

Journal de bord

19/01/24 : Choix et explication sommaire du projet (Essayer de faire un sismomètre à partir de 3 capteurs de vibrations)

Etude du fonctionnement général et des caractéristiques et composantes des sismomètres :

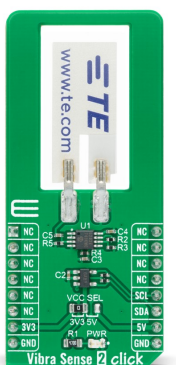
- Appareil capable de détecter de très petits mouvements du sol et de les enregistrer (analogiquement ou numériquement) en suivant une base de temps très précise (séisme = vibrations provoquées par la cassure de la croûte terrestre)
- Comporte un capteur mécanique, un transducteur, un amplificateur et un enregistreur
- Sismomètres utilisés actuellement : électromagnétiques (mécanisme de pendule avec un aimant se déplaçant dans une bobine de cuivre), chaque capteur est unidirectionnel, il en faut 3 (vertical, Est-Ouest, Nord-Sud)
- Le mouvement du sol z_{sol} est lié au mouvement de la masse z par l'équation suivante, avec α la constante d'amortissement du système, ω_0 la pulsation propre de l'oscillateur et M l'amplification :

$$\ddot{z} + 2\alpha\omega\dot{z} + \omega_0^2 z = M\ddot{z}_{\text{sol}}$$

Ici, l'idée est de faire un sismomètre à 3 composantes. Ces sismomètres sont particulièrement adaptés à l'étude de la sismicité locale et permettent de reconstituer le mouvement du sol en trois dimensions.

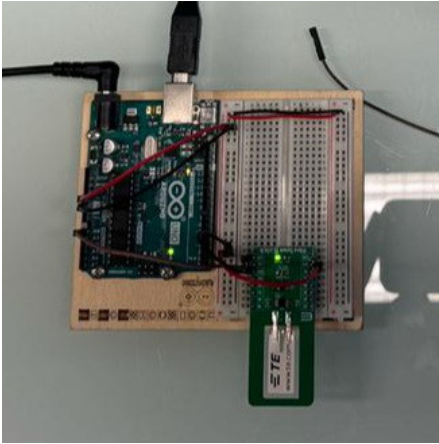
26/01/24 : Rencontre avec le responsable du projet et explication des attentes/cahier des charges, des résultats déjà obtenus et des consignes de réalisation.

Le capteur est composé d'un film piezoélectrique de 28µm d'épaisseur, laminé et monté sur un click-board :



Le but du projet est d'utiliser 3 de ces capteurs pour créer un sismomètre à 3 composantes facile à monter et peu coûteux (Les capteurs coûtent moins de 10euros chacun). Il faudra donc positionner les 3 capteurs selon les 3 axes orthogonaux de l'espace.

Début du travail avec le capteur sur une carte Arduino UNO R3. Alimentation du capteur :



Draft de code :

```
int sensorPin = A0;
int ledPin = 13;
int sensorValue = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(sensorPin);
  print(sensorValue);
  if (sensorValue > 700){
    digitalWrite(ledPin, HIGH);
  }

  else{
    digitalWrite(ledPin, LOW);
  }
}
```

02/02/24 : Travail sur le montage Arduino et le programme associé (pour l'instant pas de résultats satisfaisants)

Code dans la carte Arduino :

```
int sensorPin=A0;
int ledPin=13;
int sensorValue=0;
```

```

const int etatOn = 500;
const int etatOff = 500;

void setup(){
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}

void loop(){
  sensorValue=analogRead(sensorPin);
  Serial.print(A0);
  if (sensorValue>700){
    digitalWrite(ledPin,HIGH);
  }

  else{
    digitalWrite(ledPin,LOW);
  }
}

```

09/02/24 : Travail sur le montage Arduino et le programme associé (étude de la plage de réponse du capteur).

On a testé le capteur dans un environnement silencieux en essayant de réduire au maximum les vibrations (dans une boîte en polystyrène), mais les réponses du capteur ne changent pas. Il semblerait que le capteur sature, ou plus probablement que le programme utilisé n'est pas adapté à ce capteur. Des recherches sur Internet ne donnent rien quant à d'autres projets utilisant ce capteur menés à terme par d'autres personnes.

01/03/24 : Travail sur le montage Arduino et le programme associé

Code utilisé :

```

const int PIEZO_PIN = A0; // Sortie du piezo
int previousPiezoADC = 0; // Stocker la valeur précédente

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur ADC actuelle du piezo
  int piezoADC = analogRead(PIEZO_PIN);

```

```
// Calculer la différence entre la valeur actuelle et la précédente
int diff = piezoADC - previousPiezoADC;

// Mettre à jour la valeur précédente pour la prochaine itération
previousPiezoADC = piezoADC;

// Affichage de la différence
Serial.println(diff);

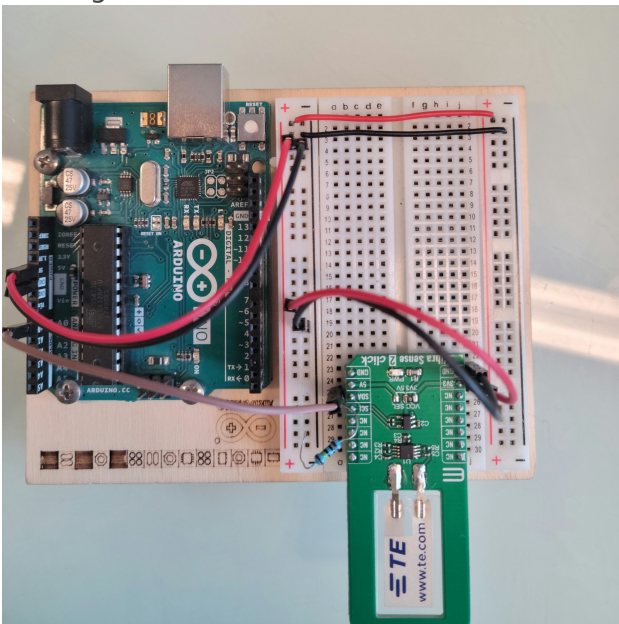
// Attente avant la prochaine lecture
delay(5);

}
```

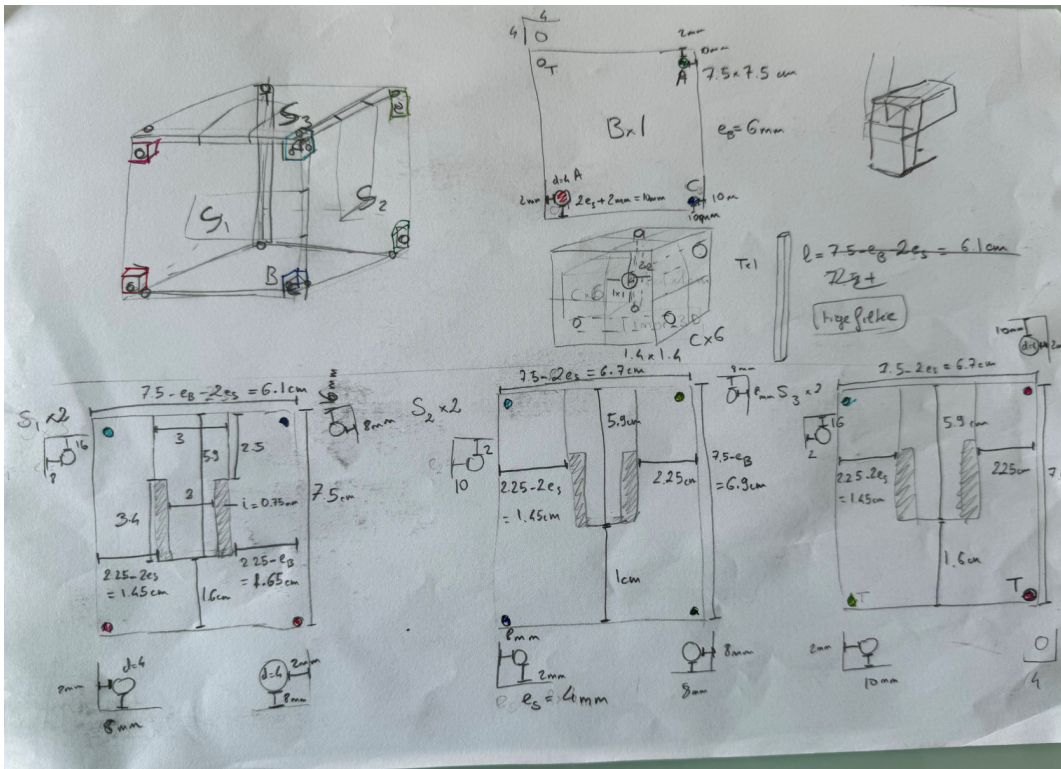
Les résultats obtenus varient selon la valeur de la résistance utilisée. Il faut donc calibrer le montage, c'est-à-dire choisir la valeur de résistance permettant de ne pas capter les vibrations parasites de l'air mais de capter celles des "séismes". Il faut ensuite convertir les valeurs de différentiel de tension obtenu en $m.s^{-2}$, ceci au moyen de l'application Phyphox, développée par l'Université Aachen d'Aix-en-Provence, qui peut servir d'accéléromètre grâce aux capteurs des smartphones.

La résistance la plus adaptée est celle de 100 kohm.

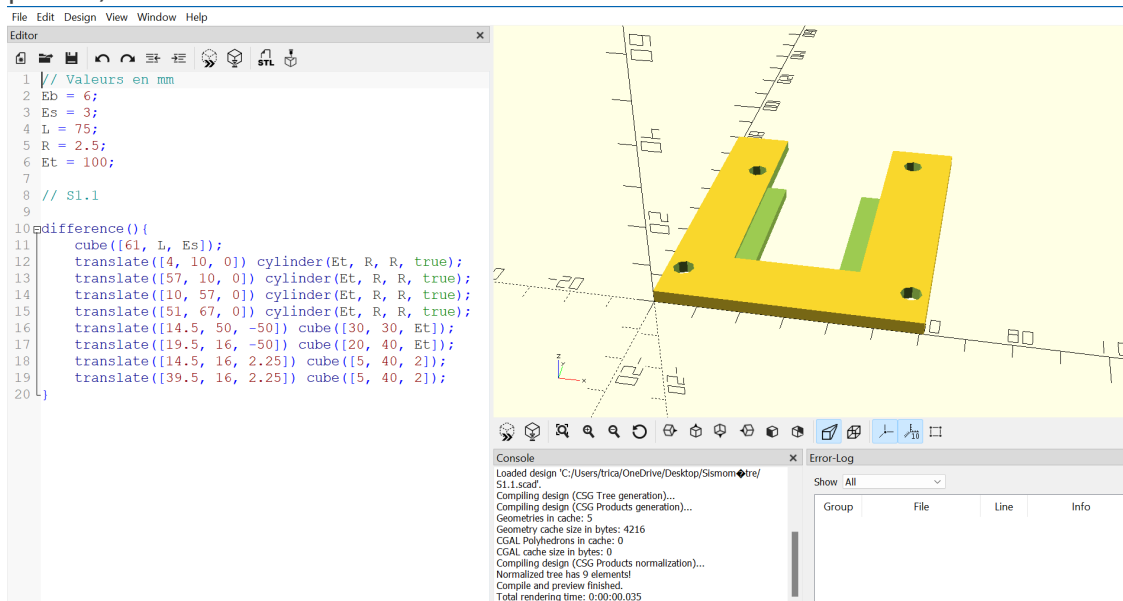
Montage :



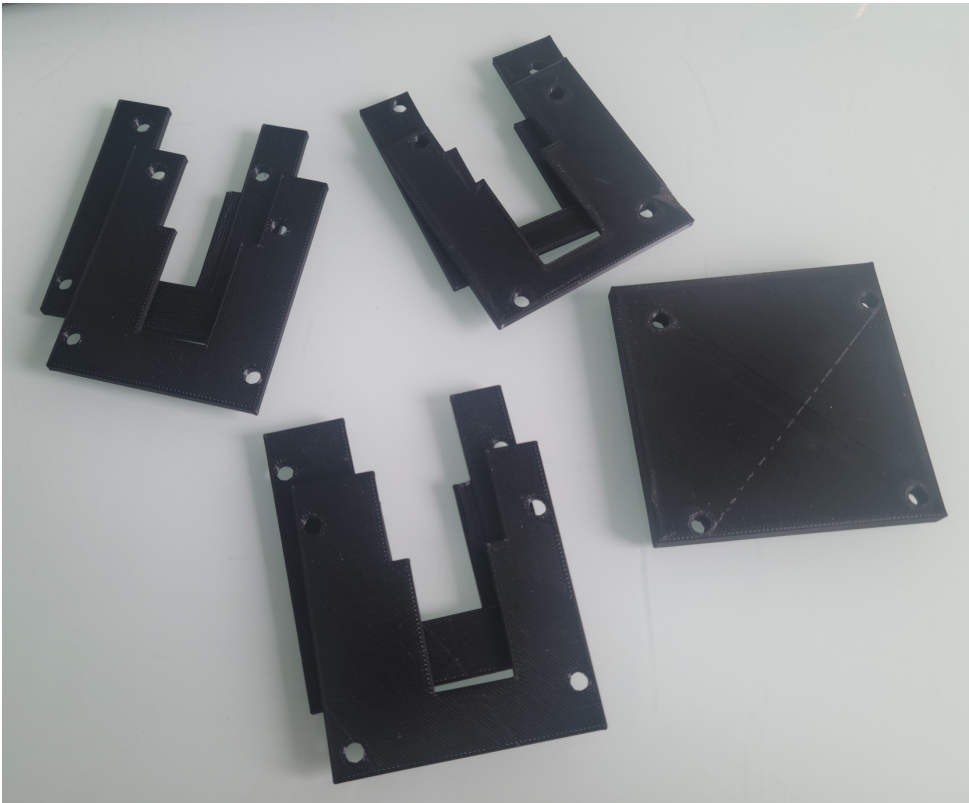
08/03/24 : Nous avons réalisé un plan pour le support du sismomètre, le résultat final est un cube avec 2 faces ouvertes, composé de 4 faces en plexiglas supportant les différents éléments du montage et reliés par des coins vissés :



15/03/25 : Nous avons modélisé les différentes pièces du sismomètre sur OpenScad et lancé la fabrication des pièces (mais le site FabLab ayant été fermé, nous n'avons pas pu faire toutes les pièces).



22/03/24 : Nous avons pu imprimer les pièces et avons commencé à prévoir le montage du sismomètre en commandant un troisième capteur. Nous avons aussi modifié le code du capteur afin d'améliorer le graphique obtenu.



```
const int PIEZO_PIN = A0; // Sortie du piezo
float previousPiezoADC = 0; // Stocker la valeur précédente

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur ADC actuelle du piezo
  float piezoADC = analogRead(PIEZO_PIN);

  // Calculer la différence entre la valeur actuelle et la précédente
  float diff = piezoADC - previousPiezoADC;

  // Mettre à jour la valeur précédente pour la prochaine itération
  previousPiezoADC = piezoADC;

  // Affichage de la différence
  Serial.println(diff);

  Serial.print(-50); // To freeze the lower limit
  Serial.print(" ");
  Serial.print(50); // To freeze the upper limit
  Serial.print(" ");

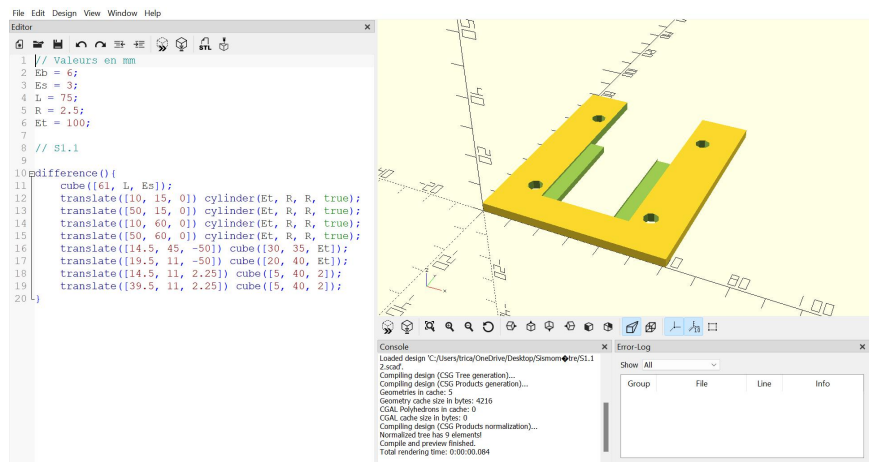
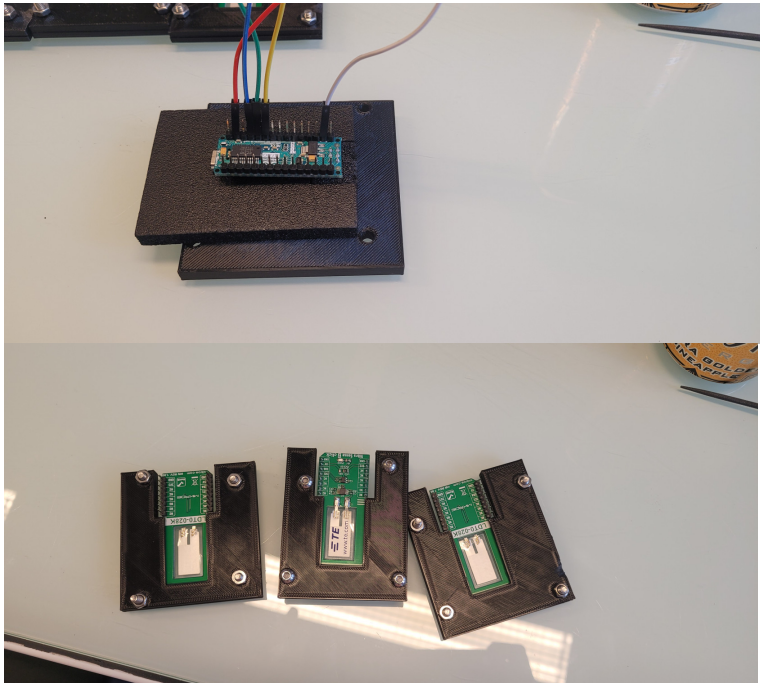
  // Attente avant la prochaine lecture
```



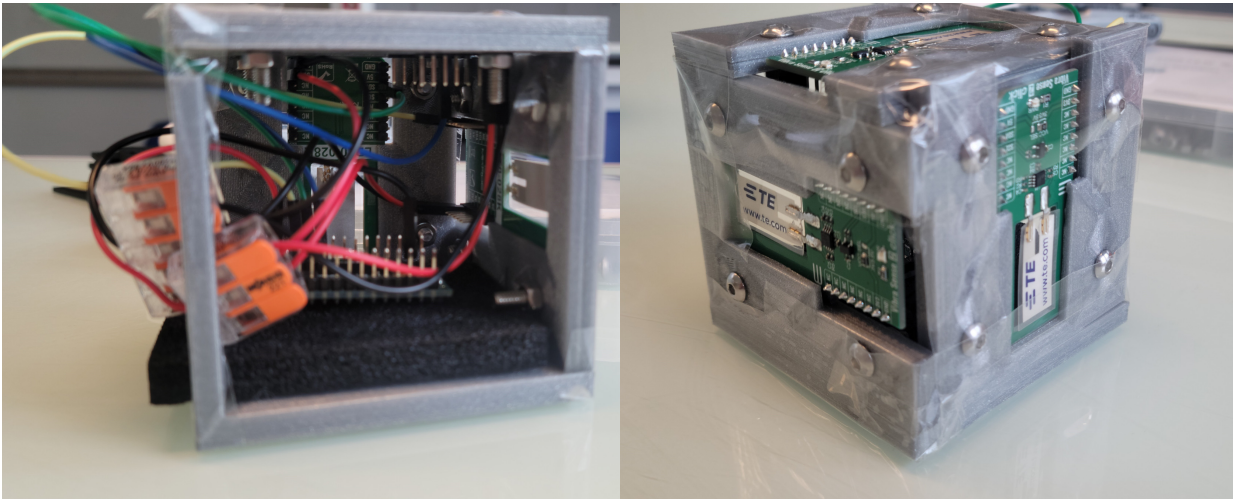
```
delay(5);
```

```
}
```

29/03/24 : Nous avons assemblé les faces du sismomètre, mais avons rencontré un problème dans l'assemblage des faces, les trous pour les vis dans les cubes utilisés pour l'assemblage étaient trop petits. Nous avons donc réimaginé les pièces et le mode d'assemblage.



05/04/24 : Nous avons assemblé sommairement les parties du sismomètre. Certaines mesures sont fausses, et l'assemblage est un peu compliqué, nous allons donc modifier une dernière fois les pièces du sismomètre. Nous avons aussi créé le code pour le sismomètre avec ses trois capteurs, mais avons eu du mal à le lancer.



```
const int PIEZO1_PIN = A0; // Sortie du piezo 1
const int PIEZO2_PIN = A1; // Sortie du piezo 2
const int PIEZO3_PIN = A2; // Sortie du piezo 3
float previousPiezo1ADC = 0; // Stocker la valeur précédente
float previousPiezo2ADC = 0; // Stocker la valeur précédente
float previousPiezo3ADC = 0; // Stocker la valeur précédente

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur ADC actuelle du piezo
  float piezo1ADC = analogRead(PIEZO1_PIN);
  float piezo2ADC = analogRead(PIEZO2_PIN);
  float piezo3ADC = analogRead(PIEZO3_PIN);

  // Calculer la différence entre la valeur actuelle et la précédente
  float diff1 = piezo1ADC - previousPiezo1ADC;
  float diff2 = piezo2ADC - previousPiezo2ADC;
  float diff3 = piezo3ADC - previousPiezo3ADC;

  // Mettre à jour la valeur précédente pour la prochaine itération
  previousPiezo1ADC = piezo1ADC;
  previousPiezo2ADC = piezo2ADC;
  previousPiezo3ADC = piezo3ADC;

  // Affichage de la différence
  Serial.println(diff1);
  Serial.println(diff2);
```



```
Serial.println(diff3);
```

```
Serial.print(-50); // To freeze the lower limit
```

```
Serial.print(" ");
```

```
Serial.print(50); // To freeze the upper limit
```

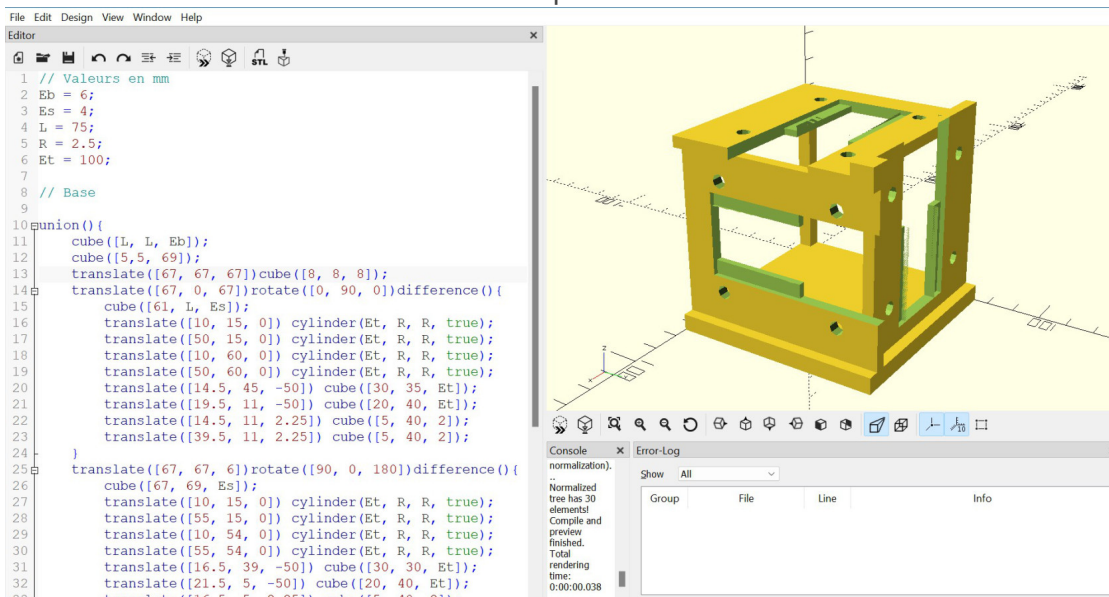
```
Serial.print(" ");
```

```
// Attente avant la prochaine lecture
```

```
delay(5);
```

```
}
```

26/04/24 : Nous avons lancé l'impression de la nouvelle version du sismomètre et avons modifié le code afin d'afficher les valeurs des trois capteurs, mais malheureusement, la sensibilité n'est pas adaptée, il faut probablement ajouter des résistances. Nous avons aussi dû utiliser une carte Arduino Uno au lieu d'une Nano, mais elle est déjà attachée à un montage avec une breadbox, il nous faudra donc en utiliser une autre pour le sismomètre final.



```
const int PIEZO1_PIN = A0; // Sortie du piezo 1
```

```
const int PIEZO2_PIN = A1; // Sortie du piezo 2
```

```
const int PIEZO3_PIN = A2; // Sortie du piezo 3
```

```
float previousPiezo1ADC = 0; // Stocker la valeur précédente
```

```
float previousPiezo2ADC = 0; // Stocker la valeur précédente
```

```
float previousPiezo3ADC = 0; // Stocker la valeur précédente
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    // Lecture de la valeur ADC actuelle du piezo
```

```
    float piezo1ADC = analogRead(PIEZO1_PIN);
```

```
    float piezo2ADC = analogRead(PIEZO2_PIN);
```

```

float piezo3ADC = analogRead(PIEZO3_PIN);

// Calculer la différence entre la valeur actuelle et la précédente
float diff1 = piezo1ADC - previousPiezo1ADC;
float diff2 = piezo2ADC - previousPiezo2ADC;
float diff3 = piezo3ADC - previousPiezo3ADC;

// Mettre à jour la valeur précédente pour la prochaine itération
previousPiezo1ADC = piezo1ADC;
previousPiezo2ADC = piezo2ADC;
previousPiezo3ADC = piezo3ADC;

// Affichage de la différence
Serial.print(diff1); Serial.print(",");
Serial.print(diff2); Serial.print(",");
Serial.print(diff3); Serial.print(",");

Serial.print(-50); Serial.print(","); // To freeze the lower limit
Serial.print(50); Serial.print(","); // To freeze the upper limit
Serial.println();

// Attente avant la prochaine lecture
delay(5);

}

```

03/05/24 : Nous avons imprimé et monté la nouvelle version du sismomètre. Malheureusement, suite à un défaut d'impression, la fente supposée maintenir les capteurs est trop large et les capteurs sont légèrement mobiles. Nous avons donc réimprimé la pièce principale. Nous avons aussi transformé trois résistances de 100k ohms (valeur utilisée dans le montage du 01/03/24) en fils pour le montage électronique.

Entre le 10/05/24 et le 24/05/24 : Nous avons imprimé la version finale du support du sismomètre et finalisé le montage. Le sismomètre fonctionne, mais il faut encore le calibrer, ajouter un algorithme pour calculer la magnitude des séismes enregistrés, et changer le delay, car le sismomètre a actuellement un fonctionnement de filtre passe-bas 200Hz. Il faut trouver un moyen d'améliorer le graphe obtenu.

Revision #17

Created 26 January 2024 12:58:22 by Tricart Elie

Updated 27 May 2024 18:05:56 by Tricart Elie