

ADXL335 Messungen

Einführung

Im folgenden Artikel habe ich beschrieben, wie ich den ADXL335 Beschleunigungsmesser mit einfachen Methoden geeicht habe. Ich verwende die gleiche Methode wie Adafruit, bestimme aber die Gleichungen für die Werte mit linearer Regression und nicht mit der map() – Funktion. Die map() Funktion hat für mich sehr schlechte Werte geliefert.

Ich habe dazu das ADXL335 Breakout Board von Adafruit (inklusive Adafruit Dokumentation), einen Arduino UNO und das Datenblatt ADXL335 verwendet. Zusätzlich habe ich einen HP32S zum Berechnen der linearen Regression eingesetzt.

Für die in diesem Dokument veröffentlichte Software gelten folgende Lizenzbestimmungen

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation without any warranty.

Messaufbau

Den Schaltungsaufbau habe ich der Adafruit Breakout Dokumentation entnommen und hier tabellarisch aufgeführt.

Kolonne: Arduino

Zeile: Breakout Board

	3V3	GND	A0	A1	A2
Vin	X				
GND		X			
zOut					x
yOut				x	
xOut			x		

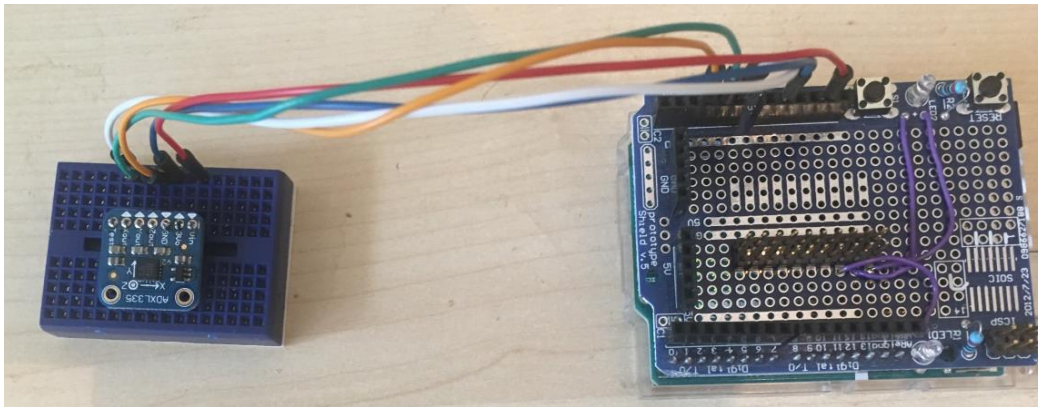
Der Schaltungsaufbau des Breakouts ist auf einem Minibreadboard gemacht.

Messprinzip

Im Datenblatt des ADXL335 ist zu sehen, wie die Gravitationskräfte auf den Sensor wirken. Bei den Bildern ist der gegebene Punkt auf dem ADXL335 zu beachten.

Liegt das Breadboard flach auf einer horizontalen Unterlage, so sind die Gravitationskräfte $g(x)$ und $g(y)$ jeweils 0 und die Gravitationskraft $g(z)$ ist ± 1 , je nach dem der Sensor nach

oben oder nach unten schaut.

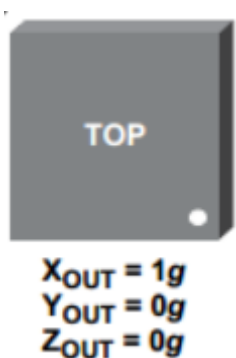


Ansicht von oben auf den Sensor. $g(x), g(y) = 0, g(z) = 1g$

Für alle Kanten, die Unterseite und die Oberseite jeweils horizontal ausgerichtet auf der Tischplatte, werden mit dem ersten Sketch die x-, y- und z-ADC-Werte für die Gravitation - 1g, 0g, 1g gemessen. Für die "Überkopflage" zum Messen des Wertes $g(z) = -1g$ das Breadboard auf die Unterkante des Tisches halten.

In jeder Position kann der zur Position gehörige Gravitationswert als ADC Ausgangswert gemessen werden wie den folgenden Kapiteln zu entnehmen ist. Die Werte für 0g habe ich jeweils nur in einer Position gemessen.

Positive Gravitation wirkt auf die x-Achse



Der Sensor steht auf der Kante, so dass der Orientierungspunkt rechts unten ist.

```
xAxisValADC(1g) = 408
```

```
yAxisValADC(0g) = 345
```

```
zAxisValADC(0g) = 363
```

Negative Gravitation wirkt auf die x-Achse

$X_{OUT} = -1g$
 $Y_{OUT} = 0g$
 $Z_{OUT} = 0g$



Der Sensor steht auf der Kante, so dass der Orientierungspunkt links oben ist.

`xAxisValADC(-1g) = 269`

Positive Gravitation wirkt auf die y-Achse

$X_{OUT} = 0g$
 $Y_{OUT} = 1g$
 $Z_{OUT} = 0g$



Der Sensor steht auf der Kante, so dass der Orientierungspunkt links unten ist.

`yAxisValADC(1g) = 408`

`xAxisValADC(0g) = 337`

Negative Gravitation wirkt auf die y-Achse



$X_{OUT} = 0g$
 $Y_{OUT} = -1g$
 $Z_{OUT} = 0g$

Der Sensor steht auf der Kante, so dass der Orientierungspunkt rechts oben ist.

`yAxisValADC(-1g) = 268`

Positive Gravitation wirkt auf die z-Achse



$X_{OUT} = 0g$
 $Y_{OUT} = 0g$
 $Z_{OUT} = 1g$

Der Sensor liegt flach auf dem Tisch, so dass der Orientierungspunkt nach oben schaut.

```
zAxisValADC(1g) = 431
```

Negative Gravitation wirkt auf die z-Achse

Der Sensor liegt flach auf dem Tisch, so dass der Orientierungspunkt nach unten schaut.



$X_{OUT} = 0g$
 $Y_{OUT} = 0g$
 $Z_{OUT} = -1g$

```
zAxisValADC(-1g) = 293
```

Auswertung

	-1g	0g	+1g
xAxisValADC	269	337	408
yAxisValADC	269	345	408
zAxisValADC	293	363	431

Diese Werte habe ich dem HP32S gefüttert und mit linearer Regression bin ich zu folgenden Gleichungen gekommen:

$$gx(xAxisValADC) = xAxisValADC * 0.0144 - 4.86$$

$$gy(yAxisValADC) = yAxisValADC * 0.0143 - 4.89$$

$$gz(zAxisValADC) = zAxisValADC * 0.0145 - 5.25$$

Damit lässt sich im Arbeitsbereich des Sensors (-3g ... +3g) der Gravitationswert für jede Achse wie folgt berechnen und per serielle Schnittstelle ausgeben:

```
Serial.print(float(xAxisValADC * 0.0144 - 4.86));  
Serial.print(float(yAxisValADC * 0.0143 - 4.89));  
Serial.print(float(zAxisValADC * 0.0145 - 5.25));
```

Sketch zur Bestimmung der x- y- und z-Werte

```
#include <Arduino.h>

int xAxisPin = A0;
int yAxisPin = A1;
int zAxisPin = A2;

//Variables to hold the returned
//ADC data from the analog input
//pins
int xAxisValADC = 0;
int yAxisValADC = 0;
int zAxisValADC = 0;

//Arduino Uno with a 10-bit resolution (max output value = 1023)
const int ADCMaxVal = 1023;

//The AD converter voltage, measured with a voltage meter in mV
const float VREF = 4810;

int run = 0;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  analogReference(EXTERNAL);
  pinMode(xAxisPin, INPUT);
  pinMode(yAxisPin, INPUT);
  pinMode(zAxisPin, INPUT);
}

void loop()
{
  //Read the x, y, and z values from
  //the analog input pins and hold them
  xAxisValADC = analogRead(xAxisPin);
  yAxisValADC = analogRead(yAxisPin);
  zAxisValADC = analogRead(zAxisPin);
  Serial.print("Gravitation run ");
  Serial.print(++run);
  if (run == 65535)
  {
    run = 0;
  }
  Serial.print("\tx: ");
  Serial.print(xAxisValADC);

  Serial.print("\ty: ");
  Serial.print(yAxisValADC);

  Serial.print("\tz: ");
  Serial.print(zAxisValADC);
  Serial.println();
  delay(2000);
}
```

Sketch mit Ausgabe der ADC Werte und der Gravitation

```
#include <Arduino.h>

int xAxisPin = A0;
int yAxisPin = A1;
int zAxisPin = A2;

//Variables to hold the returned
//ADC data from the analog input
//pins
int xAxisValADC = 0;
int yAxisValADC = 0;
int zAxisValADC = 0;

//Arduino Uno with a 10-bit resolution (max output value = 1023)
const int ADCMaxVal = 1023;

//The AD converter voltage, measured with a voltage meter in mV
const float VREF = 4810;

int run = 0;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
  analogReference(EXTERNAL);
  pinMode(xAxisPin, INPUT);
  pinMode(yAxisPin, INPUT);
  pinMode(zAxisPin, INPUT);
}

void loop()
{
  //Read the x, y, and z values from
  //the analog input pins and hold them
  xAxisValADC = analogRead(xAxisPin);
  yAxisValADC = analogRead(yAxisPin);
  zAxisValADC = analogRead(zAxisPin);
  Serial.print("Gravitation run ");
  Serial.print(++run);
  if (run == 65535)
  {
    run = 0;
  }
  Serial.print("\tx: ");
  Serial.print(xAxisValADC);
  Serial.print("\t");

  Serial.print(float(xAxisValADC * 0.0144 - 4.86));
  Serial.print("g");

  Serial.print("\ty: ");
  Serial.print(yAxisValADC);
  Serial.print("\t");
  Serial.print(float(yAxisValADC * 0.0143 - 4.89));
  Serial.print("g");

  Serial.print("\tz: ");
  Serial.print(zAxisValADC);
  Serial.print("\t");
  Serial.print(float(zAxisValADC * 0.0145 - 5.25));
  Serial.print("g");
  Serial.println();
  delay(2000);
}
```