Lesson 18 ADXL335

Introduction

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal-conditioned voltage outputs. It measures acceleration with a minimum full-scale range of ±3g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

Components

- 1 * SunFounder Uno board
- 1 * Breadboard
- 1 * ADXL335 module
- 1 * USB cable
- Jumper wires

Principle

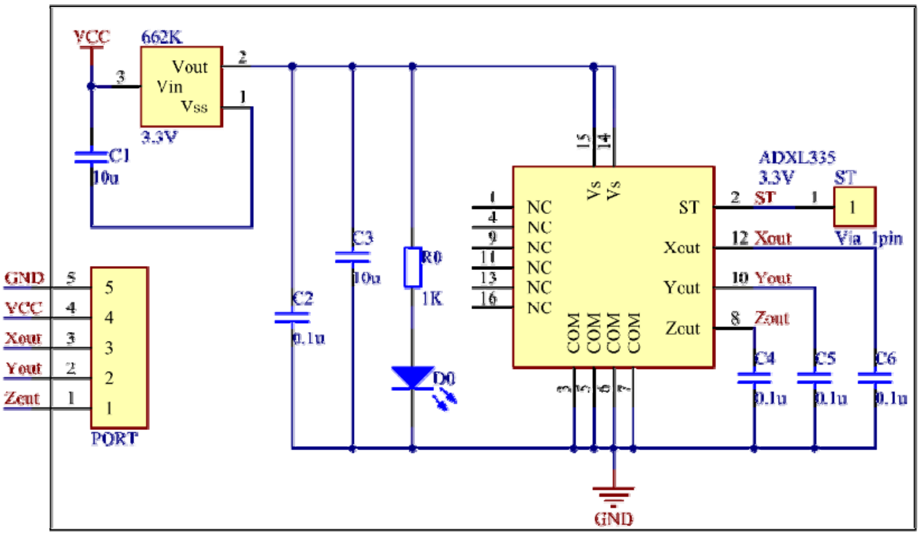
Accelerometer

An accelerometer is used to measure the force generated during the acceleration. The most fundamental is the commonly-known acceleration of gravity which is 1g.

By measuring the acceleration caused by gravity, you can calculate the tilt angle of the device to the level surface. Through analyzing the dynamic acceleration, you can tell the way how the device is moving. For example, self-balancing board or hoverboard applies the acceleration sensor and gyroscope for Kalman filter and posture correction.

ADXL335 is an accelerometer module that uses a 5*5*2 mm LCC packaging when the ambient temperature ranges from -55℃ to 125℃. The operating voltage of ADXL335 ranges from 1.8V to 3.6V. It's quite convenient for embedding hardware in engineering projects.

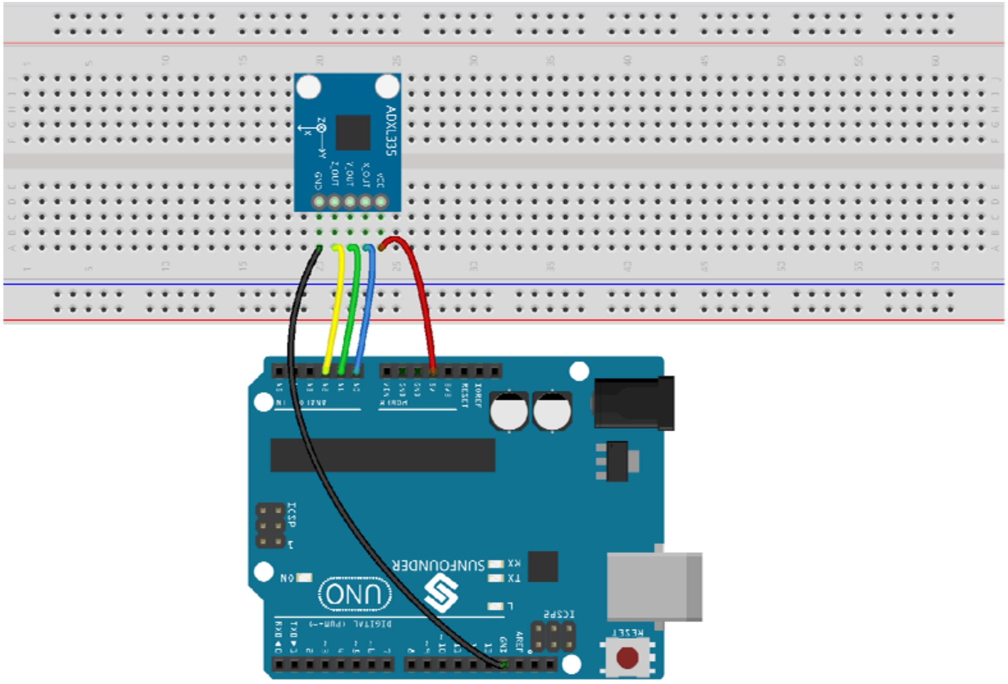
See the following figure for the schematic diagram. There is a 3.3v voltage regulator chip in the circuit, so you can power the module with 5V or 3.3V.



Principle: The ADXL335 outputs analog voltage values. Therefore, connect its pin X, Y, and Z to A0, A1, and A2 of the control board. Read the analog values of X, Y, and Z by programming, convert them into digital ones via the AD converter in the board. With some calculation, you can get the acceleration at X, Y, and Z axes when moving the ADXL335.

Experimental Procedures

Step 1: Build the circuit

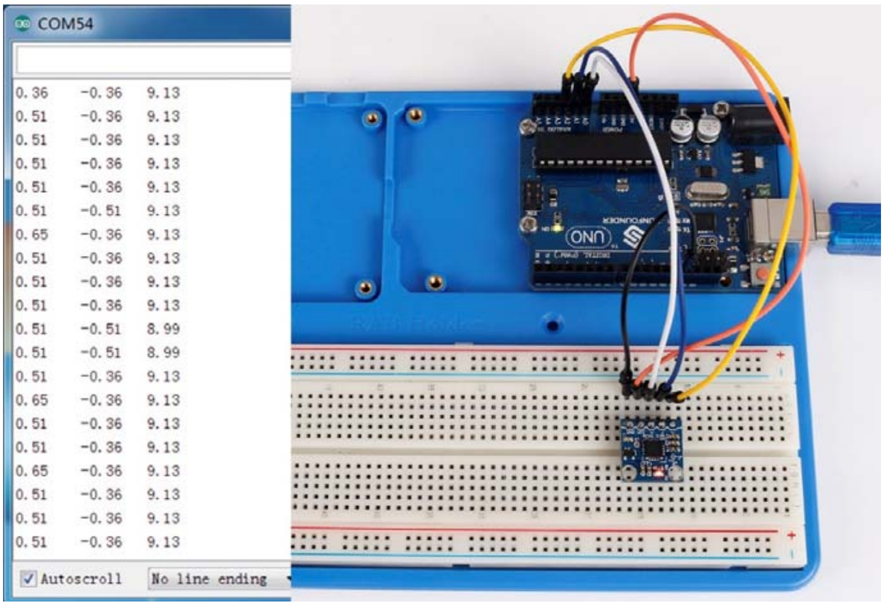


Step 2: Open the code file

Step 3: Select correct Board and Port

Step 4: Upload the sketch to the SunFounder Uno board

After uploading, open **Serial Monitor**, where you can see the data detected. When the acceleration of the module changes, the figure will change accordingly on the window.



Code

```
//ADXL335
/********************************
ADXL335
note:vcc-->5v ,but ADXL335 Vs is 3.3V
The circuit:
      5V: VCC
analog 0: x-axis
analog 1: y-axis
analog 2: z-axis
After burning the program, open the serial monitor debugging window, where you can see the data detected being displayed. When the acceleration varies, the figure will vary accordingly.
********************************/
//Email:support@sunfounder.com
//Website:www.sunfounder.com
//2015.5.7
const int xpin = A0;                  // x-axis of the accelerometer
const int ypin = A1;                  // y-axis
const int zpin = A2;                  // z-axis (only on 3-axis models)
void setup()
{
 // initialize the serial communications:
 Serial.begin(9600);
}
void loop()
{
 int x = analogRead(xpin);  //read from xpin
 delay(1); //
 int y = analogRead(ypin);  //read from ypin
 delay(1);
 int z = analogRead(zpin);  //read from zpin
```

```
 float zero_G = 338.0; //ADXL335 power supply by Vs 3.3V:3.3V/5V*1024=676/2=338
//Serial.print(x);
//Serial.print("\t");
//Serial.print(y);
//Serial.print("\t");
//Serial.print(z);
//Serial.print("\n");
float zero_Gx=331.5;//the zero_G output of x axis:(x_max + x_min)/2
float zero_Gy=329.5;//the zero_G outgput of y axis:(y_max + y_min)/2
float zero_Gz=340.0;//the zero_G output of z axis:(z_max + z_min)/2
float scale = 67.6;//power supply by Vs 3.3V:3.3v /5v *1024/3.3v *330mv/g =67.6g
float scale_x = 65;//the scale of x axis: x_max/3.3v*330mv/g
float scale_y = 68.5;//the scale of y axis: y_max/3.3v*330mv/g
float scale_z = 68;//the scale of z axis: z_max/3.3v*330mv/g
Serial.print(((float)x - zero_Gx)/scale_x);    //print x value on serial monitor
Serial.print("\t");
Serial.print(((float)y - zero_Gy)/scale_y);    //print y value on serial monitor
Serial.print("\t");
Serial.print(((float)z - zero_Gz)/scale_z);    //print z value on serial monitor
Serial.print("\n");
delay(1000);    //wait for 1 second
}
```

Code Analysis 18-1 **Calculate the acceleration of ADXL335**

```
void loop()
{
// Read the values x, y, and z from the pin x, y, and z.
 int x = analogRead(xpin);   //read from xpin
 delay(1); //
 int y = analogRead(ypin);   //read from ypin
 delay(1);
 int z = analogRead(zpin);   //read from zpin
```

//The *zero_G* output is nominally equal to Vs/2 at all supply voltages. When the power is 3.3V, convert it to a digital value, so it should be about 338 (3.3V/5V*1024=676, 676/2=338).

```
float zero_G = 338.0;  //ADXL335 power supply by Vs 3.3V:3.3V/5V*1024=676/2=338
```

To get a more accurate *zero_G* value of the three axes, you can also measure the three zero reference points separately. Uncomment the following part. This code is to print the analog value of the axes. Read the maximum value at three axes for the ADXL335 (place the module on a level surface according to the x, y, and z directions) and the minimum one. Calculate the average of the two at each axis, and assign them to *zero_Gx*, *zero_Gy*, and *zero_Gz* respectively. So you can get the zero reference point.

```
//Serial.print(x);
//Serial.print("\t");
//Serial.print(y);
//Serial.print("\t");
//Serial.print(z);
//Serial.print("\n");
float zero_Gx=331.5;//the zero_G output of x axis:(x_max + x_min)/2
float zero_Gy=329.5;//the zero_G outgput of y axis:(y_max + y_min)/2
float zero_Gz=340.0;//the zero_G output of z axis:(z_max + z_min)/2
```

// The output sensitivity (or scale factor) varies proportionally to the supply voltage. The sensitivity of the ADXL335 is 330mV/g, so we can get: scale = (3.3V/5V x 1024)/3.3V x 330mV/g = 67.6 g. For higher accuracy, you can measure and calculate the value separately: scale_x, scale_y, and scale_z.

```
float scale = 67.6;//power supply by Vs 3.3V:3.3v /5v *1024/3.3v *330mv/g =67.6g
float scale_x = 65;//the scale of x axis: x_max/3.3v*330mv/g
float scale_y = 68.5;//the scale of y axis: y_max/3.3v*330mv/g
float scale_z = 68;//the scale of z axis: z_max/3.3v*330mv/g
```

//See the calculation formula of acceleration below. Print it on Serial Monitor.

```
Serial.print(((float)x - zero_Gx)/scale_x);  //print x value on serial monitor
Serial.print("\t");
Serial.print(((float)y - zero_Gy)/scale_y);  //print y value on serial monitor
Serial.print("\t");
Serial.print(((float)z - zero_Gz)/scale_z);  //print z value on serial monitor
Serial.print("\n");
delay(1000);  //wait for 1 second
```