**Simple angle meter using ADXL335 accelerometer**

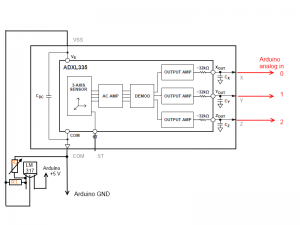
[ADXL335](http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf) is 3 axis accelerometer with analog output from Analog Devices.

ADXL335 acceleration measurement range is +/- 3 g. Supply voltage is 1.8 -  3.6 V, however all specifications at the datasheet is given at 3.0 V. This accelerometer has  3 outputs for X,Y,Z axis which voltage is proportional to acceleration on specific axis.

At midpoint when acceleration is 0 g output is typically 1/2 of supply voltage. If a supply voltage is 3V, then output is 1.5 V. Output sensitivity typically is 300 mV/g.

The connecting ADXL335 to Arduino board is simple. The accelerometer can be supplied from the 3.3 V output at Arduino board, however then midpoint voltage and sensitivity is different from specified at datasheet. I used 3.0 V supply voltage, it came from voltage[regulator LM317](http://www.national.com/ds/LM/LM117.pdf).

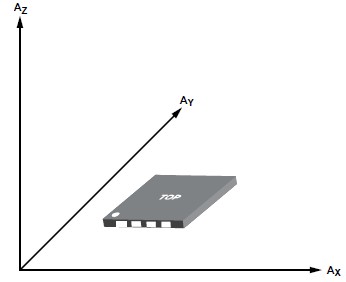
Outputs of X,Y,Z axis are connected directly to analog inputs of Arduino.

[](http://www.electronicsblog.net/wp-content/uploads/acs.png)

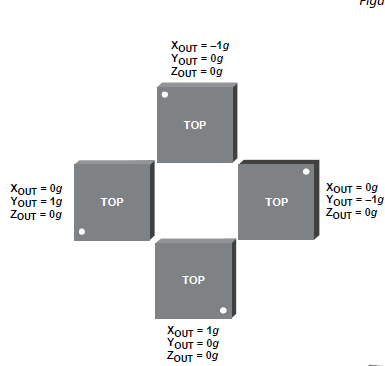
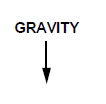
R1 is 2.7k resistor, while R2 is variable 10k resistor. Please adjust R2 to have  3.0 V  output at LM317 before connecting ADXL335 board. Connect the board and adjust again, because output voltage will drop since load is connected. If  readings of the acceleration are unstable please connect additional capacitor between VSS and COM.

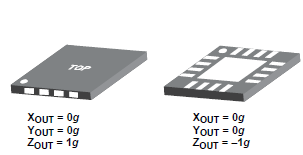
An accelerometer can be used not only measure acceleration, but also orientation. It’s because accelerometer sense the force of gravity which is pointed to the center of earth.

This is how sensing axis are orientated to ADXL335 (images from [ADXL335 datasheet](http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf))



Readings from X,Y,Z axis reflects an orientation of an accelerometer.





Interesting that if a direction of sensing axis match to gravity direction measurement result is negative (-1 g).

Let’s start with an Arduino program. This program not only sends readings of acceleration from 3 axis via serial port, but also calculates rotation of the accelerometer around X, Y, Z, axis.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #define ADC\_ref 2.56    #define zero\_x 1.569    #define zero\_y 1.569    #define zero\_z 1.569    #define sensitivity\_x 0.3    #define sensitivity\_y 0.3    #define sensitivity\_z 0.3 |

ADC reference voltage is 2.56 V. Default “on board 5V” is not selected because is slightly lower then 5.0 V and can be unstable. 0 g voltage after some simple calibration is set to be 1.569 V. The sensitivity for calculations is used as default – 300 mV/g.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | void setup()   {    analogReference(INTERNAL2V56);    Serial.begin(256000);    } |

Setting the ADC reference to 2.56 V and starting serial communication.

void loop() {

|  |  |
| --- | --- |
| 1  2  3 | value\_x = analogRead(0);  value\_y = analogRead(1);  value\_z = analogRead(2); |

At the loop A/D conversions are performed for each of 3 channels.

|  |  |
| --- | --- |
| 1 | xv=(value\_x/1024.0\*ADC\_ref-zero\_x)/sensitivity\_x; |

The acceleration is calculated at this line.

|  |  |
| --- | --- |
| 1 | angle\_z =atan2(-yv,-xv)\*57.2957795+180; |

The rotation(for Z axis) is calculated using [atan2](http://en.wikipedia.org/wiki/Atan2) function. It calculates angle from length of X, Y vectors. \*57.2957795 – is conversation of radian to degree. +180 is for offset.

Full code:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 | // Simple angle meter using ADXL335 accelerometer  //from electronicsblog.net/    #define ADC\_ref 2.56    #define zero\_x 1.569    #define zero\_y 1.569    #define zero\_z 1.569    #define sensitivity\_x 0.3    #define sensitivity\_y 0.3    #define sensitivity\_z 0.3    unsigned int value\_x;  unsigned int value\_y;  unsigned int value\_z;    float xv;  float yv;  float zv;    float angle\_x;  float angle\_y;  float angle\_z;    void setup()   {    analogReference(INTERNAL2V56);    Serial.begin(256000);    }    void loop() {    value\_x = analogRead(0);  value\_y = analogRead(1);  value\_z = analogRead(2);    xv=(value\_x/1024.0\*ADC\_ref-zero\_x)/sensitivity\_x;    Serial.print ("x= ");  Serial.print (xv);  Serial.print(" g ");    yv=(value\_y/1024.0\*ADC\_ref-zero\_y)/sensitivity\_y;    Serial.print ("y= ");  Serial.print (yv);  Serial.print(" g ");    zv=(value\_z/1024.0\*ADC\_ref-zero\_z)/sensitivity\_z;    Serial.print ("z= ");  Serial.print (zv);  Serial.print(" g ");    Serial.print("\n");    Serial.print("Rotation ");    Serial.print("x= ");    angle\_x =atan2(-yv,-zv)\*57.2957795+180;    Serial.print(angle\_x);  Serial.print(" deg");  Serial.print(" ");    Serial.print("y= ");    angle\_y =atan2(-xv,-zv)\*57.2957795+180;    Serial.print(angle\_y);  Serial.print(" deg");  Serial.print(" ");    Serial.print("z= ");    angle\_z =atan2(-yv,-xv)\*57.2957795+180;    Serial.print(angle\_z);  Serial.print(" deg");  Serial.print("\n");    delay(1000);  delay(1000);  } |