

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
جامعة بلحاج بوشعيب لعين تموشنت  
Université De Ain Témouchent - BELHADJ Bouchaib  
Département D'électronique



## Projet de fin D'étude

Domaine : SCIENCES & TECHNOLOGIES

Filière : Electronique

Spécialité : Instrumentation

### Thème :

**Evaluation des performances d'un détecteur de pluie  
a relais a base d'un capteur résistif compatible avec  
Arduino**

#### Présenté par :

-Labidine Yacine

-Benahmidane boucif abdelilleh

#### Jury:

Mr. BENDIMERAD Karim	M. A. A	C.U.B.B	Président
Mme HELLAL Melouka	M. A. A	C.U.B.B	Examineur
Mme CHAIB Amina	M. A. A	C.U.B.B	Encadrant

---

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



### **Résumé :**

Ce mémoire étudie les performances d'un détecteur de pluie à relais basé sur un capteur résistif compatible avec Arduino. Le capteur détecte l'eau en modifiant sa résistance, activant ainsi un relais pour automatiser diverses actions.

L'analyse évalue la sensibilité du capteur, sa rapidité de réaction, et sa fiabilité. Les résultats des différents tests effectués au laboratoire et en extérieur démontrent une bonne efficacité, mais soulignent des axes d'amélioration en termes de précision et de robustesse. Des recommandations sont proposées pour optimiser le système et renforcer ses performances

**Mots-clés :** Détecteur de pluie, Capteur résistif, Relais, Arduino.

### **Abstract:**

This thesis studies the performance of a relay rain detector based on an Arduino-compatible resistive sensor. The sensor detects water by changing its resistance, thus activating a relay to automate various actions.

The analysis evaluates the sensor's sensitivity, reaction speed, reliability. The results of the various tests carried out in the laboratory and outdoors demonstrate good efficiency, but highlight areas for improvement in terms of accuracy and robustness. Recommendations are proposed to optimize the system and improve its performance.

**Keywords:** Rain Detection, Resistive Sensor, Relay, Arduino.

### **المخلص**

تدرس هذه الأطروحة أداء كاشف المطر المرحل الذي يعتمد على مستشعر مقاوم متوافق مع أردوينو. يكتشف المستشعر الماء عن طريق تغيير مقاومته، وبالتالي تنشيط مرحل لأتمتة إجراءات مختلفة

يقوم التحليل بتقييم حساسية المستشعر وسرعة رد فعله وموثوقيته. تظهر النتائج كفاءة جيدة، لكنها تسلط الضوء على مجالات التحسين. من حيث الدقة والمتانة. يتم اقتراح توصيات لتحسين النظام وتحسين أدائه

الكلمات الرئيسية: اكتشاف المطر مستشعر المقاومة المرحل اردوينو

# REMERCIEMENTS

Nous remercions tout d'abord le bon Dieu pour la force et la persévérance qui nous ont permis d'achever ce travail.

Nous adressons nos sincères remerciements à Mme CHAIB Amina pour son encadrement, ses conseils précieux et ses directives qui ont grandement contribué à la réalisation de ce mémoire.

Nous exprimons également notre gratitude aux membres du jury pour leur présence et pour l'évaluation de ce travail.

J'exprime ma gratitude envers l'ensemble des enseignants du domaine d'instrumentation pour la qualité de la formation qu'ils nous ont dispensée tout au long de notre parcours.

Enfin, un grand merci à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation et à la réussite de ce projet.

# DÉDICACES

Aujourd'hui, nous tournons une belle page de notre vie. Après des années d'efforts, de patience et de persévérance, nous sommes fier d'avoir atteint cet objectif. Ce chemin n'a pas été facile, mais il nous a appris beaucoup et nous a rendu plus fort.

Nous remercions Dieu pour sa guidance, ainsi que toutes les personnes qui nous ont soutenu tout au long de cette aventure.

Nous dédions ce travail :

À nos familles pour son amour et son soutien.

À nos enseignants pour leurs efforts et leurs conseils.

À nos amis et camarades pour leur présence et leurs encouragements.

Merci à tous ceux qui ont cru en nous. Ce succès est aussi le votre.

**LABIDINE Yacine et BENAHMIDANE Boucif Abdelilleh**

# Liste des Tableaux

**Tableau I.1 :** Grandeurs d'entrée et de sortie et effet utilisé pour les capteurs actifs.

**Tableau I.3 :** Exemples des capteurs TOR, analogiques et numériques

**Tableau I.2 :** Type de matériau utilisé et caractéristique électrique des capteurs passifs

**Tableau II.1 :** Caractéristiques techniques du contrôleur de la carte Arduino UNO.

**Tableau II.2 :** Composants de la carte Arduino UNO

**Tableau II.3 :** Tableau récapitulatif des trois étages du prototype.

**Tableau II.4 :** Caractéristiques du capteur de pluie RD-MH.

**Tableau II.5 :** Caractéristiques d'une LED

**Tableau II.6 :** Caractéristiques d'un moteur DC 5V

**Tableau III.1 :** Caractéristiques du capteur DHT22

**Tableau III.2 :** La réponse du capteur à différentes quantités de pluie.

**Tableau III.3:** Données collectées lors des différents tests du détecteur de pluie.

**Tableau III.4:** Réponse du capteur à différents niveaux d'humidité

**Tableau III.5:** Influence de la température sur le fonctionnement du capteur

**Tableau III.6:** Statistiques sur la fréquence des fausses détections en laboratoire et à l'extérieur.

# Liste des figures

**Figure I.1:** Principe de Fonctionnement d'un capteur

**Figure I.2 :** Composition d'une chaîne de mesure standard.

**FigureI.3 :** Le fonctionnement du capteur actif, avec un exemple

**FigureI.4 :** Le fonctionnement du capteur passif, avec un exemple

**Figure I.5 :** Domaines d'utilisation du capteur.

**FigureI.6:** Exemple de linéarisation de caractéristiques

**FigureI.7 :** Exemple de capteur résistif

**Figure I.8 :** Exemple d'un capteur de pluie résistif YL-83.[

**Figure I.9 :** Un capteur de pluie standard.

**Figure I.10 :**La carte Arduino.

**Figure I.11 :** Le microcontrôleur d'Arduino

**Figure I.12 :** Différents types de la carte Arduino

**Figure II.1 :** Le logiciel Proteus

**Figure II.2 :** La carte Arduino UNO

**Figure II.3 :** Composants de la carte Arduino Uno

**Figure II.4 :** Interface graphique du logiciel Arduino IDE

**Figure II.5 :** Barre de boutons Arduino

**Figure II.6** Organigramme de notre prototype détecteur de pluie.

**Figure II.7 :** Schéma de câblage du détecteur de pluie sur Proteus.

**Figure II.8 :** Code Arduino associé à la simulation du détecteur de pluie.

**FigureII.9**Interaction entre **Les** composants électroniques utilisés en réalisation.

**Figure II.10 :** Capteur de pluie RD-MH compatible avec Arduino

**Figure II.11**Raccordement du capteur avec l'Arduino.

**Figure II.12 :** KY-019 Module relais 5V

**Figure II.13** Diodes électroluminescentes (Rouge, Jaune, Vert)

**Figure II.14**Moteur DC (5V)

**Figure II.15 :** Photo réelle du prototype réalisé : détecteur de pluie à relai.

**Figure III.1 :** système d'irrigation des champs

**Figure III.2 :**Fermeture du toit en fonction du capteur de pluie

**Figure III.3** Exemple de capteur de pluie pour voiture

**Figure III.4 :** capteur de température et humidité DHT 22

**Figure III.5 :** Ecran LCD

**Figure III.6 :**Prototype détecteur de pluie amélioré, au laboratoire.

**Figure III.7 :** Courbe du taux de fausses alarmes en fonction de l'humidité ambiante

**Figure III.8 :** Test du prototype de détection de pluie en extérieur.

**Figure III.9 :** Courbe illustrant l'effet de la température sur le détecteur.

**Figure III.10 :** Histogramme illustrant une comparaison de précision : laboratoire/extérieur.

**Figure III.11 :** Installation optimisée du détecteur de pluie : capteur en position inclinée.

**Figure III.12 :** Capteur de pluie capacitif IBR274

# Liste des annexes

**Annexe 1** : Schéma interne de la carte Arduino UNO

**Annexe 2** : Simulation sur Proteus du prototype détecteur de pluie amélioré muni de son code Arduino

# Table des matières

Résumé .....	I
Remerciements .....	III
Dédicaces .....	III
Liste des Tableaux.....	IV
Liste des figures.....	V
Liste des annexes .....	VII
Table des matières.....	IV
<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>Chapitre 01 : Notion de capteurs, captures de pluie, et la carte Arduino .....</b>	<b>4</b>
1.1 Introduction.....	5
1.2 Définition d'un capteur .....	5
1.3 L'utilité du capteur.....	6
1.4 Chaîne de mesure.....	6
1.5 Capteurs passifs et Capteurs actifs .....	7
1.5.1 Capteur actif.....	7
1.5.2 Capteur passif.....	8
1.6 Métrologie des performances d'un capteur.....	9
1.6.1 Étendue de la mesure .....	9
1.7 Principales caractéristiques des capteurs .....	10
1.7.1 Sensibilité.....	10
1.7.2 Rapidité.....	10
1.7.3 Précision.....	10
1.7.4 Finesse .....	10
1.7.5 Linéarité.....	11
1.8 Classification des capteurs.....	11
1.9 Technologie des capteurs résistifs.....	12
1.9.1 Avantages et limites des capteurs résistifs.....	13
1.9.2 Le capteur de pluie résistif .....	14
1.10 Les cartes programmables .....	16
1.10.1 La révolution des systèmes embarqués .....	16
1.10.2. Définition d'une carte programmable .....	17
1.10.3. La carte Arduino .....	17
1.10.3.3 Atouts de la carte Arduino [12,16].....	18
1.10.3.4.b Les avantages de la carte Arduino .....	19
1.10.4 La famille de l'Arduino.....	19

I.10.4.1	Carte Arduino UNO .....	20
I.10.4.2	Carte Arduino Mega 2560 .....	20
I.10.4.3	Carte Arduino LEONARDO .....	20
I.10.4.4	Carte Arduino Micro .....	20
I.10.4.5	Carte Arduino NANO .....	20
I.10.4.6	Carte Arduino Mini 05 .....	21
I.10.4.7	Carte Arduino DUE .....	21
I.10.4.8	Carte Arduino Esplora.....	21
I.10.4.9	Carte Arduino Yun Mini .....	21
I.11	Conclusion.....	21
	<b>Chapitre 02 : Simulation et réalisation du prototype : Détecteur de pluie à relais .....</b>	<b>22</b>
II. 1	Introduction .....	23
II.2	Présentation des technologies utilisés .....	23
II.2.1	Le logiciel Proteus.....	23
II.2.2	La carte Arduino UNO .....	24
II.2.2.1	Caractéristiques de la carte Arduino UNO.....	25
II.2.2.2	Composition de la carte Arduino .....	26
II.2.3	Environnement de programmation d'Arduino UNO (IDE).....	29
II.3	Phase de simulation du prototype .....	31
II.3.1	Description du prototype : Détecteur de pluie.....	32
II.3.2	Schéma de câblage du prototype Détecteur de pluie sur Proteus .....	34
II.3.3	Principe de fonctionnement du détecteur de pluie simulé:.....	35
II.4	Phase de réalisation du prototype .....	36
II.4.1	Description des composants électroniques utilisés en réalisation.....	37
II.4.1.1	Le capteur de pluie MH-RD.....	37
II.4.1.2	Le relais .....	39
II.4.1.3	Diode électroluminescente (LED) .....	40
II.4.1.4	Le moteur DC (5V).....	41
II.4.2	Réalisationmatérielle.....	42
II. 5	Conclusion.....	43
	<b>Chapitre 03 : Evaluation des performances du prototype : Détecteur de pluie à relais.....</b>	<b>44</b>
III. 1	Introduction.....	45
III.2	Utilisation du détecteur de pluie dans les systèmes automatisés .....	45
III.2.1	Commande automatique de l'irrigation en agriculture à l'aide d'un capteur de pluie : .	45
III.2.2	Utilisation d'un capteur de pluie pour l'automatisation domestique .....	46

III.2.3	Système d'essuie-glaces intelligent base sur un capteur de pluie.....	47
III.3	Méthodologie d'évaluation des performances .....	47
III.3.1	Sensibilité .....	47
III.3.2	Temps de réponse.....	47
III.3.3	Précision .....	47
III.3.4	Durabilité.....	47
III.4	Amélioration du prototype détecteur de pluie .....	48
III.4.1	Intégration d'un capteur de température et humidité ambiante DHT 22 .....	48
III.4.2	Intégration d'un écran LCD (16*2).....	49
III.4.3	Intégration avec Arduino .....	50
III.4.4	Réalisation du Prototypé détecteur de pluie amélioré.....	50
III.5	Protocole de Test .....	51
III.5.1	Tests au laboratoire .....	51
III.5.1.1	Temps de réponse.....	51
III.5.1.2	Seuil de déclenchement .....	51
III.5.1.1	Taux de fausses alarmes .....	52
III.5.2	Tests en extérieur .....	53
III.5.2.1	Sensibilité du capteur à l'humidité .....	54
III.5.2.2	Sensibilité du capteur à la température .....	55
III.5.3	Comparaison de précision Laboratoire/Extérieur .....	56
III.6	Analyse des Résultats .....	57
III.6.1	Sensibilité .....	57
III.6.2	Stabilité du relais.....	57
III.6.3	Résistance et durabilité.....	57
III.6.4	Performancesface à la température.....	57
III.6.5	Limites observées.....	57
III.6.6	Comparaison aux attentes théoriques .....	57
III.7	Optimisations Possibles .....	58
III.7.1	Installation optimisée du détecteur de pluie.....	58
III.7.2	Seuil de déclenchement ajustable.....	58
III.7.3	Remplacement du capteur de pluie résistif .....	58
III.7.4	Double capteur pour validation croisée .....	59
III.7.5	Algorithme de temporisation intelligente.....	59
III.8	Conclusion .....	60
	<b>Conclusion générale.....</b>	<b>62</b>

<b>Les références Bibliographiques .....</b>	<b>64</b>
<b>Les annexes.....</b>	<b>66</b>

# **INTRODUCTION GENERALE**



## Introduction générale

Dans un monde de plus en plus tourné vers l'automatisation et l'optimisation de l'usage des ressources, les capteurs environnementaux jouent un rôle crucial dans la conception de systèmes intelligents. Parmi ces capteurs, le détecteur de pluie est un élément essentiel, notamment dans les domaines de la domotique, l'agriculture automatisée et la météorologie. Dans l'agriculture, par exemple, un système de détection de pluie peut interrompre automatiquement l'irrigation lors d'une averse, contribuant ainsi à l'économie d'eau et à la préservation des sols. Dans le domaine domestique, il permet la fermeture automatique des fenêtres ou des toits ouvrants en cas de pluie, évitant ainsi des dommages matériels et assurant le confort des occupants. De même, dans des applications météorologiques locales, ces capteurs peuvent fournir des données utiles à petite échelle.

Grâce à ce type de capteur, il devient possible de réagir automatiquement aux conditions climatiques en temps réel, améliorant ainsi la performance, la sécurité et l'efficacité énergétique des systèmes.

Cependant, les systèmes de détection de pluie disponibles sur le marché sont souvent soit coûteux, soit complexes à intégrer dans des systèmes à base de microcontrôleurs simples comme Arduino, largement utilisés dans les projets éducatifs, les prototypes industriels ou les installations domotiques.

Le but de ce mémoire est donc, de concevoir et d'évaluer les performances d'un prototype détecteur de pluie à relai à base d'un capteur de pluie résistif compatible avec Arduino, qui répond à la fois aux critères de simplicité, efficacité et facilité d'intégration dans des solutions automatisées tout en maintenant un coût abordable.

Le mémoire est structuré en trois chapitres comme suit :

Le premier chapitre comporte de généralités sur les capteurs, les capteurs de pluie résistifs et les cartes programmables.

Dans le second chapitre nous exposons en détails, les deux phases de réalisation du prototype initial (détecteur de pluie à relai). Il s'agit de la phase de simulation en utilisant le logiciel Proteus, et celle de réalisation matérielle en utilisant des composants électroniques réels.

Dans le troisième et dernier chapitre nous proposons une amélioration de notre prototype, ainsi que quelques solutions optimales au profit des tests effectués au laboratoire et à l'extérieur.

Le mémoire se termine par une conclusion générale mettant en valeur nos acquis de cette expérience.

**CHAPITRE 01 :**

**NOTION DE CAPTEURS,  
CAPTEURS DE PLUIE, ET  
LA CARTE ARDUINO**

---

## 1.1 Introduction :

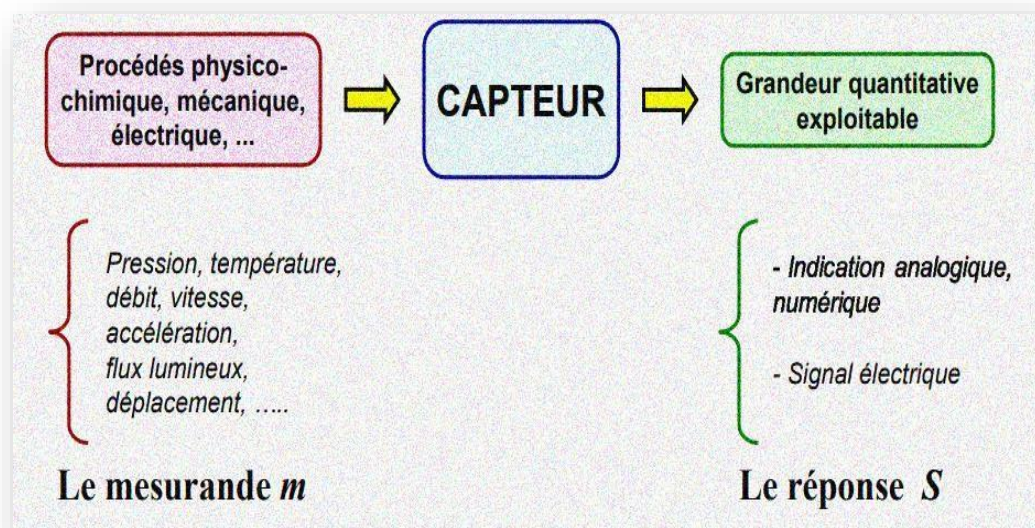
Lorsqu'il est nécessaire de convertir une grandeur physique en une autre exploitable par un système, on utilise *un capteur*. Ce dispositif a pour fonction de fournir une représentation compréhensible d'un phénomène physique, permettant ainsi son intégration dans un processus plus large.

D'autre part, personne ne peut ignorer la révolution des systèmes embarqués qui a bouleversé le monde avec une nouvelle notion d'un système électronique et informatique autonome.

De ce fait, on présente dans ce chapitre des généralités essentiels pour la compréhension des capteurs, des capteurs de pluie résistifs et de la carte Arduino.

## I.2 Définition d'un capteur :

Un **capteur** est un dispositif qui détecte une grandeur physique ou chimique (comme la température, la pression, la lumière, le mouvement, etc.) et la convertit en un signal électrique ou numérique exploitable par un système de mesure ou de commande.[1] La figure I.1 illustre le principe de Fonctionnement d'un capteur



**Figure I.1:** Principe de Fonctionnement d'un capteur

### I.3 L'utilité du capteur :[2]

Il existe plusieurs grandeurs physiques, telles que la température, le flux optique, la position, le débit, la vitesse ou encore l'humidité. Le rôle d'un capteur est de convertir ces grandeurs en données exploitables afin de permettre leur traitement ultérieur.

On peut alors citer quelques mesures effectuées par un capteur comme :

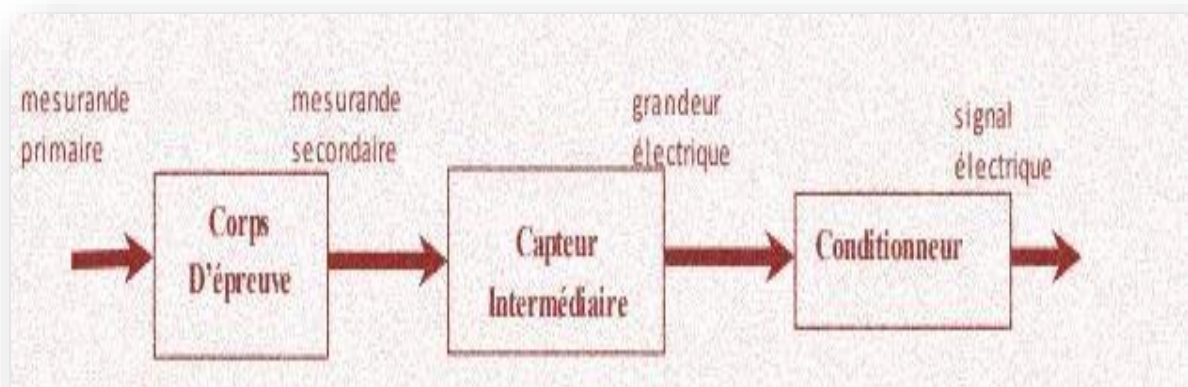
- **Mesure de présence** : détecte la proximité immédiate d'un objet.
- **Mesure de position, de déplacement ou de niveau** : détermine la position actuelle d'un objet en mouvement de rotation ou de translation.
- **Mesure de vitesse** : évalue la vitesse linéaire ou angulaire d'un objet.
- **Mesure d'accélération, de vibrations ou de chocs.**
- **Mesure de débit, de force, de couple et de pression.**
- **Mesure de température et d'humidité.**

### I.4 Chaîne de mesure :[3]

La chaîne de mesure se compose de trois étapes essentielles :

1. **Corps d'épreuve** : est en contact direct avec la grandeur à mesurer.
2. **Capteur intermédiaire** : ce transducteur, couplé à un conditionneur, génère un signal électrique exploitable en sortie.
3. **Conditionneur** : Ajuste le signal électrique en sortie tout en déterminant sa nature finale.

Ces trois éléments forment ensemble la chaîne de mesure, comme le montre la figure I.2.



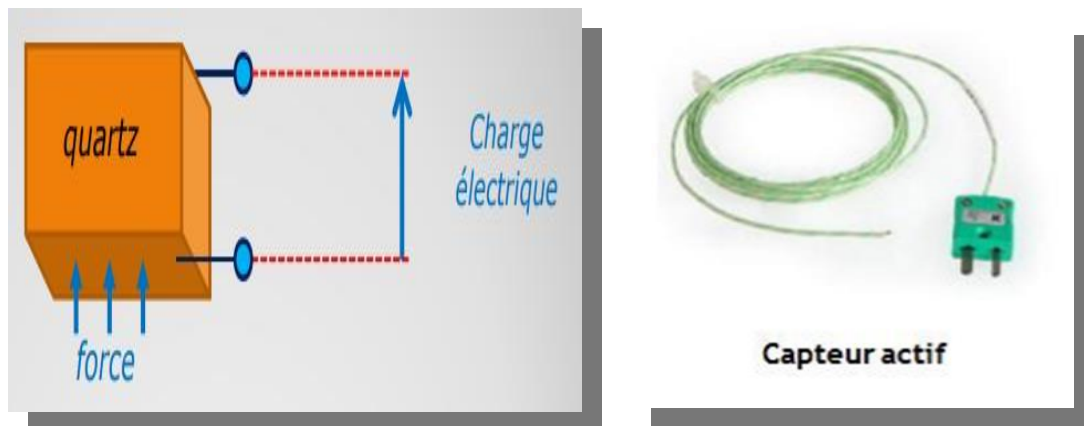
**Figure I.2** : Composition d'une chaîne de mesure standard.[3]

## I.5 Capteurs passifs et Capteurs actifs :[4]

Les capteurs peuvent être classés en deux catégories en fonction des phénomènes physiques qu'ils exploitent : **les capteurs actifs** et **les capteurs passifs** :

### I.5.1 Capteur actif :

Ce capteur repose sur un effet physique permettant la conversion d'une énergie (thermique, mécanique ou de rayonnement) en énergie électrique (voir **Figure I.3**).



**Figure I.3** : Le fonctionnement du capteur actif, avec un exemple [5].

Les effets physiques les plus couramment utilisés en instrumentation, sont regroupés dans le **Tableau I.1** :

Grandeur physique mesurée	Effet utilisé	Grandeur de sortie
Température	Thermoélectricité	Tension
Flux de rayonnement optique	Photoémission	Courant
	Effet photovoltaïque	Tension
	Effet photo-électrique	Tension
Force	Piézo-électricité	Charge électrique
Pression		
Accélération	Induction électromagnétique	Tension
Vitesse		
Position (Aimant)	Effet Hall	Tension
Courant		

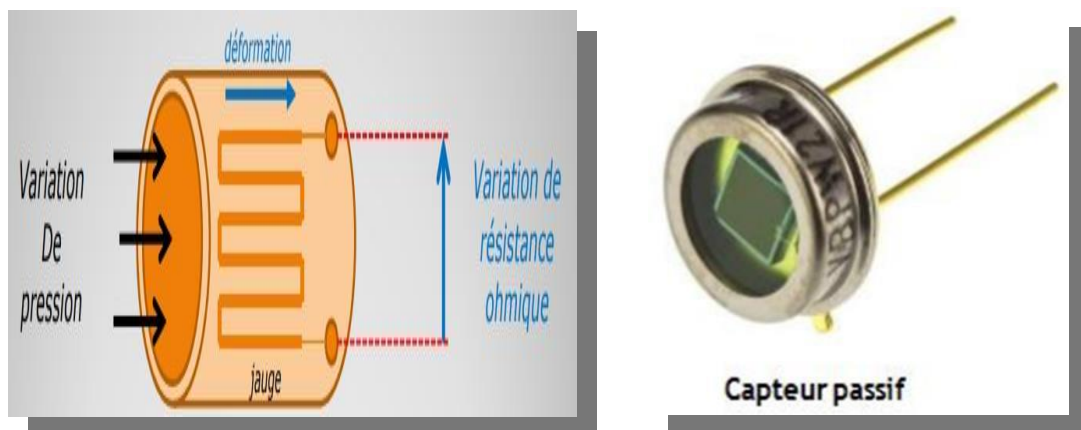
**Tableau I.1** : Grandeurs d'entrée et de sortie et effet utilisé pour les capteurs actifs.[6]

## I.5.2 Capteur passif :

Les capteurs passifs sont généralement constitués d'éléments dont l'impédance dépend d'un paramètre sensible à la grandeur mesurée. La variation de cette impédance peut être due à :

- **Une modification des dimensions du capteur** : Ce principe est utilisé dans de nombreux capteurs de position, tels que les potentiomètres, les inductances à noyau mobile ou encore les condensateurs à armature mobile.
- **Une déformation sous l'effet d'une force ou d'une grandeur associée** : Par exemple, une armature de condensateur soumise à une différence de pression, ou une jauge d'extensomètre attachée à une structure déformable pour mesurer une contrainte mécanique, la pression ou l'accélération.

La **Figure I.4** illustre le principe de fonctionnement du capteur passif, avec un exemple :



**FigureI.4** : Le fonctionnement du capteur passif, avec un exemple [5]

Le tableau **I.2** regroupe quelques types de matériaux utilisés et caractéristiques électriques des capteurs passifs.[6]

Grandeur mesurée	Caractéristique électrique	Type de matériau utilisé
Température	Résistivité	Métaux : platine, nickel, cuivre ...
Très basse température	Constante diélectrique	Verre
Flux de rayonnement optique	Résistivité	Semi-conducteur
Déformation	Résistivité	Alliage de Nickel, silicium dopé
	Perméabilité	Alliage ferromagnétique
Position (aimant)	Résistivité	Matériaux magnéto résistants : bismuth, antimoine d'indium
Humidité	Résistivité	Chlorure de lithium

**Tableau I.2** : Type de matériau utilisé et caractéristique électrique des capteurs passifs.[6]

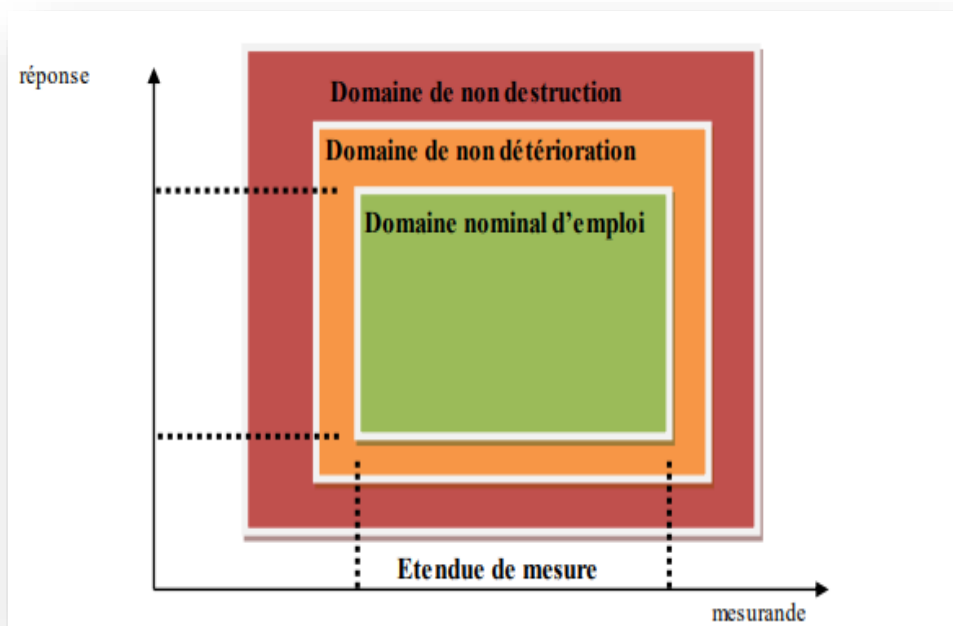
## I.6 Métrologie des performances d'un capteur:[6-7]

Afin de classer les capteurs selon leurs performances, il est essentiel de définir des paramètres permettant de les sélectionner en fonction de l'application souhaitée :

### I.6.1 Étendue de la mesure :

L'étendue de la mesure correspond à la plage dans laquelle les performances du capteur sont garanties selon des spécifications définies. Cette plage se divise en trois catégories (voir **Figure I.5**) :

- a) **Domaine nominale d'emploi** : Région où la grandeur mesurée peut varier sans altérer les caractéristiques du capteur.
- b) **Domaine de non-détérioration** : Intervalle des valeurs limites des facteurs influençant le capteur (tels que la grandeur mesurée, la température, l'environnement, etc.), dans lequel le capteur conserve ses caractéristiques une fois les surcharges éventuelles annulées.
- c) **Domaine de non-destruction** : Limite au-delà de laquelle le capteur n'est pas détruit, bien que des modifications permanentes de ses caractéristiques puissent survenir.



**Figure I.5 :** Domaines d'utilisation du capteur.

## I.7 Principales caractéristiques des capteurs :[8-9] :

**I.7.1 Sensibilité :** Elle correspond à la plus petite variation d'une grandeur physique que le capteur est capable de détecter. Pour un capteur linéaire, cette sensibilité reste constante.

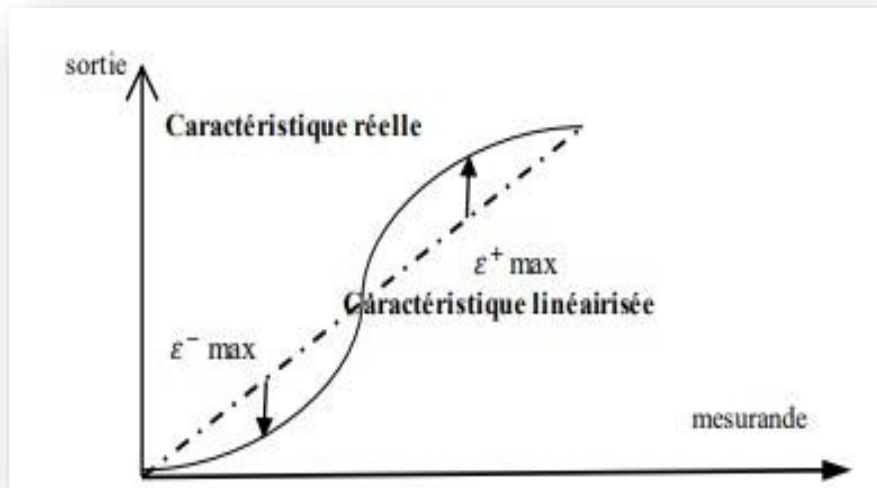
$$\text{Sensibilité(s)} = \text{grandeur de sortie/mesurande}$$

**I.7.2 Rapidité :** C'est le temps de réponse du capteur entre la variation de la grandeur mesurée et la prise en compte de cette information par le système de commande. La rapidité est directement liée à la bande passante du capteur.

**I.7.3 Précision :** C'est la capacité du capteur à fournir une mesure fidèle à la valeur réelle de la grandeur physique. Une meilleure précision implique une réduction des erreurs de mesure.

**I.7.4 Finesse :** Elle correspond à l'aptitude du capteur à ne pas perturber la grandeur qu'il mesure par sa propre présence. Cette caractéristique s'évalue en tenant compte non seulement du capteur lui-même, mais aussi de son environnement d'utilisation.

**I.7.5 Linéarité** : C'est la plage dans laquelle la sensibilité du capteur reste constante, indépendamment de la valeur de la grandeur mesurée. Elle est souvent déterminée par une droite d'approximation de la caractéristique réelle du capteur. L'écart de linéarité correspond au pourcentage maximal de divergence entre la courbe réelle du capteur et cette droite d'approximation (voir **Figure I.6**).



**Figure I.6:** Exemple de linéarisation de caractéristiques [9]

## **I.8 Classification des capteurs : [10]**

On peut classer les capteurs selon leurs sorties, qui peuvent prendre différentes formes (voir **Tableau I.3**) :

1. **Capteur Tout Ou Rien (TOR)** : Il délivre une information binaire, indiquant uniquement deux états possibles (vrai/faux ou 0/1).
2. **Capteur analogique** : Sa sortie varie de manière continue en fonction de la grandeur mesurée.
3. **Capteur numérique** : Il génère une sortie sous forme de niveaux discrets de tension ou de courant, représentant des valeurs échelonnées.

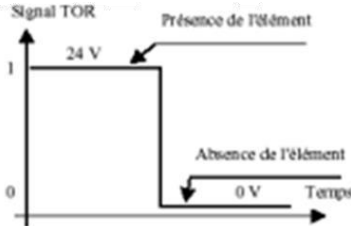
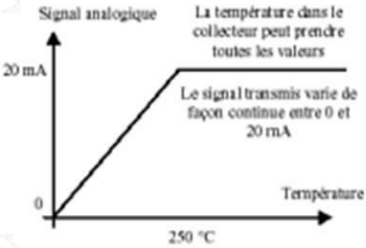
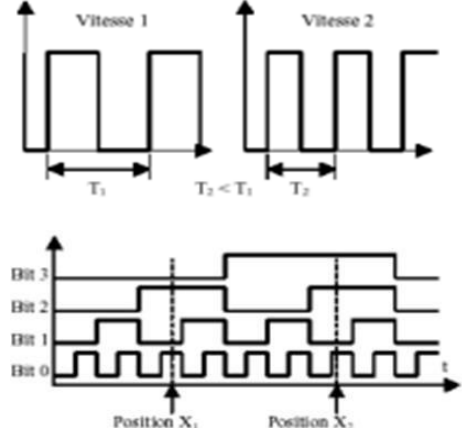
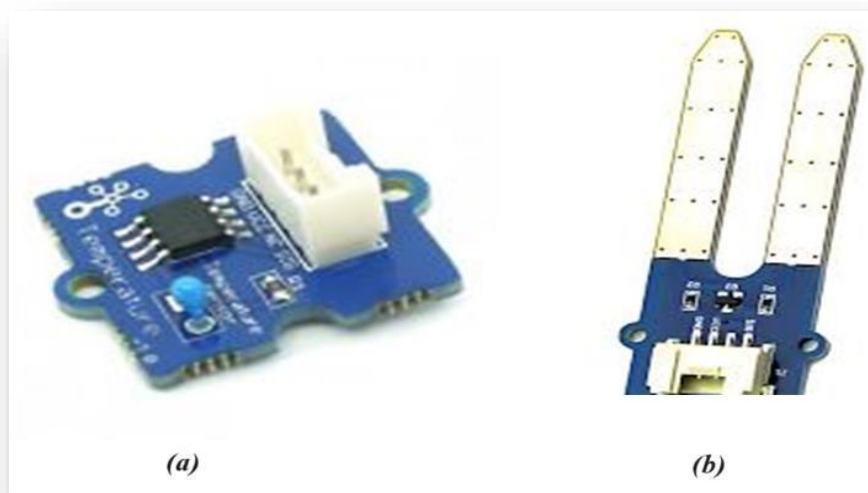
Capteur TOR	Capteur analogique	Capteur numérique
 <p>Signal TOR</p> <p>Présence de l'élément</p> <p>Absence de l'élément</p> <p>24 V</p> <p>0 V</p> <p>Temps</p>	 <p>Signal analogique</p> <p>La température dans le collecteur peut prendre toutes les valeurs</p> <p>Le signal transmis varie de façon continue entre 0 et 20 mA</p> <p>20 mA</p> <p>Température</p> <p>250 °C</p>	 <p>Vitesse 1</p> <p>Vitesse 2</p> <p><math>T_2 &lt; T_1</math></p> <p>Bit 3</p> <p>Bit 2</p> <p>Bit 1</p> <p>Bit 0</p> <p>Position <math>X_1</math></p> <p>Position <math>X_2</math></p> <p>t</p>

Tableau I.3 : Exemples des capteurs TOR, analogiques et numériques

### I.9 Technologie des capteurs résistifs :[11]

Un capteur résistif tel que celui de la **Figure I.7** fonctionne sur le principe de la variation de la résistance électrique d'un matériau conducteur ou semi-conducteur en réponse à une grandeur physique externe. Cette grandeur peut être une **température**, une **déformation mécanique**, une **pression**, ou encore un **niveau de liquide**.

Lorsque le capteur est soumis à une variation de l'environnement (par exemple une hausse de température ou une contrainte mécanique), l'un des paramètres de cette équation est modifié (souvent  $\rho$ , LLL ou AAA), ce qui entraîne un changement mesurable de la résistance. Ce changement est ensuite converti en signal électrique exploitable par un système électronique.



**Figure I.7 :** Exemple de capteur résistif : [11]

(a) Intelligent ; (b) capteur d'humidité résistif

### **I.9.1 Avantages et limites des capteurs résistifs :[12]**

#### **✚ Avantages :**

Les capteurs résistifs sont couramment utilisés dans divers domaines industriels et scientifiques grâce à leur *simplicité de conception*, leur *faible coût* et leur *facilité d'intégration* dans les systèmes.

- Leur principe de fonctionnement est simple. Par exemple, les thermistances (NTC/PTC) sont des capteurs résistifs sensibles à la température, dont la résistance varie de manière significative en fonction de la chaleur, ce qui permet de surveiller ou de réguler les processus thermiques dans les équipements industriels.
- Grâce à leur structure simple, ces capteurs peuvent être fabriqués à bas coût, ce qui les rend idéaux pour une production en grande série ou pour des applications nécessitant de nombreux points de mesure.
- Leur intégration dans les systèmes automatisés est également facilitée par leur compatibilité avec des circuits électroniques standards, tels que les ponts de Wheatstone, les microcontrôleurs ou les automates programmables (API).

- En raison de leur robustesse et de leur adaptabilité, les capteurs résistifs trouvent leur place dans des secteurs variés tels que l'automobile (détection de position ou de température), l'aéronautique, la robotique, la biomécanique, ainsi que dans les laboratoires de recherche pour la collecte de données expérimentales.

 **Limites :**

Toutefois, malgré ces avantages, ces capteurs présentent plusieurs limitations susceptibles de compromettre leur précision, leur fiabilité et leur longévité, tels que :

- ❖ **Dépendance à la propreté du capteur :** La présence d'impuretés telles que la poussière, les résidus d'huile ou l'eau sur la surface du capteur peut perturber le contact électrique ou modifier la résistivité du matériau, ce qui affecte directement la mesure.[13]
- ❖ **Vieillessement des matériaux :** L'exposition prolongée à des conditions sévères telles que la chaleur, l'humidité ou des charges électriques répétées peut altérer les propriétés physiques du matériau résistant. Ce vieillissement entraîne une dérive des caractéristiques initiales du capteur, nécessitant des recalibrages fréquents pour maintenir des mesures fiables.[14]

### **I.9.2 Le capteur de pluie résistif :**

Ce capteur est principalement utilisé pour la détection de présence de pluie (voir **Figure I.8**). Lorsque de l'eau est présente, sa conductivité augmente, entraînant une diminution de sa résistance. Ce changement peut être facilement mesuré à l'aide d'une carte Arduino permettant ainsi de surveiller la présence d'humidité du aux intempéries. Leur simplicité de *fonctionnement* et de *conception*, leur *faible coût* et leur *facilité d'intégration* dans les systèmes rend les capteurs de pluie résistifs particulièrement adaptés aux systèmes embarqués et aux environnements industriels exigeant des solutions robustes, compactes et peu coûteuses. Comme tout autre type de capteur résistif, il partage les avantages et les limites cités précédemment.



**Figure I.8** Exemple d'un capteur de pluie résistif YL-83.[15]

Le capteur de pluie présenté dans la figure (I.10) détecte de l'eau qui vient court-circuiter les pistes du circuit imprimé. Il agit comme une résistance variable qui va changer d'état : 100k ohms lorsqu'il est mouillé et 2M ohms lorsqu'il est sec. Il dispose de 2 sorties, une sortie numérique 1/0, une sortie analogique 0 à 1023. La plaque permet, au contact de l'eau, de faire passer le courant.

Un capteur de pluie standard (**Figure I.9**) se compose de deux composants essentiels [18] :

1. La plaque de détection : Cette plaque est généralement exposée à l'extérieur, installée sur le toit ou dans un emplacement propice aux précipitations. Elle détecte la pluie ou la neige fondante et agit comme un indicateur crucial pour automatiser les actions en cas de mauvais temps.
2. Le module électronique : Ce module assure la liaison entre la plaque de détection et l'Arduino. Il convertit la résistance de la plaque en une tension de sortie disponible sur une broche de sortie analogique (AO). En outre, le module utilise un comparateur haute précision LM393 pour numériser cette tension et la rendre accessible sur une broche de sortie numérique (DO).



**Figure I.9 :** Un capteur de pluie standard. [18]

Le module est également équipé d'un potentiomètre intégré pour ajuster la sensibilité de la sortie numérique (DO). En réglant un seuil avec ce potentiomètre, le module peut produire une sortie LOW lorsque la quantité d'eau dépasse la valeur seuil, et une sortie HIGH dans le cas contraire.

## **I.10 Les cartes programmables :**

### **I.10.1 La révolution des systèmes embarqués :**

L'électronique programmée, également appelée système embarqué ou informatique embarquée, désigne un système électronique intelligent qui combine étroitement du matériel (hardware) et des logiciels (software) conçus spécifiquement pour exécuter une tâche ou un ensemble de tâches bien définies. Ces systèmes sont généralement conçus pour fonctionner de manière autonome, souvent en temps réel, et sont optimisés pour répondre à des contraintes précises telles que la vitesse, la fiabilité, la consommation énergétique ou l'encombrement. On les retrouve dans de nombreux secteurs industriels et grand public : automobiles (gestion moteur, ABS), domotique (thermostats intelligents), médical (moniteurs de fréquence cardiaque), ou encore dans l'aéronautique et la robotique.

L'intégration étroite entre le logiciel et le matériel permet une grande flexibilité dans le développement de systèmes électronique complexes, tout en maintenant des coûts réduits et une grande fiabilité.

### **I.10.2. Définition d'une carte programmable :**

Une carte programmable est une carte sur laquelle sont intégrés des composants électroniques dont un ou plusieurs microcontrôleurs. Un microcontrôleur permet de contrôler la carte en y injectant un programme informatique, écrit en divers langage de programmation. Cette étape rend la carte utilisable d'une manière autonome, c'est-à-dire : sans ordinateur. [16]

Les constructeurs de cartes programmables proposent plusieurs types de cartes, comme *Arduino*, *Raspberry PI*, ou encore *Node MCU*...

Il faut alors faire le bon choix en étudiant les caractéristiques de chaque carte, En fonction du besoin souhaité, comme pour un ordinateur. L'Arduino constitue le candidat le plus populaire des cartes programmables présentant un rapport qualité-prix très abordable.

### **I.10.3. La carte Arduino :**

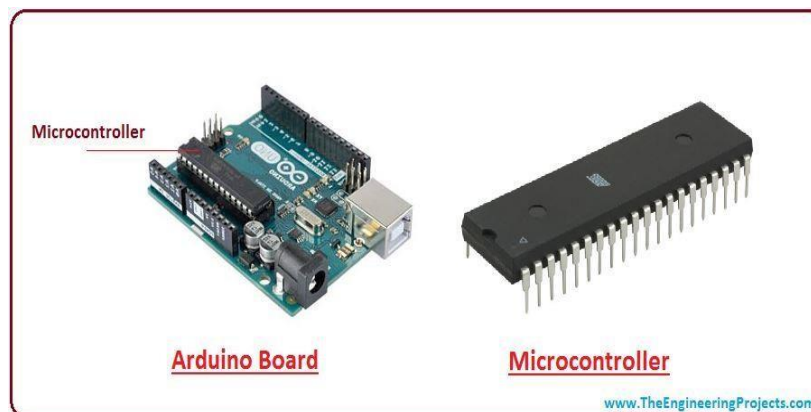
Une carte **Arduino** est une petite carte électronique équipée d'un microcontrôleur capable d'analyser et de générer des signaux électriques. Ce microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs (**Figure I.10**).



**Figure I.10 :** La carte Arduino.

### I.10.3.a Atouts de la carte Arduino [12,16]:

- ✓ **Circuit imprimé** : Cette carte est construite sur un **circuit imprimé (PCB)**, c'est-à-dire une plaque rigide sur laquelle sont soudés différents composants électroniques, connectés entre eux par des pistes conductrices formant un circuit plus ou moins complexe.
- ✓ **Matériel libre** : L'un des atouts majeurs d'Arduino est qu'il s'agit d'un matériel libre. Cela signifie que les plans de la carte (schémas électroniques et conception du circuit imprimé) sont mis à disposition gratuitement pour tous. Cette liberté est essentielle pour garantir l'ouverture, l'adaptabilité et le respect des droits liés à la propriété intellectuelle.
- ✓ **Le microcontrôleur** : L'élément central de la carte Arduino. C'est un petit ordinateur intégré qui regroupe plusieurs fonctions essentielles : un processeur, une mémoire vive (RAM), une mémoire morte (ROM), et des ports d'entrées/sorties. Sur la **Figure I.11**, il est visible sous la forme d'un grand composant noir rectangulaire muni de nombreuses broches. Une fois alimenté, il peut exécuter le programme de manière autonome, sans avoir besoin d'un ordinateur.



**Figure I.11** : Le microcontrôleur d'Arduino [12]

Arduino est donc une carte qui se connecte sur l'ordinateur pour être programmée, et qui peut ensuite fonctionner seule si elle est alimentée en énergie. Elle permet de recevoir des informations et de transmettre depuis ou vers des matériels électroniques (diodes, potentiomètres, récepteurs, servomoteurs, moteurs, détecteurs...).

### **I.10.3.b Les avantages de la carte Arduino :**

- ✓ **Simplicité d'utilisation** : Arduino est particulièrement adapté aux débutants en électronique grâce à sa facilité d'utilisation. Les cartes sont accompagnées d'un logiciel convivial qui permet de programmer et de téléverser des codes en toute simplicité.
- ✓ **Coût accessible** : Les cartes Arduino sont proposées à des prix abordables, avec une large gamme de modèles adaptés à tous les budgets, ce qui les rend accessibles au plus grand nombre.
- ✓ **Grande flexibilité** : Très polyvalente, la plateforme Arduino peut être utilisée dans une multitude de projets, allant de simples montages DIY (do-It-yourself) aux systèmes automatisés complexes dans le domaine industriel.
- ✓ **Communauté active et solidaire** : Arduino bénéficie d'une communauté vaste et dynamique qui partage conseils, tutoriels, projets et solutions. De nombreux forums, sites spécialisés et groupes d'entraide facilitent l'apprentissage et le développement de projets.
- ✓ **Compatibilité avec de nombreux composants** : Arduino prend en charge une grande variété de capteurs et de modules électroniques, ce qui permet de concevoir des projets très variés selon les besoins.
- ✓ **Plateforme open source** : Arduino étant open source, chacun peut accéder librement aux schémas des cartes, contribuer au développement de nouvelles fonctionnalités ou créer des dérivés personnalisés. Cette ouverture favorise l'innovation continue et l'enrichissement de l'écosystème.
- ✓ **Programmation accessible** : Le langage de programmation utilisé par Arduino, basé sur Wiring, est simple à appréhender, même pour les débutants, ce qui facilite la prise en main rapide de la plateforme.

### **I.10.4 La famille de l'Arduino :[17]**

Arduino n'est pas un type de carte ou un microcontrôleur spécifique. Cette carte désigne plutôt toute une famille. Choisir celle qui convient à un projet n'est pas une simple affaire et il convient de réfléchir laquelle choisir. Nous citons dans ce qui suit quelques-unes, présentées sur la **Figure I.12**.



**Figure I.12** : Différents types de la carte Arduino.[17]

**I.10.4.1 Carte Arduino UNO** : cette carte est basée sur un microcontrôleur de type ATmega328 cadencé à 16 MHz. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

**I.10.4.2 Carte Arduino Mega 2560** : cette carte est basée sur un microcontrôleur ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs.

**I.10.4.3 Carte Arduino LEONARDO** : cette carte est basée sur un microcontrôleur ATmega32u4 cadencé à 16 MHz permettant la gestion du port USB par un seul processeur.

**I.10.4.4 Carte Arduino Micro** : cette carte est basée sur un microcontrôleur ATmega32u4 cadencé à 16 MHz permettant la gestion du port USB par un seul processeur.

**I.10.4.5 Carte Arduino NANO** : cette carte est basée sur un microcontrôleur ATmega328. Sa mémoire de 32 kB et ses E/S font de ce circuit un élément idéal pour les systèmes embarqués ou pour des applications nécessitant du multitâches.

**I.10.4.6 Carte Arduino Mini 05** : cette carte est basée sur un microcontrôleur ATmega328 cadencé à 16 MHz et convient pour des applications miniatures.

**I.10.4.7 Carte Arduino DUE** : cette carte dispose de 54 E/S dont 12 PWM, 12 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que l'UNO.

**I.10.4.8 Carte Arduino Esplora** : cette carte possède une implantation à l'image d'une manette de jeux et est équipée de capteurs et actionneurs permettant de l'utiliser directement pour effectuer des essais.

**I.10.4.9 Carte Arduino Yun Mini** : cette carte Yun Mini est basée sur un ATmega32u4 (comme la Leonardo) et sur un Atheros AR9331. Le processeur Atheros supporte la distribution Linux Linino basée sur Open WRT.

## **I.11 Conclusion :**

Dans ce premier chapitre, nous avons exploré les notions fondamentales relatives aux capteurs. Une attention particulière a été portée aux capteurs résistifs (de pluie), en analysant leurs principes de fonctionnement, leurs avantages et leurs limitations. Ce chapitre a également introduit des informations basiques de la carte Arduino, un élément indispensable dans la conception d'un système de détection de pluie. Les connaissances acquises ici serviront alors de fondement aux chapitres suivants, consacrés à la mise en œuvre pratique et à l'évaluation des performances du prototype développé.

**CHAPITRE 02 :**

**SIMULATION ET**

**RÉALISATION DU**

**PROTOTYPE : DÉTECTEUR**

**DE PLUIE à RELAIS**

---

## II.1 Introduction :

Dans ce chapitre, nous présentons, en détail, la *mise en œuvre pratique* d'un système de détection automatique de pluie en utilisant une carte Arduino UNO. Cette réalisation se fait en deux phases :

- **La simulation du circuit électronique** (à l'aide du logiciel Proteus),
- **La réalisation physique** avec les composants réels.

Ce chapitre vise alors à valider le fonctionnement du montage à la fois virtuellement (simulation) et matériellement (prototype réel).

Avant de procéder aux détails de cette réalisation, il est commode de commencer le chapitre par une présentation des technologies que nous avons utilisé : **Le logiciel Proteus et la carte Arduino UNO.**

## II.2 Présentation des technologies utilisés :

### II.2.1 Le logiciel Proteus :[19]

**Proteus Professional** est une suite logicielle dédiée à la simulation électronique, développée par la société **Labcenter Electronics**. Elle permet la **Conception Assistée par Ordinateur (CAO)** dans le domaine de l'électronique.

Cette suite comprend plusieurs outils puissants, dont les principaux sont :

- ✓ **ISIS** : pour la création de schémas électroniques,
- ✓ **ARES** : pour la conception de circuits imprimés (PCB),
- ✓ **PROSPICE** : pour la simulation analogique,
- ✓ **VSM (Virtual System Modelling)** : pour la simulation de microcontrôleurs et de systèmes embarqués.

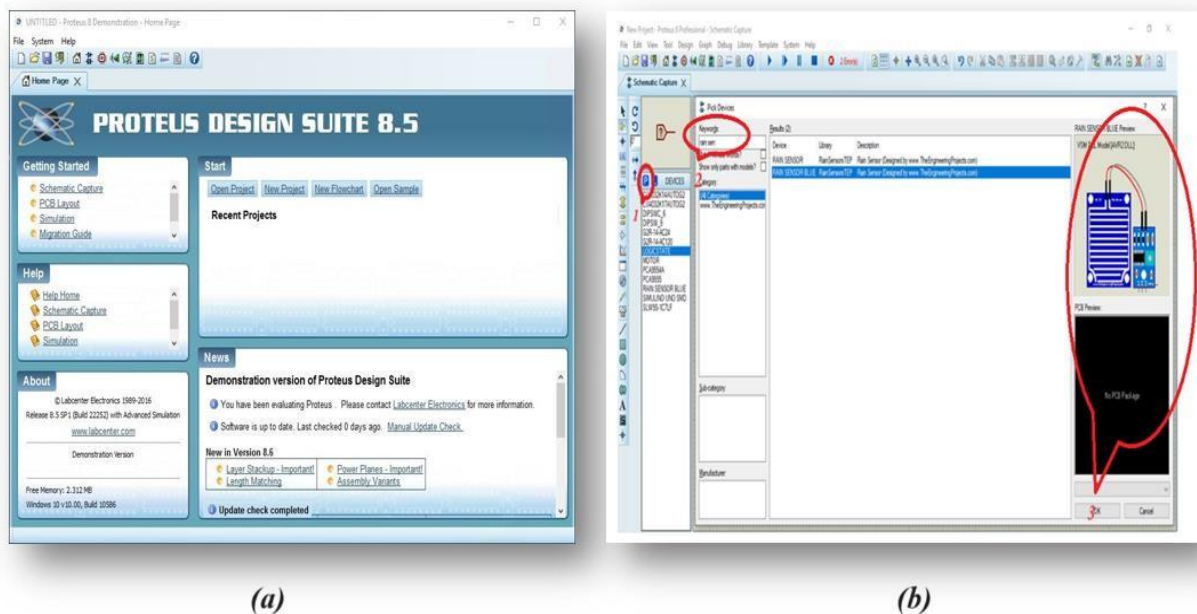
Très répandue dans le milieu de l'électronique, **Proteus Professional** est utilisée aussi bien par des entreprises que par des établissements de formation, tels que les lycées et universités.

**Parmi ses principaux atouts, on peut citer :**

1. Une suite logicielle intuitive, facile et rapide à prendre en main,
2. Un support technique efficace,

3. Un outil de prototypage virtuel permettant de réduire significativement les coûts en matériel et en développement lors de la phase de conception.

Nous avons choisi la version du logiciel **Proteus 8.5** (**Figure II. 1.a**) qui est principalement conçu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.



**Figure II.1 :** Le logiciel Proteus 8.5 : (a) Interface d'accueil ; (b) Sélection d'un composant

Pour sélectionner le composant qu'on veut utiliser, il suffit de suivre les étapes suivantes (**Figure II.1.b**) :

1. Cliquer sur l'icône 'P' (qui désigne les composants électroniques)
2. Ecrire sur la case **Keywords** le composant voulu, par exemple : Rain sensor
3. Le composant choisi s'affiche sur le volet **Preview**

## II.2.2 La carte Arduino UNO :

Parmi les types d'Arduino cités dans le premier chapitre, nous avons opté pour une carte Arduino UNO (**Figure II.2**) qui représentera le cœur de notre réalisation. Notre choix est justifié par la facilité de sa conception et la mise en œuvre ainsi que la simplicité de la façon de travailler avec son microcontrôleur. Elle est accessible à tous. Même les amateurs pourront s'en sortir aisément grâce

aux explications claires distribuées un peu partout sur internet. Elle présente aussi l'avantage d'être relativement peu coûteuse comparativement aux autres cartes programmables.



**Figure II.2 :** La carte Arduino UNO [20]

Le schéma interne de la carte Arduino UNO est en **annexe 1**.

### **II.2.2.1 Caractéristiques de la carte Arduino UNO : [20]**

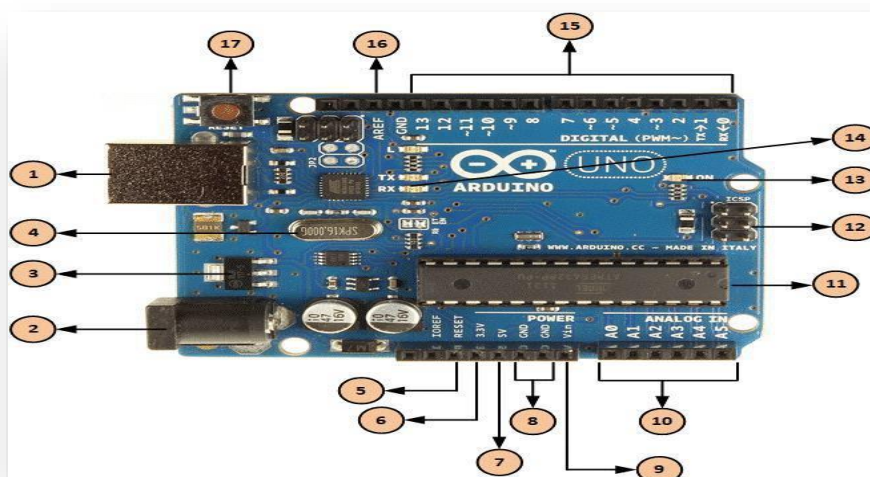
Parmi les caractéristiques techniques du contrôleur de la carte Arduino UNO, on trouve (**Tableau II.1**) :

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limitée)	6-20V
E / S numériques Pins	14 (dont 6 fournissent la sortie PWM*)
PWM numérique E / S Pins	6
Pins d'entrée analogique	6
DC Courant par E / S Pin	20 mA
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	32 KB
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB
Vitesse de l'horloge	16 MHz
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g


**Tableau II.1 :** Caractéristiques techniques du contrôleur de la carte Arduino UNO.

### II.2.2.2 Composition de la carte Arduino : [20]

Les composants de la carte Arduino représentée sur la **Figure II.3** sont détaillés dans le **Tableau II.2**.



**Figure II.3 :** Composants de la carte Arduino Uno[20]

<p style="text-align: center;">1</p>	<p><b>Le port USB a deux actions</b></p> <ul style="list-style-type: none"> <li>• Alimenter la carte en énergie électrique (5V).</li> <li>• Téléverser le programme dans la carte.</li> </ul>
<p style="text-align: center;">2</p>	<p><b>Prise jack</b></p> <ul style="list-style-type: none"> <li>• Permet de brancher une alimentation (pile, batterie, adaptateur secteur) comprise entre 5 et 12 V.</li> <li>• Est relié au régulateur de tension intégré à la carte.</li> <li>• Connexion USB.</li> </ul> 
<p style="text-align: center;">3</p>	<p><b>Régulateur de tension</b></p> <ul style="list-style-type: none"> <li>• La fonction du régulateur de tension est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.</li> </ul>
<p style="text-align: center;">4</p>	<p><b>Oscillateur à quartz</b></p> <ul style="list-style-type: none"> <li>• Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique. Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. C'est l'horloge de l'Arduino, il cadence son fonctionnement. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16 MHz.</li> </ul>

<p style="text-align: center;"><b>5,17</b></p>	<p><b>Arduino Reset</b></p> <ul style="list-style-type: none"> <li>• Permettent de redémarrer un Arduino avec un “Reset”. Cela aura pour effet de réinitialiser la carte Arduino. Il existe deux manières de redémarrer un Arduino UNO : soit en utilisant le bouton “Reset” (17), soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée “RESET” (5).</li> </ul>
<p style="text-align: center;"><b>6,7,8,9</b></p>	<p><b>Broches (3.3, 5, GND, Vin)</b></p> <ul style="list-style-type: none"> <li>• 3.3V (6) – Broche d’alimentation de tension 3.3 Volts</li> <li>• 5V (7) – Broche d’alimentation de tension 5 Volts</li> <li>• La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts.</li> <li>• GND (8) (Ground / Masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour les circuits.</li> <li>• Vin (9) – Cette broche permet d’alimenter l’Arduino depuis une source de tension extérieure. Elle est reliée au circuit d’alimentation principale de la carte Arduino.</li> </ul>
<p style="text-align: center;"><b>10</b></p>	<p><b>6 Broches analogiques</b></p> <ul style="list-style-type: none"> <li>• Permettent de brancher des capteurs et des détecteurs analogiques.</li> <li>• La carte Arduino utilise un convertisseur analogique/numérique (CAN) pour permettre la lecture du signal par le microcontrôleur. Un signal sera converti sur 10 bits. Ces six entrées peuvent donc admettre 1024 valeurs analogiques comprises entre 0 et 5 Volts.</li> </ul>
<p style="text-align: center;"><b>11</b></p>	<p><b>Microcontrôleur ATMEL ATmega328</b></p> <ul style="list-style-type: none"> <li>• Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.</li> <li>• Permet de stocker le programme et de l’exécuter. En effet c’est le cerveau de la carte.</li> </ul>

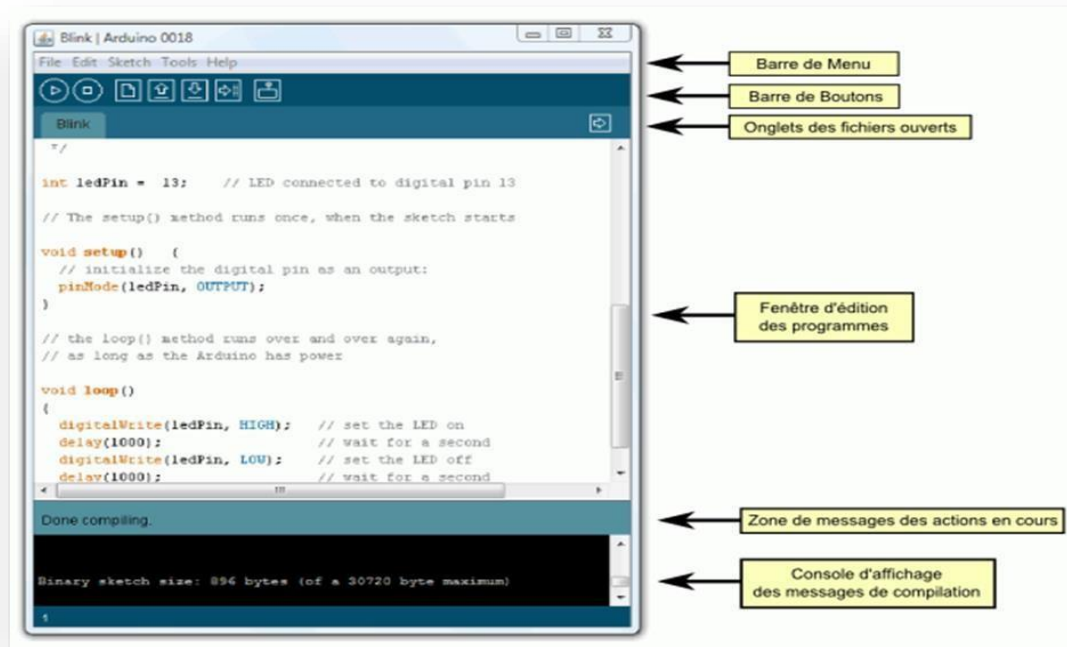
<b>12</b>	<p><b>Connecteur ICSP</b></p> <ul style="list-style-type: none"> <li>• Permet à l'Arduino de communiquer avec des composants externes via le protocole SPI (Serial Peripheral Interface).</li> </ul>
<b>13</b>	<p><b>Indicateur LED d'alimentation</b></p> <ul style="list-style-type: none"> <li>• Un voyant qui indique l'alimentation de la carte.</li> </ul>
<b>14</b>	<p><b>LEDs TX et RX</b></p> <ul style="list-style-type: none"> <li>• TX clignote lors de l'envoi des données série.</li> <li>• RX clignote pendant le processus de réception.</li> </ul>
<b>15</b>	<p><b>14 entrées et sorties numériques (digital)</b></p> <p>Ces broches peuvent être configurées pour fonctionner comme :</p> <ul style="list-style-type: none"> <li>• Broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques.</li> <li>• Broches numériques de sortie pour piloter différents modules. <ul style="list-style-type: none"> <li>❖ Remarque : Les broches étiquetées “~” peuvent être utilisées pour générer des PWM.</li> </ul> </li> </ul>
<b>16</b>	<p><b>Broche AREF</b></p> <ul style="list-style-type: none"> <li>• AREF est l'acronyme anglais de “référence analogique”. Cette broche est parfois utilisée pour définir une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.</li> </ul>

**Tableau II.2 : Composants de la carte Arduino UNO[20]**

### **II.2.3 Environnement de programmation d'Arduino UNO (IDE) [20]:**

La carte Arduino est une carte électronique programmable. Il lui faut alors un éditeur qui fonctionne sur un ordinateur et un programme qui permettra de téléverser le code vers la carte. Les créateurs d'Arduino ont développé un logiciel pour que la programmation des cartes Arduino soit visuelle, simple et complète à la fois. C'est ce que l'on appelle un **IDE**, qui signifie Integrated Development Environment ou Environnement de Développement « Intégré » en français (**EDI**). L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation (**Figure II.4**). Nous pouvons donc

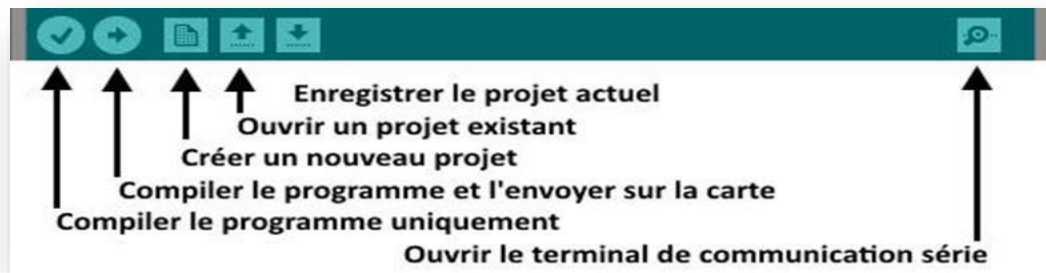
saisir notre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte Arduino.



**Figure II.4** : Interface graphique du logiciel Arduino IDE[20]

L'espace (IDE) dédié à la programmation des cartes Arduino comporte (**Figure II.4**):

- Une barre de menu comme pour tout logiciel une interface graphique (GUI),
- Une barre de boutons (**Figure II.5**) qui donnent un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,
- Un éditeur (à coloration syntaxique) pour écrire le code de vos programmes, avec onglets de navigation,
- Une zone de messages qui affiche indique l'état des actions en cours,
- Une console texte qui affiche les messages concernant le résultat de la compilation du programme.



**Figure II.5** : Barre de boutons Arduino [20]

### **+ Structure et langage d'un programme en Arduino [20]:**

Le langage Arduino est basé sur les langages C et C++. Les programmes Arduino sont intégrés dans un schéma (scketch) **setup** / **loop**. Il s'agit de deux blocs de fonctions obligatoirement présents dans tous programmes Arduino (figure II.4).

Un bloc commence par le symbole « { », et se termine par le symbole « } ». Le nom d'une fonction est toujours suivi par les symboles () .

La fonction **setup** () est appelée systématiquement au démarrage de l'Arduino, une seule fois, après un reset ou une mise sous tension. Il est utilisé pour initialiser des variables, démarrer des bibliothèques, modifier le paramétrage des broches, etc.

Après avoir utilisé la fonction **setup** (), la fonction **Loop** () exécute de manière infinie le code à l'intérieur de ce bloc afin de répondre aux interactions demandées.

Afin de rendre plus intelligible le code écrit, il est possible d'écrire du texte qui ne sera pas interpréter comme du code. Une ligne de commentaire commencera par les symboles // alors qu'un bloc de commentaires sera encadré par les symboles /\* et \*/ .

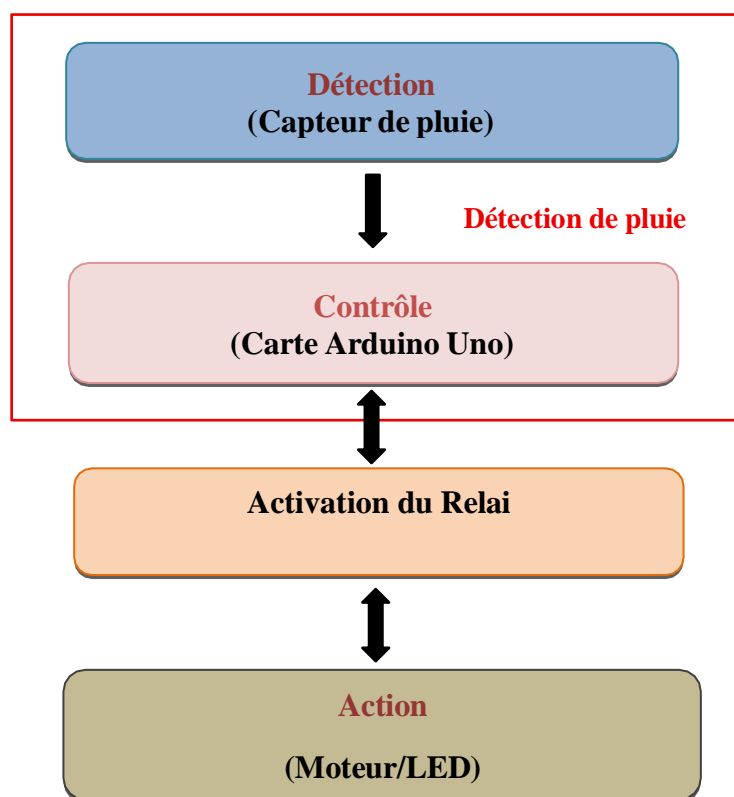
### **II.3Phase de simulation du prototype :**

La simulation est un outil primordial pour le développement scientifique et technologique, car elle offre aux fabricants de circuits la possibilité de simuler les différents circuits avant de passer à la réalisation. Cela permet alors de réduire les coûts engendrés par une suite d'essais expérimentaux, et gagner du temps.

Dans la phase de notre simulation, on a choisi le simulateur le plus utilisé : le logiciel **Proteus 8.5**, cité précédemment.

### II.3.1 Description du prototype : Détecteur de pluie :

La première partie de notre projet consiste à réaliser un détecteur de pluie à relais. Le prototype repose sur l'utilisation d'un **capteur de pluie résistif**, connecté à un **microcontrôleur Arduino**, capable de détecter la présence d'eau sur sa surface. Lorsqu'une détection est confirmée, un **relais** est activé, permettant d'allumer une LED (ou lampe) et activer d'autres systèmes (moteur DC). Son schéma (Organigramme) est représenté sur la **Figure II.6**.



**Figure II.6** : Organigramme de notre prototype détecteur de pluie.

Les composants que nous avons sélectionnés des bibliothèques **Proteus** sont les suivants :

1. Un capteur de pluie HL-83 (Rain Sensor Blue)
2. Un module de détection de pluie (compatible avec Arduino)
3. Une carte Arduino UNO

4. Un relais (RL1)
5. Un moteur DC
6. Source d'alimentation 5V

Le prototype réalisé est constitué de trois étages distincts :

- Etage de *détection de la présence de pluie* : qui comprend le capteur de pluie et la carte Arduino.
- Etage d'*activation du relai* : qui comprend le relai.
- Etage d'*action automatique* : qui comprend une LED ou un moteur DC

Le **Tableau II.3** récapitule la composition, la fonction et la connectique de chacun de ces étages :

<b>Etages du prototype</b>	<b>Constituants</b>	<b>Fonction</b>	<b>Connectique</b>
<i>Etage de détection de la présence de pluie</i>	<b>Capteur de pluie HL-83</b>	Détecte la présence de gouttes d'eau sur sa surface	Deux fils (rouge et noir) le relie au <b>module d'interface</b> : <ul style="list-style-type: none"> <li>• <b>Rouge</b> : VCC</li> <li>• <b>Noir</b> :GND</li> </ul>
	<b>Module d'interface du capteur</b>	Agit comme un <b>convertisseur de signal</b> : du capteur vers l'Arduino	<ul style="list-style-type: none"> <li>• <b>VCC</b>: connecté à la broche <b>5V</b> de l'Arduino</li> <li>• <b>GND</b>: connecté à la broche <b>GND</b> de l'Arduino</li> <li>• <b>D0 (sortie numérique)</b>:connectée à la broche <b>A0</b>de l'Arduino</li> </ul>
	<b>Carte Arduino UNO</b>	Cœur du système. Il lit le signal du module de détection et prend une décision.	<ul style="list-style-type: none"> <li>• <b>Entrée</b> :la broche <b>A0</b> d'Arduino reçoit le signal numérique de pluie (0 = pluie détectée, 1 = sec).</li> <li>• <b>Sortie</b> :la broche <b>D7</b> est utilisée pour activer le relais.</li> </ul>

<p>Etage <i>d'activation du relai</i></p>	<p><b>Relais (RL1)</b></p>	<p>Contrôle un <b>dispositif externe</b> (comme une LED ou Moteur) à partir du signal de l'Arduino</p>	<ul style="list-style-type: none"> <li>• <b>Entrée (IN)</b> du relais à la <b>broche D7</b> de l'Arduino</li> <li>• <b>VCC et GND du relais</b> respectivement <b>5V et GND</b> de l'Arduino</li> <li>• <b>Sortie du relais</b> (commutation) : connectée à une LED ou Moteur.</li> </ul>
<p>Etage <i>d'action automatique</i></p>	<p><b>LED</b></p>	<p>Avertisseur visuel qui s'allume en cas de détection de pluie</p>	<p>Elle est câblée avec la sortie du relais, alimentée via le relais lorsqu'il est activé par l'Arduino</p>
	<p><b>Moteur DC</b></p>	<p>Simule une charge à activer en cas de détection de pluie (par exemple, ouverture de fenêtre).</p>	<p>Il est commandé est câblé avec la sortie du relais.</p>

**Tableau II.3 :** Tableau récapitulatif des trois étages du prototype.

### III.3.2 Schéma de câblage du prototype Détecteur de pluie sur Proteus :

Le schéma électrique de notre prototype détecteur de pluie simulé sous le logiciel Proteus 8.5 est le suivant (**Figure II.7**) :

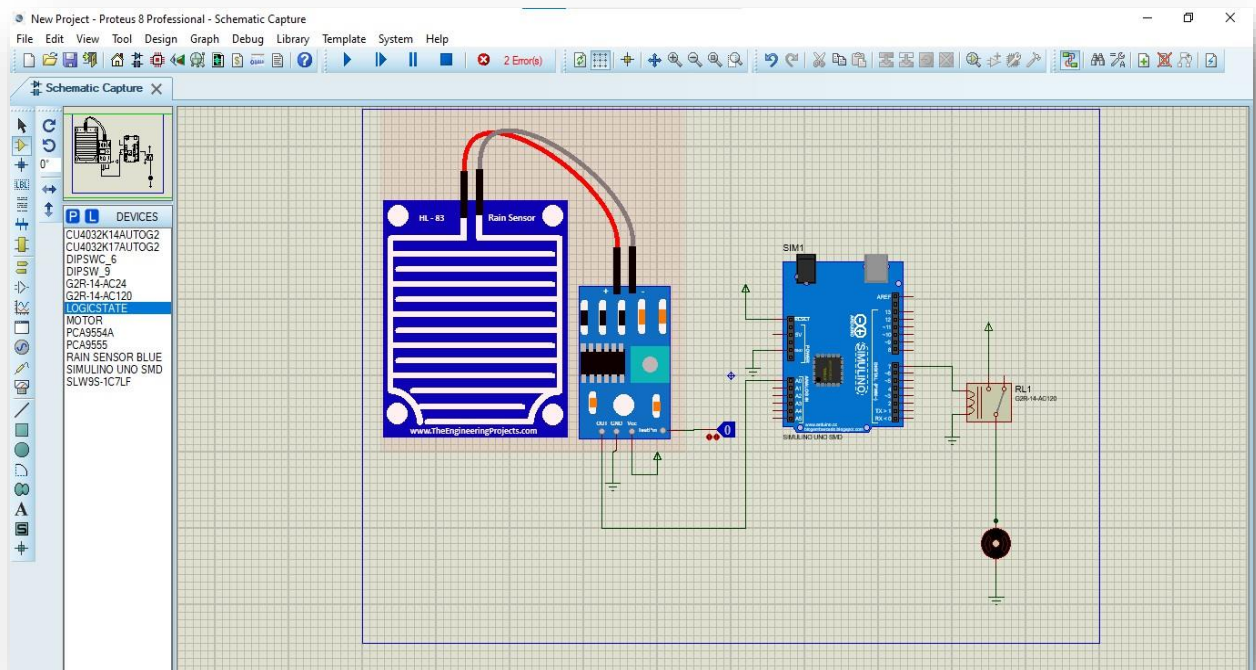


Figure II.7 : Schéma de câblage du détecteur de pluie sur Proteus.


### III.3.3 Principe de fonctionnement du détecteur de pluie simulé :

A l'aide du logiciel de simulation Proteus, on alimente le capteur de pluie par une batterie de 5V puis on injecte la sortie numérique de ce capteur A0 à la broche D2 de carte Arduino. On branche le relai avec Arduino. Le Moteur (ou LED) est relié à la sortie du relai.

Le principe de fonctionnement du prototype détecteur de pluie en *lecture tout ou rien* du capteur de pluie (sortie numérique **D0**) se résume en :

1. **S'il n'y pas de la pluie** (capteur sec) : la sortie numérique **D0=1= HIGH**, le relais est donc désactivé et la **LED** sera **éteinte** (simulé à un moteur désactivé).
2. **Si le capteur détecte la pluie** (capteur mouillé) : la sortie numérique **D0=0=LOW**, le relais est donc activé et la **LED** sera **allumée** pour signaler la pluie (simulé à un moteur activé).

Le code associé à notre simulation, que nous avons généré à l'aide du logiciel IDE Arduino cité dans la section **II.2.3**, compilé, vérifié, et transféré sur une carte Arduino, est le suivant (**Figure II.8**):

The image shows a screenshot of the Arduino IDE 2.3.6 interface. The main window displays a C++ code file named 'Blink.ino'. The code defines a constant 'relayPin' as 7, sets up pin modes for an input pin (A0) and an output pin (D7), and implements a loop that reads the input pin and controls the relay based on its state. Comments in French explain the actions: turning the relay on when the input is HIGH (turning on the lamp) and turning it off when the input is LOW (turning off the lamp). A 10ms delay is used for stabilization. The IDE's top bar shows the board set to 'Arduino Uno' and the status bar at the bottom indicates 'Output'.

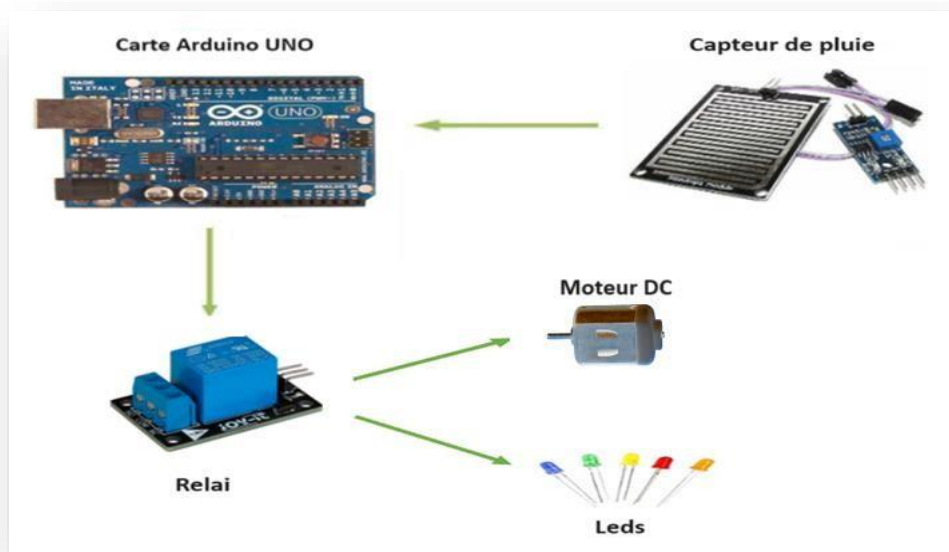
**Figure II.8** : Code Arduino associé à la simulation du détecteur de pluie.

### **II.4 Phase de réalisation du prototype :**

Après avoir exposé et présenter la phase de simulation qui précède la conception de notre projet, nous allons aborder la réalisation pratique du prototype détecteur de pluie dans sa version initiale. Cette version sera améliorée par la suite dans le troisième chapitre.

La **Figure II.9** montre les interactions entre les composants électroniques que nous avons utilisé pour la réalisation du prototype : détecteur de pluie à relai compatible avec la carte Arduino UNO. Le but étant de décrire comment se déroulent les actions entre le capteur et les actionneurs.

Dans la section ci-après (II.4.1), nous décrivons ces composants en présentant leurs caractéristiques et connectique.



**Figure II.9 :** Interaction entre Les composants électroniques utilisés en réalisation.

## II.4.1 Description des composants électroniques utilisés en réalisation :

### II.4.1.1 Le capteur de pluie MH-RD [21]

Ce capteur (**Figure II.10**) est particulièrement adapté pour détecter la présence ou l'absence de pluie, notamment dans les projets de domotique liés à l'agriculture. Il est livré avec un module d'interface comprenant un potentiomètre permettant d'ajuster la sensibilité, ainsi que des bornes offrant le choix entre une sortie numérique et une sortie analogique. Le capteur fonctionne en détectant les gouttes d'eau qui court-circuitent les pistes du circuit imprimé. Il agit alors comme une résistance variable : sa résistance chute à environ  $100\text{ k}\Omega$  lorsqu'il est mouillé, et atteint jusqu'à  $2\text{ M}\Omega$  lorsqu'il est sec.



Figure II.10 : Capteur de pluie RD-MH compatible avec Arduino [21].

Le **Tableau II.4** regroupe les caractéristiques techniques, les bornes du module d'interface ainsi que le fonctionnement des sorties du capteur de pluie RD-MH.

Caractéristiques techniques	Bornes du module d'interface	Fonctionnement des sorties
<ul style="list-style-type: none"> <li>• <b>Dimensions du capteur :</b> 39 mm x 54 mm</li> <li>• <b>Tension d'alimentation :</b> 3,3 V à 5 V</li> <li>• <b>Indicateurs :</b> LEDs de contrôle intégrées</li> </ul>	<ul style="list-style-type: none"> <li>• <b>VCC :</b> Alimentation positive (3 V ou 5 V DC)</li> <li>• <b>GND :</b> Masse (négatif)</li> <li>• <b>DO (Digital Out) :</b> Sortie numérique (valeur logique 0 ou 1)</li> <li>• <b>AO (Analog Out) :</b> Sortie analogique (valeur de 0 à 1023)</li> </ul>	<ol style="list-style-type: none"> <li>1. <b>Sortie numérique (D0) :</b> détection en tout ou rien : <ul style="list-style-type: none"> <li>• Lorsque la plaque est sèche, la sortie est à <b>1</b></li> <li>• Lorsqu'elle est mouillée, la sortie passe à <b>0</b></li> </ul> </li> <li>2. <b>Sortie analogique (A0) :</b> Fournit une valeur proportionnelle au niveau d'humidité détecté. <ul style="list-style-type: none"> <li>• <b>0V</b> correspond à une plaque complètement mouillée.</li> <li>• <b>5 V</b> indique une plaque totalement sèche.</li> </ul> </li> </ol>

Tableau II.4 : Caractéristiques du capteur de pluie RD-MH[21].

#### ✚ Raccordement du capteur avec l'Arduino:

Pour interfacer le capteur de pluie avec la carte Arduino UNO (**Figure II.11**), on suit les étapes suivantes :

1. Alimenter le capteur depuis la carte Arduino en **0V** (broche **GND** du capteur) et **5V** (broche **Vcc** du capteur). Cela permettra de fournir de l'alimentation au capteur

uniquement lors de la prise de mesures, prolongeant ainsi sa durée de vie.

2. La sortie du capteur (**D0** ou **A0**) sera reliée à l'entrée analogique **A0** de la carte Arduino.

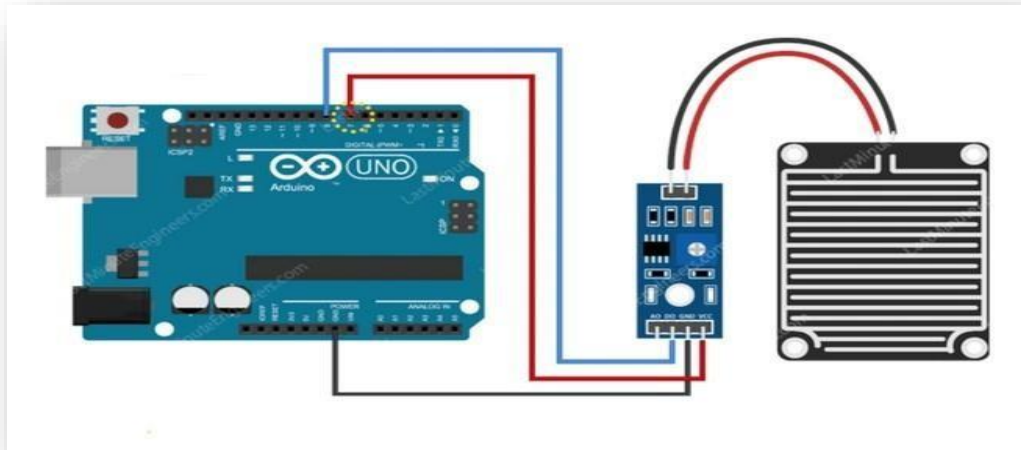


Figure II.11 : Raccordement du capteur avec l'Arduino.[21]

#### II.4.1.2 Le relais [22]

Un **relais** tel que celui de la **Figure II.12** est un interrupteur à commande électrique qui permet d'ouvrir ou de fermer un circuit de manière électromécanique ou électronique. Il est utilisé pour contrôler un circuit à l'aide d'un autre circuit, souvent avec un signal de faible puissance.

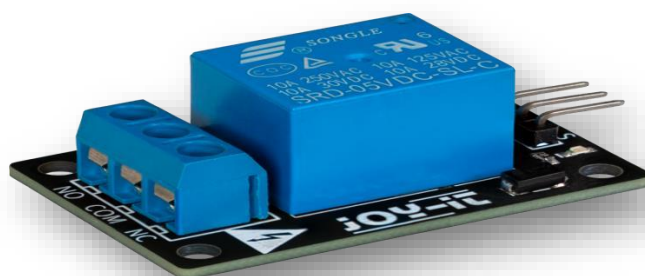


Figure II.12 : KY-019 Module relais 5V [22]

#### ✚ Principe de fonctionnement d'un relais :

Lorsqu'un courant électrique traverse la **bobine** du relais, un **champ magnétique** est généré. Ce champ attire une **armature mobile**, qui provoque le déplacement des **contacts** du

relais.

- Si le contact est **normalement ouvert (NO)**, il se **ferme** sous l'effet du courant.
- Si le contact est **normalement fermé (NF)**, il s'**ouvre** sous l'effet du courant.

Ce mécanisme permet de **commuter** un ou plusieurs circuits, souvent pour activer ou désactiver des charges de forte puissance à l'aide de commandes de faible puissance.

#### ✚ **Connectique du relais :**

Le relais dans notre projet permet de contrôler un élément externe, comme une LED ou un moteur. Il agit comme un interrupteur électronique. Sa connectique est la suivante :

- **Entrée (IN)** du relais à la **broche D7** de l'Arduino
- **VCC et GND du relais** respectivement **5V et GND** de l'Arduino
- **Sortie du relais** (commutation) : connectée à une LED et un Moteur DC.

#### **II.4.1.3 Diode électroluminescente (LED) [23]**

Une diode électroluminescente (LED) (**Figure II.13**) est un dispositif optoélectronique capable d'émettre de la lumière lorsqu' il est parcouru par un courant électrique. Comme une diode classique la LED laisse passer le courant dans un sens (polarisation direct) et le bloque dans le sens inverse (polarisation inverse). Ainsi, chaque LED possède une tension de seuil propre à elle [2].



**Figure II.13** Diodes électroluminescentes (Rouge, Jaune, Vert) [20]

✚ Principales caractéristiques d'une LED (Tableau II.5) :

<b>Directivité standard ronde de</b>	5 mm
<b>Eproxy résistant aux</b>	UV
<b>Courant direct (IF)</b>	30mA
<b>Tension directe (VF)</b>	1,8 V à 2,4 V
<b>Tension inverse</b>	5V
<b>Température de fonctionnement</b>	-30 ° à + 85 °
<b>Température de stockage</b>	-40 ° à + 100 °
<b>Intensité lumineuse</b>	20 mcd

**Tableau II.5** : Caractéristiques d'une LED [20]

#### **II.4.1.4 Le moteur DC (5V) [26] :**

Le moteur 5V DC est un dispositif électromécanique compact et puissant conçu pour fonctionner avec une tension d'alimentation de 5 volts en courant continu (**Figure II.14**). Ce moteur est de petite taille, ce qui le rend idéal pour une utilisation dans des espaces restreints et malgré sa petite taille, il peut générer une puissance significative, le rendant adapté à une variété de tâches mécaniques. Ce type de moteur offre une grande polyvalence pour générer un mouvement mécanique contrôlé et peut être intégré dans une variété d'applications électroniques et robotiques, de projets d'automatisation ou de modélisme.



**Figure II.14** : Moteur DC (5V) [26]

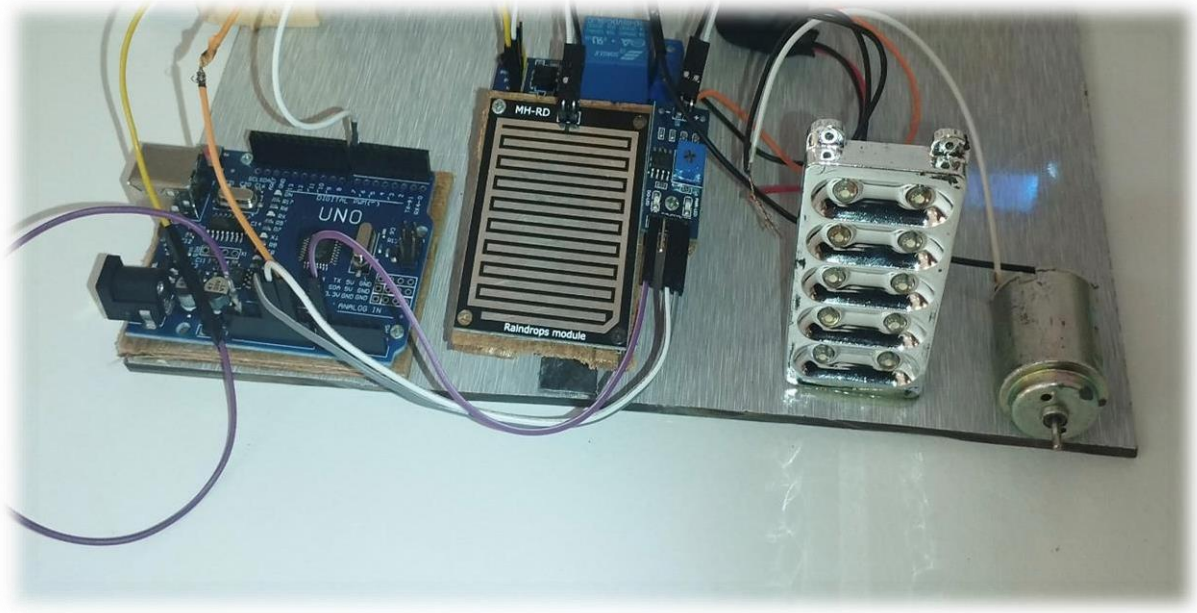
 **Caractéristiques d'un moteur DC 5V (Tableau II.6) :**

<b>Tension d'alimentation</b>	5V DC (courant continu)
<b>Charge nominale :</b>	10 g * cm
<b>Courant sans charge</b>	70 mA max
<b>Vitesse à vide</b>	9100 ± 1800
<b>Courant en charge</b>	250 mA max
<b>Vitesse en charge</b>	4500 ± 1500
<b>Couple de démarrage</b>	20 g * cm
<b>Tension de démarrage</b>	2.0
<b>Décrochage</b>	500mA max
<b>Taille du corps</b>	27.5mm x 20mm x 15mm
<b>Taille de l'arbre</b>	Diamètre x 2mm 8mm
<b>Poids</b>	17,5 grammes

**Tableau II.6 :** Caractéristiques d'un moteur DC 5V.[26]

#### **II.4.2 Réalisation matérielle :**

Nous avons présenté dans les sections précédentes les composants de notre détecteur de pluie à relai, en décrivant leur fonctionnement. Il nous ne reste donc que de présenter une photo de la réalisation de notre prototype au laboratoire, comme le montre la **Figure II.15** :



**Figure II.15** : Photo réelle du prototype réalisé : détecteur de pluie à relais.

## **II. 5 Conclusion :**

Cette mise en œuvre illustre de manière concrète comment un système simple, basé sur une carte Arduino UNO et un capteur de pluie résistif, peut permettre la détection automatique de la pluie et l'activation d'un dispositif d'alerte. Ce prototype démontre ainsi l'efficacité et la facilité d'intégration de solutions électroniques pour répondre à des besoins de surveillance environnementale.

**CHAPITRE 03 :**  
**EVALUATION DES**  
**PERFORMANCES DU**  
**PROTOTYPE : DÉTECTEUR**  
**DE PLUIE à RELAIS**

---

### **III.1 Introduction :**

Le but de ce chapitre est d'évaluer les performances du système capable de détecter la présence de pluie que nous avons réalisé dans le second chapitre. On rappelle que notre prototype est constitué d'un capteur de pluie résistif RD-MH, une carte Arduino UNO, un relais et une LED et un moteur. Le capteur déclenche le relais en cas de pluie, ce qui active le moteur et allume la LED.

On lancera alors dans un premier temps les grands axes d'utilisation d'un tel dispositif pour démontrer son utilité. On présentera par la suite, l'amélioration que nous avons adopté pour optimiser notre prototype, suivi de deux tests : au laboratoire et à l'extérieur.

A la fin de ce chapitre on propose quelques solutions pour l'optimisation et l'amélioration de notre prototype tout en respectant le compromis industriel basique.

### **III.2 Utilisation du détecteur de pluie dans les systèmes automatisés :**

Le détecteur de pluie est un dispositif conçu pour détecter la présence de précipitations et déclencher des actions appropriées afin d'optimiser le confort et la sécurité. Dans le domaine automobile, il permet l'activation et le réglage automatique des essuie-glaces. En domotique, il peut servir à contrôler l'arrosage ou à fermer automatiquement les fenêtres.

#### **III.2.1 Commande automatique de l'irrigation en agriculture à l'aide d'un capteur de pluie :**

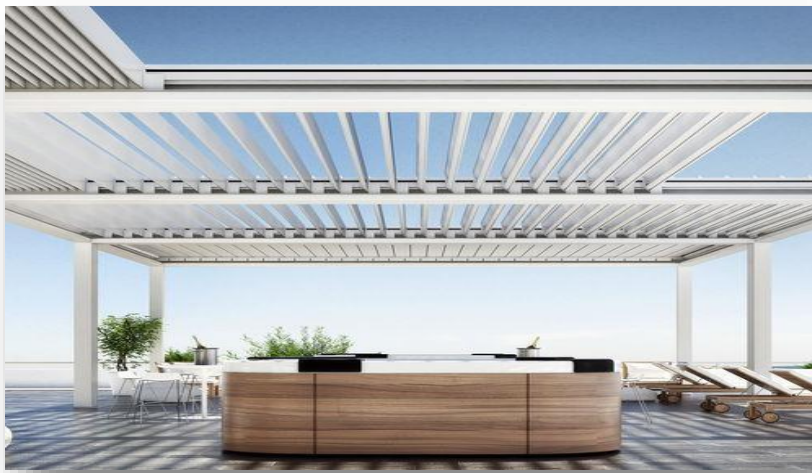
L'utilisation du détecteur de pluie permet de gérer automatiquement l'irrigation en arrêtant le système en cas de précipitations (**Figure III.1**). Cela évite un arrosage inutile, économise l'eau et protège les cultures contre l'excès d'humidité. Ce procédé favorise une agriculture plus efficace et durable.



**Figure III.1** : système d'irrigation des champs [27]

### **III.2.2 Utilisation d'un capteur de pluie pour l'automatisation domestique :**

Dans le cadre de la domotique, le détecteur de pluie permet d'automatiser certaines tâches pour améliorer le confort et la sécurité. Par exemple, il peut fermer automatiquement les fenêtres ou les toits (Figure III.2) en cas de pluie ou interrompre le système d'arrosage extérieur. Cela évite les dégâts liés à l'eau et optimise la gestion des ressources.



**Figure III.2** : Fermeture du toit en fonction du capteur de pluie [27]

### III.2.3 Système d'essuie-glaces intelligent basé sur un capteur de pluie :

Un système d'essuie-glaces intelligent dans une voiture utilise un capteur de pluie pour détecter automatiquement les gouttes sur le pare-brise (**Figure III.3**). En fonction de l'intensité des précipitations, le système ajuste la vitesse et la fréquence des essuie-glaces sans intervention du conducteur. Cela améliore la visibilité, renforce la sécurité de conduite et augmente le confort, surtout en cas de changements soudains des conditions météorologiques.



**Figure III.3 :** Exemple de capteur de pluie pour voiture [28]

### III.3 Méthodologie d'évaluation des performances :

Pour évaluer l'efficacité du détecteur de pluie, plusieurs paramètres clés doivent être définis et mesurés :

**III.3.1 Sensibilité** : mesure la capacité du capteur à détecter de faibles niveaux d'humidité ou de petites gouttes de pluie. Un capteur performant doit réagir même à un début de précipitation.

**III.3.2 Temps de réponse** : correspond au délai entre la détection de la pluie et l'activation du relais ou du système automatisé (essuie-glaces, fermeture de fenêtres, etc.). Un temps de réponse court est essentiel pour une réaction rapide.

**III.3.3 Précision** : évalue la fiabilité du capteur en termes de détection correcte, en minimisant les fausses alertes (déclenchements sans pluie réelle) ou les non-détections.

**III.3.4 Durabilité** : concerne la capacité du capteur à fonctionner efficacement sur le long terme malgré l'exposition répétée à l'eau, à l'humidité et aux conditions climatiques. Cela inclut la résistance à la corrosion et aux dépôts.

Ces paramètres peuvent être testés à l'aide d'expériences contrôlées (environnement humide simulé, exposition prolongée à l'eau, tests de déclenchement en conditions variables, etc.) pour obtenir une évaluation complète et fiable des performances du détecteur.

### III.4 Amélioration du prototype détecteur de pluie :

Avant de procéder à l'évaluation des performances de notre prototype détecteur de pluie, il s'avérait commode de l'améliorer en rajoutant un système de surveillance des niveaux de température et d'humidité, qui va accompagner le mécanisme de détection de pluie. Cette amélioration se présente par l'intégration d'un capteur de température et humidité ambiante (le DHT 22) accompagné d'un afficheur LCD, comme suit :

#### III.4.1 Intégration d'un capteur de température et humidité ambiante DHT 22 :

Cela permettrait de corrélérer les données de pluie avec les conditions environnementales et d'améliorer la prise de décision (exemple : ignorer certaines détections à basse température où la rosée est fréquente).



**Figure III.4 :** capteur de température et humidité DHT 22 [29]

Le DHT 22 (**Figure III.4**) est le capteur de température et d'humidité le plus précis. Il offre un rang de détection entre  $-40^{\circ}\text{C}$  et  $+80^{\circ}\text{C}$  et un rang en humidité situé entre 0 et 100%, le rendant bien plus versatile que le DHT11. Son signal est très stable sur le long terme. Son temps de réponse extrêmement court, sa protection contre les interférences, son encombrement réduit, sa très basse consommation et son coût bas font de ce capteur une référence pour les projets de domotique ou de monitoring environnemental nécessitant une fourchette de mesure très large. L'interfaçage série du capteur DHT22 utilise seulement 3 câbles seulement, [(-), (+) et out (signal)] rend l'utilisation de ce module facile.[29]

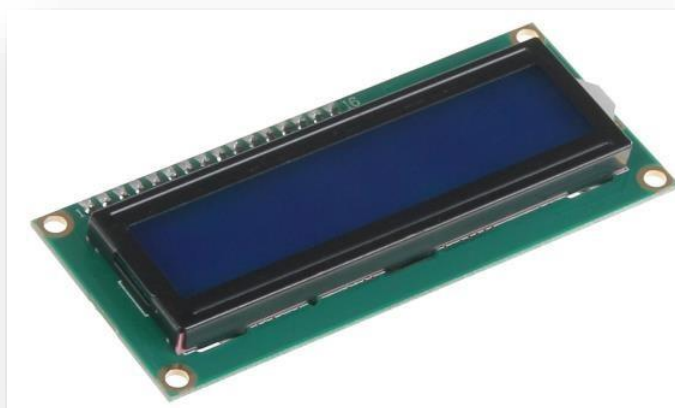
**1. Caractéristiques du capteur DHT22 (Tableau III.1) :**

<b>Tension d'alimentation</b>	3.3 à 5,5V DC
<b>Connectivité : signal digital Bus</b>	3 broches
<b>Champ de mesure d'humidité</b>	0% à 100%
<b>Champ de mesure de Température</b>	-40 à +80 °C
<b>Précision</b>	Humidité + -2%, Température + - 0,5 °C
<b>Résolution</b>	Humidité 0.1%, Température 0,1 °C
<b>Dimensions</b>	28,2 mm x 13.1 mm x 10 mm

**Tableau III.1** : Caractéristiques du capteur DHT22 [29]

**III.4.2 Intégration d'un écran LCD (16\*2) :**

Cela permettrait d'afficher en temps réel les informations collectées par capteur DHT 22, telles que la température, l'humidité. Nous pouvant ainsi suivre l'évolution des données environnementales sans avoir recours à un ordinateur, ce qui améliore l'autonomie et l'ergonomie de notre prototype.



**Figure III.5** : Ecran LCD [31]

L'écran à cristaux liquides ou LCD (**Figure III.5**) permet de la création d'un écran plat à faible consommation d'électricité. Aujourd'hui, ces écrans sont utilisés dans les affichages électroniques et ils sont devenus indispensables dans les systèmes techniques. Plusieurs types d'afficheurs LCD sont

disponibles sur le marché. Ils se différencient par leurs caractéristiques techniques, leurs tensions de fonctionnement, ainsi que leurs dimensions.

## 2. Caractéristiques d'un écran LCD [31]:

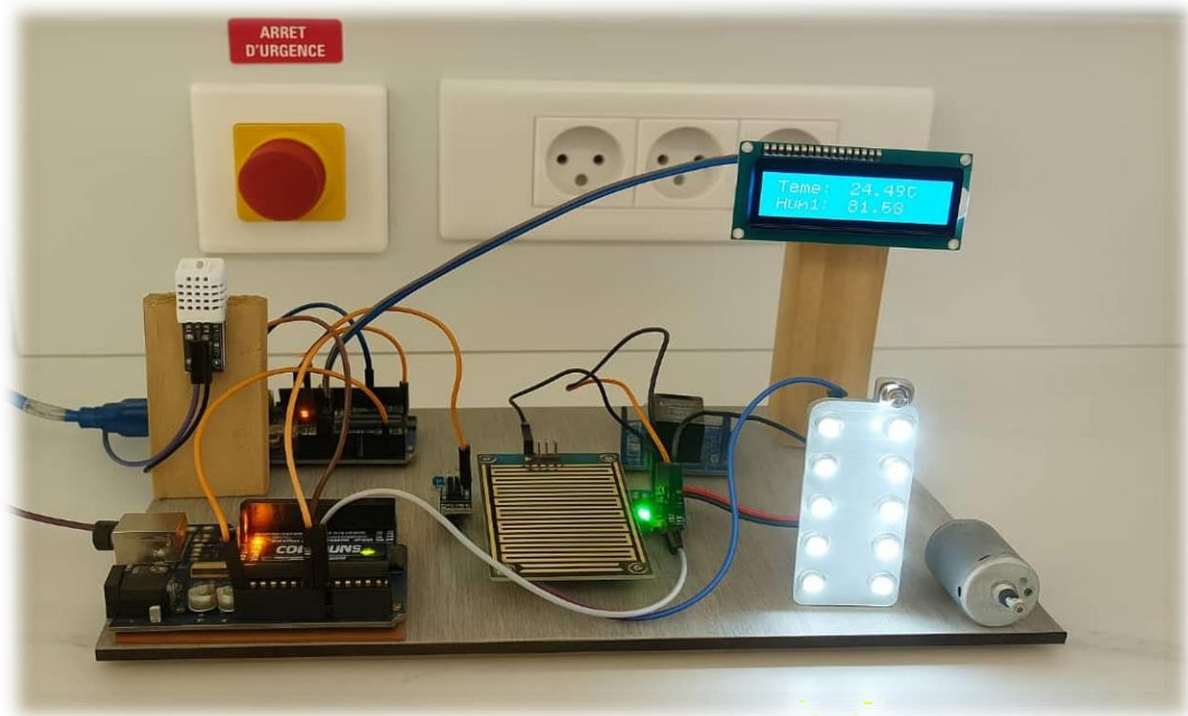
- **Alimentation** :5 V
- **Interface** :I2C (adresse 0x27)
- **Caractères blancs sur fond bleu**
- **Contraste ajustable via potentiomètre**
- **Dimensions** :80 x 38 x 18 mm

### III.4.3 Intégration avec Arduino :

Le capteur DHT22 et l'écran LCD peuvent être facilement intégrés avec une deuxième carte Arduino grâce à des bibliothèques disponibles. Le DHT22 fournit des données numériques de température et d'humidité via une seule broche, tandis que l'écran LCD (par exemple 16x2 avec interface I2C) permet d'afficher ces données en temps réel. Cette combinaison permet une visualisation directe et améliore l'interactivité du système, lors des différents tests.

### III.4.4 Réalisation du Prototype détecteur de pluie amélioré :

La photo de la **Figure III.6** (prise au laboratoire d'ELN-TLC de l'université d'Ain Temouchent) représente notre prototype détecteur de pluie amélioré :



**Figure III.6** : Prototype détecteur de pluie amélioré, au laboratoire.

La simulation de ce prototype ainsi que son code Arduino associé sont en **annexe 2**.

### **III.5 Protocole de Test :**

Après avoir réalisé et testé la fonctionnalité de notre prototype (**Figure III.6**) au laboratoire, nous avons procédé à la réalisation d'une série de tests (au laboratoire et en extérieur) pour évaluer les performances du détecteur de pluie. Le protocole de test est alors résumé dans les points suivants :

1. Effectuer un nombre des tests variable selon le type de l'expérience,
2. Collecter les données (mesures) dans des tableaux,
3. Tracer d'éventuel graphiques.

#### **III.5.1 Tests au laboratoire :**

Au laboratoire nous avons simulé la présence de pluie par le versement d'eau sur le capteur de pluie (MH-RD). Nous avons pu alors effectuer quelques mesures, tel que le temps de réponse et le seuil de déclenchement, ainsi que tester les performances de notre prototype à des taux d'humidité élevés.

##### **III.5.1.1 Temps de réponse :**

Nous avons simulé l'intensité de pluie en appliquant de l'eau sur le capteur à des intervalles réguliers. A chaque test (pratiquement 20 tests), nous avons calculé le temps de réponse en mesurant l'intervalle entre le début de la pluie et le déclenchement du système. Ce temps de réponse est en moyenne égal à **30 ms**.

##### **III.5.1.2 Seuil de déclenchement**

Nous avons simulé différentes intensités de pluie en appliquant de l'eau sur le capteur à des intervalles réguliers. Nous avons constaté que le capteur est suffisamment sensible pour détecter de faibles quantités d'eau.

Lors de chaque test, nous avons noté manuellement les informations suivantes : l'intensité de pluie (simulée au laboratoire par un versement de quantités d'eau mesuré au préalable par une seringue graduée), ainsi que l'état du relais (activé ou non).

Les résultats sont collectés dans le **Tableau III.2** :

Quantité de pluie (ml)	Détection (Oui/Non)
5	Non
10	Non
12	Non
15	Oui

**Tableau III.2** : la réponse du capteur à différentes quantités de pluie.

Nous pouvons clairement remarquer que le capteur commence à réagir efficacement à partir de **15ml** de quantité de pluie.

### III.5.1.1 Taux de fausses alarmes :

Au cours de ce test, nous voulions examiner les performances du capteur de pluie, à différents niveaux d'humidité. Nous avons essayé dans un premier de créer un milieu humide (avec les moyens de bord !!). Pour effectuer le test, nous avons procédé comme suit :

- Maintenir une valeur de taux d'humidité,
- Effectué 10 tests pour cette valeur,
- Et comptabilisé le nombre de fausses alertes de détection sur ces 10 essais.

En procédant ainsi, nous avons collecté les résultats dans le **Tableau III.3**:

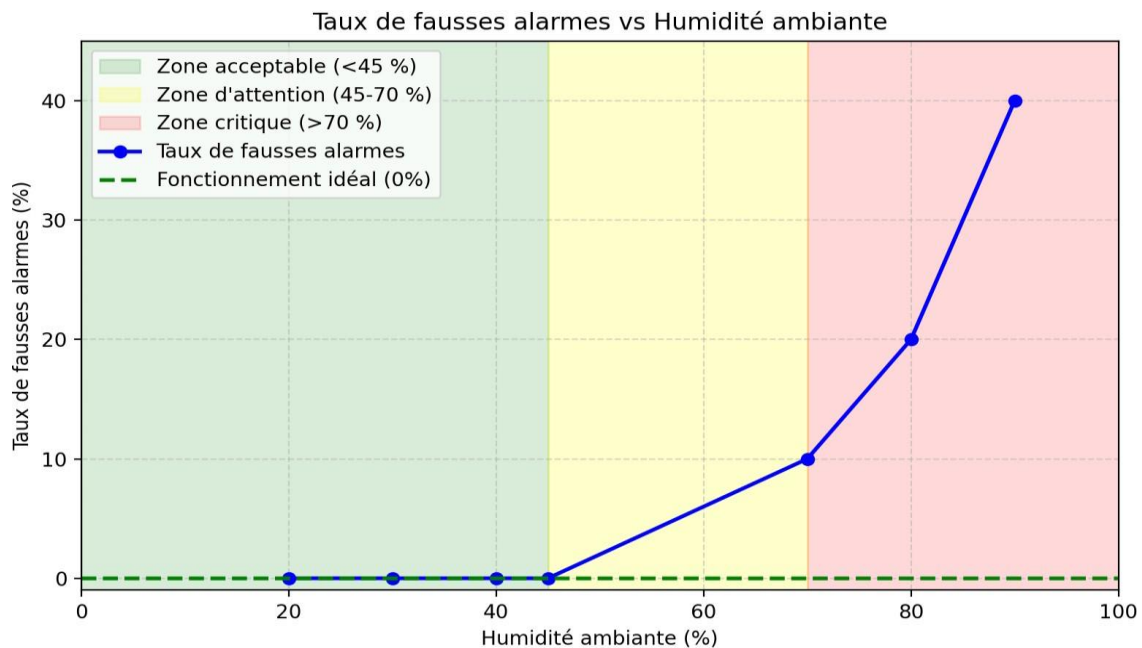
Humidité ambiante (%)	Nombre de tests	Fausses alarmes détectées	Taux de fausses alarmes (%)
20	10	0	0
30	10	0	0
40	10	0	0
45	10	0	0
70	10	1	10
80	10	2	20
90	10	4	40

**Tableau III.3** : Données collectées lors des différents tests du détecteur de pluie.

Le taux de fausses alarmes se calcule par la formule mathématique suivante :

$$\text{Taux de fausses alarmes (\%)} = (\text{Fausses alarmes détectées} / \text{Nombre de tests}) * 100$$

Dans la courbe de la **Figure III.7** (dessiné à l'aide de python), on représente à partir du **Tableau III.3**, le taux de fausses alarmes en fonction de l'humidité :



**Figure III.7** : Courbe du taux de fausses alarmes en fonction de l'humidité ambiante

Nous pouvons clairement remarquer que tant que l'humidité reste sous 70 %, le prototype affiche de bonnes performances. Au-delà, sa fiabilité chute. Cela indique que la condensation devient proche de celle d'une goutte de pluie et par conséquent le capteur n'arrive plus à les distinguer.

### III.5.2 Tests en extérieur :

Nous avons installé notre prototype dans un environnement naturel, après avoir protégé le circuit, sauf les capteurs (de pluie MH-RD et le DHT22). Le but de ce test est d'observer son comportement face à de véritables intempéries. Au cours des premiers essais, le capteur s'est montré précis dans la détection, avec peu de fausses alertes, et a démontré une bonne résistance aux conditions climatiques.



**Figure III.8 :** Test du prototype de détection de pluie en extérieur.

Pour les essais en extérieur, nous avons également relevé les performances du prototype face aux conditions météorologiques telles que l'humidité, la température.

### **III.5.2.1 Sensibilité du capteur à l'humidité :**

Nous avons mesuré la réaction du capteur à différents niveaux d'humidité présente en air.

Dans le **Tableau III.4** sont collectés les résultats obtenus :

<b>Niveau d'humidité (%)</b>	<b>Détection (Oui/Non)</b>
<b>10%</b>	Non
<b>20%</b>	Non
<b>60%</b>	Non
<b>80%</b>	Oui

**Tableau III.4 :** Réponse du capteur à différents niveaux d'humidité

Il est clair que le test à l'extérieur présente presque le même comportement face à un taux d'humidité au-delà de 60%.

### III.5.2.2 Sensibilité du capteur à la température :

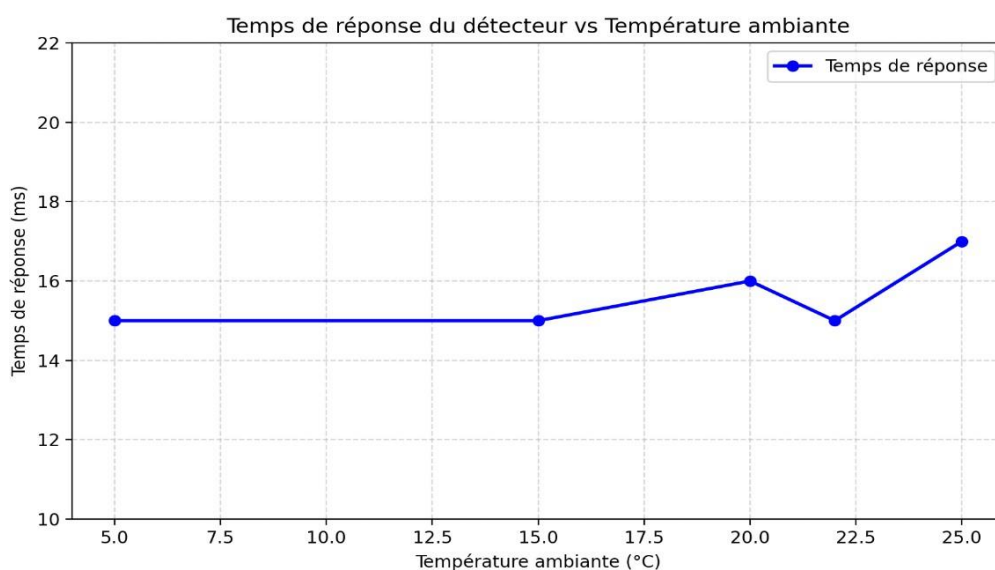
Pour évaluer la sensibilité du capteur de pluie face à l'augmentation légère de température, nous avons mesuré le temps de réponse du prototype pour quelques valeurs de température ambiante (la variation de la température étant faible aux mois pluvieux).

Les données collectées sont récapitulées dans le **Tableau III.5** :

Température ambiante (°C)	Détection de la pluie	Temps de réponse (ms)	Remarques
5 °C	Oui	15	Fonctionnement stable
15 °C	Oui	15	Aucun changement notable
20 °C	Oui	16	Conditions normales
22 °C	Oui	15	Léger gain de rapidité, fonctionnement normal
25 °C	Oui	17	Pas d'effet négatif observé

**Tableau III.5** : Influence de la température sur le fonctionnement du capteur.

A partir du **Tableau III.5**, nous avons tracé la courbe en python: temps de réponse du détecteur de pluie en fonction de la température (**Figure III.9**).



**Figure III.9** : Courbe illustrant l'effet de la température sur le détecteur.

Nous pouvons constater que le prototype conserve des performances rapides et constantes dans les conditions de température testées. Les variations légères de température ambiante n'ont eu aucun effet sur les performances du capteur, qui est resté stable et fiable dans toutes les conditions.

### III.5.3 Comparaison de précision Laboratoire/Extérieur :

En comparant les détections avec les conditions réelles (en laboratoire et en extérieur) en fixant le nombre de test à **10** et maintenir le taux d'humidité à **85%**, nous avons obtenu les résultats suivants (**Tableau III.6**):

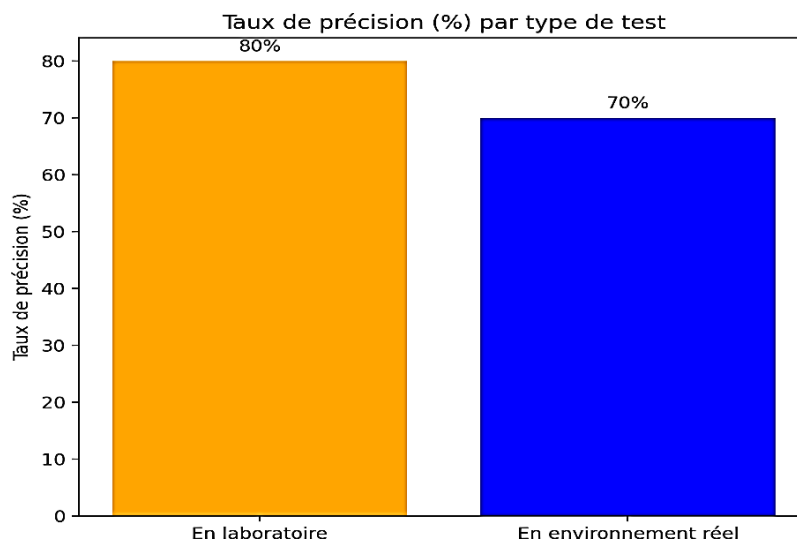
Type de test	Nombre total de détections	Fausses alertes	Taux de précision
<b>En laboratoire</b>	10	2	80 %
<b>En environnement réel</b>	10	3	70%

**Tableau III.6** : Statistiques sur la fréquence des fausses détections en laboratoire et à l'extérieur

Le taux de précision est calculé par la formule suivante :

$$\text{Taux de précision} = \left( \frac{\text{Nombre total de détections} - \text{Fausses alertes}}{\text{Nombre total de détections}} \right) * 100$$

Nous avons tracé un histogramme du taux de précision au laboratoire et à l'extérieur. Nous avons obtenu la **Figure III.10**:



**Figure III.10** : Histogramme illustrant une comparaison de précision : laboratoire/extérieur.

Ce dernier test démontre que la précision du détecteur face à des valeurs élevées d'humidité au laboratoire est quasiment proche de celle à l'extérieur. Une légère dégradation de précision en

extérieur s'interprète principalement par les conditions réelles qui introduisent du bruit et des variations sur le capteur de pluie.

### **III.6 Analyse des Résultats :**

Les résultats obtenus lors des tests ont permis de confirmer des performances globales satisfaisantes du capteur de pluie :

#### **III.6.1 Sensibilité :**

Le capteur a montré une bonne capacité à détecter des niveaux faibles d'intensité de pluie, ce qui est conforme aux attentes. Il réagit rapidement dès les premières gouttes (**30 ml**), ce qui le rend adapté à des applications sensibles comme l'irrigation ou les systèmes d'essuie-glaces intelligents.

#### **III.6.2 Stabilité du relais :**

Le relais associé au capteur s'est montré stable et fiable, avec aucun déclenchement intempestif pendant les phases de test. Le système maintient son état tant que la pluie persiste, et revient correctement à son état initial après l'arrêt des précipitations.

#### **III.6.3 Résistance et durabilité :**

Après une exposition prolongée à l'humidité, à la poussière et à des cycles répétés de pluie simulée et réelle, le capteur a conservé son bon fonctionnement. Aucun signe de corrosion ni de dysfonctionnement n'a été observé durant 30 jours de test. Alors, le capteur présente une bonne robustesse pour une utilisation extérieure de moyenne durée.

#### **III.6.4 Performances face à la température :**

Le capteur offre un bon temps de réponse ( $\approx 15\text{ms}$ ), une précision élevée ( $>80\%$ ), aux légères variations de température. Cela confirme sa robustesse pour une utilisation à long terme, en environnement extérieur durant les mois pluvieux.

#### **III.6.5 Limites observées :**

Quelques fausses alertes ont été enregistrées en extérieur, probablement dues à la condensation due aux fortes valeurs de taux d'humidité (au-delà de 70%).

#### **III.6.6 Comparaison aux attentes théoriques :**

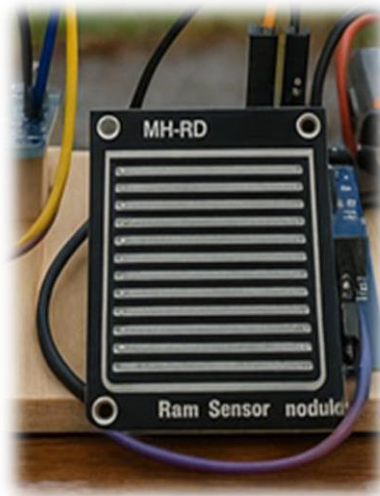
Les performances réelles du capteur sont globalement en accord avec les spécifications théoriques prévues pour ce type de dispositif. Le bon comportement en température et la faible consommation d'énergie confirment aussi sa compatibilité avec des systèmes embarqués ou autonomes.

### III.7 Optimisations Possibles :

À partir des tests réalisés et des limites observées, plusieurs pistes d'amélioration peuvent être envisagées pour augmenter la fiabilité et l'efficacité du système, tel que :

#### III.7.1 Installation optimisée du détecteur de pluie :

Pour garantir une détection plus précise et éviter les erreurs de fonctionnement, nous avons installé le capteur de pluie avec une légère inclinaison (**Figure III.11**). Cette disposition permet un écoulement naturel de l'eau de pluie, réduisant ainsi l'accumulation d'humidité sur la surface du capteur et limitant le risque de fausses détections après les précipitations.



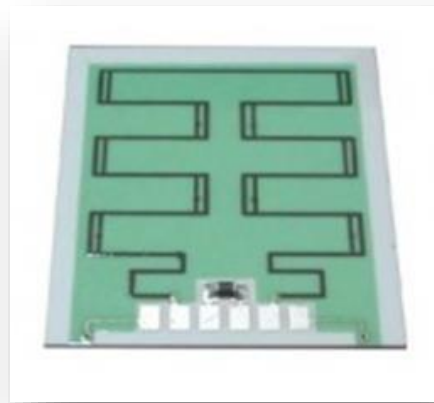
**Figure III.11 : Installation optimisée du détecteur de pluie : capteur en position inclinée.**

#### III.7.2 Seuil de déclenchement ajustable :

Au cours des tests, nous avons observé qu'avec un seuil bas la sensibilité du détecteur augmente mais peut provoquer davantage de fausses alertes. Il suffit alors d'augmenter ce seuil au prix d'une réactivité moindre au début de la pluie. Pour ce faire, nous proposons d'ajouter un potentiomètre ou une commande logicielle permettant de régler manuellement ou automatiquement le seuil d'humidité nécessaire pour activer le relais.

#### III.7.3 Remplacement du capteur de pluie résistif :

Puisque nous avons observés que le taux de fausses alarmes augmente avec des taux d'humidité élevés, il sera privilégié de remplacer un capteur de type résistif par un capteur capacitif, ce qui permet une meilleure sensibilité à l'humidité et une plus grande résistance aux interférences. Parmi les capteurs capacitifs disponibles, le capteur de pluie IBR274 (**Figure III.12**) :



**Figure III.12 : Capteur de pluie capacitif IBR274 [30]**

Le capteur "IBR274" est une platine sur substrat céramique destinée à la réalisation de détecteur de pluie à détection capacitive. Ainsi l'accumulation de pluie sur la surface de la plaque aura pour conséquence de modifier la valeur de sa capacité. Il possède une mini résistance chauffante au dos pour éliminer la condensation et la rosée, ce qui permet de s'affranchir des possibilités de détections intempestives.[30]

Ces caractéristiques sont comme suit [30]:

- Mode de détection** : capacitif
- **Technologie** : Substrat céramique (AL<sub>2</sub>O<sub>3</sub>)
- **Capacité nominale** : 105 pF (10 %)
- **Résistance chauffante** : 42 ohms (10 %)
- **NTC** : 1 Kohm / 25 °C

#### **III.7.4 Double capteur pour validation croisée :**

Utiliser deux capteurs disposés à des endroits différents permettrait d'activer le système seulement si les deux détectent la pluie. Cela minimise le risque d'erreur causé par un événement localisé (goutte isolée, insecte, poussière, etc.).

#### **III.7.5 Algorithme de temporisation intelligente :**

Mettre en place un délai de validation (par exemple 3 secondes) avant activation du relais, pour s'assurer que la pluie est bien continue et non accidentelle. Cela permet d'éviter les activations inutiles.

### **III.8 Conclusion :**

Ce projet a permis de concevoir, tester et évaluer un détecteur de pluie destiné à des d'essuie-glaces intelligents, stations météo intelligents, domotique, automobile....etc. Grâce aux différentes méthodologies d'évaluation mises en œuvre, nous avons pu mesurer des performances satisfaisantes en termes de sensibilité, de temps de réponse, de précision et de durabilité.

Les résultats ont démontré que le capteur détecte efficacement les précipitations même à faibles intensités, avec un temps de réaction rapide d'environ **15 ms**, et une précision à **70 %** en conditions réelles. De plus, le capteur a fait preuve d'une bonne résistance face aux variations climatiques, notamment aux changements de température, sans perte de performance.

Ce travail met en évidence la robustesse et la fiabilité du système, tout en soulignant quelques limites, notamment la possibilité de fausses alertes en cas de condensation due aux forts taux d'humidité. Ces observations ouvrent la voie à des optimisations futures visant à améliorer davantage la précision et la stabilité du détecteur :

#### **Perspectives :**

Afin d'améliorer les performances et les fonctionnalités du système, plusieurs axes de développement peuvent être envisagés:

##### **1. Amélioration de la durabilité :**

Une étude pourrait être menée sur l'ajout d'un revêtement protecteur pour le capteur, afin de le rendre plus résistant aux conditions climatiques extrêmes et d'allonger sa durée de vie.

##### **2. Intégration IoT :**

Il serait pertinent d'explorer l'intégration d'une connectivité sans fil (comme le Wi-Fi, le Bluetooth ou le LoRa) pour permettre une gestion à distance du capteur via une application mobile ou une interface web, offrant ainsi des fonctionnalités telles que l'alerte en temps réel ou l'enregistrement des données.

##### **3. Filtre anti-ruissellement :**

Intégrer un filtrage logiciel ou électronique pour distinguer une vraie pluie d'un simple ruissellement ou de la condensation (rosée, éclaboussures). Cela permettrait de réduire significativement les fausses alertes, surtout en extérieur.

##### **4. Détection de pluie plus sophistiquée :**

L'utilisation de capteurs plus avancés, comme des capteurs optiques ou des capteurs à infrarouges, pourrait être étudiée afin d'améliorer la précision de la détection et d'élargir les conditions de détection (intensité de la pluie, humidité ambiante, etc.).

**5. Ajout d'un capteur secondaire :**

Intégrer un capteur secondaire pour permettre une interprétation croisée des données, ce qui pourrait transformer le système en capteur intelligent, capable de distinguer différentes conditions environnementales avec plus de précision.

# CONCLUSION GÉNÉRALE



Dans un contexte où l'automatisation devient un enjeu fondamental pour améliorer la gestion des ressources et l'efficacité énergétique, ce projet s'est inscrit dans la conception et l'évaluation d'un détecteur de pluie simple, économique et compatible avec les microcontrôleurs Arduino. Destiné à des applications dans les domaines de la domotique et de l'agriculture, ce système vise à réagir automatiquement à la présence de pluie pour déclencher un relais qui active un moteur, permettant par exemple la fermeture automatique de fenêtres ou l'arrêt d'un système d'irrigation.

Tout au long de ce travail, nous avons commencé par une étude des différents types de capteurs de pluie, en mettant l'accent sur la technologie résistive pour sa simplicité, sa disponibilité et son faible coût. Malgré ses limites, cette technologie s'est révélée adaptée à une intégration avec Arduino, notamment grâce à la facilité de programmation et la flexibilité offerte par cette plateforme.

La simulation du système sur Proteus, suivie de la réalisation physique du prototype, nous a permis de tester la fonctionnalité complète du détecteur : lecture des données du capteur, détection de seuil, activation du relais, et en option, signalisation par LED ou actionner un moteur. Le code Arduino développé a été documenté et optimisé pour offrir un déclenchement fiable à partir d'un certain seuil d'humidité.

Ce projet bien qu'il soit simple, nous a été très profitable. En effet, il nous a permis d'enrichir notre formation tels que l'utilisation d'ARDUINO et la manipulation de son code, le contrôle des différents composants électroniques, ainsi que l'amélioration du prototype réalisé.

L'évaluation expérimentale des performances du détecteur a mis en lumière plusieurs éléments clés :

- ✓ Une bonne sensibilité du capteur résistif à la présence d'eau.
- ✓ Un temps de réponse rapide, généralement inférieur à une seconde.
- ✓ Une précision acceptable, bien que des fausses alarmes puissent survenir en raison de l'humidité ambiante ou de salissures sur le capteur.
- ✓ Une durabilité moyenne, le capteur étant sensible à la corrosion et nécessitant un entretien régulier.

En somme, le système conçu atteint les objectifs fixés en début de projet : il est fonctionnel, abordable, et facilement intégrable dans des systèmes automatisés simples. Cependant, comme toute autre technologie, des améliorations peuvent encore être apportées.

## Les références Bibliographiques

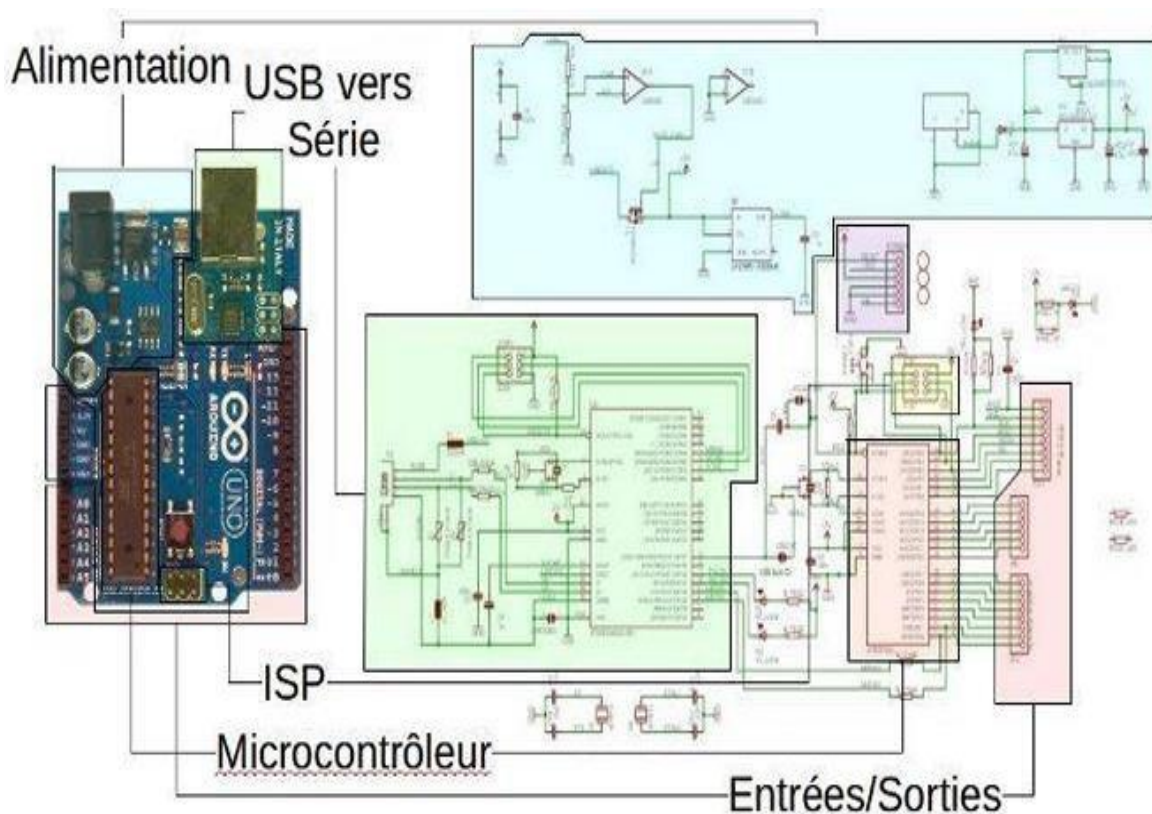
- [1] Fraden, J. (2016). Flow sensors. In *Handbook of modern sensors: Physics, designs, and applications* (pp. 453–484). Springer
- [2] Asch, G. (1998). *Sensors in industrial instrumentation: Les capteurs en instrumentation industrielle* [Bilingual edition]. Hermes Science Publications.
- [3] Asch, G., & al. (2003). *Les capteurs en instrumentation industrielle* (6<sup>e</sup> éd.). Dunod..
- [4] <https://elearning.enporan.dz/moodle/mod/resource/view.php?id=5640&redirect=1>
- [5] <http://dptgeii.iutsd.univ-lorraine.fr/cours/lpsarii/IM/Cours/Capteurs.pdf>
- [6] Correvon, M. (2012). *Capteurs inductifs de position, conditionnement et traitement du signal* [Course material]. CD\SEM\Cours\Chap09.
- [7] Ferguson, T. (2001). Tutorial : Mesurer des températures par thermocouples. *National Instruments, Note d'application, 43*.
- [7] Fraden, J. (2016). *Handbook of Modern Sensors*, Springer
- [8] Wilson, J. (2004). *Sensor technology handbook*. Elsevier.
- [9] [https://si.lycee-desfontaines.eu/IMG/pdf/capteurs\\_cr.pdf](https://si.lycee-desfontaines.eu/IMG/pdf/capteurs_cr.pdf)
- [10] <https://www.guilcor.fr/content/37-le-capteur-d-humidite-resistif>
- [11] Margolis, M. (2011). *Arduino cookbook* (2nd ed.). O'Reilly Media.
- [12] Bourdon, J. (2007). *Capteurs et systèmes d'acquisition*. Dunod
- [13] Fraden, J. (2016). *Handbook of Modern Sensors*
- [14] Bentley, J. P. (2005). *Principles of measurement systems* (4th ed.). Pearson Education.
- [15] Boulefdouai, S. (2022). *Simulation et réalisation d'un détecteur de pluie piloté par une carte Arduino Uno* (Mémoire de Master, Université d'Aïn Témouchent).
- [16] <https://learn.sparkfun.com/tutorials/arduino-comparison-guide>
- [17] <https://www.moussasoft.com/comment-utiliser-capteur-de-pluie-arduino-domoti/>

- [18] <https://elearning-facsci.univ-annaba.dz/mod/resource/view.php?id=12622>
- [19] Belmeliani, A., & Rahiel, N. (2020). *Conception et réalisation d'un prototype d'une maison intelligente* (Mémoire de Master, Université d'Oran 2 Mohamed Benahmed, Institut de maintenance et de sécurité industrielle).
- [20] <http://gilles.thebault.free.fr/spip.php?article40>
- [21] <https://kwoco-plc.com/fr/electrical-relay/>
- [22] Sadi, A. (2018). *Conception et simulation d'une carte d'acquisition de données météorologique en temps réel à base d'un microcontrôleur 18F45K22* (Mémoire de fin d'études, Université Mouloud Mammeri de Tizi-Ouzou).
- [23] <https://youpilab.com/components/product/moteur-5v-dc>
- [24] <https://www.seneca.it/media/6849170/2.jpg?width=600&height=338.0281690140845>
- [25] [https://www.fermeture-online.com/330539-thickbox\\_default/ pergola-vacances-auto-porteea-toit-ouvrant-retractable.webp](https://www.fermeture-online.com/330539-thickbox_default/ pergola-vacances-auto-porteea-toit-ouvrant-retractable.webp)
- [26] Google. (n.d.). *Photo détecteur de pluie pour voiture*.  
<https://www.google.com/search?q=photo+d%C3%A9tecteur+de+pluie+pour+voiture>
- [27] [https://www.google.com/search?q=photo+d%C3%A9tecteur+de+pluie+pour+voiture&sc\\_e sv=0c57f3453f651efe&rlz=1C1FHFK\\_arDZ1147DZ1147&sxsrf=AE3TifPCLJP9OWF3n3y3PE7LQ9eR-](https://www.google.com/search?q=photo+d%C3%A9tecteur+de+pluie+pour+voiture&sc_e sv=0c57f3453f651efe&rlz=1C1FHFK_arDZ1147DZ1147&sxsrf=AE3TifPCLJP9OWF3n3y3PE7LQ9eR-)
- [28] <https://www.powertech-dz.net/products/single/capteur-temperature-et-dhumidite-dht22-vente-composants-electronique-blida-algerie-75>
- [29] [https://www.lextronic.fr/28698-large\\_default/capteur-de-pluie-ibr274.jpg](https://www.lextronic.fr/28698-large_default/capteur-de-pluie-ibr274.jpg)
- [30] [https://www.gotronic.fr/art-afficheur-lcd-2x16-caracteres-lcd16x2i2c-25650.htm?srsltid=AfmBOopia\\_Rjl\\_5su\\_xUqv0gZ7H9pQ0v3qpFAefLyBe07zBaEC7I3nfA](https://www.gotronic.fr/art-afficheur-lcd-2x16-caracteres-lcd16x2i2c-25650.htm?srsltid=AfmBOopia_Rjl_5su_xUqv0gZ7H9pQ0v3qpFAefLyBe07zBaEC7I3nfA)
- [31] Hammoumi, A., Belhaji, A., & Laghrissi, K. (2016). *Conception et réalisation d'un thermomètre numérique à base du PIC16F877A* (Mémoire de fin d'études, Institut Supérieur d'Ingénierie & des Affaires, ISGA Rabat, École d'ingénierie).
- [32] Mokeddem, F., & Bachra, M. H. (2016). *Étude et réalisation d'une seringue électrique avec une carte Arduino* (Mémoire de fin d'études, Université Abou Bekr Belkaïd de Tlemcen).

## Les annexes

### Annexe 1 : Schéma interne de la carte Arduino UNO

L'Arduino Uno est une carte à microcontrôleur programmable conçue pour piloter divers composants mécaniques tels que des systèmes, des lumières ou des moteurs. Grâce à sa simplicité d'utilisation, elle permet à tout un chacun même sans connaissances avancées en programmation de concevoir facilement des dispositifs automatisés. Cet outil a été spécialement pensé pour les inventeurs, artistes ou passionnés désireux de développer leurs propres projets électroniques et systèmes intelligents.[32]



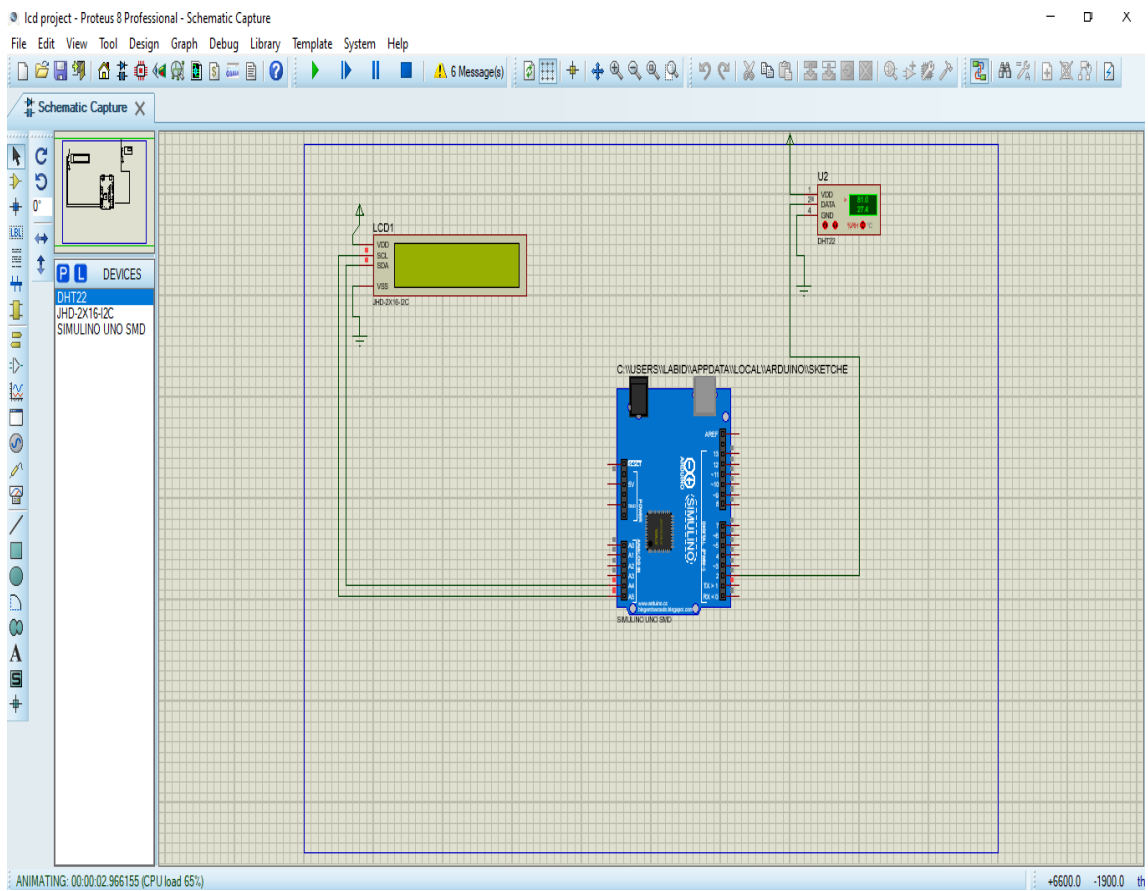
**Figure A** : Schéma interne du circuit électrique de la carte Arduino [33].

## **Annexe 2 : Simulation sur Proteus du prototype détecteur de pluie amélioré muni de son code Arduino :**

Le code Arduino pour afficher la Température et de l'humidité sur un écran LCD avec Arduino :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  lcd.init();
  lcd.backlight();
  dht.begin();
  lcd.setCursor(0, 0);
  lcd.print("Initialisation");
  delay(2000);
  lcd.clear();
}
void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  lcd.clear();
  if (isnan(temp) || isnan(hum)) {
    lcd.setCursor(0, 0);
    lcd.print("Sensor error");
```

```
} else {  
lcd.setCursor(0, 0);  
lcd.print("Temp: ");  
lcd.print(temp, 1);  
lcd.print((char)223);  
lcd.print("C");  
lcd.setCursor(0, 1);  
lcd.print("Hum: ");  
lcd.print(hum, 1);  
lcd.print(" %");  
}  
delay(2000);  
}
```



**Figure B** : Schéma de Simulation sur Proteus : Affichage de température et d'humidité sur LCD.

Le code Total du Montage complet:

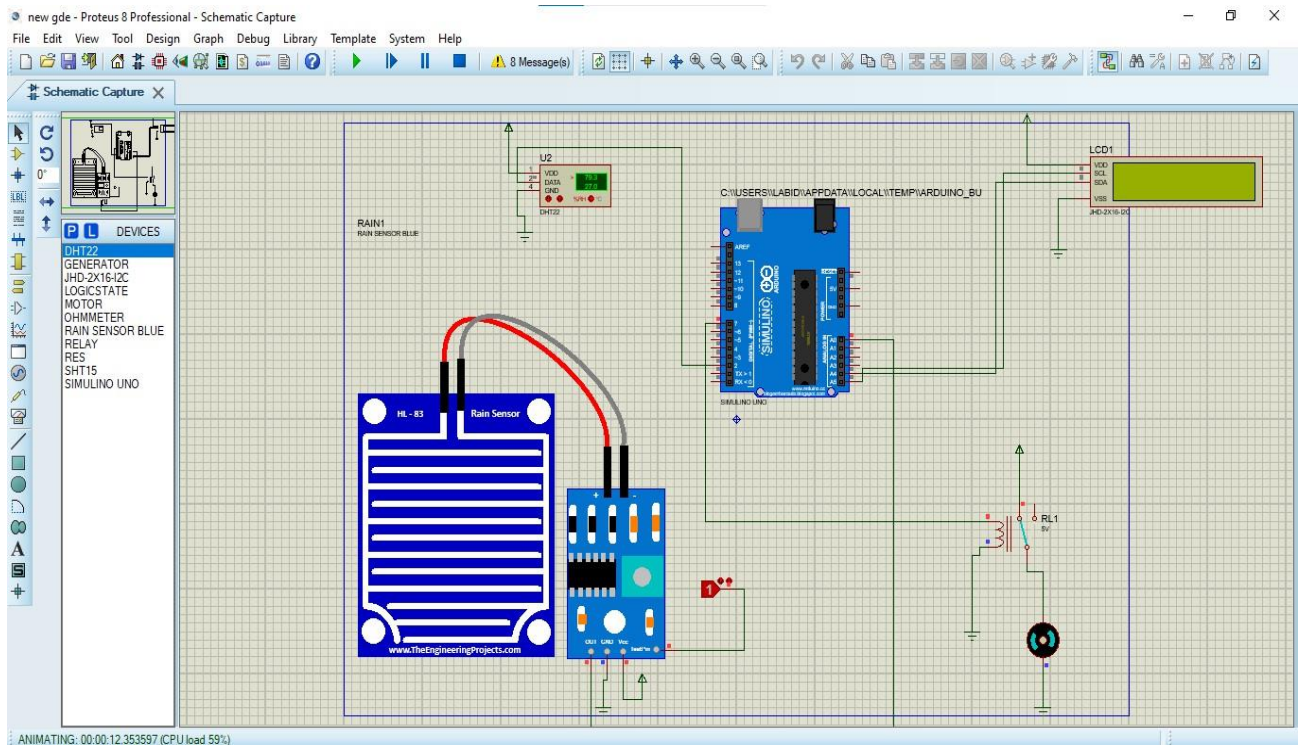
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
// DHT setup
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
// LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);
// Pins
const int rainSensorPin = A0;
const int motorPin = 7;
void setup() {
  // Init LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("System Starting...");
  delay(2000);
  lcd.clear();
  // Init DHT
  dht.begin();
```

```
// Pin modes
pinMode(rainSensorPin, INPUT);
pinMode(motorPin, OUTPUT);
digitalWrite(motorPin, LOW); // Motor OFF by default
}
void loop() {
  // Read DHT
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  // Read rain sensor
  int rainValue = analogRead(rainSensorPin); // Analog value (0 = rain, 1023 = dry)
  lcd.clear();
  // Display temp & humidity
  if (isnan(temp) || isnan(hum)) {
    lcd.setCursor(0, 0);
    lcd.print("Sensor Error");
  } else {
    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print(temp, 1);
    lcd.print((char)223);
    lcd.print("C ");
    lcd.print("H:");
    lcd.print(hum, 0);
    lcd.print("%");
  }
  // Rain detection logic
  lcd.setCursor(0, 1);
```

```

if (rainValue < 500) { // Threshold: adjust as needed
lcd.print("Rain: YES ");
digitalWrite(motorPin, HIGH); // Turn ON motor
} else {
lcd.print("Rain: NO ");
digitalWrite(motorPin, LOW); // Turn OFF motor
}
delay(2000);
}

```



**Figure C :** Schéma de Simulation du deux Montages (prototype amélioré)