TP Accelerometre ADXL345 I2C SPI

Capteur

Les capteurs MEMS sont des composants électronique qui permettent de mesurer une grandeur physique. MEMS signifie Micro Electro Mechanical System. Les MEMS sont produits avec les mêmes techniques que celles utilisées dans la fabrication de semi conducteur. Dans un MEMS, des micro dispositifs mécaniques sont gravés sur un morceau de silicium.
Les accéléromètres sont des MEMS utilisées dans de nobreux objets du quotidien (smartphone, voiture, manette de jeux vidéos, …). Ils permettent de mesurer une accélération. C'est à dire un changement de vitesse. Attention un accéléromètre ne vous donnera pas la vitesse absolue d'un objet. En revanche, si l'objet accélère ou ralenti, il vous dira combien l'objet aura gagné ou perdu en vitesse.
L'accéléromètre mesure l'accélération suivant les trois degrées de liberté X, Y et Z.
Dans ce TP, nous allons mettre en œuvre un accéléromètre.

Partie 1 :

Question n°1 :
Télechargez la Datasheet da la puce MEMS adxl345. Quel est le nom du fabricant de la puce ? A quelle page se trouve le tableau des registres ? Trouvez le chapitre qui décrit l'échange avec le protocole I2C.

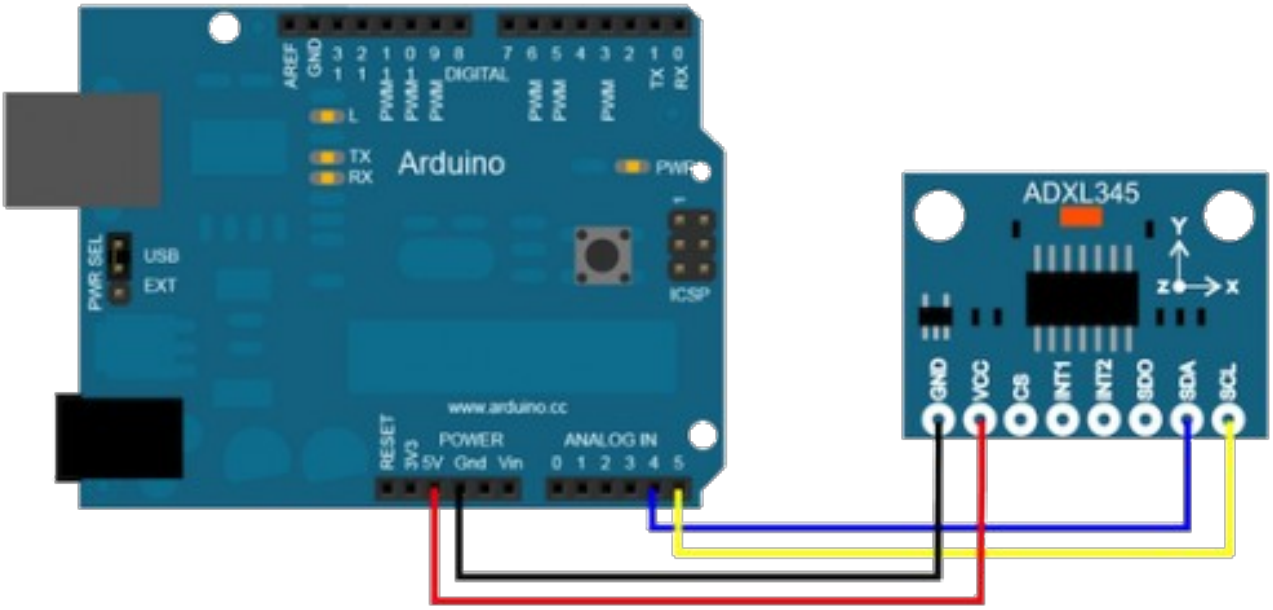
Question n°2 :
Réalisez le montage électrique afin de communiquer avec l'ADXL345 en utilisant le protocole I2C.

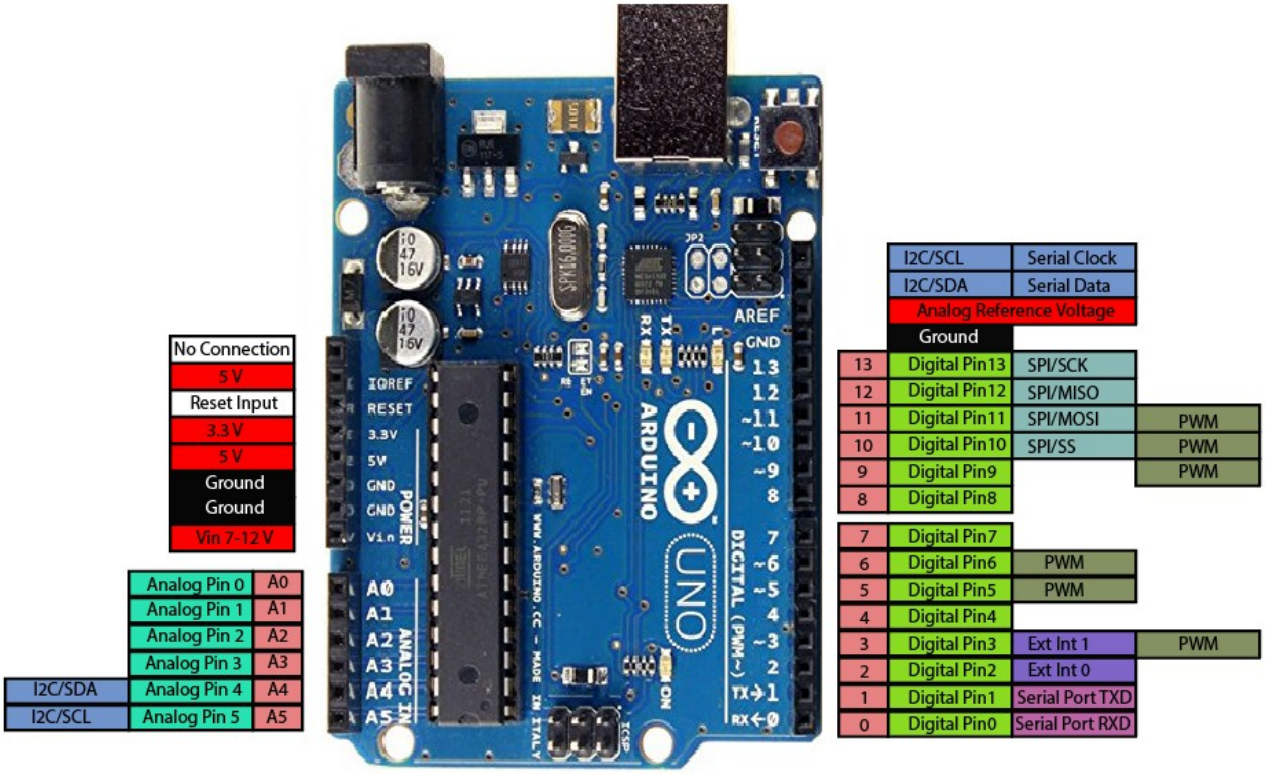
Question n°3 :
Utilisez le code Arduino (en page 3) pour communiquer avec l'ADXL345. Ouvrez un terminal et déplacez le capteur pour observer les variation dans le terminal

Question n°4 :
Utilisez le Moniteur Série (différent du terminal série) pour afficher les courbes d'accélération du capteur.

Question n°5 :
Modifiez le code précedent pour lire le registre DEVID de l'ADXL345 et vérifier que votre code fonctionne correctement

Question n°6 :
Modifiez le code du départ, pour séparer en deux fonction différentes la lecture et l'écriture dans un registre :
Read_REG() qui prend en paramètre l'adresse du composant et l'adresse d'un registre et renvois le contenus du registres
Write_REG() qui prend en paramètre l'adresse du composant, l'adresse d'un registre et un octet puis écrit cet octet à l'adresse du registre.


Arduino
ADXL345
GND
VCC
CS
INT1
INT2
SDO
SDA
SCL
POWER
ANALOG IN
RESET
www.arduino.cc


No Connection
5 V
Reset Input
3.3 V
5 V
Ground
Ground
Vin 7-12 V
Analog Pin 0 A0
Analog Pin 1 A1
Analog Pin 2 A2
Analog Pin 3 A3
I2C/SDA Analog Pin 4 A4
I2C/SCL Analog Pin 5 A5
I2C/SCL Serial Clock
I2C/SDA Serial Data
Analog Reference Voltage
Ground
13 Digital Pin13 SPI/SCK
12 Digital Pin12 SPI/MISO
11 Digital Pin11 SPI/MOSI PWM
10 Digital Pin10 SPI/SS PWM
9 Digital Pin9 PWM
8 Digital Pin8
7 Digital Pin7
6 Digital Pin6 PWM
5 Digital Pin5 PWM
4 Digital Pin4
3 Digital Pin3 Ext Int 1 PWM
2 Digital Pin2 Ext Int 0
1 Digital Pin1 Serial Port TXD
0 Digital Pin0 Serial Port RXD

```
#include <Wire.h> // Wire library - used for I2C communication
int ADXL345 = 0x53; // The ADXL345 sensor I2C address
float X_out, Y_out, Z_out; // Outputs

void setup() {
  Serial.begin(9600); // Initiate serial communication for printing the results on the Serial monitor
  while(Serial.available() != 0){
  }
  Serial.println("Start");
  Wire.begin(); // Initiate the Wire library // Set ADXL345 in measuring mode
  Wire.beginTransmission(ADXL345); // Start communicating with the device
  Wire.write(0x2D); // Access/ talk to POWER_CTL Register - 0x2D // Enable measurement
  Wire.write(8); // (8dec -> 0000 1000 binary) Bit D3 High for measuring enable
  Wire.endTransmission();
  delay(10);
}

void loop() { // === Read acceleromter data === //
  Wire.beginTransmission(ADXL345);
  Wire.write(0x32); // Start with register 0x32 (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(ADXL345, 6, true); // Read 6 registers total, each axis value is stored in 2
registers
  X_out = ( Wire.read()| Wire.read() << 8); // X-axis value
  X_out = X_out/256; //For a range of +-2g, we need to divide the raw values by 256, according to
the datasheet
  Y_out = ( Wire.read()| Wire.read() << 8); // Y-axis value
  Y_out = Y_out/256;

  Z_out = ( Wire.read()| Wire.read() << 8); // Z-axis value
  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);
}
```