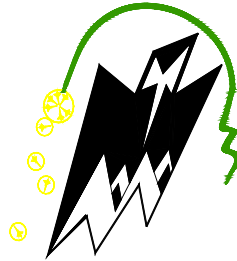


**République Algérienne Démocratique et Populaire**

**Ministère de L'Enseignement Supérieur et de la**

**Recherche Scientifique**

**Université Mouloud Mammeri de Tizi-Ouzou**



**FACULTE DE GENIE ELECTRIQUE ET  
D'INFORMATIQUE**

**Spécialité : Electronique industrielle**

**Projet de fin d'étude :**

**Réalisation d'un datalogger à base d'une carte  
Arduino Uno "application à une station  
météorologique"**

***Présenté par :***

*Tissoula Katia*

*Oulmi Lamia*

***Encadré par :***

*K. BENNAMANE*

***Promotion2023/2024***



## Sommaire

1.	Introduction :	3
2.	Définition :	3
3.	Les différentes technologies des dataloggers :	3
3.1.	Par caractéristiques techniques ou objectifs de mesure :	4
3.2.	Par technologie :	5
4.	Les avantages des dataloggers dans divers secteurs :	5
5.	Conclusion :	6
1.	Introduction :	8
2.	L'Internet des Objets (IoT) et Arduino :	8
2.1.	Avantages de la combinaison Arduino et IoT:	8
2.1.1.	Automatisation de la Maison	8
2.2.	Systèmes de Surveillance Intelligents	9
2.2.1.	Sécurité Renforcée :	9
2.2.2.	Surveillance en Temps Réel :	9
2.3.	Analyse d'Images	9
3.	La carte Arduino UNO :	9
3.1.	Microcontrôleur ATMEL ATmega328 :	10
3.1.1.	Mémoire Programme (FLASH) :	11
3.1.2.	Mémoire de Données Volatile (SRAM) :	11
3.1.3.	Mémoire de Données Non Volatile (EEPROM) :	11
3.2.	Les entrées les sorties de carte Arduino UNO :	11
4.	Sketch Arduino :	12
4.1.	Définition :	12
4.2.	La structure de base d'un Sketch (setup () et loop()) :	12
4.3.	Les Principaux Éléments d'un Sketch :	13
4.3.1.	Déclaration des Fonctions :	13
4.3.2.	Utilisation des Fonctions :	13
4.3.3.	Structure du Programme :	13
4.3.4.	Variables et Constantes :	13
4.3.5.	Instructions d'entrée/sortie (E/S):	13
4.3.6.	Commentaires :	13
5.	L'IDE Arduino :	14
5.1.	Présenter l'interface de l'IDE Arduino :	14

5.1.1.	Disposition des Fonctionnalités :	14
5.1.2.	Éditeur de Code :	14
5.1.3.	Barre de Menu et Barre d'Outils :	14
5.1.4.	Zone de Messages :	14
5.1.5.	Autres Fonctionnalités :	14
6.	Guide pratique pour débiter avec l'IDE Arduino :	15
6.1.	Création d'un nouveau Sketch :	15
6.2.	Importation de bibliothèques :	15
7.	Conclusion :	15
1.	Introduction :	16
2.	Logging de la température, de l'intensité lumineuse, de l'humidité et de la pression :	17
2.1.	Mesure de la température :	17
2.1.1.	LM35DZ :	17
2.1.2.	Principes de fonctionnement :	17
2.1.3.	Connexion des capteurs de température à la carte Arduino :	17
	Vcc est connecté à l'alimentation 5V de l'Arduino	17
2.1.4.	Affichage des mesures sur le moniteur série :	18
2.2.	Mesure d'intensité lumineuse :	18
2.2.1.	La photorésistance :	18
2.2.2.	Principe de fonctionnement :	18
2.2.3.	Connexion des capteurs de luminosité à la carte Arduino :	19
2.2.4.	Affichage des mesures sur le moniteur série :	19
2.3.	Mesure de l'humidité :	19
2.3.1.	Capteur d'humidité DHT11 :	19
2.3.2.	Principes de fonctionnement du capteur d'humidité :	20
2.3.3.	Connexion des capteurs d'humidité à la carte Arduino :	20
2.3.4.	Affichage des mesures sur le moniteur série :	20
2.4.	Mesure de pression :	21
2.4.1.	Capteur de pression MPX2010DP :	21
2.4.2.	Principes de fonctionnement du capteur de pression :	21
2.4.3.	Connexion des capteurs d'humidité à la carte Arduino :	21
2.4.4.	Affichage des mesures sur le moniteur série :	22
3.	Intégrer un module d'horloge temps réel (RTC) au datalogger :	22
3.1.	Module d'horloge temps réel (RTC) :	22
3.2.	Principes de fonctionnement :	23
3.3.	Connexion des capteurs d'humidité à la carte Arduino :	23

4.	Enregistrement des données d'une station météo et sauvegarde sur carte SD : .....	24
4.1.	Un adaptateur de carte micro SD : .....	24
4.2.	Connexion d'un adaptateur de carte micro SD à la carte Arduino : .....	24
5.	TESTS : .....	25
5.1.	Schéma de montage final : .....	25
5.2.	Exécution du programme principal : .....	26
5.3.	Les mesures dans le moniteur série : .....	27
5.4.	Vérification des données enregistrées : .....	28
6.	Conclusion : .....	31
➤	Conclusion générale .....	33

## ***Remerciements :***

*Nous tenons à remercier tout d'abord Dieu le tout puissant de nous avoir donné le courage et la santé durant toutes nos années d'étude.*

*Nous tenon à remercier notre encadreur, Pr. Bennamane Kamal pour ses encouragements et son aide précieuse.*

*Nous présentons également nos sincères remerciements au président et aux membres de jury pour l'intérêt qu'ils ont porté à ce travail en acceptant de le juger.*

*Nos vifs remerciements vont, en fin, à tous nos enseignants du département de Génie électrique et informatique qui ont veillé sur notre formation et qui ont tenu à faire accroître nos connaissances.*

## *Dédicace :*

*Je dédie ce mémoire A mes parents qui m'ont soutenu et encouragé durant ces années d'études.*

*Qu'elle trouve ici le témoignage de ma profonde reconnaissance.*

*A mes frères, et mes sœurs et Ceux qui ont partagé avec moi tous les*

*Moments d'émotion lors de la réalisation de ce travail. Ils m'ont*

*Chaleureusement supporté et encouragé tout au long de mon parcours.*

*A ma famille, mes proches et à ceux qui me donnent de l'amour et de la*

*Vivacité.*

*A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de*

*Succès.*

*A tous ceux que j'aime.*

*Merci !*

**Résumé du Mémoire :**

Ce mémoire décrit la création d'une station météo autonome qui utilise des dataloggers et des systèmes embarqués pour mesurer la température, l'humidité, la pression atmosphérique et la luminosité. En s'appuyant sur la plateforme Arduino, le projet a abouti à un prototype fonctionnel et fiable. Réalisé en trois étapes étude des dataloggers, exploration de la technologie Arduino, et conception de la station le projet a démontré que des solutions abordables peuvent efficacement collecter et stocker des données. L'intégration d'un module RTC a ajouté une dimension temporelle aux données. Les tests ont validé la précision et la fiabilité du système, tout en enrichissant les compétences en électronique et en programmation.

**Les mots clés :**

Station météo, Datalogger, Stockage de données, Autonomie.

## ➤ **Introduction générale**

## **Introduction générale**

Dans le domaine de l'observation et de la surveillance des conditions météorologiques, deux concepts clés méritent d'être définis au préalable : le "datalogger" et les "systèmes embarqués". Le "datalogger" fait référence au processus d'acquisition, d'enregistrement et de stockage automatisé de données environnementales. Les "systèmes embarqués" désignent quant à eux des dispositifs électroniques conçus pour effectuer des tâches spécifiques de manière autonome, sans nécessiter une intervention humaine continue.

C'est dans ce contexte que s'inscrit la problématique centrale de notre projet de recherche : comment concevoir et réaliser une station météo autonome capable de mesurer, d'enregistrer et de stocker de manière fiable et durable les principaux paramètres météorologiques tels que la température, l'humidité, la pression atmosphérique et la luminosité ?

Cette question soulève des enjeux techniques liés à l'intégration des capteurs, à la gestion des données et à l'autonomie de fonctionnement du système, mais aussi des défis pratiques en termes d'accessibilité et d'utilisation de la solution développée.

L'originalité et l'intérêt de notre recherche résident dans l'exploitation des technologies de l'Internet des Objets (IoT) et de la plateforme Arduino pour concrétiser cette station météo autonome.

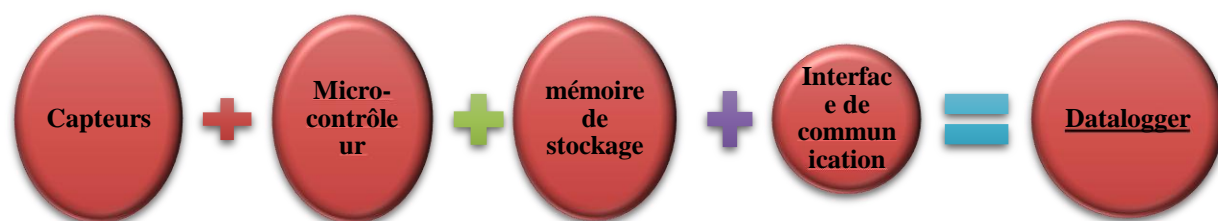
En effet, ces outils offrent des possibilités innovantes, permettant de concevoir des solutions abordables, flexibles et adaptées aux besoins d'une large gamme d'utilisateurs, qu'ils soient scientifiques, chercheurs ou simples particuliers intéressés par la météorologie. Les objectifs poursuivis par notre recherche sont multiples :

Il s'agit d'abord de concevoir et de réaliser un prototype fonctionnel de station météo autonome, capable de mesurer avec précision les paramètres clés tels que la température, l'humidité, la pression atmosphérique et la luminosité. Ensuite, nous visons à mettre en place un système fiable et durable de collecte, de stockage et de gestion des données environnementales recueillies, en exploitant notamment l'utilisation d'une carte SD. Enfin, nous souhaitons démontrer la pertinence et l'accessibilité de l'utilisation des technologies IoT et Arduino pour ce type d'application météorologique.

L'hypothèse centrale de notre recherche est que l'exploitation de la plateforme Arduino et des technologies IoT permettra de concevoir une station météo autonome, à la fois

## Introduction générale

performante, abordable et conviviale, facilitant ainsi l'observation, l'enregistrement et l'analyse des conditions météorologiques par un large public.



# **CHAPITRE I**

## **➤ Généralités sur les dataloggers**

# **CHAPITRE I Généralités sur les dataloggers**

## **1. Introduction :**

Il est essentiel de surveiller la température dans de multiples domaines, comme le contrôle de la qualité des produits alimentaires, la préservation des médicaments, la gestion des entrepôts, et même la surveillance de l'environnement dans le cadre de la recherche scientifique.

Autrefois mécanique, le datalogger est généralement autonome et peut être programmé pour enregistrer des valeurs individuelles sur des périodes allant de quelques heures à plusieurs mois.

En ce qui concerne la surveillance météorologique, la création et la mise en place d'un datalogger avec une carte Arduino représentent une solution souple, abordable et adaptable pour recueillir et enregistrer des informations météorologiques de manière précise.

## **2. Définition :**

Les dataloggers, ou enregistreurs de données, sont des dispositifs électroniques conçus pour collecter, stocker et parfois transmettre des informations sur des variables physiques ou environnementales au fil du temps. Ils peuvent mesurer une gamme étendue de paramètres tels que la température, l'humidité, la pression, la lumière et bien d'autres. Dotés de capteurs spécifiques, ces dispositifs sont souvent autonomes, alimentés par des piles, et programmables pour enregistrer les données à des intervalles définis.

Du point de vue technologique, en général, les dataloggers intègrent des capteurs sensibles aux paramètres à mesurer, un microcontrôleur pour le traitement des données, une mémoire de stockage et une interface de communication, si nécessaire. Certains modèles plus avancés offrent la possibilité de transférer les données via Bluetooth, Wi-Fi ou des cartes SIM, facilitant ainsi la surveillance à distance.

## **3. Les différentes technologies des dataloggers :**

Les dataloggers utilisent diverses technologies pour collecter, stocker, et transmettre des données. Leurs principaux composants sont :

### **a. Capteurs :**

Mesurent des paramètres comme la température, l'humidité, la pression, et la lumière, transformant ces variations en signaux électriques.

## **CHAPITRE I Généralités sur les dataloggers**

### b. Microcontrôleur :

Gère le traitement des données, contrôle le datalogger, déclenche les mesures, et gère le stockage des données.

### c. Mémoire de stockage :

Souvent une mémoire flash, conserve les données enregistrées sur une longue période.

### d. Connectivité :

Modules comme le Bluetooth, le Wi-Fi, ou des cartes SIM permettent de transférer les données vers des dispositifs externes ou des plateformes de surveillance, facilitant l'analyse à distance.

Les dataloggers se classent selon les paramètres, des combinaisons hybrides sont proposées pour des besoins spécifiques. Le choix dépend des besoins précis du projet :

### **3.1. Par caractéristiques techniques ou objectifs de mesure :**

#### a. Température :

Utilisés pour surveiller et contrôler les températures sensibles, souvent dans la chaîne d'approvisionnement.

#### b. Humidité :

Pour suivre les niveaux d'humidité relative dans des environnements où c'est crucial, comme les entrepôts ou les musées.

#### c. Pression :

Mesurent la pression de gaz ou de liquides, utile pour surveiller les variations de niveau d'eau, l'altitude, ou la vitesse.

#### d. Vibrations / Chocs :

Aussi appelés "shock" dataloggers, ils enregistrent les chocs et les vibrations, souvent utilisés dans le transport ou les équipements industriels.

#### e. Wi-Fi / Cellulaire :

Permettent de récupérer les données à distance et en temps réel, pratique pour la surveillance continue.

## **CHAPITRE I Généralités sur les dataloggers**

### f. USB :

Permettent de connecter directement le datalogger à un ordinateur pour télécharger les données après la période de surveillance.

### g. Utilisation unique vs rechargeable :

Selon que l'équipement doit être remplacé après une utilisation ou peut être rechargé et réutilisé.

## **3.2. Par technologie :**

### a. Électronique :

Comprend les capteurs, le microcontrôleur, et la mémoire de stockage. Nécessite souvent une configuration initiale via un ordinateur.

### b. Mécanique :

Imprime directement les données sur papier intégré dans l'appareil. Pratique pour les environnements où les ordinateurs ne sont pas disponibles.

### c. Sans fil / Radiofréquence (RFID) :

Permet une communication sans fil avec d'autres appareils, offrant une flexibilité supplémentaire pour la récupération des données.

Certaines entreprises proposent des dataloggers qui combinent ces technologies pour répondre à des besoins spécifiques. Le choix d'un datalogger doit être basé sur les exigences précises du projet. [1]

## **4. Les avantages des dataloggers dans divers secteurs :**

Les dataloggers ont beaucoup d'avantages qui les rendent très utiles dans différents domaines. Tout d'abord, ils peuvent collecter des données de façon continue et précise, sur une longue période. Cela permet d'analyser en détail des choses importantes comme la température, l'humidité et la pression. C'est essentiel dans des domaines comme la recherche, l'agriculture et l'industrie, où la qualité des données est cruciale.

Un autre gros point positif, c'est que les dataloggers peuvent fonctionner tout seuls, sans avoir besoin d'être contrôlés en permanence par quelqu'un. C'est très pratique dans des endroits difficiles d'accès ou quand il faut collecter régulièrement des données.

Les dataloggers ont aussi des connexions comme le Bluetooth, le Wi-Fi ou la 4G. Cela permet de les surveiller à distance en temps réel et d'être averti rapidement en cas de

## **CHAPITRE I Généralités sur les dataloggers**

problème. C'est indispensable pour garder la qualité des produits périssables et faire marcher correctement les équipements sensibles. De plus, les dataloggers ont une longue autonomie, peuvent être utilisés dans beaucoup de types d'environnements, et sont très précis dans leurs mesures. Leur interface est facile à utiliser, souvent sans avoir besoin de connaissances techniques poussées. Leur connectivité moderne facilite aussi le transfert des données.

Les dataloggers ont aussi une mémoire interne pour stocker les données sur le long terme. Ils peuvent même envoyer des alertes quand il y a des changements importants. Ils sont aussi relativement peu coûteux, ce qui les rend accessibles même avec des petits budgets.

Enfin, ils sont certifiés selon différentes normes, ce qui garantit leur fiabilité et leur conformité. Tous ces avantages font des dataloggers des outils idéaux pour de nombreuses applications dans l'industrie, la science et l'environnement. Ils sont bien plus performants et flexibles que les anciens enregistreurs de données. [2]

### **5. Conclusion :**

La surveillance précise et continue des variables environnementales est essentielle dans de nombreux domaines, de l'industrie à la recherche scientifique. Les dataloggers, avec leur capacité à collecter, stocker et transmettre des données sur des paramètres tels que la température, l'humidité et la pression, représentent pour nous une solution inestimable. Leur autonomie, leur précision, leur connectivité moderne et leur facilité d'utilisation en font des outils polyvalents et indispensables à nos yeux.

En offrant une surveillance fiable et une analyse détaillée des conditions environnementales, les dataloggers contribuent à garantir la qualité des produits, à optimiser les processus industriels et à soutenir la recherche scientifique. Nous estimons que l'adoption de ces technologies modernes marque une avancée significative par rapport aux méthodes traditionnelles d'enregistrement de données, ouvrant la voie à une surveillance plus efficace et à une prise de décision éclairée dans une multitude de secteurs.

# **CHAPITRE II**

## **Arduino : Applications et Technologies**

**CHAPITRE I Généralités sur les dataloggers**

## **1. Introduction :**

L'Internet des Objets (IoT) permet aux objets du quotidien de devenir intelligents et interconnectés grâce à des capteurs et des capacités de communication. Arduino, une plateforme de développement accessible, est largement utilisée dans les projets IoT. En combinant Arduino et l'IoT, il devient possible de connecter le monde physique au monde virtuel. Arduino collecte des données environnementales via des capteurs et les transmet au réseau.

## **2. L'Internet des Objets (IoT) et Arduino :**

Arduino et l'IoT Arduino est une plateforme matérielle open source très populaire pour les projets IoT. Elle est appréciée pour sa simplicité et sa polyvalence. Grâce à Arduino, il est possible de créer facilement des projets électroniques. L'IoT consiste à connecter des objets du quotidien au réseau pour qu'ils puissent communiquer entre eux et avec les utilisateurs. La synergie entre Arduino et l'IoT a transformé notre manière d'interagir avec notre environnement. Arduino agit comme une interface entre les appareils physiques et le réseau. Il collecte des données de l'environnement à l'aide de capteurs, comme la température, l'humidité, le mouvement, etc. Ces données sont ensuite traitées et envoyées vers le cloud via des connexions sans fil pour être stockées et accessibles à tout moment.

### **2.1. Avantages de la combinaison Arduino et IoT:**

La combinaison d'Arduino et de l'Internet des Objets (IoT) offre de nombreux avantages, tant sur le plan personnel que professionnel. Voici un aperçu détaillé de ces avantages :

#### **2.1.1. Automatisation de la Maison**

##### **2.1.1.1. Contrôle à Distance :**

Grâce à Arduino, il est possible de contrôler divers appareils domestiques, comme les lumières, les thermostats, et les serrures, à distance via une application sur votre Smartphone. Cela permet non seulement de gagner du temps mais aussi de réaliser des économies d'énergie.

### **2.1.1.2. Scénarios Personnalisés :**

Vous pouvez programmer des scénarios personnalisés, comme allumer les lumières au coucher du soleil ou ajuster la température de votre maison en fonction de votre emploi du temps.

## **2.2. Systèmes de Surveillance Intelligents**

### **2.2.1. Sécurité Renforcée :**

En utilisant Arduino, vous pouvez créer des systèmes de sécurité plus efficaces, incluant des alarmes, des caméras de surveillance, et des détecteurs de mouvement. Ces systèmes peuvent vous envoyer des alertes instantanées en cas d'activité suspecte.

### **2.2.2. Surveillance en Temps Réel :**

Les données recueillies par les capteurs peuvent être analysées en temps réel, permettant de réagir rapidement à toute anomalie ou intrusion.

## **2.3. Analyse d'Images**

### **2.3.1. Détection de Situations à Risque :**

Les caméras connectées à un Arduino peuvent capturer et analyser des images pour détecter des situations à risque, comme la présence d'intrus ou des mouvements inhabituels.

### **2.3.2. Reconnaissance de Visages et Objets :**

Grâce à des algorithmes d'apprentissage automatique, les systèmes peuvent reconnaître des visages ou des objets spécifiques, améliorant ainsi la sécurité et l'efficacité des systèmes de surveillance.

### **2.3.3. Applications Médicales :**

Dans les hôpitaux, l'analyse d'images peut surveiller les patients et détecter des situations d'urgence, comme une chute, permettant une intervention rapide. [3]

## **3. La carte Arduino UNO :**

L'Arduino Uno est une carte à microcontrôleur basée sur un ATmega328 cadencé à 16 MHz. Elle est conçue pour être simple et économique, et est la carte la plus utilisée et la mieux documentée de la famille Arduino. [4]

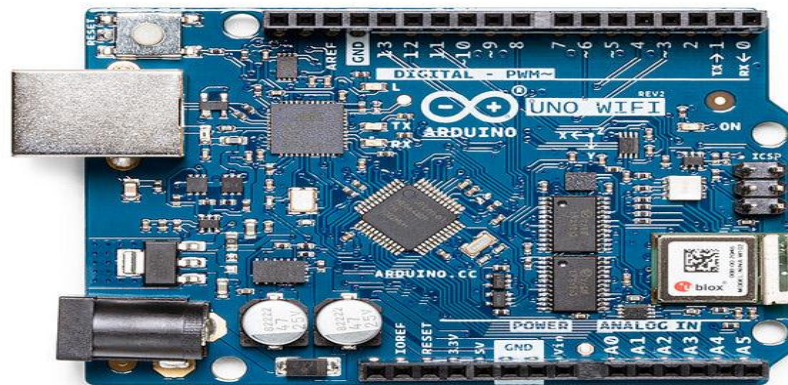
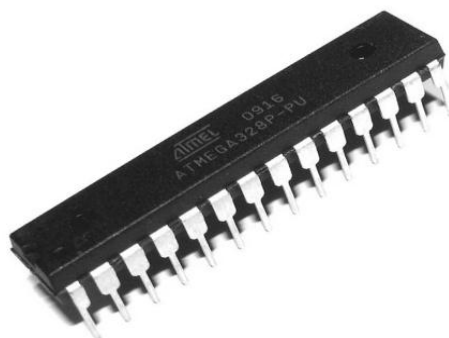


Figure 2.1 : carte Arduino Uno

### 3.1. Microcontrôleur ATMEL ATmega328 :

L'ATmega328p est un type de microcontrôleur fabriqué par Atmel et faisant partie de la famille AVR. Il est très avancé en termes d'architecture et de spécifications techniques. Grâce à sa capacité à fonctionner sur une plage de tensions variées et à sa mémoire importante, ainsi que ses performances élevées et ses nombreuses fonctionnalités intégrées, il est extrêmement utile pour les applications dans les systèmes embarqués et les microcontrôleurs. Sa popularité dans les projets Arduino Uno est due à la facilité de programmation en langage C/C++ et à la présence d'un bootloader, ce qui en fait un choix privilégié pour les développeurs électroniques.

## ATMEGA328P Microcontroller



(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

**Figure2 .2 : microcontrôleur ATMEG328P**

La mémoire d'un microcontrôleur de l'ATmega328 se compose de trois types principaux :

### **3.1.1. Mémoire Programme (FLASH) :**

La mémoire FLASH est une mémoire non volatile qui conserve les données même après la coupure de l'alimentation. Elle est utilisée pour stocker le programme principal du microcontrôleur. La mémoire FLASH est rapide et offre une capacité de stockage importante, permettant au microcontrôleur d'exécuter des instructions et des programmes.

### **3.1.2. Mémoire de Données Volatile (SRAM) :**

La SRAM (StaticRandom Access Memory) est une mémoire volatile qui perd ses données lorsque l'alimentation est coupée. Elle sert à stocker des données temporaires en cours d'exécution du programme, telles que les variables. La SRAM est rapide mais plus coûteuse que d'autres types de mémoires.

### **3.1.3. Mémoire de Données Non Volatile (EEPROM) :**

L'EEPROM (ElectricallyErasable Programmable Read-Only Memory) est une mémoire non volatile utilisée pour stocker des données persistantes qui doivent être conservées même en cas de coupure d'alimentation. Bien que plus lente que la SRAM, l'EEPROM offre une capacité de stockage suffisante pour des données telles que des paramètres de configuration ou des états à conserver durablement.

Ces trois types de mémoires jouent des rôles essentiels dans le fonctionnement et le stockage des informations nécessaires au bon déroulement des programmes et des opérations du microcontrôleur ATmega328. [5]

## **3.2. Les entrées les sorties de carte Arduino UNO :**

La carte Arduino UNO possède différents types de broches d'entrée et de sortie, permettant de se connecter à divers composants électroniques.

Les broches numériques (D0 à D13) fonctionnent comme des interrupteurs. Elles peuvent être soit en position "haute" (5 volts), soit en position "basse" (0 volt). Cela permet de commander des choses simples comme des LED ou des moteurs. Certaines de ces broches numériques (D3, D5, D6, D9, D10, D11) sont en réalité des broches "PWM" (Pulse Width

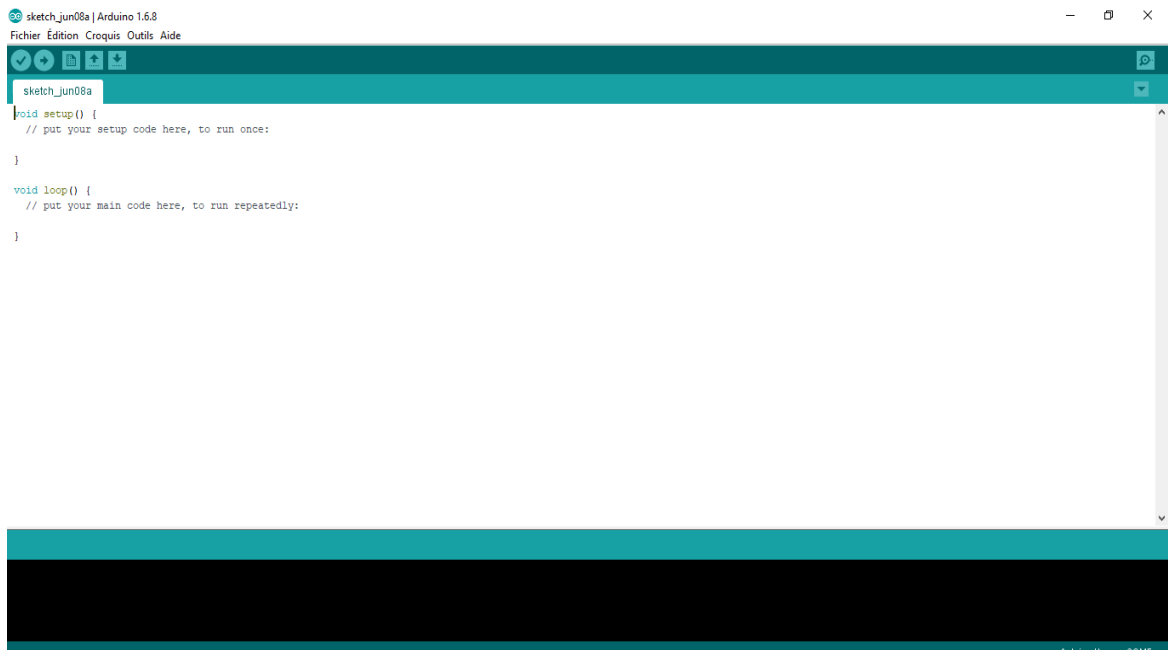
Modulation). Cela signifie qu'elles peuvent varier la vitesse ou la luminosité des composants qui y sont connectés, en modulant le signal électrique.

Les broches analogiques (A0 à A5) peuvent mesurer des signaux électriques qui varient de manière continue, comme la luminosité, la température ou la tension. Cela permet de récupérer des informations plus complexes de l'environnement.

## 4. Sketch Arduino :

### 4.1. Définition :

Un « sketch » dans le contexte Arduino est le nom que l'Arduino utilise pour un programme. Il s'agit d'un fichier écrit en utilisant le langage Processing, qui est proche du C et permet de manipuler les entrées/sorties, les communications avec le PC, les opérations mathématiques et les conversions de type.



**Figure 2.3 :** L'interface de programme Arduino sketch

### 4.2. La structure de base d'un Sketch (setup () et loop()) :

La structure de base d'un sketch en Arduino comprend deux fonctions principales : setup () et loop().

- setup() : Celle-ci s'exécute une seule fois, au tout début, quand on démarre le programme. C'est ici qu'on initialise tous les paramètres, qu'on configure les broches

d'entrée/sortie, qu'on démarre les différentes bibliothèques, etc. C'est comme la préparation avant de lancer le programme.

- `loop ()` : Une fois que la fonction `setup()` est exécutée, la fonction `loop()` est exécutée en boucle indéfiniment. C'est dans cette fonction que vous placez le code principal de votre programme, les instructions que vous souhaitez exécuter en continu.

En combinant ces deux fonctions, on peut faire fonctionner l'Arduino de manière efficace. On initialise tout ce qu'il faut au début, dans "`setup ()`", puis on exécute les instructions en boucle dans "`loop()`" pour que le programme tourne sans s'arrêter. [8]

### **4.3. Les Principaux Éléments d'un Sketch :**

#### **4.3.1. Déclaration des Fonctions :**

Les fonctions sont des blocs de code qui effectuent des tâches spécifiques. Elles doivent être déclarées avec un nom clair, sans espaces ni accents, suivies éventuellement d'arguments entre parenthèses, et d'une séquence d'instructions entre des accolades { }.

#### **4.3.2. Utilisation des Fonctions :**

Une fois déclarées, les fonctions peuvent être appelées à plusieurs reprises dans le code principal en fournissant éventuellement des paramètres d'entrée.

#### **4.3.3. Structure du Programme :**

Un sketch Arduino doit obligatoirement contenir les fonctions `setup ()` et `loop ()`. La fonction `setup ()` est exécutée en premier, suivie de la fonction `loop ()`.

#### **4.3.4. Variables et Constantes :**

En dehors des fonctions, on peut déclarer des variables (dont le contenu peut changer) et des constantes (dont la valeur reste fixe) pour stocker des données ou des paramètres.

#### **4.3.5. Instructions d'entrée/sortie (E/S):**

Ces instructions permettent de lire des données depuis des capteurs ou de contrôler des actionneurs.

#### **4.3.6. Commentaires :**

Ils aident à expliquer le code et à le rendre plus compréhensible pour les autres programmeurs.

Ces éléments sont essentiels pour écrire des sketches Arduino bien structurés et fonctionnels. Ils permettent de créer des programmes clairs, modulaires et faciles à maintenir.

## **5. L'IDE Arduino :**

L'IDE Arduino (Integrated Développement Environnement) est un logiciel utilisé pour programmer les cartes Arduino. C'est un environnement de développement open-source qui offre une interface conviviale pour écrire, compiler et téléverser du code sur les microcontrôleurs Arduino.

### **5.1. Présenter l'interface de l'IDE Arduino :**

L'interface de l'IDE Arduino est conçue pour être claire et intuitive, facilitant ainsi la navigation et l'utilisation pour les débutants et les utilisateurs expérimentés. Voici quelques-unes des caractéristiques clés de l'interface :

#### **5.1.1. Disposition des Fonctionnalités :**

L'IDE présente une disposition organisée des fonctionnalités principales, telles que la zone d'édition du code, la zone de compilation, la zone de téléversement, etc.

#### **5.1.2. Éditeur de Code :**

La zone principale de l'interface est réservée à l'édition du code. C'est là que vous écrivez votre programme en langage Arduino, également appelé croquis (sketch).

#### **5.1.3. Barre de Menu et Barre d'Outils :**

En haut de l'interface, vous trouverez la barre de menu et la barre d'outils, qui regroupent des commandes pour les actions telles que la création d'un nouveau croquis, l'ouverture et la sauvegarde de fichiers, la compilation et le téléversement du code, etc.

#### **5.1.4. Zone de Messages :**

En bas de l'interface, il y a une zone où les messages d'erreur, les avertissements et les notifications sont affichés. Cela permet de suivre le processus de compilation et de téléversement du code.

#### **5.1.5. Autres Fonctionnalités :**

L'IDE Arduino offre également des fonctionnalités supplémentaires telles que la gestion des bibliothèques, le moniteur série pour la communication avec la carte Arduino, et des outils de débogage.

Ces éléments contribuent à une expérience utilisateur claire et intuitive, rendant l'IDE Arduino accessible à tous les utilisateurs, qu'ils soient débutants ou expérimentés.

## **6. Guide pratique pour débiter avec l'IDE Arduino :**

### **6.1. Création d'un nouveau Sketch :**

- Ouvrez l'IDE Arduino sur votre ordinateur.
- Cliquez sur « Fichier » dans la barre de menu.
- Sélectionnez « Nouveau » pour créer un nouveau Sketch.
- Vous pouvez maintenant commencer à écrire votre code dans l'éditeur.

### **6.2. Importation de bibliothèques :**

- Pour importer une bibliothèque dans votre Sketch, allez dans « Croquis » dans la barre de menu.
- Sélectionnez « Inclure une bibliothèque » pour choisir parmi les bibliothèques disponibles ou pour ajouter une bibliothèque externe.
- Une fois la bibliothèque importée, vous pourrez l'utiliser dans votre code en ajoutant les instructions appropriées. [8]

## **7. Conclusion :**

Ce chapitre a exploré les concepts de l'Internet des Objets (IoT) et comment la plateforme Arduino peut concrétiser ces idées dans divers projets innovants.

Nous avons étudié la carte Arduino Uno, un microcontrôleur essentiel pour les applications IoT. Facile à utiliser, elle collecte des données via des capteurs et les transmet au cloud. Sa nature open-source la rend accessible tant aux débutants qu'aux experts.

L'IDE Arduino, un logiciel gratuit, facilite la programmation de ces cartes. Son interface conviviale simplifie l'écriture, la compilation et le téléchargement du code.

En combinant Arduino et l'IoT, de nombreuses innovations sont possibles : automatisation domestique, surveillance intelligente, analyse d'images et création d'appareils interactifs.



## **Chapitre III**

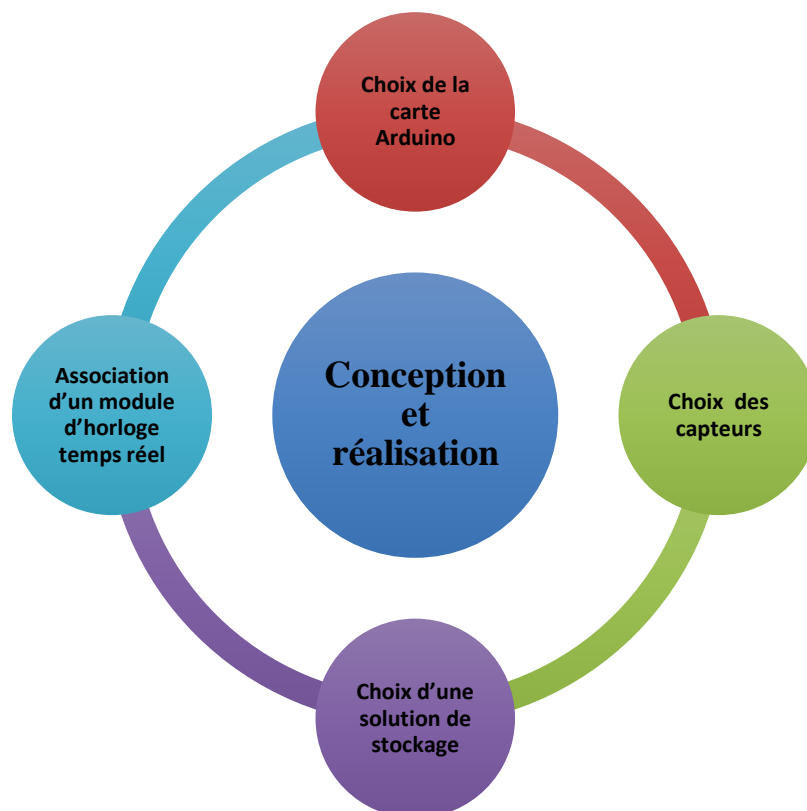
- **Conception et réalisation d'un datalogger pour une station météo**

## Chapitre III conception et réalisation d'un datalogger pour une station météo

### 1. Introduction :

Après avoir vu les bases théoriques des technologies liées à l'internet des objets (IoT) et aux dataloggers, nous allons maintenant nous concentrer sur la partie pratique de la mise en place de notre station météorologique. Tout d'abord, nous allons nous intéresser aux différents capteurs qui vont nous permettre de mesurer les paramètres météorologiques clés. Nous verrons en détail les capteurs de température, d'humidité, de pression atmosphérique et d'intensité lumineuse.

Nous apprendrons aussi comment les connecter à la plateforme Arduino UNO pour récupérer les données, nous consacrerons à la conception et à la réalisation du datalogger en lui-même, ce système, centré sur l'Arduino UNO, aura pour rôle de collecter avec précision les données des capteurs, de les horodater correctement, puis de les enregistrer sur une carte SD pour pouvoir les analyser par la suite. Tout au long de ces deux chapitres, nous mettrons en pratique nos connaissances sur les technologies IoT et les dataloggers afin de construire une station météorologique complète et fonctionnelle.





## Chapitre III conception et réalisation d'un datalogger pour une station météo

Gnd est connecté à la masse de l'Arduino.

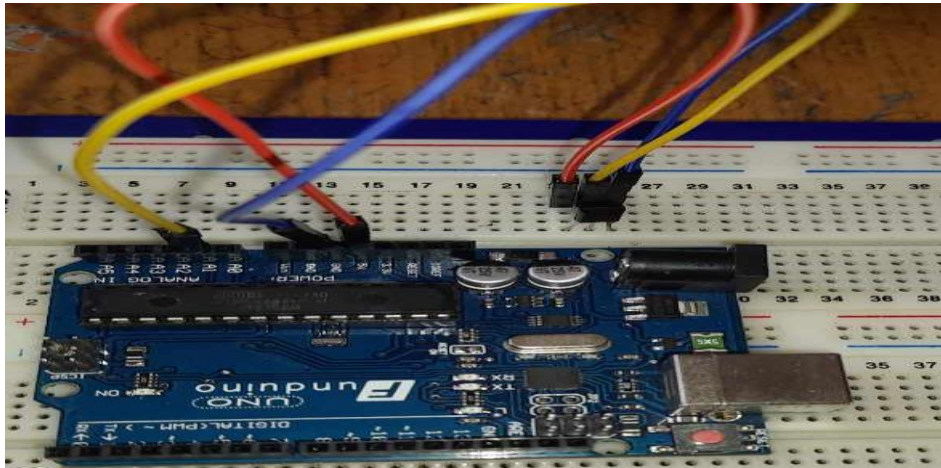


Figure 3.2 : Schéma de branchement du capteur de température

### 2.1.4. Affichage des mesures sur le moniteur série :



Figure 3.3 : Résultats des mesures de la température

## 2.2. Mesure d'intensité lumineuse :

### 2.2.1. La photorésistance :

La photorésistance, également connue sous le nom de LDR (Light-Dependent Resistor), est un composant électronique dont la résistance varie en fonction de la luminosité perçue. Elle est utilisée pour détecter la présence ou l'absence de lumière, ainsi que pour mesurer l'intensité de la lumière incidente. [9]



Figure 3.4 : La photorésistance

### 2.2.2. Principe de fonctionnement :

La photorésistance est un composant optoélectronique qui se comporte comme un transistor dont le courant de base est contrôlé par la lumière...Lorsqu'il est exposé à la

## Chapitre III conception et réalisation d'un datalogger pour une station météo

lumière, la photorésistance voit son courant collecteur-émetteur augmenter de manière proportionnelle à l'intensité lumineuse. Cette variation de courant peut être mesurée et convertie en une valeur d'intensité lumineuse. [9]

### 2.2.3. Connexion des capteurs de luminosité à la carte Arduino :

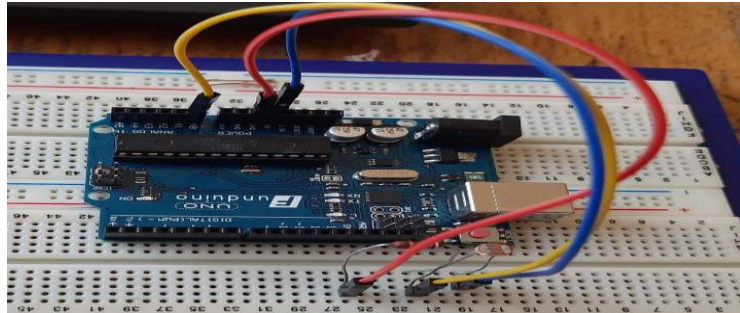


Figure 3.5 : Schéma de branchement du capteur de luminosité

### 2.2.4. Affichage des mesures sur le moniteur série :

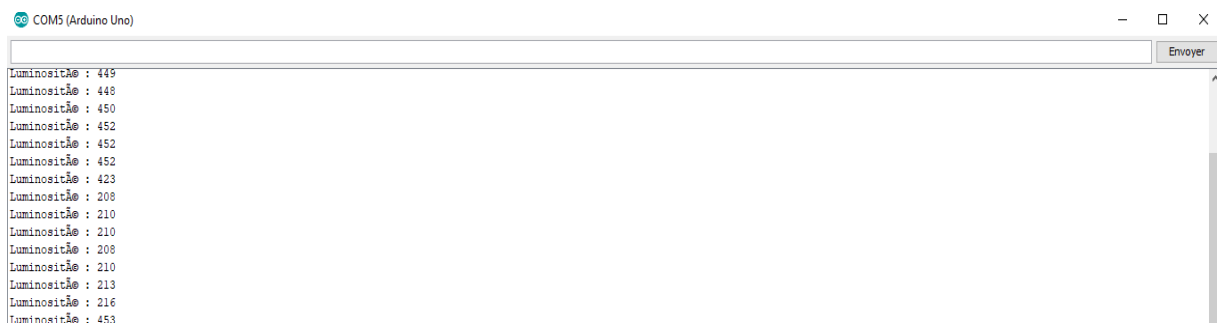


Figure 3.6 : Résultats des mesures de capteur de luminosité

## 2.3. Mesure de l'humidité :

### 2.3.1. Capteur d'humidité DHT11 :

Le DHT11 est un capteur de température et d'humidité à semi-conducteur. Il utilise un capteur capacitif pour mesurer l'humidité et une thermistance pour mesurer la température. [9]

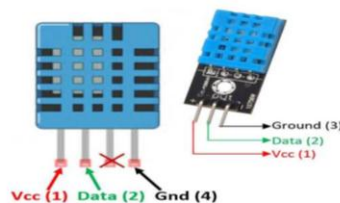


Figure 3.7 : Capteur d'humidité DHT11

## Chapitre III conception et réalisation d'un datalogger pour une station météo

### 2.3.2. Principes de fonctionnement du capteur d'humidité :

Le capteur d'humidité du DHT11 est composé d'un polymère hygroscopique (sensible à l'humidité) déposé entre deux électrodes. Lorsque l'humidité de l'air ambiant varie, les propriétés diélectriques du polymère changent, modifiant la capacité électrique du capteur. Un circuit électronique intégré au capteur convertit cette variation de capacité en un signal numérique représentant le taux d'humidité relative. [9]

### 2.3.3. Connexion des capteurs d'humidité à la carte Arduino :

Broche VCC du DHT11 relié à Broche 5V de l'Arduino

Broche DATA du DHT11 relié à une broche numérique de l'Arduino (branche 2)

Broche GND du DHT11 relié à Broche GND de l'Arduino

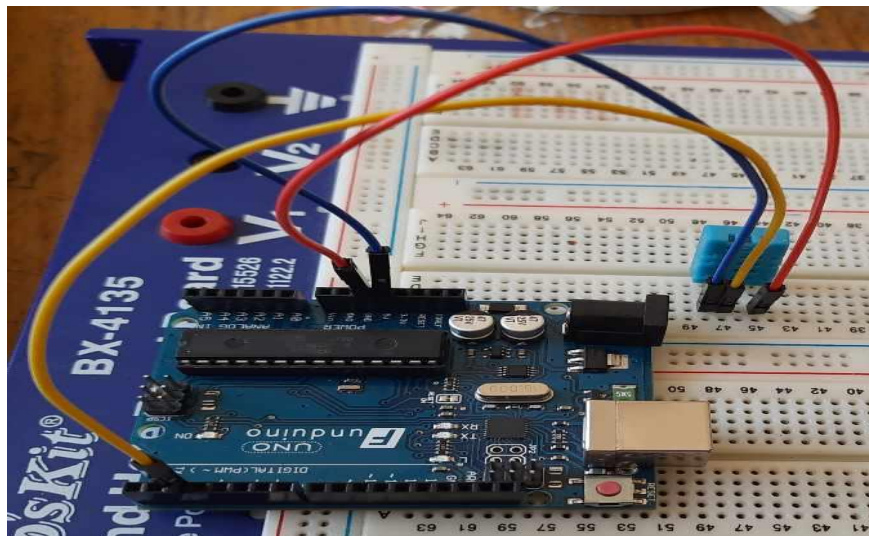


Figure 3.8 : Schéma de branchement du capteur d'humidité dht11

### 2.3.4. Affichage des mesures sur le moniteur série :

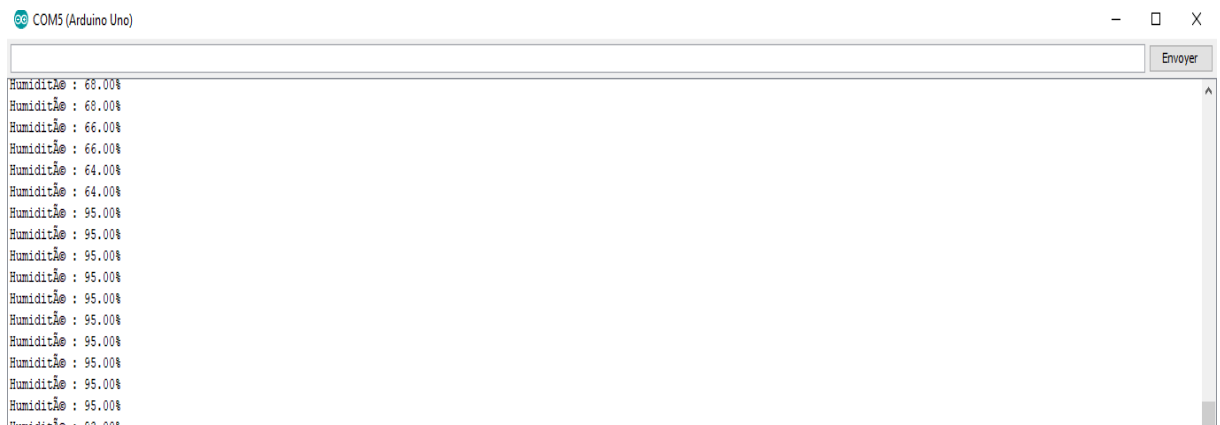
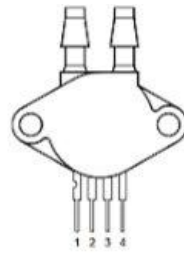


Figure 3.9 : Résultats de mesure de capteur d'humidité dht11

### 2.4. Mesure de pression :

#### 2.4.1. Capteur de pression MPX2010DP :

Le MPX2010DP est un capteur de pression qui mesure la différence de pression entre deux points. Il est souvent utilisé pour surveiller la pression dans les fluides, les systèmes de ventilation, les équipements médicaux et divers instruments de mesure industriels. [9]



**Figure 3.10 :** Capteur de pression MPX2010DP

#### 2.4.2. Principes de fonctionnement du capteur de pression :

Le fonctionnement du MPX2010DP repose sur un élément piezorésistif en silicium, dont la résistance électrique varie en fonction de la pression appliquée. Cette variation est ensuite convertie en une tension de sortie proportionnelle à la pression mesurée. Le capteur intègre également des circuits de compensation de température pour garantir une bonne linéarité et précision sur toute sa plage de fonctionnement. Grâce à sa large plage de mesure, sa sortie analogique linéaire et ses performances élevées, le MPX2010DP est couramment utilisé dans diverses applications de mesure de pression, telles que le contrôle de pression dans les systèmes pneumatiques, la mesure de pression atmosphérique, la détection de fuites dans les réseaux, et le contrôle de pression dans les appareils électroménagers. [9]

#### 2.4.3. Connexion des capteurs de pression à la carte Arduino :

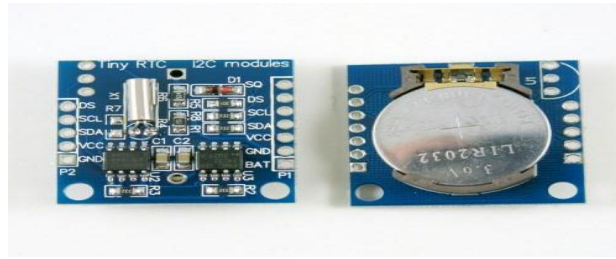
Le MPX2010DP possède 3 broches :

Vcc est connecté à l'alimentation 5V de l'Arduino

Vout est connecté à une entrée analogique (A2) de l'Arduino

Gnd est connecté à la masse de l'Arduino





**Figure 3.13:** Module d'horloge temps réel RTC\_DS130

### 3.2. Principes de fonctionnement :

Le DS1307 est un petit composant électronique très pratique. Il fait office d'horloge et de calendrier, fournissant des informations comme l'heure, la date et même l'année. Il peut même ajuster automatiquement la fin du mois, y compris pour les années bissextiles. Vous pouvez choisir entre un format d'heure de 12 ou 24 heures, avec un indicateur AM/PM pour le mode 12 heures. En cas de coupure de courant, il bascule automatiquement sur une alimentation de secours grâce à une pile bouton CR1220. De plus, il offre différentes options de sorties programmables pour une utilisation flexible dans divers projets électroniques.[10]

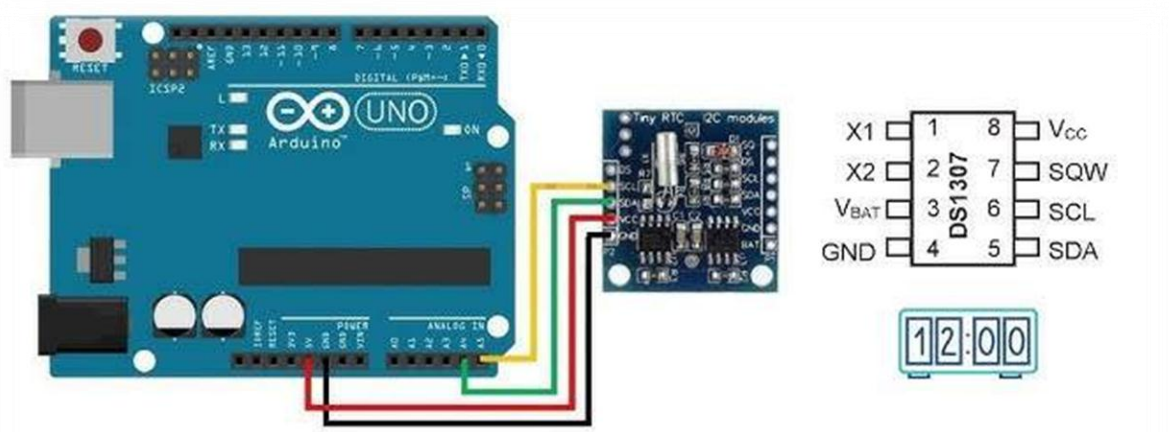
### 3.3. Connexion des capteurs d'humidité à la carte Arduino :

Broche VCC du RTC\_DS1307 ver 5V de l'Arduino

Broche GND du RTC\_DS1307 ver GND de l'Arduino2.

Broche SDA (données) du RTC\_DS1307 ver l'Arduino (A4)

Broche SCL (horloge) du RTC\_DS1307 ver l'Arduino (A5)



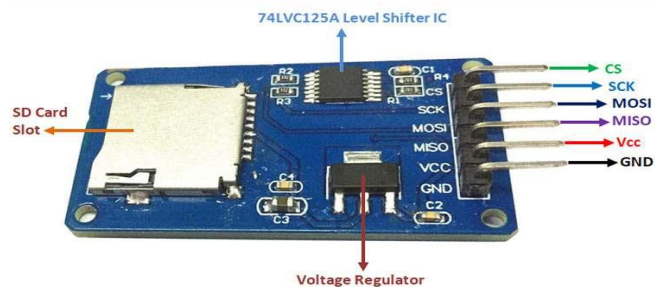
**Figure 3.14 :** Schéma de branchement d'horloge temps réel RTC\_DS1307

## Chapitre III conception et réalisation d'un datalogger pour une station météo

### 4. Enregistrement des données d'une station météo et sauvegarde sur carte SD :

#### 4.1. Un adaptateur de carte micro SD :

Un adaptateur de carte micro SD est un dispositif qui permet de convertir une carte micro SD en une carte SD standard. Cela permet d'utiliser la carte micro SD dans des appareils qui ne supportent que les cartes SD, mais pas les cartes micro SD. Les cartes micro SD sont spécialement conçues pour les téléphones portables et autres appareils mobiles, mais grâce à cet adaptateur, elles peuvent également être utilisées dans des appareils plus anciens ou ceux qui ne prennent en charge que les cartes SD standard. Ce module adaptateur est compact et pratique, offrant une solution flexible pour la compatibilité des cartes mémoire entre différents appareils. [11]



**Figure 3.15 :** Un adaptateur de carte micro SD

#### 4.2. Connexion d'un adaptateur de carte micro SD à la carte Arduino :

SD carte GND relie a Arduino GND

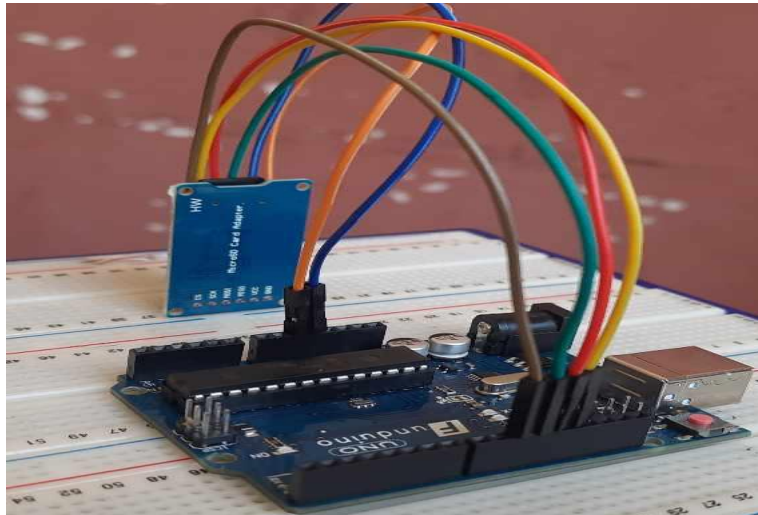
SD carte VCC relie a Arduino 5V

SD carte MISO relie a Arduino pin12

SD carte MOSI relie a Arduino pin13

SD carte SCK relie a Arduino pin10

SD carte SCS relie a Arduino 5V



**Figure 3.16 :** Schéma de branchement d'un adaptateur de carte micro SD

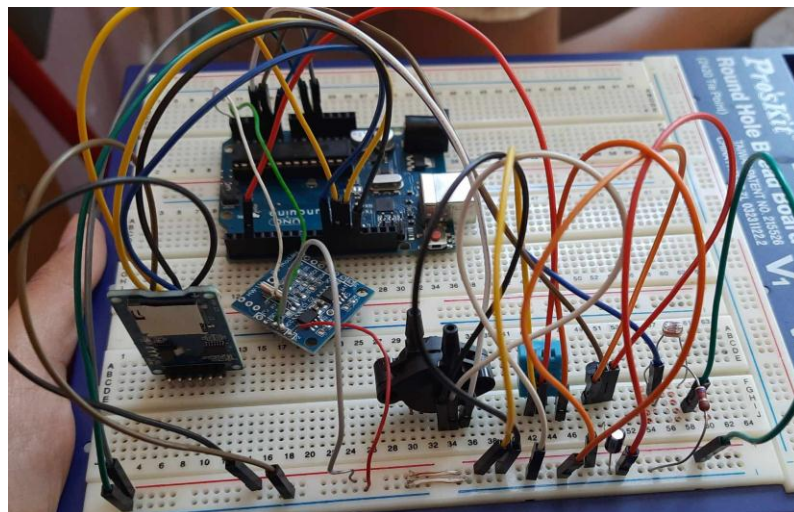
### 5. TESTS :

#### 5.1. Schéma de montage final :

Pour tester le fonctionnement complet de la station météo, nous avons suivi plusieurs étapes :

Tout d'abord, nous avons rassemblé tous les composants et les avons montés ensemble selon le schéma final du circuit.

Ensuite, nous avons chargé le programme principal dans l'Arduino. Ce programme permet de faire fonctionner tous les capteurs en même temps et d'enregistrer les données sur la carte micro SD.



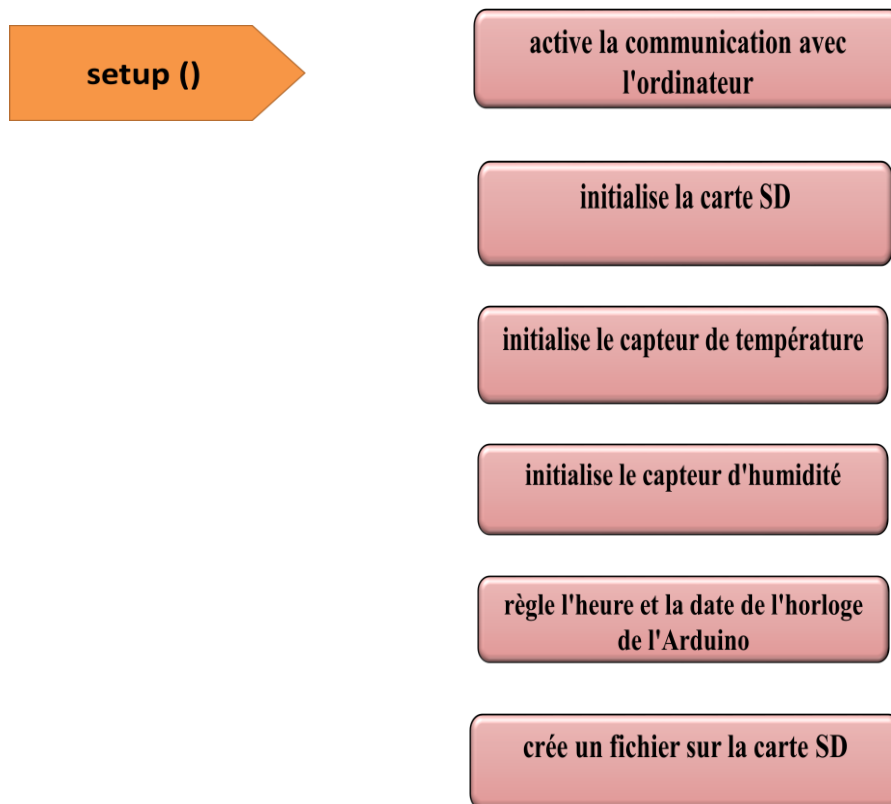
**Figure 3.17 :** Schéma de branchement station météo autonome

## Chapitre III conception et réalisation d'un datalogger pour une station météo

### 5.2. Exécution du programme principal :

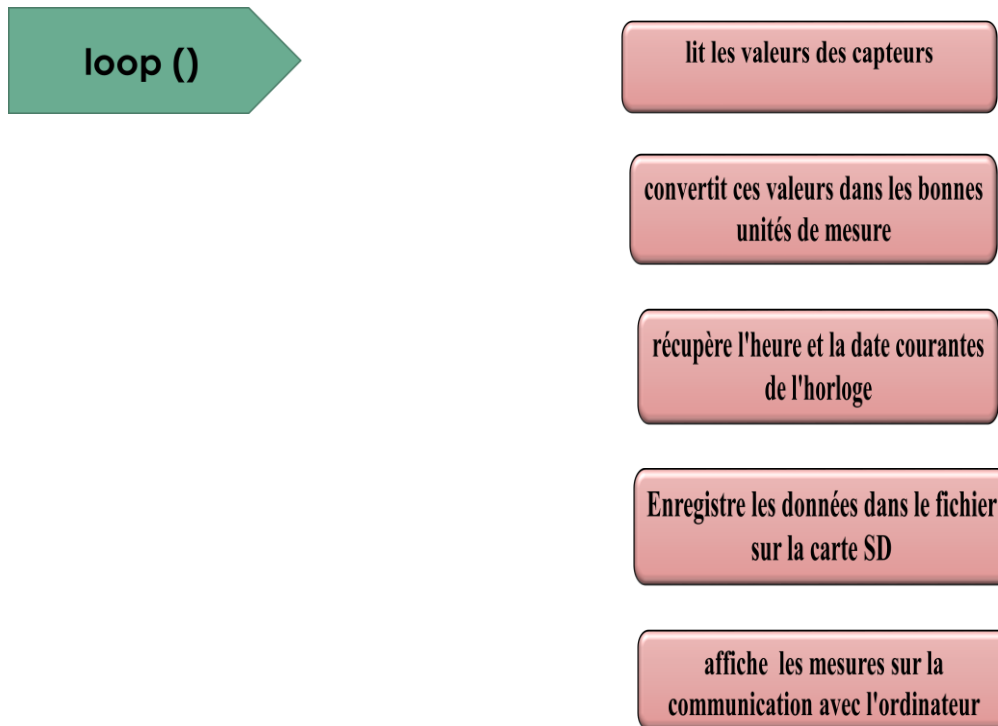
Le programme principal de la station météo commence par inclure les bibliothèques nécessaires pour communiquer avec la carte SD, le module RTC et le capteur de température, humidité, pression, luminosité.

Dans la première partie du programme, appelée "setup ()", le programme fait plusieurs choses, il active la communication avec l'ordinateur, pour pouvoir afficher des informations, il initialise la carte SD, pour pouvoir y enregistrer les mesures et il initialise le capteur de température et d'humidité et il règle l'heure et la date de l'horloge de l'Arduino et crée un fichier sur la carte SD pour y écrire les mesures.



### Exécution du programme principal de la station autonome

Dans la deuxième partie du programme, appelée "loop ()", le programme lit les valeurs des capteurs de luminosité, de température et de pression et il convertit ces valeurs dans les bonnes unités de mesure, il récupère l'heure et la date courantes de l'horloge et enregistre toutes ces données dans le fichier sur la carte SD puis il affiche les mesures sur la communication avec l'ordinateur. Tout ce processus de mesure et d'enregistrement se répète chaque un heure.



### Exécution du programme principal de la station autonome

#### 5.3. Les mesures dans le moniteur série :

Pendant que le programme fonctionnait, nous avons regardé attentivement les mesures de température, de luminosité, de pression, de l'humidité de date et d'heure qui s'affichaient dans le moniteur série de l'Arduino. Cela nous a permis de vérifier que chaque capteur fonctionnait bien et que les valeurs étaient logiques.

Voici les résultats obtenus de mesure de station météo dans le moniteur série :

```
COM5 (Arduino Uno)
Date : 2024/6/4
Luminosité : 424
Température : 26.88°C
Humidité : 58.00%
Pression : 1.13hPa
Heure : 11:43
Date : 2024/6/4
Luminosité : 424
Température : 26.88°C
Humidité : 58.00%
Pression : 1.13hPa
Heure : 11:43
Date : 2024/6/4
Luminosité : 424
Température : 26.88°C
Humidité : 58.00%
Pression : 1.13hPa
Heure : 11:43
Date : 2024/6/4
Luminosité : 424
Température : 26.88°C
Humidité : 58.00%
```

The screenshot shows the Arduino Serial Monitor window for COM5 (Arduino Uno). The output displays a repeating set of sensor data: Date (2024/6/4), Luminosité (424), Température (26.88°C), Humidité (58.00%), Pression (1.13hPa), and Heure (11:43). The data is printed in a consistent format across multiple lines.

**Figure 3.18 :** Résultats de mesure de station météo

### 5.4. Vérification des données enregistrées :

Nous avons retiré la carte micro SD et nous avons regardé le fichier de données enregistrées. Nous avons vu que les mesures étaient bien sauvegardées sur la carte. De plus, quand nous avons affiché ces mesures sous forme de tableau, elles correspondaient bien à ce que nous avons vu s'afficher dans le moniteur série de l'Arduino.

Ce tableau représente les valeurs enregistrées dans le fichier data.csv :

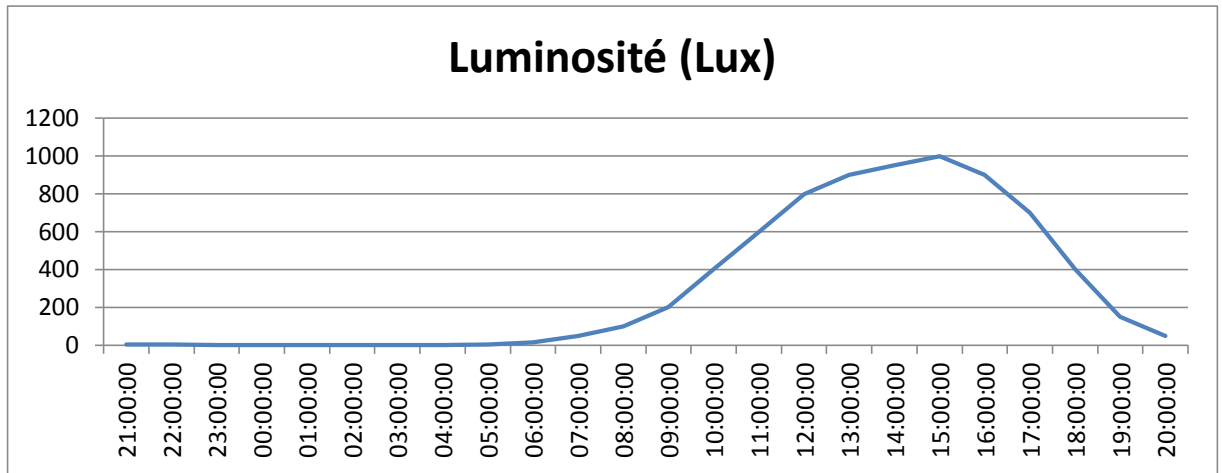
Date	Heure	Température (°C)	Luminosité (lux)	Humidité (%)	Pression (hPa)
14/06/2024	21:00:00	29,33	5	61	1012
15/06/2024	00:00:00	28,35	1	47	1010
15/06/2024	03:00:00	28,35	1	47	1010
15/06/2024	06:00:00	26,88	15	53	1010
15/06/2024	09:00:00	29,33	200	51	1014
15/06/2024	13:00:00	30,3	900	57	1011
15/06/2024	16:00:00	29,33	900	48	1009
15/06/2024	20:00:00	28,84	50	50	1010

**Figure 3.19 :** Tableau représente les valeurs enregistrées dans le fichier data.csv

Les données de température, de pression, d'humidité et de luminosité ont été collectées et enregistrées dans le fichier data.csv. Ces données ont ensuite été utilisées pour générer une série de courbes afin de visualiser l'évolution de ces différents paramètres au fil du temps.

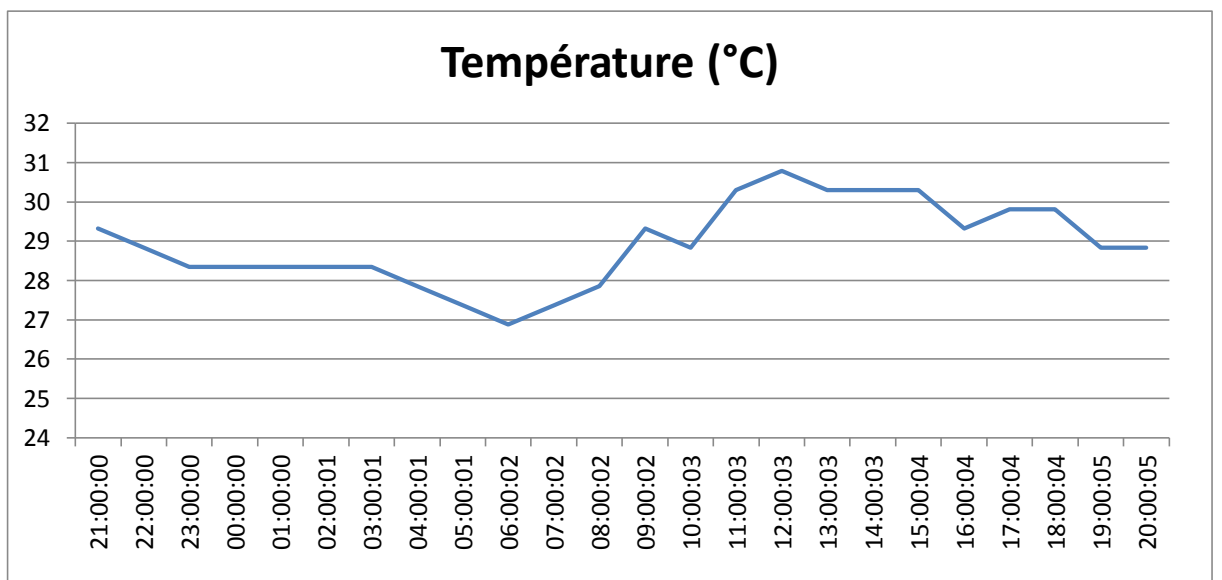
Ces courbes représentent les valeurs enregistrées dans le fichier data.csv :

La première courbe montre les variations de la luminosité ambiante. Cette courbe suit logiquement le rythme jour/nuit, avec des valeurs maximales pendant la journée et des valeurs minimales la nuit.



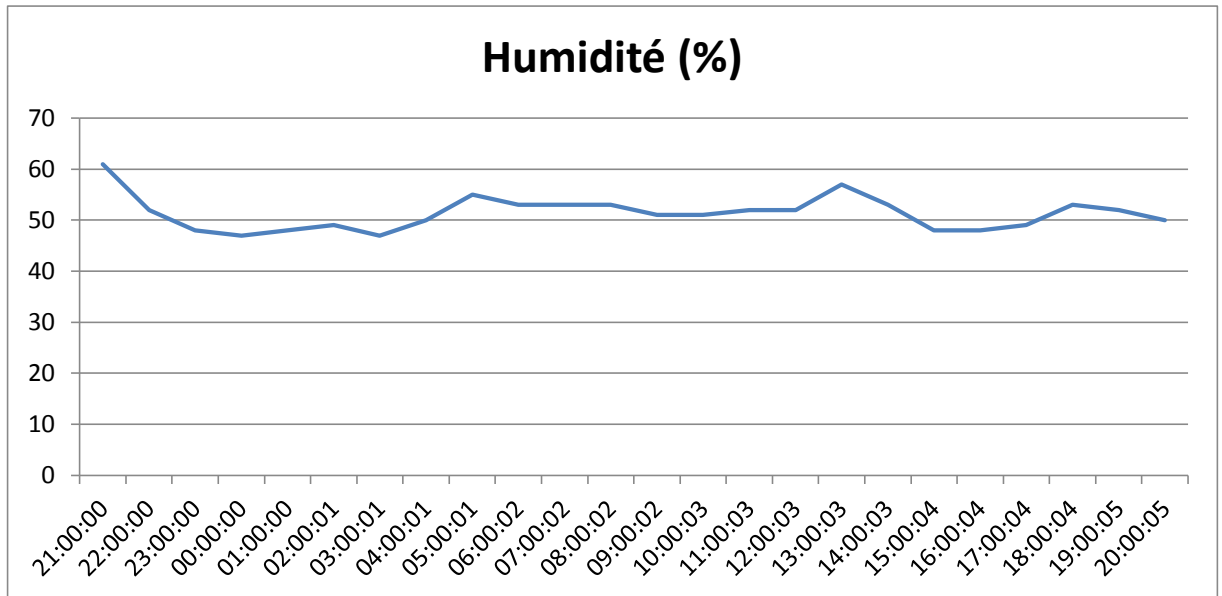
**Figure 3.20 :** les variations de la luminosité ambiante

La deuxième courbe montre la variation de la température au cours de la période de mesure. On peut observer des fluctuations régulières, avec des pics et des creux correspondant probablement aux cycles jour/nuit et aux changements de conditions météorologiques.



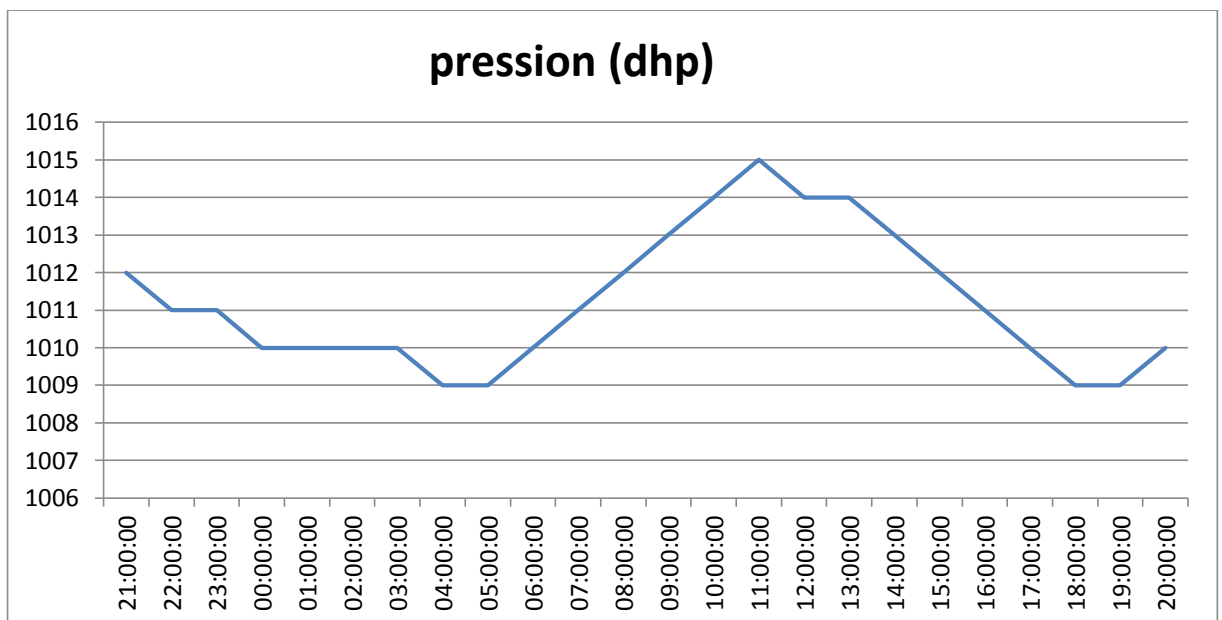
**Figure 3.21:** Les variations de température

La troisième courbe illustre les variations du taux d'humidité. On remarque une certaine corrélation entre les courbes de température et d'humidité, avec des pics d'humidité lorsque la température est plus basse.



**Figure 3.22 :** Les variations du taux d'humidité

Enfin, la quatrième courbe présente l'évolution de la pression atmosphérique. Celle-ci semble relativement stable, avec quelques variations mineures. Ces variations de pression sont souvent liées aux mouvements des masses d'air et aux conditions météorologiques.



**Figure 3.23 :** L'évolution de la pression atmosphérique

L'analyse de ces différentes courbes permet d'avoir une vision globale de l'évolution des conditions environnementales au cours de la période de mesure. Ces informations peuvent être utiles pour comprendre les interactions entre les différents paramètres et leur impact sur l'environnement.

## **Chapitre III conception et réalisation d'un datalogger pour une station météo**

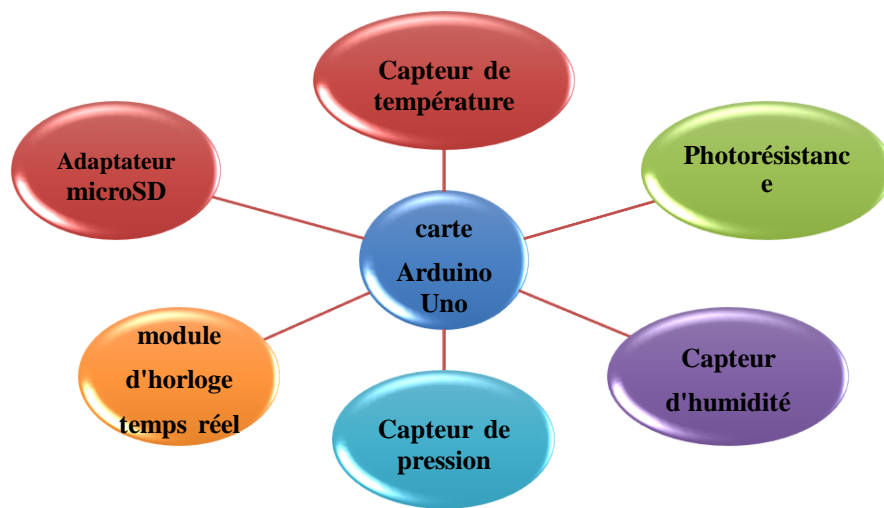
### **6. Conclusion :**

Dans ce dernier chapitre, nous avons testé et vérifié le bon fonctionnement complet de la station météo. Tout d'abord, nous avons assemblé tous les composants selon le schéma final, puis nous avons exécuté le programme principal sur l'Arduino. Pendant que le programme tournait, nous avons suivi attentivement les mesures de température, de luminosité, de date et d'heure qui s'affichaient dans le moniteur série. Cela nous a permis de vérifier que chaque capteur fonctionnait correctement et que les valeurs étaient cohérentes.

Ensuite, nous avons retiré la carte micro SD et nous avons regardé le fichier de données enregistrées. Nous avons vu que les mesures étaient bien sauvegardées et que les valeurs correspondaient bien à ce que nous avions vu dans le moniteur série.

Cette évaluation finale nous a permis de s'assurer de la fiabilité, de la précision et de la cohérence de tout le système. Nous avons vérifié que les capteurs fonctionnaient de manière stable, que les mesures étaient justes et que les données enregistrées étaient complètes.

Au final, nous pouvons dire que la station météo a atteint ses objectifs de conception et offre de bonnes performances. Les résultats obtenus montrent que le système peut mesurer avec précision les paramètres environnementaux et les enregistrer de façon fiable sur la carte micro SD. Cette validation finale nous conforte dans la qualité du travail réalisé.



**Schéma synoptique de dataloggers de la carte Arduino**

➤ **Conclusion générale**

Notre projet de conception d'une station météo autonome avec systèmes embarqués a suivi une démarche structurée en trois étapes principales. Tout d'abord, nous avons exploré les généralités sur le datalogger, présentant les différentes technologies disponibles et l'importance des dataloggers pour la collecte et le stockage de données environnementales. Cette base théorique était essentielle pour comprendre le fonctionnement et l'intérêt de tels systèmes.

Ensuite, nous nous sommes penchés sur les technologies de l'Internet des Objets, en particulier l'utilisation de la carte Arduino UNO, examinant les composants matériels et logiciels nécessaires. Cette étape nous a permis de nous familiariser avec les outils clés pour la réalisation de notre projet.

Nous avons conçu et réalisé une station météo autonome basée sur une plateforme Arduino, qui permis de mesurer et d'enregistrer de manière fiable différents paramètres environnementaux avec l'utilisation de capteurs de haute qualité et à une conception rigoureuse du circuit, nous avons obtenu des mesures précises et cohérentes, avec une carte microSD pour stocker des données permet une autonomie et une capacité de stockage élevées. L'intégration du module RTC\_DS1307 ajouté une dimension temporelle essentielle aux relevés, permettant une analyse plus complète et contextuelle des données enregistrées. Les tests et la validation du système ont confirmé le bon fonctionnement de chaque composant ainsi que la fiabilité et la cohérence globale des performances de la station météo, comme en témoignent les résultats concordants affichés dans le moniteur série et les données sauvegardées sur la carte micro SD.

Ce projet nous a permis d'acquérir des compétences techniques dans les domaines de l'électronique, de l'informatique embarquée et de la programmation. La réalisation de cette station météo autonome représente une étape importante dans notre apprentissage et notre maîtrise des technologies liées aux systèmes embarqués et à l'Internet des Objets.

Au-delà de l'aspect technique, ce travail nous a également sensibilisés à l'importance des mesures environnementales et de la surveillance météorologique. La possibilité d'exploiter les données recueillies pour des analyses plus approfondies ouvre de nombreuses perspectives d'utilisation, que ce soit dans un cadre personnel, scientifique ou éducatif.

La réalisation de cette station météo autonome a été une expérience enrichissante, nous permettant de mettre en pratique nos connaissances théoriques et de développer de nouvelles

compétences. Nous sommes fiers du résultat obtenu et envisageons d'explorer davantage les possibilités offertes par ce type de système embarqué dans le cadre de futurs projets.

## Référence bibliographie:

- [1] <https://www.coldchaintech.com/blog/best-practices/what-is-a-data-logger>
- [2] <https://www.influxtechnology.com/post/data-loggers-and-their-importance>
- [3] <https://polaridad.es/fr/Internet-des-objets-IoT-avec-Arduino-Manuel-pratique-PDF/>
- [4] <https://www.gotronic.fr/>
- [5] <https://learn.adafruit.com/memories-of-an-arduino/arduino-memories>
- [6] <https://arduino-france.site/entrees-sorties-arduino/>
- [7] <https://bentek.fr/4-structure-programme-arduino/>
- [8] <https://ledisrupteurdimensionnel.com/presentation-de-la-carte-arduino-et-de-son-ide/>
- [9] [https://www.alldatasheet.com/view.jsp?Searchword=LM324&gad\\_source=1&gclid=CjwKCAjwvIWzBhAlEiwAHHWgvVCyjp1eVUv\\_r3decdLJ6XDN2fpJuJ6V\\_yzFMaXuRfBFuzULX84fSRoCMMkQAvD\\_BwE](https://www.alldatasheet.com/view.jsp?Searchword=LM324&gad_source=1&gclid=CjwKCAjwvIWzBhAlEiwAHHWgvVCyjp1eVUv_r3decdLJ6XDN2fpJuJ6V_yzFMaXuRfBFuzULX84fSRoCMMkQAvD_BwE)
- [10] <https://www.electroduino.com/introduction-to-ds1307-rtc-module/>
- [11] <https://www.dzduino.com/module-spi-pour-carte-m%C3%A9moire-micro-sd-tf-fr>