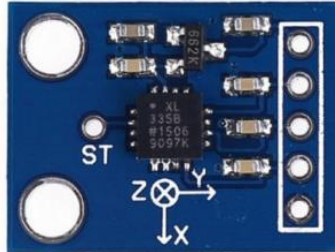


# GY-61 ADXL335 Triple Axis Accelerometer



## ADXL335 Module Triple Axis Accelerometer Breakout

The 3 axis ADXL335 from Analog Devices is the latest in a long, proven line of analog sensors - the holy grail of accelerometers. The ADXL335 is a triple axis MEMS accelerometer with extremely low noise and power consumption - only 320uA! The sensor has a full sensing range of +/-3g. The Board comes fully assembled and tested with external components installed.

### Quick Spec:

Sensor Chip: ADXL335

Power Supply: 1.8V- 6V (Onboard 3.3V Regulator)

Integrated X, Y, and Z axis accelerometer on a single chip

X and Y axis has a 0.5Hz to 1600Hz bandwidth

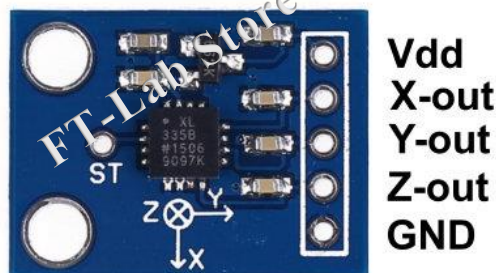
Z axis has a 0.5Hz to 550Hz bandwidth

Hermetically sealed for temp and humidity resistance

10,000 g shock tolerant

Dimensions: 21 x 16 mm

### Board Layout:





## Sample Code

```
//Analog read pins
const int xPin = 0;
const int yPin = 1;
const int zPin = 2;

//The minimum and maximum values that came from
//the accelerometer while standing still
//You very well may need to change these
int minVal = 265;
int maxVal = 402;

//to hold the caculated values
double x;
double y;
double z;

void setup(){
  Serial.begin(9600);
}

void loop(){

  //read the analog values from the accelerometer
  int xRead = analogRead(xPin);
  int yRead = analogRead(yPin);
  int zRead = analogRead(zPin);

  //convert read values to degrees -90 to 90 - Needed for atan2
  int xAng = map(xRead, minVal, maxVal, -90, 90);
  int yAng = map(yRead, minVal, maxVal, -90, 90);
  int zAng = map(zRead, minVal, maxVal, -90, 90);
```

```
//Calculate 360deg values like so: atan2(-yAng, -zAng)
//atan2 outputs the value of  $-\pi$  to  $\pi$  (radians)
//We are then converting the radians to degrees
x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI);
y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);
z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);

//Output the calculations
Serial.print("x: ");
Serial.print(x);
Serial.print(" | y: ");
Serial.print(y);
Serial.print(" | z: ");
Serial.println(z);

delay(100); //just here to slow down the serial output - Easier to read
}
```