

# Introduction à l'électronique digitale

## Microcontrôleurs

### **TP n. 3 Capteurs 1: accéléromètre et gyroscope**

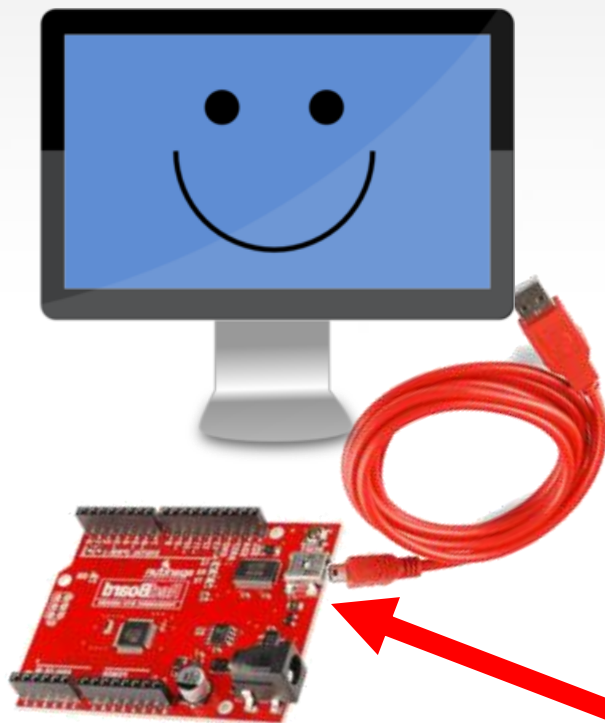


Section de Physique

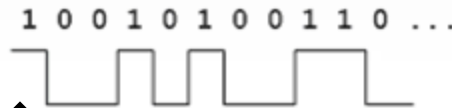
D. Mari, C. Meinen

# Communication sériele

## Méthode utilisée pour transférer les données entre deux appareils



Entre l'ordinateur et et Arduino on utilise le BUS USB.  
Les données sont transmises séquentiellement  
comme ('0') et ('1').

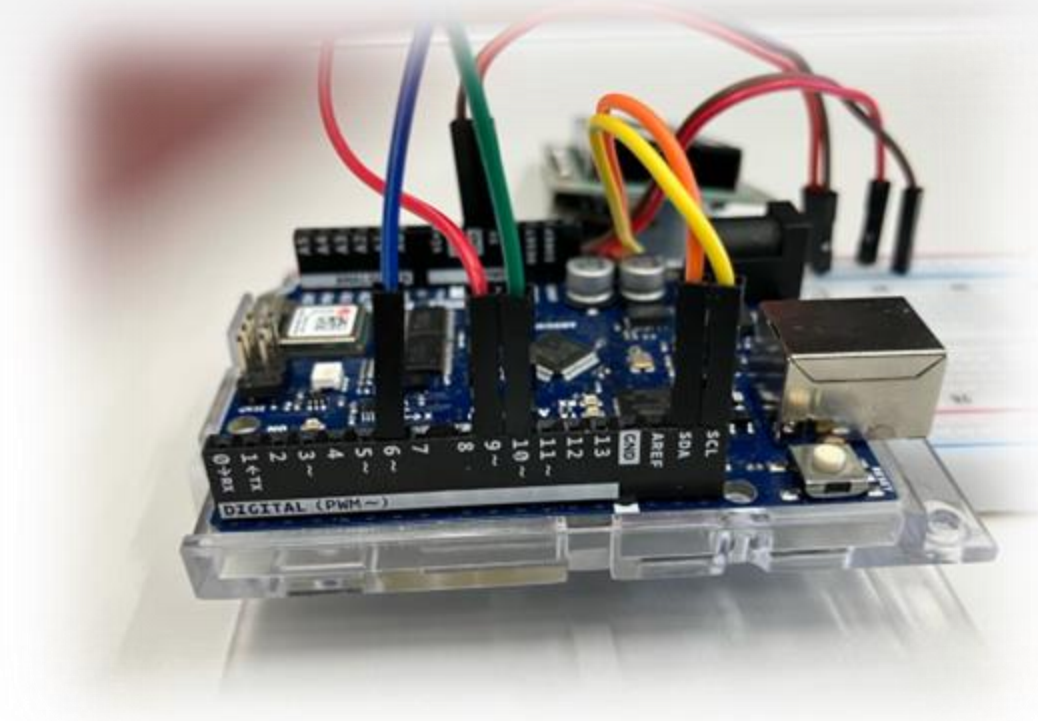


Arduino réserve le Digital I/O pin # 0 à la  
reception (RX) et le Digital I/O pin #1 à la  
transmission (TX)

Mais on a également un socle  
pour le cable USB

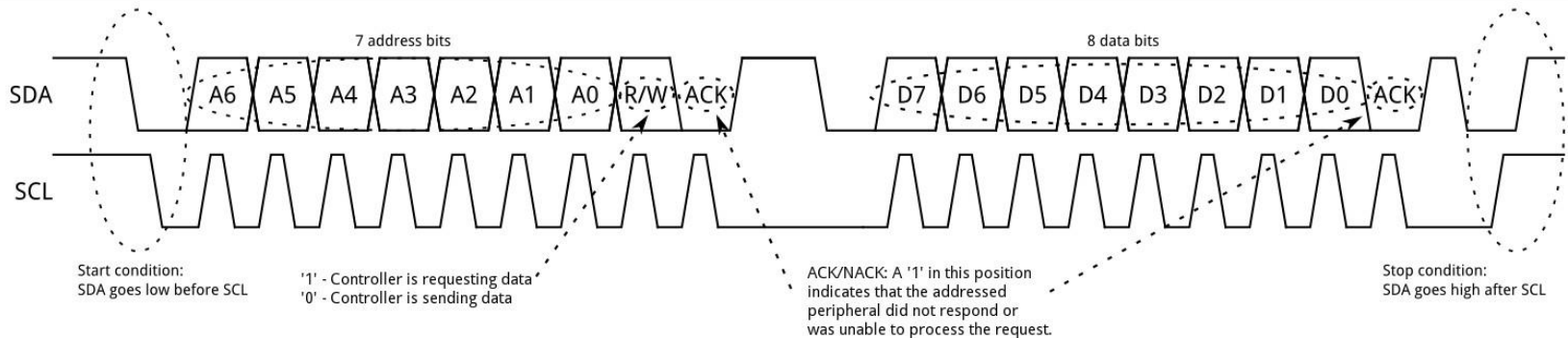
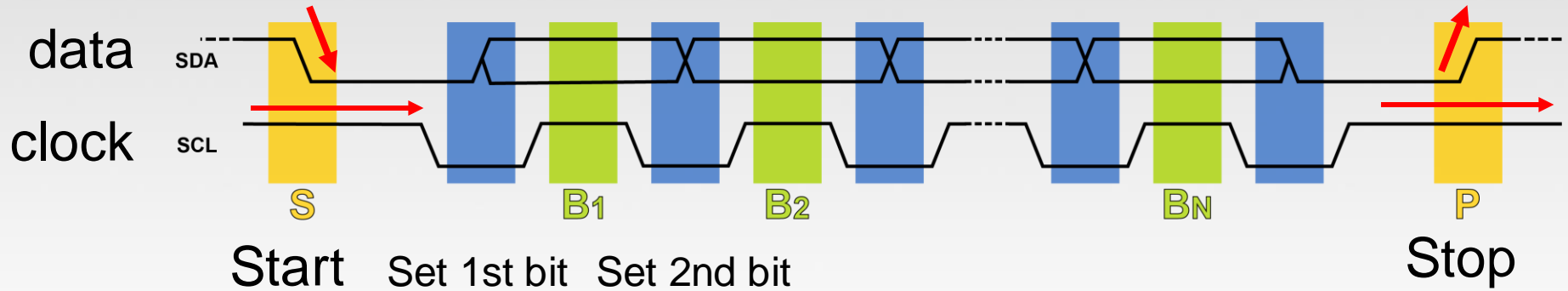
# On utilise l'interface IIC pour communiquer avec les périphériques

IIC transmet deux signaux :  
SCL (clock) et SDA (data)



# BUS de données

## IIC ou I<sup>2</sup>C ((Inter-Integrated Circuit)



# Tester le protocole IIC

Activer IIC avec un programme (p.ex ColourRGB)  
contenant des instructions pour IIC: `lcd.print(" ")`

Connecter la sonde à un pin lié à SCL  
et analyser les signaux du clock et  
data avec l'oscilloscope



- Sur quel canal faut il mettre la source du trigger ?
- Quelle est la fréquence du signal clock ?
- Quelle est la période des impulsions ?
- ..et celle d'une instruction ?

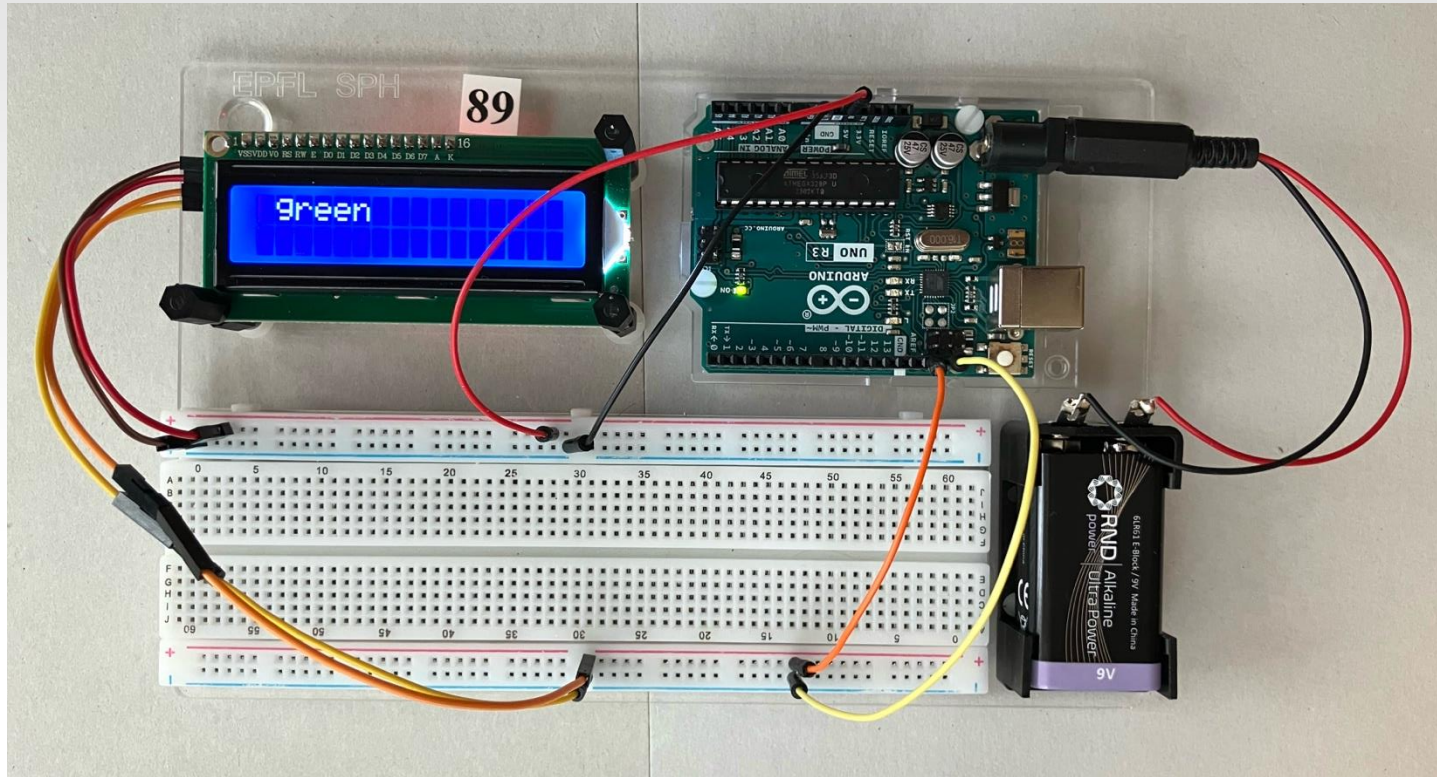
7 bit Address		Acknowledge		Acknowledge	
Start	0110101	1	0	01100101	0
Read/Write			Data pack 1		

# Brancher le système de capteurs LSM9SD1 (avec interface I2C)



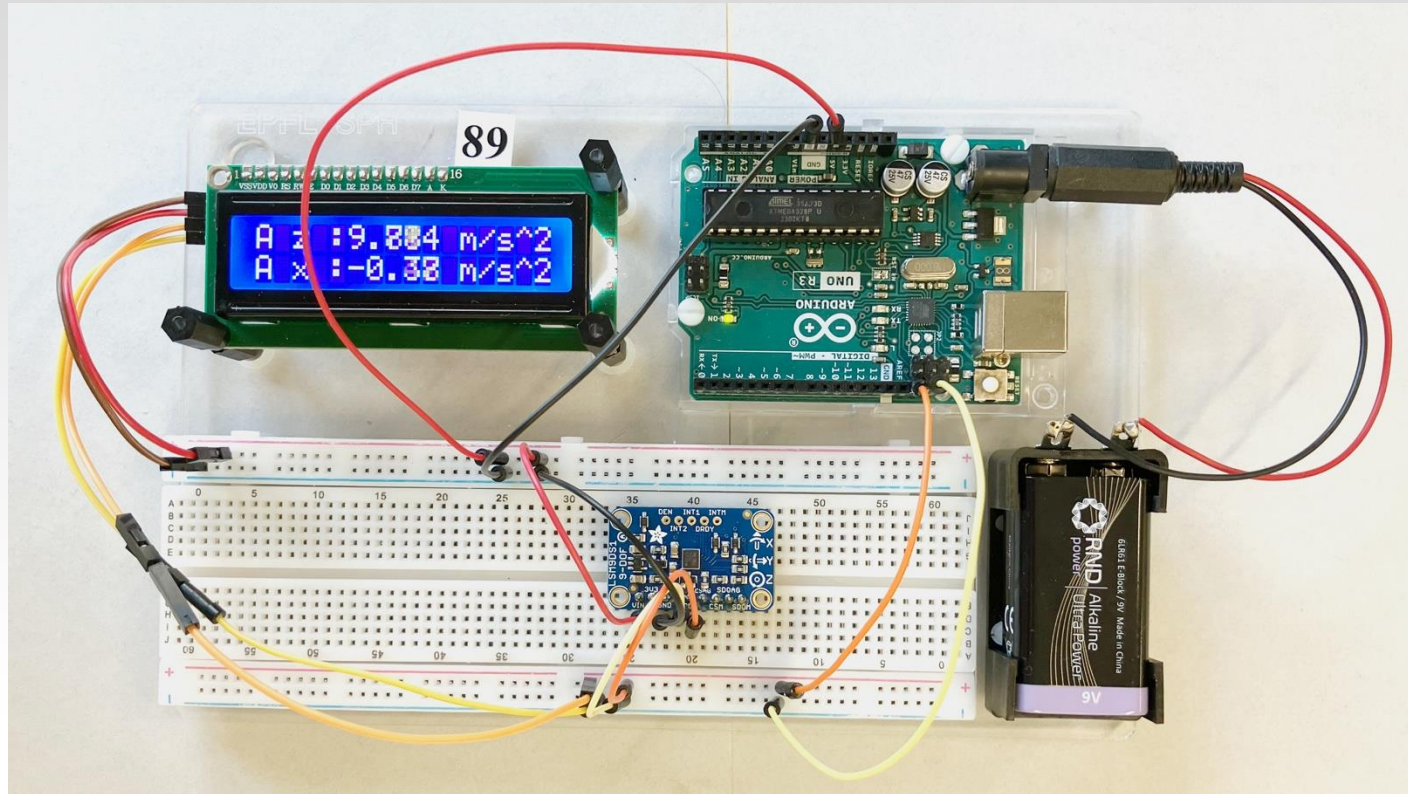


# Brancher l'écran LCD sur I2C



Conseil : utiliser deux pistes pour l'alimentation  
et 2 pistes pour SCL et SDA

# Accéléromètre 3 axes



Se familiariser avec le programme  
Accelerometre.ino

Afficher les accélérations selon x,y,z



# Accelerometre.ino

```
1  #include <Wire.h>
2  #include <SPI.h>
3  #include <Adafruit_LSM9DS1.h>
4  #include <Adafruit_Sensor.h>
5  #include <LiquidCrystal_I2C.h>
6
7  LiquidCrystal_I2C lcd(0x27, 16, 2);
8  Adafruit_LSM9DS1 lsm = Adafruit_LSM9DS1();
9
10 sensors_event_t a, m, g, temp;
11
12 void setupSensor()
13 {
14     // 1.) Set the accelerometer range
15     lsm.setupAccel(lsm.LSM9DS1_ACCELRange_2G);
16     //lsm.setupAccel(lsm.LSM9DS1_ACCELRange_4G);
17     //lsm.setupAccel(lsm.LSM9DS1_ACCELRange_8G);
18     //lsm.setupAccel(lsm.LSM9DS1_ACCELRange_16G);
19
20     // 2.) Set the magnetometer sensitivity
21     lsm.setupMag(lsm.LSM9DS1_MAGGAIN_4GAUSS);
22     //lsm.setupMag(lsm.LSM9DS1_MAGGAIN_8GAUSS);
23     //lsm.setupMag(lsm.LSM9DS1_MAGGAIN_12GAUSS);
24     //lsm.setupMag(lsm.LSM9DS1_MAGGAIN_16GAUSS);
25
26     // 3.) Setup the gyroscope
27     lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_245DPS);
28     //lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_500DPS);
29     //lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_2000DPS);
30 }
```

Charger les librairies

- Adafruit LSM9DS1
- Adafruit Unified Sensor
- Adafruit BusIO
- LiquidCrystal I2C

Range

# Accelerometre.ino

```
36 Serial.println("LSM9DS1 data read demo");
37
38 // Try to initialise and warn if we couldn't detect the chip
39 if (!lsm.begin())
40 {
41   Serial.println("Oops ... unable to initialize the LSM9DS1. Check your wiring!");
42   while (1);
43 }
44 Serial.println("Found LSM9DS1 9DOF");
45 setupSensor();
46 lcd.init();
47 lcd.backlight();
48
49 void loop() {
50   lsm.read();
51   lsm.getEvent(&a, &m, &g, &temp);
52
53   lcd.setCursor(0, 0);
54   lcd.print("A z : ");
55   lcd.setCursor(5, 0);
56   lcd.print(a.acceleration.z);
57   lcd.setCursor(10, 0);
58   lcd.print(" m/s^2");
59
60   lcd.setCursor(0, 1);
61   lcd.print("A x : ");
62   lcd.setCursor(5, 1);
63   lcd.print(a.acceleration.x);
64   lcd.setCursor(10, 1);
65   lcd.print(" m/s^2");
66   delay(500);
67 }
```



# Accéléromètre 3 axes

## exercices

Exercices suggérés:

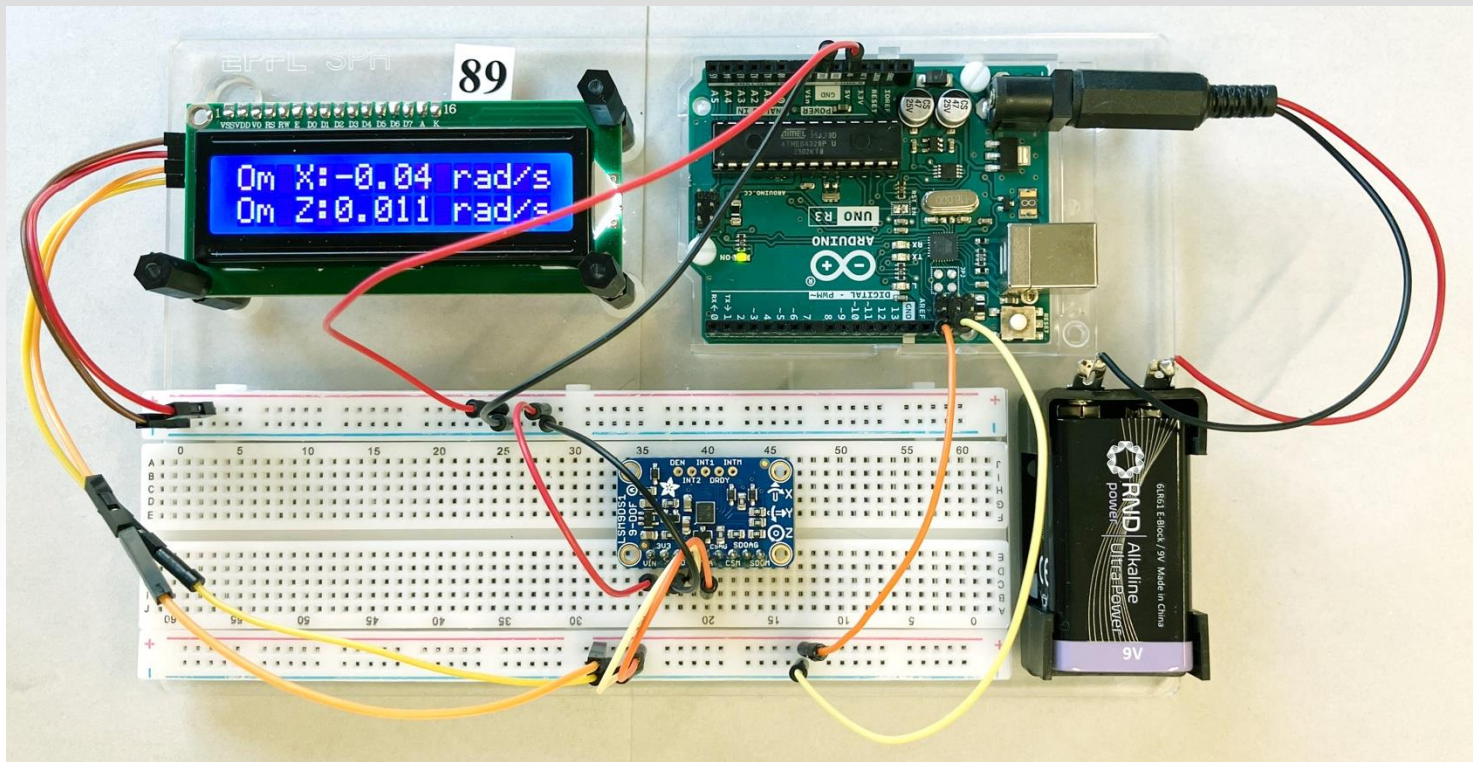
- Contrôler l'accélération de gravité
- Utiliser une LED pour afficher l'intensité d'une accélération avec un seuil (condition if )
- Éventuellement 3 LED
- Peut-on montrer l'accélération tangentielle et radiale lors de la chute (avec rotation) du kit ?

Pour les manipulations avec mouvement charger le programme et travailler en autonomie sur batterie

# Programmation : Boolean Operators

<Boolean>	Description
( ) == ( )	is equal?
( ) != ( )	is not equal?
( ) > ( )	greater than
( ) >= ( )	greater than or equal
( ) < ( )	less than
( ) <= ( )	less than or equal

# Gyroscope 3 axes



Se familiariser avec le programme  
Vitesse angulaire-acc\_rad.ino

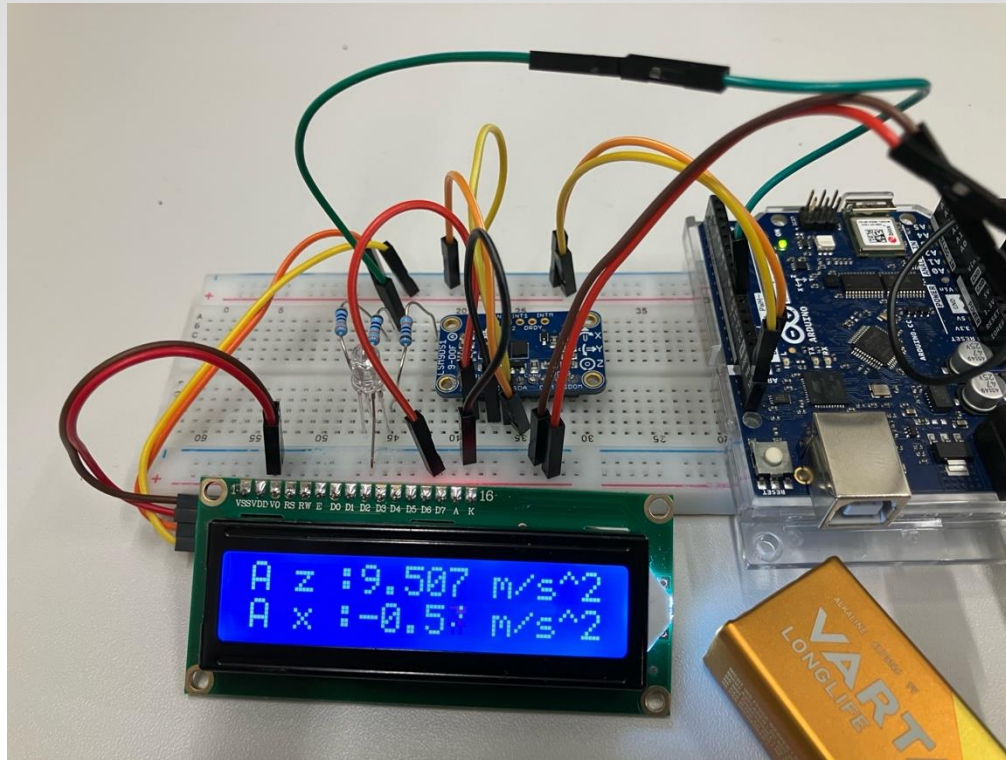
But montrer les accélérations lors d'un mouvement circulaire



# Vitesse angulaire-acc\_rad.ino

```
52 void loop() {  
53     lsm.read();  
54     lsm.getEvent(&a, &m, &g, &temp);  
55  
56     lcd.setCursor(0, 0);  
57     lcd.print("A Y : ");  
58     lcd.setCursor(6, 0);  
59     lcd.print(a.acceleration.y);   
60     lcd.setCursor(10,0);  
61     lcd.print(" m/s^2");  
62  
63     lcd.setCursor(0,1);  
64     lcd.print("Om R: ");  
65     lcd.setCursor(5, 1);  
66     lcd.print(g.gyro.z);   
67     lcd.setCursor(10,1);  
68     lcd.print(" rad/s");  
69     delay(500);  
70 }
```



# Gyroscope 3 axes



Se familiariser avec le programme  
gyroscope.ino

Afficher les vitesse de rotation selon x,y,z

# gyroscope.ino

```
void loop() {  
  lsm.read();  
  lsm.getEvent(&a, &m, &g, &temp);  
  
  lcd.setCursor(0, 0);  
  lcd.print("Om X: ");  
  lcd.setCursor(5, 0);  
  lcd.print(g.gyro.x);  
  Serial.print(g.gyro.x);   
  lcd.setCursor(10,0);  
  lcd.print(" rad/s");  
  
  lcd.setCursor(0,1);  
  lcd.print("Om Z: ");  
  lcd.setCursor(5, 1);   
  lcd.print(g.gyro.z);  
  lcd.setCursor(10,1);  
  lcd.print(" rad/s");  
  delay(500);  
}
```