ADXL345 accelerometer breakout board

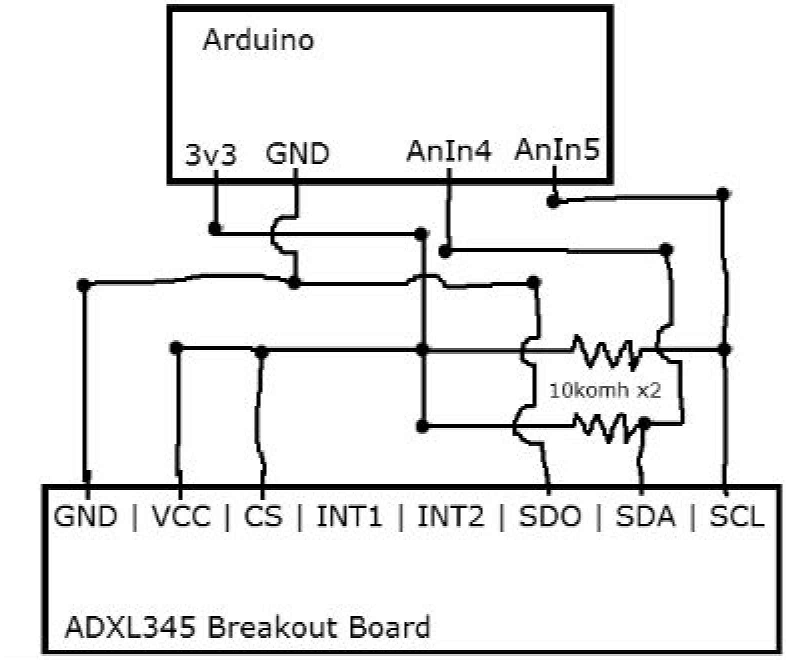
Introduction:
The ADXL345 is well suited to measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0°.Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins.

"An accelerometer measures acceleration. Which may seem obvious, however there are two types of acceleration. Static acceleration - such as gravity - which is what you will use your accelerometer to measure most often. Also dynamic acceleration, which is basically movement, that normally gets in the way of most of our tilt sensing applications we usually use accelerometers for."

Source:
Sparkfun.com for $15.
https://www.sparkfun.com/products/9045

Applications:
It is commonly used to track motion, because of this it can be used for many different reasons. It is used in activity trackers like the Jawbone UP or the Fitbit.

Datasheet:
2 - 10k ohn resistors
https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf



Code Example:

```
#include <Wire.h>

#define DEVICE (0x53)    //ADXL345 device address
#define TO_READ (6)        //num of bytes we are going to read each time (two bytes for each axis)

byte buff[TO_READ] ;    //6 bytes buffer for saving data read from the device
char str[512];                      //string buffer to transform data before sending it to the serial port

void setup()
{
  Wire.begin();        // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output

  //Turning on the ADXL345
  writeTo(DEVICE, 0x2D, 0);
  writeTo(DEVICE, 0x2D, 16);
  writeTo(DEVICE, 0x2D, 8);
}

void loop()
```

```
{
 int regAddress = 0x32;    //first axis-acceleration-data register on the ADXL345
 int x, y, z;

 readFrom(DEVICE, regAddress, TO_READ, buff); //read the acceleration data from the
ADXL345

  //each axis reading comes in 10 bit resolution, ie 2 bytes.  Least Significat Byte first!!
  //thus we are converting both bytes in to one int
 x = (((int)buff[1]) << 8) | buff[0];
 y = (((int)buff[3])<< 8) | buff[2];
 z = (((int)buff[5]) << 8) | buff[4];

 //we send the x y z values as a string to the serial port
 sprintf(str, "%d %d %d", x, y, z);
 Serial.print(str);
 Serial.write(10);

 //It appears that delay is needed in order not to clog the port
 delay(15);
}

//---------------- Functions
//Writes val to address register on device
void writeTo(int device, byte address, byte val) {
  Wire.beginTransmission(device); //start transmission to device
  Wire.write(address);        // send register address
  Wire.write(val);        // send value to write
  Wire.endTransmission(); //end transmission
}

//reads num bytes starting from address register on device in to buff array
void readFrom(int device, byte address, int num, byte buff[]) {
 Wire.beginTransmission(device); //start transmission to device
 Wire.write(address);        //sends address to read from
 Wire.endTransmission(); //end transmission

 Wire.beginTransmission(device); //start transmission to device
 Wire.requestFrom(device, num);    // request 6 bytes from device

 int i = 0;
 while(Wire.available())    //device may send less than requested (abnormal)
 {
  buff[i] = Wire.read(); // receive a byte
  i++;
 }
 Wire.endTransmission(); //end transmission
}
```

Typical Behavior:
You have to define a threshold for movement and it will work whenever that threshold is triggered.

Personal Application:
I would use this to track motion or the balance of something. This sensor could be used to trigger alarms for objects that have to be stable.

References:
http://codeyoung.blogspot.com/2009/11/adxl345-accelerometer-breakout-board.html
http://wiki.bildr.org/index.php/Triple_Axis_Accelerometer_Breakout_-_ADXL345

Really Helpful Example/Tutorial:
ADXL345 Accelerometer and Arduino Introduction

https://www.loveelectronics.co.uk/Tutorials/12/adxl345-accelerometer-arduino-tutorial

Setup:

- An ADXL345 Breakout Board.
- An Arduino Uno or Arduino Mega.
- A small breadboard.
- Two resistors (between 4.7k and 10k Ohms).
- A bundle of wires.
- A green LED.
- Some male headers.

Software:

- The Arduino development environment.
- The Love Electronics ADXL345 Arduino Library.