

ЛАБОРАТОРНА РОБОТА №2

Проектування та дослідження комп'ютеризованого вимірювача прискорень

1.1 Мета роботи

Зпроектувати та відкалібрувати вимірювач прискорень на базі ардуіно та датчика ADXL345, обчислити крен та кут тангажу.

1.2. Короткі теоретичні відомості

ADXL345 - це 3-осьовий акселерометр, який може вимірювати як статичні, так і динамічні сили прискорення. Сила земного тяжіння є типовим прикладом статичної сили, в той час як динамічні сили можуть бути викликані вібраціями, рухами і так далі.

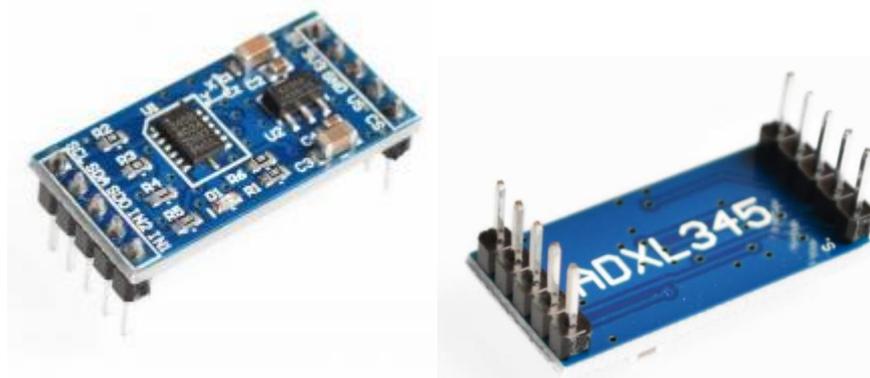


Рисунок 2.1 – 3-осьовий акселерометр ADXL345

3-х осьовий цифровий акселерометр на основі чіпа ADXL345 є датчиком, що вимірює проекції прискореного на три просторові осі (x, y, z). Знаючи ці вимірювання и з огляду на величину вільного падіння, можна визначити орієнтацію самого акселерометра в просторі. Цифрові результати вимірювання представляються у вигляді 16-розрядно чисел в додатковий коді та доступні через цифрові інтерфейси SPI (трьох- або чотирьохпровідні) або I2C.

Чутливість сенсора налаштовується (максимальна 4 mg/LSB). Частота оновлення інформації від 0,1 до 3200 Гц. . Напруга живлення чіпа 2.0 - 3.6В (можна живить з виходом 3.3В Ардуіно), плата може бути живитися від 5В (є вбудований регулятор напруги), споживання струму під час роботи 23мкА. Датчик працює як з SPI так и I2C інтерфейсами.

Дуже простий в роботі. Вивід CS використовується для Вибори інтерфейсу. Якщо на CS низьких рівень - використовується SPI інтерфейс. Якщо ж на вході CS високий рівень - використовується I2C інтерфейс.

Характеристики:

- Напруга живлення: 3.3 – 5.0 В;

- Інтерфейс шини: I2C / SPI;
- Формат пакета даних: 13 біт;
- Додаткові виводи переривань;
- Робочий діапазон: $\pm 2 \text{ g} \dots \pm 16 \text{ g}$.

Загальновідома одиниця вимірювання прискорення - метр на секунду в квадраті (m/s^2). Однак акселерометри зазвичай видають значення в «g» або гравітаційне прискорення. Один «g» - це величина прискорення надається тілу силою тяжіння на поверхні землі і в середньому дорівнює $9,8 \text{ m} / \text{s}^2$.

Отже, якщо у нас акселерометр, розташований горизонтально, з його віссю Z, спрямованої вгору, (протилежної силі тяжіння) вихідний сигнал по осі Z буде дорівнює $1g$. Вихідні сигнали по осях X і Y дорівнюватимуть нулю, тому що гравітаційна сила перпендикулярна до цих осей і ніяк на них не впливає.

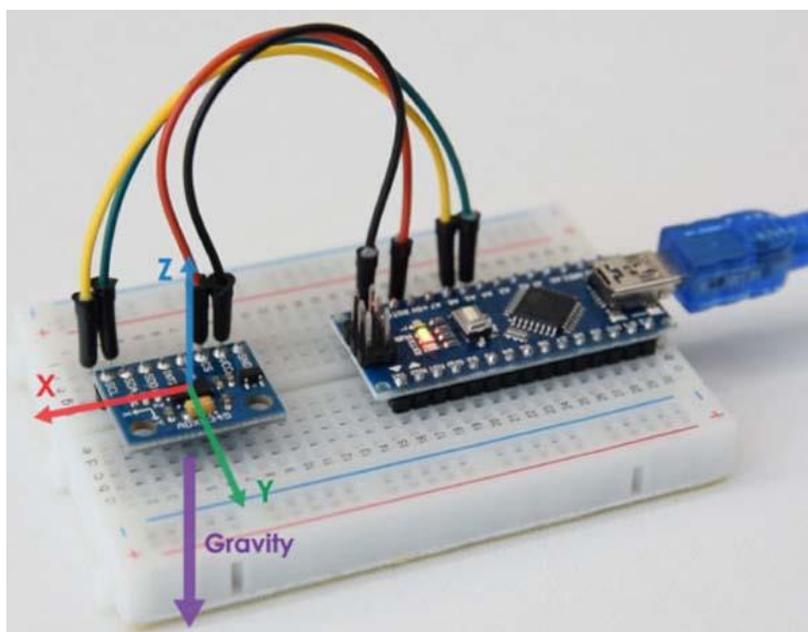


Рисунок 2.2 – Розташування осей акселерометра

Якщо перевернути датчик догори ногами, то вихідний сигнал по осі Z буде $-1g$. Це означає, що вихідні дані датчика через його орієнтації під дією сили тяжіння можуть варіюватися від $-1g$ до $+1g$.

Добре, тепер давайте подивимося, як ми можемо прочитати дані акселерометра ADXL345 за допомогою Arduino. Для зв'язку з Arduino датчик ADXL345 використовує I2C протокол. Отже, нам потрібні тільки два дроти для його підключення (лінії даних) і два дроти для його харчування.

Arduino and ADXL345 Accelerometer

I²C Communication

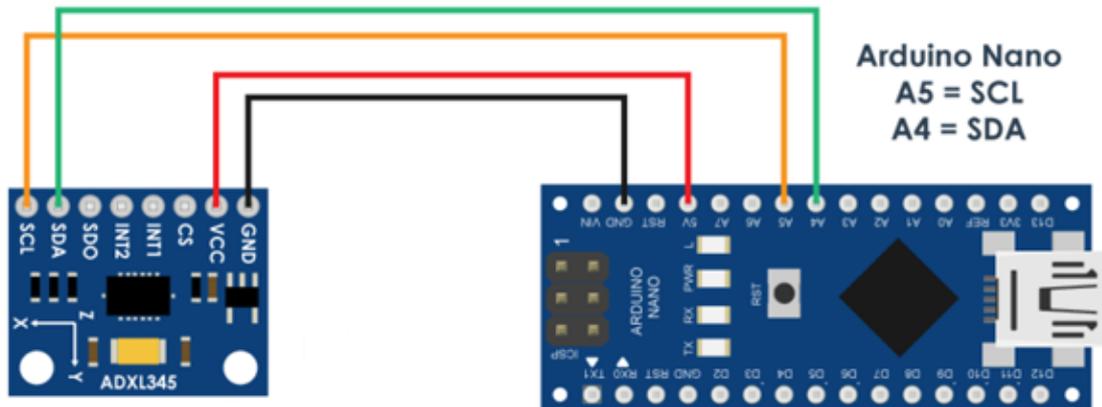


Рисунок 2.3 – Схема підключення акселерометра до Ардуіно

Скетч для читання даних акселерометра ADXL345 представлений нижче:

```
#include <Wire.h> // підключення бібліотеки Wire
int ADXL345 = 0x53; // I2C адреса датчика ADXL345
float X_out, Y_out, Z_out; // Виходи
void setup() {
  Serial.begin(9600);
  Wire.begin(); // Ініціалізація бібліотеки Wire
  // Встановить ADXL345 у режим вимірювання
  Wire.beginTransmission(ADXL345); // Почати зв'язок з пристроєм
  Wire.write(0x2D); // робота з регістром POWER_CTL - 0x2D
  // Увімкнути вимірювання
  Wire.write(8); // (8dec -> 0000 1000 двійковий) Біт D3 High для дозволу вимірювання
  Wire.endTransmission();
  delay(10);
}
void loop() {
  // === Зчитати дані акселерометра === //
  Wire.beginTransmission(ADXL345);
  Wire.write(0x32); // Почати з регістра 0x32 (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(ADXL345, 6, true); // Читання всіх 6 регістрів, значення кожної осі
  зберігається у 2 регістрах
  X_out = ( Wire.read() | Wire.read() << 8); // Значення по осі X
  X_out = X_out/256; // Для діапазону ±2g нам потрібно розділити отримані значення на
  256 відповідно до datasheet
  Y_out = ( Wire.read() | Wire.read() << 8); // Значення по осі Y
  Y_out = Y_out/256;
  Z_out = ( Wire.read() | Wire.read() << 8); // Значення по осі Z
  Z_out = Z_out/256;
  Serial.print("Xa= ");
  Serial.print(X_out);
  Serial.print(" Ya= ");
  Serial.print(Y_out);
  Serial.print(" Za= ");
  Serial.println(Z_out);
}
```

Вихідні дані датчика фактично залежать від обраної чутливості, яка може варіюватися від $+2g$ до $-16g$. Чутливість за замовчуванням становить $+2g$, тому нам потрібно розділити результат на 256, щоб отримати значення від -1 до $+1g$. $256 \text{ LSB} / g$ означає, що у нас 256 відліків на $1g$.

Залежно від нашої потреби ми можемо вибрати відповідну чутливість. У разі для стеження орієнтації цілком підійде чутливість $+2g$, але для додатків, де нам потрібно фіксувати вищу силу прискорення, наприклад, від раптових рухів, ударів і т. д. Можна вибрати інші діапазони чутливості, використовуючи регістр `DATA_FORMAT` і його біти `D1` та `D0`.

Після того як ми прочитали дані, ми можемо просто вивести їх в моніторі послідовного порту, щоб перевірити, чи відповідають значення очікуваним.

Для того щоб відкалібрувати акселерометр потрібно використати 3 регістра калібрування зміщення. Отже, потрібно розташувати датчик рівно і роздрукувати значення RAW, що не ділячи їх на 256 рис.2.4.

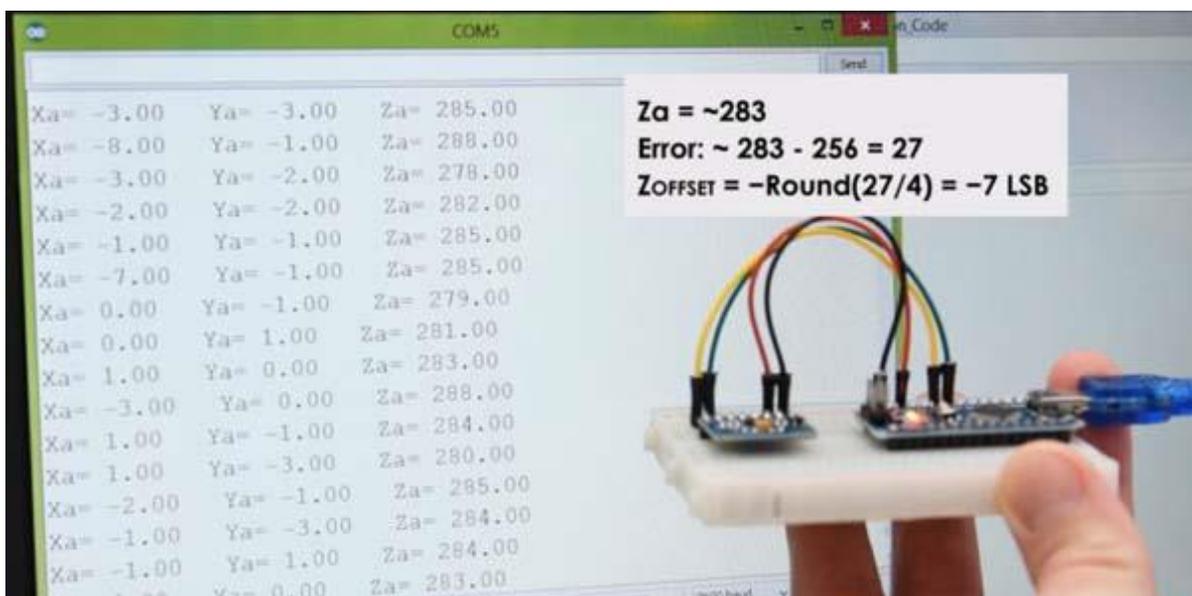


Рисунок 2.4 – Роботи акселерометра з відхиленням

Тут ми можемо бачити відхилення від норми, в нашому випадку значення по осі Z було близько 283. Це позитивна різниця в 27. Тепер нам потрібно розділити це значення на 4, і це дасть нам число, яке нам потрібно записати в регістр зсуву осі Z . якщо після цього ми завантажимо скетч, то значення по осі Z буде рівно 256, або $1g$, як і повинно бути.

Частина скетчу для калібрування акселерометра представлений нижче.

```
Wire.beginTransmission(ADXL345); // режим калібрування
Wire.write(0x1E); // Регістр зміщення (offset) осі X
Wire.write(-7);
Wire.endTransmission();
delay(10);
```

```

// вісь Y
Wire.beginTransmission(ADXL345);
Wire.write(0x1F); // Регістр зміщення (offset) осі Y
Wire.write(-8);
Wire.endTransmission();
delay(10);

// вісь Z
Wire.beginTransmission(ADXL345);
Wire.write(0x20); // Регістр зміщення (offset) осі Z
Wire.write(-12);
Wire.endTransmission();
delay(10);

```

При необхідності нам слід відкалібрувати і інші осі, використовуючи той же метод. І відразу зауважимо, що ця калібрування не записує в регістри постійно. Нам потрібно записувати ці значення в регістри при кожному включенні датчика.

Після того, як ми завершили калібрування, можна обчислити крен і кут тангажу, використовуючи дві формули:

1. // Розрахунок крену та тангажу (обертання навколо осі X, обертання /навколо осі Y)
2. $roll = atan(Y_out / \sqrt{pow(X_out, 2) + pow(Z_out, 2)}) * 180 / PI;$
3. $pitch = atan(-1 * X_out / \sqrt{pow(Y_out, 2) + pow(Z_out, 2)}) * 180 / PI;$

1.3. Підготовка до роботи

При підготовці до роботи необхідно:

- ознайомитись з рекомендованою літературою;
- вивчити короткі теоретичні відомості.

1.4. Порядок роботи:

1. Підключити модуль ArduinoUno до акселерометра ADXL345 по протоколу I2C.
2. Скачати та підключити в скетчі необхідні бібліотеки для роботи з акселерометра ADXL345.
3. Відкалібрувати акселерометр по 3-ом осям.
4. Обчислити крен і кут тангажу.
5. Виконати вимірювання прискорення при повільному русі, при швидкому русі, при поступальному русі.
6. Представити виміряні дані в вигляді таблицок.
7. Побудувати графіки залежності часу від прискорення та часу від швидкості при трьох різних видів руху.
8. Оформити звіт та зробити висновки.