

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université MOULOU D MAMMERI Tizi-Ouzou

Faculté de génie électrique et d'informatique

Département d'électronique

Mémoire de fin d'études pour l'obtention du diplôme de
Master en Electronique option industrielle



Thème

Conception et réalisation d'une
station météorologique mobile

➤ *Réalisé par :*

KECILI NEDJMA

SAIDANI SABRINA

Encadré par :

Mr IDJERI BOUSSAD

Promotion : 2017-2018

Remerciements

*En premier lieu, nous tenons à remercier « **DIEU** » le miséricordieux le tout-puissant, et qui nous a m'aidé pour que ce modeste travail soit achevé et pour que nous réussissons.*

Nous tenons à remercier vivement tous ceux qui nous orientées et nous encouragées.

*Nous pensons en particulier de notre encadreur: **IDJERI BOUSSAAD**, d'avoir nous bien suivi et dirigé notre travail et de faire profités de son savoir, ainsi de ses conseils, et pour toute l'aide, les remarque qui nous avons d'améliorer ce travail, et qui grâce a lui nous pouvons réaliser notre Objectif, et aussi nous remercier le trinôme étudiant en L3 automatique (**HAMOUM MOUKRANE, HAMADANE MOULOUD, HACHEMI MEZIANE**) pour leurs aides.*

Nous adressons nos sincères remerciements, pour membres du jury pour l'honneur Qu'ils nous ont fait en participant à l'évaluation de ce travail :

- Monsieur : **ATTAF YUCEF***
- Monsieur : **HAMMEG SLIMANE***

*Nous grands remerciment aussi tous les enseignants
qui ont contribué à notre formation, et a tout les étudiants
d'électronique industrielle.*

Dédicaces

C'est avec profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d'étude à la mémoire de mon père, ma chère maman et mon grand père, qui ont sacrifié leur vie pour ma réussite et m'ont éclairé le chemin par leurs conseils judicieux.

J'espère qu'un jour, Je pourrai leur rendre un peu de ce qu'ils ont fait pour moi, que dieu leur prête bonheur et longue vie.

Je dédie aussi ce travail à mon frère Yacine, et mes sœurs Cyria, Samira et ma tante Zohra, ma grand mère et mes oncles.

J'espère que ce travail soit un exemple pour vous de persévérance, de courage et de générosité.

Je tiens de dédier ce travail à toute ma famille ainsi mes ami(e)s, Je dédie ce travail à ma chère binôme Saïdani Sabrina.

« Le succès n'est pas la clé du bonheur. Le bonheur est la clé de succès. Si vous aimez ce que vous faites vous réussirez. »

Albert Schweitzer

Dédicaces

Du plus profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers.

- *A MES CHÈRES PARENTS,*

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.

Je vous remercie pour tous les soutiens et l'amour que vous me portez et j'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tant formolés, le fruit de vos innombrables sacrifices. Puisse Dieu, le très haut, vous accorder santé, bonheur et longue vie.

- *A MES CHÈRES SOEURS,*

Ghenima, Lamia, Drifa, Zohra, et pour les encouragements, et leur soutien.

- *A MES CHÈRES FRÈRES,*

Cherif, Essaid et Imed.

Je tiens de dédier ce travail à tous les membres de ma famille mes neveux et mes nièces ainsi mon grand père, ma grand-mère et ma tante Fatima.

*Je dédie ce travail à ma chère binôme MlleKeciliNedjmaqui
a trouvé la patience de me supporterdurant ce mémoire, et
m'encouragé.*

Sommaire

Sommaire

Introduction générale	1
-----------------------------	---

Chapitre I: Les variables météorologiques

I)- Préambule	3
II)- Généralités sur la météorologie	3
III)- Les variables météorologiques mesurées.....	3
III-1)- Température	4
III-2)- L'humidité.....	4
III-3)- L'ensoleillement :	5
III-4)- Le vent	5
III-5)- La précipitation	6
III-6)- Luminosité	6
III-7)- La pression atmosphérique	6
IV)- Les instruments de mesure météorologique.....	7
IV-1)- Capteurs de température.....	7
IV-2)- Capteur d'humidité	8
IV-3)- Capteur du vent	9
IV-4)- Pluviomètre	10
IV-5)- Capteur de rayonnement solaire.....	10
V)- Les normes d'emplacement des différents capteurs	11
V-1)- Le capteur de température et d'humidité.....	12
V-2)- Le capteur du vent	12
V-3)- Le pluviomètre	13
V-4)- Le pyranomètre	13
VI)- Conclusion :.....	13

Chapitre II : Présentation des différents instruments utilisés

I)- Préambule	15
II)- Module Arduino	15

Sommaire

II.1)- Partie matérielle	15
II.2)- Partie logicielle	19
III)-Présentation des shields	21
IV)- Capteurs utilisés	22
IV.1)- Capteur de température et d'humidité de l'air DHT11	22
IV.2)- Capteur de luminosité TSL2561	24
IV.3)- Capteur de vibration SW-18010P	26
IV.4)- Capteur d'humidité du sol YL69	27
IV.5)- Capteur de pluie	29
IV.6)- Capteur de pression (BMP280)	31
V)- Le module Bluetooth HC05	32
V.1)- Interface Smartphone et Arduino	33
VI)- Conclusion	34

Chapitre III : Conception de la station de mesure

I)- Préambule	36
II)- Architecture du système de mesure	36
III)- Conception de la station de mesure	37
III.1)- Réalisation des mesures	38
III-1-1)- Programmation de microcontrôleur	39
III.2)- Interfaçage Arduino/ Smartphone	41
III-2-1)- Présentation du système Android	41
III-2-2)- Présentation de l'application Inventor MIT	41
III-2-3)- Bluetooth	42
III-3) -Transmission Arduino/ Smartphone.....	44
III-3-1)- Programmation application Android	44
IV)- Installation de la carte à l'extérieur	47
V)-Conclusion	49
Conclusion générale	51

Sommaire

ANNEXES

ANNEXE A.....	53
ANNEXE B.....	68
ANNEXE C.....	69
Bibliographie	
Liste des figures	
Liste des abréviations	

INTRODUCTION GENERALE

Introduction générale

La météorologie est une science qui traite principalement des lois physiques régissant la dynamique du fluide atmosphérique (air sec + vapeur d'eau), considéré comme un gaz parfait pour expliquer le comportement de l'atmosphère et prévoir son comportement futur. Elle permet donc d'établir des prévisions météorologiques en s'appuyant sur des modèles mathématiques à court et à long terme.

Elle est également appliquée pour la prévision de la qualité de l'air, pour les changements climatiques et pour l'étude de plusieurs domaines de l'activité humaine (construction, trafic aérien, secteur maritime, etc). Afin de prévoir le temps, les météorologues ont besoin de rassembler des informations concernant l'atmosphère dans le monde entier.

Dans cette optique, il est donc important de développer des systèmes permettant l'estimation des différents paramètres météorologiques. A cet effet, nous proposons dans ce mémoire la conception d'une station météorologique mobile constitué de deux parties distinctes émission et réception. La partie émission est composée d'un ensemble de capteurs permettant de recueillir les différentes données sur les paramètres à mesurer et d'une carte à microcontrôleur Arduino munie d'un module de transmission de type Bluetooth. Quant à la partie réception elle est constituée d'un téléphone portable munie à son tour d'un module de réception.

Le mémoire résumant le travail réalisé est organisé autour de trois chapitres :

- Dans le premier chapitre, nous présentons des généralités sur les variables météorologiques et les instruments utilisés pour les mesures .
- Le deuxième chapitre est consacré à la présentation du système Arduino UNO, et les modules utilisés pour le développement de notre station météo.
- Le troisième chapitre est réservé à la conception et la réalisation de la station météo.
- La conclusion résume le travail réalisé et donne un aperçu sur quelques perspectives qui peuvent être développées.

CHAPITRE I :

Les variables météorologiques

I)-Préambule :

L'observation météorologique a une très longue histoire. Dès le début de l'agriculture végétale, mais toujours été difficile à prévoir sans instruments conçus pour mesurer des conditions spécifiques. Dans ce chapitre, nous allons donner tous les détails sur la météo et l'innovation dans ce domaine.

II)- Généralités sur la météorologie :

La météorologie est l'étude des phénomènes atmosphériques tels que les nuages, les pressions et les précipitations pour comprendre comment ils se forment et évoluent. Elle est fondée sur l'observation quotidienne du temps et sur la mesure de divers paramètres.

La station météorologique personnelle a la possibilité d'afficher un tableau étonnamment large de conditions météorologiques, que ce soit la température, l'humidité interne et externe, la précipitation, le vent, le refroidissement éolien, la pression atmosphérique, le point de rosée, l'indice de chaleur, ou même solaire et un rayonnement ultraviolet. Et toutes ces informations peuvent être recueillies sans même quitter la maison. Ce type de technologie de pointe peut sembler comme il serait très complexe et trop sophistiqué pour l'homme commun à utiliser, mais en réalité qui est loin de la vérité. Bien entendu, les experts météorologiques vont tirer la meilleure partie des machines, mais l'équipement est plus qu'assez facile pour les amateurs d'utiliser aussi bien. Les stations météorologiques sont soit par câble ou sans fil. Dans la plupart des cas, les câblés utilisent un câble pour fixer la console d'affichage à l'intérieur d'un boîtier de capteur intégré. Ceux-ci sont généralement moins chers que leurs homologues sans fil.

III)- Les variables météorologiques mesurées :

Les paramètres météorologiques les plus utiles pour définir le temps qu'il fait sont : [1]

III-1)- La température :

La température est considérée comme une grandeur physique liée à la notion immédiate de chaud et froid, est la manifestation à l'échelle macroscopique du mouvement des atomes et molécules, ainsi une température élevée signifie une grande « agitation » atomique.

L'unité internationale de température est le kelvin (K), le degré Celsius (°C) est une autre unité très répandue en Europe. Certains pays anglo-saxons et les Etats-Unis utilisent une autre unité: Le degré Fahrenheit (°F), la plus basse température du système Celsius est -273,15 °C correspondant à 0 K.

Les formules de transformations d'unités sont les suivantes :

$$^{\circ}\text{C} = 0,55 \times (^{\circ}\text{F} - 32), \quad \text{K} = ^{\circ}\text{C} + 273,15, \quad ^{\circ}\text{F} = 32 + (1,8 \times ^{\circ}\text{C}) \quad (\text{I.1})$$

Généralement, la température est représentée sous forme de courbe.

III-2)- L'humidité :

L'humidité représente la quantité de vapeur d'eau présente dans l'air, sans compter l'eau liquide et la glace. On doit distinguer l'humidité relative et l'humidité absolue. L'humidité relative joue un rôle sur la formation du brouillard, de la rosée et des nuages. En général, quand on parle de mesure d'humidité, on fait allusion à l'humidité relative exprimée en %, l'humidité relative de l'air correspond au rapport de la pression partielle de vapeur d'eau contenue dans l'air sur la pression de vapeur saturante à la même température et pression. Ce rapport changera si on change la température ou la pression, l'humidité relative est souvent appelée degré hygrométrique, elle suit une échelle allant de 0 à 100 %. Un air saturé en vapeur d'eau a une humidité relative de 100 %, un air très sec, une humidité de 10 à 20 %. La pression de vapeur saturante et l'humidité relative dépendent de la température, plus la température de l'air est élevée, plus il peut contenir de vapeur d'eau. On définit l'humidité absolue comme le rapport de la masse de vapeur d'eau, généralement en (g) sur le volume d'air humide en (m^3) à la pression et la température considérées. On peut aussi la définir comme le produit de l'humidité relative par l'humidité absolue de saturation.

III-3)- L'ensoleillement :

Aussi appelé insolation, est la mesure du rayonnement solaire que reçoit une surface au cours d'une période donnée, s'exprimant en mégajoules par mètre carré, MJ/m² (comme recommandé par l'organisation météorologique mondiale) ou en watts-heures par mètre carré, Wh/m².

La durée d'ensoleillement ou durée de l'insolation effective est un indicateur climatique qui mesure la durée temporelle pendant laquelle un endroit, un lieu est soumis à l'insolation effective, c'est-à-dire soumis à un rayonnement solaire suffisamment intense et puissant pour produire des ombres bien distinctes. Cet indicateur est directement lié à la nébulosité, la part du ciel couvert par les nuages, du lieu ou de l'endroit en question. Il ne doit pas être confondu avec l'ensoleillement qui exprime la quantité d'énergie reçue par le rayonnement solaire sur une surface donnée et sur une certaine période. La durée d'ensoleillement est généralement exprimée en une moyenne d'heures par an ou en une moyenne d'heures par jour.

III-4)- Le vent :

En météorologie, le vent désigne le mouvement horizontal de l'air, sa mesure comprend deux paramètres : sa direction et sa vitesse, exprimée communément en km/h ou m/s. La mesure du vent est toujours une moyenne sur une période donnée. L'anémomètre permet de mesurer la vitesse du vent. Celui-ci traduit le déplacement de l'air d'une zone de haute pression vers les basses pressions. Le vent a pour origine la différence de températures observée en surface, provoquée par le rayonnement solaire. Dans la couche limite atmosphérique, le vent est un écoulement turbulent d'air, constitué d'une multitude de tourbillons de tailles diverses, imbriqués les un dans les autres. La vitesse de vent en un point donnée de l'espace, présente la forte variation plus ou moins irrégulière d'amplitude et de fréquence différentes. La structure du vent est proche de celle de tous les écoulements fortement cisailés. Sa vitesse moyenne varie avec la hauteur, la rugosité du sol, la topographie du site. En pratique, elle se mesure sur des périodes de temps qui varient entre dix minute et une heure (pour que le résultat ne soit sensible ni aux fluctuations lentes ni aux variations rapides du vent).

III-5)- La précipitation :

Les précipitations désignent les gouttes d'eau ou les cristaux de glace, formés après condensation et agglomération dans les nuages, deviennent trop lourds pour se maintenir en suspension dans l'air et tombent au sol ou s'évaporent avant de l'atteindre. Ces précipitations sont de plusieurs natures : la pluie, la neige et la grêle comptent parmi les plus fréquentes.

- **La pluie** : se forme depuis des gouttelettes ou des cristaux de glace qui dans leur chute, ne sont pas soumis à des températures inférieures au seuil de congélation.
- **la grêle** : correspond à des billes de glaces pouvant tomber jusqu'à la vitesse de 160 km/h, avec des tailles parfois impressionnantes, certaines fois proches de celle d'une balle de tennis.
- **Le grésil** : contrairement à la grêle, passe à l'état liquide avant de rencontrer une couche plus froide et inférieure à 0 °C au cours de sa chute, poussant l'enveloppe à geler.
- **La neige** : se forme dans des conditions particulières, lorsque la vapeur d'eau se transforme directement en cristaux de glace qui s'agglomèrent de telle façon qu'ils forment des flocons, tout en traversant des couches dont la température leur permet de ne pas fondre au cours de leur chute.

III-6)- La luminosité :

La lumière est une partie extrêmement étroite du rayonnement électromagnétique dans lequel nous baignons, elle se déplace à une vitesse 'c' de 300 000 km/s. Ce rayonnement est caractérisé par sa longueur d'onde λ en mètres (ou sa fréquence f en hertz ($\lambda = c/f$)). Ce que l'on appelle la "lumière" est la partie de ce rayonnement que perçoit l'œil humain, elle est comprise entre 0,38 μm et 0,78 μm (380 nm et 780 nm). En fait, notre œil ne perçoit pas la lumière directement, il ne perçoit que celle qui est émise ou réfléchiée par des objets (entre ce document et votre œil, il y a de la lumière que vous ne pouvez voir).

III-7)- La pression atmosphérique :

La pression atmosphérique correspond à la pression générée par une colonne d'air en un point donné, elle s'exprime en pascal (Pa), unité équivalente au newton par mètre carré (N/m^2). En

moyenne, les météorologues parlent de basses pressions, synonymes de mauvais temps dans les régions tempérées. Au-dessus de 1.020 HPA, correspond aux hautes pressions. L'équation (I.2) représente l'équivalence entre les unités bar et pascal.

$$1 \text{ bar} = 1 \times 10^5 \text{ Pa} \quad (\text{I.2})$$

IV)- Les instruments de mesure météorologique:

Parmi les capteurs météorologiques, les capteurs de type Cimel sont les plus utilisés. [2]

IV-1)- Le capteur de température:

C'est un capteur de haute précision ($\pm 0,05^\circ\text{C}$) à résistance platine (Pt100) pour la mesure de la température. Il existe un modèle pour chaque type de mesure de température: sous abri, dans le sol, Il est conforme aux recommandations de l'OMM (Organisation Mondiale de la Météorologie) en termes de qualité et de disposition.

Ce capteur permet des mesures précises, rapides et aisées des différents types de température météorologiques et agro météorologiques

- Température de l'air sous abri
- Température dans le sol à différentes profondeurs en général entre (5 et 100 cm)

La figure I.1 représente le capteur de température Cimel.



Figure I.1 : Capteur de température Cimel.

IV-2)- Le capteur d'humidité:

C'est un capteur numérique de grande précision à cellule capacitive destiné à la mesure de l'humidité relative de l'air température de rosée dans un abri thermique. Le capteur CES191 est un capteur intelligent, Plug-and-Play, économique sans aucune maintenance. Ce capteur est conforme aux recommandations de l'OMM en termes de qualité et de disposition.

Le capteur est équipé d'un processeur qui assure la mesure, le traitement du signal et la communication avec l'unité d'acquisition. Il est conçu selon la technologie Cimel de micro-consommation (MicroAmps) pour être monté spécifiquement dans les stations météorologiques automatiques Cimel. Il jouit d'une grande stabilité sur une longue période : de 2 à 5 ans. Au-delà, il est conseillé de le remplacer sans réétalonnage.

La figure I.2 représente le capteur d'humidité Cimel :

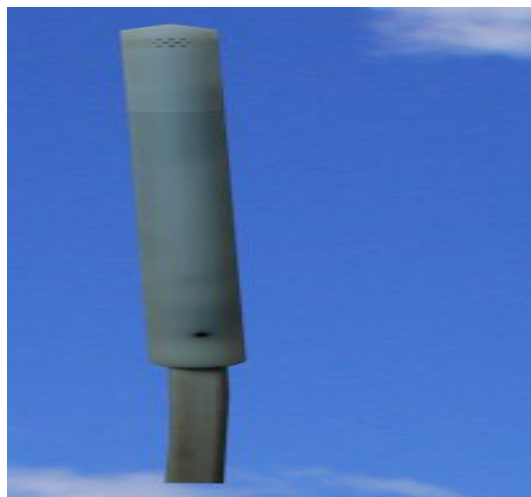


Figure I.2: Capteur d'humidité Cimel CES 191.

➤ **Principe de mesure :**

- L'élément sensible est un polymère hygroscopique dont la capacité électrique varie en fonction de l'humidité relative
- Les corrections de linéarité et d'effet de température sont effectuées par le logiciel embarqué du capteur
- Le support de montage dans le mini-abri assure le positionnement du capteur d'humidité au même niveau que le capteur de température afin d'améliorer la précision de la mesure de la température de rosée
- Conçu pour être remplacé sans étalonnage

IV-3)- Le capteur du vent:

Le capteur du vent Cimel est composé en deux partie : utiliser pour mesure de vitesse à l'aide d'un anémomètre et direction du vent à l'aide d'une girouette, l'unité de mesure c'est m/s ou km/h. Les figures I.3 représente le capteur du vent cimel.



(a). Anémomètre CES155

(b).Girouette CES157

Figure I.3 : Capteur du vent Cimel.

✓ **Anémomètre à codage optique CES 155 :**

Ce capteur est destiné à mesurer la vitesse du vent avec une grande précision dans des conditions climatiques rudes. Il répond aux recommandations de l'OMM en termes de qualité et de disposition.

✓ **Girouette à codage optique CES 157:**

Ce capteur est destiné à mesurer la direction du vent avec une grande précision et dans des conditions climatiques rudes. La technologie à codage optique assure une grande durabilité au système. Il est conforme aux recommandations de l'OMM en termes de qualité et de disposition.

IV-4)- Le pluviomètre :

C'est un capteur automatique à augets basculants pour la mesure de la quantité de précipitation et de l'intensité pendant un temps donné. Il se présente sous la forme d'une boîte cylindrique. L'eau le pénétrant, fait basculer l'auget d'un côté puis s'évacue sous son poids. L'eau s'accumule sur l'autre partie de l'auget qu'il le fait basculer de l'autre côté et ainsi de suite.

Selon les constructeurs, la surface du réceptacle du pluviomètre est différente, une impulsion de comptage correspond à une quantité d'eau propre à chacun. La figure I.4 représente le pluviomètre.



Figure I.4 : Le pluviomètre.

IV-5)- Le capteur de rayonnement solaire :

L'instrument utilisé pour mesurer la quantité du rayonnement est un Pyranomètre. Son fonctionnement est basé sur le principe de détection thermoélectrique, utilisé de sorte que le rayonnement est presque complètement absorbé par une surface noire horizontale, sur une plage de longueurs d'onde très large. L'augmentation de température est mesurée par l'intermédiaire de thermocouples connectés en série ou en série-parallèle pour effectuer une thermopile.

Les jonctions actives (à chaud) sont situées sous la surface du récepteur noir et sont chauffées par le rayonnement absorbé dans le revêtement noir, les jonctions passives (à froid) de la thermopile sont en contact thermique avec le boîtier du pyranomètre, qui sert de dissipateur de chaleur.

Les pyranomètre passifs à thermopile ne nécessitent pas l'alimentation électrique. Le détecteur génère une petite tension proportionnelle à la différence de température entre la surface absorbante noir et le boîtier de l'appareil, c'est de l'ordre de 10 mV (microvolts) par W / m^2 , donc sur une journée ensoleillée, la sortie sera autour de 10 mV (millivolts). Chaque pyranomètre a une sensibilité particulière, définie au cours du processus d'étalonnage, qui est utilisé pour convertir le signal de sortie du microvolt en éclairement global W / m^2 . La figure I.5 représente le capteur de rayonnement



Figure I.5: Capteur de rayonnement.

V)- Les normes d'emplacement des différents capteurs :

Afin d'installer les différents capteurs il faut respecter les normes d'emplacement correspondantes à chaque capteur. Donc il faut tous les regrouper pour choisir un site où on peut implanter une station. [3]

V-1)- Le capteur de température et d'humidité :

Les capteurs à l'intérieur de l'abri doivent être installés à une hauteur standard de 1,5 m. Le lieu doit être parfaitement ventilé c'est à dire qu'aucun obstacle ne doit bloquer l'écoulement d'air.

En effet, un lieu confiné va avoir des températures soit trop élevée soit trop basse. Le lieu doit être suffisamment ensoleillé, car si le terrain présente une faible part d'ensoleillement, les températures vont être sous-estimées. par conséquent on doit éviter les lieux à proximité des bassins, les piscines, les murs dépassant 1m20, les arbres pouvant venir ombrager l'abri

V-2)- Le capteur du vent :

L'anémomètre et la girouette sont des dispositifs qui doivent être installés principalement sur un toit d'habitation. Ils permettent de connaître la direction et la vitesse du vent selon les règles usuelles, les capteurs sont généralement placés à 10 m au-dessus du sol et sur un terrain dégagé, c'est-à-dire une surface où les obstacles se situent à une distance d'au moins dix fois leur hauteur.

Un emplacement libre de tout obstacle environnant afin de minimiser les erreurs, habituellement, sur le toit d'un bâtiment est l'idéal une fois l'emplacement déterminé, la pose de la girouette et l'anémomètre doit respecter certains critères d'emplacement afin de ne pas fausser les lectures des vents et de leurs directions.

Premièrement il faut vérifier et s'assurer que les mâts auxquels la girouette et l'anémomètre sont fixés, sont au niveau, par la suite il faut calibrer la girouette en prenant comme point de départ le nord avec une boussole, lorsque tout est en ordre, on peut appuyer sur le reset pour la mettre en opération OMM préconise une distance entre le pylône anémométrique et l'obstacle de 10 fois la hauteur de l'obstacle, de plus, les capteurs doivent demeurer accessibles pour étalonnage (girouette) ou entretien.

V-3)- Le pluviomètre :

Lors de l'installation du pluviomètre, on doit respecter des règles élémentaires afin de recueillir dans l'appareil une quantité de précipitations égale à celle tombée sur une surface horizontale. Pour cela le pluviomètre doit être disposé sur une surface parfaitement horizontale.

V-4)- Le pyranomètre :

Le choix du lieu d'installation de ce capteur n'est pas compliqué, il doit être exposé au soleil, tout en évitant les ombres portées, notamment au lever et au coucher du soleil pendant plus de 30 % de la journée, quel que soit le jour de l'année.

VI)- Conclusion :

La mesure des paramètres météorologique est le point de départ de toute prévision météorologique, qui permet les descriptions qualitatives du ciel, et mieux comprendre et prévoir les phénomènes météorologiques. Toutes les observations doivent être méticuleusement définies, normalisées, sélectionnées et organisées. Les stations météorologiques doivent faire appel à des instruments fiables de hautes qualités pour avoir des mesures justes. Le chapitre suivant porte sur quelques instruments utilisés pour la réalisation de mesures.

CHAPITRE II :

Arduino et différents capteurs utilisés

I)- Préambule :

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée, on parle aussi du système embarqué ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit. Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui.

L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation, on va parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance d'électronique.

II)- Module Arduino :

II-1)- Partie matérielle :

Arduino, et son récent synonyme Genuino, est une marque qui couvre des cartes matériellement libres sur lesquelles se trouve un microcontrôleur (D'architecture Atmel comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due). Les schémas de ces cartes sont publiés en licence libre. Cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas sous licence libre.

Arduino est une plate - forme électronique open-source basé sur le matériel et le logiciel facile à utiliser. Les cartes Arduino sont capables de lire les entrées - La lumière sur un capteur, un doigt sur un bouton ou un message Twitter - et la transformer en une sortie - activation d'un moteur, d'allumer une LED, afficher une écriture. On peut commander la carte Arduino en envoyant un ensemble d'instructions au microcontrôleur disposé sur la carte. Pour ce fait, on utilise le langage de programmation Arduino (Basé sur le câblage) et le logiciel Arduino IDE (Integrated Development Environment) basé sur le traitement.

Au fil des années, Arduino a été le cerveau de milliers de projets, des objets de tous les jours à des instruments scientifiques complexes. Une communauté mondiale des décideurs, étudiants,

amateurs, artistes, programmeurs et professionnels - sont rassemblés autour de cette plate - forme open source, leurs contributions ont ajouté une quantité incroyable de connaissances accessibles qui peuvent être d'une grande aide.

Toutes les cartes Arduino sont complètement open source, permettant aux utilisateurs de les construire de façon autonome et les adapter à leurs besoins particuliers.

Le logiciel IDE Arduino est aussi en open-source, et il se développe à travers les contributions des utilisateurs dans le monde entier. [4]

La figure II.1 présente une carte Arduino UNO.

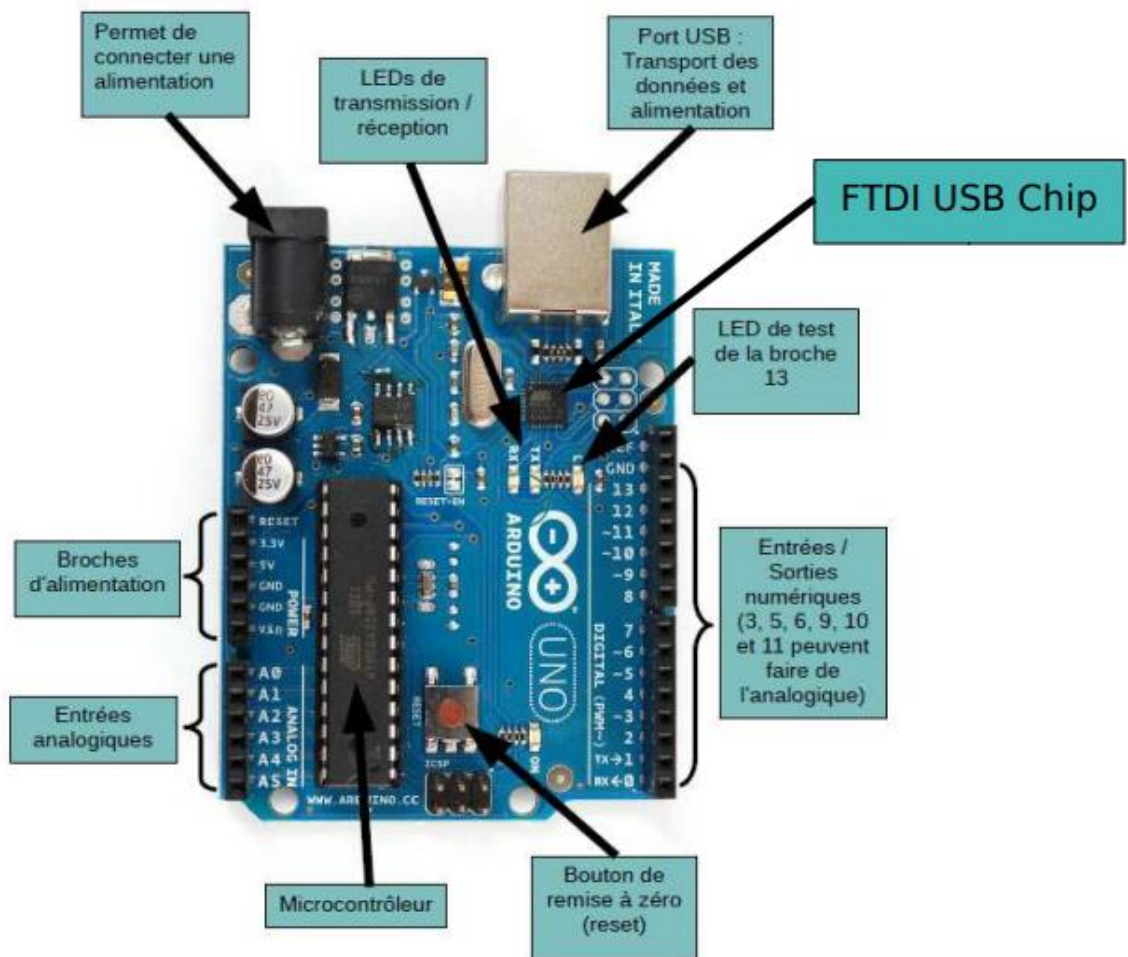


Figure II.1: Vue de la carte Arduino UNO.

➤ **Le Microcontrôleur ATmega328 :**

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit.

Aujourd'hui, en soudant un grand nombre de composants encombrants; tels que les transistors; les résistances et les condensateurs 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.

La figure II.2 montre deux types de microcontrôleur ATmega328, disponibles sur les cartes Arduino.

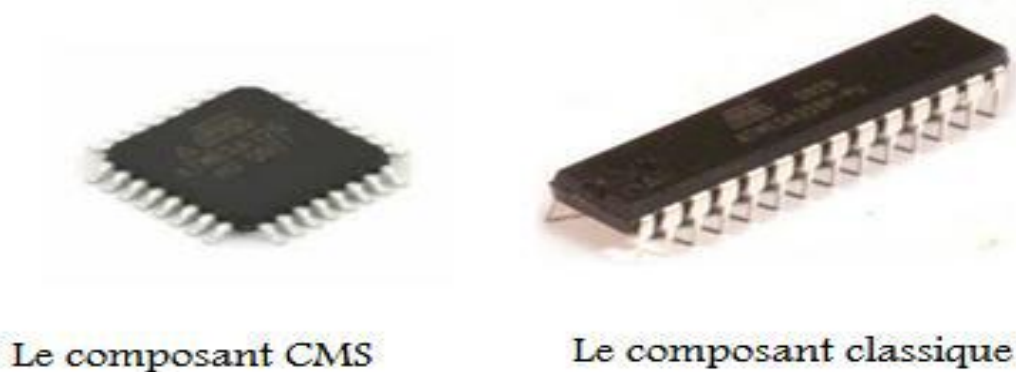


Figure II.2 : Microcontrôleur ATmega328.

Le microcontrôleur ATmega328 est constitué d'un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable est composée essentiellement de :

- ✓ **Mémoire Flash :** C'est celle qui contiendra le programme à exécuter. Cette mémoire effaçable et réinscriptible est une mémoire programmée de 32Ko.

- ✓ **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.

✓ **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

➤ **Les sources d'alimentation de la carte :**

La carte Arduino UNO, 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.

La carte peut fonctionner avec une alimentation externe de (6 à 20 volts). 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 on utilise plus de (12V), le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino UNO est entre (7V et 12V).

Les broches d'alimentation sont les suivantes :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée), On peut alimenter la carte à l'aide de cette broche.

- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino).

- **3V3** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible, l'intensité maximale sur cette broche est de 50mA.

- **GND** : Broche de masse (Ou 0V).

➤ **Les entrées ou sorties numériques:**

Chacun des connecteurs D0 a D13 peut être configuré par programmation en entrée ou en sortie, nous pouvons donc avoir par exemple les connecteurs 2 et 3 configurés comme des entrées et les connecteurs 7, 8 et 9 configurés comme des sorties. Il est par conséquent possible de connecter côte à côte des capteurs logiques (Interrupteurs par exemple) aux connecteurs 2 et 3 et des actionneurs aux connecteurs 7, 8 et 9. Les signaux véhicules par ces connecteurs sont des signaux logiques, c'est-à-dire qu'ils ne peuvent prendre que deux états : HAUT (5 Volts) ou BAS (0 Volt), par rapport au connecteur de masse GND, qui lui est toujours, par définition, à 0 Volt.

➤ **Les entrées analogiques :**

Contrairement aux entrées/sorties numériques qui ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre un millier de valeurs (1024 exactement) analogiques comprises entre 0 et 5 Volts. Nous pourrions donc avoir des valeurs de tension précises à 5 mV près ($\approx 5V/1024$)

II-2)- Partie logicielle :

L'interface IDE Arduino est plutôt simple, elle offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Elle est dotée d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console pour afficher les résultats de la compilation du code source, des opérations sur la carte, etc.

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le **C**, le **C++**, le **Java** et le **Processing**. Le langage impose une structure particulière typique de l'informatique embarquée.

- La fonction « **Setup** » contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).

- La fonction « **Loop** », elle est exécutée en boucle après l'exécution de la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée par le bouton reset. Cette boucle est absolument nécessaire sur les microcontrôleurs, étant donné qu'ils n'ont pas du système d'exploitation. En effet, si cette boucle est omise, à la fin du code produit. La figure II.3 représente l'interface IDE Arduino.

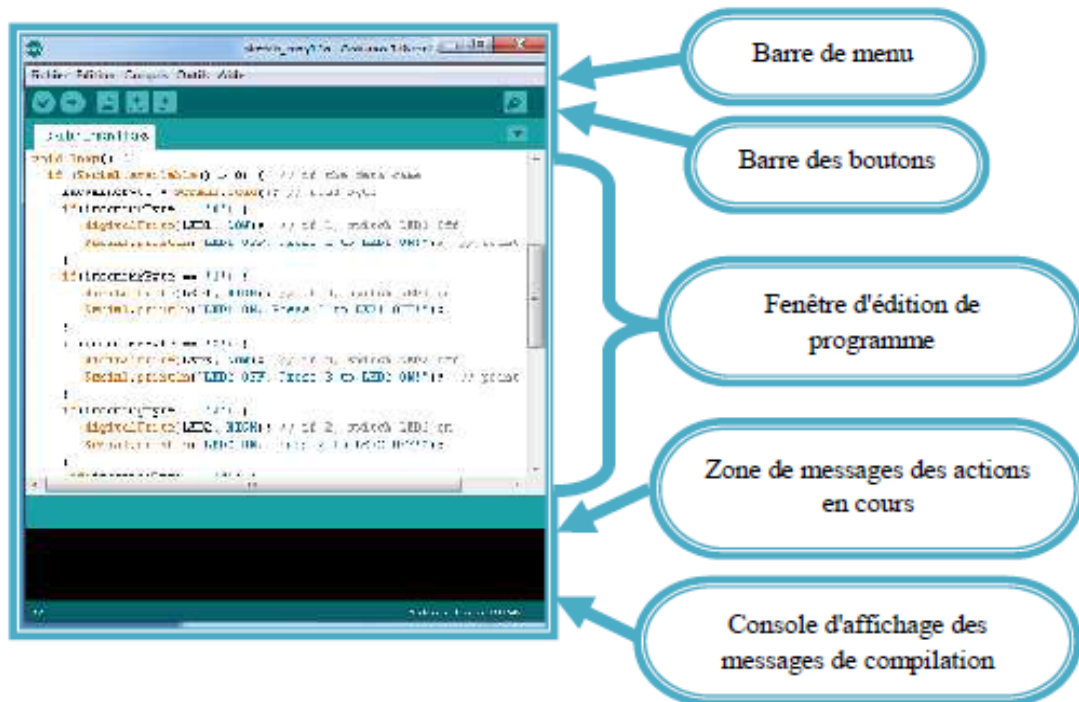


Figure II.3: Interface IDE Arduino.

➤ Structure générale d'un programme sur l'IDE :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis exécute les instructions les unes après les autres, dans un ordre défini par les lignes du code.

La structure d'écriture d'un programme sous Arduino est représentée par la figure II.4.

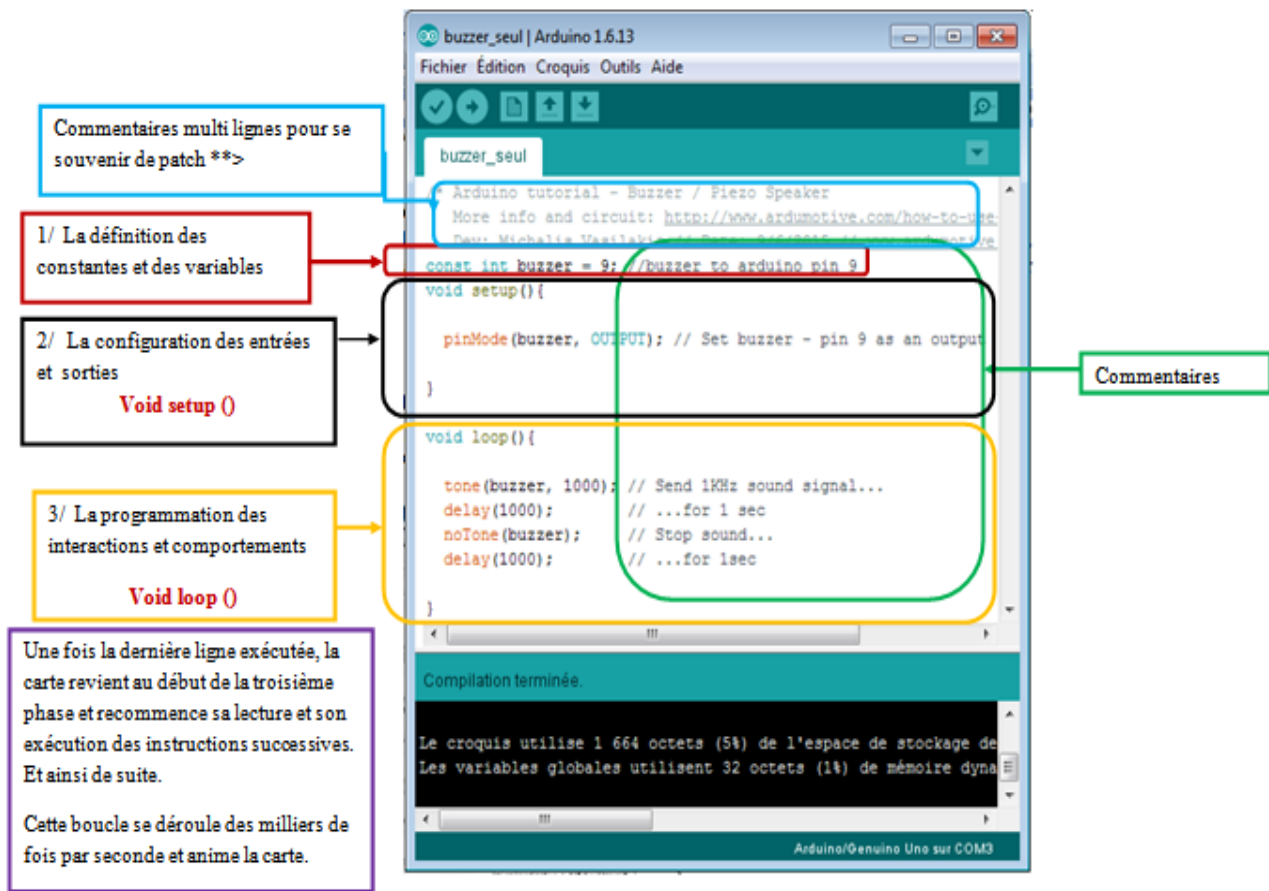


Figure II.4: Structure générale d'un programme sur l'IDE.

III)-Présentation des shields :

Pour la plupart des projets, il est souvent nécessaire d'ajouter des fonctionnalités aux cartes Arduino. Plutôt que d'ajouter soit même des composants extérieurs (sur une platine d'essai, circuit imprimé, etc.), il est possible d'ajouter des shields.

Un shield est une carte que l'on connecte directement sur la carte Arduino qui a pour fonction d'étendre les fonctionnalités de la carte. Ces shields viennent généralement avec une librairie permettant de les contrôler. On retrouve par exemple, des shields Ethernet, de contrôle de moteur, lecteur de carte SD. Le principal avantage de ces shields est leurs simplicités d'utilisation. Il suffit des les emboîter sur la carte Arduino pour les connecter. Les circuits électroniques et les

logiciels sont déjà faits et on peut en empiler plusieurs. C'est un atout majeur pour ces cartes pour ajouter de nouvelles fonctionnalités. La figure II.5 représente une liste des différents shields.



Figure II.5: Liste des différents shields.

IV)- Capteurs utilisés :

IV-1)- Capteur de température et d'humidité de l'air DHT11:

Le capteur DHT11 fournit une sortie numérique proportionnelle à la température et à l'humidité mesurée par le capteur, la technologie utilisée pour produire le capteur DHT11 garantit une grande fiabilité, une excellente stabilité à long terme et un temps de réponse très rapide. L'interface physique du capteur est réalisée par un connecteur à 3 broches: +5V, GND et DATA. Les deux premières broches sont l'alimentation et la terre, elles sont utilisées pour alimenter le capteur, la troisième représente le signal de sortie numérique du capteur. [5]

La figure II.6 représente le capteur DHT11.

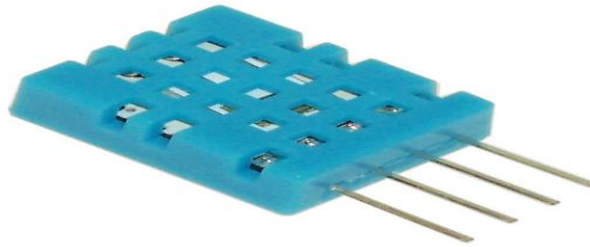


Figure II.6 : Capteur de température et d'humidité DHT11.

➤ **Caractéristiques :**

- Alimentation : 5V.
- Consommation : 0.5 mA en minimal/ 2.5 mA en maximum.
- Etendue de mesure de température : 0°C à $50^{\circ}\text{C} \pm 2^{\circ}\text{C}$.
- Etendue de mesure humidité : 20-90% RH $\pm 5\%$.

➤ **Description des entrées / sorties :**

- Broche 1: 5V.
- Broche 2: sortie numérique.
- Broche 3: non-utilisée
- Broche 4: GND

La figure II.7 présente le câblage du capteur DHT11 avec la carte Arduino.

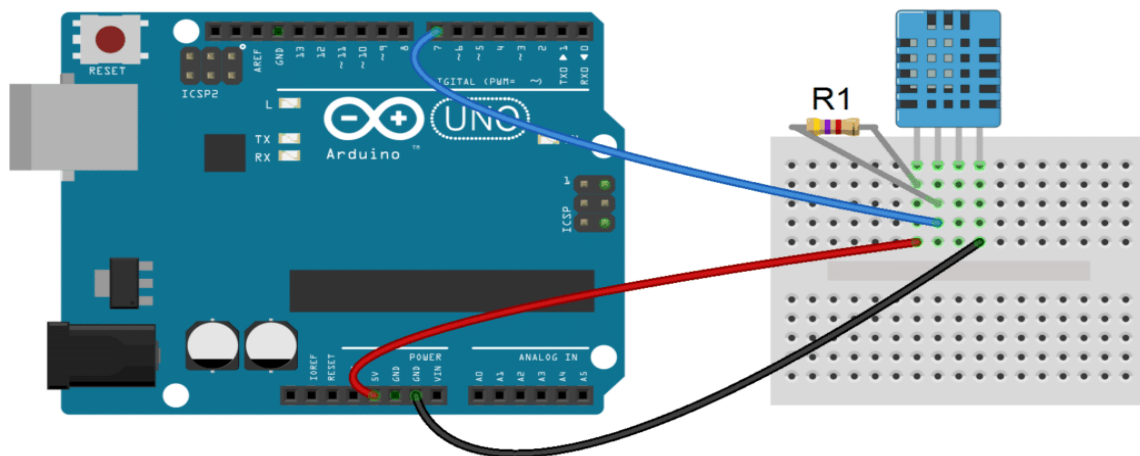


Figure II.7: Câblage du capteur DHT11 sur la carte Arduino UNO.

IV-2)-Capteur de luminosité TSL2561:

Le TSL2561 est un capteur de lumière avancé qui présente un champ d'application important. Il est plus précis qu'une photo résistance. Il est équipé de deux photodiodes: une pour l'infrarouge et l'autre pour une pour le spectre lumineux. Il est léger, muni d'un convertisseur analogique numérique qui transforme l'intensité lumineuse en un signal numérique. Cette sortie numérique peut être injectée dans un microprocesseur pour être calculée en Lux. [6]

Le capteur de luminosité TSL2561 est représenté par la figure II.8.

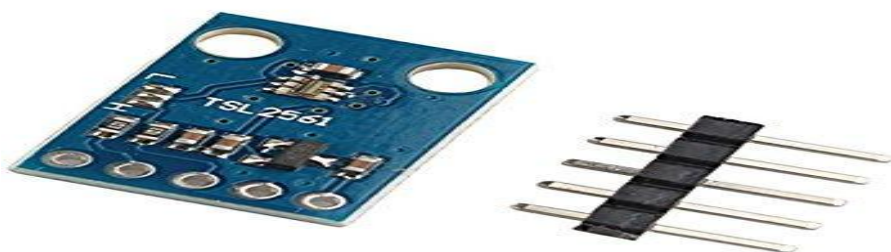


Figure II.8 : Capteur de Luminosité TSL2561.

➤ **Caractéristiques :**

- Le TSL2561 est un capteur de lumière sophistiqué qui a une réponse précise dans le spectre visible
- Capteur : magnétique.
- Sortie : numérique.
- Matériau: Mélange.

➤ **Description des entrées / sorties :**

- VIN → 5V
- GND → GND
- SCL → SCL
- SDA → SDA
- INT → non-utilisée.

La figure II.9 représente le câblage du capteur de luminosité TSL2561 avec la carte Arduino.

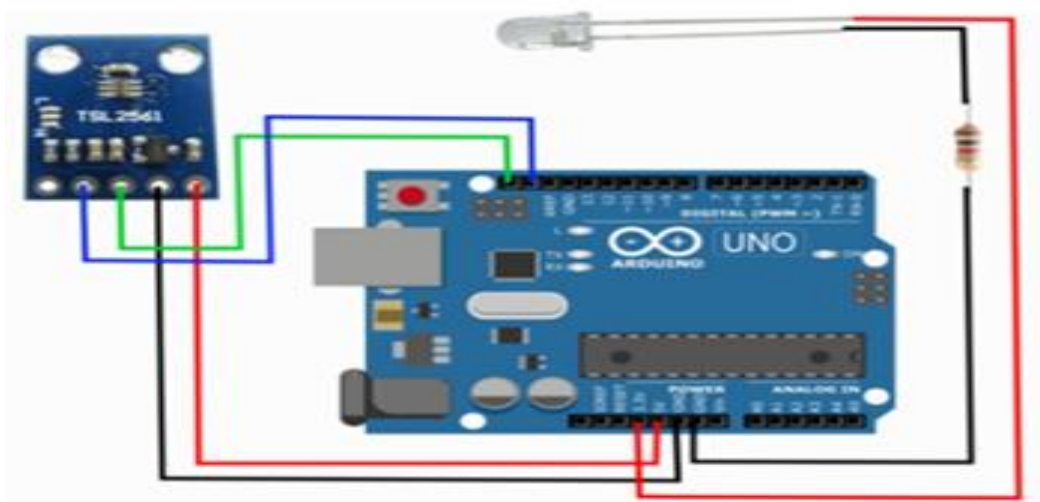


Figure II.9 : Câblage du capteur de luminosité sur l'Arduino UNO.

IV-3)- Capteur de vibration SW-18010P :

Ce capteur est utilisé pour déclencher l'effet de vibration tels que : une alarme antivol, une voiture intelligente, une alarme de tremblement de terre, une alarme de moto, etc. Ce module est comparable avec le module du capteur de vibration de type normalement ouvert où les vibrations de déclenchement, peuvent entraîner le module de relais pendant des périodes de temps longues. La figure II.10 présente le capteur de vibration SW-18010P. [7]



Figure II.10 : Capteur de vibration SW-18010P.

➤ **Caractéristiques:**

- Temps de commutation: 2 ms.
- Résistance de commutation: $<10 \Omega$.
- Température de fonctionnement: -40 à 80°C .
- Durée de vie: 100.000 cycles.
- Dimensions: 05 x 15 mm.
- Longueur des broches: 10 mm.

➤ **Description des entrées / sorties :**

- D0 → Sortie numérique.
- GND → GND.
- VCC → +5V.

La figure II.11 présente le câblage du capteur de vibration SW -18010P.

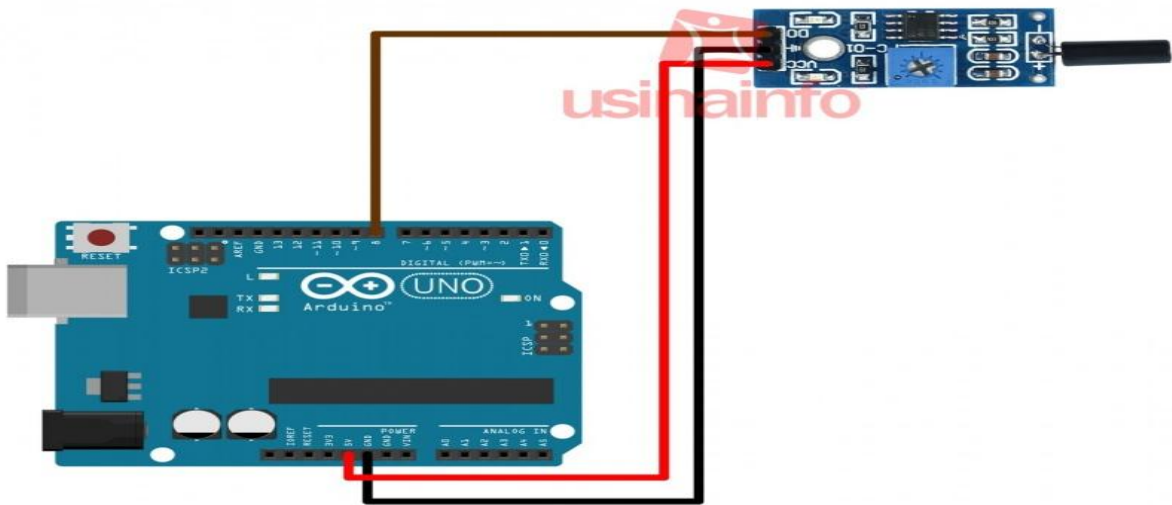


Figure II.11: Câblage du capteur vibration sur Arduino UNO.

IV- 4)- Capteur d'humidité du sol YL69 :

Ce capteur agit comme une résistance dont la valeur dépend de la quantité d'humidité. La figure II.12 présente le capteur d'humidité du sol YL69.[8]

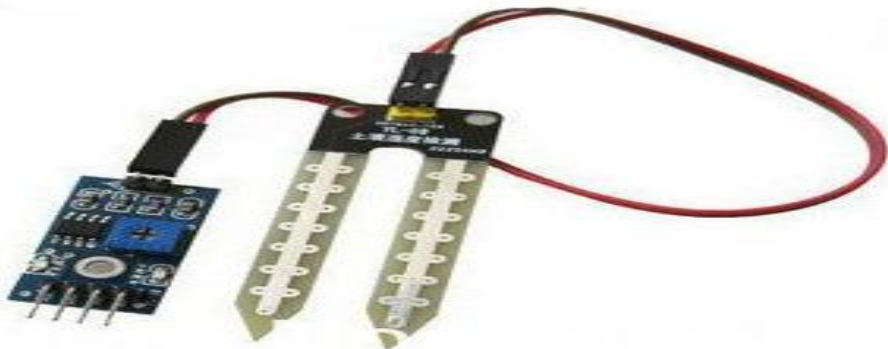


Figure II.12 : Capteur d'humidité du sol YL69.

➤ Caractéristiques:

C'est un capteur d'humidité simple qui peut être utilisé dans un sol est en déficit d'eau (niveau haut) et vice versa (niveau bas). Ce module peut être utilisé pour réaliser des systèmes d'arrosage automatique.

- La sensibilité de ce module est ajustable via un potentiomètre numérique (en bleu sur la photo)
- Tension de fonctionnement : 3,3V-5V
- Module avec 2 sorties: une analogique et une numérique, la sortie numérique est plus précise.
- Le module comprend des perçages pour faciliter sa fixation.
- Taille du PCB: 3cm * 1.6cm.
- Contient un indicateur de tension (LED rouge) et un indicateur de sortie numérique (LED verte).
- Le comparateur est à base d'un LM393 (très stable).

➤ Description des entrées / sorties:

- VCC → 3 V- 5V.
- GND → GND.
- DO → Sortie numérique.
- A1 → Sortie analogique.

La figure II.13 présente le câblage de ce capteur avec la carte Arduino UNO.

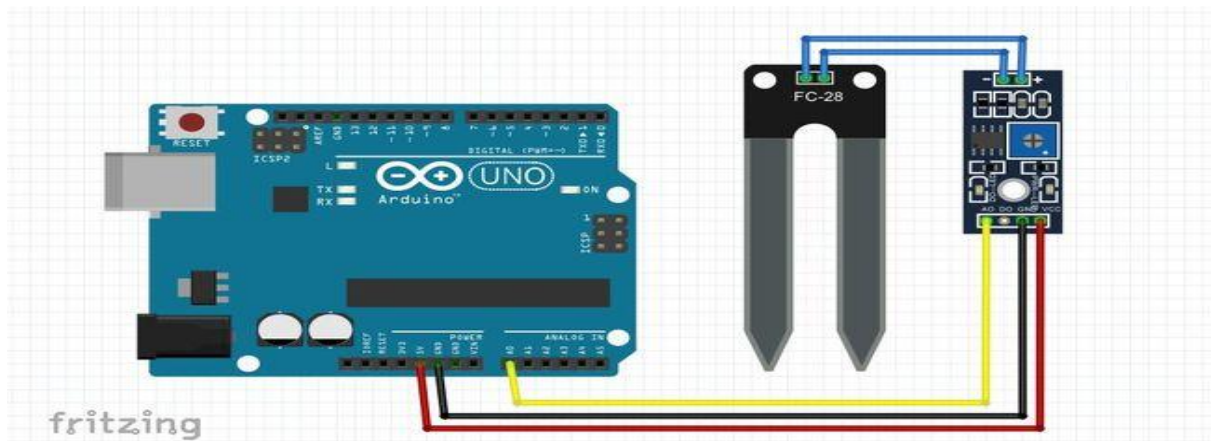


Figure II.13: Câblage du capteur d'humidité du sol sur l'Arduino UNO.

IV-5)- Capteur de pluie :

Ce capteur utilise deux matériaux de hautes qualités sur une plaque de 5,5 x 4,0 cm², protégé contre les oxydations tout en optimisant la conductivité, la durée de vie ainsi que les performances. La sortie du comparateur délivre un courant de plus de 15mA dont le seuil est ajustable via un potentiomètre. [9]

La figure II.14 présente le câblage de ce capteur avec la carte Arduino UNO.



Figure II.14 : Capteur de pluie.

➤ **Caractéristiques :**

Tension d'alimentation : 3,3V à 5V.

Le PCB du comparateur mesure 3.2 cm x 1.5 cm.

Il est composé de 2 parties :

- La carte électronique qui gère le fonctionnement.
- La plaque de détection de pluie.

La plaque permet, au contact de l'eau, de faire passer le courant. Le capteur dispose de 2 sorties :

- Numérique (0 ou 1)
- Analogique (de 0 à 1024)

➤ **Descriptions des entrées / sorties :**

- VCC : 5V
- GND : GND
- D0 : Sortie digital, qui envoi une valeur numérique 0 ou 1.
- A0 : Sortie analogique, qui permet d'avoir une valeur variable et graduelle.

La figure II.15 présente le câblage de ce capteur avec la carte Arduino UNO.

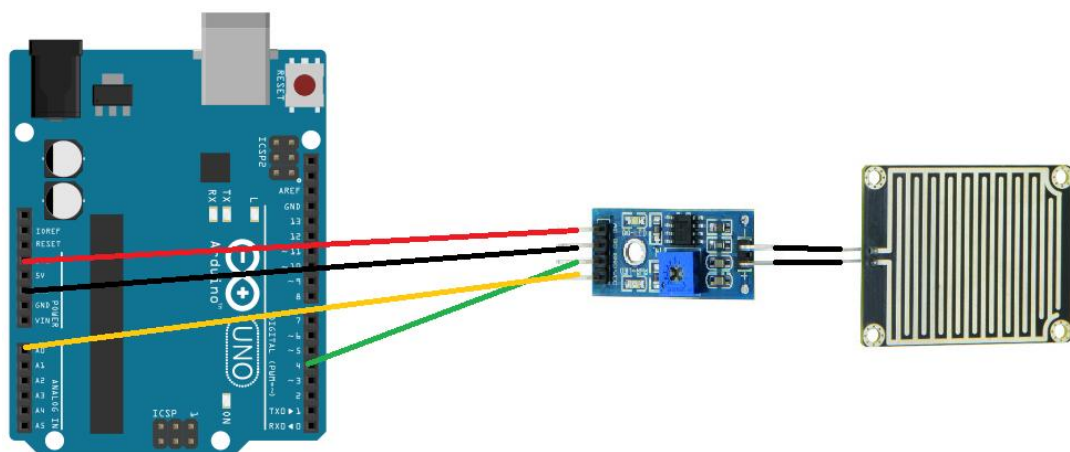


Figure II.15 : Câblage du capteur de pluie sur l'Arduino UNO.

IV-6)- Capteur de pression (BMP280):

Le BMP 280 est un capteur de pression barométrique absolu, spécialement conçu pour les applications mobiles. Ses petites dimensions et sa faible consommation d'énergie permettent la mise en œuvre dans des appareils portables. Le BMP280 est basé sur la technologie éprouvée des capteurs de pression piézo-résistif Bosch. [10]

La figure II.16 présente le capteur de pression BMP 280.

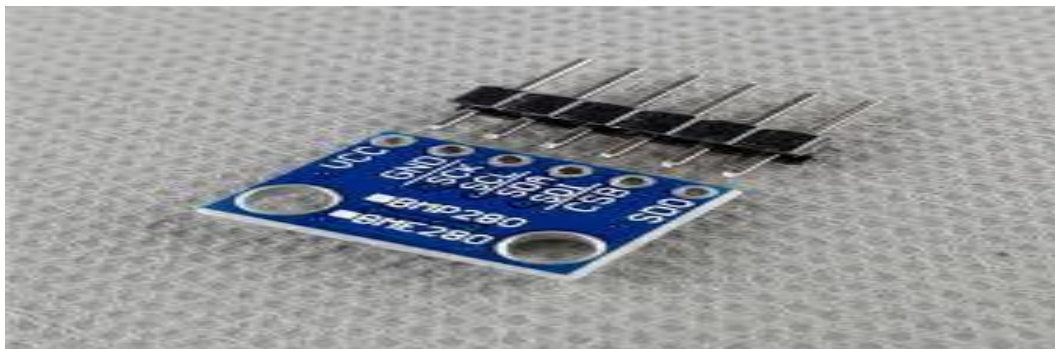


Figure II.16 : Capteur de pression BMP 280.

➤ **Caractéristiques :**

- Alimentation : 3 à 5V.
- Mesure de la température de -40°C à 85°C avec une précision de +/- 1°C.
- Mesure de la pression de 300 hPa à 1100 hPa avec une précision de +/- 1hPa (de 0°C à 65°C) et +/- 1hPa (de -40°C à 0°C).
- Mesure d'humidité relative de 0% à 100%.
- Sortie numérique sur bus I2C ou bus SPI.

➤ **Description des entrées / sorties :**

- VIN → (+3<5v)
- GND → GND
- SCL → SCL (A4)
- SDA → SDA (A5)
- CSB → non connecté
- SD0 → non connecté.

La figure II.17 présente le câblage du capteur BMP280 avec la carte Arduino UNO.

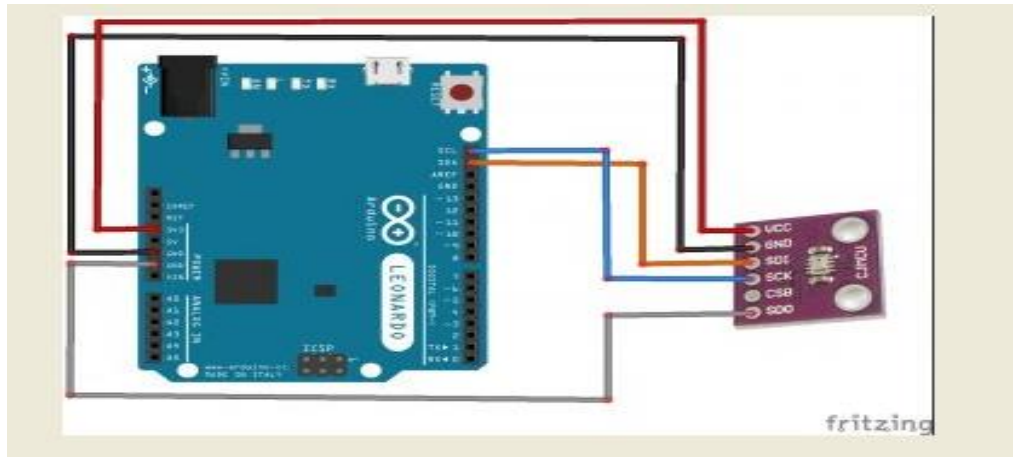


Figure II.17: Câblage du capteur de Pression sur l'Arduino UNO.

V)- Le module Bluetooth HC05:

Le module Bluetooth HC-05 est un module SPP (Serial Port Protocol) facile à utiliser, conçu pour une configuration de connexion série sans fil transparente, sa communication s'effectue via une communication série qui facilite l'interfaçage avec le contrôleur ou le PC. Le module Bluetooth HC-05 fournit le mode de commutation entre le mode maître et le mode esclave. Le module Bluetooth HC-05 est représenté par la figure II.18. [11]

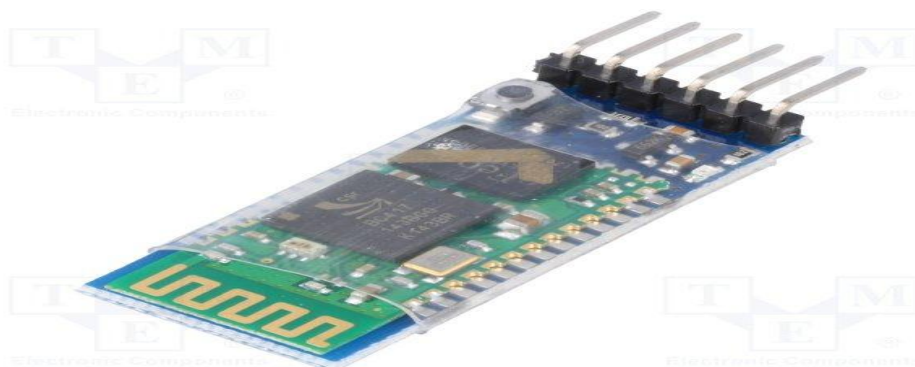


Figure II.18: Module Bluetooth HC-05.

➤ **Caractéristiques :**

- Tension d'entrée: 3,6 à 6 V.
- Portée: jusqu'à 10 mètres
- Liaison série: 4800 à 1382400 bauds
- Sorties: Key, VCC, GND, TXD, RXD et State
- Configuration via commandes AT.
- Dimensions: 35 x 16 x 12 mm.
- Led indique le fonctionnement : clignote si non connecté, éclairage fixe quand il est connecté.

La figure II.19 présente le câblage du module Bluetooth HC-05 avec la carte Arduino UNO.

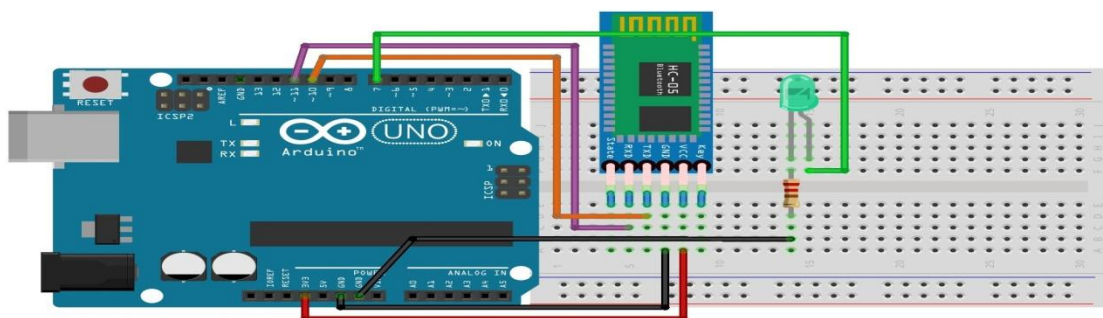


Figure II.19 : Câblage du module HC-05 sur l'Arduino UNO.

VI)- Interface Smartphone et Arduino :

Pour assurer la communication entre les deux modules Arduino/Smartphone, il faut installer une application sur le Smartphone dite le Bluetooth Terminal HC05 ayant pour code pin par défaut : 1234. La figure II.20 représente l'interface Bluetooth Terminal HC05.

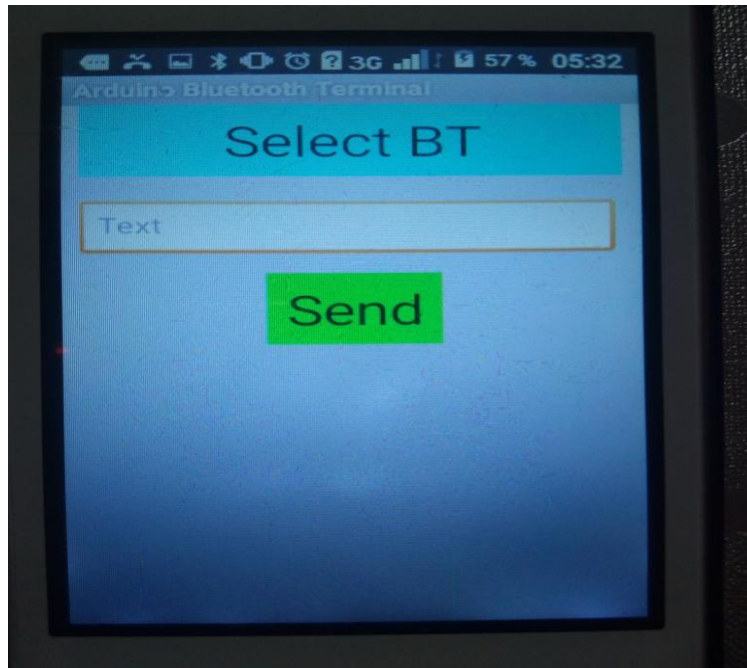


Figure II.20: Interface Bluetooth terminal HC05.

Une fois cette application est installée, la communication Smartphone / Arduino est assurée. La vitesse de transfert série doit être paramétrée convenablement en bauds sinon une série de caractères illisibles apparaîtra dans le moniteur.

VII)- Conclusion :

Dans ce chapitre nous avons exploré les différents outils tels que la carte Arduino, les différents capteurs et le module Bluetooth nécessaires au développement du système de mesure météorologique. Dans ce contexte nous allons exploiter les modules précités pour la réalisation de la station mobile qui sera l'objet du chapitre suivant.

CHAPITRE III :

Conception de la station météo

I)- Préambule :

Le besoin d'observer et de contrôler des phénomènes physiques tels que la température, l'humidité de l'air, du sol, les vibrations ou encore la luminosité et pression est essentiel pour de nombreuses applications industrielles et scientifiques. Cette nécessité nous a amené à réfléchir à la réalisation d'une station météo mobile en utilisant l'application Inventor MIT (Massachusetts Institute of Technologie).

Dans ce chapitre nous allons exploiter les différents modules décrits précédemment pour le développement dans notre propre station. Ensuite introduire l'application Android et son interfaçage avec la carte Arduino UNO pour la transmission des grandeurs mesurées vers un Smartphone via le module Bluetooth.

II)- Architecture du système de mesure:

Ce système est composé de trois parties principales : la première partie représente la partie acquisition composée des différents capteurs (température, humidité, pluie, vibration, pression et luminosité), la deuxième concerne l'unité de traitement des données (Arduino UNO) et la dernière est l'unité de gestion à distance (le module Bluetooth (HC-05)).

La figure III.1 représente le schéma synoptique de la station météorologique mobile.

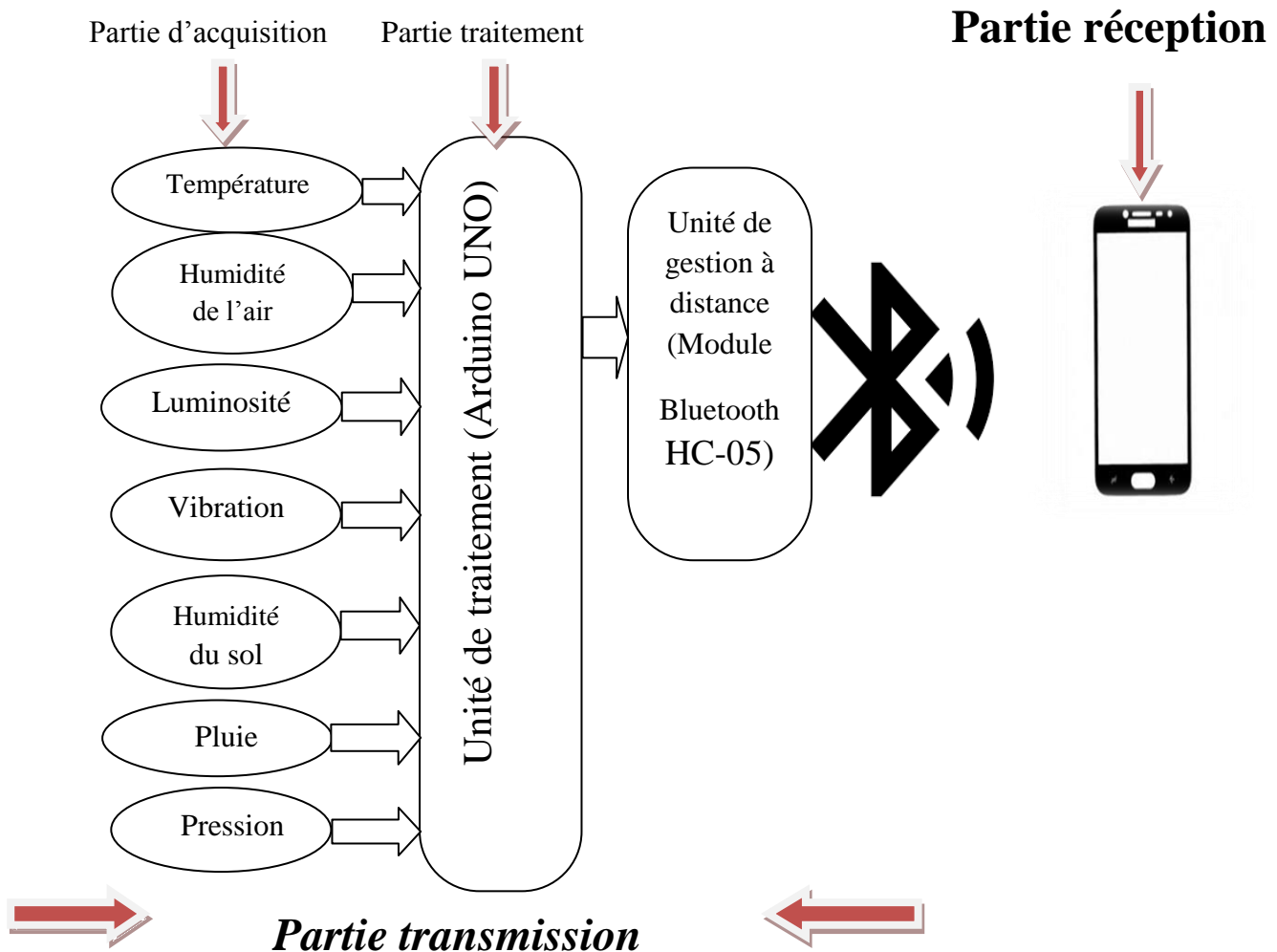


Figure III.1 : Schéma synoptique de la station mobile.

III)- Conception de la station de mesure :

La conception de la station doit passer par les étapes suivantes :

- 1- La réalisation des mesures.
- 2- L'interfaçage Arduino Smartphone.
- 3- Le transfert à distance des mesures de l'Arduino vers le Smartphone.

III-1)- Réalisation des mesures :

Les mesures sont réalisées après le branchement des différents composants avec la carte Arduino et sa programmation. La figure III.2 représente le schéma de connexion de la carte Arduino avec les différents capteurs sur Fritzing.

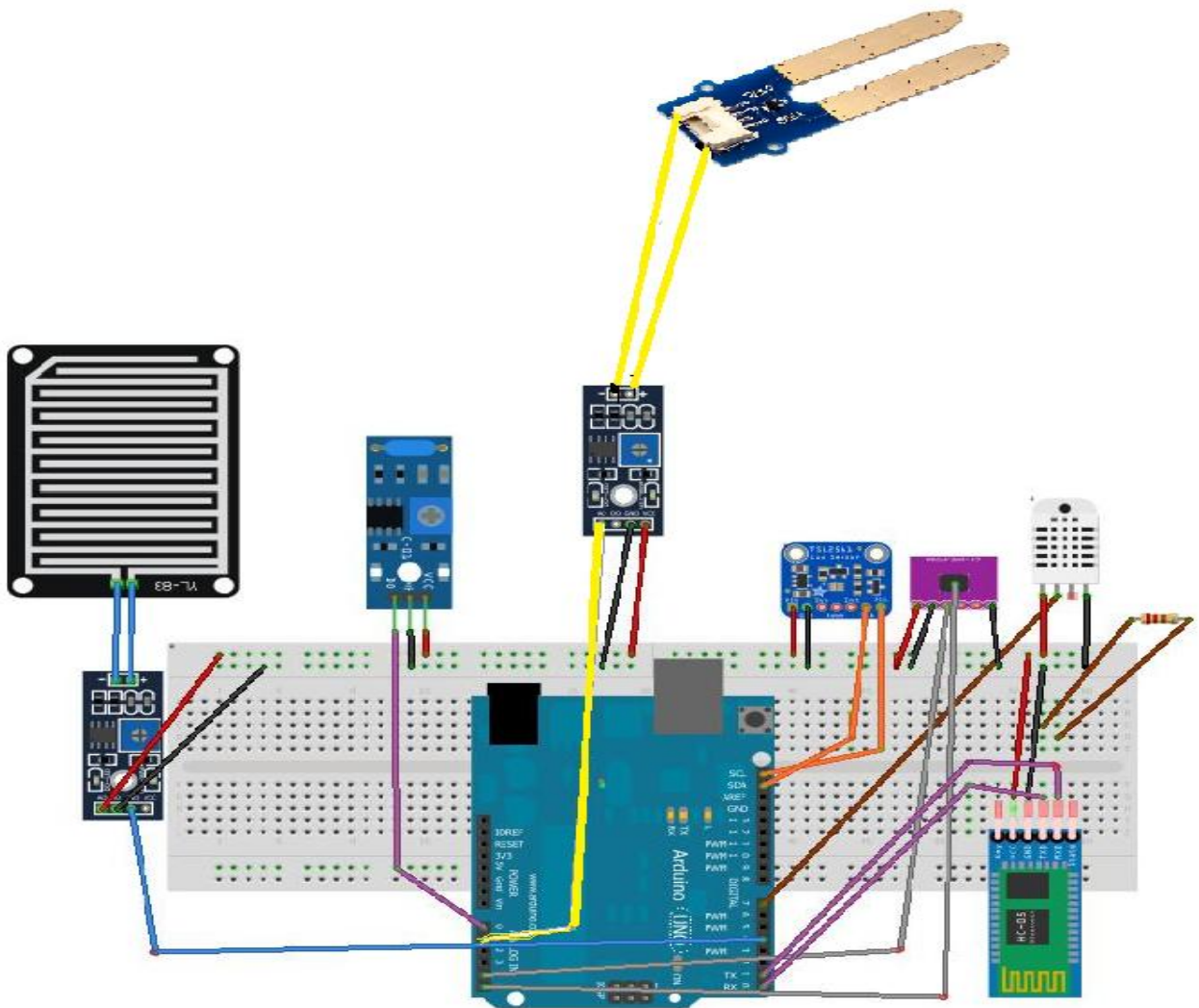


Figure III.2 : Schéma de connexion de la carte Arduino avec les différents capteurs sur Fritzing.

III-1-1)- Programmation du microcontrôleur:

Les étapes de programmation du microcontrôleur sont:

- La création d'un projet.
- L'écriture du programme ensuite enregistrement.
- La vérification de la syntaxe et correction d'éventuelles erreurs.
- La téléversement vers le microcontrôleur

La figure III.3 représente l'organigramme résumant les étapes d'exécution du programme sur le microcontrôleur.

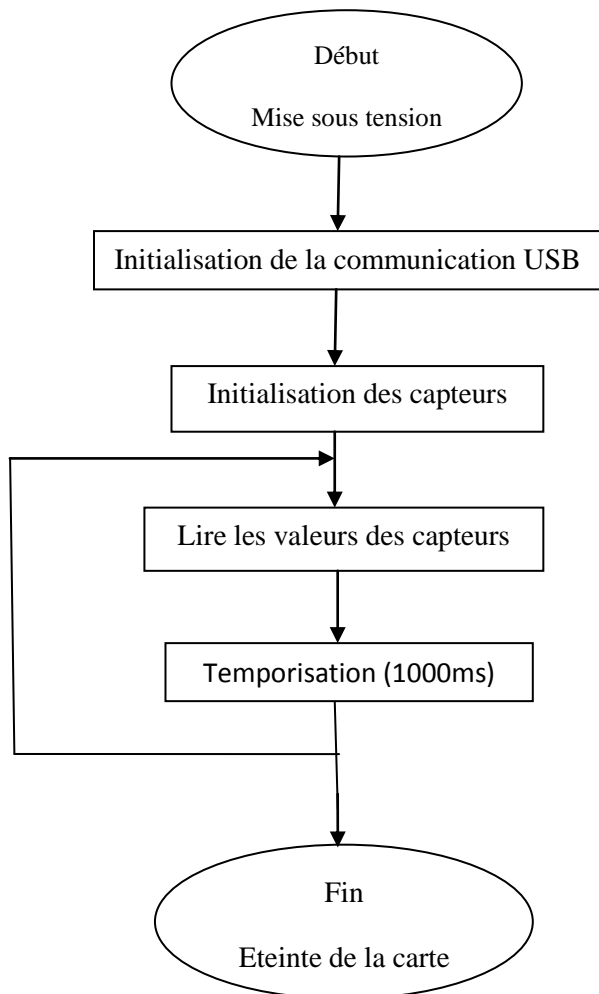


Figure III.3: Organigramme résumant les étapes d'exécution du programme sur le microcontrôleur.

Pour tester le fonctionnement des capteurs utilisés: (DHT11, TSL2561, SW-18010P, YL69, capteur de pluie, BMP280, nous avons réalisé un câblage entre ces capteurs et la carte Arduino UNO. La carte Arduino est connectée au port USB d'un PC pour la visualisation des résultats de mesure sur écran.

Une image de la carte météo réalisée sur une plaque d'essai et connectée par un câble USB est présentée à la figure III.4.

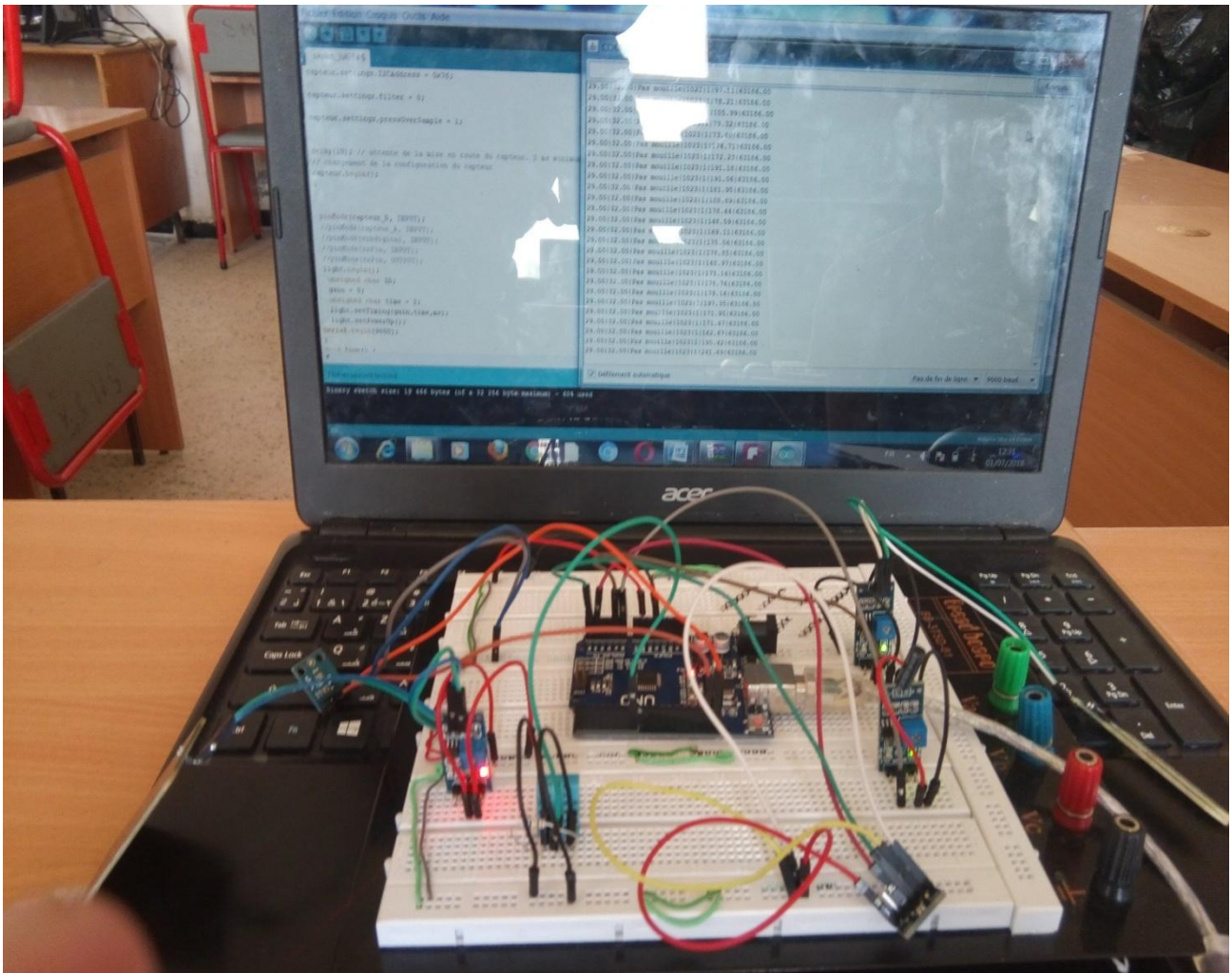


Figure III.4: Image du montage réalisé connecté par USB.

La figure III.5 représente quelques résultats obtenus affichés sur le moniteur du PC.

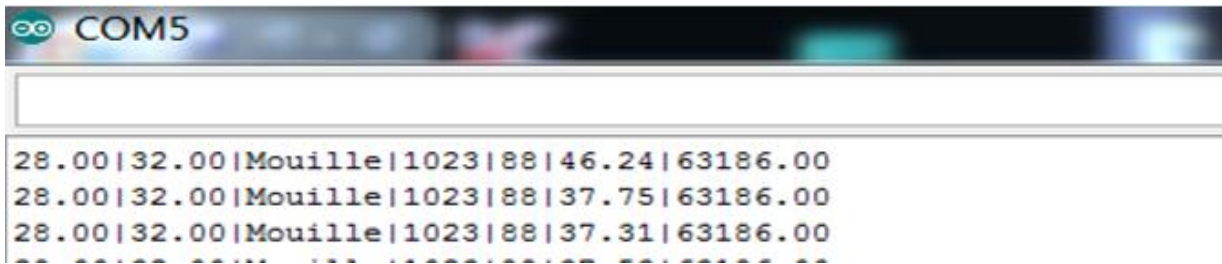


Figure III.5: Résultats affichés sur le moniteur du PC.

III-2)- Interfaçage Arduino/ Smartphone :

III-2-1)- Présentation du système Android :

Android est un système d'exploitation développé initialement pour les Smart phones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC et intègre tous les utilitaires et les périphériques nécessaires à un Smartphone. Il est optimisé pour les outils Gmail. Aussi, Android est libre et gratuit et a été ainsi rapidement adopté par des fabricants.

III-2-2)- Présentation de l'application Inventor MIT:

L'application Inventor est un outil de développement des applications en ligne pour les smart phones sous Android et permet à chacun de créer son application personnelle pour le système d'exploitation Android qui est développé par Google. La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail, elle rappelle certains langages de programmation simplifiés basés sur une interface graphique similaire à Scratch, les informations des applications sont stockées sur des serveurs distants elles sont actuellement entre tenues par le Massachusetts Institute of Technology (MIT). L'environnement d'App Inventor contient trois fenêtres qui sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : permet de créer l'allure d'application (Application Inventor Designer).
- Une pour la programmation par elle-même : elle permet, par l'assemblage des blocs de créer le comportement de l'application (App Inventor Block Editor).

- Une pour l'émulateur: qui permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

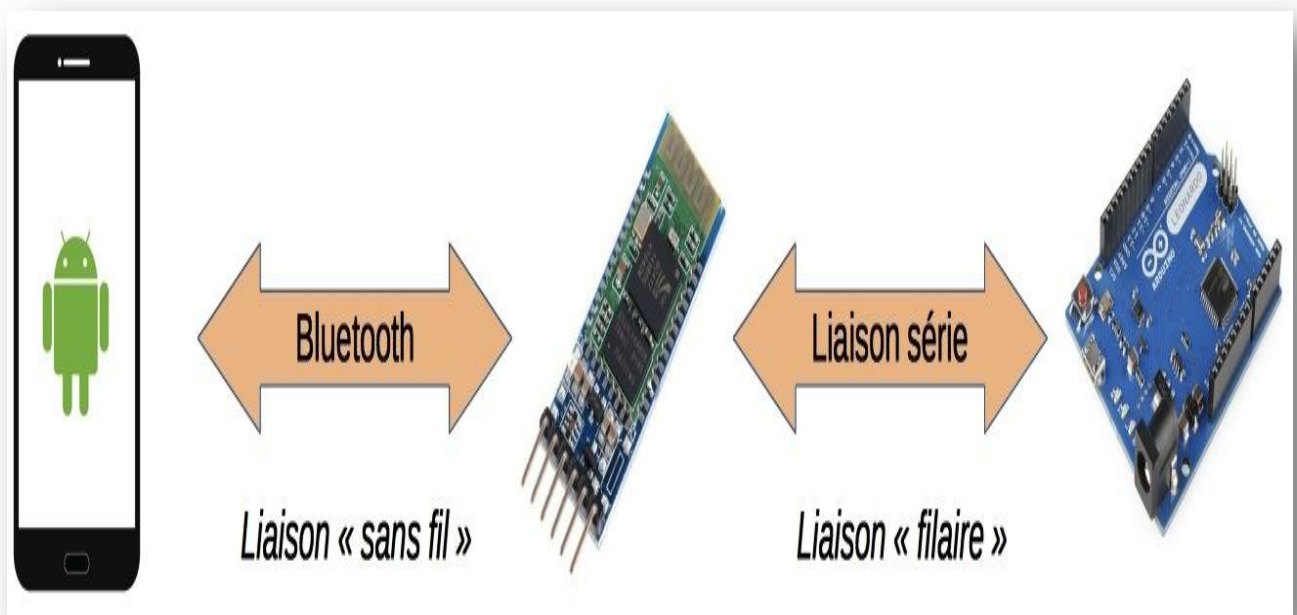
La connexion d'un terminal réel sous Android permettra en suite d'y télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette. Le comportement du programme sera identique. [12]

III-2-3)- Bluetooth :

Le Bluetooth est un standard de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF. Son objet est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires.

Elle peut remplacer par exemple les câbles entre ordinateurs, tablettes, téléphones mobiles entre eux ou avec des imprimantes, scanners, claviers, souris, manettes de jeu vidéo, téléphones portables.

La figure III.6 représente l'interfaçage Arduino/Smartphone :



La figure III.6: Interfaçage Arduino/Smartphone

Pour le principe de fonctionnement, un programme qu'on a développé est exécuté dans le microcontrôleur de la carte Arduino, qui permettant de récupérer toutes les données envoyés par les capteurs, avec un intervalle de temps défini. Une fois que les données reçues, elles seront transmises via Bluetooth à un Smartphone Android qui exécute une application Inventor MIT qu'on a développé pour recevoir les données mesurées par chaque capteur et de les traiter si nécessaire.

La figure III.7 représente l'organigramme de l'interfaçage Arduino/ Smartphone.

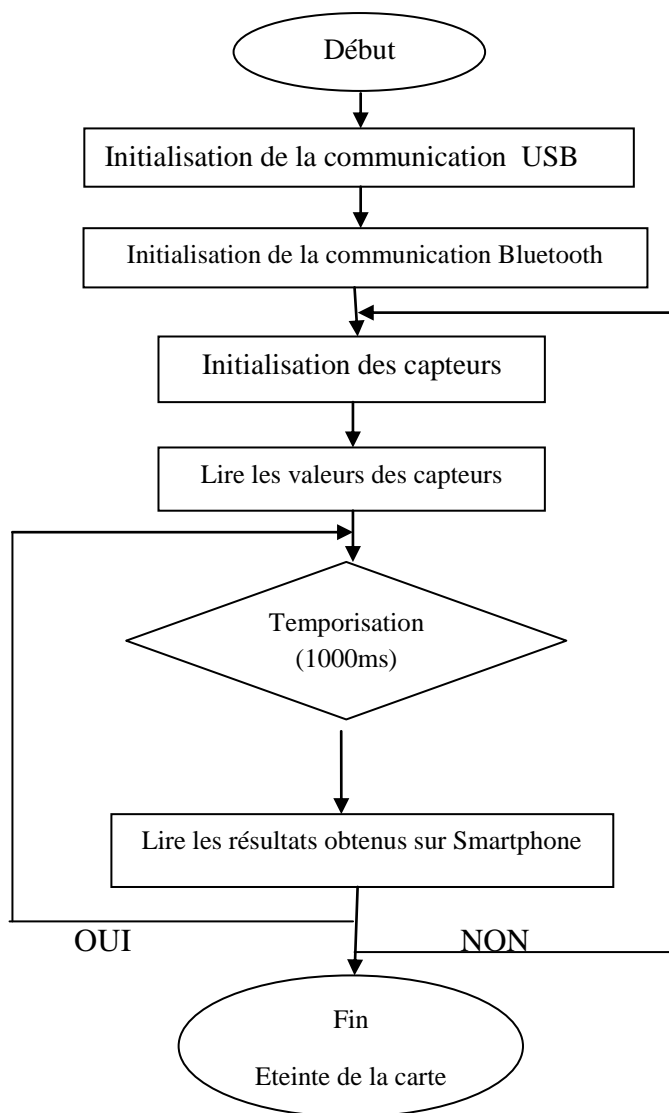


Figure III.7: Organigramme de l'interfaçage Arduino/Smartphone.

III-3)- Transmission Arduino/ Smartphone:

III-3-1)- Programmation Application Android :

Dans le but de contrôler les différents équipements utilisés, nous avons créé une application Android pour smartphone. Pour ce faire nous avons opté pour le « MIT App inventor». Elle simplifie le développement des applications sous Android et les rend accessibles pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Elle est basée sur une interface graphique. La figure III.7 représente l'interface de l'application MIT développée.



Figure III.8:Interface de l'application MIT.

Dans les paragraphes qui vont suivre nous allons détailler cette application (MIT) (sa programmation, son fonctionnement). Dans le bloc de l'interface, nous avons deux boutons pour les connecter/déconnecter au Bluetooth HC-05 et une petite zone de texte qui affiche l'état du Bluetooth.

La figure III.9 représente les étapes de programmation du bloc 1.

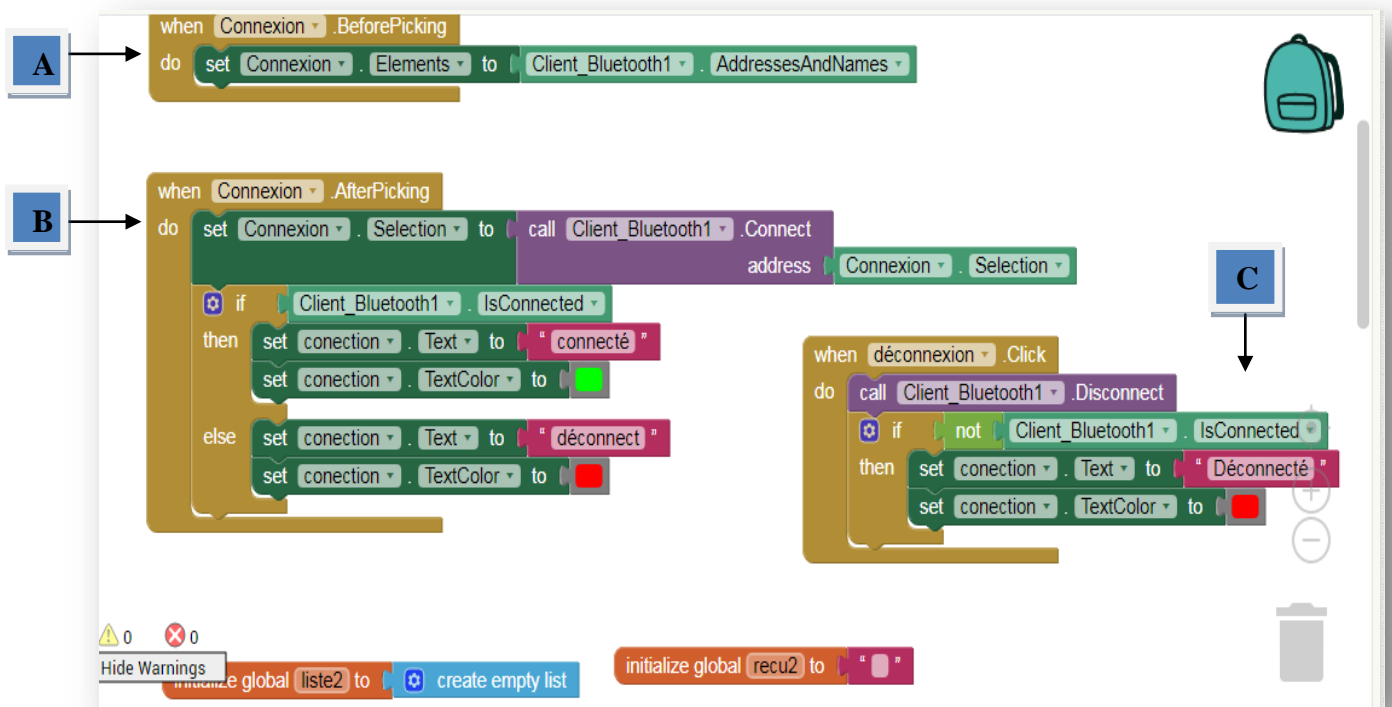


Figure III.9: Etapes de la programmation du bloc 1.

En utilisant des structures conditionnelles, nous avons créé les 2 boutons «connexion » et « déconnexion ».

Le bloc **A** : on cliquant sur le bouton connexion, la liste des Bluetooth s'affichera.

Le bloc **B** : se connecter à l'élément Bluetooth sélectionné par l'utilisateur et afficher le texte « Connecté » si l'application s'est connecté au Bluetooth, ou « Déconnecté » dans le cas contraire.

Le bloc **C** : on cliquant sur le bouton déconnexion l'application se déconnecte du Bluetooth distant et affiche le texte « Déconnecté ».

La figure III.9 représente les étapes de programmation du bloc 2.

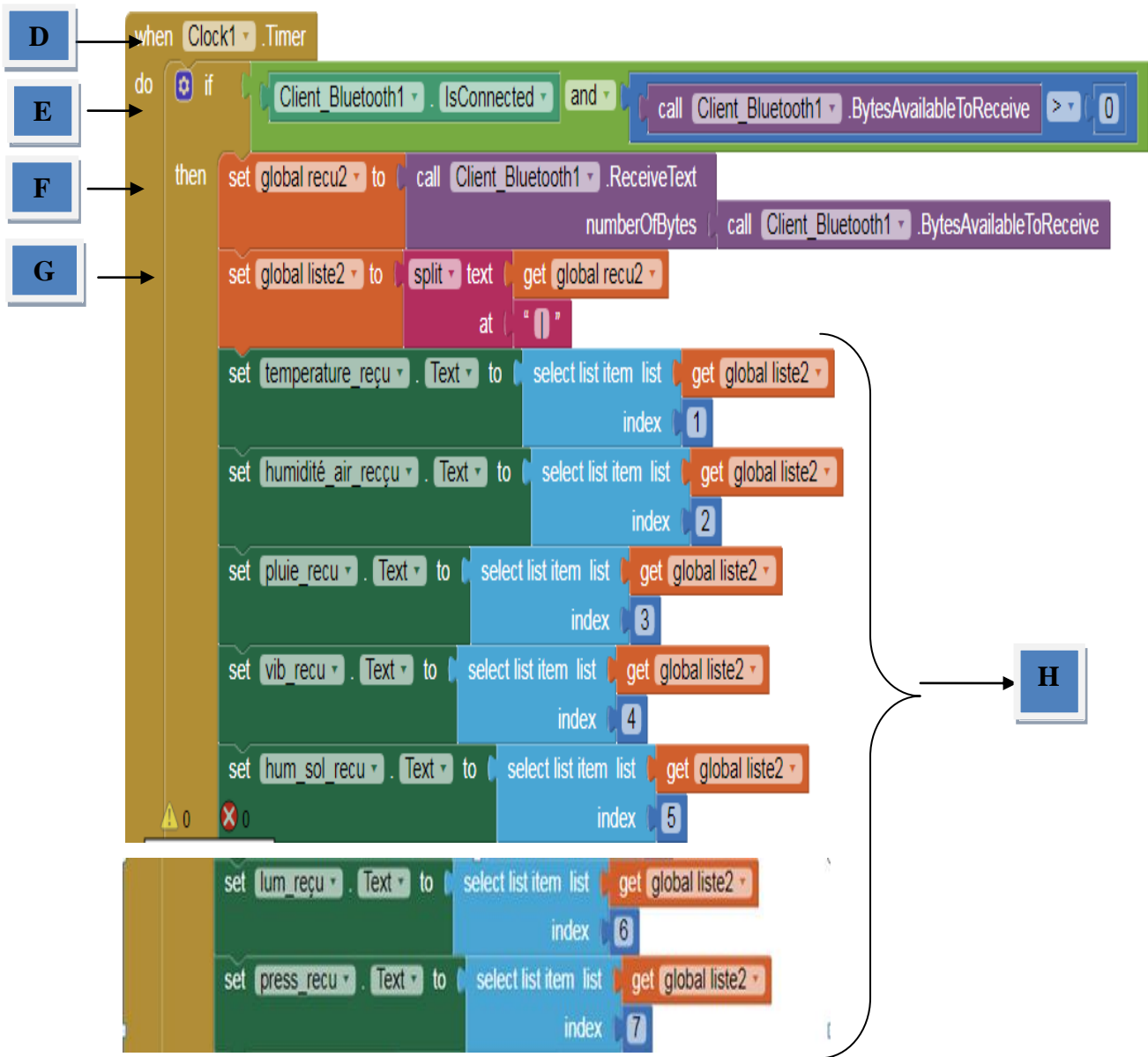


Figure III.10: Etapes de la programmation du bloc 2.

Le bloc **D** : on utilisant l'horloge pour lancer le temps.

Le bloc **E** : connecter au Bluetooth, et appller a toutes les variables qui sont (≥ 0).

Le bloc **F** : mettre toutes les variables dans l'étiquette « reçu2 », et appeler au Bluetooth pour recevoir toutes les données en bytes.

Le bloc **G** :mettre toutes les mesures dans l'étiquette « liste 2 » et les sèparer avec (|).

Le bloc **H** : contient 7 zones de texte (température, humidité de l'air, pluie, vibration, humidité du sol, luminosité, pression), et obtenir des mesures.

La figure III.11 représente les étapes de programmation du bloc 3.

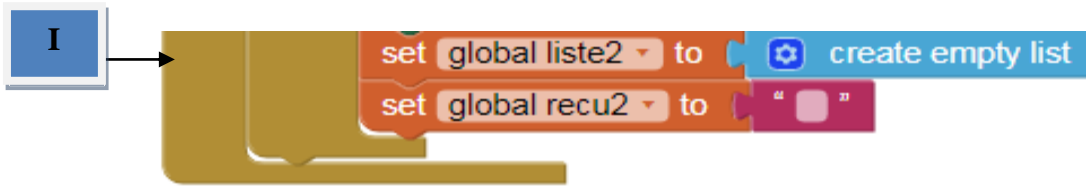


Figure III.11: Etapes de la programmation du bloc 3.

A la fin de chaque information soit mise dans sa place, on va créer une liste vide pour ne pas afficher plusieurs résultats.

IV)-Tests effectués:

Nous avons installé la carte à l'environnement extérieure afin d'effectuer des mesures sur quelques paramètres météorologiques à différents moments de la journée pour évaluer leurs changements tout au long de la journée. Les tests ont été effectués le 01/07/2018 à la commune Yakouren de la wilaya de Tizi-Ouzou, où la journée fut ensoleillée. Les figures ci-dessous représentent les résultats de mesures obtenus sur les paramètres choisis où les mesures sont enregistrées chaque heure de temps.

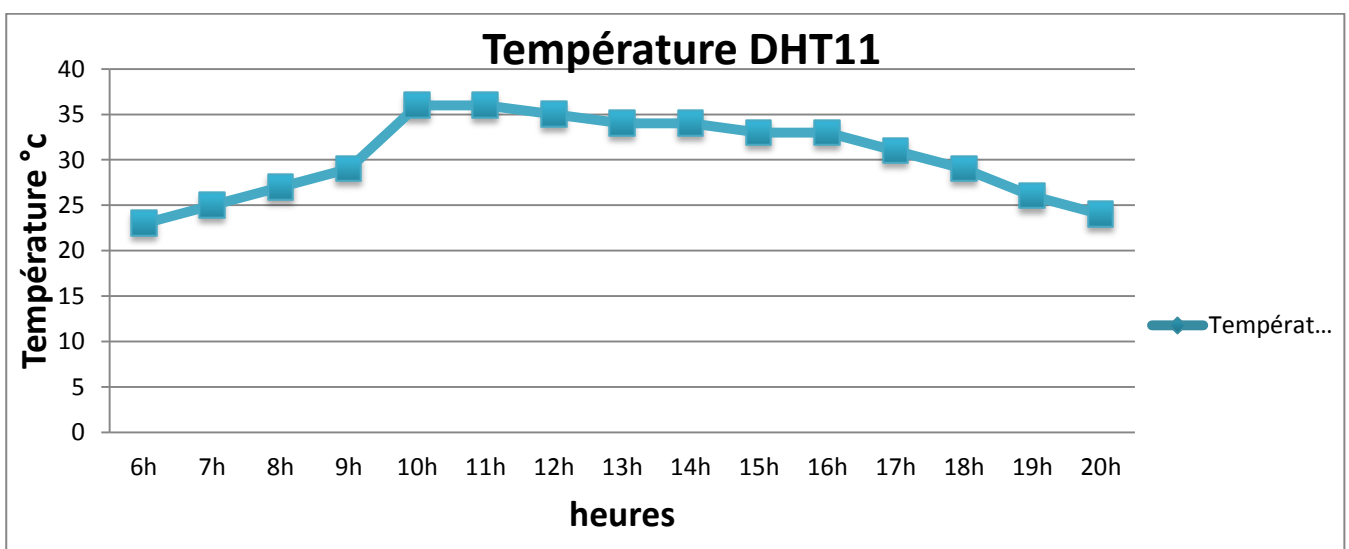


Figure III.12: Courbe graphique du changement de la température en fonction du temps.

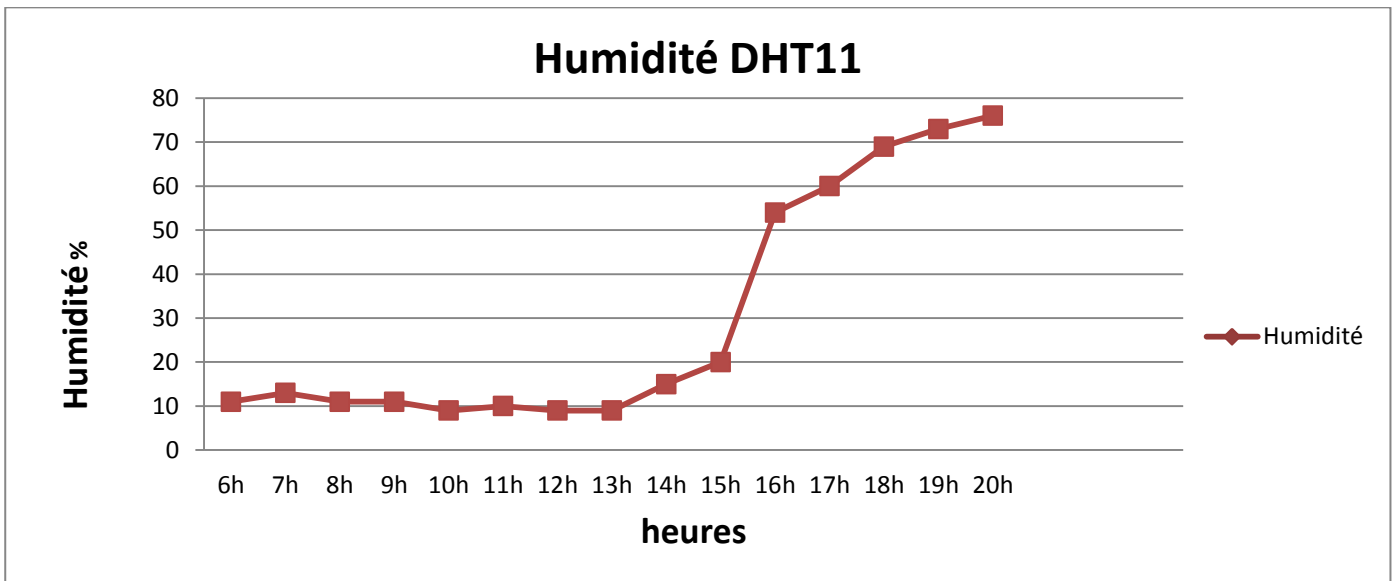


Figure III.13 : Courbe graphique du changement de l'humidité en fonction du temps.

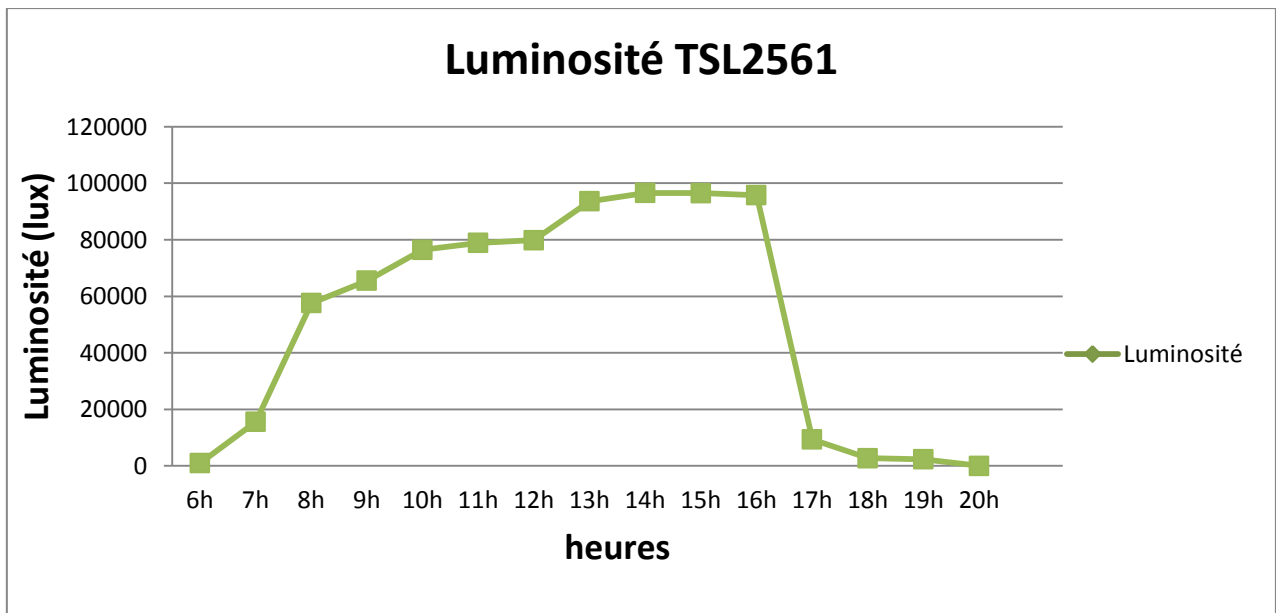


Figure III.14: Courbe graphique du changement de luminosité en fonction du temps.

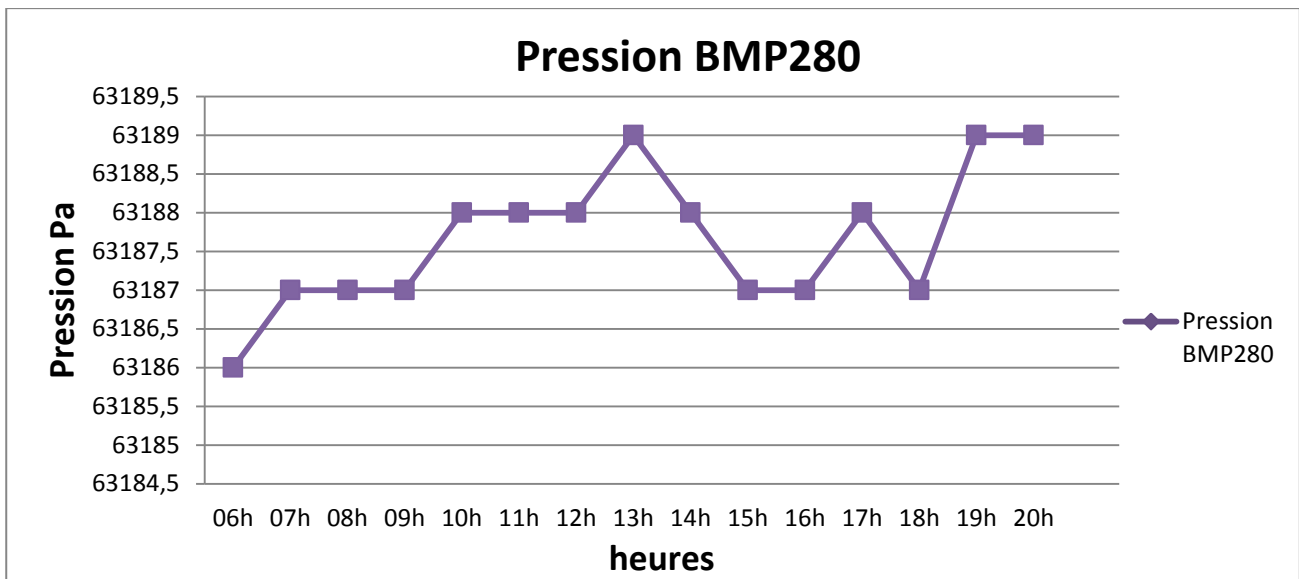


Figure III.15 : Courbe graphique du changement de la pression.

Les courbes obtenues montrent que les paramètres enregistrés montrent des variations à certains moments de la journée pour chacun. D'une façon générale, les grandeurs enregistrées sont étroitement liées à la présence du soleil. Hormis quelques petites variations aléatoires de l'humidité et de la pression.

V-Conclusion :

Dans ce chapitre, nous avons détaillé les étapes de la conception de la station de mesure des paramètres météorologiques. Ces étapes ont consisté d'abord en la réalisation des mesures avec la carte à microcontrôleur et affichage des résultats sur PC, l'interfaçage Arduino Smartphone et le transfert à distance des mesures de l'Arduino vers le Smartphone. Enfin nous avons réalisé des tests avec la station réalisée à l'air libre sur une certaine période de temps sur une journée choisie. Les mesures obtenues ont montré une certaine variation à des moments précis au cours de cette journée dont les résultats obtenus s'avèrent satisfaisant.

CONCLUSION ET PERSPECTIVES

Conclusion et perspectives

Conclusion perspectives

Tout au long de ce travail nous avons cherché à concevoir une station météorologique mobile. Cette station est composée d'un ensemble de capteurs qui sont : un capteur de température et d'humidité de l'air, un capteur d'humidité au sol, un capteur de pression atmosphérique, un capteur de luminosité, un capteur de vibration et un capteur de pluie ayant pour rôle l'acquisition des données, d'une carte à microcontrôleur Arduino dont le rôle est le traitement des données issus des différents capteurs, d'un module Bluetooth pour la transmission à distance en temps réel des mesures obtenues et d'un Smartphone pour la réception et l'affichage des différentes mesure obtenues.

La conception de la station est composée de plusieurs étapes qui sont : le branchement des capteurs avec la carte Arduino, la programmation de la carte sur PC et l'affichage des résultats sur moniteur. L'étape suivante a consisté à assurer la transmission à distance vers un Smartphone, pou cela nous avons installé l'application Bluetooth terminal HC05 permettant d'assurer l'interfaçage entre les deux module Arduino et Smartphone. Pour conclure, nous avons effectué des tests à l'air libre avec la station et obtenu des mesures sur une certaine période de temps dont les résultats obtenus s'avèrent satisfaisants.

Comme perspectives de ce travail, des extensions sur la station réalisée peuvent être considérées tenant comptes d'autres grandeurs météorologiques comme la vitesse et la direction du vent. Le champ d'application de cette station peut être exploité pour développer un réseau de stations afin d'obtenir des mesures spatiales sur une grande échelle.

En outre les méthodes et les techniques développées au cours de ce travail peuvent être étendues à d'autres types de stations, afin d'améliorer leurs performances et obtenir des mesures précises sur les grandeurs météorologiques à mesurer.

Nous espérons que ce mémoire sera une référence et enrichi par les promotions futures.

ANNEXES

I- Programme capteur de température et d'humidité de l'air DHT11 :

```
#include <dht.h>

dht DHT;

#define DHT11_PIN 7

void setup(){

Serial.begin(9600);

}

void loop()

{

int chk = DHT.read11(DHT11_PIN);

Serial.print("Temperature = ");

Serial.println(DHT.temperature);

Serial.print("Humidity = ");

Serial.println(DHT.humidity);

delay(1000);

}
```

II- Programme capteur de luminosité TSL2561 :

```
#include <SFE_TSL2561.h>

#include <Wire.h>

// Create an SFE_TSL2561 object, here called "light":

SFE_TSL2561 light;

// Global variables:

boolean gain; // Gain setting, 0 = X1, 1 = X16;
```

```
unsigned int ms; // Integration ("shutter") time in milliseconds

void setup()

{

// Initialize the Serial port:

Serial.begin(9600);

Serial.println("TSL2561 example sketch");

// Initialize the SFE_TSL2561 library

// You can pass nothing to light.begin() for the default I2C address (0x39),

// or use one of the following presets if you have changed

// the ADDR jumper on the board:

// TSL2561_ADDR_0 address with '0' shorted on board (0x29)

// TSL2561_ADDR default address (0x39)

// TSL2561_ADDR_1 address with '1' shorted on board (0x49)

light.begin();

// Get factory ID from sensor:

// (Just for fun, you don't need to do this to operate the sensor)

unsigned char ID;

if (light.getID(ID))

{

Serial.print("Got factory ID: 0X");

Serial.print(ID,HEX);

Serial.println(", should be 0X5X");

}

}
```

```
// Most library commands will return true if communications was successful,  
  
// and false if there was a problem. You can ignore this returned value,  
  
// or check whether a command worked correctly and retrieve an error code:  
  
else  
  
{  
  
byte error = light.getError();  
  
printError(error);  
  
}  
  
gain = 0;  
  
// If time = 0, integration will be 13.7ms  
  
// If time = 1, integration will be 101ms  
  
// If time = 2, integration will be 402ms  
  
// If time = 3, use manual start / stop to perform your own integration  
  
unsigned char time = 2;  
  
// setTiming() will set the third parameter (ms) to the  
  
// requested integration time in ms (this will be useful later):  
  
Serial.println("Set timing...");  
  
light.setTiming(gain,time,ms);  
  
// To start taking measurements, power up the sensor:  
  
Serial.println("Powerup...");  
  
light.setPowerUp();  
  
// The sensor will now gather light during the integration time.  
  
// After the specified time, you can retrieve the result from the sensor.
```

```
// Once a measurement occurs, another integration period will start.

}

void loop()

{

delay(ms);

unsigned int data0, data1;

if (light.getData(data0,data1))

{

// getData() returned true, communication was successful

Serial.print("data0: ");

Serial.print(data0);

Serial.print(" data1: ");

Serial.print(data1);

double lux; // Resulting lux value

boolean good; // True if neither sensor is saturated

// Perform lux calculation:

good = light.getLux(gain,ms,data0,data1,lux);

// Print out the results:

Serial.print(" lux: ");

Serial.print(lux);

if (good) Serial.println(" (good)"); else Serial.println(" (BAD)");

}

else
```

```
{  
  
// getData() returned false because of an I2C error, inform the user.  
  
byte error = light.getError();  
  
printError(error);  
  
}  
  
}  
  
void printError(byte error)  
  
// If there's an I2C error, this function will  
  
// print out an explanation.  
  
{  
  
Serial.print("I2C error: ");  
  
Serial.print(error,DEC);  
  
Serial.print(", ");  
  
switch(error)  
  
{  
  
case 0:  
  
Serial.println("success");  
  
break;  
  
case 1:  
  
Serial.println("data too long for transmit buffer");  
  
break;  
  
case 2:  
  
Serial.println("received NACK on address (disconnected?);");  
  
}
```

```
break;

case 3:

Serial.println("received NACK on data");

break;

case 4:

Serial.println("other error");

break;

default:

Serial.println("unknown error");

}

}
```

III- Programme capteur de vibration SW-18010P:

```
#define ANALOG_IN_PIN 0
#define VIBRATION_SENSOR_PIN 5
int motionDetected = LOW;
int sensorVal = 0;
void setup() {
Serial.begin(9600);
}
void loop() {
motionDetected = digitalRead(VIBRATION_SENSOR_PIN);
sensorVal = analogRead(ANALOG_IN_PIN);
Serial.print("Anaologue: ");
Serial.print(sensorVal);
Serial.print(" Digital :");
Serial.println(motionDetected);
Delay (100);
}
```

IV- Programme de capteur d'humidité du sol :

```
int PinSensor = A0;

int ValeurSensor = 0;

int pourcentage = 0;

void setup(){

Serial.begin(9600);

}

void loop(){

ValeurSensor = analogRead(PinSensor);

pourcentage = ConvertEnPercent(ValeurSensor);

AfficheValeurEtPourcentage();

delay(1000);

}

int ConvertEnPercent(int value){

int ValeurPorcentage = 0;

ValeurPorcentage = map(value, 1023, 465, 0, 100);

return ValeurPorcentage;

}

void AfficheValeurEtPourcentage(){

Serial.print("Valeur: ");

Serial.println(ValeurSensor);

Serial.print("Pourcentage: ");

Serial.print(pourcentage);
```

```
Serial.println("%");  
  
Serial.println("-----");  
  
}
```

V-Programme capteur de pluie :

```
const int capteur_D = 4; // pin connecté à la sortie digital du capteur  
  
const int capteur_A = A1; // pin connecté à la sortie analogique du capteur  
  
int val_analogique;  
  
void setup()  
{  
  
pinMode(capteur_D, INPUT);  
  
pinMode(capteur_A, INPUT);  
  
Serial.begin(9600);  
  
}  
  
void loop()  
{  
  
//Partie Digital  
  
if(digitalRead(capteur_D) == LOW) // Le capteur est actif à l'état bas  
{  
  
Serial.println("Valeur digital : mouille");  
  
delay(10); // Tempo  
  
}  
  
else
```

```
{  
  
Serial.println("Valeur digital : Pas mouille");  
  
delay(10); // Tempo  
  
}  
  
// Partie analogique  
  
val_analogique=analogRead(capteur_A); // on lit la broche capteur "analogique"  
  
Serial.print("Valeur analogique : ");  
  
Serial.println(val_analogique); // afficher la valeur de detection sur la liison série  
  
Serial.println("");  
  
delay(1000);  
  
}
```

VI- Programme capteur de pression :

```
#include "SparkFunBME280.h"  
  
//Library allows either I2C or SPI, so include both.  
  
#include "Wire.h"  
  
#include "SPI.h"  
  
BME280 capteur;  
  
void setup() {  
  
Serial.begin(9600);  
  
while (!Serial) {  
  
// Attente de l'ouverture du port série pour Arduino LEONARDO  
  
}
```

```
//configuration du capteur

capteur.settings.commInterface = I2C_MODE;

capteur.settings.I2CAddress = 0x76;

capteur.settings.tStandby = 0;

capteur.settings.pressOverSample = 1;

Serial.println("Starting BME280... ");

delay(10); // attente de la mise en route du capteur. 2 ms minimum

// chargement de la configuration du capteur

capteur.begin();

}

void loop() {

Serial.print("\t Pression: ");

Serial.print(capteur.readFloatPressure(), 2);

Serial.print(" Pa");

delay(1000);

}
```

VII- Programme du module Bluetooth :

```
#include <SoftwareSerial.h>

#define rxPin 11 // Broche 11 en tant que RX, à raccorder sur TX du HC-05

#define txPin 10 // Broche 10 en tant que TX, à raccorder sur RX du HC-05

SoftwareSerial mySerial(rxPin, txPin);

void setup()
```

```
{  
// define pin modes for tx, rx pins:  
pinMode(rxPin, INPUT);  
pinMode(txPin, OUTPUT);  
mySerial.begin(9600);  
Serial.begin(9600);  
}  
void loop()  
{  
int i = 0;  
char someChar[32] = {0};  
// when characters arrive over the serial port...  
if(Serial.available()) {  
do{  
someChar[i++] = Serial.read();  
delay(3);  
}while (Serial.available() > 0);  
mySerial.println(someChar);  
Serial.println(someChar);  
}  
while(mySerial.available())  
Serial.print((char)mySerial.read());  
}
```

VIII- Programme final:

```
#include <SFE_TSL2561.h>

#include <Wire.h>

SFE_TSL2561 light;

#include<dht.h>

dht DHT;

#define DHT11_PIN 7

#include "SparkFunBME280.h"

//Library allows either I2C or SPI, so include both.

#include "Wire.h"

#include "SPI.h"

BME280 capteur;

boolean gain; // Gain setting, 0 = X1, 1 = X16;

unsigned int ms; // Integration ("shutter") time in milliseconds

const int capteur_D=4; // capteur digital pluie (0 ou 1)

//const int capteur_A=A1; // capteur analogique pluie (0 à 1023)

const int vibanalog=A0; // capteur analogique vibration

//const int vibdigital=7; // capteur digital pluie

bool vibration=LOW; // valeur du capteur de vibration digital (0 ou 1)

int vibration_valeur; // valeur du capteur de vibration analogique (0 à 1023)

const int humidite_sol=A1; // humidité du sol (0 à 1023)

void setup() {{

Serial.begin(9600);
```

```
while (!Serial) {  
  
// Attente de l'ouverture du port série pour Arduino LEONARDO  
  
}  
  
//configuration du capteur  
  
capteur.settings.commInterface = I2C_MODE;  
  
capteur.settings.I2CAddress = 0x76;  
  
capteur.settings.filter = 0;  
  
capteur.settings.pressOverSample = 1;  
  
Serial.println("Starting BME280... ");  
  
delay(10); // attente de la mise en route du capteur. 2 ms minimum  
  
// chargement de la configuration du capteur  
  
capteur.begin();  
  
}  
  
pinMode(capteur_D, INPUT);  
  
//pinMode(capteur_A, INPUT);  
  
//pinMode(vibdigital, INPUT);  
  
//pinMode(rxPin, INPUT);  
  
//pinMode(txPin, OUTPUT);  
  
light.begin();  
  
unsigned char ID;  
  
gain = 0;  
  
unsigned char time = 2;  
  
light.setTiming(gain,time,ms);
```

```
light.setPowerUp();

Serial.begin(9600);

}

void loop() {

int chk= DHT.read11(DHT11_PIN);

//vibration= digitalRead(vibdigital);

vibration_valeur= analogRead(vibanalog);

int valeur_humidite_sol=analogRead(humidite_sol);

valeur_humidite_sol=constrain(valeur_humidite_sol, 120, 1023);

int pourcentage_humidite=map(valeur_humidite_sol, 1023, 120, 0, 100);

delay(ms);

unsigned int data0, data1;

Serial.print(DHT.temperature);

Serial.print("|");

Serial.print(DHT.humidity);

Serial.print("|");

if(digitalRead(capteur_D)==LOW)

{ Serial.print("Mouille"); }

else

{ Serial.print("Pas mouille");}

Serial.print("|");

Serial.print(vibration_valeur);

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

```
Serial.print(pourcentage_humidite);

Serial.print("|");

if (light.getData(data0,data1))

{

double lux; // Resulting lux value

boolean good; // True if neither sensor is saturated

// Perform lux calculation:

good = light.getLux(gain,ms,data0,data1,lux);

// Print out the results:

Serial.print(lux);

}Serial.print("|");

Serial.println(capteur.readFloatPressure(), 2);

delay(1000);

}
```

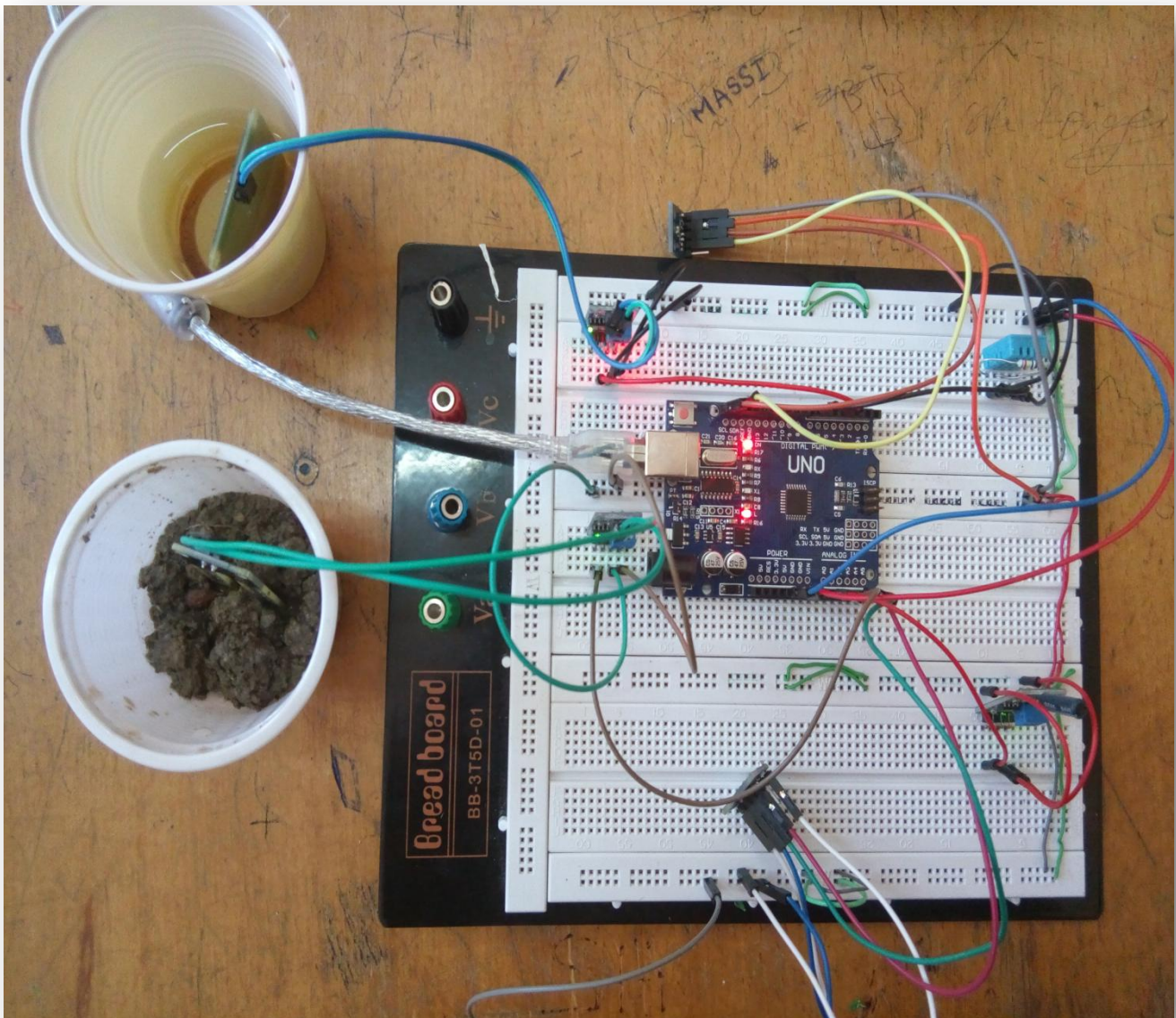


Photo de la réalisation pratique

Les logiciels utilisés



Pour programmer la carte Arduino UNO



Pour la programmation de l'application MIT



Pour l'implantation des composants électroniques

BIBLIOGRAPHIE

Bibliographie

[1] : <http://www.meteolafleche.com/cours.html>

[2] : <http://www.cimel.fr/?acquisition-unit=unite-dacquisition>

[3] : <https://fr.wikipedia.org/wiki/photographie>

[4] : <https://www.arduino.cc>

[5] : <https://www.eagle-robotics.com/accueil/85-capteur-de-DHT11.html>

[6] : [Arduino-passion/le-module-capteur-de-luminosite](http://Arduino-passion.com/le-module-capteur-de-luminosite)

[7] : https://www.gotronic.fr/art-capteur-de-vibrations-haute-sensibilite-sw18010p-22186.htm#complte_desc

[8] : <https://www.google.dz/search?q=Capteur+d'humidité+du+sol+YL69&oq>

[9] : <https://www.kzenjoy.net/2016/capteur-de-pluie-arduino/>

[10] : <https://www.eagle-robotics.com/accueil/85-capteur-de-pression-bmp280.html>

[11] : <http://tiptopboards.com/347-module-bluetooth-hc05-.html>

[12] : Mathieu DUMONT; Loick ATTAGNANT; Killian JACQUET. *Arduino* [en ligne].

Publié le 13/01/2017. KRAMA Abdelbasset, GOUGUI Abdelmoumen. 08/06/2015.

Liste des figures :

Figure I.1 : Capteur de température Cimetl	7
Figure I.2: Capteur d'humidité Cimetl	8
Figure I.3 : Capteur du vent Cimetl.....	9
Figure I.4 : Le pluviomètre.....	10
Figure I.5: Capteur de rayonnement.....	11
Figure II.1: Vue de la carte Arduino UNO.....	16
Figure II.2 : Microcontrôleur ATmega328	17
Figure II.3:Interface IDE Arduino.....	20
Figure II.4: Structure générale du programme (IDE	21
Figure II.5: Liste des différents shields.....	22
Figure II.6 : Capteur de Température et d'Humidité DHT11	23
Figure II.7: Câblage du capteur d'humidité d'aire sur l'Arduino UNO.....	24
Figure II.8 : Capteur de Luminosité TSL2561	24
Figure II.9 : Câblage du capteur Luminosité sur l'Arduino UNO	25
Figure II.10 : Capteur de vibration SW-18010P	26
Figure II.11: Câblage du capteur Vibration sur Arduino UNO.....	27
Figure II.12 : Capteur d'humidité du sol YL69	27
Figure II.13:Câblage du capteur d'humidité du sol sur l'Arduino UNO	29
Figure II.14 : Capteur de pluie	29
Figure II.15 :Câblage du capteur de pluie sur l'Arduino UNO	30
Figure II.16 : Capteur de pression BMP 280.....	31
Figure II.17:Câblage du capteur de Pression sur l'Arduino UNO	32
Figure II.18: Module Bluetooth HC-05	32
Figure II.19 : Câblage du module HC-05 sur l'Arduino UNO.....	33
Figure II.20: Interface Bluetooth terminal HC05	34
Figure III.1 : Schéma synoptique de la station mobile.....	37
Figure III.2 : Schéma de connexion de l'Arduino avec les différents capteurs sur Fritzing	38
Figure III.3: Organigramme des étapes d'exécution sur le microcontrôleur	39
Figure III.4 : Image du montage réalisé connecté par USB	40
Figure III.5: Résultats affichés sur le moniteur du PC	41
Figure III.6: Interfaçage Arduino/Smartphone.....	42

Figure III.7: Organigramme de l'interfaçage Arduino Smartphone.....	43
Figure III.8 : Interface de l'application MIT	44
Figure III.9: Etapes de la programmation du bloc 1	45
Figure III.10: Etapes de la programmation du bloc 2	46
Figure III.11: Etape de la programmation du bloc 3	47
Figure III.12: Courbe graphique du changement de la température.....	47
Figure III.13: Courbe graphique du changement de l'humidité.....	48
Figure III.14 : Courbe graphique du changement de luminosité.....	48
Figure III.15 : Courbe graphique du changement de la Pression	49

Liste des abréviations :

HPA : hecto pascal

OMM : Organisation Météorologique Mondiale

ARM : Adjustable Rate Mortgage.

IDE : Interactive Development Environment.

AVR: Automatic Volume Recognition.

USB: Universal Serial Bus.

I2C : protocole de communication.

SPP: Serial Port Protocol.

MIT: Massachusetts Institute of Technologie

UHF : Ultra High Frequency.