

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la  
Recherche Scientifique

Université Mohammed Seddik BENYAHIA – Jijel –



Faculté des Sciences et de la Technologie  
Département d'Electronique

---

## MEMOIRE DE FIN D'ETUDES

---

Présenté en vue de l'obtention du diplôme  
de **MASTER** en Electronique

Option : **Electronique des Systèmes Embarqués**

Thème :

Simulation d'un détecteur de  
tremblement de terre

Encadré par :

**Pr. Abdelmalek  
Bouhadjera**

Réalisé par :

**Mr. Bourouissa Yasser**

Année Universitaire  
2020/2021

# REMERCIEMENTS

*J'offre ma grande gratitude à Dieu qui m'a aidé à faire ce travail. J'exprime ma profonde gratitude à mes parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.*

*Je remercie mon encadreur Mr Abdelmalek Bouhadjera pour les efforts qu'il a déployé, pour m'aider, conseiller, encourager et corriger. Je voudrais remercier les membres de jury d'avoir accepté d'examiner mon travail.*

*Je remercie aussi tout le corps enseignant dans le département d'Electronique qui a contribué à ma formation universitaire. Sans oublier tous mes amis, Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici ma sincère reconnaissance.*

# *DEDICACES*

*Je dédie ce modeste travail :*

*A ma plus belle étoile qui puisse exister dans l'univers ma chère mère « Houria ».*

*A mon meilleur ami : mon père, le plus beau et bon de tous les pères « Chaabane ».*

*Mes frères : mohammed ; ibrahim.*

*Mes sœurs : nafissa ; hadjer ; selma.*

*A mon amie d'enfance: Abd Elnnour ; Wassime ; Omar.*

*A mes tous amis de l'université.*

# Sommaire

|  |    |
|--|----|
| <b>Introduction générale</b> .....                 | 1  |
| <b>Chapitre 1 : Généralités sur la sismologie.</b> |    |
| 1.1 Introduction.....                              | 3  |
| 1.2 Cause des séismes.....                         | 4  |
| 1.2.1 Tectonique des plaques.....                  | 4  |
| 1.2.2 Autres causes de séismes.....                | 5  |
| 1.3 Caractéristiques principales .....             | 6  |
| 1.3.1 L'hypocentre ou foyer .....                  | 6  |
| 1.3.2 L'épicentre.....                             | 6  |
| 1.3.3 La magnitude (M).....                        | 6  |
| 1.3.4 L'intensité (I).....                         | 8  |
| 1.4 Type d'ondes sismiques.....                    | 10 |
| 1.4.1 Ondes primaires « P ».....                   | 10 |
| 1.4.2 Ondes transversales « S ».....               | 10 |
| 1.4.3 Ondes de surface .....                       | 11 |
| 1.4.3.1 Ondes de Rayleigh ou ondes R.....          | 11 |
| 1.4.3.2 Ondes de Love .....                        | 11 |
| 1.5 Les failles .....                              | 12 |
| 1.5.1 Type des failles .....                       | 12 |
| 1.5.1.1 Faille normal .....                        | 12 |
| 1.5.1.2 Faille inverse .....                       | 12 |

|         |  |    |
|---------|--|----|
| 1.5.1.3 | Faille décrochant .....                        | 12 |
| 1.6     | Classification des séismes .....               | 13 |
| 1.6.1   | Les séismes superficiels .....                 | 13 |
| 1.6.2   | Les séismes intermédiaires .....               | 13 |
| 1.6.3   | Les séismes profonds .....                     | 13 |
| 1.7     | La localisation des séismes .....              | 13 |
| 1.7.1   | Les séismes inter-plaque .....                 | 13 |
| 1.7.2   | Les séismes intra-plaque .....                 | 13 |
| 1.7.3   | Les séismes liés à l'activité volcanique ..... | 14 |
| 1.7.4   | Les séismes liés à l'activité humaine.....     | 14 |
| 1.8     | La structure interne de la Terre.....          | 14 |
| 1.9     | Enregistrement des séismes.....                | 15 |
| 1.10    | Action sismique.....                           | 15 |
| 1.11    | Conclusion.....                                | 16 |

## **Chapitre 2 : Description du matériel et logiciel du projet.**

|         |  |    |
|---------|--|----|
| 2.1     | Introduction.....  | 17 |
| 2.2     | Étude de la partie matérielle.....                         | 17 |
| 2.2.1   | Carte Arduino .....  | 17 |
| 2.2.1.1 | Carte Arduino UNO .....                                    | 18 |
| 2.2.1.2 | Avantages de la carte Arduino UNO .....                    | 18 |
| 2.2.1.3 | Caractéristiques technique de la carte Arduino UNO ...     | 19 |
| 2.2.1.4 | Brochage de la carte Uno .....                             | 20 |
| 2.2.1.5 | Différents éléments de la carte Arduino Uno .....          | 21 |
| 2.2.2   | Accéléromètre ADXL335 .....                                | 26 |
| 2.2.2.1 | Caractéristiques et spécification Accéléromètre ADXL335... | 27 |
| 2.2.2.2 | Broches de connexion .....                                 | 28 |
| 2.2.2.3 | Fonctionnement de l'accéléromètre ADXL335 .....            | 29 |

|         |  |    |
|---------|--|----|
| 2.2.3   | L'afficheur LCD .....  | 30 |
| 2.2.3.1 | L'afficheur utilisé dispose de .....                               | 31 |
| 2.2.3.2 | Rôle des différentes broches de l'afficheur LCD .....              | 32 |
| 2.2.3.3 | Brochage de l'afficheur LCD .....                                  | 32 |
| 2.2.3.4 | Principe des cristaux liquides .....                               | 34 |
| 2.2.3.5 | Le fonctionnement l'afficheur LCD .....                            | 34 |
| 2.2.4   | LED.....   | 35 |
| 2.2.4.1 | Fonctionnement de LED .....  | 36 |
| 2.2.4.2 | Les caractéristiques lumineuses des LED .....                      | 36 |
| 2.2.5   | Buzzer .....   | 37 |
| 2.2.5.1 | Le Buzzer est Caractérisé par .....                                | 37 |
| 2.3     | Étude de la partie logicielle .....                                | 38 |
| 2.3.1   | Arduino IDE .....  | 38 |
| 2.3.1.1 | Présentation du logiciel .....                                     | 39 |
| 2.3.1.2 | Interface de programmation Arduino .....                           | 41 |
| 2.3.1.3 | Structure d'un programme Arduino .....                             | 43 |
| 2.3.2   | Logiciel Proteus ISIS.....   | 43 |
| 2.3.3   | Le software utilisée "Processing" .....                            | 44 |
| 2.3.3.1 | Les bases de l'interface de Processing .....                       | 45 |
| 2.3.3.2 | Communication entre la carte Arduino & logiciel<br>Processing..... | 47 |
| 2.4     | Conclusion.....  | 48 |

## **Chapitre 3 : Simulations et tests pratiques.**

|  |           |
|--|-----------|
| 3.1 Introduction .....                                   | 49        |
| 3.2 Description du schéma fonctionnel .....              | 49        |
| 3.3 Déroulement du projet .....                          | 50        |
| 3.4 Simulation sur le logiciel Proteus .....             | 50        |
| 3.4.1 Présentation .....                                 | 50        |
| 3.4.2 Démarche de la simulation .....                    | 50        |
| 3.4.2.1 Les bibliothèques nécessaires pour Proteus ..... | 50        |
| 3.4.3 Circuit de simulation sous Proteus .....           | 52        |
| 3.5 Réalisation du projet .....                          | 54        |
| 3.5.1 Composants utilisés .....                          | 54        |
| 3.5.2 Alimentation et connexion des broches .....        | 56        |
| 3.5.3 Explication de la programmation .....              | 56        |
| 3.5.4 Principe de fonctionnement.....                    | 57        |
| 3.5.5 Résultats et discussion .....                      | 59        |
| 3.6 Conclusion.....                                      | 60        |
| <br>   |           |
| <b>Conclusion générale.....</b>                          | <b>61</b> |
| <b>Bibliographie.....</b>                                | <b>62</b> |
| <b>Annexe .....</b>                                      | <b>65</b> |

# Liste des figures

|   |    |
|---|----|
| <b>Figure 1.1 :</b> Mouvement des plaques tectoniques.....                | 5  |
| <b>Figure 1.2 :</b> Caractéristiques d'un séisme.....                     | 8  |
| <b>Figure 1.3 :</b> Onde de type P.....                                   | 10 |
| <b>Figure 1.4 :</b> Onde de type S.....                                   | 10 |
| <b>Figure 1.5 :</b> Onde de Rayleigh.....                                 | 11 |
| <b>Figure 1.6 :</b> Ondes de surface (l'onde de love).....                | 11 |
| <b>Figure 1.7 :</b> Les types des failles sismiques.....                  | 12 |
| <b>Figure 1.8 :</b> La structure interne de la terre.....                 | 15 |
| <b>Figure 1.9 :</b> Décomposition de l'action sismique.....               | 16 |
| <b>Figure 2.1 :</b> La carte Arduino Uno.....                             | 18 |
| <b>Figure 2.2 :</b> Brochage de la carte Arduino Uno.....                 | 20 |
| <b>Figure 2.3 :</b> Les interfaces d'E/S de la carte Arduino Uno.....     | 21 |
| <b>Figure 2.4:</b> Microcontrôleur ATmega328.....                         | 21 |
| <b>Figure 2.5:</b> Microcontrôleur ATmega328 (type classique.....         | 22 |
| <b>Figure 2.6:</b> Sources de l'Alimentation de la carte Arduino UNO..... | 25 |
| <b>Figure 2.7:</b> Accéléromètre ADXL335 Module GY-61.....                | 26 |
| <b>Figure 2.8:</b> Broches de connexion Accéléromètre ADXL335 .....       | 28 |
| <b>Figure 2.9:</b> Accéléromètre effet piézoélectrique.....               | 30 |
| <b>Figure 2.10:</b> L'afficheur LCD16 x 2.....                            | 31 |
| <b>Figure 2.11:</b> Brochage de l'afficheur LCD.....                      | 33 |
| <b>Figure 2.12:</b> Symbole de la LED.....                                | 35 |
| <b>Figure 2.13:</b> Fonctionnement de LED .....                           | 36 |
| <b>Figure 2.14:</b> magnétique buzzer.....                                | 37 |
| <b>Figure 2.15:</b> L'interface de l'IDE Arduino.....                     | 38 |



## Liste des figures

|  |    |
|--|----|
| <b>Figure 2.16:</b> L'interface de l'IDE Arduino en détail.....  | 39 |
| <b>Figure 2.17:</b> La barre d'outils.....   | 40 |
| <b>Figure 2.18:</b> Moniteur série de L'Arduino IDE.....   | 41 |
| <b>Figure 2.19:</b> Fenêtre de l'IDE.....  | 42 |
| <b>Figure 2.20:</b> Les étapes de base du programme.....   | 42 |
| <b>Figure 2.21:</b> Simulation sur logiciel Proteus.....   | 43 |
| <b>Figure 2.22:</b> Logo du Processing .....   | 44 |
| <b>Figure 2.23:</b> l'interface de Processing .....  | 45 |
| <b>Figure 2.24:</b> La fenêtre « Préférence » de Processing.....   | 46 |
| <b>Figure 2.25:</b> Communication entre Arduino & Processing.....  | 47 |
| <b>Figure 2.26 :</b> Communication entre PROTEUS et PROCESSING.....  | 48 |
| <b>Figure 3.1 :</b> Schéma fonctionnel du travail proposé.....   | 49 |
| <b>Figure 3.2 :</b> Recherche d'articles Pot-HG par Proteus .....  | 51 |
| <b>Figure 3.3 :</b> Accéléromètre a par Proteus .....  | 51 |
| <b>Figure 3.4 :</b> Circuit électronique d'une pratique .....  | 52 |
| <b>Figure 3.5 :</b> L'exécution de la pratique a par Proteus .....   | 53 |
| <b>Figure 3.6 :</b> L'exécution de la pratique a par Proteus .....   | 53 |
| <b>Figure 3.7 :</b> Projet de travail d'écriture de mots ("EarthQuake" "Detector") sur l'écran LCD.....            | 54 |
| <b>Figure 3.8:</b> Projet de travail d'écriture de mots ("Calibrating....." "Please wait...") sur l'écran LCD..... | 55 |
| <b>Figure3.9:</b> Projet de travail d'écriture de mots ("X Y Z") sur l'écran LCD.....                              | 55 |
| <b>Figure 3.10 :</b> Schéma de circuit et connexions.....  | 56 |
| <b>Figure 3.11 :</b> Projet de travail en sommeil.....   | 57 |

## Liste des figures

|   |    |
|---|----|
| <b>Figure 3.12</b> : Graphique à travers les axes X, Y et Z à travers le système de traitement (au repos).....            | 58 |
| <b>Figure 3.13</b> : Travail de projet en mouvement (vibration).....  | 58 |
| <b>Figure 3.14</b> : Graphique à travers les axes X, Y et Z à travers le système de traitement (en cas de mouvement)..... | 59 |

### Liste des tableaux

|  |    |
|--|----|
| <b>Tableau 1.1 :</b> Energie libérée pour les différentes magnitudes de Richter..... | 7  |
| <b>Tableau 1.2 :</b> Echelle d'intensité de Mercalli.....                            | 8  |
| <b>Tableau 2.1:</b> Caractéristiques d'une carte Arduino Uno.....                    | 19 |
| <b>Tableau 2.2:</b> Caractéristiques et spécification Accéléromètre ADXL335.....     | 27 |
| <b>Tableau 2.3:</b> la description des broches de LCD.....                           | 32 |
| <b>Tableau 3.1 :</b> les différents composants utilisés dans la simulation.....      | 52 |

### Liste des abréviations

|                |   |
|----------------|---|
| <b>USB</b>     | Universal Serial Bus                                |
| <b>IDE</b>     | Integrated development environment                  |
| <b>ICSP</b>    | In-Circuit Serial Programming                       |
| <b>I2C</b>     | Integrated - Integrated Circuit                     |
| <b>CA / CC</b> | Courant Alternatif / Courant Continu                |
| <b>LED</b>     | Light-Emitting Diode                                |
| <b>RX</b>      | Reception   |
| <b>TX</b>      | Transmission  |
| <b>LCD</b>     | Liquid Crystal Display                              |
| <b>SRAM</b>    | Static random-access memory                         |
| <b>CAO</b>     | Conception Assisté par Ordinateur                   |
| <b>E/S</b>     | Entrées/Sorties                                     |
| <b>EEPROM</b>  | Electrically-erasable programmable read-only memory |
| <b>TTL</b>     | Time To Live  |
| <b>SPI</b>     | Serial Peripheral Interface                         |
| <b>TWI</b>     | Two-Wire Interface.                                 |
| <b>R/W</b>     | Read/Write  |
| <b>GND</b>     | Ground  |

**Introduction**

**Générale**

### Introduction générale

Un tremblement de terre est le tremblement de la surface de la Terre, résultant de la libération soudaine d'énergie dans la lithosphère terrestre qui crée des ondes sismiques. Un tremblement de terre est ce qui se produit lorsque deux blocs de terre glissent soudainement l'un sur l'autre. La surface où ils glissent est appelée faille ou plan de faille. L'emplacement sous la surface de la terre où le séisme commence s'appelle l'hypocentre, et l'emplacement directement au-dessus de lui à la surface de la terre s'appelle l'épicentre [1]. Pendant le séisme, le degré des dommages causés est basé sur la magnitude qui indique la quantité d'énergie libérée par la croûte terrestre. La magnitude du séisme qui est inférieure à 5 est mesurée à l'aide d'une échelle de magnitude locale appelée échelle de magnitude de Richter. Il mesure la magnitude du séisme en observant l'amplitude sur un sismogramme. Ces dernières années, une échelle de magnitude standard est utilisée qui représente l'énergie libérée au moment du tremblement de terre en incluant plus précisément les événements à grande échelle. Cette technique utilise des appareils tels que le sismomètre, le géophone et l'accéléromètre. En attendant, avant de sélectionner un capteur sismique, nous devons savoir que le capteur sismique doit fournir des signaux qui ne sont pas affectés par les caractéristiques inhérentes aux capteurs et reflètent aussi fidèlement que possible la véritable réponse du sol à l'onde source sismique qui le traverse. En termes de réponse en fréquence du récepteur, sa sortie doit être constante pour toutes les fréquences d'entrée. De plus, la phase de la fréquence d'entrée ne doit pas être affectée afin que la forme de l'onde ne change pas. D'une manière générale, il est souhaitable d'avoir un capteur sismique avec un temps de réponse rapide et un temps d'établissement faible [2]. En ce qui concerne la sélection d'un capteur sismique, les géophones ont un temps de pointe et un temps de stabilisation plus longs que l'accéléromètre, mais les accéléromètres sont sélectionnés pour les activités sismiques en raison de leur faible bruit, de leurs temps de réponse rapides et de leurs bandes passantes élevées par rapport aux géophones.

Le problème dont nous font face est le prix de ces appareils qui est chère est pas facile à obtenir.

Le but de mon travail consiste à réaliser un sismographe à base d'un arduino uno, et un accéléromètre et qui affiche en temps réel et enregistre les données obtenues.

## Introduction générale

Dans le premier chapitre, j'ai parlé des généralités sur la sismologie, de sorte que j'ai donc expliqué les causes les plus importantes des séismes et mentionné les caractéristiques et les types les plus importants, ainsi que les classifications et quelques autres j'ai accompagné l'explication de ces généralités des tableaux, des fugueurs et de courbes pour simplifier et souligner l'idée.

Quant au le deuxième chapitre je suis entré dans la description du matériel et du logiciel et ce à travers deux études, c'était la première étape étude de la partie matérielle, J'ai mentionné les dispositifs les plus importants, qui étaient carte Arduino et je ne me suis pas limité à cela, mais j'ai abordé ses caractéristiques et ses avantages, avec le support de chaque élément avec de figure et de tableaux , aussi les différents éléments de la carte Arduino Uno Deuxième étape c'est Accélération ADX335 c'est le même travail que la première étape en plus de cela sont des fonctionnement de l'accélération ADX335, ainsi que les élément l'afficheur LCD, LED , aussi Buzzer. Dans chaque élément il y a des caractéristiques, de fonctionnement, des principes

Dans le deuxième partie c'est étude de la logicielle, j'ai mentionné Arduino IDE Logiciel proteus ISIS, aussi le soflawer utilisé «processing » et utilisé des figures et des tableaux pour confirmer les informations. Finale j'ai représenté petite conclusion en résumé du chapitre.

Dans troisième et chapitre et dernier sujet relève du titre simulations est testes pratiques, nous effectuons la simulation via la logicielle proteus isis, via les appareils électroniques (arduino, accéléromètre etc...) situés dans la bibliothèque du programme proteus, puis nous effectuons immédiatement les travaux pratiques via le programme processing.

**Chapitre 1 :**  
**Généralités sur la**  
**sismologie**



## 1.1 Introduction

Un séisme est une vibration du sol transmise aux bâtiments, causée par le frottement et le choc brutal, en profondeur, de plaques rocheuses coulissant entre elles le long de failles de l'écorce terrestre, et libérant de ce fait une énergie considérable. Les séismes sont, avec le volcanisme, l'une des manifestations de la tectonique des plaques. L'activité sismique est concentrée le long de failles, en général à proximité des frontières entre ces plaques. Lorsque les frottements au niveau d'une de ces failles sont importants, le mouvement entre les deux plaques est bloqué. De l'énergie est alors stockée le long de la faille. La libération brutale de cette énergie permet de rattraper le retard du mouvement des plaques. Le déplacement instantané qui en résulte est la cause des séismes. Après la secousse principale, il y a des répliques, parfois meurtrières, qui correspondent à des petits réajustements des blocs au voisinage de la faille.

Le tremblement de terre est la catastrophe naturelle la plus meurtrière et la plus destructrice. Chaque année plus de cent mille séismes, dont la magnitude sur l'échelle de Richter est comprise entre 3 à 4, et un à deux séisme de magnitude supérieur à 8 secouent la terre.

Les risques font l'objet d'études depuis de nombreuses années. Pendant longtemps, ils ont été considérés sous l'angle du phénomène dangereux, et ce n'est qu'après quelques décennies que des chercheurs se sont davantage interrogés sur les éléments exposés. Si l'on considère le risque comme le résultat de la combinaison entre un aléa et des éléments vulnérables, il est important de l'aborder dans une démarche intégrée prenant en compte ces deux composantes.

A cause de la multiplicité des études sur les risques, on a souhaité de faire un travail qui regroupe trois exemples sur les risques sismiques :

- Risque sismique et développement
- Vulnérabilité sismique des constructions
- Les barrages et le risque sismique

## 1.2 Cause des séismes

### 1.2.1 Tectonique des plaques

La compréhension du mécanisme responsable de l'activité sismique du globe terrestre est récente, puisque ce n'est qu'en 1968 que J. Morgan, D. McKenzie et X. Le Pichon a formulé la théorie de la tectonique des plaques qui fournit un modèle cinématique cohérent des déformations de l'écorce terrestre. Le moteur de ces déformations est (**Figure1.1**) l'expansion des fonds océaniques (proposée en 1960 par H. Hess après l'échec d'autres tentatives d'explication de la « dérive des continents », imaginée en 1915 Par Wegener) ; cette expansion, qui peut atteindre 170 mm/an pour les dorsales les plus actives, pousse les unes contre les autres, les différentes plaques rigides (une douzaine au total (**Figure1.1**) qui constituent l'écorce terrestre. Plusieurs types de mouvements peuvent résulter de ces affrontements entre plaques :

- **la subduction** : c'est-à-dire la plongée d'une plaque sous une autre ce qui permet de compenser l'augmentation de surface plaque Nazca sous l'Amérique du Sud, ou de la plaque Philippines sous l'Eurasie au niveau du Japon.
- **le décrochement** : c'est-à-dire le coulissage horizontal d'une plaque contre une autre, dont l'exemple le plus connu est la célèbre faille de San Andreas en Californie (contact entre les plaques Pacifique et Amérique du Nord) ;
- **la compression** : c'est-à-dire la collision frontale sans subduction qui se traduit par la formation de chaînes de montagnes, comme l'Himalaya, résultat de la collision des plaques Inde et Eurasie.

Ces mouvements relatifs entre plaques ne se font pas, en général, de manière progressive, mais par à-coups et chacun de ces à-coups constitue un séisme, plus ou moins intense suivant l'amplitude et la rapidité du mouvement, ainsi que l'étendue de la zone concernée [3].

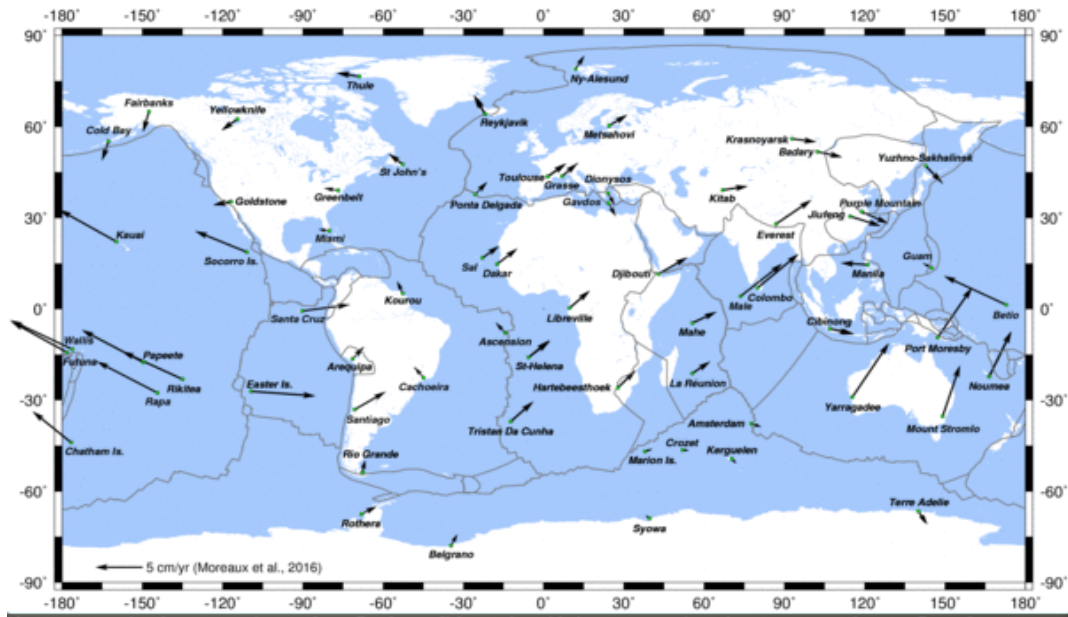


Figure 1.1 : Mouvement des plaques tectoniques.

Cette théorie de la tectonique des plaques est maintenant bien établie et fournit une explication immédiate des séismes qui se produisent au voisinage des limites des plaques (séismes inter plaques). Si ce type de séisme représente effectivement la plus grande partie de l'activité sismique, qui est donc concentrée dans certaines zones bien définies, nous observons aussi des séismes, moins nombreux mais pouvant être violents, à l'intérieur de certaines plaques.

Cette sismicité inter plaque, plus diffuse et plus difficile à prévoir, il résulte de l'état de contrainte qui règne à l'intérieur des plaques du fait de leurs interactions mutuelles ; l'hypothèse des plaques parfaitement rigides ne constitue en effet qu'une première approximation, et celles-ci sont en réalité susceptibles de subir des ruptures locales sous l'effet des champs de contraintes qui agissent sur elles.

### 1.2.2 Autres causes de séismes

Presque toute l'activité sismique correspond à des séismes tectoniques ; ils existent Mais d'autres types de séismes, naturels ou artificiels, dont les suivants :

- **séismes volcaniques**, associés à la montée du magma, pendant les éruptions ces séismes sont généralement faibles et localisés au voisinage des volcans actifs.
- **séismes artificiels** résultant des explosions souterraines (tirs de carrières et de mines, essais nucléaires) ou de l'exploitation des mines.

- **séismes dits induits**, souvent associés à un apport massif d'eau dans les terrains (mise en eau d'un grand barrage ou injection à grande échelle dans le sol) ; ces séismes induits, dont on connaît quelques exemples, paraissent devoir être attribués à la diminution de résistance à la rupture des roches lors de l'infiltration d'eau dans leur réseau de micro fractures [4].

### 1.3 Caractéristiques principales

Un séisme est caractérisé par :

#### 1.3.1 L'hypocentre ou foyer

Le point d'origine d'un séisme, il peut se trouver entre la surface et jusqu'à 700 kilomètres de profondeur (limite du manteau supérieur) pour les événements les plus profonds.

C'est la région de la faille où se produit la rupture et d'où partent les ondes sismiques.

#### 1.3.2 L'épicentre

C'est le point situé, en surface, à la verticale du foyer, et où l'intensité est la plus grande, et donc le point le plus proche.

Il existe trois classes de séismes, selon la profondeur de champ :

**a) les séismes superficiels** : qui se produisent à une faible profondeur, soit dans les premières dizaines de kilomètre (<60km), et qui se trouvent autant aux frontières divergentes.

**b) Les séismes intermédiaires** : qui se produisent entre quelques dizaines de kilomètres de profondeur (60 à 300km), et se concentrent uniquement au voisinage des limites convergentes.

**c) Les séismes profonds** : qui se produisent à des profondeurs pouvant atteindre les 700km, soit en pratique la base de l'asthénosphère, ce sont des séismes de grande intensité, ils surviennent au niveau des zones de subduction et se sont les plus destructeurs.

#### 1.3.3 La magnitude (M)

La puissance d'un tremblement de terre correspond à l'énergie libérée pendant ce séisme. La puissance est exprimée grâce à une échelle des magnitudes créée par le géophysicien américain Charles Richter en 1935 : « l'échelle de Richter ». C'est l'échelle la plus utilisée par les géologues. Cette échelle donne 8.9 (degré9) pour les grands tremblements de terre.

$$M = \log A + B \log \Delta + C \quad (1.1)$$

$A$  : Amplitude maximale.

$B, C$  : Coefficient de stations

$\Delta$  : Distance entre deux stations.

De par sa magnitude  $M$  (échelle de Richter), c'est une évaluation de la quantité d'énergie libérée au niveau du foyer du tremblement de terre. Cette échelle, définie en 1935, est basée sur le principe d'enregistrement des mouvements du sol effectué en un certain nombre de points d'observations situés à des distances quelconques de l'épicentre. A partir d'une étude statistique, nous avons établi une corrélation entre l'énergie libérée par un séisme et la magnitude [5].

**Tableau 1.1** : Energie libérée pour les différentes magnitudes de Richter.

| Magnitude | Energie en J    |
|-----------|-----------------|
| 1         |                 |
| 2         | $\sim 10^3$     |
| 3         | $\sim 10^6$     |
| 4         | $\sim 10^9$     |
| 5         | $\sim 10^{10}$  |
| 6         | $\sim 10^{13}$  |
| 7         | $\sim 10^{15}$  |
| 8         | $\sim 10^{17}$  |
| 9         | Sup à $10^{18}$ |

1.3.4 L'intensité (I)

L'intensité d'un séisme est estimée à partir des effets visibles (dégâts) produits à la surface (mouvements d'objets suspendus, fissures dans les maisons, etc.). Plusieurs échelles de mesure existent. La plus connue est l'échelle modifiée de Mercalli (appelée échelle MSK depuis sa révision en 1964) qui comprend 12 degrés depuis le I (vibrations ressenties uniquement par les instruments), jusqu'au XII (dévastation totale) [5].



Figure 1.2 : Caractéristiques d'un séisme.

| Intensité | Dégâts Matériels   |
|-----------|--|
| Degré I   | secousse imperceptible par l'homme, inscrite seulement par les sismographes.   |
| Degré II  | secousse ressentie par un petit nombre de personne, surtout par ceux situés aux étages supérieurs des maisons.   |
| Degré III | secousse faible ressentie par un certain nombre d'habitants, comme le serait l'ébranlement produit par une voiture lancée à grande vitesse; la direction et la durée de la secousse peuvent parfois être appréciées. |
| Degré IV  | ébranlement constaté par quelques personnes en plein air, mais beaucoup à l'intérieur des maisons; vibration de vaisselle, craquement de planchers et des plafonds.  |

|            |  |
|------------|--|
| Degré V    | ébranlement constaté par toute la population; réveil des dormeurs; ébranlement démeubles et de lits.   |
| Degré VI   | des personnes effrayées sortent des habitations; tintement général des sonnettes, arrêt des pendules, crépis fendillés, vaisselle brisée, cloches mises en branle, chute de plâtras.   |
| Degré VII  | maisons légèrement endommagées, lézardes dans les murs, chutes de cheminées isolées en mauvais état; écroulement de minarets, de mosquées ou d'églises mal construites.  |
| Degré VIII | sérieux dommages, fentes béantes dans les murs, chute de la plupart des cheminées, des clochers; renversement ou rotation des statues, des monuments funéraires; fissures dans les pentes raides ou dans les terrains humides; chutes de rochers en montagne.  |
| Degré IX   | de solides maisons en construction européenne sont sérieusement endommagées, un grand nombre, rendues inhabitables; d'autres s'écroulent plus ou moins complètement.   |
| Degré X    | la plupart des bâtiments en pierre et en charpente sont détruits avec leur fondations; fentes dans les murs en briques; rails de chemins de fer légèrement recourbés; dommages aux ponts; tuyaux de conduites brisés, fentes et plis ondulés dans les rues; éboulements; l'eau des rivières et des lacs es projetée sur le rivage. |
| Degré XI   | destruction totale des bâtiments en pierre, des ponts des digues; larges déchirures et crevasses dans le sol; grands éboulements de terrain  |
| Degré XII  | rien ne demeure plus des œuvres humaines; changement dans la topographie; formations de grandes failles, dislocations horizontales et cisaillement du sol; rivières détournées de leur cours.  |

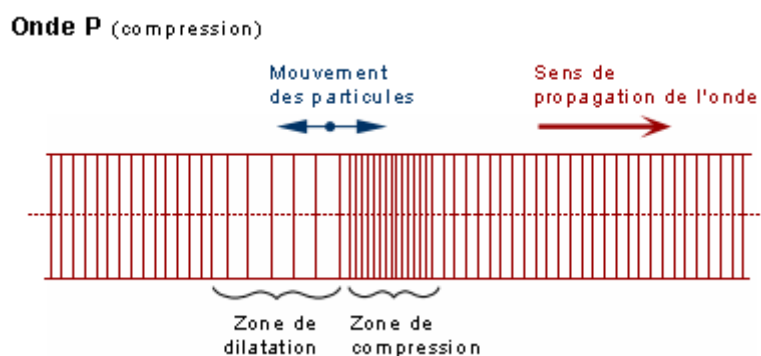
**Tableau 1.2 :** Echelle d'intensité de Mercalli.

## 1.4 Type d'ondes sismiques

### 1.4.1 Ondes primaires « P »

Les ondes P ou ondes primaires sont des ondes de compression (ou longitudinales) : elles compressent puis dilatent successivement le sol, parallèlement à la direction de leur propagation (**Figure 1.3**).

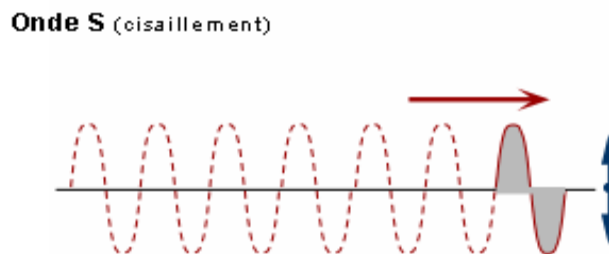
Ce sont les ondes les plus rapides pouvant atteindre jusqu'à 14km/s. On les enregistre bien sur la composante verticale du sismologue.



**Figure 1. 3** : Onde de type P.

### 1.4.2 Ondes transversales « S »

Les ondes S, ou ondes secondaires sont des ondes de cisaillement (ou transversales). Les vibrations s'effectuent perpendiculairement au sens de propagation de l'onde avec une vitesse de 4 à 5 km/s. Plus lentes que les ondes P, elles apparaissent en second sur les Sismogrammes. (**Figure 1. 4**).



**Figure 1.4** : Onde de type S.

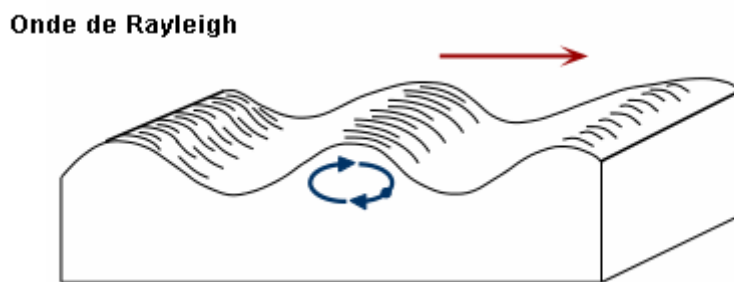


### 1.4.3 Ondes de surface

Ce sont des ondes de volume qui arrivent à la surface de la terre produisant des ondes de surface affectant le sol sur une profondeur extrêmement faible. On distingue à ce niveau deux types d'ondes :

#### 1.4.3.1 Ondes de Rayleigh ou ondes R

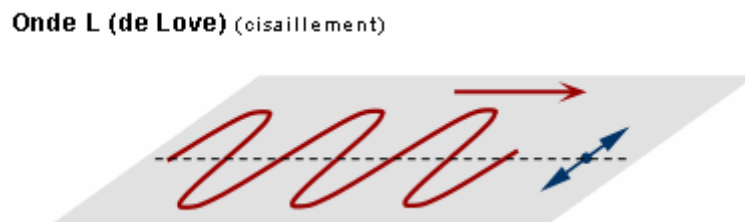
Elles sont assimilables à une vague ; les particules du sol se déplacent selon une ellipse dans le plan vertical de propagation et entraînent des compressions ou des tractions, ainsi que des cisaillements dans le sol. (**Figure 1. 5**).



**Figure 1. 5** : Onde de Rayleigh.

#### 1.4.3.2 Ondes de Love

Ce sont des ondes pour lesquelles les points du sol se déplacent dans un plan perpendiculaire à la direction de propagation ; elles n'engendrent que des contraintes de cisaillement. Elles impriment au sol un mouvement de vibration latéral. (**Figure 1. 6**) [6].



**Figure 1. 6** : Ondes de surface (l'onde de love).

## 1.5 Les failles

Le mouvement relatif continu des plaques tectoniques qui existe en profondeur dans la partie ductile du manteau est généralement accommodé par des ruptures successives dans la partie fragile superficielle de l'écorce terrestre. Les surfaces de ruptures sont nommées des failles tectoniques.

Une faille est donc une cassure avec déplacement de masse dans les matériaux rigides de l'écorce terrestre, en réaction à une contrainte. Les forces qui génèrent cette contrainte peuvent agir en compression, en extension ou en couple, définissant autant de types de failles [3].

### 1.5.1 Type des failles

L'analyse des ondes sismiques permet de comprendre quel type de faille est à l'origine du séisme, de trouver l'orientation de cette faille, ainsi que la direction du Glissement sur cette faille.

#### 1.5.1.1 Faille normal

Une faille normale résulte de mouvements d'extension. La déformation entraîne un affaissement des roches initiales.

#### 1.5.1.2 Faille inverse

Une faille inverse apparaît lors de mouvements de compression. La déformation entraîne un raccourcissement des terrains initiaux.

#### 1.5.1.3 Faille décrochant

Une faille en décrochement provoque un déplacement des blocs uniquement dans le sens horizontal [3].

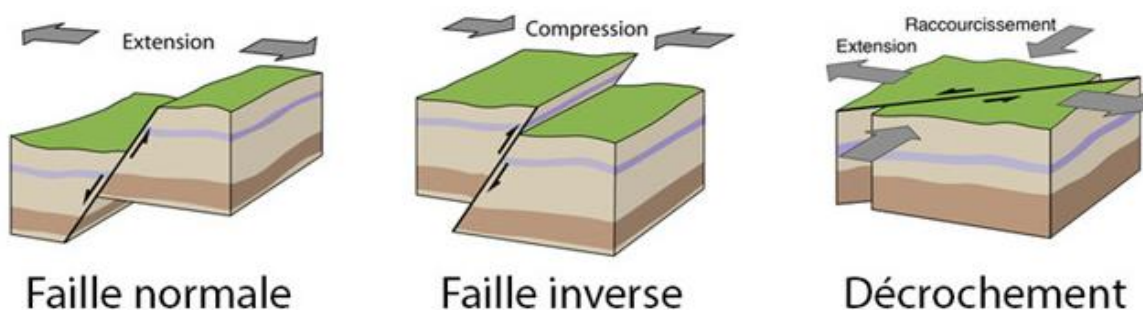


Figure 1.7 : Les types des failles sismiques [3].

## 1.6 Classification des séismes

Le foyer ou l'hypocentre est le lieu qui constitue l'origine de l'énergie de l'ébranlement des roches. Selon la profondeur des foyers on classe les séismes en [7]:

- Les séismes superficiels.
- Les séismes intermédiaires.
- Les séismes profonds.

### 1.6.1 Les séismes superficiels

La croûte terrestre étant rigide, elle peut casser. Etant donné que les plaques lithosphériques ont une épaisseur moyenne d'environ 70 Km, la majorité des séismes sont évidemment superficiels c'est à dire que les foyers ne dépassent pas une profondeur de quelques dizaines de kilomètre (0 – 70 Km environ).

### 1.6.2 Les séismes intermédiaires

Ce sont des ébranlements dont la profondeur des foyers est comprise entre 70 et 350 Km. Cette catégorie de séismes constitue environ 25% de la totalité des séismes.

### 1.6.3 Les séismes profonds

Ils sont rares que les deux catégories précédentes et ne représentent qu'environ 5%. Les séismes profonds ont des foyers qui se situent entre 350 et 700 Km.

## 1.7 La localisation des séismes

L'activité sismique est principalement liée à la tectonique des plaques [8].

### 1.7.1 Les séismes inter-plaquent

Dans la majorité des cas, les séismes se déclenchent en limite de plaques. C'est en effet au niveau de ces contacts que les contraintes occasionnées par la dérive des continents sont les plus fortes. Séismes et volcanisme sont souvent associés sur ces limites de plaques.

### 1.7.2 Les séismes intra-plaque

Même à l'intérieur des plaques tectoniques, des failles peuvent jouer et occasionner des séismes. Généralement moins violents que les précédents, ils correspondent à des réajustements des pressions dans la croûte terrestre.

### 1.7.3 Les séismes liés à l'activité volcanique

Les éruptions volcaniques, autre phénomène associé à la tectonique des plaques, occasionnent une multitude de séismes. Ces derniers peuvent permettre de prédire l'imminence d'une éruption.

### 1.7.4 Les séismes liés à l'activité humaine

Certaines activités humaines peuvent occasionner des séismes, généralement modérés. Il s'agit notamment de la mise en eau des barrages ou de l'exploitation des gisements souterrains (Gaz, minerais, .....etc.).

## 1.8 La structure interne de la Terre

On distingue trois parties principales : la croûte, d'épaisseur allant de 10 à 70 kilomètres, puis le manteau, qui s'étend de la base de la croûte jusqu'à une profondeur de 2900 kilomètres et enfin le noyau.

- **La croûte (ou écorce):** elle est à l'état solide. On distingue deux types de croûte, océanique et continentale.
  - **La croûte océanique :** se situe essentiellement sous les océans. Son épaisseur est de 5 km à 10 km en moyenne. Elle est constituée de roches basaltiques.
  - **La croûte continentale :** se trouve au niveau des continents. Son épaisseur est de 30 à 35 Km au niveau des plaines continentales et de 50 à 65 km sous les chaînes de montagnes. Moins dense que la croûte océanique, elle est formée de roches granitiques essentiellement.
- **Le manteau :** constitué par un manteau supérieur rigide jusqu'à 70 à 150 km, puis plastique jusqu'à 700 km de profondeur, et un manteau inférieur, solide et situé entre 700 et 2 900 km de profondeur.
- **Le noyau :** constitue la partie centrale de la Terre. Divisé en un noyau interne solide et un noyau externe liquide. Le centre de la Terre se trouve à 6 370 km de profondeur [9].

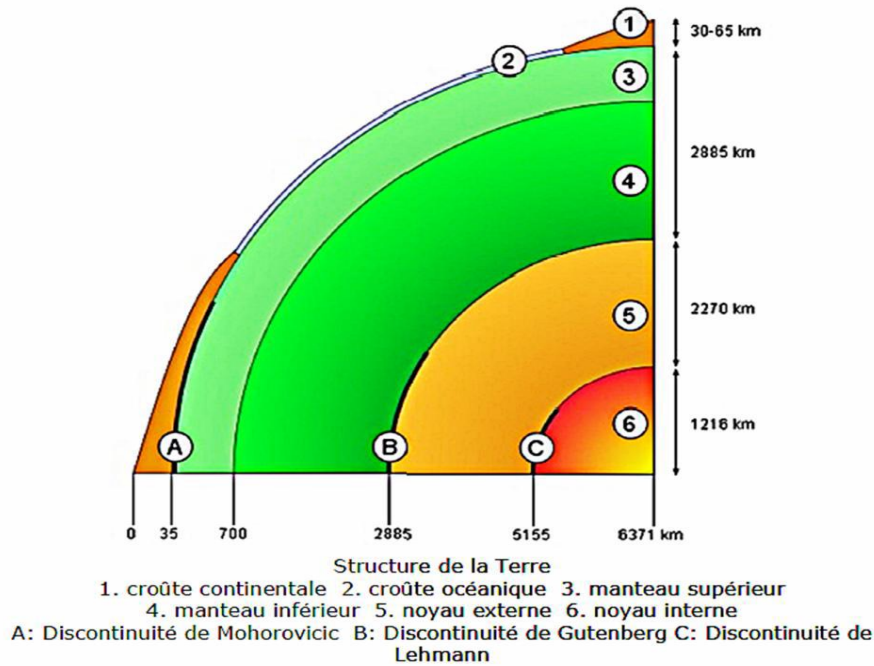


Figure 1.8 : Structure interne de la terre [9].

## 1.9 Enregistrement des séismes

Nous mesurons les ondes sismiques à l'aide d'appareils appelés : sismographe. Le principe est basé sur un appareil capable de sentir les vibrations ; ces vibrations sont transmises à une aiguille qui les inscrit sur cylindre qui tourne à une vitesse constante.

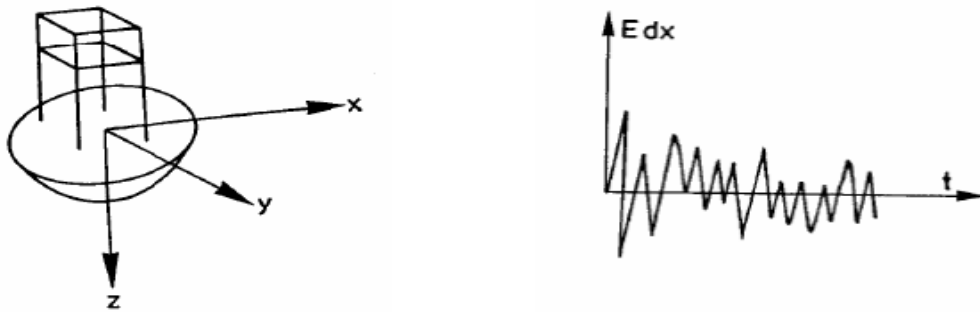
Donc, les sismographes enregistrent, sous forme de sismogrammes, les vibrations du sol provoquées par les ondes sismiques. L'amplitude de ces vibrations en un point ainsi que distance de ce point à l'épicentre permettent de calculer la magnitude du séisme, c'est-à-dire sa puissance.

## 1.10 Action sismique

Le mouvement sismique (ou action sismique) peut se décomposer en la somme d'une composante horizontale (suivant deux directions principales  $H_x$  et  $H_y$ ) et d'une composante verticale  $V_z$  (**Figure 1.7**). Il est utile de savoir que la composante verticale est, à l'aplomb de l'épicentre, égale à la composante horizontale.

Plus nous nous éloignons de cette épicentre, plus la composante verticale est moins importante par rapport à la composante horizontale, elle perd environ 50 à 70% de son influence. L'importance du mouvement vertical est souvent négligeable, car moins dommageable, que le mouvement horizontal. D'autant plus, du fait, que les structures soient réalisées pour supporter leur poids propre, cette composante verticale est moins importante

alors que la composante horizontale, elle, sera préférée pour dimensionner les structures à la résistance aux efforts horizontaux [10].



**Figure 1.9 :** Décomposition de l'action sismique.

### 1.11 Conclusion

Les tremblements de terre sont un phénomène destructeur qui résulte du mouvement des plaques tectoniques qui sont directement ressentis à la surface de la Terre et peuvent causer de nombreux dégâts. Il est impossible d'empêcher leur apparition. De nombreux sismologues étudient ce phénomène pour tenter de le prédire.

En utilisant des outils sismométriques, nous pouvons détecter un tremblement de terre. Il est possible de découvrir un moyen de contrer les effets dévastateurs du séisme, mais une étude très approfondie est nécessaire. Pour que l'homme puisse enfin faire face à un tremblement de terre.

# **Chapitre 2 :**

## **Description du matériel et logiciel du projet**

## 2.1 Introduction

Dans ce chapitre, nous allons faire la lumière sur l'aspect scientifique autre que le chapitre précédent dans lequel nous avons parlé des généralités sur les tremblements de terre et les types de tremblements de terre qui existent sur la planète Terre. Où nous expliquerons et décrirons les dispositifs électroniques et logiciels existants qui fonctionnent pour détecter les tremblements de terre. Parmi ces appareils on trouve l'Arduino, l'accéléromètre...etc.

## 2.2 Étude de la partie matérielle

### 2.2.1 Carte Arduino

Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée. Ces cartes permettent un accès simple et peu coûteux à l'informatique embarquée. De plus, elles sont entièrement libres de droit, autant sur l'aspect du code source (Open Source) que sur l'aspect matériel (Open Hardware). Ainsi, il est possible de refaire sa propre carte Arduino dans le but de l'améliorer ou d'enlever des fonctionnalités inutiles au projet.

Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de bibliothèques et de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications dans le cadre de hobby [11].

Le système Arduino permet de :

- Contrôler les appareils domestiques
- Fabriquer votre propre robot
- Faire un jeu de lumières
- Communiquer avec l'ordinateur
- Télécommander un appareil mobile (modélisme) Etc... [11].

Arduino est un microcontrôleur. Un microcontrôleur est un circuit intégré capable de recevoir de l'information tel celle d'un capteur de position ou d'un capteur de température. En fonction de cette information, le circuit intégré commande d'autres composants électroniques tels qu'un moteur ou un écran LCD [12].



### 2.2.1.1 Carte Arduino UNO

La carte Arduino Uno (**Figure 2.1**) est l'une des cartes les plus populaires de la famille Arduino dont le cœur est un microcontrôleur de référence ATmega328.

Il dispose de 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un résonateur céramique 16 MHz (CSTCE16M0V53-R0), une connexion USB, une prise d'alimentation, un connecteur ICSP et un bouton de réinitialisation. Il contient toute les interfaces nécessaires pour prendre en charge le microcontrôleur ; connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur CA / CC ou une batterie pour commencer. Vous pouvez bricoler votre UNO sans trop vous soucier de faire quelque chose de mal, dans le pire des cas, vous pouvez remplacer la puce pour quelques dollars et recommencer [13].

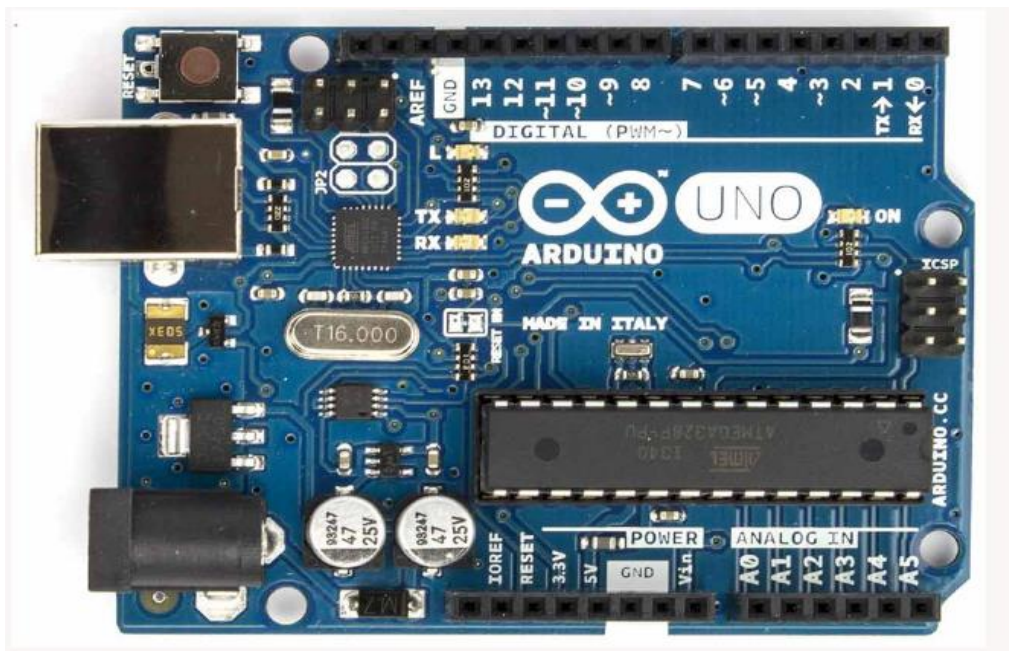


Figure 2.1 : La carte Arduino Uno.

### 2.2.1.2 Avantages de la carte Arduino UNO

Arduino Uno est le plus populaire et il offre vraiment un large éventail de possibilités et est également l'option la plus appropriée. Il présente de nombreux avantages :

- ❖ Réaliser un programme complexe, une connexion facile entre la carte et les composants externes, le stockage et la lecture du programme, peut recevoir des

informations analogiques et numériques, et équipé d'une carte de protection du moteur (permettant à plus de pilotes de spécifier).

- ❖ Utilisons Scratch pour simuler le comportement de notre bot sur une instance virtuelle du chemin.
- ❖ **Multiplateforme** : le logiciel Arduino, écrit en JAVA, fonctionne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes de microcontrôleurs sont limités à Windows.
- ❖ **Environnement de programmation clair et simple** : L'environnement de programmation Arduino (Arduino IDE) est facile à utiliser, mais suffisamment flexible pour tirer parti d'applications plus complexes.
- ❖ Le matériel est également open source et extensible : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168 et ATMEGA328, et les concepteurs de circuits expérimentés peuvent créer leur propre version des cartes Arduino, les compléter et les améliorer [14].

### 2.2.1.3 Caractéristiques technique de la carte Arduino UNO

Dans ce tableau 2.1, on présentera les caractéristiques de la carte Arduino UNO

| Désignation                                 | Description                                   |
|---|---|
| <b>Microcontrôleur</b>                      | ATmega328                                     |
| <b>Tension d'alimentation interne</b>       | 5V  |
| <b>Tension d'alimentation (recommandée)</b> | 7 à 12V, limites =6 à 20 V                    |
| <b>Entrées/sorties numériques</b>           | 14 dont 6 sorties PWM                         |
| <b>Entrées analogiques</b>                  | 6   |
| <b>Courant max par broches E/S</b>          | 40 mA   |
| <b>Courant max sur sortie 3,3V</b>          | 50mA  |
| <b>Mémoire Flash</b>                        | 32 KB dont 0.5 KB utilisée par le boot loader |
| <b>Mémoire SRAM</b>                         | 2 KB  |

|                          |                 |
|--------------------------|-----------------|
| <b>mémoire EEPROM</b>    | 1 KB            |
| <b>Fréquence horloge</b> | 16 MHz          |
| <b>Dimensions</b>        | 68.6mm x 53.3mm |

Tableau 2.1: Caractéristiques d'une carte Arduino Uno.

2.2.1.4 Brochage de la carte Arduino Uno

Les brochages de la carte Arduino Uno sont présentés ci-dessous :

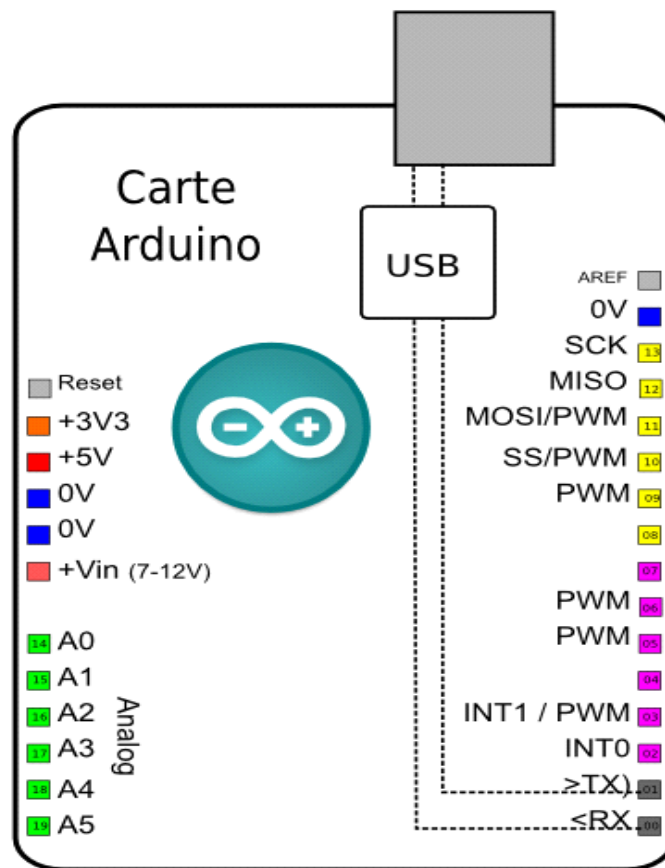


Figure 2.2 : Brochage de la carte Arduino Uno [15].

2.2.1.5 Différents éléments de la carte Arduino Uno

Les différents éléments de l'Arduino Uno sont présentés comme suit :

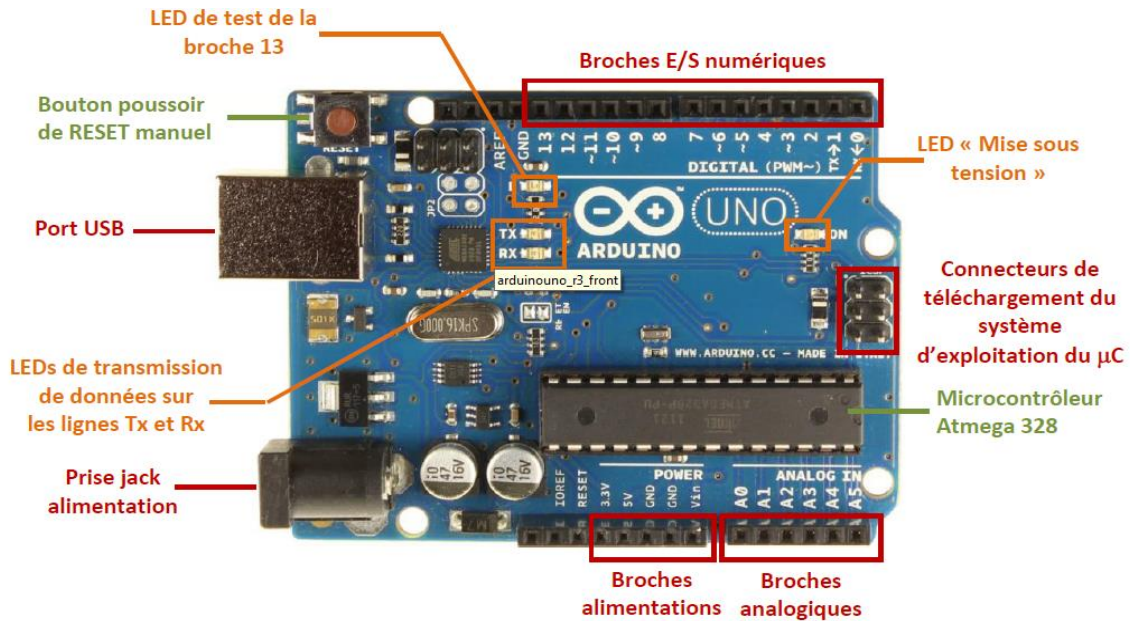


Figure 2.3 : Les interfaces d’E/S de la carte Arduino Uno.

a) Microcontrôleur ATMEL ATmega328

Le microcontrôleur de la carte Arduino UNO est un ATmega328. C'est un Microcontrôleur ATMEL de la famille AVR 8bits.

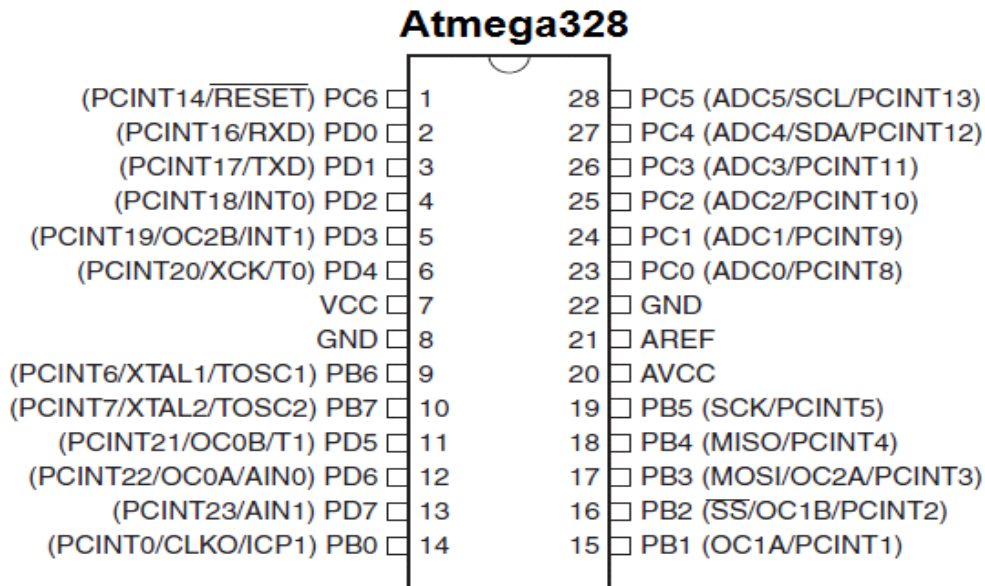


Figure 2.4: Microcontrôleur ATmega328 [16].

- ❖ Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit. Aujourd'hui, on n'a pas besoin de souder un grand nombre de composants encombrants ; tels que les transistors ; les résistances et les condensateurs. Le tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C [17].



**Figure 2.5:** Microcontrôleur ATmega328 (type classique).

- **Gestion des mémoires dans la carte Arduino**

L'ATmega 328 a 32ko de mémoire FLASH pour stocker le programme (dont 0.5ko également utilisé par le bootloader). L'ATmega 328 a également 2kcode mémoire SRAM (volatile) et 1ko d'EEPROM (non volatile : mémoire qui peut être lue à l'aide de la librairie EEPROM) [18].

- b) Interfaces d'Entrées/Sorties**

Chacune des 14 broches numériques de la carte Uno (numérotées de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()`, et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne « résistance de rappel » (pull-up) (déconnectée par défaut) de 20 à 50kOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite (broche, HIGH)`.

Il y a entre ces broches celles qui ont des fonctionnalités en plus :

- **Communication série** : broche 0 (RX) et 1 (TX). Utilisée pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes de circuit intégré ATmega 16U2 programmé en convertisseur USB/Série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel à la transmission série travers ces broches avec l'instruction `Serial.Print ()`, à condition que le câble USB déconnecté, sinon il va y avoir un chevauchement.
- **Interruptions externes** : broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt ()` pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée)** : broches 3, 5, 6, 9, 10 et 11. Fournissant une impulsion PWM 8 bits à l'aide de l'instruction `analogWrite ()`.
- **SPI (Interface Série Périphérique)** : broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI disponible avec une librairie pour la communication SPI. Les broches SPI sont également connectées sur le connecteur ISCP.
- **I2C** : Les broches 4 (SDA) et 5 (SCL). Prise en charge des communications I2C (TWI Two-Wire Interface) à l'aide de la bibliothèque `Wire`.
- **LED** : broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsqu'elle est au niveau BAS, la LED est éteinte. [18]

### c) Les entrées analogiques

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead ()` du langage Arduino.

- ❖ les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19, aux cas où le nombre de broches numériques n'est suffisant.



- **Autre broches** : il y a deux autres broches disponibles sur la carte :
- ✓ **AREF** : tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction analog Reference (). Elle utilisée pour comparer la valeur d'une tension d'entrée par rapport à la valeur d'une tension de référence choisie.
- ✓ **Reset** : mettre cette broche au niveau BAS entraine la réinitialisation du microcontrôleur. Cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

#### d) Alimentation de carte arduino

La carte ArduinoUno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus).

L'adaptateur secteur peut être connecté en branchant une prise de 2.1mm, dédié de la pile positive au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et Vin (tension positive en entrée) du connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20V. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si en utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Ainsi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V [18].

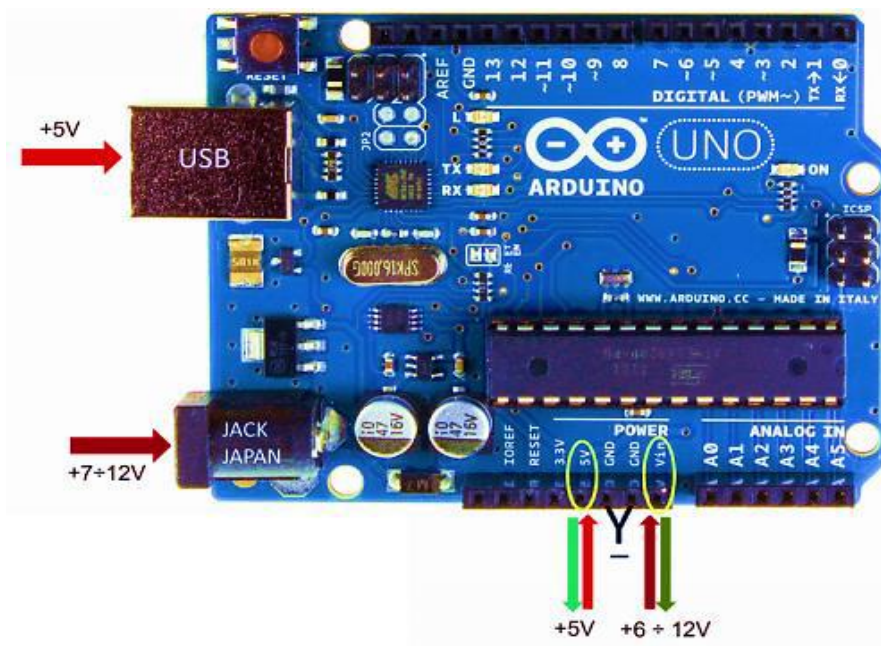
Il est noté qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jack et avoir le câble USB connecté.

Les broches d'alimentation sont les suivantes :

- ✓ **RESET** : Cette broche permet de réinitialiser l'Arduino et donc de redémarrer le programme au début. Pour réinitialiser l'Arduino, il suffit de mettre cette broche à 0V puis de la repasser à 5V, ou 3,3V selon l'Arduino.
- ✓ **IOREF** : Cette broche est destinée à indiquer aux « shields » la tension de fonctionnement de l'Arduino. Sur un Arduino 5V, elle aura une tension de 5V et sur un 3,3V une tension de 3,3V.
- ✓ **5V** : la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (un régulateur de tension est intégré dans la carte L'Arduino

S'il fonctionne sous une tension de 5V, est équipé d'un régulateur 5V. La sortie de ce régulateur est connectée sur cette broche qui peut donc être employée pour alimenter des circuits externes en 5V. Si on désire construire sa propre alimentation 5V, on peut également fournir cette tension à l'Arduino via cette broche.

- ✓ **3.3V** : Un Arduino Uno est équipé d'un régulateur 3,3V dont la sortie est connectée à cette broche. On peut s'en servir pour alimenter un circuit externe en 3,3V ; ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu de 5V). L'intensité maximale disponible sur cette broche de 50mA.
- ✓ **VIN** : L'entrée du régulateur embarqué sur l'Arduino (à distinguer du 5V de la connexion USB ou autre source (5V) régulée). On peut choisir d'alimenter l'Arduino via cette broche au lieu d'utiliser la prise d'alimentation.
- ✓ **GND** : La masse ou 0V.



**Figure 2.6:** Sources de l'Alimentation de la carte Arduino UNO.



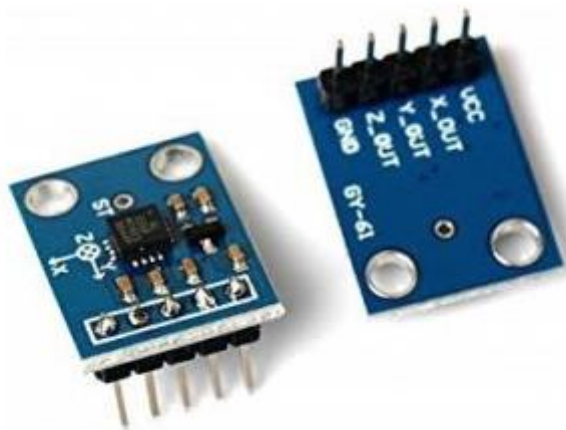
### 2.2.2 Accéléromètre ADXL335

Ce module d'accéléromètre est basé sur le populaire circuit intégré d'accéléromètre analogique à trois axes ADXL335, qui lit les accélérations X, Y et Z sous forme de tensions analogiques. En mesurant la quantité d'accélération due à la gravité, un accéléromètre peut déterminer l'angle auquel il est incliné par rapport à la terre.

En détectant la quantité d'accélération dynamique, l'accéléromètre peut déterminer à quelle vitesse et dans quelle direction l'appareil se déplace. En utilisant ces deux propriétés, vous pouvez créer toutes sortes de projets sympas, des instruments de musique (imaginez jouer et avoir l'inclinaison connectée au niveau de distorsion ou au pitch-bend) à un moniteur de vitesse sur votre voiture (ou la voiture de vos enfants). L'accéléromètre est très facile à interfacer avec un microcontrôleur Arduino à l'aide de 3 broches d'entrée analogiques et peut être utilisé avec la plupart des autres microcontrôleurs, tels que le PIC ou l'AVR [19].

ADXL335 3-Axis Accelerometer Module GY-61 is a three axis accelerometer sensor module based on ADXL335 integrated circuit. The ADXL335 is a triplé axis accelerometer with extremely low noise and power consumption. The sensor has a full sensing range of +/-3g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

There is an on-board 3.3V voltage regulator to power the ADXL335 so power provided should be between 3.3V and 6V DC [20].



**Figure 2.7:** Accéléromètre ADXL335 Module GY-61.

### 2.2.2.1 Caractéristiques et spécification Accéléromètre ADXL335

- Les caractéristiques et en détail du Accéléromètre ADXL335 sont présentées ci-dessous :
  - Tension d'alimentation CC 3V-6V
  - Régulateur de tension LDO embarqué
  - Il peut être interface avec un microcontrôleur 3V3 ou 5V.
  - Tous les composants nécessaires sont remplis.
  - Ultra-Low Power: 40uA en mode mesure, 0.1uA en veille @ 2.5V
  - Appuyez sur / Double Tap Détection
  - Détection de chute libre
  - Sortie analogique [19].
- **General Specifications :**
  - ADXL335 3-axis Accelerometer
  - On-board 3.3V Voltage Regulator
  - Analog voltage output centered at 1.65V
  - Suitable for connection to 5V and 3.3V système [20]

- **Techniques Spécifications :**

Les spécifications techniques sont dans le tableau suivant :

|                             |  |
|-----------------------------|--|
| Sensor Chip                 | ADXL335                                    |
| Operating Voltage Range     | 3V ~ 5V                                    |
| Supply Current              | 400uA                                      |
| Interface                   | Analog quantity output                     |
| Full scale range            | +/-3g                                      |
| Operating Temperature       | -40'C~ +85'C                               |
| Sensitivity                 | 300mv /g                                   |
| Sensitivity of accuracy (%) | +/- 10                                     |
| Application                 | Various electronic products or DIY project |

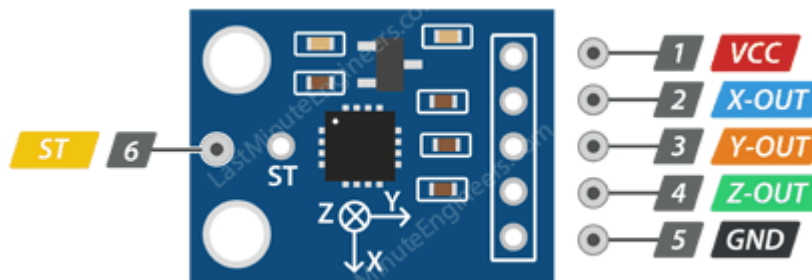
|            |   |
|------------|---|
| Material   | PCB + Brass                               |
| Dimensions | 21 x 16 x 10 mm / 0.83 x 0.63 x 0.39 inch |
| Weight     | 2 g / 0.07 oz                             |
| Color      | Blue                                      |

**Tableau 2.2:** Caractéristiques et spécification Accéléromètre ADXL335 [20]

### 2.2.2.2 Broches de connexion

Pin Définitions:

- 1- **VCC:** 3.3V or 5V
  - 2- **X\_OUT:** Analog Output
  - 3- **Y\_OUT:** Analog Output
  - 4- **Z\_OUT:** Analog Output
  - 5- **GND:** Ground
  - 6- **ST :** pin contrôle la fonction d'auto-test
- L'accéléromètre ADXL335 fournit une tension analogique aux broches de sortie X, Y, Z ; qui est proportionnel à l'accélération dans les directions respectives, c'est-à-dire X, Y, Z.



**Figure 2.8:** Broches de connexion Accéléromètre ADXL335 [21].

### 2.2.2.3 Fonctionnement de l'accéléromètre ADXL335

Le dispositif le plus couramment utilisé est l'accéléromètre piézoélectrique. Comme son nom l'indique, il utilise le principe de l'effet piézoélectrique. Le dispositif est constitué d'un cristal de quartz piézoélectrique sur lequel est appliquée une force d'accélération, dont la valeur doit être mesurée.

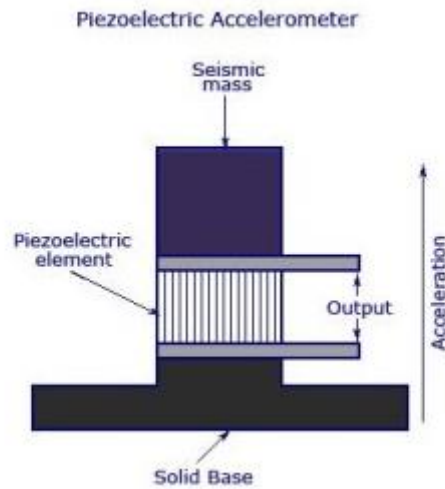
Au repos, l'accéléromètre mesure 1 g, ce qui équivaut à l'attraction gravitationnelle de la Terre à 9,81 mètres par seconde. Certains accéléromètres fonctionnent sur l'effet piézoélectrique qui peut calculer de petits changements de tension tandis que d'autres calculent le changement de capacité de deux composants.

Un accéléromètre de base se compose de plusieurs axes, deux pour déterminer la plupart des mouvements bidimensionnels avec l'option d'un troisième pour le positionnement 3D. De nos jours, de nombreux Smartphones et appareils photo utilisent des modèles à trois axes, tandis que les voitures utilisent essentiellement deux axes pour déterminer le moment de l'impact. Les accéléromètres sont sensibles, ils parviennent même à mesurer le moindre changement d'accélération. L'accéléromètre détecte les forces statiques ou dynamiques d'accélération, c'est-à-dire les forces statiques telles que la gravité et les forces dynamiques telles que les vibrations et le mouvement.

De nos jours, l'accéléromètre peut mesurer plusieurs axes jusqu'à 3 axes. Fondamentalement, l'accéléromètre se compose de plaques capacitives internes, certaines sont fixes tandis que d'autres sont attachées à de minuscules ressorts qui se déplacent à l'intérieur lorsque la force d'accélération agit sur le capteur. Au fur et à mesure que les plaques se déplacent, l'accélération peut être déterminée en changeant la capacité.

Alors que les accéléromètres capacitifs utilisent un élément en silicium. Ici, il détecte les changements de capacité lorsque l'accélération se produit. Ce changement se transforme en une tension pour mesurer la vitesse [22].

En raison de la propriété particulière d'auto génération, le cristal produit une tension proportionnelle à la force d'accélération. Le fonctionnement de base et la disposition sont illustrés dans la figure suivante :



**Figure 2.9:** Accéléromètre effet piézoélectrique [19].

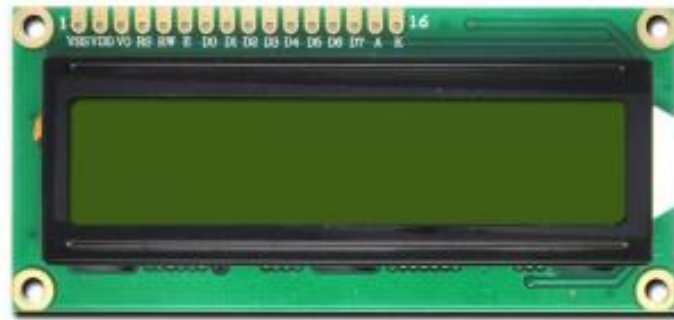
### 2.2.3 L'afficheur LCD

L'afficheur LCD alphanumérique est le composant idéal pour ce type d'application :

Le nombre de caractères étant limité mais suffisant, il se contrôle aisément au travers d'une arduino. Le modèle utilisé comporte quatre lignes de vingt caractères qui permettent de créer une petite interface utilisateur efficace et dispose d'un rétro éclairage offrant la possibilité de lire des informations dans l'obscurité et qui ajoute une touche moderne et attractive au boîtier.

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquide Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu courant (de 1 à 5 mA), bons marchés et s'utilisent avec beaucoup de facilité.

Tout projet qui nécessite tant de convivialité ou de contrôle pour l'utilisateur doit comporter un afficheur. En effet, celui-ci permet de manière très rapide de révéler n'importe quelle information qui pourrait être utile au programmeur ou à l'utilisateur.



**Figure 2.10:** L'afficheur LCD16 x 2.

### 2.2.3.1 L'afficheur utilisé dispose de

- Une RAM (DDRAM : DATA RAM) de 80 caractères correspondant.
- Une RAM (CGRAM : CHARACTER GRAPHIC RAM) permettant de créer de nouveaux caractères.
- 4 lignes de 20 caractères.
- Le registre d'instruction I R: (Instruction Register) :

C'est le registre de contrôle, suivant la valeur que l'on met dedans l'afficheur exécute des opérations de configurations. Il permet aussi de positionner le curseur parmi les 80 adresses de l'afficheur.

- Le registre de données D R: (Data Register) :

Suivant la valeur que l'on met dedans l'afficheur peut afficher un caractère (Code ASCII ou spécifiques) ou créer une ligne d'une matrice d'un nouveau caractère [23].

Cet afficheur nécessite une alimentation de 5V pour pouvoir alimenter son pilote interne et ainsi permettre l'affichage des caractères sur l'écran. La résistance variable permet de pouvoir modifier le contraste à tout moment. Ce LCD possède huit bits de données. Le composant propose en effet un mode de transmission sur 8 bits ou sur 4 bits (Dans ce mode, seul les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisés pour transmettre les données et les lire.

Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse). Il possède également trois lignes de contrôle : une de lecture ou d'écriture de données sur le LCD, une qui informe si les données servent à la configuration de l'afficheur ou à l'affichage, et une qui avertit l'afficheur qu'un caractère est présent sur le port.

### 2.2.3.2 Rôle des différentes broches de l'afficheur LCD

- Les PIN 1 et 2 (VSS et VDD) : Alimentation de l'afficheur LCD en 0V- 5V.
- Le PIN 3 (VEE) : Entrée permettant de régler le contraste de l'afficheur LCD.
- Le PIN 4 (RS: Registre Select) : cette entrée permet d'indiquer à l'afficheur si l'on souhaite réaliser une commande (RS=0) par des instructions spécifiques ou écrire une donnée (envoi du code du caractère à afficher) sur le bus (RS=1).
- Le PIN 5 (R/W: Read/write) : entrée de lecture (R/W=1) et d'écriture (R/W=0). Lorsqu'on commande l'afficheur LCD il faut se placer en écriture.
- Le PIN 6 (E: ENABLE) : entrée de validation, elle permet de valider les données sur un front descendant. Lorsque E=0 alors le bus de données est à l'état haute impédance.
- Les PIN 7 à 14 (D0...D7) : bus de données bidirectionnel, il permet de transférer les instructions ou les données à l'afficheur LCD.

### 2.2.3.3 Brochage de l'afficheur LCD

Les Broches de l'afficheur LCD est dans le tableau suivant :

| Broche | Nom | Niveau | Fonction   |
|--------|-----|--------|--|
| 1      | Vss | -      | Masse  |
| 2      | Vdd | -      | Alimentation positive +5V  |
| 3      | Vo  | 0-5V   | Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.   |
| 4      | RS  | TTL    | Sélection du registre (Registre Select)<br>Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée. |
| 5      | R/W | TTL    | Lecture ou écriture (Read/Write)<br>L : Écriture<br>H : Lecture  |
| 6      | ET  | TTL    | Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.   |
| 7      | J0  | TTL    | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)  |
| 8      | D1  | TTL    | Bus de données bidirectionnel 3 états (haute impédance lorsque   |

|    |     |     |   |
|----|-----|-----|---|
|    |     |     | E=0)  |
| 9  | D2  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 10 | D3  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 11 | D4  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 12 | D5  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 13 | D6  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 14 | D7  | TTL | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) |
| 15 | UNE | -   | Anode rétro éclairage (+5V)   |
| 16 | À   | -   | Cathode rétro éclairage (masse)                                     |

Tableau 2.3: la description des broches de LCD [24].

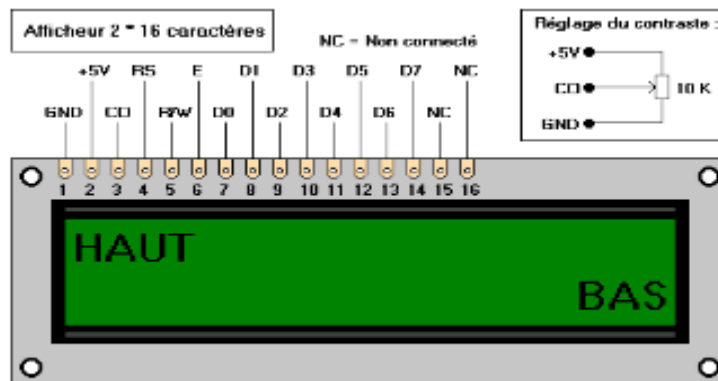


Figure 2.11: Brochage de l’afficheur LCD.



### 2.2.3.4 Principe des cristaux liquides

L'afficheur est constitué de deux lames de verre, distantes de 20  $\mu\text{m}$  environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réfléchissants). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réfléchissant ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Les modèles transmissifs fonctionnent différemment: normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité; pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétro éclairés.

### 2.2.3.5 Le fonctionnement l'afficheur LCD

On envoie deux types d'information à l'afficheur :

- Les commandes qui permettent de l'initialiser : positionnement du curseur, effacement écran, etc. ;
- Les données à afficher.

L'entrée RS permet de spécifier si on envoie une commande ou une donnée :

- RS=0 : instruction (commande) ;
- RS=1 : caractère (donnée).

L'afficheur dispose d'une entrée R/W pour spécifier une lecture ou une écriture :

- R/W=0 : écriture vers l'afficheur ;
- R/W=1 : lecture de l'afficheur.

Pour valider tous les échanges sur le bus de données (D7-D0) on utilise l'entrée E de l'afficheur. Un front descendant sur cette entrée valide la donnée. En programmation, il faudra placer un court instant E à l'état haut puis à l'état bas.

Il est possible d'utiliser l'afficheur LCD en mode 8 bits normal ou en mode 4 bits pour économiser les broches de son  $\mu$ Contrôleur par exemple, c'est assez pratique :

- **Mode 8 bits:**

En mode 8 bits on place la donnée ou la commande sur le bus D7 à D0 et on valide avec E

- **Mode 4 bits:**

En mode 4 bits on place déjà les poids forts de la donnée ou la commande sur les bits de D7 à D4 et on valide une première fois avec E. Puis on va mettre le poids faibles sur les bits de D7 à D4 et on valide une seconde fois avec E.

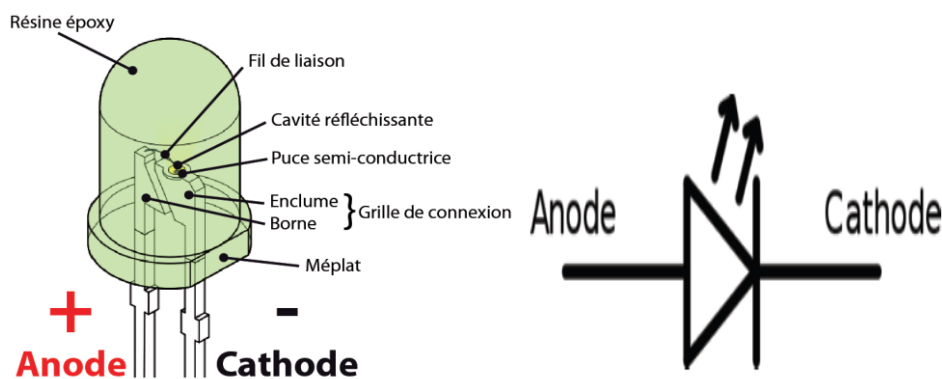
L'envoi ou la lecture d'un octet s'effectue donc en 2 temps dans ce mode [25].

### 2.2.4 LED

La LED (Light Emitting Diode) est textuellement une “diode émettant de la lumière”. Elle est également appelée DEL (Diode électroluminescente) en français et SSL (Solid State Lighting) en anglais. Il s'agit d'un composant (opto) électronique qui, d'une part, ne laisse passer le courant électrique que dans un sens (définition de la diode) et d'autre part, émet de la lumière.

La LED a une histoire relativement récente par rapport aux autres sources d'éclairage : elle a commencé à se développer d'un point de vue industriel dans un premier temps dans la signalétique à partir des années 1970 et s'est vraiment diffusée dans l'éclairage général depuis les années 2000/2010.

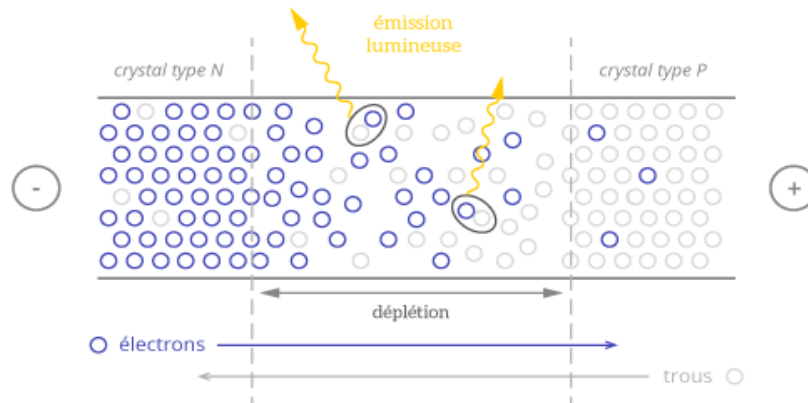
Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens (le sens passant). Lorsqu'elle est traversée par un courant, la LED oppose une tension fixe, L'intensité admissible dans la LED est aussi limitée (pour les LED ordinaires de 5mm, 24mA environ) [26].



**Figure 2.12:** Symbole de la LED.

### 2.2.4.1 Fonctionnement de LED

Les LED fonctionnent suivant le principe de la luminescence et plus exactement de l'électroluminescence, puisque la lumière est émise suite au passage d'un courant électrique. Le principe est de produire un déficit d'électron dans une zone (matériau dopé p) et un excédent dans une autre (matériau dopé n) : à la jonction entre les deux matériaux, les "trous" d'électrons se recombinent avec les électrons et génèrent un photon [26].



**Figure 2.13:** Fonctionnement de LED.

### 2.2.4.2 Les caractéristiques lumineuses des LED

Alors que le rayonnement lumineux, avec les lampes classiques, se répartit plus ou moins dans toutes les directions (d'où les luminaires), les LED fournissent normalement un flux lumineux très directionnel.

Mais ce flux lumineux n'est pas naturellement «blanc». Pour obtenir une lumière de couleur acceptable il faut généralement combiner plusieurs couches luminescentes : une diode bleue, une diode verte et une diode rouge.

L'ennui est que les trois puces de couleurs différentes n'ont pas exactement le même comportement dans le temps, ce qui entraîne une certaine évolution de la couleur dans le temps, le blanc ne restant pas strictement identique. Actuellement le rendu des couleurs des diodes luminescentes correspond à un indice de rendu de couleur allant de 75 à 85. On espère, néanmoins, pouvoir le porter à 85-90, voire même 95 [27].

### 2.2.5 Buzzer

Le Buzzer est une structure intégrée de transducteurs électroniques, alimentation en courant continu, largement utilisé dans les ordinateurs, les imprimantes, les photocopieurs, les alarmes, jouets électroniques, matériel électronique automobile et d'autres produits électroniques pour les appareils sonores. Le Buzzer passif module d'alarme utilisé pour l'Arduino [28].



**Figure 2.14:** magnétique buzzer.

Le Buzzer interne active avec la source de choc, de sorte que sera appelé à une tension. Les sources internes passives sans chocs, donc si un signal continu ne peut pas faire tweet. Doit 2K ~ 5K ave place pour le conduire. Buzzer passif est souvent coûteux passif. Les avantages des Buzzer passifs sont les suivants:

- 1- Bon marché,
- 2- contrôle de la fréquence sonore, vous pouvez faire un "plus que l'efficacité d'un cheveu mètre Soula Xi 'Fruit.
- 3- Dans certains cas particuliers, vous pouvez réutiliser un contrôle et un port LED Buzzer actif.

#### 2.2.5.1 Le Buzzer est Caractérisé par

- **Type :** Buzzer passive
- **Tension de travail:** 3.5-5.5v
- **Courant de travail:** < 25mA
- **Dimension PCB:** 18.5mm x 15mm (L x P)
- **Fonction de Buzzer :** buzz

## 2.3 Étude de la partie logicielle

### 2.3.1 Arduino IDE

Arduino IDE est un logiciel open source, conçu par Arduino.cc et principalement utilisé pour écrire, compiler et télécharger du code sur presque tous les modules Arduino.

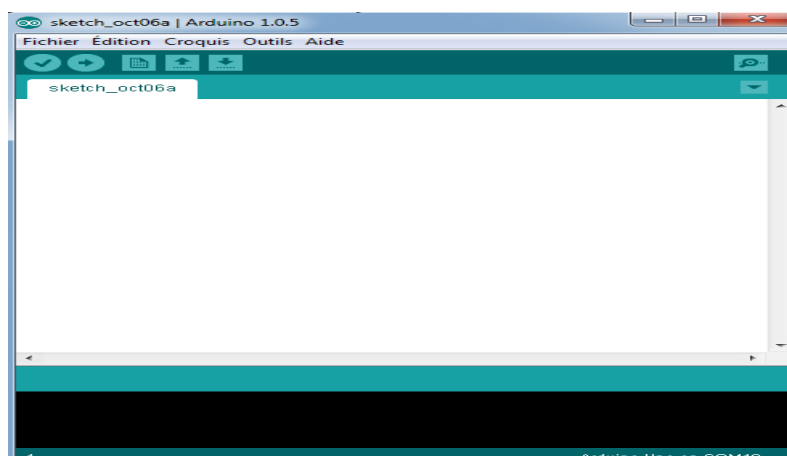
Il s'agit d'un logiciel officiel Arduino, ce qui rend la compilation de code trop facile pour que même une personne ordinaire sans aucune connaissance technique préalable puisse se familiariser avec le processus d'apprentissage.

Il est disponible pour tous les systèmes d'exploitation, c'est-à-dire MAC, Windows, Linux et s'exécute sur la plate-forme Java dotée de fonctions et de commandes intégrées qui jouent un rôle essentiel dans le débogage, l'édition et la compilation du code.

L'environnement IDE contient principalement deux parties de base : l'éditeur et le compilateur où le premier est utilisé pour écrire le code requis et le dernier est utilisé pour compiler et télécharger le code dans le module Arduino donné.

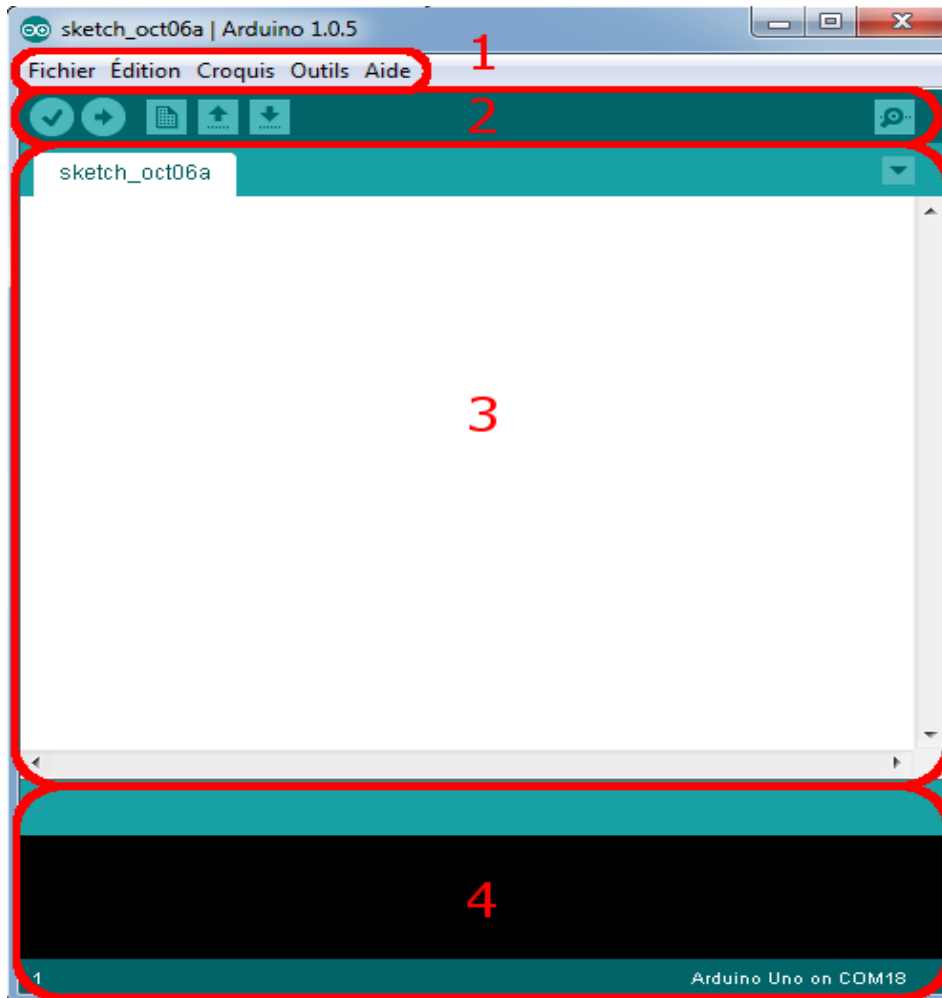
Chacun d'eux contient un microcontrôleur sur la carte qui est réellement programmé et accepte les informations sous forme de code.

Le code principal, également connu sous le nom d'esquisse, créé sur la plate-forme IDE générera finalement un fichier hexadécimal qui sera ensuite transféré et téléchargé dans le contrôleur de la carte [29].



**Figure 2.15:** L'interface de l'IDE Arduino.

## 2.3.1.1 Présentation du logiciel



**Figure 2.16:** L'interface de l'IDE Arduino en détail.

**Le cadre numéro 1 :** ce sont les options de configuration du logiciel.

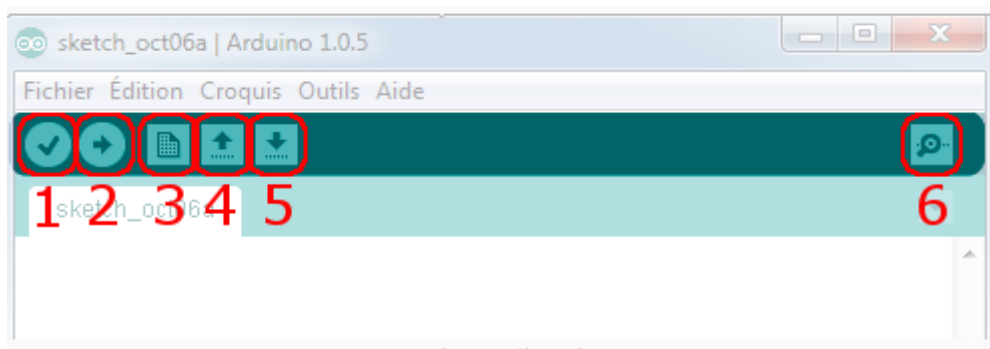
**Le cadre numéro 2 :** il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes.

**Le cadre numéro 3 :** ce bloc va contenir le programme que nous allons créer, Programme (croquis ou sketch).

**Le cadre numéro 4 :** celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur (console) [31].

Certains microcontrôleurs disposent d'une plateforme de programmation. La carte Arduino utilise un logiciel open source gratuit appelé Arduino IDE (pour Mac OS, Windows et Linux).

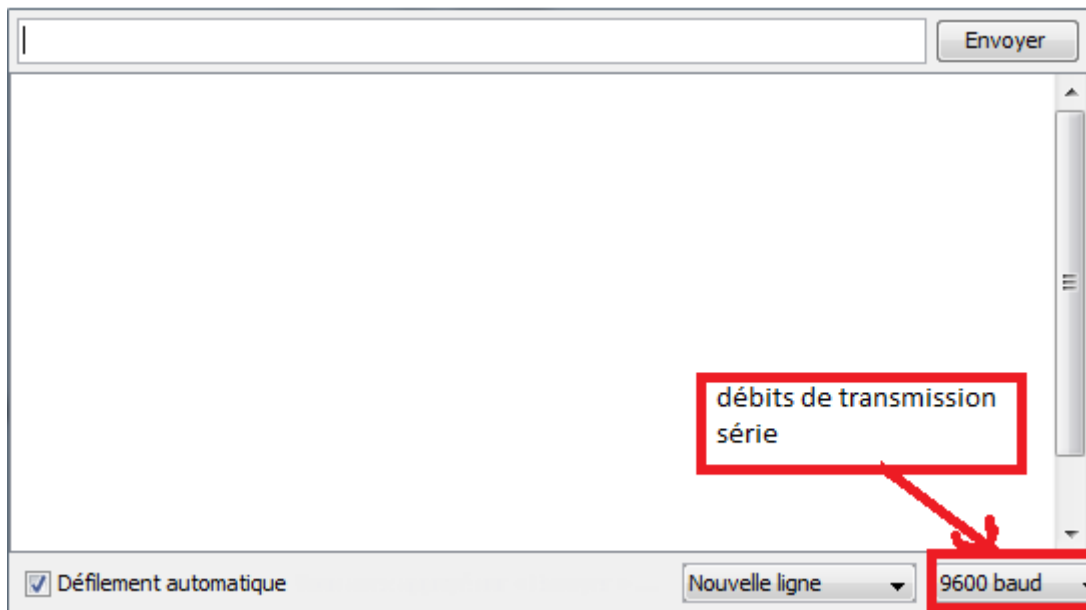
Arduino IDE est minimaliste, mais il fournit un environnement presque complet pour la plupart des projets basés sur Arduino. La barre de menu du haut contient les options standard, notamment « Fichier » (nouveau, chargement de sauvegarde, etc.), « Modifier » (police, copier, coller, etc.), « Sketch » (pour la compilation et la programmation), « Outils » (options utiles pour tester des projets) et « Aide ». La section centrale de l'IDE est un simple éditeur de texte dans lequel vous pouvez entrer le code du programme. La section inférieure de l'IDE est dédiée à une fenêtre de sortie qui est utilisée pour voir l'état de la compilation, la quantité de mémoire utilisée, toutes les erreurs trouvées dans le programme et divers autres messages utiles [30].



**Figure 2.17:** La barre d'outils.

- **Bouton 1 :** Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme
- **Bouton 2 :** Charge (téléverse) le programme dans la carte Arduino.
- **Bouton 3 :** Crée un nouveau fichier.
- **Bouton 4 :** Ouvre un fichier.
- **Bouton 5 :** Enregistre le fichier.
- **Bouton 6 :** Ouvre le moniteur série (on verra plus tard ce que c'est) [31].

Le bouton apparaissant dans le coin supérieur droit est le Serial Monitor. Une fenêtre pop-up séparée qui agit comme un terminal indépendant et joue un rôle vital pour l'envoi et la réception des données série. Il est possible d'accéder au panneau Outils et sélectionner Serial Monitor, ou appuyer simultanément sur (Ctrl + Maj + M) pour l'ouvrir instantanément. Le Serial Monitor aidera en fait à déboguer les croquis écrits où vous pourrez avoir une idée du fonctionnement de votre programme. Votre module Arduino doit être connecté à l'ordinateur par un câble USB afin d'activer le moniteur série [29].



**Figure 2.18:** Moniteur série de L'Arduino IDE.

### 2.3.1.2 Interface de programmation Arduino

La façon dont le microcontrôleur gère ses entrées / sorties est fixée par un programme, contenu dans le microcontrôleur. Ce programme doit être écrit par l'utilisateur. En pratique, l'utilisateur écrit le programme en langage C, en utilisant un environnement de développement spécialisé (IDE) installé sur un ordinateur. Ce programme est ensuite compilé et téléversé dans le microcontrôleur par liaison série (USB).

Nous utiliserons l'IDE standard Arduino (arduino.exe). Il suffit de taper le code dans la fenêtre dédiée, de compiler et de téléverser le programme sur la carte arduino. La carte doit être reliée à l'ordinateur par un câble USB. Le modèle de la carte arduino (il y a plusieurs types de carte) ainsi que le port série sur lequel est branchée la carte doivent être déclarés dans le menu de l'IDE Outils/type de carte et Outils/port série [32].



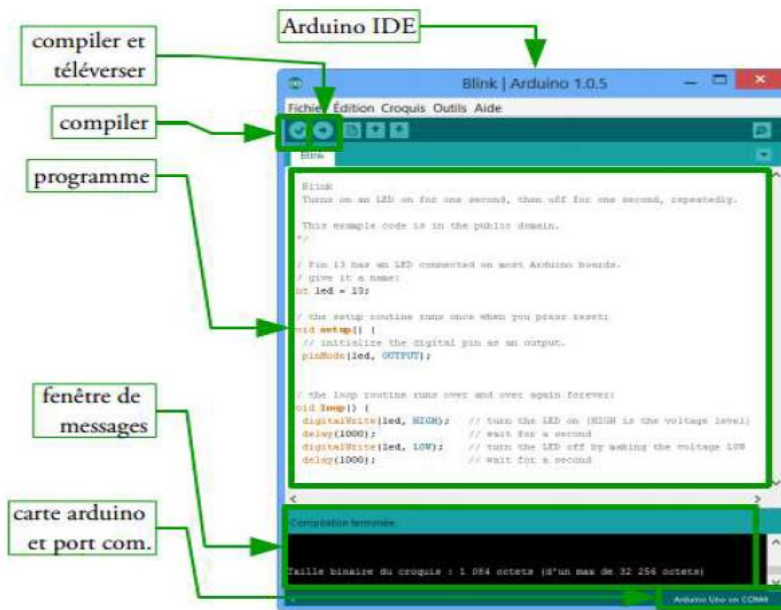


Figure 2.19: Fenêtre de l'IDE.

La fonction « setup () » est appelée au démarrage du programme. Cette fonction est utilisée pour initialiser les variables, le sens des broches, les bibliothèques utilisées. La fonction « setup () » n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

Après avoir créé une fonction « setup () », qui initialise et fixe les valeurs de démarrage du programme, la fonction « loop () » (boucle en français), fait exactement ce que son nom suggère et s'exécute en boucle sans fin, permettant au programme de s'exécuter et de répondre. On utilise cette fonction pour contrôler activement la carte Arduino [33].

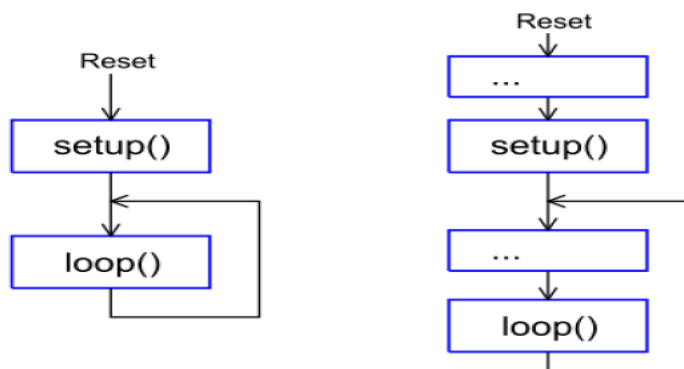


Figure 2.20: Les étapes de base du programme

### 2.3.1.3 Structure d'un programme Arduino

Les programmes Arduino peuvent être divisés en 3 parties : la Structure, les Valeurs (variables et constantes) et les Fonctions. Dans cet article, nous allons parler de la Structure qui correspond au squelette de notre programme Arduino.

Dans le jargon Arduino, un fichier Arduino est appelé « Sketch » en anglais et « Croquis » en français.

La Structure d'un programme Arduino est obligatoirement composée de ces deux fonctions principales :

- ✓ La fonction « setup () » ;
- ✓ La fonction « loop () ».

### 2.3.2 Logiciel Proteus ISIS

Il existe plusieurs logiciels de simulation des circuits électroniques tels que Workbench, PROTEUS, Tina...etc. Les premiers tests de simulation du circuit Radar sont faits sur Proteus ISIS, le logiciel fameux des simulations des montages électroniques. En plus de sa capacité de simuler des montages à base de microcontrôleur, il permet davantage de donner une idée sur la réalisation matérielle et la conception des circuits imprimés. Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'une carte Arduino et son interaction avec les composants qui l'entourent (capteur ultrason, écran LCD...ex).

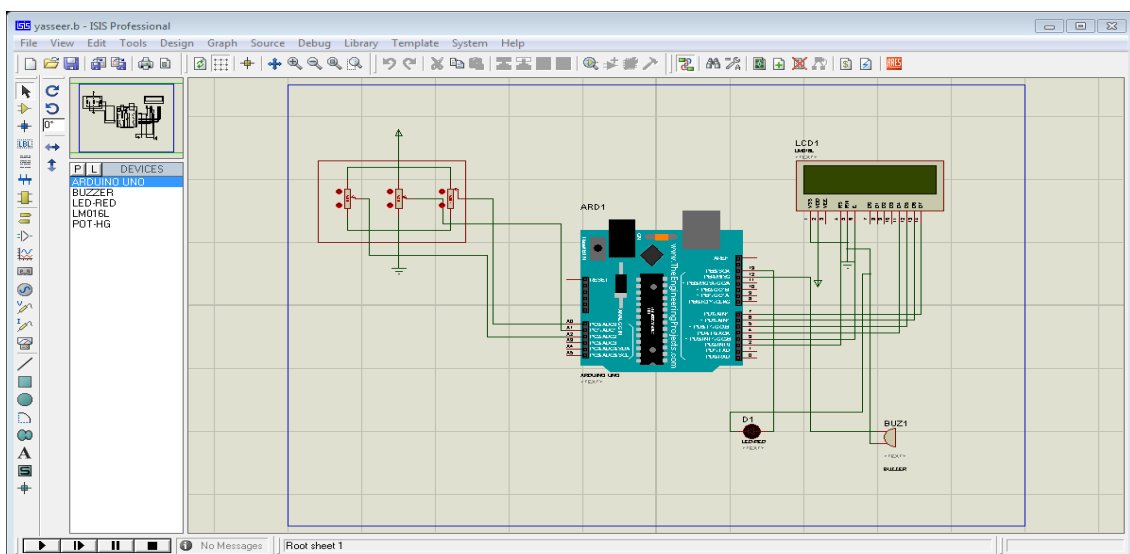


Figure 2.21: Simulation sur logiciel Proteus.

### 2.3.3 Le software utilisée "Processing"

Processing est une excellente source de création des graphiques, est un langage de programmation et une interface de développement intégré (IDE) écrite en Java.

Processing relie les concepts de logiciel aux principes de la forme visuelle, du mouvement et de l'interaction. Il intègre un langage de programmation, un environnement de Développement et une méthodologie D'enseignement dans un système unique. Processing a été créé pour enseigner les fondamentaux de la programmation informatique dans un contexte visuel, être utilisé comme outil de production pour des contextes spécifiques. Les étudiants les artistes, les professionnels du design et les chercheurs l'utilisent pour l'apprentissage, le prototypage et la production [34].



**Figure 2.22:** Logo du Processing.

Les programmes réalisés avec Processing peuvent donc fonctionner sur toute machine possédant un dispositif virtuelle Java ainsi que dans les navigateurs Web équipé du plugin Java.

Les bibliothèques de Processing étendent facilement la capacité d'envoyer / recevoir des données dans divers formats et importer et exporter des formats de fichiers 2D et 3D [34].

### 2.3.3.1 Les bases de l'interface de Processing

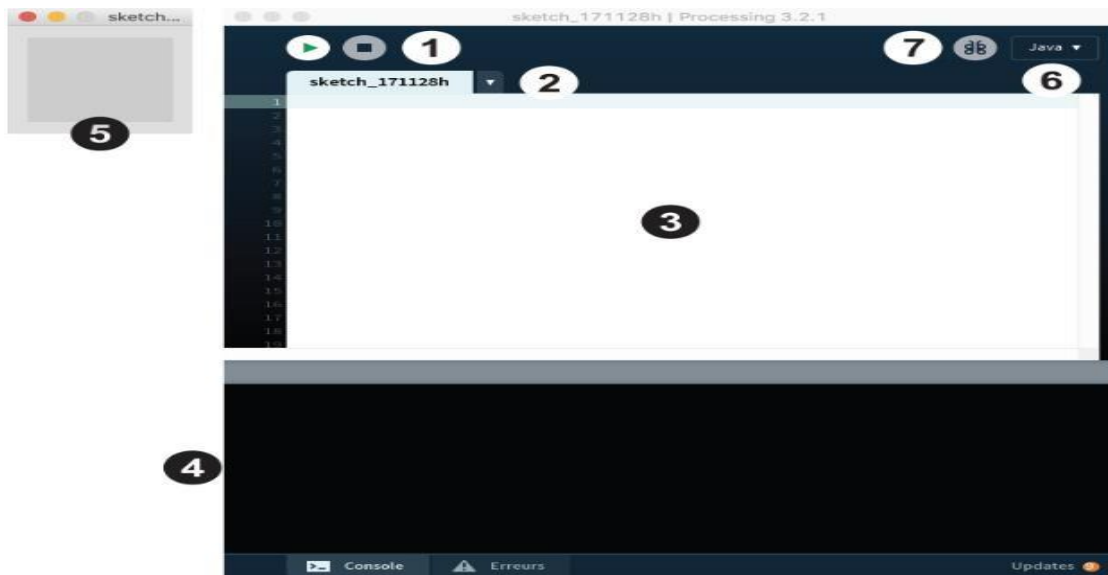
- **L'interface**

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes : La fenêtre principale dans laquelle vous allez créer votre projet et la fenêtre de visualisation dans laquelle vos créations (dessins, animations, vidéos) apparaissent.

On trouve plus précisément les éléments suivants dans l'interface [35] :

- 1- Barre d'actions.
- 2- Barre d'onglets.
- 3- Zone d'édition (pour y saisir votre programme).
- 4- Console, qui comprend un onglet les messages affichés par programme un onglet pour les erreurs. Cette console indique aussi si des mises à jour (« updates») sont disponibles pour les bibliothèques et les modes.
- 5- Fenêtre de visualisation (espace de dessin).
- 6- Liste déroulante pour les modes.
- 7- Bouton pour activer le mode debug (pas à pas).

La Figure 2.23 suivante indique bien l'emplacement de ces éléments.



**Figure 2.23:** l'interface de Processing.

- **Barre d'actions**



**Bouton "Run"** : exécute votre sketch (votre programme).



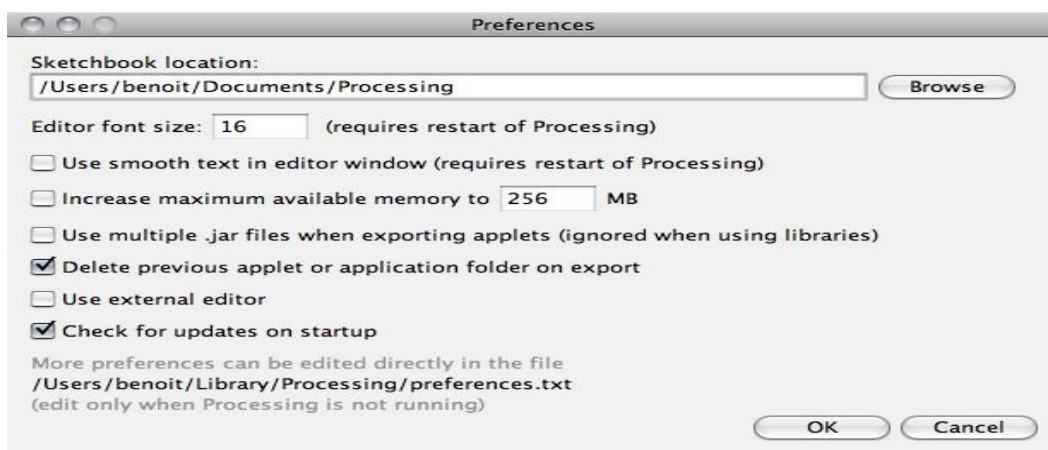
**Bouton "Stop"** : arrête l'exécution de votre sketch.

Processing permet de travailler dans plusieurs modes, un mode permettant de programmer dans un environnement spécifique à chaque plateforme visée (ex : application pour tablette Android, ...). Ces modes peuvent être gérés depuis une interface spécifique le « Contribution Manager ».

Vous pouvez changer ce mode à tout moment depuis l'interface, en ayant au préalable sauvegardé votre sketch.

- **Le dossier de travail**

C'est le dossier dans lequel seront enregistrés les sketches et les bibliothèques (des modules externes proposant des fonctionnalités supplémentaires). Par défaut ce dossier se nomme Processing et se trouve dans Documents (sous Mac) ou Mes Documents (sous Windows). Sous GNU/Linux, il est dans votre dossier personnel sous le nom de sketchbook. Pour modifier ce dossier, allez dans le menu Processing > Préférences. Dans la boîte de dialogue qui apparaît, il suffit de cliquer sur Naviguer pour choisir le dossier qui convient [36].



**Figure 2.24:** La fenêtre « Préférence » de Processing.

### 2.3.3.2 Communication entre la carte Arduino & logiciel Processing

L'IDE de Processing est similaire à Arduino en termes de structure. Il a des fonctions de configuration et des fonctions de dessin comme un Arduino IDE a une fonction de configuration et de boucle. L'IDE de traitement peut communiquer avec l'IDE Arduino via une communication en série. De cette façon, nous pouvons envoyer des données de l'Arduino à l'IDE de Processing et aussi de l'IDE de Processing à l'Arduino. Le déroulement de la communication entre le logiciel Arduino IDE et le Processing, se fait à travers une liaison série half duplex, les données sont téléversés vers la carte Arduino où se passe le traitement ensuite les résultats de traitement seront transférés vers le logiciel Processing sous la même liaison, voir Figure 2.25.

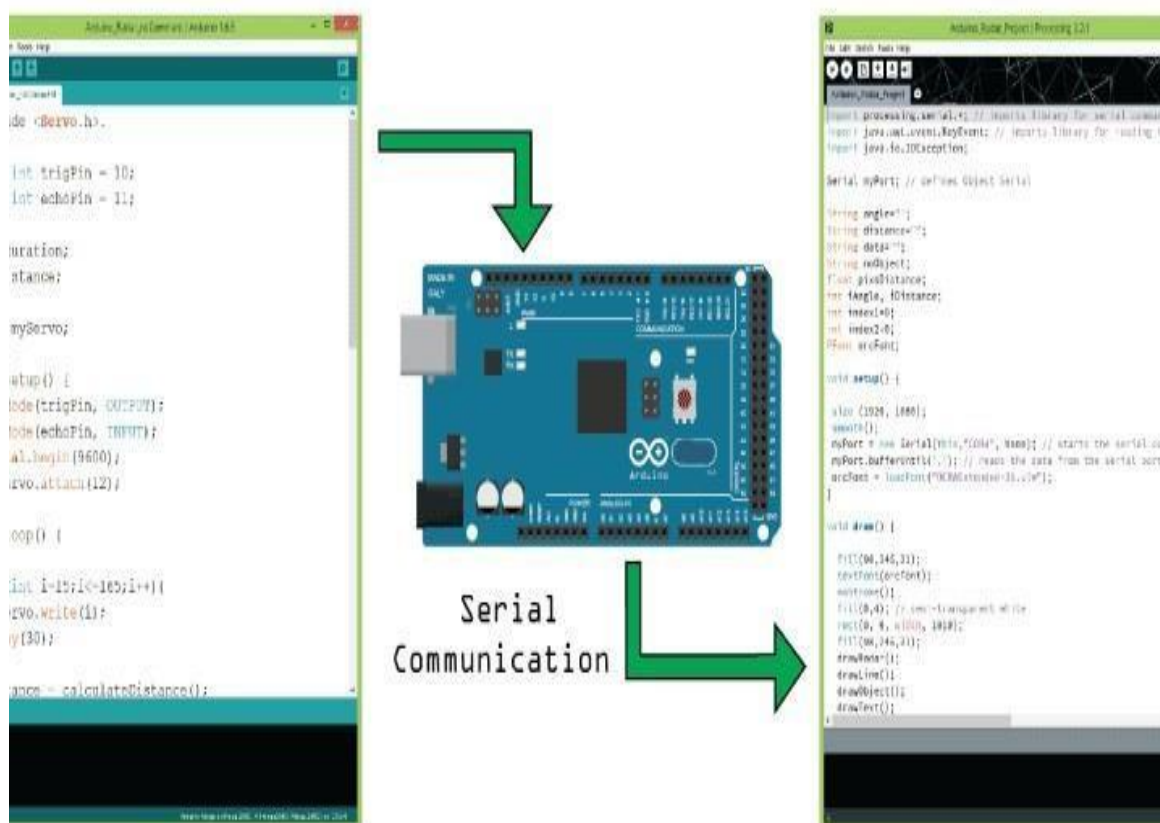
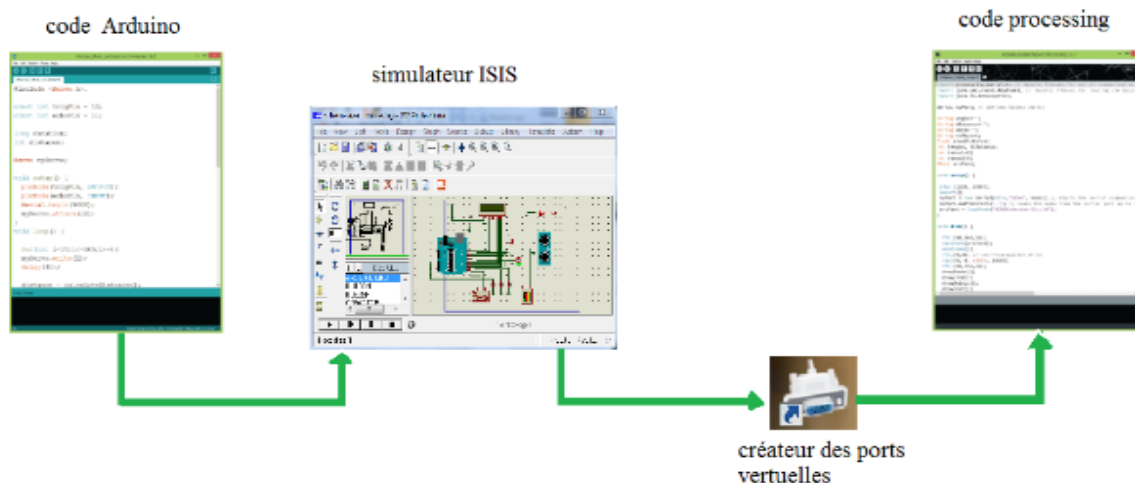


Figure 2.25 : Communication entre Arduino & Processing.

Dans le cas de la simulation, et après la compilation de code Arduino, il faut exporter les données de type binaire en hexadécimale pour l'exécuter par la carte Arduino au niveau de simulateur ISIS, cette dernière va traiter les données reçues puis elle va orienter les

résultat obtenues vers les processing mais il n'existe pas une liaison entre eux, pour cela il faut créer une liaison virtuelle à l'aide de logiciel «Virtual Serial Port Driver » en identifiant deux ports l'un à côté de processing et l'autre pour la carte Arduino d'une part et d'autre part il faut rajouter un adaptateur pin-port à côté de la carte Arduino pour être prêt d'émettre des données à travers cette liaison. La communication entre Proteus et Processing est présentée par la figure suivante:



**Figure 2.26 :** Communication entre PROTEUS et PROCESSING.

## 2.4 Conclusion

Dans ce chapitre, nous avons vu les principaux composants utilisés dans notre projet basé sur la carte d'acquisition Arduino Uno ainsi que les raisons du choix de l'accéléromètre ADXL 335, LCD, Buzzer et LED où nous avons discuté de l'explication de la description détaillée de l'IDE Arduino, ainsi que le programme de traitement Processing et la relation entre eux .

Dans le prochain chapitre, nous allons conduire le projet et étudier l'expérience de détection de tremblement de terre.

# **Chapitre 3 :**

## **Simulations et tests pratiqués**



### 3.1 Introduction

Dans ce chapitre, nous allons expliquer en détail la partie expérimentale de notre projet. Il s'agit d'un appareil capable de détecter et de prédire les tremblements de terre à l'aide du module de capteur accéléromètre ADXL335 GY-61.

Nous allons assembler le matériel, et présenterons également les aspects du logiciel Arduino IDE et Processing ainsi que le logiciel Proteus ISIS.

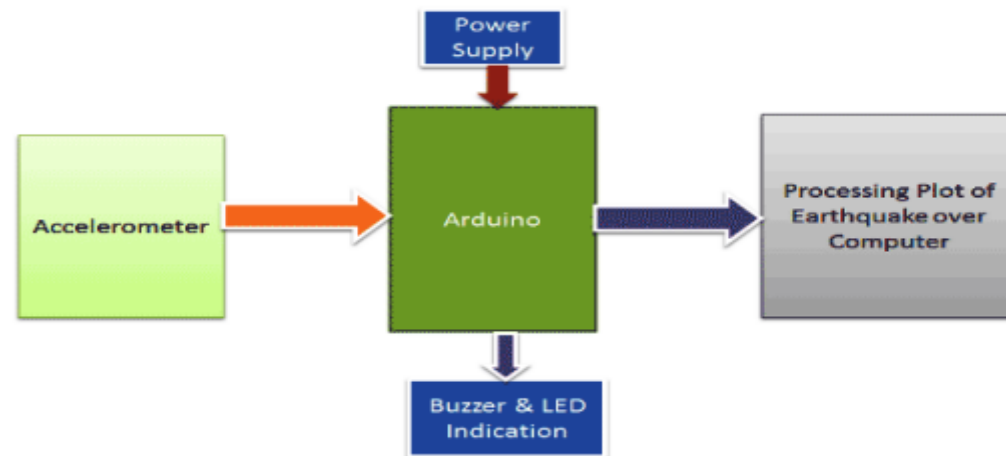


Figure 3.1 : Schéma fonctionnel du travail proposé.

### 3.2 Description du schéma fonctionnel

Le fonctionnement de ce détecteur de tremblement de terre est simple. Comme nous l'avons mentionné précédemment, nous avons utilisé l'accéléromètre pour détecter les vibrations des tremblements de terre le long de l'un des trois axes de sorte que chaque fois que des vibrations se produisent, l'accéléromètre détecte les vibrations et les convertit en valeur ADC équivalente. Ensuite, ces valeurs ADC sont lues par Arduino et affichées sur l'écran LCD 16x2.

Nous devons d'abord calibrer l'accéléromètre en prélevant des échantillons de vibrations environnantes chaque fois qu'Arduino se met sous tension. Ensuite, nous devons soustraire ces valeurs d'échantillon des lectures réelles pour obtenir les lectures réelles. Cet étalonnage est nécessaire pour qu'il n'affiche pas d'alertes concernant ses vibrations environnantes normales. Après avoir trouvé des lectures réelles, Arduino compare ces valeurs avec des valeurs max et min prédéfinies. Si Arduino constate que les valeurs de changement sont supérieures ou inférieures aux valeurs prédéfinies de n'importe quel axe dans les deux sens (négatif et positif), Arduino déclenche le buzzer et affiche l'état de l'alerte sur l'écran

LCD 16x2 et une LED est également allumée. Nous pouvons ajuster la sensibilité du détecteur de tremblement de terre en modifiant les valeurs prédéfinies dans le code Arduino.

### 3.3 Déroulement du projet

La pratique a été faite en deux parties :

- ✓ La première partie est la conception assistée par ordinateur (CAO), la simulation avec PROTEUS.
- ✓ La deuxième partie est la réalisation, le montage pratique.

Pour les deux parties simulation et réalisation, on passe par :

- ✓ Programmation de la carte Arduino (ATMega328).
- ✓ Affichez les résultats du graphique qui se trouvent sur l'écran de l'ordinateur à l'aide du logiciel Processing.

## 3.4 Simulation sur le logiciel Proteus

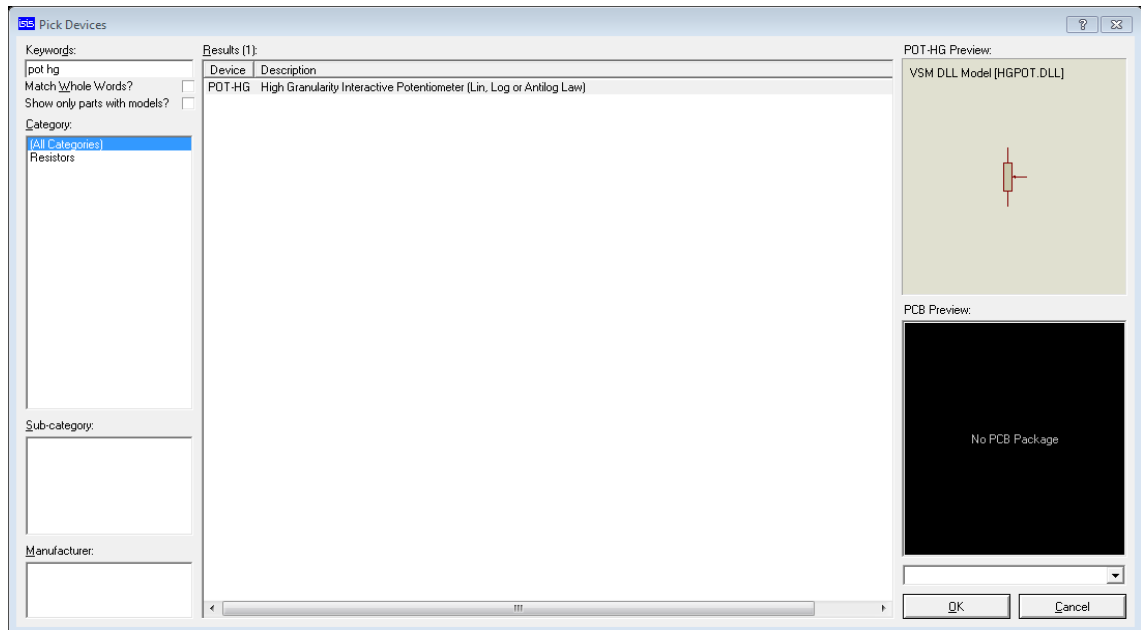
### 3.4.1 Présentation

Avant le début de la réalisation du dispositif il faut passer par la simulation, car elle nous informe sur les risques à éviter (protéger nos composants des dommages). Dans cette simulation, on va travailler avec les programmes mentionnés au chapitre 2. Nous allons donc simuler le circuit dans Proteus et envoyer les données via un port série virtuel pour enfin l'afficher sous forme d'un graphe.

### 3.4.2 Démarche de la simulation

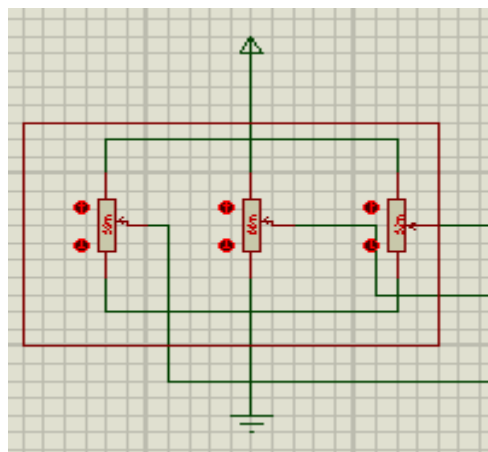
#### 3.4.2.1 Bibliothèque Accéléromètre ADXL335

L'accéléromètre ADXL335 n'étant pas sur le Proteus, on peut le faire en allant chercher l'élément Pot-HG comme le montre la figure suivante :



**Figure 3.2 :** Recherche d'articles Pot-HG par Proteus.

Nous introduisons le type de 3 éléments de POT-HG puis nous associons à VCC et GND et cela forme pour nous le dispositif Accéléromètre ADXL335 comme le montre la figure suivante :



**Figure 3.3 :** Accéléromètre a par Proteus.

3.4.3 Circuit de simulation sous Proteus

La (Figure 3.4) suivante représente le circuit global de notre simulation sous Proteus.

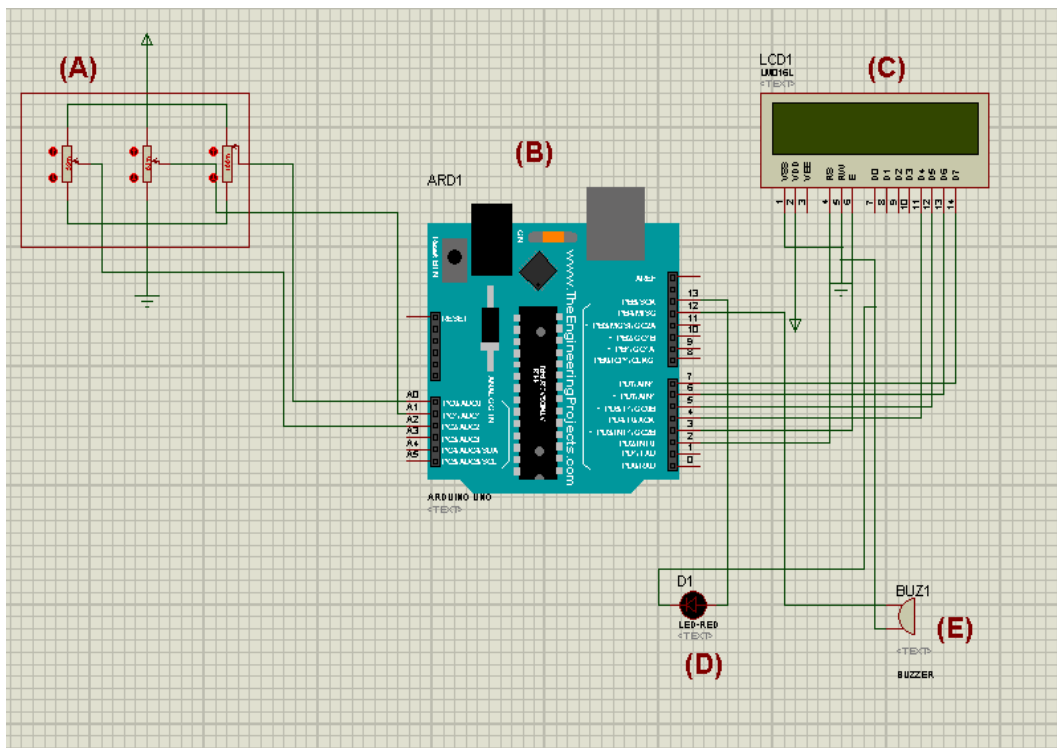


Figure 3.4 : Circuit électronique d’une pratique.

Les notations aperçues dans la Figure 3.5 sont expliqués dans le tableau ci-dessous :

|     |                       |
|-----|-----------------------|
| (A) | Accéléromètre ADXL335 |
| (B) | Carte Arduino UNO     |
| (C) | Afficheur LCD         |
| (D) | LED                   |
| (E) | Buzzer                |

Tableau 3.1 : Les différents composants utilisés dans la simulation.

Nous écrivons le code sur l'IDE Arduino UNO et extrayons le code via l'Arduino UNO de Proteus, puis nous lisons et écrivons sur l'écran LCD de l'appareil écrivons le mot (Bonjour...) comme indiqué dans la figure suivante :

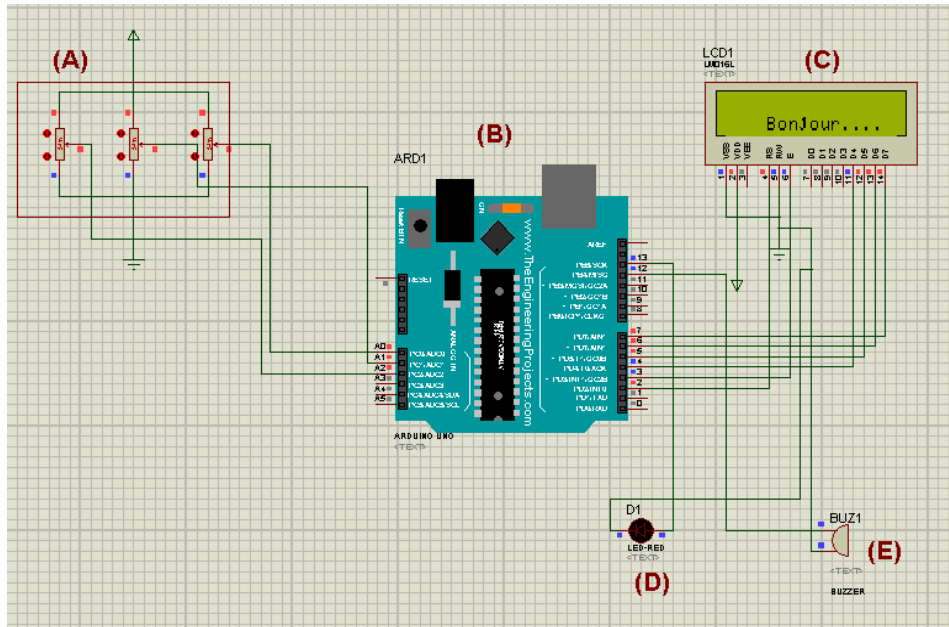


Figure 3.5 : L'exécution de la pratique a par Proteus.

Nous effectuons également des modifications via l'accéléromètre ADXL335 et effectuons des modifications X, Y ainsi que Z via l'appareil POT-HG

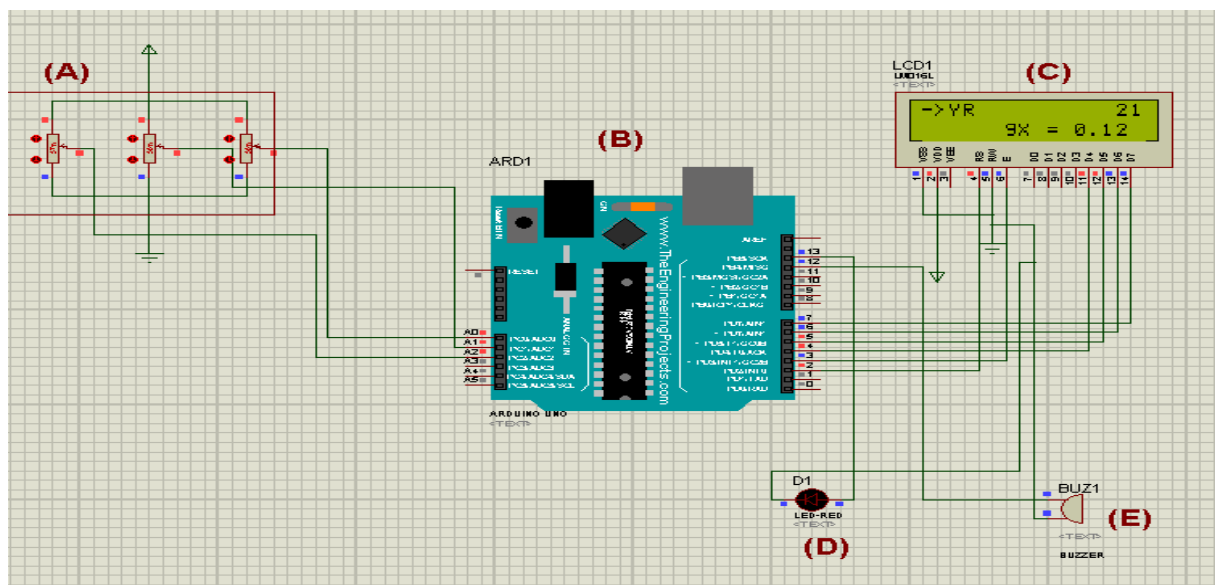


Figure 3.6 : L'exécution de la pratique a par Proteus.

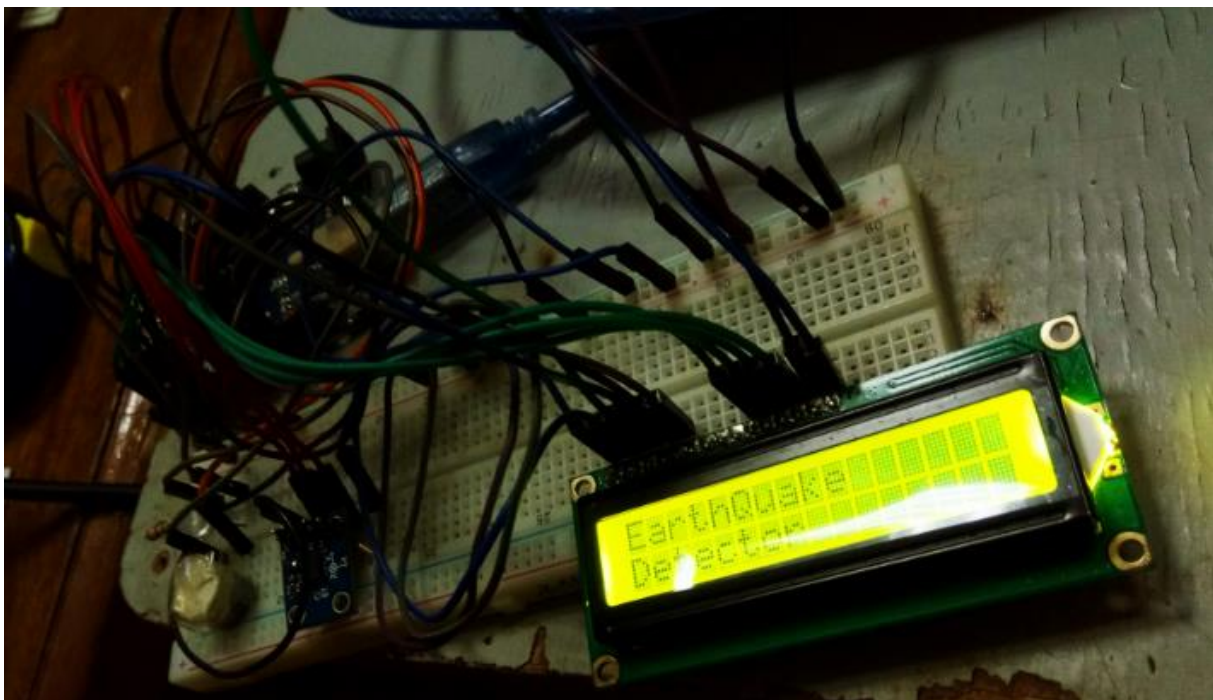
### 3.5 Réalisation du projet

#### 3.5.1 Composants utilisés

Les principaux composants du projet sont :

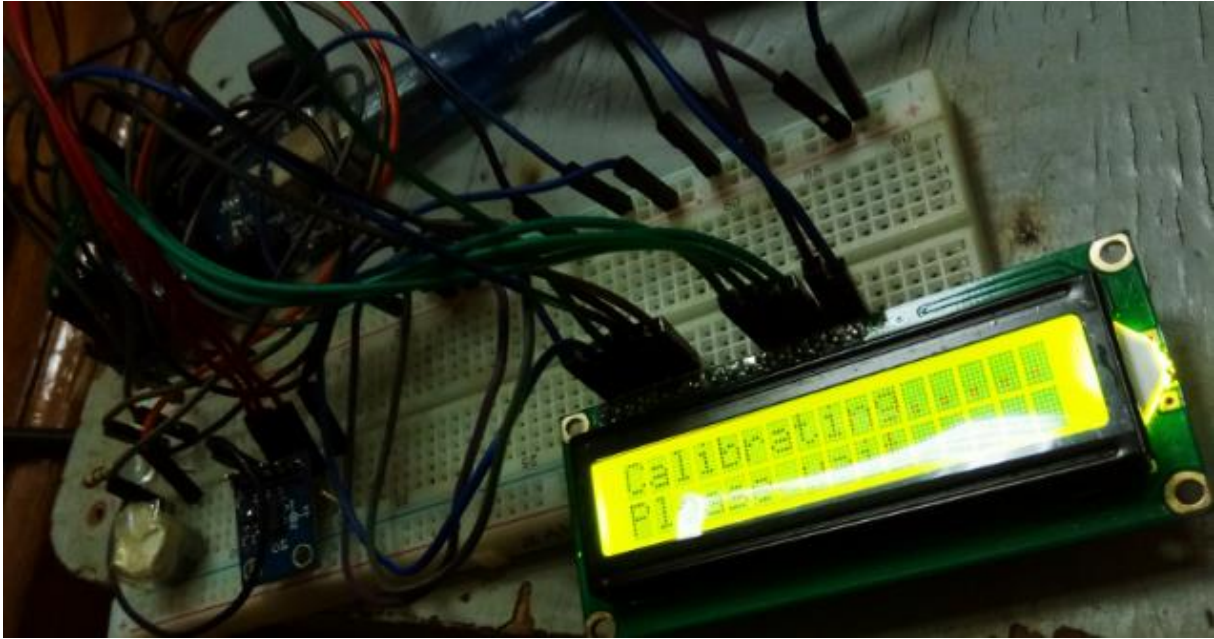
- ✓ Accéléromètre ADXL335
- ✓ Carte Arduino UNO
- ✓ L'afficheur LCD
- ✓ LED
- ✓ Buzzer

Après avoir écrit le code via le programme Arduino IDE et nous envoyons le code à l'appareil Carte Arduino UNO et nous implémentons le projet et installons les appareils. Nous remarquons qu'il écrit ("EarthQuake " "Detector ") sur l'appareil L'afficheur LCD, puis écrit le mot ("Calibrating....." "Please wait...») puis nous donnés les coordonnées (X), (Y) et (Z) d'après les trois images suivantes :

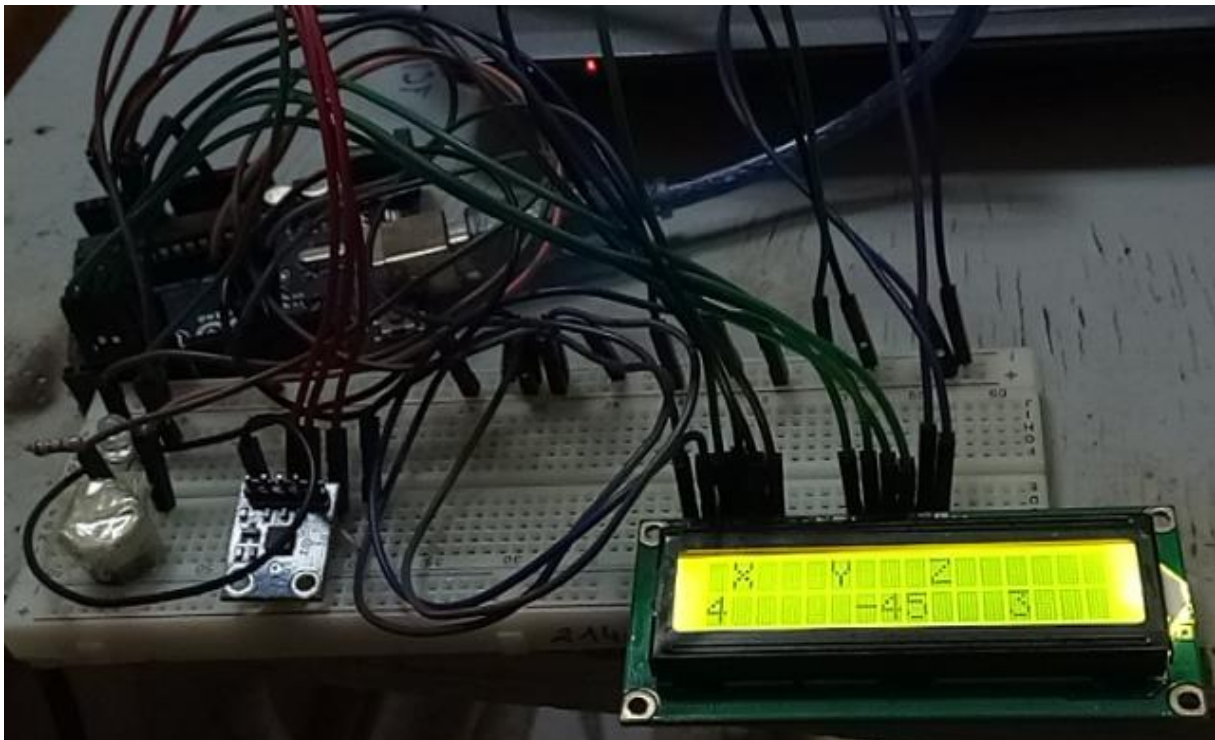


**Figure 3.7 :** Projet de travail d'écriture de mots ("EarthQuake" "Detector") sur l'écran LCD.





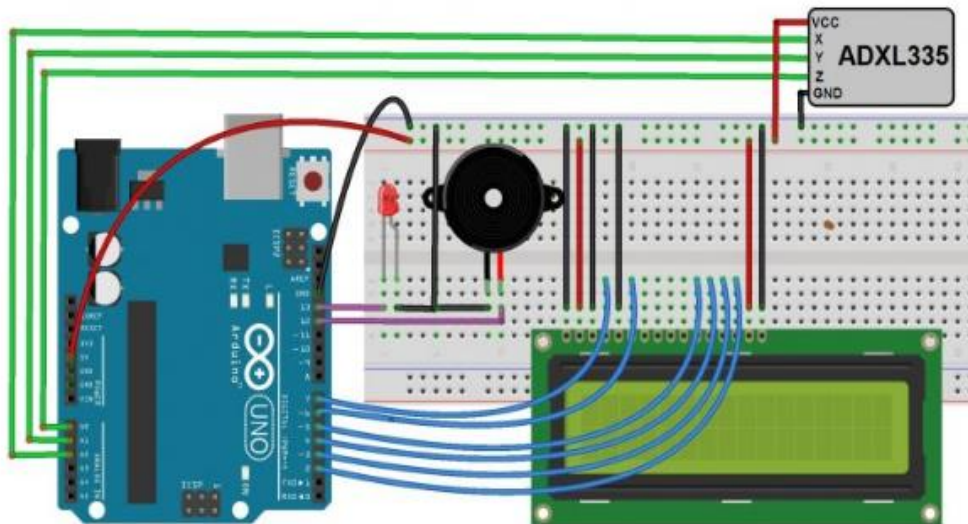
**Figure 3.8:** Projet de travail d'écriture de mots ("Calibrating....." "Please wait...") sur l'écran LCD.



**Figure 3.9:** Projet de travail d'écriture de mots ("X Y Z") sur l'écran LCD.

### 3.5.2 Alimentation et connexion des broches

La sortie stable (+5V) du port USB de l'ordinateur est utilisée pour alimenter tous les appareils et capteurs Arduino et Accéléromètre ADXL335 et L'afficheur LCD. Les broches des composants sont connectées selon le schéma suivant (Figure 3.10), le programme est compilé sous l'IDE Arduino et le programme est téléchargé sous forme de fichier HEX sur la carte pour exécution via un câble USB.



**Figure 3.10** : Schéma de circuit et connexions.

### 3.5.3 Explication de la programmation

Dans ce bouclier Arduino Uno de détecteur de tremblement de terre, nous avons créé deux codes : un pour Arduino pour détecter un tremblement de terre et un autre pour Processing IDE pour tracer les vibrations du tremblement de terre sur le graphique sur ordinateur. Nous allons découvrir les deux codes un par un :

#### A. un code Arduino

Tout d'abord, nous étalonnons l'accéléromètre par rapport à sa surface de pose, afin qu'il n'affiche pas d'alertes par rapport à ses vibrations environnantes normales. Dans cet étalonnage, nous prélevons quelques échantillons, puis nous en faisons une moyenne et les stockons dans une variable. Désormais, chaque fois que l'accéléromètre prendra des lectures, nous soustrairons ces valeurs d'échantillon des lectures afin qu'il puisse ignorer les vibrations environnantes. Ensuite, Arduino compare ces valeurs calibrées (soustraites) avec des limites prédéfinies. Et agit en conséquence. Si les valeurs sont supérieures aux valeurs prédéfinies, le



buzzer émettra un bip et tracera le graphique de vibration sur l'ordinateur à l'aide du traitement.

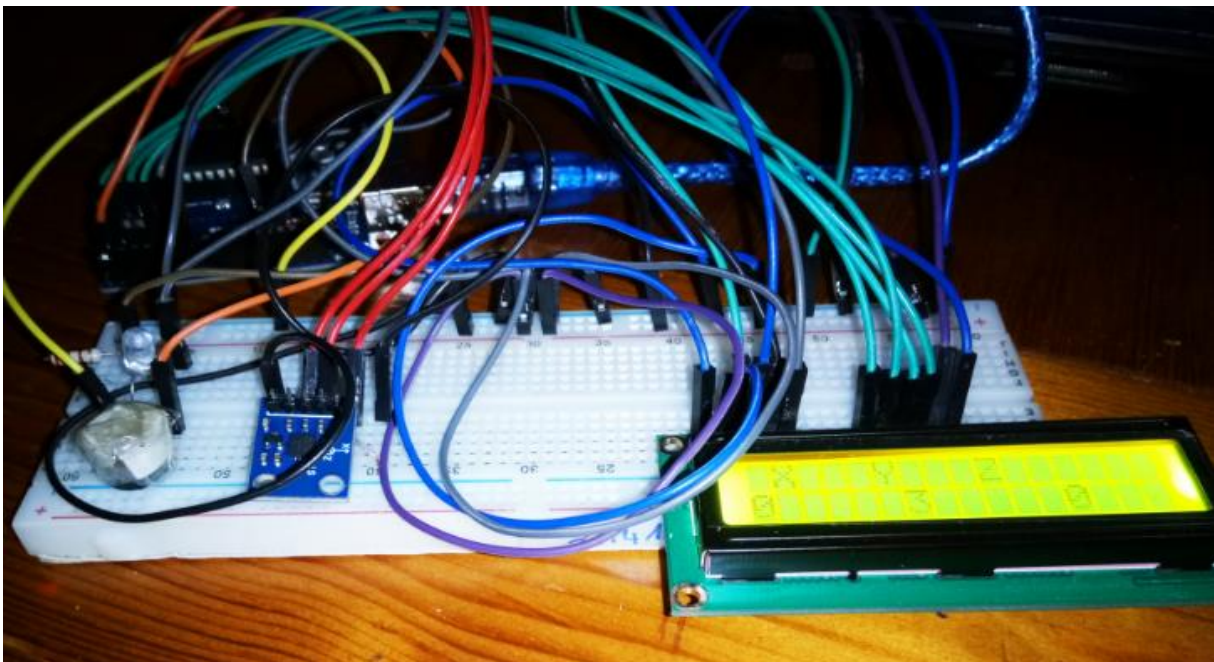
### B. Code de Processing

Nous avons conçu un graphique utilisant Processing, pour les vibrations du tremblement de terre, dans lequel nous avons défini la taille de la fenêtre, les unités, la taille de la police, l'arrière-plan et la lecture et l'affichage des ports série, l'ouverture du port série sélectionné, etc. Dans la fonction ci-dessous, nous avons reçu données du port série et extraire les données requises, puis les mapper avec la taille du graphique. Après cela, nous avons tracé l'espace unitaire, les limites max et min, les valeurs des axes x, y et z. Après cela, nous traçons les valeurs sur le graphique en utilisant 3 couleurs différentes comme le bleu pour la valeur de l'axe des x, la couleur verte pour l'axe des y et z est représenté par la couleur rouge.

### 3.5.4 Principe de fonctionnement

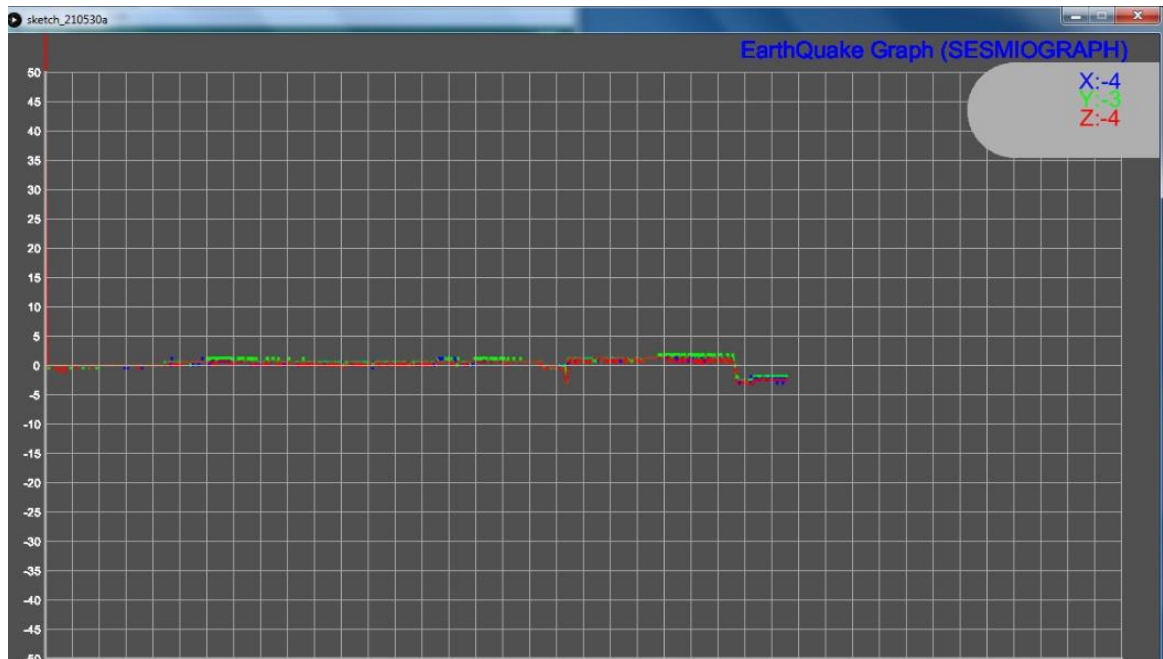
- Première opération

Après avoir écrit le code dans Arduino IDE et l'avoir envoyé via USB, la Carte Arduino UNO lit et traduit le code et commence à travailler avec l'appareil L'afficheur LCD ainsi qu'avec l'appareil Accéléromètre ADXL335, puis commence à fonctionner comme il est enregistré pour nous au début de l'expérimentation via Les coordonnées X, Y et Z dans le cas du repos restent constantes comme le montre l'image suivante :



**Figure 3.11** : Projet de travail en sommeil.

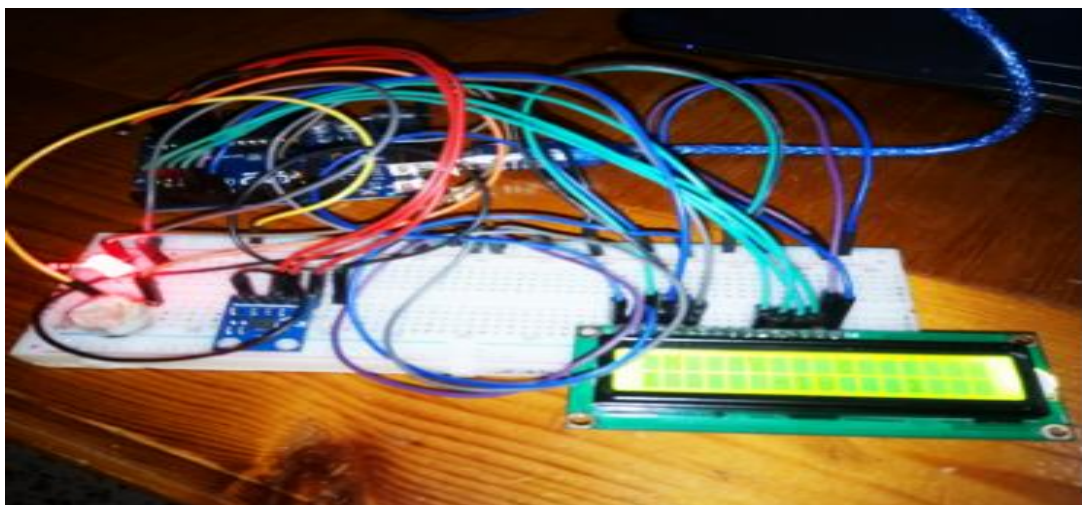
On constate qu'il y a une stabilité sur le graphe dans le programme de PROCESSING et cela est dû à l'immobilité des appareils électroniques (à l'état de repos).



**Figure 3.12 :** Graphique à travers les axes X, Y et Z à travers le système de traitement (au repos).

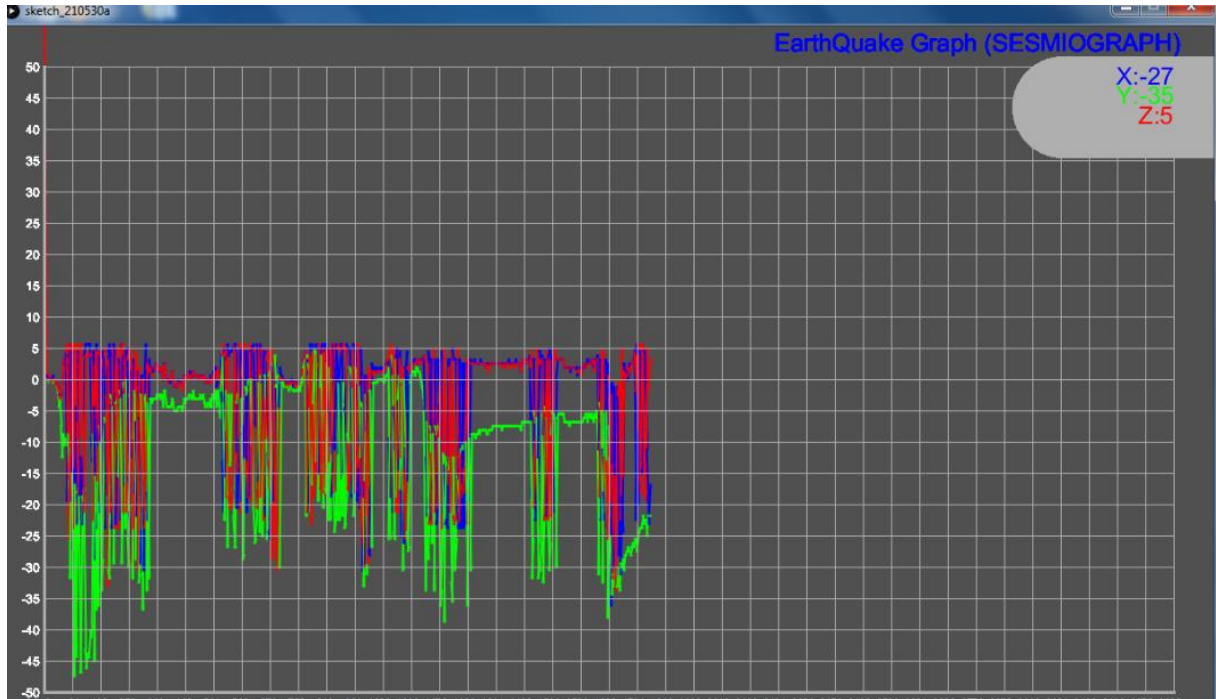
- **Deuxième opération**

Dans la deuxième étape, nous déplaçons et faisons vibrer les appareils sur différents axes et coordonnées X, Y et Z, le bip retentit de l'appareil Buzzer, et l'appareil LED s'allume et modifie les coordonnées X, Y ainsi que Z (en le cas du mouvement).



**Figure 3.13 :** Travail de projet en mouvement (vibration).

Nous remarquons la différence et la variance lors du déplacement des appareils, car le mouvement et le changement d'axes et de coordonnées X, Y et Z sont enregistrés pour nous dans le système PROCESSING et le graphique qu'il contient constitue la grande différence entre les coordonnées comparées à la première étape et cela est dû à la vibration et au mouvement de ce projet.



**Figure 3.14** : Graphique à travers les axes X, Y et Z à travers le système de traitement (en cas de mouvement).

### 3.5.5 Résultats et discussion

Les graphiques présentés précédemment indiquent l'alerte au tremblement de terre sur les axes X, Y et Z. En l'absence de vibration au sol, les valeurs sont aux axes zéro. Si le mouvement est suffisamment violent lors du séisme et dépasse un certain seuil, c'est-à-dire que le maximum atteint (20) au niveau de l'axe des abscisses (couleur bleue), la LED s'allume, un buzzer retentit et un message d'alerte apparaît sur l'écran LCD filtré. Le sismographe basé sur Arduino a été testé avec un accéléromètre et il fonctionne de manière satisfaisante. Il a été constaté que tous les composants du système fonctionnent correctement. La détection pré-séisme à l'aide d'un accéléromètre et d'un message d'alerte à une heure prédéfinie sera réalisée avec succès. L'accéléromètre ADXL335 détecte les vibrations pré-séisme et donne une tension analogique équivalente à l'accélération imposée sur les axes X, Y et Z.

Les trois sorties analogiques sont appliquées aux broches Arduino Uno. Toute accélération due au mouvement dans l'un des axes est détectée par l'accéléromètre et donc par l'Arduino Uno. Si le mouvement est suffisamment violent pendant le séisme et dépasse un certain seuil, la LED s'allume, un buzzer retentit, et il affiche également un message d'alerte sur l'écran LCD et le graphique peut être tracé à l'aide de l'IDE pour le traitement logiciel.

### **3.6 Conclusion**

Dans ce chapitre, nous avons fait l'expérience de simulation via le programme Proteus, puis nous effectuons les travaux pratiques via le programme de Processing à l'aide d'appareils électroniques, où le programme nous permet de donner un graphique qui nous permet d'étudier les tremblements de terre.

Grâce aux travaux pratiques, nous concluons que nous pouvons prédire et détecter les tremblements de terre grâce à l'expérience que nous avons faite.

# **Conclusion générale**

### Conclusion générale

Avec le développement de la science moderne, il était nécessaire d'essayer de gérer l'effet destructeur des tremblements de terre et de trouver des moyens de réduire leur impact destructeur, et cela a été représenté par des stations de surveillance sismique qui tentent d'estimer le moment des tremblements de terre sur la base de nombreux indicateurs terrestres, tectoniques ou volcaniques, ainsi que la présence d'appareils de mesure et de surveillance sismiques à travers lesquels les séismes sont classés en fonction de l'impact destructeur sur une échelle appelée échelle sismique de Richter.

À la fin de la rédaction d'un sujet sur les tremblements de terre, une personne doit traiter les tremblements de terre comme l'une des manifestations du pouvoir de Dieu Tout-Puissant, car certains tremblements de terre atteignent l'intensité de leur destruction au point de pouvoir changer le visage de villes entières en quelques instants. , qui indique la grandeur de Dieu Tout-Puissant, et sa capacité à diriger cette vie comme il veut et quand il veut, Gloire à Dieu, le Grand Créateur de l'univers.

Ce travail Détection de tremblement de terre basée sur Arduino à l'aide d'un accéléromètre, est développé avec succès et fonctionne de manière satisfaisante. Ainsi pour résumer nous avons introduit ce produit dans le but de réduire les destructions causées par le tremblement de terre en alertant les gens. Il est économique et son prix est indiqué de manière à ce qu'il soit abordable pour chaque individu. Nous avons présenté une nouvelle technique pour résoudre le problème de détection automatique et de classification des tremblements de terre en une seule étape en utilisant un dispositif de détection de tremblement de terre basé sur Arduino.

. Ce travail utilise un accéléromètre pour détecter les vibrations pré-sismiques. L'accéléromètre est très sensible aux vibrations et aux vibrations sur les trois axes. L'avantage de construire un détecteur de tremblement de terre basé sur Arduino avec un accéléromètre est de réduire les pertes destructrices.

En plus de l'expérimentation, les travaux pratiques que nous avons réalisés permettent de détecter des séismes et d'étudier le type de séismes connus à travers le mouvement de la Terre.



### BIBLIOGRAPHE

- [1] ChandamThousaina Singh, Ngangbam Michael Sing, "Anti-Theft Security System using Seismic Sensor ADXL335 (GY-61)", IJESC, (Vol. 9, No. 7), July 2019.
- [2] A.R.Jadha, Gaurav. S. Gaikwad, Amol.B. Chavan, Kavaljeet. K. Tivale, Aniket. D. Patil "ARM Controller Based Earthquake & Flood Detection and Alert System", International Journal of Advanced Research in Engineering & Management (IJAREM), Vol. 3, No. 04, 2017.
- [3] M. Terrier, G. Bertrand, and A. Roullé. "Document d'information sur le risque sismique a destination des médias." Rapport BRGM/RP-55012-FR (2006).
- [4] Bisch. Philippe, Jacques. Betbeder-Matibet, and Jean-Louis. Doury, Constructions parasismiques, Ed. Techniques Ingénieur, 1995.
- [5] Garcia, J. F. "Rapport final de PFE." Elaboration D'une Méthode D'évaluation De La Vulnérabilité Sismique Des Bâtiments. INSA-Strasbourg (2007).
- [6] LOUNAS Boualem, "Analyse du comportement des structures élancées en plan", Mémoire de fin d'étude de master, Université Mouloud Mammeri TIZI-OUZOU, 2012.
- [7] AMOURI Chahinez, Mémoire de magister en génie civil, Construction des spectres de réponse pour la région de Constantine, Université Mentouri de Constantine(2005).
- [8] M.BADAOU, B.BENAHMED, H.TAREK, D.HAMZA, "Effets de caractéristiques sismiques sur les spectres de réponse", Mémoire de Fin d'Etudes, Université Ziane Achour de Djelfa, 2017.
- [9] LAHOUAL, Abderahmane, and Lakhdar GUERROUD. Effets de caractéristiques sismiques sur les spectres de réponse. Diss. 2017.
- [10] Belmouden, Youssef. "Analyse spectrale non itérative des oscillateurs simples sous l'action des tremblements de terre." Bulletin de l'institut scientifique, Rabat, section Sciences de la Terre 26: 1-10. (2004).
- [11] Walid Benlahcene : Un éclairage redondant, mémoire de fin d'étude pour d'ingénieur d'état en instrumentation, Université Batna, (2007).
- [12] <https://www.arduino-france.com/tutoriels/quest-ce-que-arduino/>.(18/07/2021)
- [13] <https://store.arduino.cc/arduino-uno-rev3>.(19/07/2021)
- [14] [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php).(22/07/2021)

## BIBLIOGRAPHE

- [15] [http://www.monclubelec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.Mat%C3%A9rielUno](http://www.monclubelec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Mat%C3%A9rielUno) (22/07/2021)
- [16] <https://www-lisic.univ-littoral.fr/~hebert/microcontroleur/atmel/>.(23/07/2021)
- [17] C. Tavernier, Arduino Applications avancées, Dunod, ISBN 978-2-10 058205-1, Paris 2012
- [18] L.GUENFOUD, C.MOUZARINE, "Conception et réalisation d'un système de contrôle à distance de la qualité de l'air", Mémoire de fin d'études, UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU, (2017/2018).
- [19] <https://how2electronics.com/arduino-earthquake-detector-accelerometer/>(01/09/2021)
- [20] <https://pdf1.alldatasheet.com/datasheet-pdf/view/1181923/ETC/GY-61.html>(01/09/2021)
- [21] <https://lastminuteengineers.com/adxl335-accelerometer-arduino-tutorial/>(02/09/2021)
- [22] <https://robu.in/accelerometer-sensor-arduino/>(03/09/2021)
- [23] BELHOCINE Zakia, KHELIFA OUI Kamelia, "Conception et réalisation de la carte de commande d'une tournette", Mémoire de Fin d'Etudes De MASTER ACADEMIQUE, (le 13 juillet 2016).
- [24] <https://www.aurel32.net/elec/lcd.php>(04/09/2021)
- [25] <https://sciencesappliquees.com/noncato/346-cours-les-afficheurs-lcd>(04/09/2021)
- [26] [https://leclairage.fr/led/#Presentation\\_des\\_LED](https://leclairage.fr/led/#Presentation_des_LED)(04/09/2021)
- [27] [https://media.xpair.com/auxidev/nD43a\\_LED.pdf](https://media.xpair.com/auxidev/nD43a_LED.pdf)(05/09/2021)
- [28] [https://dl.ummtto.dz/bitstream/handle/ummtto/6260/ZerroukiMohamedAmine\\_NesnasRiadh.pdf](https://dl.ummtto.dz/bitstream/handle/ummtto/6260/ZerroukiMohamedAmine_NesnasRiadh.pdf)(05/09/2021)
- [29] <https://www.theengineeringprojects.com/2018/10/introduction-to-arduinoide.html> (06/09/2021)
- [30] A. KIHAL, I. BOULASSEL, "Générateur de cartes à ultrasons" mémoire de fin d'études Université de Jijel 2020.
- [31] [https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquée/742\\_decouverte-de-larduino/3416\\_le-logiciel/](https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquée/742_decouverte-de-larduino/3416_le-logiciel/) (07/09/2021)



## BIBLIOGRAPHE

[32] P. PUJADES « Utilisation d'Ardu Block : Programmation Arduino », Mars 2015

[33] [https://www.mon-fablab.fr/arduino/structure/fonctions\\_de\\_base/loop.\(07/09/2021\)](https://www.mon-fablab.fr/arduino/structure/fonctions_de_base/loop.(07/09/2021))

[34] Reas and Ben Fry. Processing: A Programming Herd-book for Visual Designers and Artists (Second Edition) The MIT Press. Publié Décembre 2014.

[35] POORAZIZI, Ebrahim et HUNTER, Andrew JS. " Evaluation of web processing service frameworks". OSGeo Journal, vol. 14, no 1, p. 29-42, 2015.

[36] Makni, Aida. Fusion de données inertielles et magnétiques pour l'estimation de l'attitude sous contrainte énergétique d'un corps rigide accéléré. Diss. Université Grenoble Alpes, 2016.

## ANNEXE A

### Code Arduino

```
#include <LiquidCrystal.h> // lcd Header

LiquidCrystal lcd(7,6,5,4,3,2); // pins for LCD Connection

#define buzzer 12 // buzzer pin

#define led 13 //led pin

#define x A0 // x_out pin of Accelerometer

#define y A1 // y_out pin of Accelerometer

#define z A2 // z_out pin of Accelerometer

/*variables*/

int xsample=0;

int ysample=0;

int zsample=0;

long start;

int buz=0;

/*Macros*/

#define samples 50

#define maxVal 5 // max change limit

#define minVal -5 // min change limit

#define buzTime 5000 // buzzer on time

void setup()

{

  lcd.begin(16,2); //initializing lcd

  Serial.begin(9600); // initializing serial

  delay(1000);

  lcd.print("EarthQuake ");

  lcd.setCursor(0,1);
```

```
lcd.print("Detector ");
delay(2000);
lcd.clear();
lcd.print("Calibrating.....");
lcd.setCursor(0,1);
lcd.print("Please wait...");
pinMode(buzzer, OUTPUT);
pinMode(led, OUTPUT);
buz=0;
digitalWrite(buzzer, buz);
digitalWrite(led, buz);
for(int i=0;i<samples;i++) // taking samples for calibration
{
xsample+=analogRead(x);
ysample+=analogRead(y);
zsample+=analogRead(z);
}
xsample/=samples; // taking avg for x
ysample/=samples; // taking avg for y
zsample/=samples; // taking avg for z
delay(3000);
lcd.clear();
lcd.print("Calibrated");
delay(1000);
lcd.clear();
lcd.print("Device Ready");
delay(1000);
```

```
lcd.clear();

lcd.print(" X  Y  Z ");

}

void loop()

{

int value1=analogRead(x); // reading x out

int value2=analogRead(y); //reading y out

int value3=analogRead(z); //reading z out

int xValue=xsample-value1; // finding change in x

int yValue=ysample-value2; // finding change in y

int zValue=zsampl-value3; // finding change in z

/*displying change in x,y and z axis values over lcd*/

lcd.setCursor(0,1);

lcd.print(xValue);

lcd.setCursor(6,1);

lcd.print(yValue);

lcd.setCursor(12,1);

lcd.print(zValue);

delay(500);

/* comparing change with predefined limits*/

if(xValue < minVal || xValue > maxVal || zValue < minVal || zValue > maxVal)

{

if(buz == 0)

start=millis(); // timer start

buz=1; // buzzer / led flag activated

}

else if(buz == 1) // buzzer flag activated then alerting earthquake
```

```
{  
  lcd.setCursor(0,0);  
  lcd.print("Earthquake Alert ");  
  if(millis()>= start+buzTime)  
  buz=0;  
}  
else  
{  
  lcd.clear();  
  lcd.print(" X Y Z ");  
}  
digitalWrite(buzzer, buz); // buzzer on and off command  
digitalWrite(led, buz); // led on and off command  
/*sending values to processing for plot over the graph*/  
Serial.print("x=");  
Serial.println(xValue);  
Serial.print("y=");  
Serial.println(yValue);  
Serial.print("z=");  
Serial.println(zValue);  
Serial.println(" $");  
}
```

## ANNEXE B

### Processing IDE Code

```
import processing.serial.*;

PFont f6,f8,f12,f10;

PFont f24;

Serial myPort; // The serial port

int xPos = 0; // horizontal position of the graph

float y1=0;

float y2=0;

float y3=0;

void setup ()

{

// set the window size: and Font size

f6 = createFont("Arial",6,true);

f8 = createFont("Arial",8,true);

f10 = createFont("Arial",10,true);

f12 = createFont("Arial",12,true);

f24 = createFont("Arial",24,true);

size(1200, 700);

// List all the available serial ports

println(Serial.list());

myPort = new Serial(this, "COM3", 9600);
```

```
println(myPort);

myPort.bufferUntil('\n');

background(80);

}

void draw ()

{

serial ();

}

void serial()

{

String inString = myPort.readStringUntil('$'); // reading incoming data from serial

if (inString != null)

{

// extracting all required values of all three axis:

int l1=inString.indexOf("x")+2;

String temp1=inString.substring(l1,l1+3);

l1=inString.indexOf("y")+2;

String temp2=inString.substring(l1,l1+3);

l1=inString.indexOf("z")+2;

String temp3=inString.substring(l1,l1+3);

//mapping x, y and z value with graph dimensions

float inByte1 = float(temp1+(char)9);

inByte1 = map(inByte1, -80,80, 0, height-80);
```

```
float inByte2 = float(temp2+(char)9);

inByte2 = map(inByte2,-80,80, 0, height-80);

float inByte3 = float(temp3+(char)9);

inByte3 = map(inByte3,-80,80, 0, height-80);

float x=map(xPos,0,1120,40,width-40);

//ploting graph window, unit

strokeWeight(2);

stroke(175);

Line(0,0,0,100);

textFont(f24);

fill(0,00,255);

textAlign(RIGHT);

xmargin("EarthQuake Graph (SESMIOGRAPH)",200,100);

fill(100);

strokeWeight(100);

line(1050,80,1200,80);

strokeWeight(1);

textAlign(RIGHT);

fill(0,0,255);

String temp="X:"+temp1;

Text(temp,100,95);

fill(0,255,0);

temp="Y:"+temp2;
```



```
Text(temp,100,92);

fill(255,0,0);;

temp="Z:" +temp3;

Text(temp,100,89);

//ploting x y and z values over graph

strokeWeight(2);

int shift=40;

stroke(0,0,255);

if(y1 == 0)

y1=height-inByte1-shift;

line(x, y1, x+2, height-inByte1-shift) ;

y1=height-inByte1-shift;

stroke(0,255,0);

if(y2 == 0)

y2=height-inByte2-shift;

line(x, y2, x+2, height-inByte2-shift) ;

y2=height-inByte2-shift;

stroke(255,0,0);

if(y2 == 0)

y3=height-inByte3-shift;

line(x, y3, x+2, height-inByte3-shift) ;

y3=height-inByte3-shift;

xPos+=1;
```

```
if (x >= width-30) // go back to begining
{
  xPos = 0;
  background(80);
}
}
}

void Line(int x1, int y1, int x2, int y2)
{
  float xx1=map(x1,0,100,40,width-40);
  float xx2=map(x2,0,100,40,width-40);
  float yy1=map(y1,0,100,height-40,40);
  float yy2=map(y2,0,100,height-40,40);
  line(xx1,yy1,xx2,yy2);
  xx2=map(100,0,100,40,width-40);
  yy2=map(0,0,100,height-40,40);
  line(xx1,yy1,xx2,yy2);
  strokeWeight(1);
  for(int i=1;i<21;i++)
  {
    yy2=map(i*10,0,200,height-40,40);
    yy1=yy2;
    line(xx1,yy1,xx2,yy2);
```

```
}  
  
yy2=map(100,0,100,height-40,40);  
  
yy1=map(0,0,100,height-40,40);  
  
for(int i=1;i<41;i++)  
  
{  
  
xx1=map(i*5,0,200,40,width-40);  
  
xx2=map(i*5,0,200,40,width-40);  
  
line(xx1,yy1,xx2,yy2);  
  
}  
  
textAlign(RIGHT); // 100 degree  
  
// result+=yy1;  
  
fill(255);  
  
strokeWeight(1);  
  
textFont(f12);  
  
for(int i=-10;i<11;i++)  
  
{  
  
String result="";  
  
result+=5*i;  
  
ymargin(result, x1,y1);  
  
y1+=5;  
  
}  
  
x1=0;  
  
y1=0;
```

```
strokeWeight(1);

textFont(f10);

for(int i=0;i<41;i++)

{

String result="";

result+=28*3*i;

xmargin(result, x1,y1);

x1+=5;

}

textAlign(RIGHT);

textAlign(RIGHT);

}

void ymargin(String value, int x1, int y1)

{

float xx1=map(x1,0,100,40,width-40);

float yy1=map(y1,0,100,height-40,40);

text(value,xx1-5,yy1+5);

}

void xmargin(String value, int x1, int y1)

{

float xx1=map(x1,0,200,40,width-40);

float yy1=map(y1,0,100,height-25,25);

text(value,xx1+7,yy1);
```

```
}  
  
void Text(String value, int x1, int y1)  
  
{  
  
float xx1=map(x1,0,100,40,width-40);  
  
float yy1=map(y1,0,100,height-25,25);  
  
text(value,xx1,yy1);  
  
}
```

## **Résumé:**

Un mouvement inattendu de la surface de la Terre s'appelle un tremblement de terre. Lorsque deux parties de la surface de la Terre se déplacent soudainement l'une par rapport à l'autre le long de la ligne de faille, en raison des forces tectoniques, un tremblement de terre se produit. Un tremblement de terre (également connu sous le nom de tremblement ou tremblement) est le résultat de la libération soudaine d'énergie dans la croûte terrestre entraînant des ondes sismiques. Les tremblements de terre sont enregistrés à l'aide d'un sismomètre, également appelé sismographe. La magnitude d'un séisme conventionnellement signalée, ou la magnitude associée et souvent obsolète de Richter, est celle où les séismes de magnitude 3 ou moins sont souvent imperceptibles et de magnitude 7 causant des dommages importants sur de vastes zones. L'intensité des vibrations est mesurée sur une échelle de Marcella modifiée. Un tremblement de terre est une catastrophe naturelle imprévisible qui cause des dommages à la vie et aux biens. Cela arrive soudainement et nous ne pouvons pas l'arrêter mais nous pouvons en être alertés. De nos jours, il existe plusieurs techniques qui peuvent être utilisées pour détecter les petites vibrations et les percussions, nous pouvons donc prendre des précautions avant que certaines vibrations majeures du sol ne se produisent. Ce travail utilise un accéléromètre pour détecter les vibrations pré-sismiques. L'accéléromètre est très sensible aux vibrations et aux vibrations sur les trois axes. L'avantage de construire un détecteur de tremblement de terre basé sur Arduino avec un accéléromètre est de réduire les pertes destructrices.

**Mots-clés :** Tremblement de terre, Vibration, Catastrophe, Arduino, Accéléromètre, Détection.

## **Abstract:**

An unexpected movement of the earth's surface is called an earthquake. When two parts of the earth surface move suddenly in relation to each other along a fault line, due to tectonic forces, an earthquake occurs. An earthquake (also known as a tremor or temblor) is the result of a sudden release of energy in the earth's crust that creates seismic waves. Earthquakes are recorded with a seismometer, also known as a seismograph. The moment magnitude of an earthquake is conventionally reported, or the related and mostly obsolete Richter magnitude, with magnitude 3 or lower earthquakes being mostly imperceptible and magnitude 7 causing serious damage over large areas. Intensity of shaking is measured on the

modified Marcella scale. An earthquake is an unpredictable natural disaster that causes damage to lives and property. It happens suddenly and we cannot stop it but we can be alerted from it. Now-a-days, there are many technologies which can be used to detect the small shakes and knocks, so that we can take precautions prior to some major vibrations in earth. This work uses an accelerometer to detect the pre-earthquake vibrations. Accelerometer is highly sensitive to shakes and vibrations along with all the three axes. The advantage of building an Arduino based Earthquake Detector using Accelerometer is to reduce its destructivelosses.

**Keywords:** Earthquake, Vibration, Disaster, Arduino, Accelerometer, Detection.

### الملخص:

تسمى الحركة غير المتوقعة لسطح الأرض بالزلازل. عندما يتحرك جزءان من سطح الأرض فجأة فيما يتعلق ببعضهما البعض على طول خط الصدع ، بسبب القوى التكتونية ، يحدث زلزال. الزلزال (المعروف أيضًا باسم الهزة أو الرعدة) هو نتيجة الإطلاق المفاجئ للطاقة في القشرة الأرضية مما يؤدي إلى حدوث موجات زلزالية. يتم تسجيل الزلازل باستخدام مقياس الزلازل ، المعروف أيضًا باسم جهاز قياس الزلازل. يتم الإبلاغ عن حجم الزلزال بشكل تقليدي ، أو حجم ريختر المرتبط به والذي عفا عليه الزمن في الغالب ، حيث تكون الزلازل التي تبلغ قوتها 3 أو أقل من الزلازل غير محسوسة في الغالب وتسبب قوتها 7 أضرارًا جسيمة على مناطق واسعة. يتم قياس شدة الاهتزاز على مقياس ماركسيلا المعدل. الزلزال هو كارثة طبيعية لا يمكن التنبؤ بها تسبب أضرارًا في الأرواح والممتلكات. يحدث فجأة ولا يمكننا إيقافه ولكن يمكن تنبيهنا منه. في الوقت الحاضر ، هناك العديد من التقنيات التي يمكن استخدامها للكشف عن الاهتزازات الصغيرة والقرع ، حتى نتمكن من اتخاذ الاحتياطات قبل حدوث بعض الاهتزازات الرئيسية في الأرض. يستخدم هذا العمل مقياس التسارع لاكتشاف اهتزازات ما قبل الزلزال. مقياس التسارع حساس للغاية للاهتزازات والاهتزازات مع جميع المحاور الثلاثة. تتمثل ميزة بناء كاشف للزلازل على أساس اردوينو باستخدام مقياس التسارع في تقليل الخسائر المدمرة.

**الكلمات الرئيسية:** زلزال، اهتزاز، كارثة، أردوينو ، مقياس تسارع ، كشف.