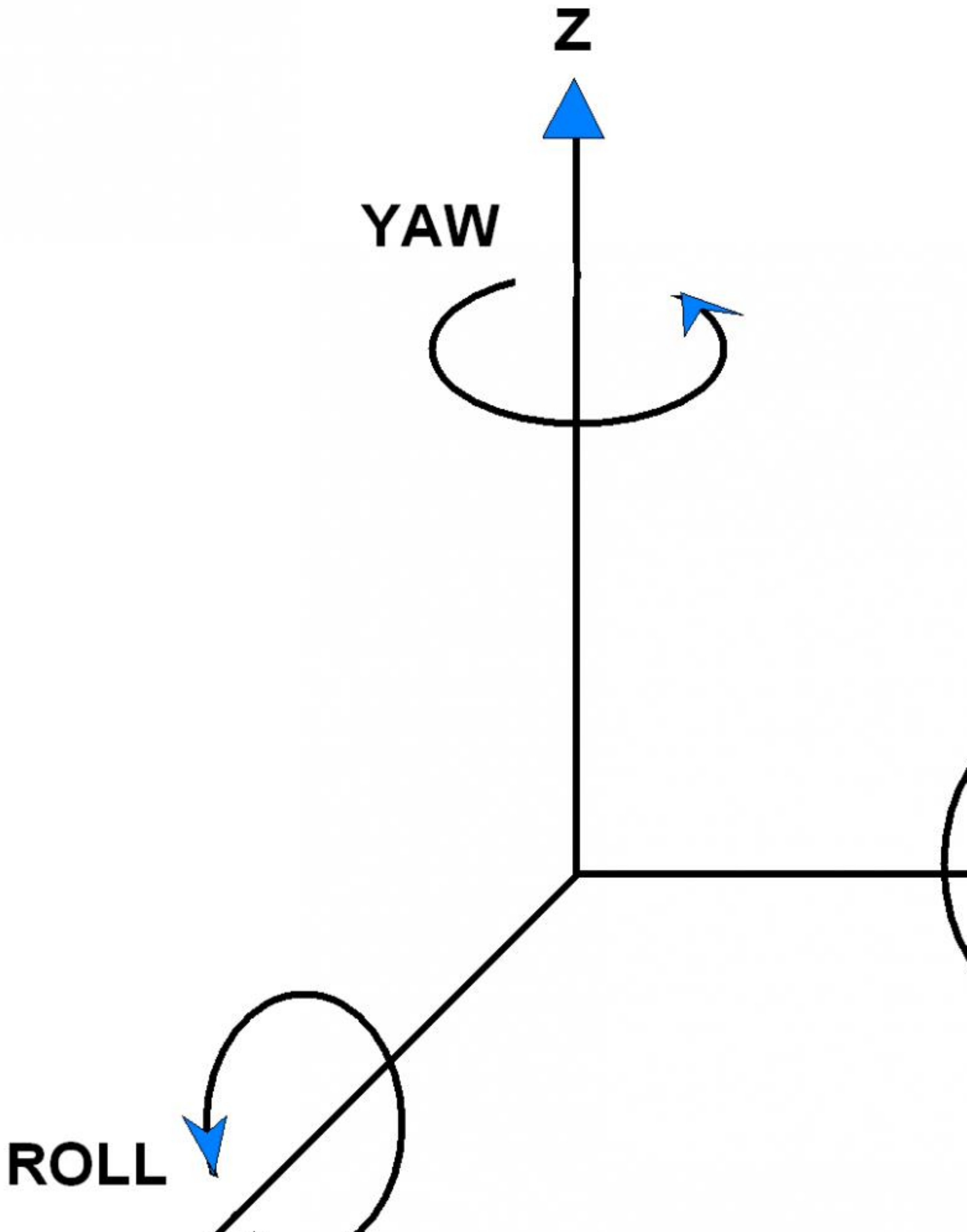
We get these angles in radians. So, multiply these values by (180/π) to get angle in degrees within range of -90° to +90° each axis.

Angle of Rotation

Now let’s find a complete angle of rotation (0° to 360°) around X, Y, Z axis, which we can also call as,

- Roll - Angle of rotation along the X axis
- Pitch - Angle of rotation along the Y axis
- Yaw - Angle of rotation along the Z axis

All of them are shown in below conceptual diagram.


Z
YAW
ROLL

These angles are in degrees and can give readings of a complete rotation.
Now let’s calculate these angles. As we get Θ, Ψ and Φ in the range of -90° to +90°. Here we need to make these values in the range of -180° to +180° so that we can calculate a complete 360° angle of rotation. Let's calculate these with the arc tangent function which can be expressed as

$$\mathrm{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < \\ +\frac{\pi}{2} & \text{if } x = \\ -\frac{\pi}{2} & \text{if } x = \\ \text{undefined} & \text{if } x = \end{cases}$$

This function will produce the result in the range of -π to π. These values in radians we can convert into degree by multiplying it with (180/π ≈ 57.29577951) factor. So here we get values in -180° to +180°, and we can convert it to complete 0° to 360° by just adding 180° to range.
Hence, we get roll, pitch and yaw angles as,

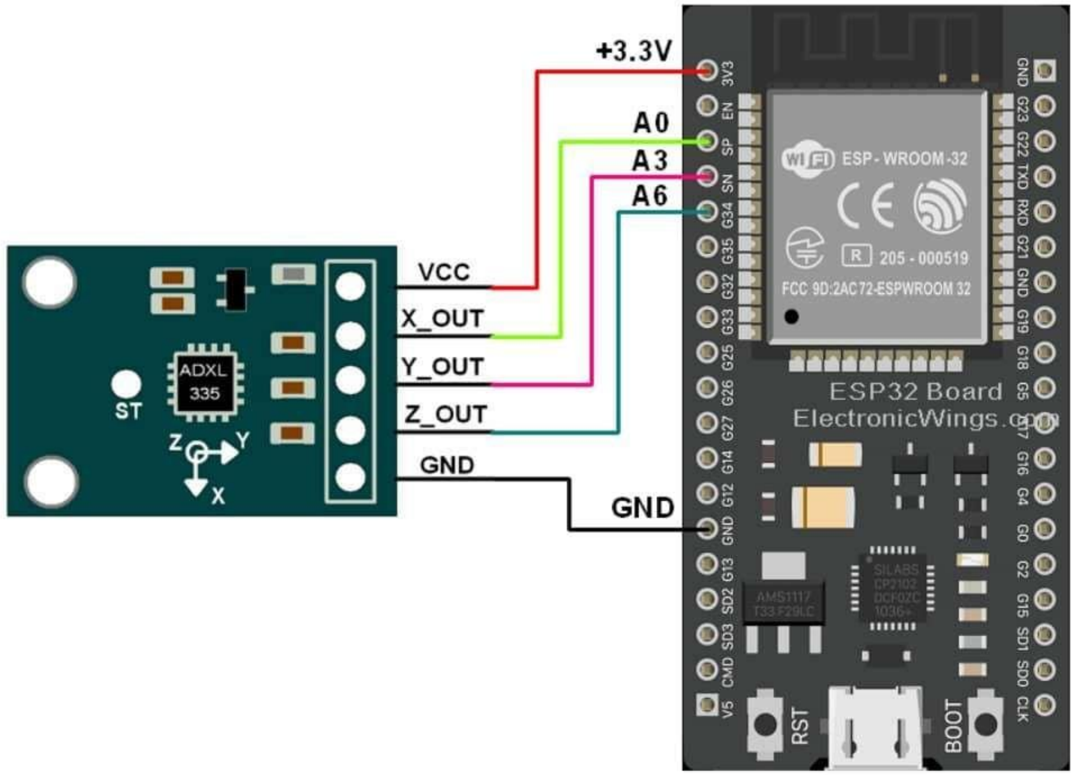
Roll = (atan2(Ayout, Azout)) *57.29577951+180

Pitch = (atan2(Azout, Axout)) *57.29577951+180

Yaw = (atan2(Axout, Ayout)) *57.29577951+180

Note that, rotation along X (roll) and Y (pitch) axis will produce change in acceleration but rotation along with Z axis (yaw) will not produce any change in acceleration as it is perpendicular to the plane of surface. Hence using only accelerometer, yaw cannot be calculated.

ADXL335 Accelerometer Hardware Connection with the ESP32



ESP32

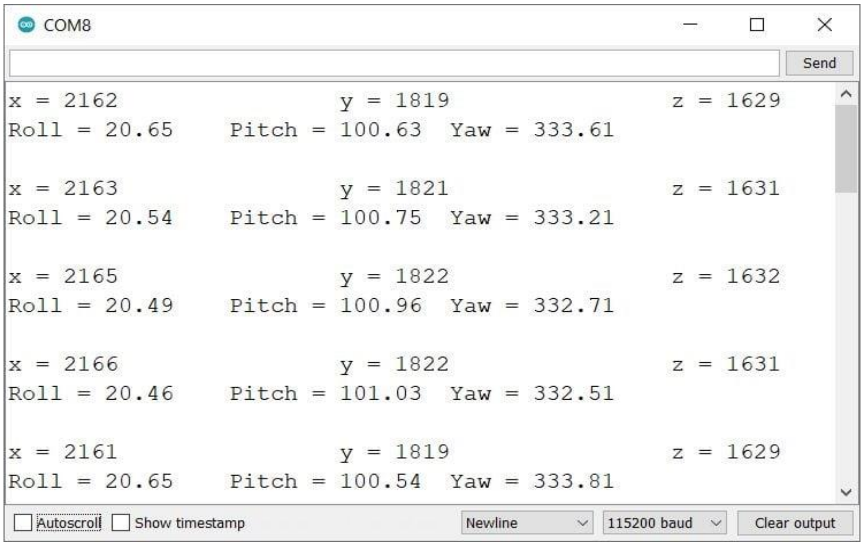
Interfacing with ADXL335 Accelerometer

Code for Read ADXL335 Accelerometer Serially using NodeMCU

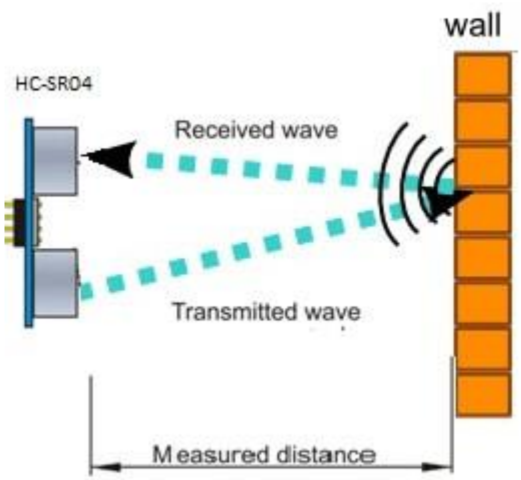
```
#include <math.h>

double roll, pitch, yaw;

const int x_out = A0; /* connect x_out of module to A1 of UNO board */
const int y_out = A3; /* connect y_out of module to A2 of UNO board */
const int z_out = A6; /* connect z_out of module to A3 of UNO board */

void setup(void){
  Serial.begin(115200);           /*Set the baudrate to 115200*/
}

void loop() {
  int x_adc_value, y_adc_value, z_adc_value;
  double x_g_value, y_g_value, z_g_value;

  x_adc_value = analogRead(x_out); /* Digital value of voltage on x_out pin */
  y_adc_value = analogRead(y_out); /* Digital value of voltage on y_out pin */
  z_adc_value = analogRead(z_out); /* Digital value of voltage on z_out pin */
  Serial.print("x = ");
  Serial.print(x_adc_value);
  Serial.print("\t\t");
```

```
  Serial.print("y = ");
  Serial.print(y_adc_value);
  Serial.print("\t\t");
  Serial.print("z = ");
  Serial.print(z_adc_value);
  Serial.println("");

  x_g_value = ( ( ( (double)(x_adc_value * 3.3)/4096) - 1.65 ) / 0.330 ); /* Acceleration in x-direction in g units */
  y_g_value = ( ( ( (double)(y_adc_value * 3.3)/4096) - 1.65 ) / 0.330 ); /* Acceleration in y-direction in g units */
  z_g_value = ( ( ( (double)(z_adc_value * 3.3)/4096) - 1.80 ) / 0.330 ); /* Acceleration in z-direction in g units */

  roll = ( ( (atan2(y_g_value,z_g_value) * 180) / 3.14 ) + 180 ); /* Formula for roll */
  pitch = ( ( (atan2(z_g_value,x_g_value) * 180) / 3.14 ) + 180 ); /* Formula for pitch */
  yaw = ( ( (atan2(x_g_value,y_g_value) * 180) / 3.14 ) + 180 ); /* Formula for yaw */
/* Display the results (acceleration is measured in m/s^2) */
  Serial.print("Roll = ");
  Serial.print(roll);
  Serial.print("\t");
  Serial.print("Pitch = ");
  Serial.print(pitch);
  Serial.print("\t");
  Serial.print("Yaw = ");
  Serial.print(yaw);
  Serial.print("\n\n");

  /*wait a second*/
  delay(1000);
}
```

ADXL335 Output on the serial monitor



Ultrasonic Sensor Working Principle



Ultrasonic Sensor HC-SR04

Hcsr04 ultrasonic sensor is composed of ultrasonic transmitter, ultrasonic receiver and a control circuit. Hscr04 ultrasonic transmitter transmits ultrasound waves at 40,000 Hz. Transmitted waves bounce back if they hit any flat surface/object in their path. Bounced back waves reaches the ultrasonic receiver. Ultrasonic receiver receives the bounced back waves and notifies the control circuit about it. Control circuit than calculates the time taken by waves to reach back after transmission. Time is than manipulated to approximate the distance traveled by waves or what is the distance between the sensor and the object? from which ultrasound waves bounced back.
Hcsr04 can measure distance between an active range of 2 cm to 4 meters. Hcsr04 requires 5 volts and 15 mA of power for operation. Hcsr04 has four pins. Two are power pins. Vcc is +ve pin apply 5v to this pin and Gnd is ground pin connect -ve of 5v power source with it. The other two pins are **Trigger** and **Echo**.

- **Trigger pin** is triggered by external controller to out burst an ultrasound wave.
- **Echo pin** notifies external controller when receiver receives back the bounced back wave.

Trigger and echo pins of hcsr04 ultrasonic sensor is directly connected to GPIO-2 and GPIO-0 or D4 and D3 pins of nodemcu. You people might have a question here why pins are directly connected if voltage levels of both devices are different. The answer is simple, trigger and echo pins output can easily be read by 3.3 volt devices. It worked for me with out any problem and the diy project is running continuously from 72 hours. If it did not work for you then you might insert a logic converter between the two modules.
Ultrasonic sensor HC-SR04 with nodemcu esp8266 WiFi module

Time is converted in to distance using the speed of sound in air formula. According to universal speed of sound in air formula

Time = Distance / Speed

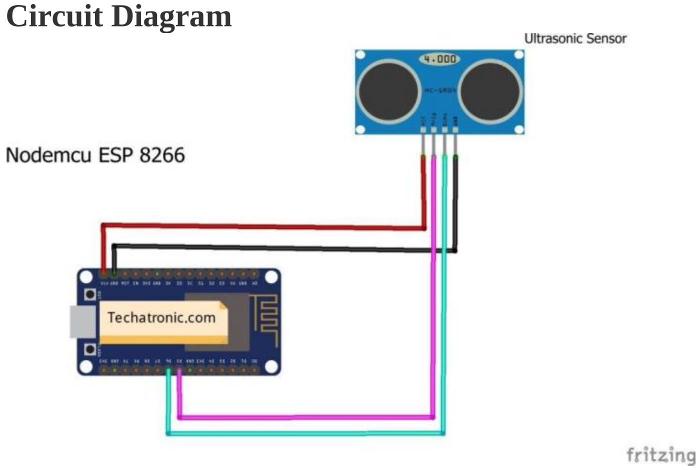
Where **Speed = speed of sound in air. Which is 340 m/s**

Hence reorganizing the formula **Distance= Time * 340(speed of sound in air).**

The above formula is used in the below code to measure distance from time. At the beginning of the code i defined the nodemcu pins (**trigP,echoP**) interacting with the hcsr04 ultrasonic sensor. Onward two variables(**distance,duration**) are defined for storing the time and distance values. In the setup function trigger pin is declared output and echo pin is declared input. Serial communication of nodemcu is also initialized in setup function at 9600 baud rate.
In the loop function trigger pin remain low for 2 seconds after 2 seconds it is made high for 10 us. 10us is time in which the trigger pin sends an output ultrasound signal in 8 cycles. Then the pulse in function reads the bounce back waves and approximates the time. After wards the statement **distance= duration*0.034/2** calculates the real distance between the ultrasonic sensor and object.

Ultrasonic sensor interfacing with ESP 8266 board

Circuit Diagram



Connection Table

Nodemcu esp8266	Ultrasonic Sensor
VV, Vin	VCC
G, GND	GND
D5 Pin	Trig Pin
D6 Pin	Echo

- The HC-SR04 ultrasonic sensor has four pins out of which two are used to provide power to the sensor and the other two are for the data.
- Connect the VCC pin of the ultrasonic sensor to the VIN pin of the nodemcu.
- Join the GND pin of the ultrasonic sensor with the GND pin of the nodemcu.
- Connect the TRIG and ECHO pins of the ultrasonic sensor with the digital-5 and digital-6 pins of the nodemcu as shown above in the diagram.
- Now connect the USB cable to the nodemcu so that the ultrasonic sensor starts working and you can see the values on the serial monitor screen.

Code for ultrasonic sensor interfacing

```
const int trigPin = D5;
 const int echoPin = D6;
 long duration;
 int distance;
 void setup() {
 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
 Serial.begin(9600); // Starts the serial communication
 }
```

```
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
delay(2000);
}
```