

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 عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département d'Électronique et Télécommunications



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : électronique
Domaine : Sciences et Technologies
Filière : Electronique
Spécialité : Instrumentation
Thème

Conception et Réalisation d'un Système de Surveillance
Météorologique basé sur l'Internet des Objets (IoT)

Présenté Par :

- 1) Mr FRIFRA Aymen
- 2) Mr SEFIR Houcine

Devant le jury composé de :

Dr. BENZINA Amina	MCB	UAT.B.B (Ain Temouchent)	Président
Dr. GHEZALI Sanaa	MCB	UAT.B.B (Ain Temouchent)	Examineur
M ^{me} . CHAIB Amina	MAA	UAT.B.B (Ain Temouchent)	Encadrante
Dr. MERADI Abdelhafid	MCA	UAT.B.B (Ain Temouchent)	Co-Encadrant

Année Universitaire 2025/2026



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 عين تموشنت - بلحاج بوشعيب
Université Ain Témouchent- Belhadj Bouchaib -
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie



Formulaire de déclaration sur l'honneur
Relatif à l'engagement pour respecter les règles
d'authenticité scientifique dans l'élaboration d'un travail de
recherche

(Annexe de l'arrêté n°933 du 28 juillet 2016 fixant les règles relatives à la prévention et la lutte contre le plagiat)

Je soussigné,

Etudiant, M. Frifra Aymen

Détenteur d'une carte d'étudiant N° : 212137075157 , délivrée le : 2021 à Ain Témouchent

Inscrit à la faculté : **Sciences et de Technologie**

Département : **Électronique et Télécommunications**

Et chargé de préparer un mémoire de fin d'études Master 2 en **Instrumentation** intitulé :
Conception et Réalisation d'un Système de Surveillance Météorologique basé sur l'Internet des Objets (IoT)

Déclare sur l'honneur, m'engager à respecter les règles scientifiques, méthodologiques, et
les normes de déontologie professionnelle et de l'authenticité académique requise dans
l'élaboration du projet de recherche suscité.

À Ain Temouchent, le : 28/06/2026

Signature



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 عين تموشنت - بلحاج بوشعيب
Université Ain Témouchent- Belhadj Bouchaib -
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie



Formulaire de déclaration sur l'honneur
Relatif à l'engagement pour respecter les règles
d'authenticité scientifique dans l'élaboration d'un travail de
recherche

(Annexe de l'arrêté n°933 du 28 juillet 2016 fixant les règles relatives à la prévention et la lutte contre le plagiat)

Je soussigné,

Etudiant, M. **Sefir Houcine**

Détenteur d'une carte d'étudiant N° : 212137072606 , délivrée le : 2021 à Ain Témouchent

Inscrit à la faculté : **Sciences et de Technologie**

Département : **Électronique et Télécommunications**

Et chargé de préparer un mémoire de fin d'études Master 2 en **Instrumentation** intitulé :
Conception et Réalisation d'un Système de Surveillance Météorologique basé sur l'Internet des Objets (IoT)

Déclare sur l'honneur, m'engager à respecter les règles scientifiques, méthodologiques, et
les normes de déontologie professionnelle et de l'authenticité académique requise dans
l'élaboration du projet de recherche suscité.

À Ain Temouchent, le : 28/06/2026

Signature

Remercîments

Tout d'abord, Nous remercions « ALLAH » pour la force et la persévérance qui nous a donné pour achever ce travail.

Nous remercions chaleureusement notre encadrante **Mme CHAIB Amina** et notre co-encadrant **Mr MERADI Abdelhafid**, dont le soutien, les conseils avisés et l'expertise ont été essentiels tout au long de cette aventure. Votre disponibilité, votre patience et votre encouragement nous ont permis de surmonter les défis rencontrés et d'aboutir à ce résultat.

Je tiens également à remercier les membres du jury **Dr BENZINA Amina** (présidente) et **Dr GHEZALI Sanaa** (examinatrice), pour leur temps, leur intérêt et leurs précieuses critiques. Vos retours et suggestions ont grandement enrichi ce travail et m'ont permis de l'améliorer considérablement.

Enfin, nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce Projet de Fin d'Études.

Dédicaces

Je voudrais avant tout remercier ALLAH de m'avoir
donné la force et le courage d'accomplir cette humble
œuvre.

Je voudrais dédier cet humble travail à mes chers parents
Et à mes frères, à toute la famille FRIFRA et mes amis, et à
tous ceux qui m'aiment, et à tous ceux qui ont contribué
directement ou indirectement à rendre ce projet possible
pour nous. Merci

- Frifra aymen –

Je voudrais avant tout remercier ALLAH de m'avoir
donné la force et le courage d'accomplir cette humble
œuvre.

Je voudrais dédier cet humble travail à mes chers parents
Et à mes frères, à toute la famille de SEFIR et mes amis, et
à tous ceux qui m'aiment, et à tous ceux qui ont contribué
directement ou indirectement à rendre ce projet possible
pour nous. Merci

- Sefir Houcine -

« La mesure est la première étape qui mène au contrôle et éventuellement à l'amélioration. »

- H. James Harrington -

الملخص

يعرض هذا العمل تصميم وتنفيذ محطة أرصاد جوية ذكية مبنية على إنترنت الأشياء (IoT). يعتمد النظام المطور على لوحة Arduino Uno WiFi R3 وعدد من مستشعرات البيئة، ويتيح المراقبة في الوقت الحقيقي لدرجة الحرارة، والرطوبة النسبية، وشدة الإضاءة، وهطول الأمطار. يتم إرسال البيانات المجمعة إلى منصتي ThingSpeak و Blynk للعرض والتخزين والمراقبة عن بُعد. بالإضافة إلى ذلك، تم تطوير واجهة ويب أصلية باستخدام HTML/CSS/JavaScript لتوفير عرض بيانات في الوقت الحقيقي، وأرشفة بيانات دائمة، والتحقق المترولوجي التلقائي اعتمادًا على مؤشرات إحصائية. أظهرت الاختبارات التجريبية استقرارًا جيدًا للنظام، ومعدل نجاح في نقل البيانات يتجاوز 98%، واستعادة سريعة بعد انقطاعات الطاقة. تؤكد النتائج المحققة الجدوى الفنية والاقتصادية للحل المقترح لتطبيقات أكاديمية وتعليمية ورصد البيئة المحلي.

الكلمات المفتاحية: إنترنت الأشياء (IoT)، محطة أرصاد ذكية، Arduino Uno WiFi R3، واجهة ويب، ThingSpeak، Blynk، رصد بيئي، التحقق المترولوجي.

Résumé

Ce travail présente la conception et la réalisation d'une station météorologique connectée basée sur les technologies de l'Internet des Objets (IoT). Le système développé, fondé sur une carte Arduino Uno WiFi R3 et plusieurs capteurs environnementaux, permet la mesure en temps réel de la température, de l'humidité relative, de la luminosité et des précipitations. Les données acquises sont transmises vers les plateformes ThingSpeak et Blynk pour assurer leur visualisation, leur archivage et leur supervision à distance. En complément, une interface Web originale a été développée en HTML/CSS/JavaScript afin d'assurer la visualisation en temps réel, l'archivage persistant des mesures ainsi que leur validation métrologique automatique à l'aide d'indicateurs statistiques. Les essais expérimentaux ont mis en évidence une bonne stabilité du système, un taux de succès de transmission supérieur à 98 % ainsi qu'une reprise rapide après coupure d'alimentation. Les résultats obtenus confirment la faisabilité technique et économique de la solution proposée pour des applications académiques, pédagogiques et de surveillance environnementale locale.

Mots-clés : Internet des Objets (IoT), station météorologique connectée, Arduino Uno WiFi R3, interface Web, ThingSpeak, Blynk, surveillance environnementale, validation métrologique.

Abstract

This work presents the design and implementation of an Internet of Things (IoT)-based smart weather station. The developed system, built around an Arduino Uno WiFi R3 board and several environmental sensors, enables real-time monitoring of temperature, relative humidity, light intensity, and rainfall. The acquired data are transmitted to the ThingSpeak and Blynk platforms for visualization, storage, and remote monitoring. In addition, an original web interface was developed using HTML/CSS/JavaScript to provide real-time data visualization, persistent data archiving, and automatic metrological validation based on statistical indicators. Experimental tests demonstrated good system stability, a data transmission success rate exceeding 98%, and rapid recovery after power interruptions. The obtained results confirm the technical and economic feasibility of the proposed solution for academic, educational, and local environmental monitoring applications.

Keywords: Internet of Things (IoT), smart weather station, Arduino Uno WiFi R3, web interface, ThingSpeak, Blynk, environmental monitoring, metrological validation.

Liste des figures	Page
Figure 1.1 : Station météo manuelle	4
Figure 1.2 : Station météo automatique	4
Figure 1.3 : Principaux domaines d'application de la surveillance météorologique en temps réel	5
Figure 1.4 : Exemple d'application de la surveillance météorologique connectée en agriculture de précision	7
Figure 1.5 : Comparaison indicative des coûts entre stations commerciales et solution	9
Figure 1.6 : Système d'Internet des Objets	11
Figure 1.7 : Architecture générale d'un système de surveillance météorologique basé sur l'IoT	11
Figure 1.8 : Architecture fonctionnelle en couches du système embarqué IoT météorologique	12
Figure 1.9 : Principe simplifié des modèles HTTP et MQTT utilisés en IoT.	13
Figure 1.10 : Comparaison fonctionnelle des Platforms Cloud IoT (ThingSpeak) et (Blynk).	14
Figure 2.1 : Architecture matérielle adaptée de la station météorologique connectée.	18
Figure 2.2 : principaux composants de la carte Arduino Uno WiFi R3	19
Figure 2.3 : Fiche technique de la carte Arduino Uno WiFi R3	20
Figure 2.4 : Fiche technique (à gauche), et image (à droite) du module ESP8266	21
Figure 2.5 : Les quatre différents modèles de configuration des commutateurs DIP.	22
Figure 2.6 : Fiche technique (à gauche), et image (à droite) du capteur DHT22.	23
Figure 2.7 : Fiche technique (à gauche), et image (à droite) du capteur BME280.	24
Figure 2.8 : Fiche technique (à gauche), et image (à droite) du module LDR5516	25
Figure 2.9 : Fiche technique (à gauche), et image (à droite) du capteur de pluie V2	26
Figure 2.10 : Fiche technique (à gauche), et image (à droite) de l'afficheur LCD	27

1602 I2C

Figure 2.11 : Plaque d'essai	28
Figure 2.12 : Fiche technique (à gauche), et image (à droite) de la plaque d'essai	29
Figure 2.13 : Fils de connexion	30
Figure 2.14 : Schéma de câblage complet du système (vue Fritzing).	32
Figure 2.15 : Schéma de câblage complet du système (vue Proteus).	33
Figure 2.16 : Prototype physique de la station météorologique	34
Figure 2.17 : Courbe de calibration du capteur de pluie V2	35
Figure 2.18 : Dispositif de protection environnementale des composants (à gauche), et Recommandations de placement et de protection des capteurs (à droite).	36
Figure 3.1 : Diagramme de séquence du système de surveillance météorologique IoT.	39
Figure 3.2 : Diagramme d'activité du système de surveillance météorologique connectée.	40
Figure 3.3 : Sigle de l'Arduino IDE	41
Figure 3.4 : Interface graphique du logiciel Arduino IDE	41
Figure 3.5 : Organigramme du Firmware embarqué adopté pour notre projet.	43
Figure 3.6 : Principe de moyennage des mesures acquises par le capteur LDR5516.	45
Figure 3.7 : Captures d'écran de la configuration du canal ThingSpeak utilisé dans notre projet	47
Figure 3.8 : Capture d'écran de l'interface de la plateforme Blynk IoT utilisé dans notre projet.	50
Figure 3.9 : Interface mobile Blynk développée pour la supervision de la station météorologique connectée.	51
Figure 4.1 : Connexion du capteur DHT22	54
Figure 4.2 : Évolution des mesures de température et d'humidité du capteur DHT22 lors des essais expérimentaux.	55
Figure 4.3 : Connexion du capteur LDR5516	56
Figure 4.4 : Valeurs affichées par le capteur LDR5516 en fonctionnement	57

Figure 4.5: le capteur LDR5516 en fonctionnement : Lorsque la lumière est allumée (à gauche), Quand il n'y a pas de lumière (à droite)	57
Figure 4.6 : Module Pluie V2 connecté dans ces deux états : lorsqu'il ne pleut pas (à gauche) et lorsqu'il pleut (à droite).	58
Figure 4.7 : Valeurs affichées par le Module Pluie V2 en fonctionnement	59
Figure 4.8 : Afficheur LCD 1602 I ² C connecté	59
Figure 4.9 : Le message « Station météo », avec la légende «LCD ok » en dessous, bien affiché sur l'afficheur LCD 1602 I ² C.]	60
Figure 4.10 : Prototype complet de la station météorologique en fonctionnement	61
Figure 4.11: Interface ThingSpeak connectée : Température (22,1 °C), Humidité (60,8 %)	62
Figure 4.12 : Interface mobile Blynk utilisée pour la supervision en temps réel de la station météorologique	63
Figure 4.13 : Tableau de bord temps réel de l'interface web MeteoIoT développée (T°=23,4°C, H=75%, Lum=20%, Pluie détectée)	65
Figure 4.14 : Tableau récapitulatif des mesures en temps réel affiché dans l'interface web	66
Figure 4.15 : Graphiques historiques de température, luminosité et précipitations (session du 17/05/2026, 18h32–18h43)	67
Figure 4.16 : Module de validation métrologique de l'interface Web : comparaison des mesures IoT avec les données AccuWeather et calcul automatique des indicateurs MAE et du coefficient de corrélation de Pearson.	70
Figure 4.17 : Prototype final de la station météorologique en fonctionnement avec zoom sur l'afficheur LCD 1602 I ² C en fonctionnement rétroéclairage bleu actif, affichant les paramètres météorologiques en temps réel	75

Liste des tableaux	Page
Tableau 1.1 : Principales applications d'une station météorologique connectée.	6
Tableau 1.2 : Typologie des stations météorologiques existantes	8
Tableau 1.3 : Capteurs usuels d'un système météorologique	13
Tableau 1.4 : Comparaison des systèmes météorologiques existants avec la solution IoT proposée (en vert : avantages de la solution proposée)	15
Tableau 2.1 : Affectation des broches de l'ESP8266 utilisées dans la station météorologique développée.	22
Tableau 2.2 : Brochage du capteur DHT22	23
Tableau 2.3 : Brochage du module LDR5516	25
Tableau 2.4 : Seuils de détection du module Pluie V2	26
Tableau 2.5 : Récapitulatif des composants utilisés et coût estimatif du système	30
Tableau 2.6 : Consommation électrique estimée des différents composants du système	31
Tableau 2.7 : Description des liaisons électriques représentées sur le schéma Fritzing	33
Tableau 2.8 : Points de calibration du module LDR5516	35
Tableau 2.9 : Fréquences d'échantillonnage des différents capteurs.	36
Tableau 3.1 : Principales bibliothèques utilisées et leur rôle au sein du système.	42
Tableau 3.2 : Stratégie de gestion des erreurs intégrée au firmware.	46
Tableau 3.3 : Configuration des champs du canal ThingSpeak	48
Tableau 3.4 : Règles d'alerte ThingSpeak React configurées	49
Tableau 3.5: Correspondence Virtual Pins / widgets Blynk	51
Tableau 3.6 : Mécanismes de sécurité mis en œuvre dans la station météorologique connectée	52
Tableau 4.1 : Matériel de référence et critères d'acceptation par capteur	53
Tableau 4.2 : Résultats du scan I2C (GPIO4=SDA, GPIO12=SCL)	54
Tableau 4.3 : Résultats des tests unitaires du capteur DHT22	54

Tableau 4.4 : Indicateurs de performance DHT22	55
Tableau 4.5 : Résultats test LDR5516 (données ADC réelles, ordre croissant d'éclairement)	56
Tableau 4.6 : Résultats test Module Pluie V2	58
Tableau 4.7 : Résultats des essais de validation de l'afficheur LCD 1602 I2C.	60
Tableau 4.8 : Résultats du test d'intégration système complet	61
Tableau 4.9 : Analyse des phases de connectivité ThingSpeak	62
Tableau 4.10 : Résultats du test de coupure d'alimentation	64
Tableau 4.11 : Récapitulatif des mesures météorologiques acquises (valeurs induites par simulation physique).	64
Tableau 4.12 : Analyse de la stabilité et des phénomènes observés	65
Tableau 4.13 : Architecture fonctionnelle de l'interface Web développée	66
Tableau 4.14 : Métriques du stockage IndexedDB au 17/05/2026	67
Tableau 4.15 : Fonctionnalités du module de validation métrologique intégré à l'interface Web.	68
Tableau 4.16 : Principales fonctionnalités avancées intégrées à l'interface Web développée	69
Tableau 4.17 : Comparaison journalière des mesures IoT et des données de référence AccuWeather	70
Tableau 4.18 : Synthèse des indicateurs de qualité métrologique obtenus lors de la comparaison avec les données de référence	71
Tableau 4.19 : Analyse des performances globales du système	72
Tableau 4.20 : Problèmes identifiés et solutions appliquées	73
Tableau 4.21 : Consommation électrique estimée et mesurée du système	74
Tableau 4.22 : Bilan de réalisation des objectifs spécifiques	74

ADC (Analog-to-Digital Converter)

Convertisseur analogique-numérique. Composant qui transforme un signal électrique analogique continu en une valeur numérique discrète. L'ESP8266 dispose d'un ADC 10 bits (plage 0–1023) acceptant une tension maximale de 1,0 V.

API (Application Programming Interface)

Interface de programmation applicative. Ensemble de règles et de protocoles permettant à des logiciels de communiquer entre eux. Dans ce projet : Write API Key ThingSpeak.

Arduino IDE

Environnement de développement intégré (IDE) d'Arduino. Permet d'écrire, de compiler et de téléverser du code C/C++ sur les cartes Arduino et compatibles (dont l'ESP8266).

Blynk IoT

Plateforme IoT native iOS/Android permettant la surveillance à distance via des Virtual Pins. Les données sont mises à jour toutes les 10 secondes dans ce projet.

DHT22 (AM2302)

Capteur numérique de température (plage -40 à $+80$ °C, précision $\pm 0,5$ °C) et d'humidité relative (0–100 %, précision $\pm 2-5$ %) utilisant le protocole One-Wire.

ESP8266

Microcontrôleur Wi-Fi SoC (System on Chip) d'Espressif Systems, intégré à la carte Arduino Uno WiFi R3. Protocole Wi-Fi 802.11 b/g/n, pile TCP/IP complète, fréquence 80 MHz.

Firmware

Programme logiciel embarqué directement dans la mémoire Flash d'un microcontrôleur. Dans ce projet, le firmware est développé sous Arduino IDE et téléversé sur l'ESP8266.

GPIO (General Purpose Input/Output)

Broche numérique configurable en entrée ou en sortie sur un microcontrôleur. L'ESP8266 dispose de plusieurs GPIO utilisées pour les capteurs numériques.

I²C (Inter-Integrated Circuit)

Protocole de communication série synchrone sur 2 fils (SDA = données, SCL = horloge) permettant la connexion de plusieurs périphériques via un bus partagé grâce à l'adressage unique.

IndexedDB

API de stockage de données structurées côté navigateur, permettant la persistance des données sans serveur. Utilisée dans l'interface web pour archiver les mesures localement.

IoT (Internet of Things)

Internet des Objets. Réseau d'objets physiques connectés capables de collecter et d'échanger des données via Internet sans intervention humaine directe.

LDR5516

Module de mesure de luminosité intégrant une photorésistance LDR5516, un pont diviseur de tension et un comparateur LM393. Fournit une sortie analogique (0–5 V) et numérique (HIGH/LOW).

LCD 1602 I2C

Écran d'affichage à cristaux liquides de 16 colonnes × 2 lignes, équipé d'un adaptateur I²C réduisant le câblage à 4 fils (VCC, GND, SDA, SCL). Adresse I²C par défaut : 0x27.

MAE (Mean Absolute Error)

Erreur Absolue Moyenne. Indicateur métrologique calculé par : $MAE = (1/n) \times \sum |IoT(i) - \text{référence}(i)|$. Exprime l'écart moyen entre les mesures du capteur et la valeur de référence.

millis()

Fonction Arduino retournant le nombre de millisecondes écoulées depuis le démarrage du programme. Utilisée dans ce projet pour cadencer la boucle principale sans recours à `delay()`.

MQTT (Message Queuing Telemetry Transport)

Protocole de messagerie léger basé sur le modèle publication/abonnement (publish/subscribe). Adapté aux environnements IoT à ressources limitées. Port standard : 1883 (non sécurisé), 8883 (TLS).

ONM (Office National de la Météorologie)

Organisme algérien chargé de la surveillance météorologique nationale. Ses données servent de référence de comparaison dans ce projet.

r : variable de Pearson

Coefficient de corrélation linéaire. Mesure la force et la direction de la relation entre deux variables. Valeurs : $r = 1$ (corrélation parfaite positive), $r = 0$ (aucune corrélation).

ThingSpeak

Plateforme IoT open source de MathWorks permettant la collecte, le stockage et la visualisation de données en temps réel, avec intégration native MATLAB pour l'analyse avancée.

TLS (Transport Layer Security)

Protocole cryptographique assurant la confidentialité et l'intégrité des communications sur Internet. La version 1.3 (RFC 8446) est utilisée pour sécuriser les échanges avec ThingSpeak.

WMO (World Meteorological Organization)

Organisation Météorologique Mondiale. Organisme de l'ONU définissant les normes internationales en matière de surveillance météorologique (Guide WMO-No. 8).

WPA2-PSK

Wi-Fi Protected Access 2 avec clé pré-partagée. Protocole de sécurité Wi-Fi utilisant le chiffrement AES pour protéger les communications entre l'ESP8266 et le routeur.

Résumé	
Listes des figures	
Listes des tableaux	
Glossaire	
Sommaire	
Introduction générale	1
Chapitre 01 : État de l’art et positionnement du projet	
I.1 Introduction.....	3
I.2 Définition d’une station météorologique.....	3
I.3 Importance de la surveillance météorologique en temps réel.....	4
I.4 Applications de la surveillance météorologique.....	5
I.4.1 Agriculture de précision.....	6
I.4.2 Gestion des risques climatiques et environnementaux.....	7
I.5 Systèmes météorologiques commerciales et traditionnelles existants.....	8
I.6 Limites des stations météorologiques conventionnelles.....	9
I.6.1 Coût d’acquisition et de maintenance élevé.....	9
I.6.2 Complexité d’installation et accessibilité limitée.....	10
I.7 Technologies IoT pour la surveillance météorologique.....	10
I.7.1 Définition de l’internet des objets.....	10
I.7.2 Architectures des systèmes embarqués IoT.....	12
I.7.3 Choix de Microcontrôleur : ESP8266 et Arduino Uno.....	12
I.7.4 Choix des capteurs.....	13
I.7.5 Protocoles de communication : MQTT et HTTP.....	13
I.7.6 Plateformes Cloud IoT (ThingSpeak) et (Blynk).....	14
I.7.7 Cybersécurité dans les systèmes IoT.....	14
I.8 Synthèse Comparative et Positionnement de la Solution Proposée.....	15
I.9 Conclusion.....	15
Chapitre 02 : Conception matérielle du projet	
II.1 Introduction.....	17

II.2 Critères de sélection des composants matériels.....	17
II.3 Architecture matérielle du système proposé.....	18
II.4 Présentation des composants matériels.....	19
II.4.1 La carte Arduino Uno WiFi R3.....	19
II.4.2 Le capteur DHT22.....	23
II.4.3 Le capteur BMP280.....	24
II.4.4 Le module LDR5516.....	24
II.4.5 Capteur de pluie V2.....	25
II.4.6 L’afficheur LCD 1602 I2C.....	27
II.4.7 Plaque d’essai (Breadboard).....	28
II.4.8 Les fils de connexion (Jumper Wires).....	29
II.5 Récapitulatif des composants et budget.....	30
II.6 Alimentation et consommation énergétique.....	31
II.7 Intégration matérielle, calibration et protection environnementale.....	31
II.7.1 Schéma de câblage du système sous Fritzing.....	31
II.7.2 Schéma de câblage du système sous Proteus.....	33
II.7.3 Prototype réalisé.....	33
II.8 Calibration et protection environnementale.....	34
II.8.1 Calibration des capteurs.....	34
II.8.2 Stratégie d’échantillonnage.....	35
II.8.3 Protection environnementale.....	36
II.9 Conclusion.....	37

Chapitre 03 : Développement logiciel et Fireware

III.1 Introduction.....	38
III.2 Modélisation fonctionnelle du système.....	38
III.2.1 Le diagramme de séquence.....	38
III.2.2 Diagramme d’activité.....	39

III.3 Environnement de développement Arduino IDE.....	41
III.3.1 Définition.....	41
III.3.2 Structure et langage d'un programme en Arduino	42
III.3.3 Utilisation de l'IDE dans notre projet.....	42
III.4 Conception du firmware embarqué.....	43
III.4.1 Algorithme général de fonctionnement.....	43
III.4.2 Acquisition des données météorologiques.....	44
III.4.3 Algorithme de moyennage.....	44
III.4.4 Gestion des erreurs capteurs.....	45
III.5 Plateforme ThingSpeak : archivage et analyse des données.....	46
III.5.1 Configuration du canal ThingSpeak.....	47
III.5.2 Envoi des données vers ThingSpeak.....	48
III.5.3 Alertes React automatisées.....	49
III.6 Application mobile Blynk IoT.....	49
III.6.1 Présentation de la plateforme Blynk IoT.....	49
III.6.2 Configuration des Virtual Pins.....	50
III.6.3 Interface de supervision développée.....	51
III.7 Sécurité des échanges.....	52
III.8 Conclusion	52

Chapitre 04 : Tests, validation et résultats

IV.1 Introduction.....	53
IV.2 Tests unitaires des capteurs.....	53
IV.2.1 Validation du bus I2C.....	53
IV.2.2 Validation du capteur DHT22 (température et humidité).....	54
IV.2.3 Validation du module LDR5516 (luminosité).....	56
IV.2.4 Validation du module Pluie V2 (précipitations).....	58
IV.2.5 Validation de l'afficheur LCD 1602 I²C.....	59

IV.3 Tests d'intégration et de connectivité.....	60
IV.3.1 Test d'intégration du système complet.....	60
IV.3.2 Test de connectivité ThingSpeak.....	62
IV.3.3 Test de connectivité Blynk.....	62
IV.3.4 Robustesse de reprise après erreur DHT22.....	63
IV.3.5 Scénario de coupure et rétablissement de l'alimentation.....	64
IV.4 Données climatiques mesurées.....	64
IV.4.1 Tableau récapitulatif des mesures.....	64
IV.4.2 Stabilité et phénomènes observés.....	64
IV.5 Interface Web de visualisation, d'archivage et de validation.....	65
IV.5.1 Architecture fonctionnelle de l'interface Web.....	66
IV.5.2 Archivage et gestion des données.....	66
IV.5.3 Module de validation métrologique intégré.....	68
IV.5.4 Fonctionnalités avancées de l'interface Web.....	68
IV.6 Comparaison avec une station de référence externe.....	69
IV.6.1 Résultats de la comparaison.....	69
IV.6.2 Analyse et interprétation des résultats.....	71
IV.7 Analyse des performances globales.....	72
IV.7.1 Performances du système.....	72
IV.7.2 Problèmes rencontrés et solutions appliquées.....	72
IV.7.3 Consommation électrique.....	73
IV.7.4 Validation finale et bilan des objectifs.....	74
IV.8 Conclusion.....	75
Conclusion générale et perspectives.....	77
Références bibliographiques.....	79
Annexes.....	81

INTRODUCTION GÉNÉRALE

Introduction générale

L'évolution rapide des technologies numériques, en particulier celles liées à l'Internet des Objets (Internet of Things, IoT), a profondément transformé les méthodes de surveillance, d'acquisition et de traitement des données dans de nombreux domaines. Grâce à l'intégration de capteurs intelligents, de systèmes embarqués et de réseaux de communication sans fil, il est désormais possible de collecter, transmettre et exploiter des informations en temps réel avec une grande flexibilité et à un coût relativement réduit.

Parmi les domaines ayant largement bénéficié de ces avancées figure la surveillance météorologique. Les paramètres climatiques tels que la température, l'humidité relative, la pression atmosphérique, la luminosité ou encore les précipitations jouent un rôle essentiel dans plusieurs secteurs stratégiques, notamment l'agriculture de précision, la gestion des ressources naturelles, la protection de l'environnement, la production énergétique ainsi que la prévention des risques climatiques. Dans ce contexte, la disponibilité de données météorologiques locales, fiables et actualisées constitue un enjeu majeur pour la prise de décision.

Toutefois, les stations météorologiques professionnelles commercialisées présentent généralement un coût élevé et nécessitent des infrastructures spécifiques ainsi qu'une maintenance spécialisée. Ces contraintes limitent leur déploiement dans les établissements d'enseignement, les projets académiques ou les applications locales à budget réduit. Face à cette problématique, les technologies IoT apparaissent comme une alternative prometteuse permettant de développer des systèmes de surveillance météorologique économiques, modulaires et facilement déployables.

Le présent travail s'inscrit dans cette perspective et a pour objectif la conception et la réalisation d'une station météorologique connectée basée sur une carte Arduino Uno WiFi R3 associée à différents capteurs environnementaux. Le système développé permet l'acquisition en temps réel de plusieurs paramètres climatiques, notamment la température, l'humidité relative, la luminosité et les précipitations. Les données mesurées sont transmises vers des plateformes cloud dédiées afin d'assurer leur visualisation, leur archivage et leur supervision à distance.

Par ailleurs, une interface Web originale a été développée afin de compléter les fonctionnalités de la station. Cette interface assure non seulement la visualisation et l'archivage des données météorologiques, mais également leur analyse statistique et leur validation météorologique à travers le calcul automatique d'indicateurs de performance tels que l'erreur absolue moyenne (MAE) et le coefficient de corrélation de Pearson.

Afin d'évaluer les performances du système proposé, plusieurs campagnes expérimentales ont été menées dans des conditions réelles de fonctionnement. Ces essais ont permis d'analyser la fiabilité du prototype, la stabilité de la communication sans fil ainsi que la cohérence des mesures acquises.

Le présent mémoire est structuré en quatre chapitres. Le premier chapitre présente l'état de l'art relatif aux systèmes de surveillance météorologique et aux technologies IoT associées. Le deuxième chapitre est consacré à la conception matérielle de la station et au choix des différents composants. Le troisième chapitre décrit le développement logiciel, le firmware embarqué ainsi que les plateformes de supervision mises en œuvre. Enfin, le quatrième chapitre expose les résultats expérimentaux obtenus, analyse les performances globales du système et présente les principales perspectives d'amélioration.

CHAPITRE 01

État de l'art et positionnement du projet

I.1 Introduction

L'évolution des technologies de l'Internet des Objets (IoT) a favorisé le développement de systèmes intelligents capables d'assurer la surveillance et la transmission en temps réel de données environnementales. Dans le domaine météorologique, ces technologies offrent de nouvelles possibilités pour la collecte, le traitement et la visualisation des paramètres climatiques tout en réduisant les coûts d'installation et d'exploitation.

Dans ce contexte, les stations météorologiques connectées constituent une alternative intéressante aux systèmes conventionnels, notamment pour les applications académiques, agricoles et environnementales nécessitant des solutions accessibles et facilement déployables.

Ce chapitre présente, après avoir défini une station météorologique basique, les notions fondamentales relatives à la surveillance météorologique et aux technologies IoT associées. Une analyse des principales solutions existantes est ensuite menée afin d'identifier leurs caractéristiques et leurs limites. Enfin, une étude comparative est réalisée dans le but de positionner la solution développée dans ce mémoire par rapport aux systèmes actuellement disponibles.

I.2 Définition d'une station météorologique

Une station météorologique est un ensemble de capteurs qui enregistrent et fournissent les mesures physiques et les paramètres météorologiques liés aux changements climatiques. Ces capteurs sont placés dans un caisson (résistant au vent et aux intempéries), offrant l'équilibre de la chaleur avec l'air et protège le thermomètre du rayonnement solaire.

Les variables à mesurer sont la température, la pression, la vitesse et la direction du vent, humidité, précipitations, hauteur et type de nuages, type et intensité précipitations et visibilité. La station peut inclure tous les capteurs ou une partie. Ces informations dépendent de leur nature : météorologie agricole, aéroport, météo routière, climat, etc.

Les stations météorologiques sont généralement situées à des coordonnées fixes sur l'île et sur le continent, mais ils peuvent aussi être utilisés dans les avions, les navires (surtout navire météo).

Ils peuvent également être déplacés à d'autres fins tels que des expériences scientifiques ou des conditions météo déterminantes lors de catastrophes telles que les incendies de forêt. Les données que nous obtenons peuvent être, dans le cas d'une station automatique, envoyées directement sous forme de bulletin météo, ou faites à partir de l'observation humaine. [1]

Il existe deux types de station météorologique : manuelle et automatique et elles sont définies comme suit :

➤ Station météo manuelle :

La mesure de cette station est effectuée par un technicien, les stations météorologiques font des observations météorologiques régulières.



Figure 1.1 : Station météo manuelle

➤ Station météo automatique : [2]

Développée pour les endroits difficiles d'accès (zones isolées ou bouées météorologiques marines), une station automatique est une station dont les capteurs rapportent une série de données météorologiques sans intervention manuelle.

L'instrument est généralement le même dans les deux cas, mais certaines observations sont plus fiables que l'observateur humain.

Par exemple, il est plus facile de remarquer la quantité et le type de précipitations que les humains à l'aide d'un instrument électronique.

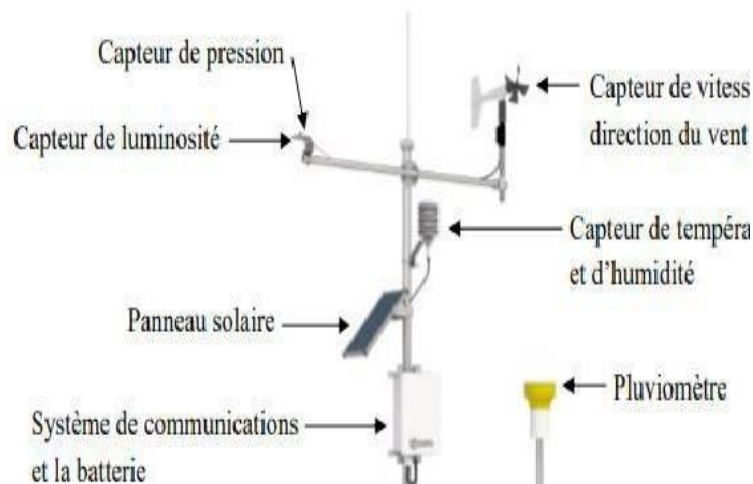


Figure 1.2 : Station météo automatique

I.3 Importance de la surveillance météorologique en temps réel

Depuis l'apparition des premiers instruments scientifiques, tels que le thermomètre et le baromètre, au XVIIe siècle, la météorologie s'est progressivement orientée vers la mesure quantitative des grandeurs physiques de l'atmosphère.

Aujourd'hui, la surveillance météorologique est devenue essentielle pour comprendre le comportement de l'atmosphère et analyser ses fluctuations. Elle consiste à mesurer et à suivre

de manière continue les principaux paramètres atmosphériques tels que la température, l'humidité relative, la pression atmosphérique, les précipitations et le rayonnement solaire. Ces informations constituent une source essentielle pour l'analyse des conditions climatiques et la compréhension des phénomènes environnementaux [3].

Avec l'accélération des changements climatiques et l'augmentation de la fréquence des événements météorologiques extrêmes, la disponibilité de données fiables et actualisées est devenue un enjeu majeur. Les systèmes de surveillance modernes permettent d'acquérir des mesures en temps réel et de fournir des informations utiles à la prise de décision dans plusieurs domaines d'activité [4].

Les données météorologiques jouent notamment un rôle important dans l'agriculture de précision, où elles contribuent à optimiser les opérations d'irrigation, à améliorer la gestion des cultures et à réduire les pertes liées aux aléas climatiques. Elles interviennent également dans la gestion des ressources énergétiques, la planification des transports, la protection de l'environnement ainsi que dans les dispositifs de prévention et d'alerte face aux risques naturels [5].

1.4 Applications de la surveillance météorologique

Les données météorologiques constituent un outil d'aide à la décision dans de nombreux domaines d'activité. L'amélioration des technologies de mesure et de communication permet aujourd'hui d'exploiter ces informations en temps réel afin d'optimiser les performances des systèmes techniques, d'améliorer la gestion des ressources naturelles et de renforcer les dispositifs de prévention face aux risques environnementaux.

L'émergence des technologies IoT a considérablement élargi les possibilités d'utilisation des données météorologiques grâce à la collecte automatisée des mesures, leur transmission à distance et leur intégration dans des plateformes d'analyse et de supervision. Les applications concernent aussi bien les secteurs agricoles, énergétiques et industriels que les domaines liés à la sécurité civile et à la protection de l'environnement.



Figure 1.3 : Principaux domaines d'application de la surveillance météorologique en temps réel.

Le tableau suivant récapitule les différentes exploitations possibles des systèmes météorologiques selon leurs domaines d'application

Chapitre 01 : État de l'art et positionnement du projet

Domaine d'application	Paramètres utiles	Exploitation possible
Agriculture de précision	Température, humidité, pluie, luminosité	Planification de l'irrigation, suivi du stress hydrique, aide à la décision
Gestion des risques	Pluie, humidité, pression	Surveillance des épisodes pluvieux, alerte locale, suivi des variations rapides
Énergie	Luminosité, température, vent si disponible	Estimation des conditions favorables à la production solaire ou éolienne
Recherche et pédagogie	Toutes les grandeurs mesurées	Acquisition de données, traitement statistique, expérimentation instrumentale
Surveillance locale	Température, humidité, pluie	Description d'un microclimat et constitution d'un historique de mesures

Tableau 1.1 : Principales applications d'une station météorologique connectée.

Ces applications montrent que la valeur d'une station connectée dépend de la qualité de ses données et de la capacité de l'utilisateur à les interpréter.

L'agriculture de précision et la gestion des risques climatiques sont les domaines les plus directement concernés par les systèmes de mesure connectés développés dans le cadre de ce travail. Elles sont alors développées dans ce qui suit :

I.4.1 Agriculture de précision

L'agriculture de précision repose sur l'utilisation des technologies de l'information et de la communication afin d'optimiser la gestion des exploitations agricoles. Cette approche vise à adapter les pratiques culturales aux conditions réelles du terrain grâce à l'acquisition et à l'analyse de données environnementales mesurées localement [6].

Dans ce contexte, les stations météorologiques connectées jouent un rôle essentiel en fournissant des informations fiables et actualisées sur les paramètres climatiques influençant directement le développement des cultures. La température de l'air constitue un indicateur important pour l'évaluation des conditions de croissance des plantes et la détection des situations de stress thermique. L'humidité relative permet d'estimer les besoins hydriques des cultures et d'anticiper l'apparition de certaines maladies phytosanitaires. Les données relatives aux précipitations contribuent à optimiser les programmes d'irrigation et à améliorer la gestion des ressources en eau, tandis que les mesures de luminosité permettent d'évaluer l'énergie disponible pour les processus de photosynthèse [7].

L'intégration de ces données dans des plateformes numériques facilite la prise de décision et permet aux agriculteurs d'intervenir de manière plus efficace et plus rationnelle. Les systèmes de surveillance météorologique basés sur l'Internet des Objets offrent ainsi une solution adaptée aux exigences de l'agriculture moderne en favorisant une utilisation plus durable des ressources naturelles tout en améliorant la productivité agricole [8].



Figure 1.4 : Exemple d'application de la surveillance météorologique connectée en agriculture de précision.

I.4.2 Gestion des risques climatiques et environnementaux

La surveillance météorologique joue un rôle essentiel dans la prévention et la gestion des risques climatiques et environnementaux. L'augmentation de la fréquence des événements météorologiques extrêmes, tels que les fortes précipitations, les vagues de chaleur, les sécheresses ou les tempêtes, nécessite la mise en place de systèmes de surveillance capables de fournir des informations fiables et actualisées en temps réel [9].

Les stations météorologiques connectées permettent de suivre en continu l'évolution des principaux paramètres atmosphériques et de détecter rapidement les variations susceptibles d'annoncer des situations à risque. Les mesures de température et d'humidité contribuent à l'identification des périodes de stress climatique, tandis que le suivi des précipitations permet d'évaluer les risques d'inondation ou de sécheresse. De même, l'analyse des tendances météorologiques locales facilite l'élaboration de stratégies de prévention adaptées aux spécificités de chaque région [10].

Grâce aux technologies de l'Internet des Objets, les données collectées peuvent être transmises instantanément vers des plateformes de supervision où elles sont analysées et exploitées pour la génération d'alertes précoces. Cette capacité de réaction rapide améliore considérablement la gestion des situations d'urgence et contribue à réduire les impacts humains, économiques et environnementaux associés aux phénomènes météorologiques extrêmes [11].

I.5 Systèmes météorologiques commerciales et traditionnelles existants

Les systèmes météorologiques existants regroupent un ensemble d'installations destinées à mesurer, enregistrer et analyser les paramètres atmosphériques. Selon la World Meteorological Organisation (WMO) [12] : « Une station météorologique est une installation équipée d'instruments conçus pour observer et mesurer les conditions atmosphériques telles que la température, l'humidité, la pression, le vent et les précipitations. ».

Les stations météorologiques existantes peuvent être regroupées en trois catégories. Les stations professionnelles sont conçues pour fournir des mesures fiables, stables et exploitables dans un cadre normatif. Elles sont généralement plus coûteuses et exigent une installation rigoureuse. Les stations commerciales grand public proposent un compromis entre prix et fonctionnalités, mais elles sont souvent limitées par des architectures propriétaires. Les stations académiques ou open source visent principalement l'apprentissage, la recherche appliquée et l'adaptation à des besoins locaux.

Catégorie	Avantages	Limites	Exemple
Professionnelle	Robustesse, précision, stabilité, documentation technique	Coût élevé, Maintenance spécialisée	Réseaux météorologiques, stations certifiées
Commerciale grand public	Installation relativement simple, interface conviviale	Architecture souvent propriétaire, personnalisation limitée	Davis, Ambient Weather, La Crosse
Académique / IoT	Faible coût, modularité, apprentissage, adaptation locale	Précision dépendante des capteurs et de la calibration	Arduino / ESP8266 / ESP32 avec cloud IoT

Tableau 1.2 : Typologie des stations météorologiques existantes.

La station Davis Vantage Pro2 constitue un exemple de station commerciale avancée fréquemment utilisée dans des applications exigeantes. Davis Instruments annonce notamment des mesures de vent, pluie, température et humidité avec des fréquences d'actualisation différentes selon les grandeurs [13]. Elle fournit des mesures précises en temps réel de la température, de l'humidité relative, de la pression atmosphérique, de la vitesse et de la direction du vent et des précipitations. Ses capteurs sont conçus pour résister à des conditions météorologiques extrêmes, garantissant une fiabilité élevée sur le long terme. Cependant, son coût élevé (~1 000 USD) limite son adoption dans des projets à petite échelle.

À titre de comparaison, la plage de température opérationnelle de la Davis Vantage Pro2 est de -40 °C à +65 °C, avec des capteurs certifiés par l'Organisation Météorologique Mondiale (WMO) pour certaines applications professionnelles.

D'autres stations, telles que les gammes **Ambient Weather** ou **La Crosse**, répondent davantage au marché grand public. Elles offrent une interface utilisateur simple et un coût inférieur à celui des systèmes professionnels, mais elles sont généralement moins ouvertes pour l'expérimentation électronique et logicielle. En comparaison, une station IoT réalisée dans un cadre académique offre une plus grande personnalisation : choix des capteurs, modification du firmware, ajout d'une interface web et adaptation des alertes.

L'histogramme suivant illustre une comparaison indicative des coûts entre quelques stations commerciales et notre solution proposée :

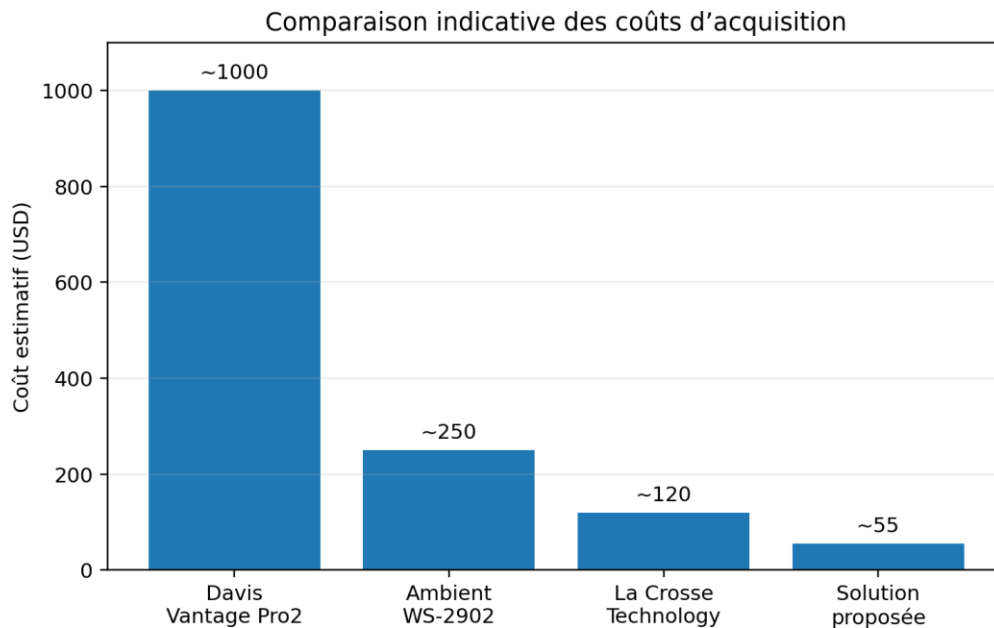


Figure 1.5 : Comparaison indicative des coûts entre stations commerciales et solution proposée.

I.6 Limites des stations météorologiques conventionnelles

Les stations météorologiques conventionnelles sont largement utilisées pour la surveillance des paramètres atmosphériques grâce à leur précision et à leur fiabilité. Elles équipent notamment les services météorologiques nationaux, les centres de recherche et certaines infrastructures industrielles nécessitant des mesures de haute qualité [14]. Cependant, elles présentent plusieurs limitations qui restreignent leur utilisation dans divers contextes.

I.6.1 Coût d'acquisition et de maintenance élevé

Le coût élevé ne concerne pas seulement l'acquisition initiale, mais inclut également la maintenance, la calibration des capteurs et le remplacement périodique des modules. À titre d'exemple, une station météorologique professionnelle complète peut dépasser plusieurs milliers de dollars, tandis qu'une station commerciale telle que la Davis Vantage Pro2 Wireless se situe dans une gamme de prix élevée. À l'inverse, une solution IoT à faible coût

peut être réalisée pour un budget nettement plus accessible, tout en répondant à des besoins locaux de surveillance.

I.6.2 Complexité d'installation et accessibilité limitée

La nécessité d'un entretien spécialisé pour les stations professionnelles limite leur déploiement à grande échelle, en particulier dans les zones rurales ou en développement où les ressources financières et techniques sont restreintes. Cette contrainte économique limite l'accès aux données climatiques pour les exploitations agricoles de petite taille, les programmes éducatifs et les projets de recherche locaux.

L'innovation dans les stations automatiques à faible coût constitue une réponse stratégique aux limitations financières des systèmes traditionnels et favorise des réseaux de surveillance météorologique plus inclusifs.

Face à ces contraintes, l'émergence des technologies de l'Internet des Objets a favorisé le développement de solutions alternatives basées sur des composants électroniques à faible coût, capables d'assurer la collecte et la transmission des données météorologiques tout en offrant une plus grande flexibilité de déploiement. Ces nouvelles approches visent à rendre la surveillance météorologique plus accessible tout en conservant des performances compatibles avec de nombreuses applications locales.

I.7 Technologies IoT pour la surveillance météorologique

Les technologies de l'Internet des Objets (IoT) occupent aujourd'hui une place essentielle dans le développement des systèmes intelligents de surveillance environnementale. Selon l'Union internationale des télécommunications : « L'IoT est une infrastructure mondiale pour la société de l'information, permettant des services avancés grâce à l'interconnexion d'objets physiques et virtuels. ».

I.7.1 Définition de l'internet des objets

L'Internet des objets (IOT) est un réseau d'objets physiques intégrés avec des capteurs, des logiciels via un réseau, intégrant les fonctionnalités informatiques et de mise en réseau dans tout type de matériel. Nous utilisons ces capacités pour examiner l'état de l'objet et le modifier si possible, par collecter et échanger des données selon des protocoles spécifiés. Ces données collectées peuvent être analysées et utilisées pour optimiser les processus, réaliser des tâches complexes, améliorer la prise de décision et améliorer l'efficacité globale ou encore de créer des liens entre les mondes virtuels et physiques. [15] Les systèmes IoT rassemblent de nombreux acteurs et composants technologiques. Il se compose d'objets connectés, d'un réseau de communication sans fil et d'une plateforme de collecte, d'hébergement et de traitement des données [16].

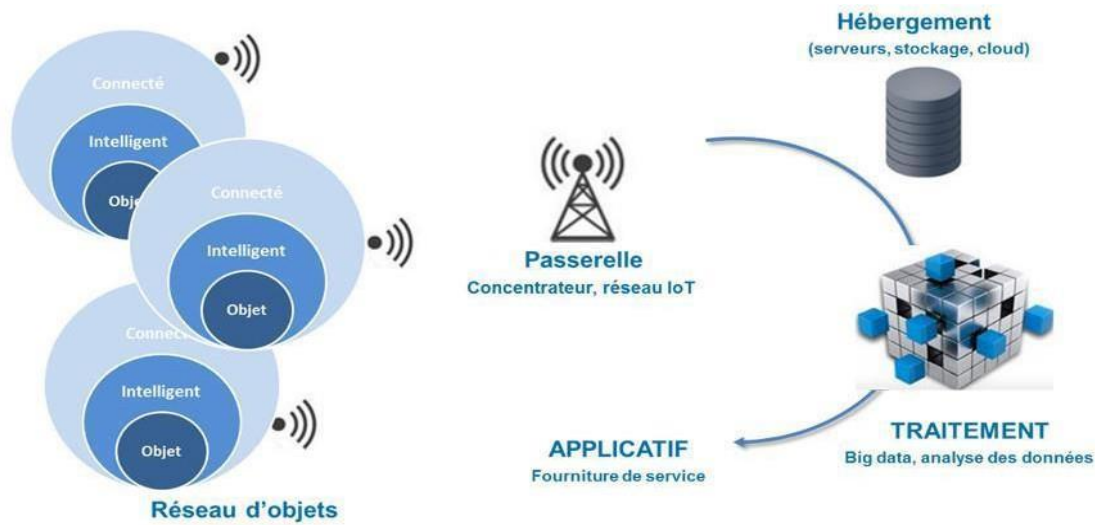


Figure 1.6 : Système d'Internet des Objets [16]

L'intégration des technologies de l'Internet des Objets (IoT) a considérablement amélioré les capacités de surveillance météorologique. Grâce à l'utilisation de capteurs connectés, de systèmes embarqués et de plateformes cloud, il est désormais possible de collecter, transmettre et visualiser les données météorologiques à distance avec une grande réactivité. Cette évolution a favorisé le développement de solutions de surveillance plus accessibles, flexibles et adaptées aux besoins locaux.

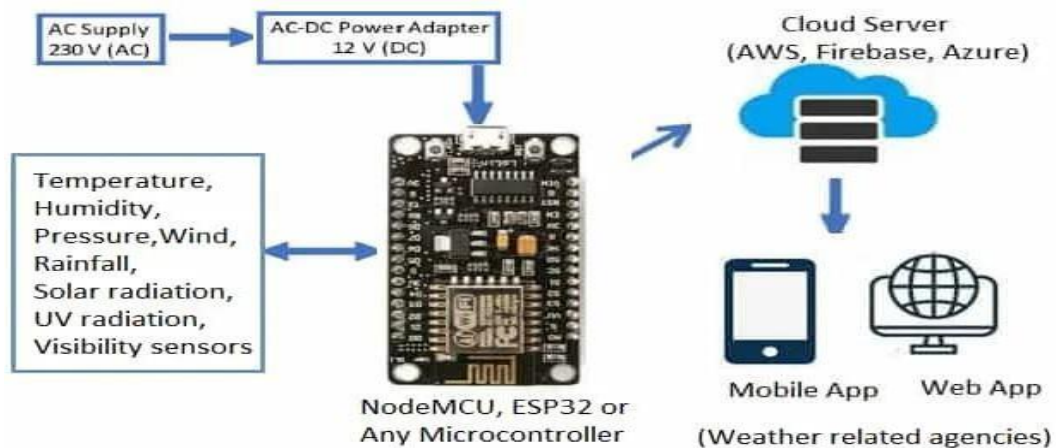


Figure 1.7 : Architecture générale d'un système de surveillance météorologique basé sur l'IoT [6].

Concrètement, relier un capteur à Internet ne suffit pas : encore faut-il que les données soient fiables, continues et directement exploitables. C'est ce que permettent les architectures IoT actuelles, à condition de maîtriser les contraintes de chaque composant.

I.7.2 Architectures des systèmes embarqués IoT

Un système embarqué est une combinaison de matériels et de logiciels permettant de remplir une ou plusieurs fonctions spécifiques avec des contraintes de consommation d'énergie, de température et de performances.

Dans le cadre de ce projet, l'architecture repose sur quatre couches fonctionnelles complémentaires :

1. **La couche de perception** : qui a pour rôle l'acquisition des paramètres climatiques via les capteurs (DHT22, LDR, YL-83)
2. **La couche de traitement** assurant l'acquisition, le filtrage et la gestion des erreurs.
3. **La couche de communication** qui transmet les données vers une plateforme distante : HTTP vers ThingSpeak, Wi-Fi via ESP8266
4. Enfin, **la couche** qui permet l'affichage, l'archivage, l'analyse et éventuellement l'envoi d'alertes : visualisation sur ThingSpeak et alertes automatisées via Blynk.

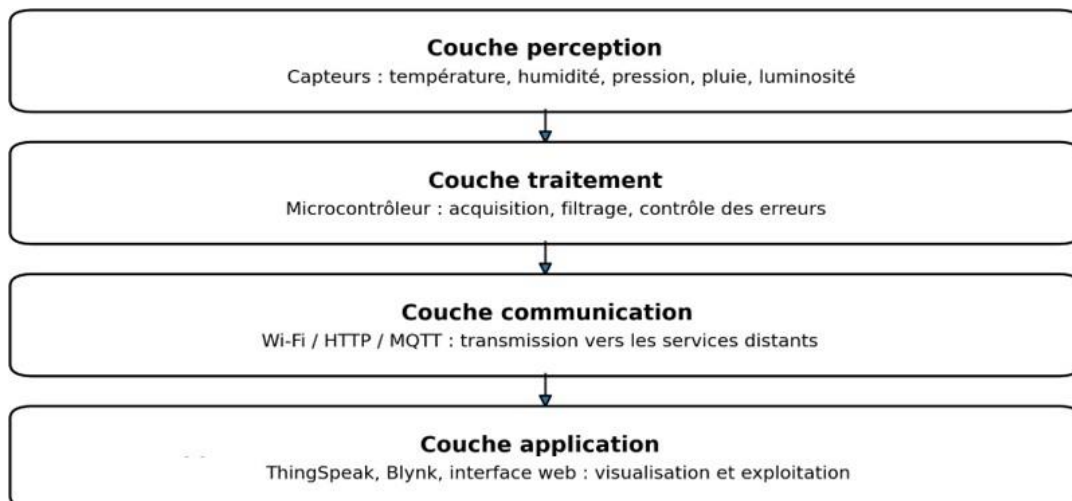


Figure 1.8 : Architecture fonctionnelle en couches du système embarqué IoT météorologique

I.7.3 Choix de Microcontrôleur : ESP8266 et Arduino Uno

Les projets de stations météorologiques IoT s'appuient généralement sur des microcontrôleurs accessibles, peu coûteux et compatibles avec les capteurs courants. Les plateformes Arduino sont très utilisées dans l'enseignement et le prototypage, car elles offrent un environnement simple et une documentation abondante. L'ESP8266, de son côté, est un système sur puce Wi-Fi largement adopté pour les objets connectés ; Espressif le présente comme une solution Wi-Fi à faible consommation, adaptée aux équipements mobiles et aux applications IoT [17].

Dans ce projet, la carte Arduino Uno WiFi R3 présente un intérêt pédagogique : elle permet de combiner l'environnement Arduino avec une connectivité Wi-Fi assurée par un module ESP8266.

Chapitre 01 : État de l'art et positionnement du projet

Or, la carte Arduino Uno WiFi R3 réunit sur un même PCB un microcontrôleur ATmega328P et un module ESP8266 distinct. Les deux puces communiquent via une liaison série interne, mais restent deux entités physiquement séparées. Cette architecture à double composant permet de répartir les tâches : l'ATmega328P exécute le programme principal, tandis que le module Wi-Fi assure la connectivité réseau.

Ce choix n'est pas nécessairement le plus performant par rapport à un ESP32 moderne, mais il reste cohérent avec notre objectif de concevoir, câbler, programmer et valider un prototype complet avec des composants disponibles localement.

I.7.4 Choix des capteurs

La qualité d'une station météorologique dépend directement de la qualité des capteurs et de leur intégration. Les capteurs à faible coût permettant de réaliser des prototypes d'un système météorologique accessible sont regroupés dans le tableau suivant :

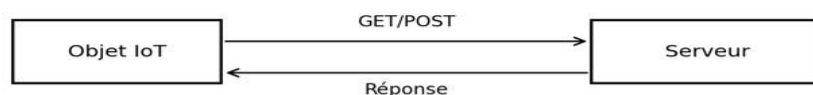
Grandeur	Capteur courant	Rôle dans le projet
Température	DHT22 / BME280	Suivi thermique local
Humidité relative	DHT22 / BME280	Suivi de l'humidité ambiante
Pression	BMP280 / BME280	Indication barométrique
Luminosité	LDR	Estimation de l'éclairement
Pluie	Module Rain V2 / YL-83	Détection de précipitations

Tableau 1.3 : Capteurs usuels d'un système météorologique.

I.7.5 Protocoles de communication : MQTT et HTTP

La communication représente une partie importante de la valeur ajoutée des stations IoT. Les données ne sont plus simplement affichées localement ; elles peuvent être enregistrées, représentées sous forme graphique et consultées à distance. Les deux protocoles les plus courants dans ce type de projet sont HTTP et MQTT.

HTTP : modèle requête/réponse



MQTT : modèle publication/abonnement



Figure 1.9 : Principe simplifié des modèles HTTP et MQTT utilisés en IoT.

Le protocole MQTT (Message Queuing Telemetry Transport) est un protocole léger standardisé par OASIS, basé sur le modèle publication/abonnement, particulièrement adapté aux environnements à ressources limitées [18].

Le protocole HTTP est plus universel, simple à mettre en œuvre et compatible avec de nombreux services web, mais consomme davantage de bande passante. C'est celui que nous avons adopté dans notre projet.

I.7.6 Platforms Cloud IoT (ThingSpeak) et (Blynk)

ThingSpeak est une plateforme IoT proposée par MathWorks, permettant d'agréger, visualiser et analyser des flux de données en direct dans le cloud [19], gratuite pour un usage non commercial et limité. Elle permet de collecter, stocker, analyser et visualiser les données en temps réel, avec une intégration native MATLAB pour le traitement avancé. Contrairement à une idée répandue, ThingSpeak n'est pas open source : son code serveur est propriétaire, mais son API publique est ouverte et largement documentée.

Blynk, de son côté, est une plateforme IoT permettant de prototyper, déployer et superviser des dispositifs connectés à distance. Elle permet de contrôler et de surveiller à distance des dispositifs via une application mobile à interface graphique simple et intuitive.



Figure 1.10 : Comparaison fonctionnelle des Plateforms Cloud IoT (ThingSpeak) et (Blynk).

I.7.7 Cybersécurité dans les systèmes IoT

La sécurité constitue un enjeu majeur dans les systèmes IoT en raison de la transmission des données via Internet. Les mécanismes de sécurité utilisés dans ce projet incluent le chiffrement des données via TLS/SSL lorsque cela est nécessaire, l'authentification par clés API pour les services cloud, la protection du réseau Wi-Fi par protocole WPA2-PSK, et la validation des entrées au niveau du firmware afin de rejeter les commandes mal formées. Dans notre implémentation, ThingSpeak est interrogé via HTTP (port 80), les données météorologiques transmises n'étant pas confidentielles. Le port 8883 serait réservé à un broker MQTT dédié comme HiveMQ, non utilisé ici.

I.8 Synthèse Comparative et Positionnement de la Solution Proposée :

Le projet que nous avons développé et proposé comme solution, s'inscrit dans la catégorie des systèmes météorologiques académiques à faible coût. Il repose sur une carte Arduino Uno WiFi R3 associée à des capteurs de température, d'humidité, de luminosité et de pluie. Les données sont transmises vers ThingSpeak et Blynk afin d'assurer une supervision à distance en temps réel.

Le tableau suivant met en évidence que la solution proposée se positionne comme une alternative économique, flexible et adaptée aux besoins locaux, tout en conservant une architecture IoT complète. Si elle ne rivalise pas encore avec les stations professionnelles en termes de gamme de capteurs, elle répond parfaitement aux besoins de surveillance locale à faible coût.

Critère	Davis Vantage Pro2	Ambient Weather WS-2902	Solution IoT proposée
Coût estimé	~1 000 USD	~200–300 USD	~50–150 USD Coût local réel : 8 600 DZD
Capteurs intégrés	T°, H, P, Vent, Pluie, UV	T°, H, P, Vent, Pluie	T°, H, P, Pluie, Luminosité
Connectivité	Wi-Fi / RF / Bluetooth	Wi-Fi	Wi-Fi (ESP8266)
Plateforme cloud	Propriétaire	Ambient Weather Network	ThingSpeak / Blynk
Protocole communication	RF propriétaire	Wi-Fi / HTTP	MQTT / HTTP
Maintenance	Spécialisée / Coûteuse	Standard	Simple / Autonome
Personnalisation	Limitée	Partielle	Totale (open source)
Cybersécurité	Intégrée	Basique	TLS / API Key / WPA2
Public cible	Professionnel	Grand public	Académique / Local

Tableau 1.4 : Comparaison des systèmes météorologiques existants avec la solution IoT proposée (en vert : avantages de la solution proposée)

I.9 Conclusion

Ce chapitre a permis de présenter les concepts essentiels liés à la surveillance météorologique et de mettre en évidence l'intérêt des solutions connectées basées sur l'IoT. L'analyse des systèmes existants a montré les limites des stations traditionnelles en termes de coût et d'accessibilité, ce qui justifie le développement d'une solution adaptée, économique et flexible.

Le tableau comparatif confirme que la solution IoT proposée basée sur l'Arduino Uno WiFi R3, les capteurs DHT22/BMP280/LDR/YL-83 et la plateforme ThingSpeak répond aux

Chapitre 01 : État de l’art et positionnement du projet

besoins de surveillance locale tout en restant accessible financièrement. Le chapitre suivant traduit ces choix en composants réels : quels capteurs, pourquoi ceux-là, et comment les contraintes du marché local ont parfois imposé des alternatives inattendues.

CHAPITRE 02

Conception matérielle du projet

II.1 Introduction

Pour tout projet, le choix des outils appropriés est essentiel. Ce chapitre présente ces choix comme une série de décisions dictées par trois contraintes principales : la disponibilité locale des composants, leur compatibilité avec l'Arduino IDE, et le compromis inévitable entre précision et coût.

En outre, ce chapitre démontre comment ces décisions, en choisissant les composants matériels pour le prototype, leur câblage avec la carte Arduino Uno WiFi R3, ainsi que les procédures de calibration et de protection physique, façonnent l'architecture complète du programme.

II.2 Critères de sélection des composants matériels

La conception d'un système de surveillance météorologique connecté repose en grande partie sur le choix judicieux des composants matériels qui le constituent. La performance globale du système, sa fiabilité, son coût ainsi que sa facilité de mise en œuvre dépendent directement des caractéristiques des capteurs et de la plateforme de traitement retenus.

Dans le cadre de ce projet, la sélection des composants a été réalisée en tenant compte de plusieurs critères techniques et pratiques :

1. Le premier critère concerne la compatibilité matérielle et logicielle avec l'environnement de développement Arduino IDE, afin de garantir une intégration simple des différents modules et de faciliter le développement du firmware. La disponibilité de bibliothèques logicielles fiables et bien documentées a également été prise en considération.
2. Le second critère est lié à la disponibilité des composants sur le marché local. Ce facteur revêt une importance particulière dans les projets à budget limité, car il conditionne non seulement le coût de réalisation du prototype, mais également la facilité de maintenance et de remplacement des composants en cas de défaillance. Certaines décisions ont ainsi été influencées par les contraintes d'approvisionnement rencontrées au moment de la réalisation du projet.
3. La précision des mesures constitue également un critère essentiel, notamment pour les capteurs chargés de surveiller les paramètres météorologiques.

Toutefois, dans le cadre d'une solution destinée à la surveillance environnementale locale, le choix ne repose pas uniquement sur les performances métrologiques. Un compromis entre précision, simplicité d'utilisation et coût d'acquisition a été recherché afin de concevoir une solution techniquement satisfaisante et économiquement accessible.

La simplicité de câblage et l'intégration matérielle ont été aussi prises en compte afin de limiter la complexité de réalisation du prototype. Les composants retenus devaient permettre une mise en œuvre rapide tout en garantissant une bonne fiabilité de fonctionnement.

L'ensemble de ces critères a conduit à la sélection d'une architecture matérielle basée sur une carte Arduino Uno WiFi R3 associée à plusieurs capteurs environnementaux destinés à la mesure de la température, de l'humidité relative, de la luminosité et des précipitations.

II.3 Architecture matérielle du système proposé

Le système développé repose sur une architecture modulaire destinée à assurer l'acquisition, le traitement et la transmission des données météorologiques en temps réel. Il est constitué d'une unité centrale de traitement, de plusieurs capteurs environnementaux ainsi que d'un module de communication permettant la transmission des données vers des plateformes de supervision distantes.

La carte Arduino Uno WiFi R3 constitue le cœur du système. Elle assure l'acquisition des données issues des différents capteurs, leur traitement et leur transmission via le réseau Wi-Fi intégré. Les paramètres surveillés comprennent la température, l'humidité relative, la pression atmosphérique, la luminosité ambiante ainsi que la détection des précipitations.

Les données collectées sont transmises vers les plateformes ThingSpeak et Blynk, qui assurent respectivement le stockage, la visualisation et la supervision à distance des mesures. Cette architecture permet d'obtenir un système de surveillance météorologique simple, évolutif et adapté aux applications de suivi environnemental en temps réel.

La figure 2.1 présente l'architecture générale du système développé :

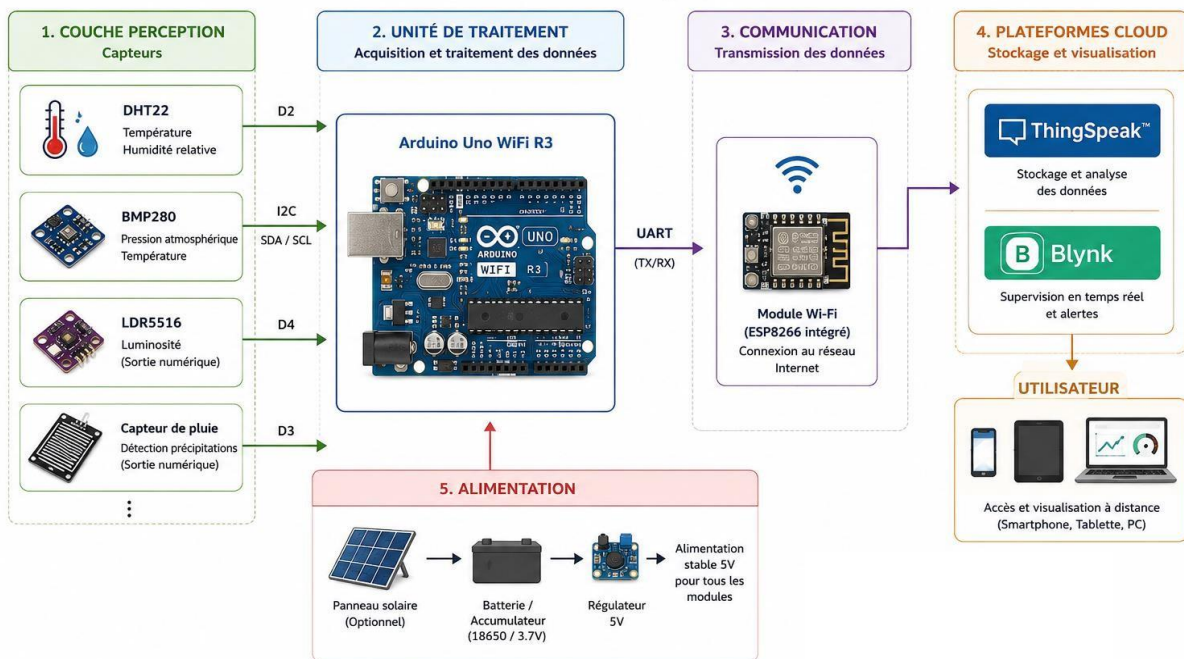


Figure 2.1 : Architecture matérielle adaptée de la station météorologique connectée.

II.4 Présentation des composants matériels

Après avoir présenté l'architecture générale du système, il convient de décrire les différents composants matériels retenus pour sa réalisation ainsi que les principales caractéristiques ayant motivé leur sélection :

II.4.1 La carte Arduino Uno WiFi R3

La carte Arduino Uno WiFi R3 constitue l'unité centrale de la station météorologique développée. Elle assure l'acquisition des données issues des différents capteurs, leur traitement ainsi que leur transmission vers les plateformes de supervision à distance. Cette carte se distingue par l'intégration, sur un même circuit imprimé, d'un microcontrôleur ATmega328P et d'un module Wi-Fi ESP8266, ce qui permet de combiner les fonctionnalités d'une carte Arduino classique avec les capacités de communication requises pour les applications de l'Internet des Objets (IoT).

Afin de mieux comprendre l'architecture de la carte, la figure 2.2 met en évidence ses principaux composants.

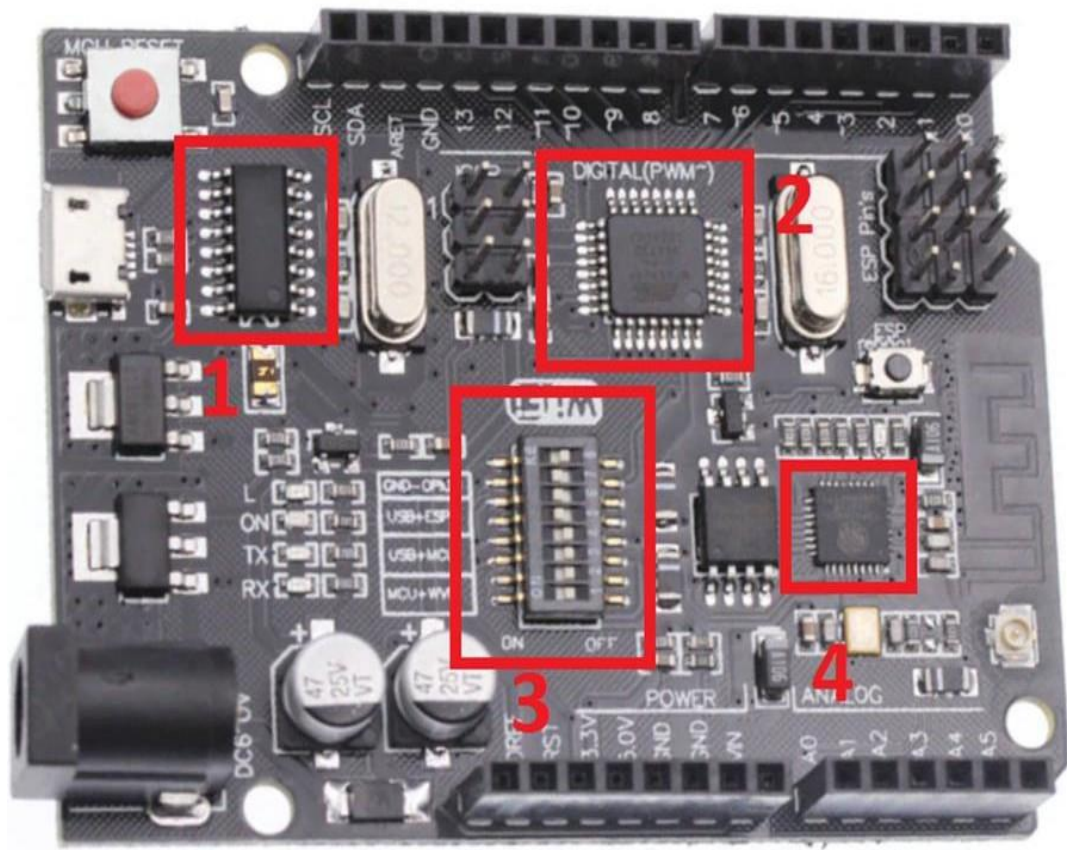


Figure 2.2 : principaux composants de la carte Arduino Uno WiFi R3 [20]

L'élément (1) correspond au convertisseur USB-série CH340, utilisé pour la programmation et la communication avec l'ordinateur. **L'élément (2)** désigne le microcontrôleur ATmega328P, qui constitue le cœur de la plateforme Arduino. **L'élément (3)** correspond aux commutateurs DIP permettant la configuration des différents modes de fonctionnement de la carte. Enfin, **l'élément (4)** identifie le module ESP8266, chargé des fonctions de communication Wi-Fi.

Dans le cadre de ce projet, les commutateurs DIP ont été configurés de manière à exécuter directement le programme sur l'ESP8266. Cette approche permet d'exploiter les bibliothèques natives dédiées aux applications IoT et d'assurer la communication avec les plateformes ThingSpeak et Blynk sans recourir à un module Wi-Fi externe. Elle contribue également à simplifier l'architecture matérielle du système tout en réduisant son coût de réalisation.

Les principales caractéristiques techniques de la carte sont résumées dans la figure 2.3 :

Arduino Uno WiFi R3 — Fiche Technique	
Microcontrôleur	ATmega328P (8 bits, 16 MHz)
Module Wi-Fi intégré	ESP8266 (IEEE 802.11 b/g/n)
Mémoire Flash	32 KB (0.5 KB bootloader)
SRAM	2 KB
EEPROM	1 KB
Tension de travail	5 V
Tension d'entrée	7 – 12 V (jack DC)
Broches numériques	14 (dont 6 PWM)
Broches analogiques	6 (résolution 10 bits)
Courant par broche	40 mA max
Courant 3.3 V	50 mA max
Dimensions	68.6 mm × 53.4 mm

Figure 2.3 : Fiche technique de la carte Arduino Uno WiFi R3 [20].

II.4.1.1 Le module ESP8266

Le module ESP8266 intégré à la carte Arduino Uno WiFi R3 assure les fonctions de communication sans fil du système. Il s'agit d'un microcontrôleur doté d'une connectivité Wi-Fi conforme à la norme IEEE 802.11 b/g/n et intégrant une pile TCP/IP complète. Grâce à ses capacités de traitement et de communication, il peut être utilisé de manière autonome pour le développement d'applications IoT.

Dans cette étude, l'ESP8266 a été retenu comme unité principale d'exécution du programme, permettant ainsi l'acquisition des données issues des capteurs et leur transmission vers les plateformes ThingSpeak et Blynk. Cette solution simplifie l'architecture matérielle du système tout en offrant une connectivité réseau native adaptée aux applications de surveillance environnementale en temps réel.

ESP8266 — Fiche Technique	
CPU	Xtensa LX106 — 80 MHz (mode éco)
Wi-Fi standard	IEEE 802.11 b/g/n (2.4 GHz)
Sécurité Wi-Fi	WEP, WPA, WPA2-PSK, WPA2-Enterprise
Pile réseau	TCP/IP complète intégrée
Tension de travail	3.3 V
Courant émission	≈ 170 mA (pic lors des émissions)
Courant réception	≈ 60 mA
Courant veille	< 1 mA (mode sleep)
GPIO disponibles	16 (dont 1 ADC 10 bits)
Protocoles	MQTT, HTTP, HTTPS, WebSocket
Portée Wi-Fi	≈ 100 m (espace ouvert)
Avantage principal	Wi-Fi intégré — coût très faible

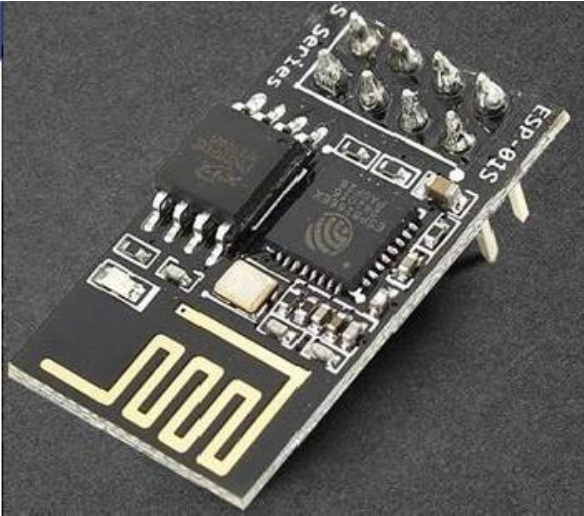


Figure 2.4 : Fiche technique (à gauche), et image (à droite) du module ESP8266 [21]

Il convient toutefois de noter que l'ESP8266 ne dispose que d'une seule entrée analogique (TOUT/A0), limitée à une tension maximale de 1 V. Cette contrainte a été prise en compte lors de l'intégration du capteur de luminosité à travers l'utilisation d'un circuit d'adaptation de tension approprié.

II.4.1.2 Configuration des commutateurs DIP

Dans notre implémentation, l'ATmega328P n'est pas utilisé. Alors, les commutateurs DIP ont été configurés pour programmer directement l'ESP8266, qui exécute l'intégralité du firmware. Ce choix permet d'exploiter nativement les bibliothèques ESP8266 (BlynkSimpleEsp8266.h, ESP8266WiFi.h) et d'accéder directement aux GPIO de l'ESP8266. En conséquence, les numéros de broches référencés dans le code correspondent aux GPIO de l'ESP8266 et non aux broches Arduino.

La figure 2.5 illustre les quatre différents modèles de configuration des commutateurs DIP :

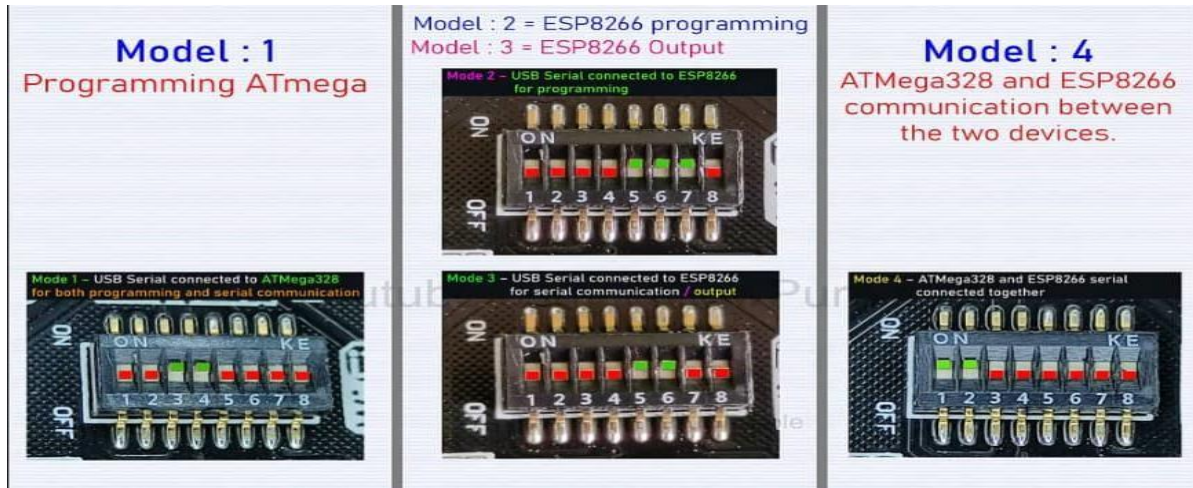


Figure 2.5 : Les quatre différents modèles de configuration des commutateurs DIP.[20]

Dans le cadre de notre projet, deux configurations ont été principalement utilisées :

- Pour transférer le code depuis l'ordinateur vers l'ESP8266, nous avons choisis (*MODEL 2 : EPS8266 programming* dans la figure 2.6), dans lequel les commutateurs 5, 6 et 7 sont positionnés sur ON afin d'établir une communication directe entre le port USB et le module Wi-Fi.
- Une fois le téléversement terminé, le commutateur 7 est placé en position OFF (*MODEL 3 : EPS8266 Output* dans la figure 2.6), et les commutateurs 5 et 6 restent activés permettant ainsi l'exécution normale du firmware tout en conservant l'accès au moniteur série de l'Arduino IDE.

Ce flux en deux phases : commutateur 7 ON pour programmer, commutateur 7 OFF puis Reset pour exécuter a été reproduit à chaque itération du développement.

Ainsi, le tableau 2.1 présente les principales broches de l'ESP8266 utilisées pour l'interfaçage des différents capteurs et périphériques de la station météorologique.

GPIO ESP8266	Broche Arduino équivalente	Composant	Rôle
GPIO2	D2	DHT22	One-Wire DATA(pull-up interne active)
GPIO0	D3	Pluie V2 DO	Détection seuil (high = pluie)
GPIO16	D0	LDR5516 DO	Seuil luminosité numérique
TOUT (A0)	A0	LDR5516 AO	Luminosité 0–1023 (unique ADC ESP8266)
GPIO4	D2(SDA)	BMP280 + LCD	I2C SDA Wire.begin(4, 12)
GPIO12	D6(SCL)	BMP280 + LCD	I2C SCL Wire.begin(4, 12)

Tableau 2.1 : Affectation des broches de l'ESP8266 utilisées dans la station météorologique développée.

Après avoir présenté la plateforme matérielle assurant le traitement et la transmission des données (**la carte Arduino Uno WiFi R3**), il convient désormais de décrire les différents capteurs intégrés à la station météorologique :

II.4.2 Le capteur DHT22

Le DHT22 (AM2302) est un capteur numérique utilisé pour la mesure de la température ambiante et de l'humidité relative. Il communique avec le microcontrôleur à travers un protocole numérique de type One-Wire nécessitant une seule ligne de données. Ce capteur a été retenu en raison de sa simplicité d'intégration, de sa faible consommation énergétique et de sa précision supérieure à celle du DHT11, fréquemment utilisé dans les applications de surveillance environnementale.

DHT22 (AM2302) — Fiche Technique	
Grandeurs mesurées	Température + Humidité relative
Plage température	-40 °C à +80 °C
Précision température	± 0.5 °C
Plage humidité	0 % à 100 % HR
Précision humidité	± 2 % HR
Fréquence max	0.5 Hz (1 mesure / 2 s)
Protocole	One-Wire numérique
Alimentation	3.3 V - 5 V
Consommation	1.5 mA (mesure) / 50 µA (veille)
Broche DATA	Résistance pull-up 10 kΩ requise

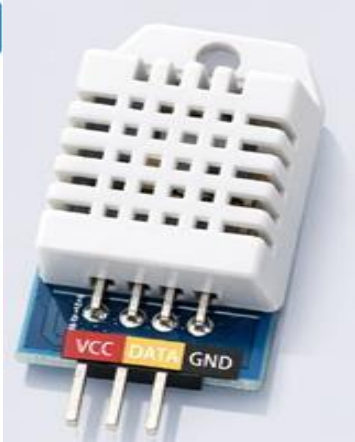


Figure 2.6 : Fiche technique (à gauche), et image (à droite) du capteur DHT22.[22]

Comme illustré à la figure 2.6, le DHT22 permet la mesure de températures comprises entre -40 °C et +80 °C avec une précision de $\pm 0,5$ °C, tandis que la plage de mesure de l'humidité relative s'étend de 0 % à 100 % HR avec une précision de ± 2 % HR. Ces performances le rendent particulièrement adapté aux applications de surveillance météorologique et environnementale.

Le module utilisé dans cette étude intègre directement une résistance de tirage (*pull-up*) sur son circuit imprimé, ce qui simplifie son raccordement à la carte Arduino Uno WiFi R3. Le signal de données est connecté à la broche GPIO2 de l'ESP8266, tandis que l'alimentation est assurée sous 5 V. Le tableau suivant illustre le brochage du capteur DHT22 :

Broche	Arduino	Esp8266
VCC	5 V	5 V
DATA	D2	GPIO2
NC	/	/
GND	GND	GND

Tableau 2.2 : Brochage du capteur DHT22

Le choix de ce capteur a été motivé par son bon compromis entre précision, coût et facilité de mise en œuvre. Il constitue une solution adaptée à l'acquisition des paramètres climatiques de base nécessaires au fonctionnement de la station météorologique développée.

II.4.3 Le capteur BMP280

Le BMP280 est un capteur environnemental numérique de Bosch Sensortec, qui mesure la pression atmosphérique et la température ambiante grâce à une interface I²C. Le BME280 peut remplacer efficacement le BMP280, en ajoutant une troisième mesure d'humidité relative sans coût supplémentaire ni complexité accrue du câblage. [23]

Comme illustré à la figure 2.7, le BMP280 présente une large plage de mesure et une faible consommation énergétique, ce qui le rend particulièrement adapté aux applications de surveillance environnementale et aux systèmes IoT alimentés à faible puissance.

BMP280 — Fiche Technique (Bosch Sensortec)	
Grandeurs mesurées	Pression atmosphérique + Température
Plage pression	300 hPa à 1100 hPa
Précision pression	± 1 hPa (mode standard)
Résolution pression	0.0016 hPa (mode ultra haute rés.)
Plage température	-40 °C à +85 °C
Précision température	± 0.5 °C
Protocoles	I2C (0x76 / 0x77) ET SPI
Alimentation	1.71 V - 3.6 V (module : 3.3/5 V)
Consommation mesure	2.7 µA (mode normal, 1 Hz)
Consommation veille	0.1 µA (mode sleep)
Modes de fonctionnement	Sleep, Forced, Normal
Dimensions puce	2.0 mm × 2.5 mm × 0.95 mm



Figure 2.7 : Fiche technique (à gauche), et image (à droite) du capteur BMP280.[23]

II.4.4 Le module LDR5516

La mesure de la luminosité ambiante constitue un paramètre important dans une station météorologique, notamment pour l'évaluation des conditions d'ensoleillement et leur influence sur l'environnement surveillé. Dans notre projet, nous avons choisis le module LDR5516 basé sur une photorésistance dont la valeur varie en fonction de l'intensité lumineuse reçue.

Le module LDR5516 regroupe sur un même PCB la photorésistance LDR5516, un pont diviseur de tension, un comparateur LM393 et un potentiomètre de seuil. Par rapport à une LDR nue, il supprime le besoin d'une résistance externe et fournit deux sorties : une analogique (0–5 V, proportionnelle à l'éclairement) et une numérique (HIGH/LOW selon le seuil réglé). [24]

Comme illustré à la figure 2.8, le module dispose de deux sorties complémentaires : une sortie analogique (AO) fournissant une tension proportionnelle à l'éclairement et une sortie numérique (DO) permettant la détection d'un seuil lumineux prédéfini. Dans cette étude, la

sortie analogique a été utilisée pour estimer le niveau de luminosité ambiante tandis que la sortie numérique a servi à la détection d'un seuil lumineux configurable.

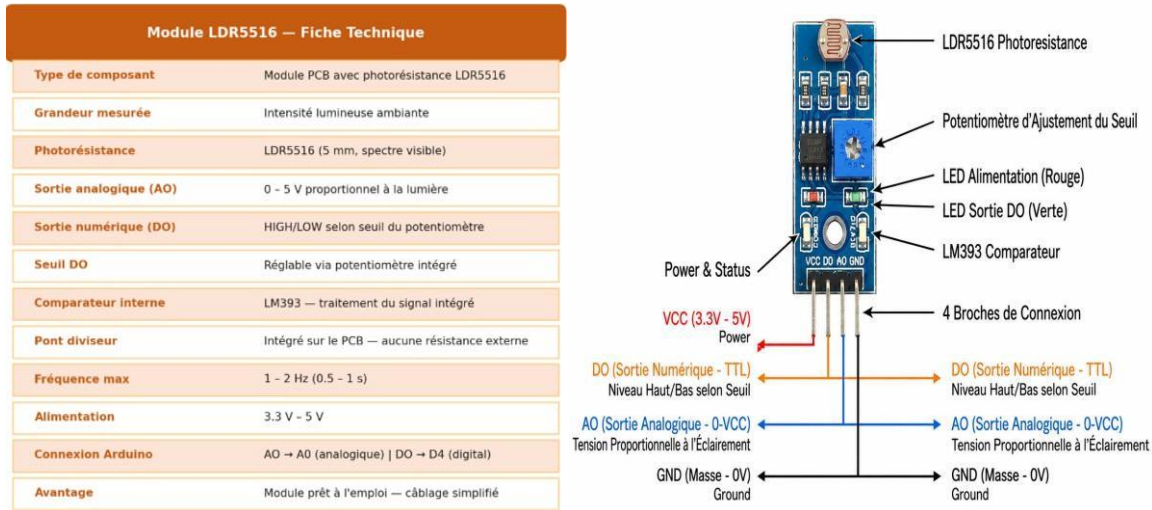


Figure 2.8 : Fiche technique (à gauche), et image (à droite) du module LDR5516 [24]

Le module est raccordé à l'ESP8266 à travers son entrée analogique TOUT (A0). Toutefois, cette entrée étant limitée à une tension maximale de 1 V, un circuit d'adaptation a été utilisé afin de garantir la compatibilité électrique entre le capteur et le microcontrôleur.

Il convient de noter que la relation entre la résistance de la photorésistance et l'éclairement n'est pas linéaire. Par conséquent, les valeurs obtenues représentent une estimation relative de la luminosité plutôt qu'une mesure photométrique absolue exprimée en lux. Malgré cette limitation, le module LDR5516 demeure une solution simple, économique et adaptée aux objectifs du projet.

Le brochage du module est présenté dans le tableau 2.3 :

Broche	Arduino	Esp8266	Signal
VCC	5 V	5 V	Alimentation
GND	GND	GND	Masse
AO	A0	TOUT(A0)	Luminosité 0–1023(100%–0%)
DO	D4	GPIO16	Seuil HIGH/LOW

Tableau 2.3 : Brochage du module LDR5516

II.4.5 Capteur de pluie V2

La détection des précipitations constitue une fonction essentielle de la station météorologique développée. Pour cette tâche, nous avons choisis un module de pluie V2 afin d'identifier la présence et l'intensité des précipitations à partir de l'accumulation d'eau sur une surface conductrice.

Le capteur est constitué d'une plaque sensible reliée à un module électronique intégrant un comparateur LM393. Lorsque des gouttelettes d'eau entrent en contact avec la surface de détection, la conductivité électrique de la plaque varie, entraînant une modification du signal délivré par le capteur. Cette variation peut être exploitée sous forme analogique ou numérique selon les besoins de l'application.

Comme illustré à la figure 2.9, la version V2 utilisée dans cette étude présente une surface de détection nickelée offrant une meilleure résistance à la corrosion que la version (V1). Cette amélioration augmente la durée de vie du capteur et permet une meilleure sensibilité à l'humidité et aux fines précipitations. Le câblage et le code restent identiques entre les deux versions.



Figure 2.9 : Fiche technique (à gauche), et image (à droite) du capteur de pluie V2[25]

Le module dispose de deux sorties complémentaires : une sortie analogique (AO) permettant d'estimer l'intensité des précipitations et une sortie numérique (DO) destinée à la détection d'un seuil réglable à l'aide d'un potentiomètre intégré. Cette double fonctionnalité facilite son intégration dans les systèmes de surveillance météorologique.

Les seuils de détection de ce capteur sont résumés dans le tableau 2.4. Ils sont valables avec le potentiomètre du module réglé en position médiane. Toute modification de ce réglage déplace les valeurs ADC de bascule et nécessite une recalibration complète.

Valeur AO	État	Action
< 200	Sec	Aucune
200 – 800	Pluie modérée	Log ThingSpeak
> 800	Forte pluie	Alerte MQTT

Tableau 2.4 : Seuils de détection du module Pluie V2

Le choix de ce capteur a été motivé par sa simplicité de mise en œuvre, son faible coût et sa capacité à fournir une information rapide sur l'occurrence des précipitations. Il constitue ainsi une solution adaptée à la surveillance météorologique en temps réel.

Les broches du module sont :

- VCC : alimentation (3,3 V ou 5 V)
- GND : masse
- AO : sortie analogique
- DO : sortie digitale

Son câblage avec l'Arduino Uno est :

- VCC → 5 V
- GND → GND
- AO → A1 (mesure du niveau d'eau)
- DO → broche digitale (ex : D2)

II.4.6 L'afficheur LCD 1602 I2C

L'afficheur LCD 1602 I2C est utilisé pour la visualisation locale des données météorologiques acquises par les différents capteurs. Il permet l'affichage de deux lignes de seize caractères et constitue une interface simple pour le suivi en temps réel des paramètres mesurés.

Comme illustré à la figure 2.10, l'intégration d'un adaptateur I2C à l'arrière du module réduit le câblage à quatre connexions seulement : VCC, GND, SDA et SCL. Cette configuration facilite son interfaçage avec la carte Arduino Uno WiFi R3 tout en limitant l'occupation des broches disponibles.

Écran LCD 16x2 I2C — Fiche Technique	
Format d'affichage	16 colonnes × 2 lignes de caractères
Interface	I2C via module PCF8574
Adresse I2C	0x27 (par défaut)
Broches utilisées	4 seulement : VCC, GND, SDA, SCL
Connexion Arduino	SDA → A4 SCL → A5
Rétroéclairage	LED bleue (contrôlable par firmware)
Contraste	Réglable via potentiomètre intégré
Alimentation	5 V
Bibliothèque Arduino	LiquidCrystal_I2C.h
Avantage principal	Réduction du câblage (16 fils → 4 fils)



Figure 2.10 : Fiche technique (à gauche), et image (à droite) de l'afficheur LCD 1602 I2C [26]

Dans notre projet, l'afficheur partage le même bus I2C que le capteur BMP280. L'absence de conflit est assurée par l'utilisation d'adresses distinctes pour les deux périphériques. Grâce à sa simplicité de mise en œuvre, son faible coût et sa bonne lisibilité, le LCD 1602 I2C constitue une solution adaptée à l'affichage local des informations météorologiques.

Le module I2C possède généralement 4 broches :

- VCC : alimentation (5 V)
- GND : masse
- SDA : ligne de données I2C
- SCL : ligne d'horloge I2C

Son câblage avec Arduino Uno(esp8266) :

- VCC → 5 V
- GND → GND
- SDA → GPIO4 (Wire.begin(4, 12))
- SCL → GPIO12 (Wire.begin(4, 12))

II.4.7 Plaque d'essai (Breadboard)

La réalisation du prototype a nécessité l'utilisation d'une plaque d'essai (*Breadboard*) permettant l'assemblage et l'interconnexion des différents composants sans recourir à des opérations de soudure. Cette solution facilite les phases de développement, de test et de modification du circuit électronique.

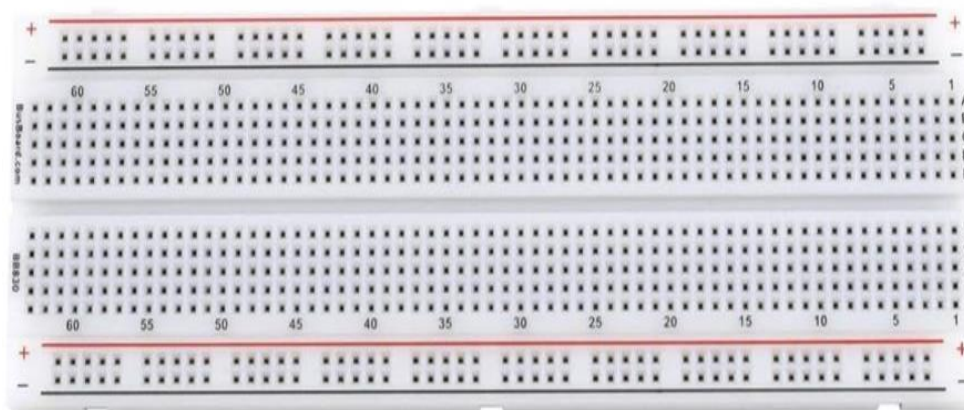


Figure 2.11 : Plaque d'essai

Comme illustré à la figure 2.12, la breadboard est constituée de zones de connexion internes organisées en rangées et colonnes conductrices, ainsi que de rails d'alimentation destinés à la distribution des tensions d'alimentation et de masse. Son pas standard de 2,54 mm assure la compatibilité avec l'ensemble des composants utilisés dans cette étude.

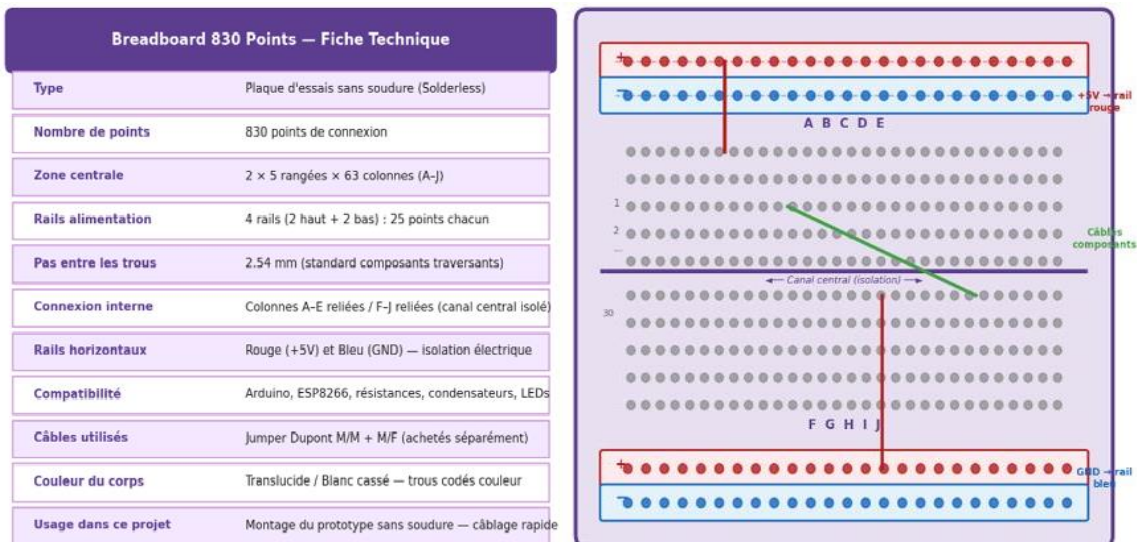


Figure 2.12 : Fiche technique (à gauche), et image (à droite) de la plaque d'essai

II.4.8 Les fils de connexion (Jumper Wires) :

Les fils de connexion sont des conducteurs utilisés pour relier différents composants électroniques entre eux dans les circuits expérimentaux. [27]

✓ Caractéristiques et avantages :

- Flexibles et faciles à manipuler
- Disponibles en différentes longueurs et couleurs
- Permettent un câblage rapide sans soudure
- Compatibles avec les breadboards et cartes Arduino
- Réutilisables pour plusieurs montages

✓ Utilisation :

- Prototypage électronique sur breadboard
- Connexion de capteurs et modules
- Réalisation de circuits temporaires
- Projets Arduino et IoT

✓ Types (entrées / sorties) :

Ces fils ne possèdent pas de broches électroniques internes, mais des connecteurs aux extrémités :

- Male–Male : broches rigides des deux côtés
- Male–Female : une broche et un connecteur femelle
- Female–Female : connecteurs femelles des deux côtés

✓ Nomenclature des extrémités :

- Pin (Male) : tige métallique utilisée pour insertion dans breadboard ou carte
- Socket (Female) : connecteur creux pour recevoir une broche mâle.



Figure 2.13 : Fils de connexion [27]

II.5 Récapitulatif des composants et budget

Les différents composants retenus pour la réalisation de la station météorologique ont été sélectionnés de manière à répondre aux exigences fonctionnelles du projet tout en maintenant un coût de réalisation réduit. L'association de ces éléments permet d'assurer l'acquisition des principaux paramètres météorologiques, leur traitement, leur affichage local ainsi que leur transmission vers les plateformes de supervision à distance.

Le tableau 2.5 présente les composants utilisés, les grandeurs mesurées ou fonctions assurées, ainsi que leur coût estimatif. L'investissement total nécessaire à la réalisation du prototype est d'environ 8 600 DZD, ce qui demeure largement inférieur au coût des stations météorologiques commerciales tout en offrant les fonctionnalités essentielles de surveillance environnementale.

Composant	Fonction / Grandeur mesurée	Fréquence d'acquisition	Prix (DZD)
DHT22	Température et humidité relative	1 mesure / 10 s	1 300
BME280	Pression atmosphérique, température et humidité	1 mesure / 10 s	800
LDR5516	Luminosité ambiante	10 mesures / 10 s	500
Capteur de pluie V2	Détection des précipitations	10 mesures / 10 s	600
LCD 1602 I²C	Affichage local des données	Temps réel	1 300
Breadboard	Support de prototypage	—	800
Câbles Dupont	Interconnexion des composants	—	800
Total	—	—	8 600

Tableau 2.5 : Récapitulatif des composants utilisés et coût estimatif du système

II.6 Alimentation et consommation énergétique

L'alimentation de la station météorologique est assurée par un adaptateur continu de 9 V / 1 A raccordé à la carte Arduino Uno WiFi R3. Le régulateur de tension intégré fournit les niveaux de tension nécessaires au fonctionnement des différents composants du système, notamment les capteurs environnementaux, l'afficheur LCD et le module de communication Wi-Fi.

La consommation énergétique constitue un critère important pour les systèmes de surveillance connectés, en particulier lorsqu'un fonctionnement autonome sur batterie est envisagé. Dans le système que nous avons développé, la consommation électrique est principalement dominée par le module ESP8266 lors des phases de transmission des données, tandis que les autres composants présentent des besoins énergétiques relativement faibles.

Le tableau 2.6 présente une estimation de la consommation des principaux éléments du système en mode actif et en mode veille. Il apparaît que la consommation totale atteint environ 240 mA en fonctionnement normal et diminue à près de 27 mA lorsque les composants sont placés en mode veille.

Composant	Mode actif	Veille
Arduino (hors ESP8266)	~45 mA	~20 mA
ESP8266 (émission)	~170 mA	< 1 mA
LCD 1602	~20 mA	~5 mA
DHT22 + BME280	~2,5 mA	< 0,1 mA
LDR5516 + Pluie V2	~2,5 mA	~1 mA
Total estimé	~240 mA	~27 mA

Tableau 2.6 : Consommation électrique estimée des différents composants du système

Cette analyse montre que la station météorologique peut être alimentée par une source d'énergie autonome telle qu'une batterie rechargeable. À titre indicatif, l'utilisation d'une batterie Li-Po de 3,7 V et 2000 mAh associée à un convertisseur élévateur de tension permet d'obtenir une autonomie d'environ sept heures en fonctionnement continu. Ces résultats confirment l'adéquation du système pour des applications de surveillance environnementale nécessitant une faible consommation énergétique.

II.7 Intégration matérielle, calibration et protection environnementale

II.7.1 Schéma de câblage du système sous Fritzing

La réalisation de la station météorologique connectée a nécessité l'interconnexion des capteurs environnementaux choisis, d'un afficheur LCD et de la carte Arduino Uno WiFi R3.

Chapitre 02 : Conception matérielle du projet

Le schéma de câblage réalisé sous Fritzing (figure 2.14) permet de visualiser l'ensemble des connexions matérielles et constitue une référence pour le montage du prototype.

L'un des aspects importants de cette architecture est l'utilisation d'un bus I2C partagé entre le capteur BMP280 et l'afficheur LCD 1602 I2C. Cette approche réduit le nombre de connexions nécessaires tout en simplifiant le câblage du système. Le DHT22 est connecté à une broche numérique dédiée, tandis que les modules LDR5516 et Pluie V2 exploitent des entrées spécifiques pour l'acquisition des données environnementales.

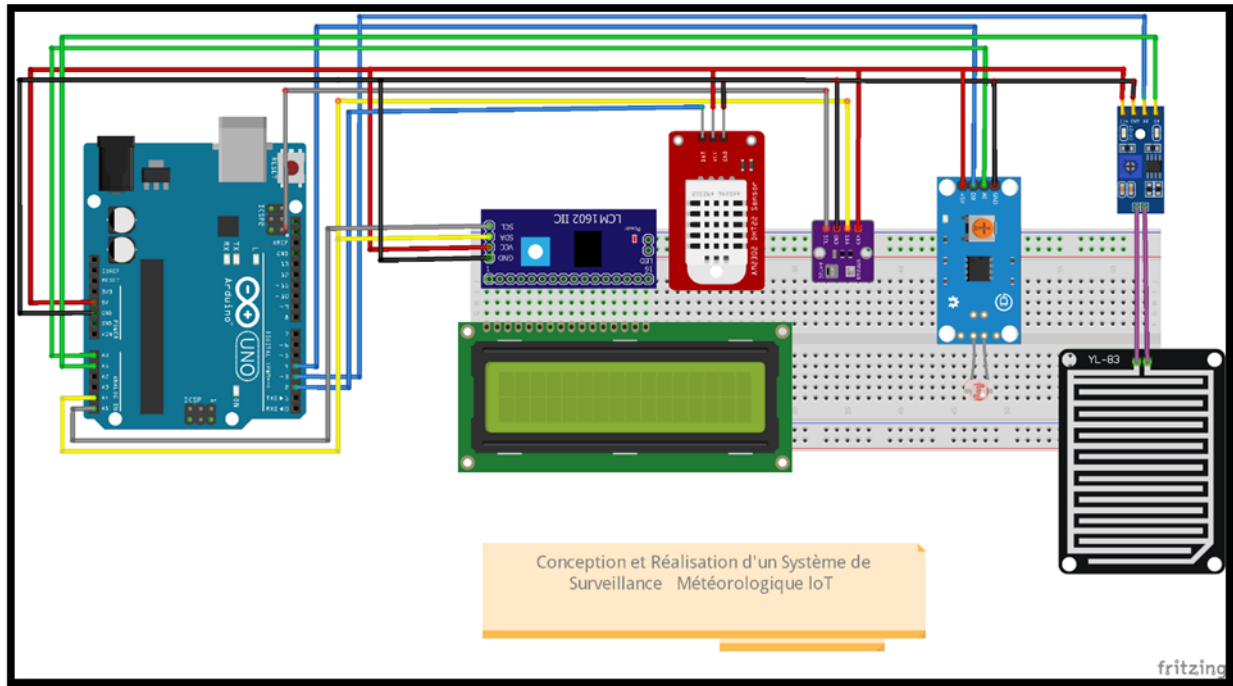


Figure 2.14 : Schéma de câblage complet du système (vue Fritzing).

Le tableau 2.7 décrit les fils du montage, leurs couleurs et les connexions entre les différents composants visibles sur le schéma de câblage réalisé sous Fritzing de la figure 2.14 :

Fil	Couleur	(Arduino)(esp8266)	Vers(composant)	Broche composant
Alimentation	rouge	5V	Tous les composants	VCC
Masse	noir	GND	Tous les composants	GND
SDA(I2C)	jaune	A4,GPIO4	LCD 1602 I2C	SDA
SDA(I2C)	jaune	A4,GPIO4	BMP280	SDA
SCL(I2C)	gris	A5,GPIO12	LCD 1602 I2C	SCL
SCL(I2C)	gris	A5,GPIO12	BMP280	SCL
DHT22.DATA	bleu	D2,GPIO2	DHT22	DATA
LDR.analogique	vert	A0,TOUT(A0)	Module.LDR5516	A0

LDR.numérique	bleu	D4,GPIO16	Module.LDR5516	D0
Pluie.analogique	vert	A1, —	YL-83	A0
Pluie.numérique	bleu	D3,GPIO0	YL-83	D0

Tableau 2.7 : Description des liaisons électriques représentées sur le schéma Fritzing

II.7.2 Schéma de câblage du système sous Proteus

Proteus 8 Professional permet de valider le circuit par simulation avant le montage physique. [28]. Le moteur SPICE vérifie les niveaux de tension et les courants sur chaque branche. Les deux outils sont complémentaires : Fritzing pour la documentation, Proteus pour la validation.

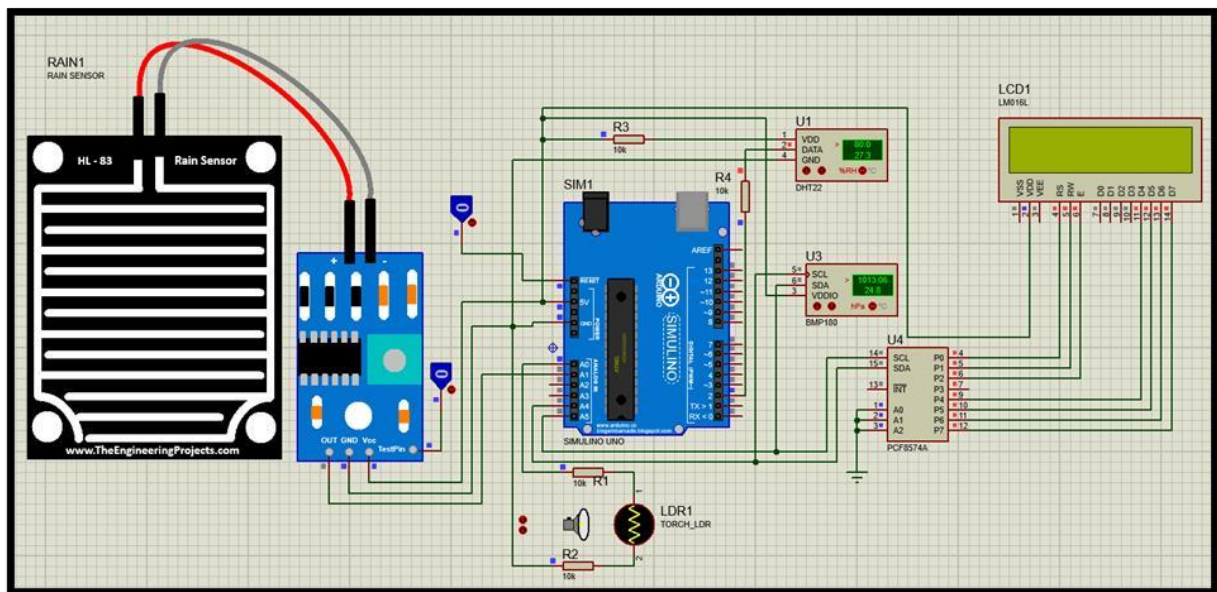


Figure 2.15 : Schéma de câblage complet du système (vue Proteus).

Cette organisation facilite la maintenance du prototype et permet de vérifier rapidement la cohérence du câblage lors des phases de test et de validation.

II.7.3 Prototype réalisé

Après validation du schéma de câblage, un prototype physique a été assemblé sur plaque d'essai. Cette étape a permis de vérifier le bon fonctionnement des capteurs, de l'afficheur et du système de communication avant toute phase de déploiement.

La figure 2.16 présente le prototype final réalisé. Les différents modules sont interconnectés à l'aide de câbles Dupont et alimentés par la carte Arduino Uno WiFi R3. Les essais expérimentaux ont confirmé la bonne acquisition des données météorologiques et leur affichage local sur l'écran LCD.

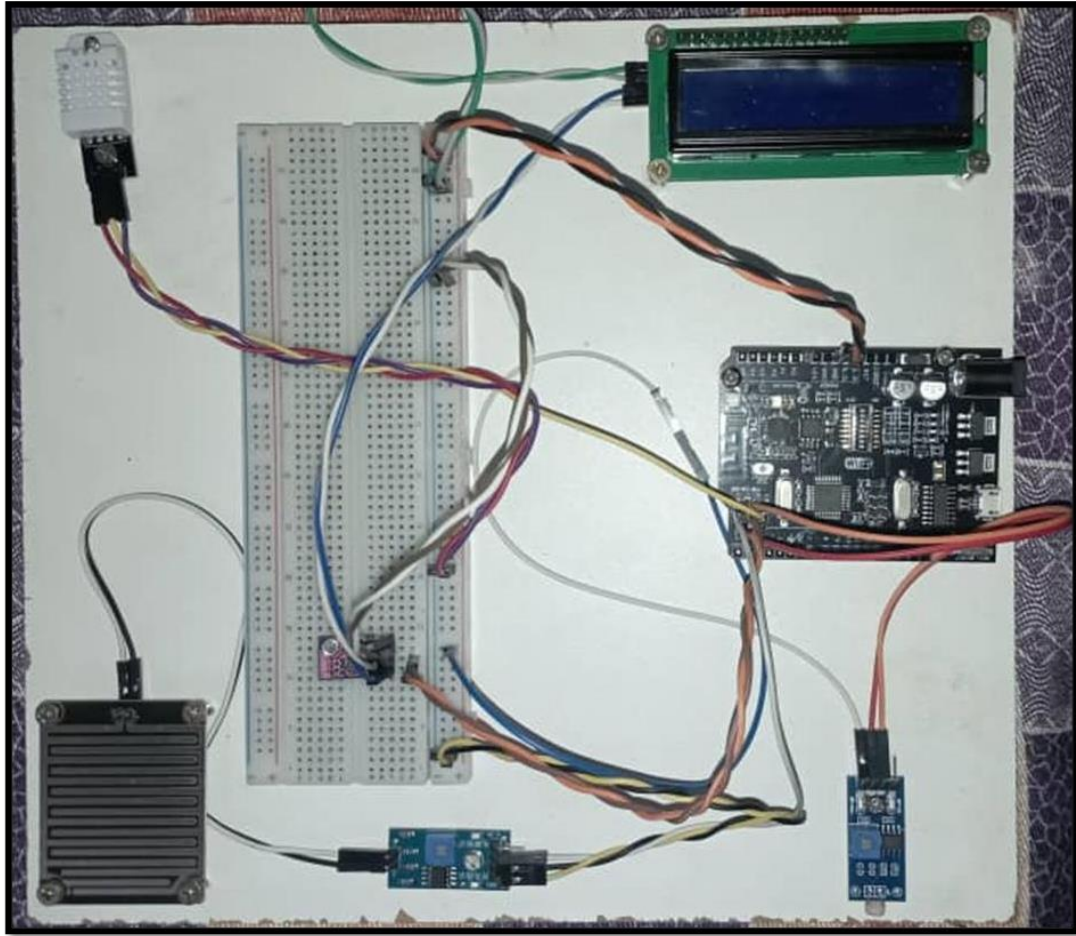


Figure 2.16 : Prototype physique de la station météorologique.

II.8 Calibration et protection environnementale

Afin d'améliorer la qualité des mesures et la fiabilité du système, une phase de calibration des capteurs a été réalisée. Des dispositions particulières ont également été prévues pour protéger les composants contre les conditions environnementales extérieures.

II.8.1 Calibration des capteurs

Pour le capteur de pluie V2, les seuils de détection ont été déterminés expérimentalement en humidifiant progressivement la plaque sensible et en observant l'évolution des valeurs mesurées. Ces essais ont permis d'identifier différents états correspondant à l'absence de pluie, à une pluie faible, modérée ou intense. Ainsi, en notant les valeurs ADC correspondantes, le potentiomètre du module est réglé de sorte que la sortie DO bascule à 600 (seuil moyen). [29]

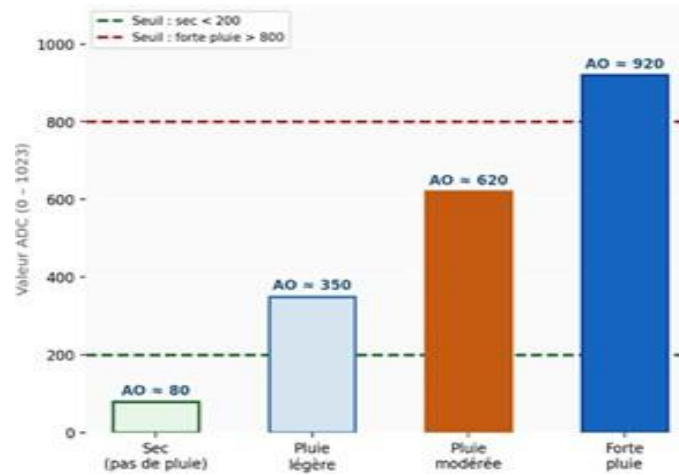


Figure 2.17 : Courbe de calibration du capteur de pluie V2.

Pour le module LDR5516, quatre points de référence ont été relevés expérimentalement :

Condition	Éclairement (lux)	ADC	Valeur %
Obscurité	~0	~1000	~5 %
Intérieur	~100	~820	~49 %
Extérieur ombragé	~1 000	~500	~80 %
Plein soleil	~10 000	~50	~98 %

Tableau 2.8 : Points de calibration du module LDR5516

Remarque : ces valeurs sont indicatives. Elles dépendent de l'âge du composant, de la température ambiante et de l'angle d'incidence de la lumière. Pour notre prototype, les relevés ont été effectués par temps clair à Oran (Algérie) en mai 2025, potentiomètre en position médiane. [30]

luminosity = map(analogRead(A0), 0, 1023, 100, 0);

II.8.2 Stratégie d'échantillonnage

L'acquisition des données repose sur une stratégie d'échantillonnage périodique permettant de concilier précision des mesures et économie des ressources matérielles. Les capteurs DHT22 et BME280 sont interrogés une fois par cycle d'acquisition, tandis que plusieurs lectures successives sont effectuées pour les modules LDR5516 et Pluie V2 afin d'obtenir une valeur moyenne plus représentative.

Les données collectées sont ensuite transmises périodiquement vers la plateforme ThingSpeak. Cette approche permet de réduire la charge de communication tout en garantissant une surveillance régulière des paramètres météorologiques.

La boucle principale repose sur `millis()` pour éviter tout appel bloquant à `delay()`. Le cycle d'acquisition dure 10 secondes [31] les données sont transmises vers ThingSpeak toutes les 30 secondes, après trois cycles.

Capteur	Lectures/cycle	Traitement	Envoi
DHT22	1	Directe	30 s
BMP280	1	Directe	30 s
LDR5516	10	Moyenne	30 s
Pluie V2	10	Moyenne	30 s
LCD	1	Affichage	Continu

Tableau 2.9 : Fréquences d'échantillonnage des différents capteurs.

II.8.3 Protection environnementale

Le fonctionnement en extérieur impose la mise en place de mesures de protection adaptées afin de préserver la fiabilité des capteurs et des circuits électroniques. Le boîtier principal est conçu pour offrir une protection contre les poussières et les projections d'eau, avec un niveau de protection recommandé de type IP54.

Les capteurs de température et d'humidité sont placés dans un abri ventilé afin de limiter l'influence du rayonnement solaire direct sur les mesures. Le capteur de pluie est quant à lui installé avec une légère inclinaison afin de favoriser l'écoulement de l'eau et d'éviter l'accumulation de dépôts susceptibles d'altérer son fonctionnement.

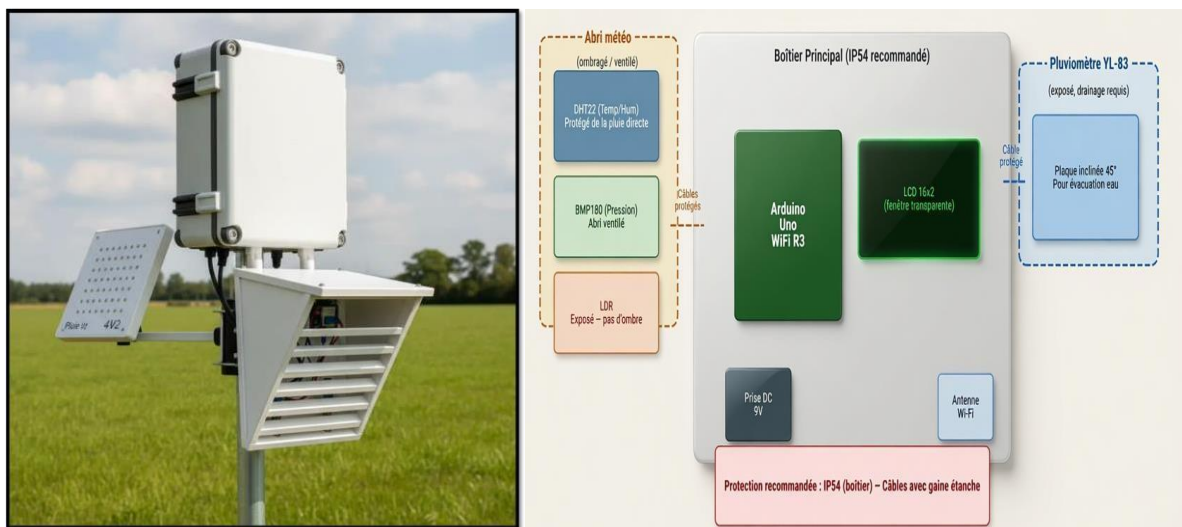


Figure 2.19 : Dispositif de protection environnementale des composants (à gauche), et Recommandations de placement et de protection des capteurs (à droite).

II.9 Conclusion

Cette partie a présenté l'intégration matérielle de la station météorologique ainsi que les principales étapes de validation du prototype. Les schémas de câblage, la réalisation pratique du système et les opérations de calibration ont permis de confirmer le bon fonctionnement des différents modules retenus. Les mesures de protection environnementale adoptées contribuent également à améliorer la robustesse et la fiabilité du dispositif dans des conditions réelles d'utilisation. Ces résultats constituent une base solide pour le développement logiciel et l'exploitation des données météorologiques présentés dans le chapitre suivant.

CHAPITRE 03

Développement logiciel et Firewall

III.1 Introduction

Après avoir présenté l'architecture matérielle de la station météorologique connectée, ce chapitre est consacré au développement logiciel du système. Le programme embarqué, exécuté sur le module ESP8266, assure l'acquisition des données issues des différents capteurs, leur traitement, leur affichage local ainsi que leur transmission vers les plateformes de supervision à distance.

Afin de garantir un fonctionnement fiable et efficace, le firmware a été conçu pour gérer simultanément les opérations de mesure, de communication et de surveillance en temps réel. Ce chapitre présente la modélisation du système, l'architecture du programme développé, ainsi que les mécanismes de communication mis en œuvre pour l'envoi et la visualisation des données météorologiques.

III.2 Modélisation fonctionnelle du système

Avant d'aborder l'implémentation logicielle de la station météorologique connectée, il est nécessaire de modéliser le fonctionnement global du système. Cette étape permet de représenter de manière structurée les interactions entre l'utilisateur, les capteurs, le microcontrôleur et les différentes plateformes de supervision.

La modélisation a été réalisée à l'aide du langage UML (Unified Modeling Language), largement utilisé pour la conception et la documentation des systèmes informatiques et embarqués. Les diagrammes présentés dans cette section décrivent les principales fonctionnalités du système sous différents angles : temporel (séquence) et comportemental (activité).

III.2.1 Le diagramme de séquence

Le diagramme de séquence présenté à la figure 3.1 décrit l'enchaînement temporel des opérations réalisées par la station météorologique lors d'un cycle de fonctionnement complet. Il met en évidence les interactions entre l'utilisateur, l'unité de traitement, les capteurs ainsi que les différents services de supervision et de communication, à travers les tâches suivantes :

1. Le processus débute par l'initialisation du système à partir de l'interface de supervision. Cette étape active les différents modules logiciels chargés de coordonner les opérations d'acquisition, de traitement et de transmission des données météorologiques.
2. La phase suivante correspond à l'acquisition des mesures provenant des différents capteurs intégrés à la station. Les paramètres environnementaux, tels que la température, l'humidité relative, la pression atmosphérique, la luminosité et les précipitations, sont collectés puis transmis au module de traitement.
3. Les données acquises font ensuite l'objet d'une phase de traitement visant à convertir les signaux mesurés en valeurs physiques exploitables. Une vérification de cohérence est également réalisée afin d'identifier d'éventuelles situations anormales. Lorsque certaines mesures dépassent les seuils prédéfinis, le système génère automatiquement une alerte destinée à l'utilisateur.

4. Après traitement, les informations sont diffusées simultanément vers les différents services du système. Les données sont d'une part affichées localement sur l'écran LCD afin de fournir une visualisation immédiate des conditions météorologiques et, d'autre part, transmises vers les plateformes IoT pour leur stockage, leur consultation et leur analyse à distance.

Enfin, en cas d'anomalie ou de dépassement de seuil, une notification est envoyée à l'interface de supervision afin d'informer rapidement l'utilisateur de l'événement détecté. Ce mécanisme garantit une surveillance continue et une réaction rapide face aux variations importantes des paramètres environnementaux.

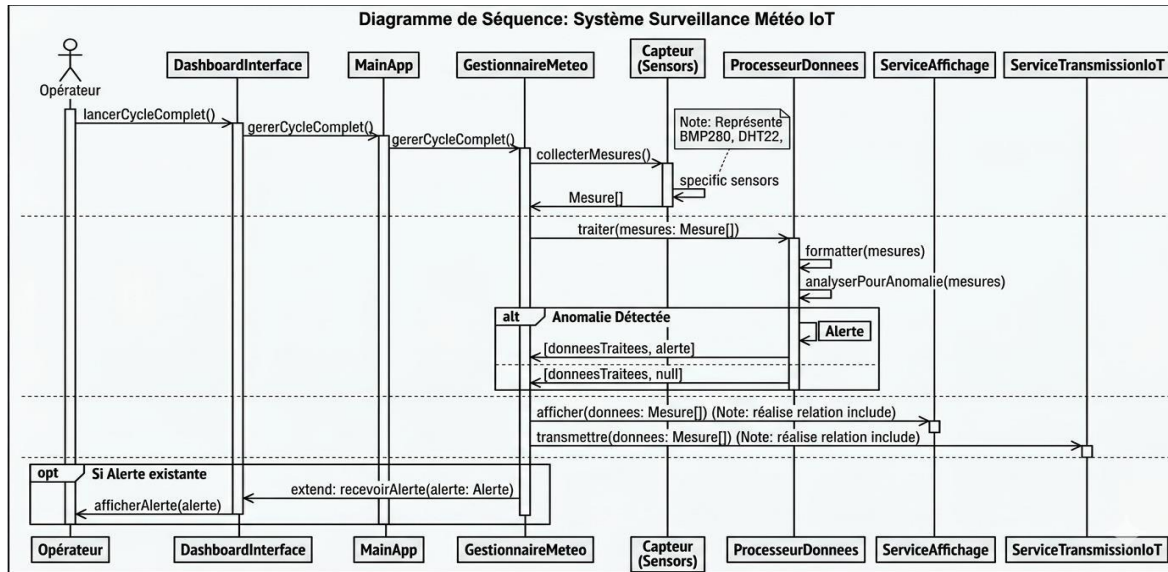


Figure 3.1 : Diagramme de séquence du système de surveillance météorologique IoT.

III.2.2 Diagramme d'activité

Le diagramme d'activité présenté à la figure 3.2 décrit le déroulement opérationnel de la station météorologique connectée ainsi que les différentes étapes exécutées au cours d'un cycle de surveillance. Il permet de représenter les flux de données et les décisions prises par le système depuis l'acquisition des mesures jusqu'à leur exploitation par l'utilisateur, à travers les tâches suivantes :

1. Le fonctionnement débute par l'initialisation des différents modules matériels et logiciels du système. Une fois cette étape achevée, les capteurs environnementaux sont sollicités afin de mesurer les principaux paramètres météorologiques, notamment la température, l'humidité relative, la pression atmosphérique, la luminosité et les précipitations.
2. Les données acquises sont ensuite traitées localement par le microcontrôleur afin d'éliminer les éventuelles erreurs de mesure et de préparer leur transmission. Les informations validées sont alors envoyées vers les plateformes de supervision et simultanément affichées sur l'interface locale du système.

3. Une étape de contrôle est ensuite réalisée afin de comparer les valeurs mesurées aux seuils prédéfinis. Lorsque les paramètres restent dans les plages de fonctionnement normales, le système poursuit son cycle de surveillance. En revanche, lorsqu'un dépassement de seuil est détecté, une alerte est générée afin d'informer l'utilisateur de la situation observée.

4. Après l'exécution des opérations de traitement et de communication, le système entre dans une phase d'attente avant le démarrage d'un nouveau cycle d'acquisition. Cette organisation cyclique permet d'assurer une surveillance continue tout en optimisant l'utilisation des ressources matérielles et énergétiques.

Le diagramme met également en évidence la possibilité de modifier certains paramètres de fonctionnement, notamment les seuils d'alerte. Ces paramètres peuvent être ajustés par l'utilisateur à travers les interfaces de supervision, puis synchronisés avec le système afin d'adapter le comportement de la station aux conditions d'exploitation souhaitées.

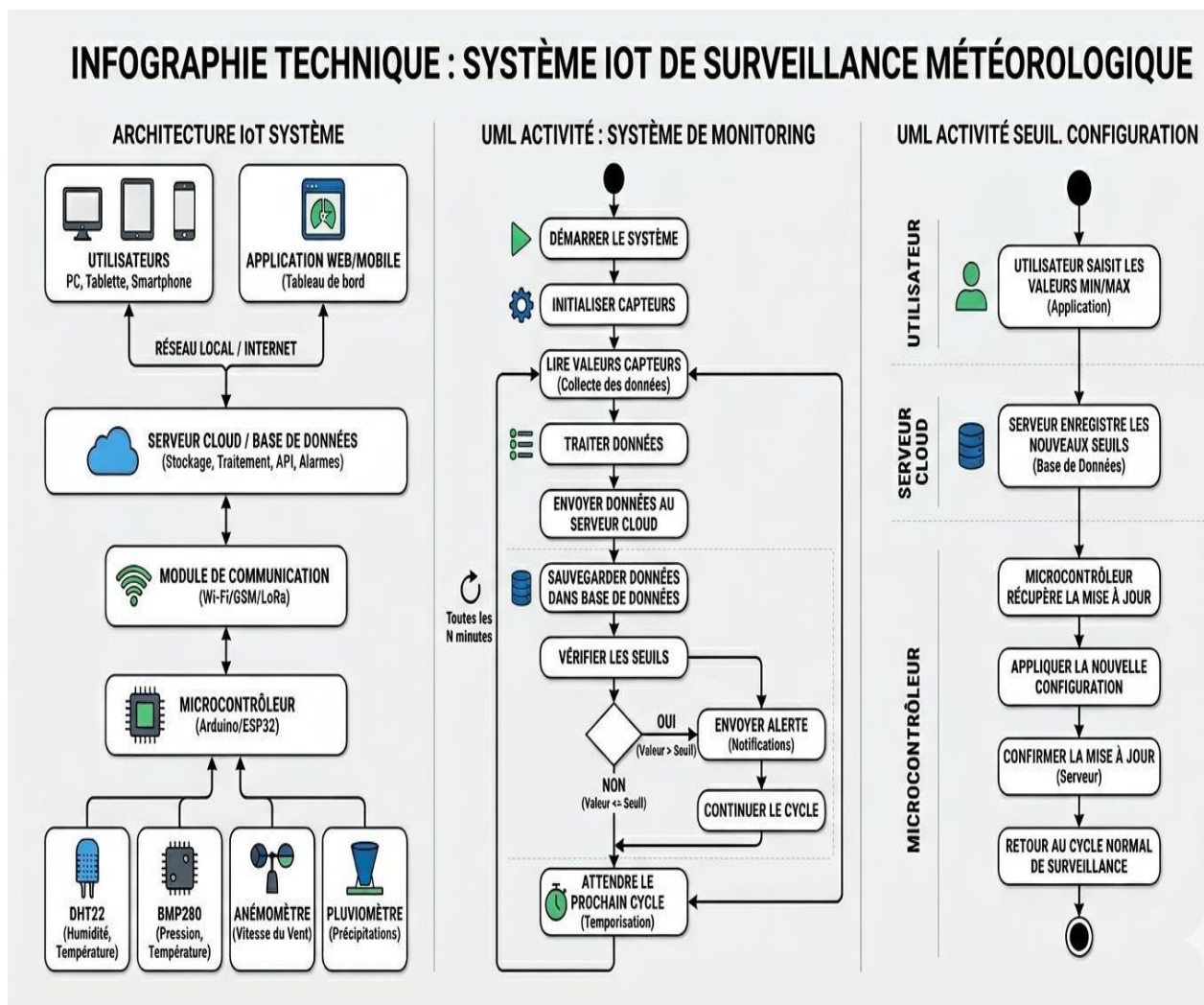


Figure 3.2 : Diagramme d'activité du système de surveillance météorologique connectée.

III.3 Environnement de développement Arduino IDE

Le développement logiciel de la station météorologique a été réalisé à l'aide de l'environnement Arduino IDE, largement utilisé pour la programmation des systèmes embarqués basés sur les cartes Arduino et les microcontrôleurs ESP8266.

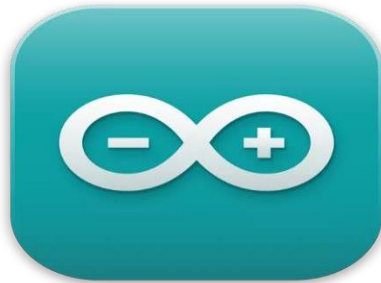


Figure 3.3 : Sigle de l'Arduino IDE

III.3.1 Définition

Arduino IDE constitue l'environnement principal utilisé pour le développement du firmware embarqué. Il permet la création, la compilation et le téléversement du programme vers les cartes Arduino. Son interface simple ainsi que la disponibilité d'un grand nombre de bibliothèques facilitent l'intégration des différents capteurs et modules de communication utilisés dans un système embarqué. Nous pouvons donc saisir notre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte Arduino.

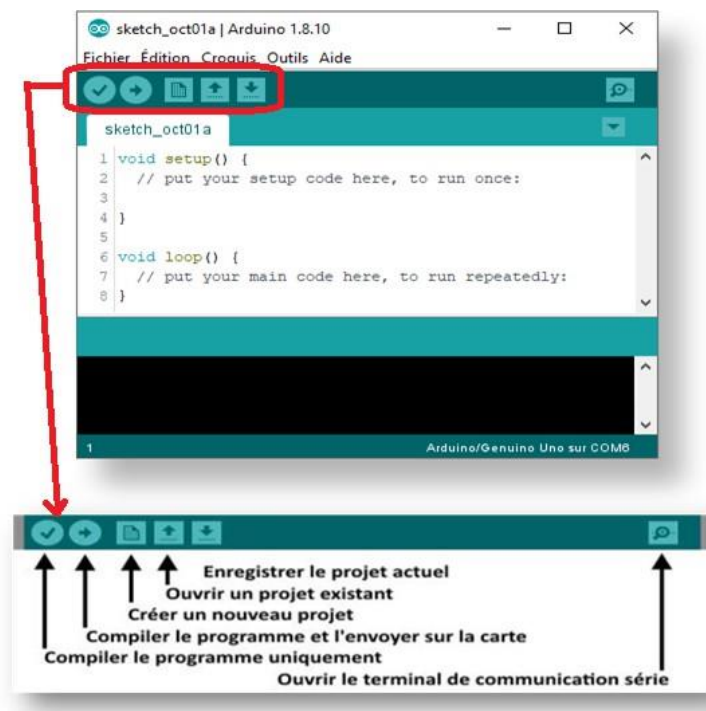


Figure 3.4 : Interface graphique du logiciel Arduino IDE [32]

L'outil intègre également un moniteur série permettant de visualiser les données acquises en temps réel et de faciliter les opérations de test et de débogage du système.

III.3.2 Structure et langage d'un programme en Arduino [32] :

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 3.4).

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 */.

III.3.3 Utilisation de l'IDE dans notre projet

L'environnement IDE nous a offert les outils nécessaires à l'écriture, à la compilation et au téléversement du programme embarqué vers la carte Arduino Uno WiFi R3.

Afin de faciliter la mise en œuvre des différentes fonctionnalités du système, plusieurs bibliothèques spécialisées ont été utilisées pour la gestion des capteurs, de l'afficheur LCD, de la communication Wi-Fi ainsi que des plateformes IoT.

Le tableau 3.1 présente les principales bibliothèques utilisées ainsi que leur rôle au sein du système :

Bibliothèque	Fonction
DHT.h	Lecture du capteur DHT22
Adafruit_BME280.h	Gestion du capteur BME280
LiquidCrystal_I2C.h	Pilotage de l'afficheur LCD
ESP8266WiFi.h	Connexion au réseau Wi-Fi
Wire.h	Communication I2C
BlynkSimpleEsp8266.h	Communication avec Blynk
ESP8266HTTPClient.h	Transmission HTTP vers ThingSpeak

Tableau 3.1 : Principales bibliothèques utilisées et leur rôle au sein du système.

III.4 Conception du firmware embarqué

Le firmware constitue le cœur logiciel de la station météorologique connectée. Il assure la coordination des différentes opérations réalisées par le système, notamment l'acquisition des données issues des capteurs, leur traitement, leur affichage local ainsi que leur transmission vers les plateformes de supervision à distance.

Le programme a été développé selon une architecture logicielle modulaire permettant de séparer les différentes fonctions du système. Cette organisation facilite la maintenance du code, améliore sa lisibilité et favorise l'évolution future de la station par l'ajout de nouveaux capteurs ou de nouvelles fonctionnalités.

III.4.1 Algorithme général de fonctionnement

Le fonctionnement de la station météorologique repose sur une boucle d'exécution continue permettant de réaliser périodiquement les opérations d'acquisition, de traitement et de transmission des données.

La figure 3.5 présente l'organigramme général du firmware développé :

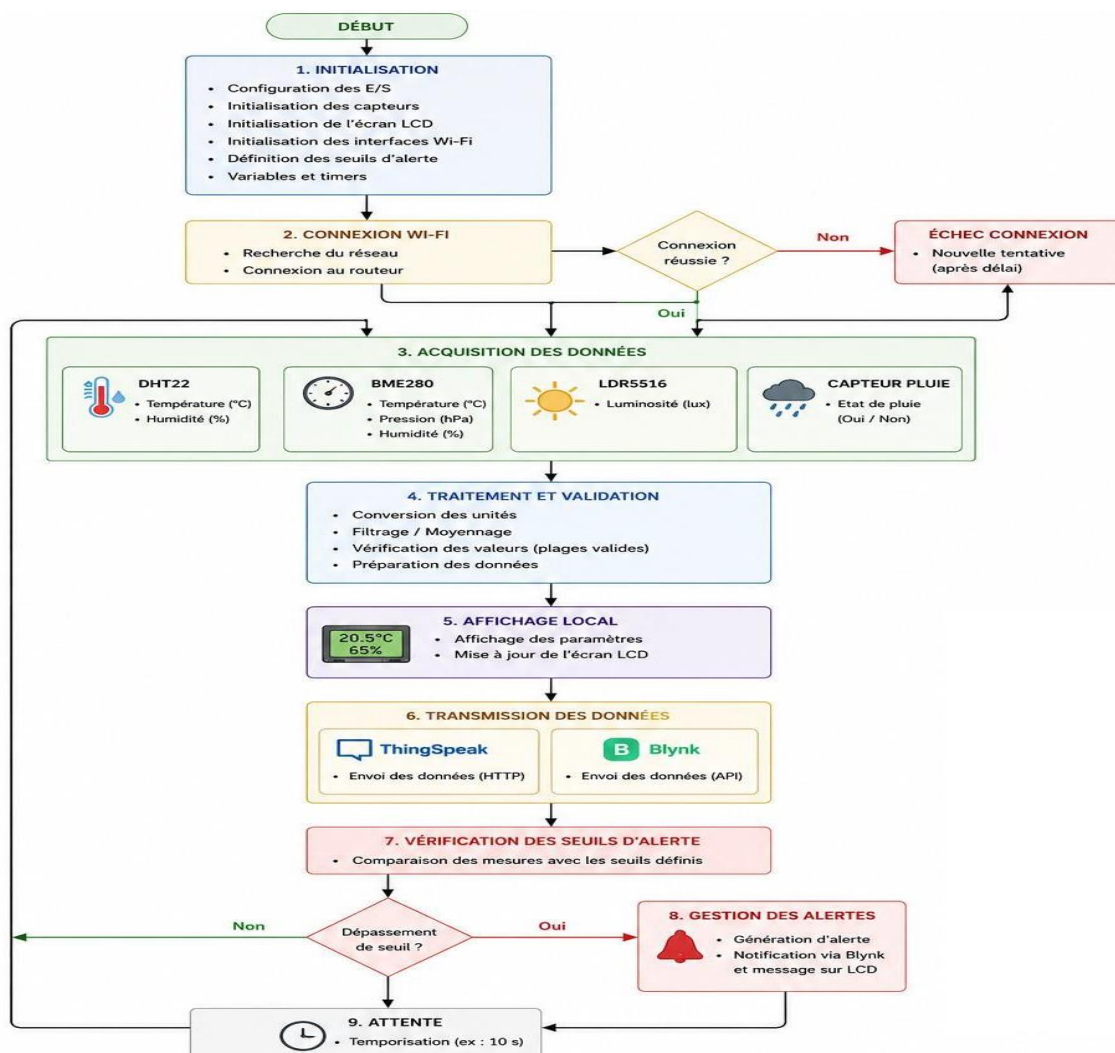


Figure 3.5 : Organigramme du Firmware embarqué adopté pour notre projet.

Après l'initialisation des périphériques matériels et l'établissement de la connexion réseau, le système interroge successivement les différents capteurs afin de collecter les paramètres météorologiques.

Les mesures obtenues sont ensuite traitées afin d'éliminer les éventuelles incohérences et de préparer les données destinées à l'affichage et à la transmission. Les résultats sont affichés localement sur l'écran LCD puis envoyés vers les plateformes ThingSpeak et Blynk pour le stockage et la supervision à distance.

Une vérification des seuils est également effectuée à chaque cycle afin de détecter d'éventuelles situations anormales. En cas de dépassement des limites définies, une notification est générée et transmise à l'utilisateur.

III.4.2 Acquisition des données météorologiques

L'acquisition des données constitue la première étape du fonctionnement de la station météorologique. Elle consiste à interroger périodiquement les différents capteurs afin de mesurer les paramètres environnementaux nécessaires à la surveillance des conditions météorologiques.

Le capteur DHT22 fournit les mesures de température et d'humidité relative, tandis que le BME280 assure la mesure de la pression atmosphérique. La luminosité est évaluée à l'aide du module LDR5516 et les précipitations sont détectées par le capteur de pluie V2. Les données collectées sont ensuite transmises au module de traitement afin d'être validées avant leur exploitation.

III.4.3 Algorithme de moyennage

Afin d'améliorer la stabilité des mesures de luminosité, un algorithme de moyennage a été appliqué aux données issues du capteur LDR5516. À chaque cycle d'acquisition, cinq lectures successives sont réalisées avec un intervalle de 10 ms entre deux mesures consécutives. La valeur finale de luminosité correspond à la moyenne arithmétique de ces cinq échantillons.

Cette méthode permet de réduire l'influence des fluctuations momentanées de l'éclairement, telles que le passage d'un nuage ou les variations rapides de la lumière ambiante. La durée totale du processus de moyennage est de 50 ms, ce qui reste négligeable devant la période d'acquisition de 10 s retenue pour le système.

En revanche, le capteur de pluie V2 fournit directement une information numérique issue du comparateur intégré au module. Son état logique indique immédiatement la présence ou l'absence de précipitations selon le seuil réglé par le potentiomètre du capteur. Une seule lecture est donc suffisante et aucun traitement de moyennage n'est nécessaire.

La figure 3.6 illustre le principe de traitement appliqué aux mesures de luminosité avant leur exploitation par le système.

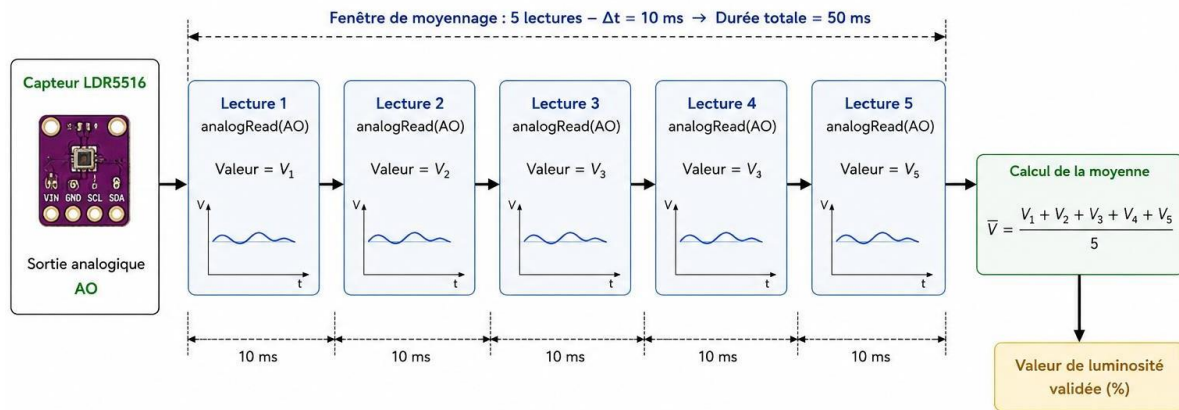


Figure 3.6 : Principe de moyennage des mesures acquises par le capteur LDR5516.

III.4.4 Gestion des erreurs capteurs

Afin de garantir la fiabilité des données transmises par la station météorologique, plusieurs mécanismes de détection et de correction des erreurs ont été intégrés au firmware.

Pour le capteur DHT22, les mesures acquises sont systématiquement vérifiées avant leur utilisation. En cas d'échec de lecture, la dernière valeur valide est conservée afin d'assurer la continuité du fonctionnement. Si plusieurs erreurs consécutives sont détectées, le capteur est automatiquement réinitialisé.

Le système surveille également l'état de la connexion Wi-Fi. Lorsqu'une déconnexion est détectée, une procédure de reconnexion est automatiquement déclenchée. Les données ne sont transmises vers les plateformes IoT que lorsque la liaison réseau est disponible.

Les valeurs analogiques acquises sont également contrôlées afin d'écarter les mesures incohérentes susceptibles de résulter d'un dysfonctionnement matériel ou d'une perturbation temporaire. Cette stratégie permet d'améliorer la robustesse du système et de garantir une meilleure qualité des informations collectées.

Le tableau 3.2 présente les principales erreurs susceptibles d'être rencontrées ainsi que les actions correctives mises en œuvre dans le firmware.

Capteur	Type d'erreur	Action corrective dans le code
DHT22	isnan(t) ou isnan(h)	Conserver la valeur du cycle précédent après 3 échecs : dht.begin()
BMP280	Absent à l'adresse 0x76	Mode dégradé : utiliser la T° DHT22 uniquement
Wi-Fi	WL_DISCONNECTED	Blynk gère la reconnexion automatiquement ThingSpeak est ignoré

Capteur	Type d'erreur	Action corrective dans le code
CallMeBot	HTTP code $\neq$ 200	Log dans le moniteur série — pas de retry (évite le spam WA)
MQTT	!client.connected()	reconnect() avec backoff
ADC	Valeur > 1023	Rejeter , conserver lecture précédente

Tableau 3.2 : Stratégie de gestion des erreurs intégrée au firmware.

Note importante : Tous les codes associés à l'implémentation du Firmware, ainsi que la corp générale du programme IDE associé au projet sont regroupés en annexe A.

De plus, le capteur BMP280 (pression atmosphérique) a été retiré de la version finale. Les raisons techniques et les résultats obtenus sans ce capteur sont présentés au chapitre 4. Par ailleurs, la sortie analogique du module Pluie V2 n'est pas exploitée : l'ESP8266 ne dispose que d'un seul convertisseur ADC (broche TOUT), déjà utilisé par le module LDR5516. La détection de pluie repose donc exclusivement sur la sortie numérique DO (seuil réglé par le potentiomètre du module)

III.5 Plateforme ThingSpeak : archivage et analyse des données

ThingSpeak est une plateforme IoT développée par MathWorks permettant la collecte, le stockage et la visualisation de données en temps réel. Dans le cadre de ce projet, elle est utilisée pour assurer l'archivage des mesures météorologiques acquises par la station et leur consultation à distance à travers une interface web.

Les données transmises par la carte Arduino Uno WiFi R3 sont enregistrées dans un canal (Channel) dédié contenant plusieurs champs correspondant aux différents paramètres surveillés. ThingSpeak reçoit les données du système toutes les 30 secondes et les stocke dans un channel à cinq champs. C'est la mémoire longue du projet : les données y sont conservées indéfiniment, exportables en CSV pour le chapitre 4, et visualisables sous forme de graphiques temporels directement depuis le navigateur. Cette approche permet de conserver un historique complet des mesures et de générer automatiquement des graphiques illustrant l'évolution temporelle des grandeurs observées.

L'utilisation de ThingSpeak présente, alors, plusieurs avantages pour ce type d'application, comme :

- ✓ La simplicité de mise en œuvre,
- ✓ Visualisation en temps réel,
- ✓ Exportation des données au format CSV
- ✓ Intégration native avec les outils d'analyse de MathWorks.



Figure 3.7 : Captures d'écran de la configuration du canal ThingSpeak utilisé dans notre projet.

III.5.1 Configuration du canal ThingSpeak

Après la création d'un compte utilisateur, nous avons configuré un canal (channel) de communication afin de recevoir les données transmises par la station météorologique, comme

le montre la figure 3.7. Chaque grandeur mesurée est associée à un champ (field) spécifique du canal, facilitant ainsi l'organisation et l'exploitation des informations collectées.

Dans la version finale du prototype, quatre champs sont utilisés pour le stockage des données météorologiques : la température, l'humidité relative, la luminosité et l'état de détection des précipitations. Les mesures sont enregistrées périodiquement et deviennent immédiatement accessibles sous forme de tableaux ou de graphiques temporels.

Le tableau 3.3 présente l'affectation des différents champs du canal ThingSpeak.

Champ	Paramètre mesuré	Unité
Field 1	Température	°C
Field 2	Humidité relative	%
Field 3	Luminosité	%
Field 4	Détection de pluie	0 / 1

Tableau 3.3 : Configuration des champs du canal ThingSpeak

Remarque : Le Field 3, initialement réservé à la pression atmosphérique, n'est pas utilisé suite au retrait du capteur BMP280.

Après sauvegarde, l'onglet API Keys fournit la Write API Key , c'est la clé d'authentification que le Firmware utilise pour chaque envoi.

III.5.2 Envoi des données vers ThingSpeak

L'envoi des données vers ThingSpeak est réalisé périodiquement à travers la connexion Wi-Fi intégrée à la carte Arduino Uno WiFi R3. Les mesures acquises par les capteurs sont regroupées puis transmises vers le canal ThingSpeak à l'aide d'une requête HTTP contenant la clé d'authentification du projet.

Dans notre projet, nous avons adopté une requête HTTP GET, en raison de sa simplicité de mise en œuvre et du faible volume de données transmis. Les paramètres météorologiques sont envoyés sous forme de champs associés au canal ThingSpeak : température, humidité relative, luminosité et état de pluie. Après réception, la plateforme enregistre automatiquement les données et les affiche sous forme de graphiques temporels.

Cette méthode est plus lisible qu'une requête POST et suffit pour le volume de données produit (une trame de 120 octets toutes les 30 secondes) [33]. Elle permet donc d'assurer un archivage continu des mesures tout en offrant une visualisation à distance accessible depuis une interface web.

III.5.3 Alertes React automatisées

ThingSpeak offre également une fonctionnalité appelée React permettant de déclencher automatiquement une action lorsqu'une condition définie est vérifiée. Dans le cadre de ce projet, nous avons tiré profit de cette option pour surveiller certains paramètres météorologiques critiques, tels qu'une température élevée ou la détection de fortes précipitations.

Le Tableau 3.4 présente les principales règles React configurées pour la surveillance automatique des conditions météorologiques.

Règle React	Condition	Action ThingSpeak
Alerte chaleur	Température > 38 °C pendant deux mesures successives	Notification par e-mail ou action ThingHTTP
Alerte pluie	Pluie détectée pendant deux mesures successives	Déclenchement d'une notification ou d'un webhook

Tableau 3.4 : Règles d'alerte ThingSpeak React configurées.

III.6 Application mobile Blynk IoT

Afin d'assurer une supervision à distance de la station météorologique, nous avons intégré la plateforme Blynk IoT au système développé. Cette plateforme permet de créer rapidement des interfaces graphiques interactives destinées au suivi en temps réel des données acquises par les capteurs.

Blynk répond alors au problème que ThingSpeak ne résout pas : l'accès mobile rapide. L'interface web de ThingSpeak est conçue pour un écran d'ordinateur et demande plusieurs gestes de navigation. Blynk est une application native iOS/Android dont le dashboard s'affiche immédiatement à l'ouverture, avec des valeurs qui se rafraîchissent toutes les 10 secondes sans action de l'utilisateur. [34]

III.6.1 Présentation de la plateforme Blynk IoT

Blynk IoT est une plateforme de développement dédiée aux applications de l'Internet des Objets (IoT). Elle permet d'établir une communication bidirectionnelle entre les objets connectés et une interface utilisateur accessible depuis un smartphone ou un navigateur web.

Dans ce projet, Blynk est utilisé comme outil de supervision en temps réel. Les données acquises par la station météorologique sont transmises via le réseau Wi-Fi puis affichées sur une interface graphique conviviale composée de différents widgets de visualisation.

Donc, grâce à l'application mobile Blynk, l'utilisateur peut consulter instantanément les différents paramètres météorologiques, tels que la température, l'humidité relative, la luminosité et l'état des précipitations, à partir d'un smartphone connecté à Internet. Cette

approche améliore l'accessibilité du système et facilite la surveillance des conditions environnementales à distance.

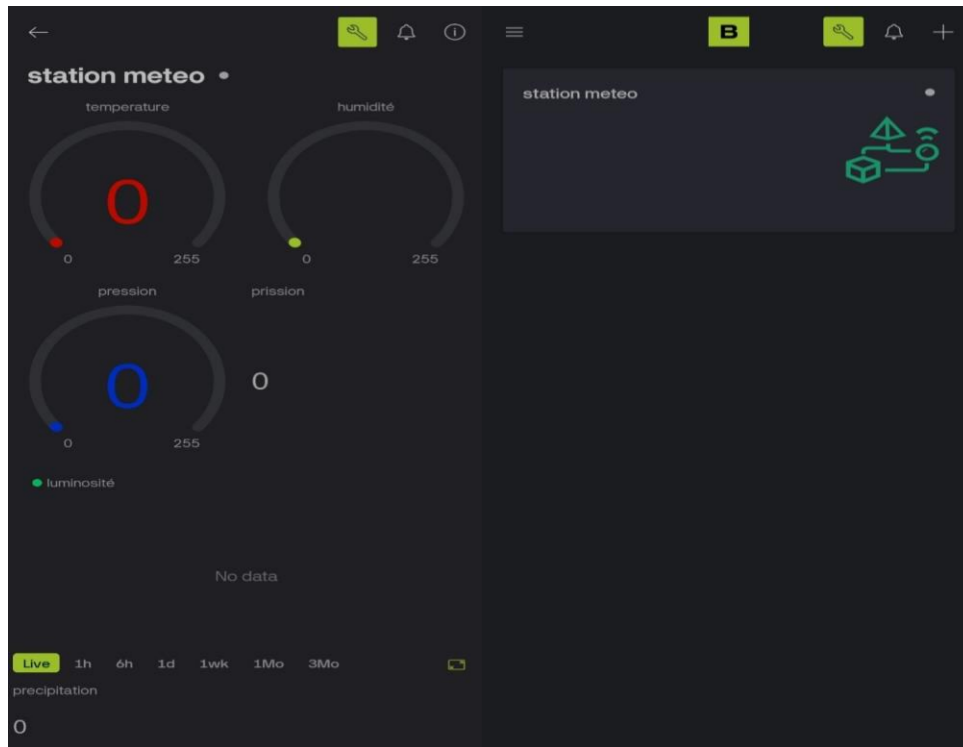


Figure 3.8 : Capture d'écran de l'interface de la plateforme Blynk IoT utilisé dans notre projet.

III.6.2 Configuration des Virtual Pins

La communication entre le firmware embarqué et l'application Blynk repose sur l'utilisation de broches virtuelles (Virtual Pins). Chaque paramètre météorologique est associé à une broche virtuelle spécifique permettant de transmettre les données vers l'interface de supervision.

Cette approche simplifie l'échange d'informations entre la station météorologique et l'application mobile tout en offrant une grande flexibilité pour l'ajout de nouvelles fonctionnalités.

Dans notre projet, Blynk communique avec l'Arduino via un concept de Virtual Pins des canaux logiques numérotés V0 à V4 qui transportent les valeurs des capteurs vers l'application. Chaque appel à `Blynk.virtualWrite()` envoie une valeur vers le widget correspondant sur le téléphone. Les affectations des principales broches virtuelles sont illustrées dans le tableau 3.5 :

Virtual Pin	Paramètre	Widget recommandé	Configuration (plage)
V0	Température (°C)	Gauge(0-80)	0 à 80 °C couleur rouge au-dessus de 35
V1	Humidité (%)	Gauge(0-100)	0 à 100 % couleur bleue
V2	Luminosité (%)	Gauge(0-100)	0 à 100 % historique 24h
V3	Pluie (0=sec / 1=pluie)	LED	0 = éteint (sec) / 1 = allumé (pluie)

Tableau 3.5: Correspondence Virtual Pins / widgets Blynk

III.6.3 Interface de supervision développée

L'interface de supervision développée sous Blynk a été conçue afin d'offrir une visualisation claire et intuitive des paramètres météorologiques mesurés par la station. Elle regroupe différents widgets permettant l'affichage en temps réel des grandeurs acquises ainsi que l'état général du système.

Les informations sont actualisées automatiquement à chaque cycle de transmission, offrant ainsi à l'utilisateur une vision instantanée des conditions environnementales. Cette interface constitue un outil efficace pour le suivi et la surveillance à distance des données météorologiques.



Figure 3.9 : Interface mobile Blynk développée pour la supervision de la station météorologique connectée.

III.7 Sécurité des échanges

La sécurité constitue un aspect important des systèmes IoT, notamment lorsqu'ils reposent sur des communications sans fil et des services cloud. Dans le cadre de ce projet, trois mécanismes ont été mis en œuvre afin d'assurer un niveau de protection adapté aux contraintes matérielles du prototype développé :

1. Au niveau du réseau local, la communication entre la carte Arduino Uno WiFi R3 et le point d'accès Wi-Fi est sécurisée à l'aide du protocole WPA2-PSK associé au chiffrement AES. Ce mécanisme permet d'empêcher les accès non autorisés au réseau et garantit la confidentialité des échanges sur le lien radio.
2. Les données météorologiques sont transmises vers la plateforme ThingSpeak à travers des requêtes HTTP. Le choix du protocole HTTP a été privilégié afin de limiter la consommation mémoire du module ESP8266.
3. Enfin, l'accès en écriture au canal ThingSpeak est protégé par une clé d'authentification (Write API Key) propre au projet. Cette clé permet d'identifier la station météorologique et d'empêcher toute modification non autorisée des données stockées sur la plateforme.

Le tableau 3.6 récapitule les principaux mécanismes de sécurité intégrés au système développé.

Couche	Mécanisme de sécurité	Rôle
Réseau local	WPA2-PSK / AES	Sécurisation des échanges Wi-Fi entre la station et le routeur
Transport Internet	HTTP avec authentification API	Transmission des données vers ThingSpeak
Application Cloud	Write API Key	Protection de l'accès en écriture au canal ThingSpeak

Tableau 3.6 : Mécanismes de sécurité mis en œuvre dans la station météorologique connectée.

III.8 Conclusion :

Ce chapitre explique comment le matériel du chapitre 2 évolue pour devenir un système de communication. La principale décision architecturale consiste à distribuer les données sur trois canaux complémentaires : le LCD pour un accès local instantané, Blynk pour la surveillance en temps réel via mobile, et ThingSpeak pour l'archivage et l'analyse des données. La fonctionnalité WhatsApp, développée et testée, a été retirée de la version déployée finale.

Le chapitre 4 présente des tests expérimentaux de ce système. La question posée n'est plus simplement 'comment le système fonctionne-t-il ?' mais plutôt 'quelle est la précision de ses mesures comparée à une référence externe ?'

CHAPITRE 04

Tests, validation et résultats

IV.1 Introduction

Après avoir présenté, dans les chapitres 2 et 3, la conception matérielle de la station météorologique connectée ainsi que le développement de son firmware et des plateformes de supervision IoT, il est nécessaire d'évaluer les performances réelles du système développé.

Le présent chapitre est consacré à la validation expérimentale du prototype que nous avons réalisé. Il a pour objectif de vérifier le bon fonctionnement des différents composants, d'évaluer la fiabilité des mesures acquises et d'analyser le comportement global de la station dans des conditions réelles d'utilisation.

Les essais effectués portent sur les tests unitaires des capteurs, les tests d'intégration du système complet, l'analyse des données météorologiques mesurées ainsi que leur comparaison avec des données de référence externes. Enfin, une étude des performances globales du prototype et des principales contraintes rencontrées au cours du développement est présentée.

IV.2 Tests unitaires des capteurs

Avant l'intégration des différents modules au sein de la station météorologique, des tests unitaires ont été réalisés sur chaque capteur afin de vérifier individuellement leur bon fonctionnement. Cette étape permet d'identifier d'éventuelles anomalies propres à chaque composant et d'évaluer leurs performances dans des conditions expérimentales contrôlées.

Les essais ont porté sur le capteur DHT22, le module LDR5516 et le capteur de pluie V2. Pour chaque dispositif, des mesures ont été effectuées puis comparées à des critères d'acceptation préalablement définis. Les protocoles expérimentaux adoptés sont résumés dans le tableau 4.1 :

Capteur	Instrument de référence	Durée	Fréquence	Critère d'acceptation
DHT22	Thermomètre certifié + hygromètre	1 heure	10 min	$\Delta T < 0,5 \text{ }^{\circ}\text{C}$ $\Delta H < 3 \%$
LDR5516	Variation lumineuse contrôlée	30 min	5 min	Relation monotone ,cohérence ADC
Module Pluie V2	Simulation par dépôt de gouttes	20 min	Manuel	DO = LOW / PLUIE : DO = HIGH

Tableau 4.1 : Matériel de référence et critères d'acceptation par capteur [35] [36]

IV.2.1 Validation du bus I²C :

Avant l'intégration des composants utilisant le protocole I²C, un test de validation du bus de communication a été réalisé à l'aide d'un programme de détection (« I²C Scanner »). Ce test a permis de vérifier le bon fonctionnement des liaisons matérielles et de confirmer l'adresse des périphériques connectés.

Le scan effectué sur les broches GPIO4 (SDA) et GPIO12 (SCL) de l'ESP8266 a permis de détecter correctement l'afficheur LCD 1602 I²C à l'adresse 0x27, confirmant ainsi la validité du câblage et le bon fonctionnement du bus I²C.

Le tableau 4.2 présente les résultats obtenus lors du scan :

Composant	Adresse I2C détectée	Résultat
LCD 1602 I ² C	0x27	✓ Détecté et opérationnel

Tableau 4.2 : Résultats du scan I²C (GPIO4=SDA, GPIO12=SCL).

IV.2.2 Validation du capteur DHT22 (température et humidité)

Le capteur DHT22 a été soumis à une série d'essais expérimentaux afin d'évaluer sa sensibilité, sa stabilité et sa capacité à suivre les variations des conditions ambiantes. Les tests ont été réalisés dans différentes situations, notamment en conditions ambiantes, sous l'effet d'une source de chaleur et en présence d'une augmentation locale de l'humidité.

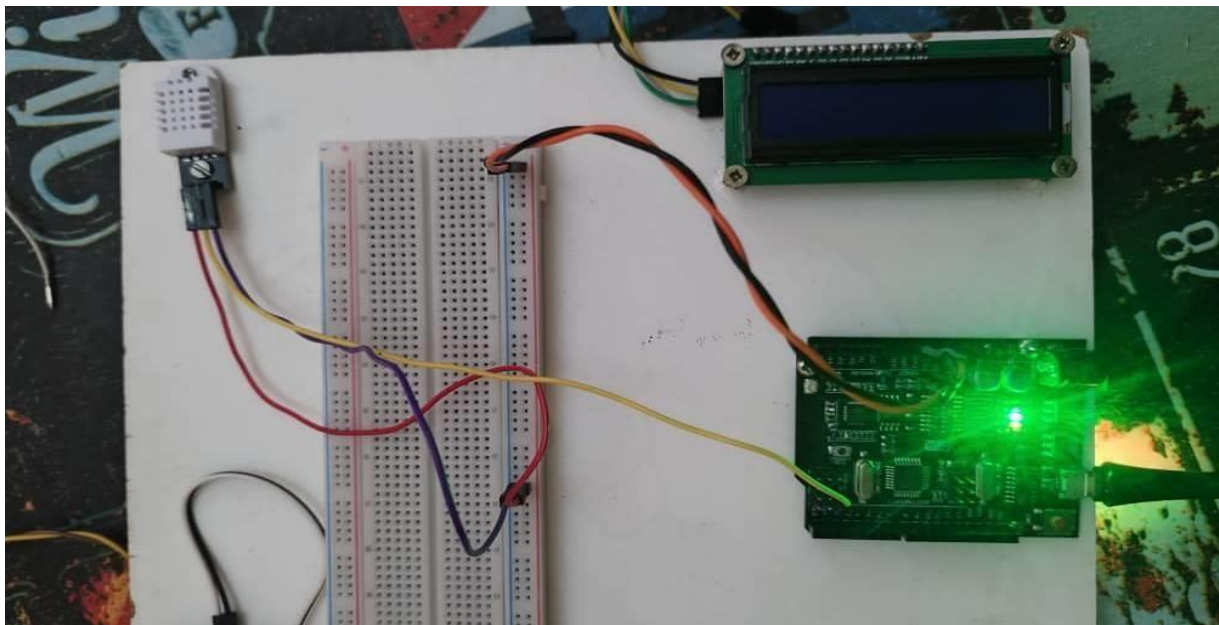


Figure 4.1 : Connexion du capteur DHT22

Le tableau 4.3 présente les résultats expérimentaux obtenus dans les différentes conditions de test.

Condition de test	Température mesurée	Humidité mesurée	Conformité
Température ambiante (repos)	22,6 °C	80,0 %	■ Nominal
Main posée sur le capteur	27,6 °C	68,4 %	■ Réactif
Haleine proche du capteur	23,8 °C	82,8 %	■ Réactif
Retour aux conditions ambiantes	23,0 – 23,2 °C	76 – 78 %	■ Stabilisé

Tableau 4.3 : Résultats des tests unitaires du capteur DHT22.

Les résultats obtenus montrent que le capteur réagit correctement aux variations thermiques et hygrométriques. Une augmentation de la température a été observée lors de l'application d'une source de chaleur, tandis qu'une hausse de l'humidité relative a été enregistrée lors d'une exposition à l'haleine. Après suppression des perturbations, les mesures sont revenues progressivement vers leurs valeurs initiales, traduisant un comportement stable du capteur.

Par ailleurs, le temps de réponse thermique du DHT22 a été estimé à environ 40 s, correspondant à quatre cycles d'acquisition successifs. La stabilité observée au repos, de l'ordre de $\pm 0,2$ °C pour la température et $\pm 0,5$ % pour l'humidité relative, confirme l'aptitude du capteur à la surveillance environnementale en temps réel.

Plage de température observée	22,6 °C – 27,6 °C	Plage nominale : -40 à +80 °C ✓
Plage d'humidité observée	68,4 % – 82,8 %	Plage nominale : 0 – 100 % ✓
Lectures invalides (is nan)	1 sur toute la session	< 3 lectures invalides/heure ✓
Temps de réponse thermique	40 s (4 cycles)	< 60 s ✓
Stabilité sur 10 cycles au repos	$\pm 0,2$ °C $\pm 0,5$ %	Spec. fabricant : $\pm 0,5$ °C $\pm 2-5$ % ✓

Tableau 4.4 : Indicateurs de performance DHT22

```

DHT22testdone.ino
1 // = بدون مقاومة خارجية DHT22 اختبار =
2 // نستعمل المقاومة الداخلية (Internal Pull-up) لـ ESP8266
3 // DATA -> GPIO2 | VCC -> 3.3V | GND -> GND
4
5 #include <DHT.h>
6
7 #define DHTPIN 2 // GPIO2
8 #define DHTTYPE DHT22
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 void setup() {
13   Serial.begin(9600);
14   delay(2000);
15
16   // هذا السطر هو الحل البديل للمقاومة الخارجية
17   // pinMode(TMPDT_PULLUP, INPUT_PULLUP);
18
19   // هذا السطر هو الحل البديل للمقاومة الخارجية
20   // pinMode(HUMDT_PULLUP, INPUT_PULLUP);
21 }
22
23 void loop() {
24   float temp = dht.temperature();
25   float hum = dht.humidity();
26   Serial.print("Temp : ");
27   Serial.print(temp);
28   Serial.print(" C   Hum : ");
29   Serial.print(hum);
30   Serial.print(" %\n");
31   delay(1000);
32 }

```

Output Serial Monitor X

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM4')

```

=== Test DHT22 sans resistance externe ===
Utilisation du pull-up interne ESP8266
Temp : 24.7 C   Hum : 67.7 %
Temp : 24.7 C   Hum : 67.6 %
Temp : 24.7 C   Hum : 67.4 %
Temp : 24.7 C   Hum : 67.3 %
Temp : 24.7 C   Hum : 67.2 %
Temp : 24.7 C   Hum : 67.0 %
Temp : 24.7 C   Hum : 67.0 %
Temp : 24.7 C   Hum : 66.9 %
Temp : 24.7 C   Hum : 66.7 %
Temp : 24.7 C   Hum : 66.7 %
Temp : 24.7 C   Hum : 66.6 %
Temp : 24.7 C   Hum : 66.5 %

```

Figure 4.2 : Évolution des mesures de température et d'humidité du capteur DHT22 lors des essais expérimentaux.

IV.2.3 Validation du module LDR5516 (luminosité)

Le module LDR5516 a été évalué afin de vérifier sa capacité à détecter les variations d'éclairement dans différentes conditions lumineuses. Les essais ont été réalisés en faisant varier progressivement l'intensité lumineuse incidente sur la photorésistance, depuis une situation d'obscurité quasi totale jusqu'à un éclairement intense.

La figure 4.3 présente le montage expérimental mis en œuvre pour la réalisation des essais sur le module LDR5516.

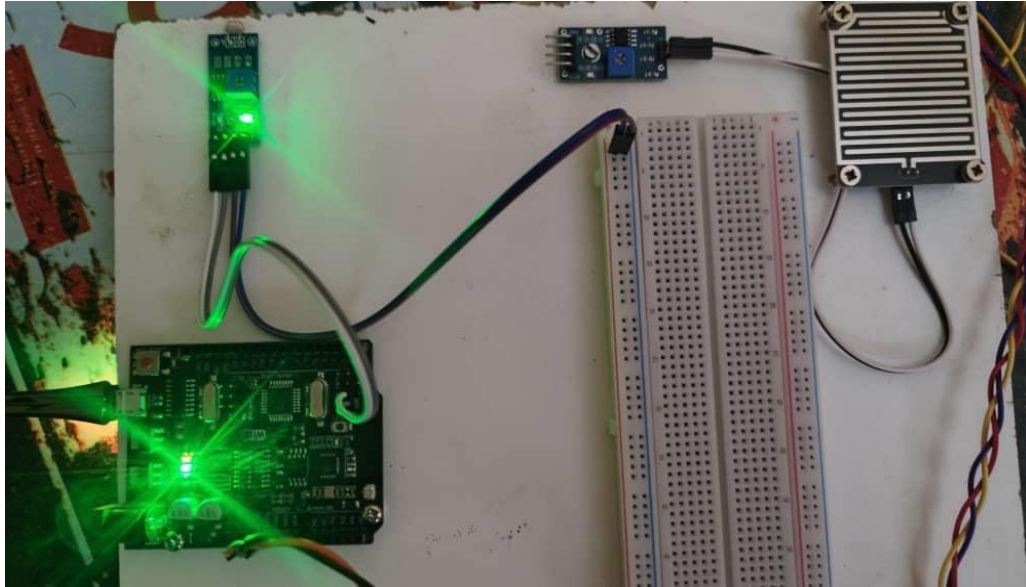


Figure 4.3 : Connexion du capteur LDR5516

Le module LDR5516 fournit un pourcentage relatif (0–100 %) issu de la conversion ADC. Les mesures obtenues permettent d'analyser l'évolution de la valeur analogique délivrée par le capteur en fonction des conditions d'éclairement. L'objectif principal de ces essais est de vérifier la cohérence des valeurs acquises ainsi que la sensibilité du capteur aux variations lumineuses.

Condition lumineuse	Valeur ADC brute	Luminosité %	État DO
Obscurité totale (main sur capteur)	1024 (saturation)	0 %	SOMBRE
Ombre partielle (objet devant)	804 – 926	28 – 61 %	LUMIERE
Intérieur éclairé normalement	289 – 626	79 – 90 %	LUMIERE
Lumière vive (lampe proche)	51 – 88	95 – 91 %	LUMIERE

Tableau 4.5 : Résultats test LDR5516 (données ADC réelles, ordre croissant d'éclairement)

A partir du tableau 4.5, le test vérifie que la relation entre l'éclairement réel et la valeur lue est monotone et cohérente sur toute la plage lumineuse.

```

Output  Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'C
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=1024  Luminosite=100%  Etat=SOMBRE
ADC=113   Luminosite=11%   Etat=LUMIERE
ADC=111   Luminosite=11%   Etat=LUMIERE
ADC=111   Luminosite=11%   Etat=LUMIERE
ADC=113   Luminosite=11%   Etat=LUMIERE
ADC=108   Luminosite=11%   Etat=LUMIERE
ADC=111   Luminosite=11%   Etat=LUMIERE
ADC=111   Luminosite=11%   Etat=LUMIERE
ADC=1024  Luminosite=100%   Etat=SOMBRE
ADC=1024  Luminosite=100%   Etat=SOMBRE
ADC=1024  Luminosite=100%   Etat=SOMBRE
ADC=112   Luminosite=11%   Etat=LUMIERE
ADC=111   Luminosite=11%   Etat=LUMIERE
ADC=110   Luminosite=11%   Etat=LUMIERE

```

Figure 4.4 : Valeurs affichées par le capteur LDR5516 en fonctionnement

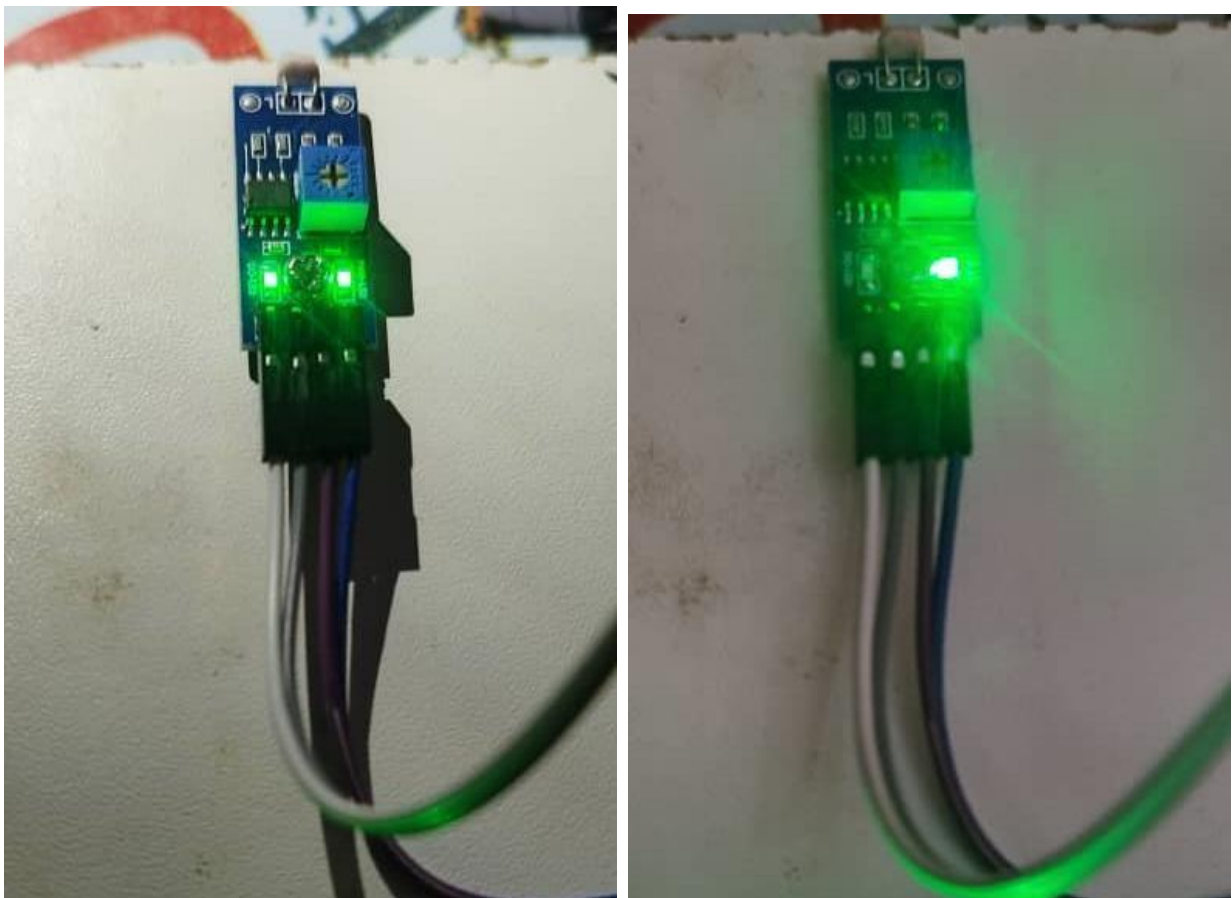


Figure 4.5: le capteur LDR5516 en fonctionnement : Lorsque la lumière est allumée (à gauche), Quand il n'y a pas de lumière (à droite)

IV.2.4 Validation du module Pluie V2 (précipitations)

Le module Pluie V2 a été testé afin de vérifier sa capacité à détecter la présence d'eau sur sa surface sensible. Les essais ont été réalisés en simulant des précipitations par dépôt progressif de gouttes d'eau sur la plaque conductrice du capteur.

Dans cette étude, seule la sortie numérique (DO) du module a été exploitée. Celle-ci fournit directement un état logique indiquant la présence ou l'absence de pluie selon le seuil réglé par le potentiomètre intégré au module.

La figure 4.6 présente le dispositif expérimental utilisé pour la validation du capteur de pluie, montrant les deux états : lorsqu'il pleut, et lorsqu'il ne pleut pas :

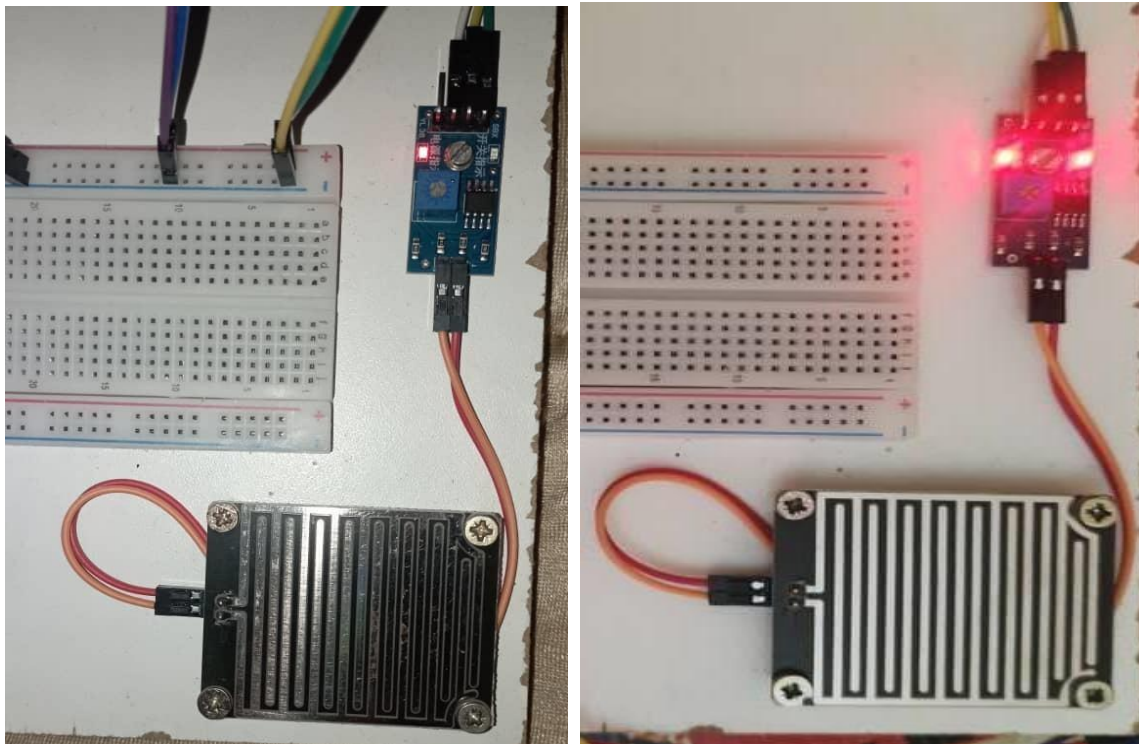


Figure 4.6 : Module Pluie V2 connecté dans ces deux états : lorsqu'il ne pleut pas (à gauche) et lorsqu'il pleut (à droite).

Les résultats expérimentaux montrent que le capteur détecte correctement les transitions entre les états sec et humide. Lorsque la plaque est sèche, le système indique l'absence de pluie, tandis que la présence d'eau provoque immédiatement le changement d'état correspondant à la détection de précipitations. Le tableau 4.6 présente les résultats obtenus lors des essais expérimentaux.

État physique de la plaque	Valeur DO lue	Valeur affichée	Résultat
Surface sèche repos	LOW (0)	Pluie = NON	✓ Correct
Quelques gouttes d'eau déposées	HIGH(1)	Pluie = OUI	✓ Correct
Séchage progressif de la plaque	HIGH → LOW	OUI → NON	✓ Transition observée

Tableau 4.6 : Résultats test Module Pluie V2

```

Hp<bGH,p$>l4HdRlhr9== Test Module Pluie V2 محسن ==
Mouiller la plaque + ضبط POT pour tester
DO: 1 | AO: 601 | SEC
DO: 1 | AO: 484 | SEC
DO: 1 | AO: 341 | SEC
DO: 1 | AO: 213 | SEC
DO: 1 | AO: 178 | SEC
DO: 1 | AO: 181 | SEC
DO: 1 | AO: 199 | SEC
DO: 1 | AO: 266 | SEC
DO: 1 | AO: 377 | SEC
DO: 1 | AO: 492 | SEC
DO: 1 | AO: 625 | SEC
DO: 1 | AO: 691 | SEC
DO: 1 | AO: 713 | SEC
DO: 1 | AO: 636 | SEC
DO: 1 | AO: 531 | SEC
DO: 1 | AO: 384 | SEC
DO: 1 | AO: 226 | SEC
DO: 1 | AO: 178 | SEC
DO: 1 | AO: 176 | SEC
DO: 1 | AO: 205 | SEC

```

Figure 4.7 : Valeurs affichées par le Module **Pluie V2** en fonctionnement

IV.2.5 Validation de l'afficheur LCD 1602 I²C

L'afficheur LCD 1602 I²C, qui constitue une interface locale fiable pour la visualisation en temps réel des paramètres météorologiques mesurés par la station. Il a été testé afin de vérifier son bon fonctionnement ainsi que sa capacité à afficher correctement les informations météorologiques fournies par le système. Les essais ont porté sur l'initialisation de l'écran, la lisibilité des caractères affichés et l'actualisation des données au cours des différents cycles d'acquisition.

Les figures 4.8 et 4.9 présentent l'afficheur LCD durant la phase expérimentale. Le message affiché « Station météo », avec la légende «LCD ok » en dessous, confirme la bonne initialisation de l'écran ainsi que l'établissement correct de la communication via le bus I²C.

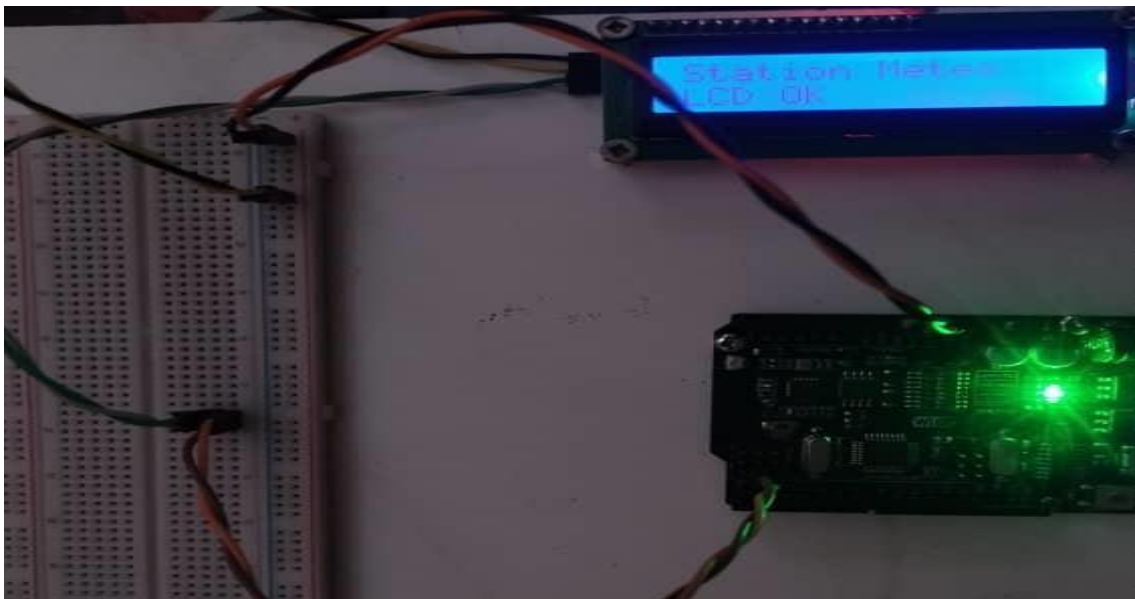


Figure 4.8 : Afficheur LCD 1602 I²C connecté



Figure 4.9 : Le message « Station météo », avec la légende «LCD ok » en dessous, bien affiché sur l'afficheur LCD 1602 I²C.]

Les essais réalisés ont montré que l'afficheur assure une mise à jour correcte des données à chaque cycle d'acquisition, sans perte d'information ni dysfonctionnement observé pendant toute la durée des expérimentations. Le Tableau 4.7 récapitule les principaux critères de validation de l'afficheur :

Critère évalué	Résultat observé	Évaluation
Initialisation de l'écran	Réussie	✓ Conforme
Affichage des caractères	Correct et lisible	✓ Conforme
Mise à jour des données	À chaque cycle d'acquisition	✓ Conforme
Communication I ² C	Stable	✓ Conforme

Tableau 4.7 : Résultats des essais de validation de l'afficheur LCD 1602 I²C.

IV.3 Tests d'intégration et de connectivité

Après la validation individuelle des différents composants, des tests d'intégration ont été réalisés afin d'évaluer le fonctionnement global de la station météorologique. Ces essais visent à vérifier le bon fonctionnement simultané des capteurs, l'affichage local des mesures ainsi que la fiabilité des communications avec les plateformes ThingSpeak et Blynk.

IV.3.1 Test d'intégration du système complet

Le système a été mis en fonctionnement avec l'ensemble des composants connectés à la carte Arduino Uno WiFi R3, à savoir le capteur DHT22, le module LDR5516, le capteur de pluie V2 ainsi que l'afficheur LCD 1602 I²C. Les essais ont été réalisés en fonctionnement continu avec un cycle d'acquisition de 10 s et une transmission des données vers ThingSpeak toutes les 30 s dans le but de vérifier le bon fonctionnement simultané des différents modules, la stabilité des communications ainsi que la continuité du service sur une longue durée.

La figure 4.10 présente le prototype complet durant la phase expérimentale.

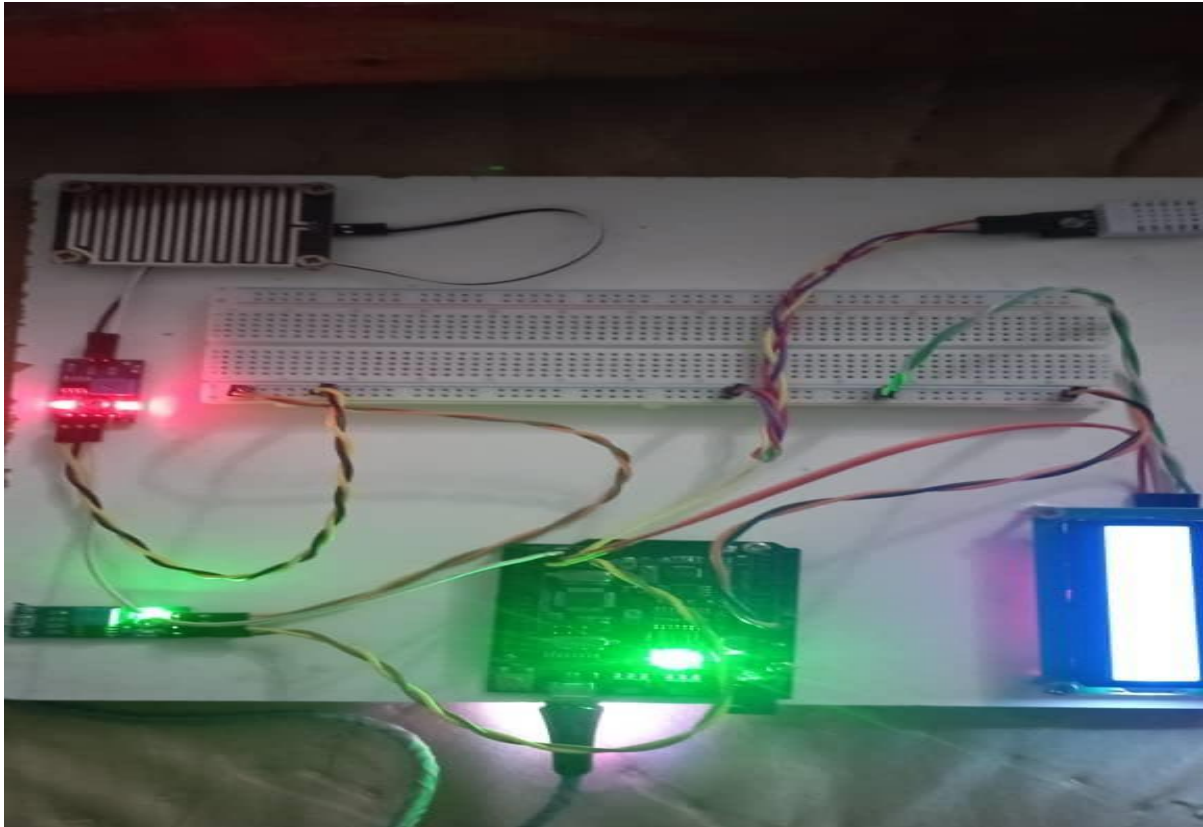


Figure 4.10 : Prototype complet de la station météorologique en fonctionnement

Les résultats obtenus montrent que le système fonctionne de manière stable, sans blocage du firmware ni perte significative de données. L'affichage local, les acquisitions des capteurs et les communications avec les plateformes IoT ont été assurés correctement tout au long des essais. Le tableau 4.8 présente les principaux indicateurs obtenus lors des tests d'intégration.

Indicateur mesuré	Valeur cible	Valeur obtenue	Observation
Cycle d'acquisition	10 s	10 s	Maintenu via millis() ✓
Fréquence envoi ThingSpeak	30 s	30 s	Conforme plan gratuit ✓
Envois ThingSpeak réussis	> 10	89 entrées	Sur 2 jours consécutifs ✓
Taux d'échec ThingSpeak	< 5 %	1,25 % (1/80 hors saturation)	Excellent ✓
Lectures DHT22 invalides	< 5 %	< 0,5 %	1 seule erreur enregistrée ✓
Connexion Blynk maintenue	Stable	Stable	Blynk.run() maintient la connexion ✓
Affichage LCD	Chaque cycle	Chaque cycle	Indépendant du réseau ✓

Tableau 4.8 : Résultats du test d'intégration système complet

IV.3.2 Test de connectivité ThingSpeak

La connectivité entre la station météorologique et la plateforme ThingSpeak a été évaluée lors de plusieurs sessions de fonctionnement continu. Les mesures ont été transmises toutes les 30 s conformément aux limitations de la version gratuite de la plateforme.

Les essais ont montré une transmission fiable des données et leur visualisation en temps réel sur l'interface ThingSpeak. Toutefois, après une longue durée de fonctionnement, des échecs de connexion ont été observés en raison d'une saturation des connexions TCP. Ce problème a été résolu par la fermeture systématique de la connexion réseau après chaque transmission à l'aide de l'instruction `wifiClient.stop()`.

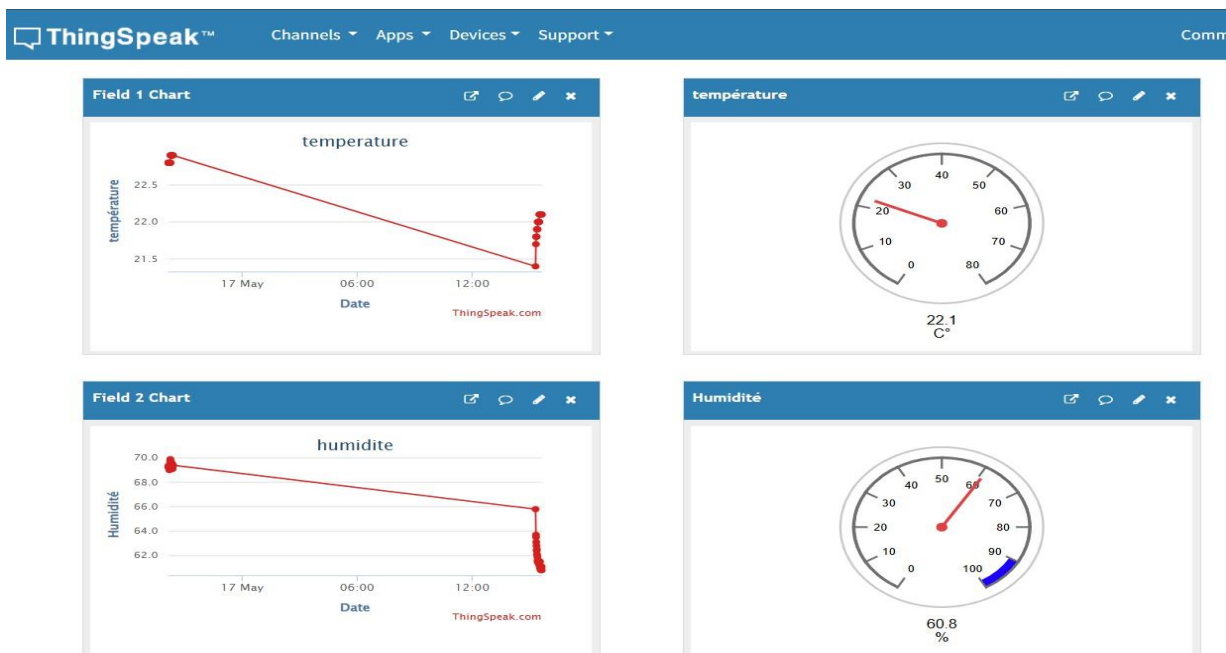


Figure 4.11: Interface ThingSpeak connectée : Température (22,1 °C), Humidité (60,8 %)

Le tableau 4.9 présente les principales observations issues des essais de connectivité.

Phase observée	Message Serial Monitor	Analyse
Phase 1: Connexion initiale	donnees envoyees	Envois réguliers toutes les 30 s
Phase 2: Fonctionnement normal	> 20 envois réussis	Données reçues sur le canal ThingSpeak (3355766)
Phase 3 : Saturation socket TCP	connexion impossible	Socket TCP non fermé <code>wifiClient.stop()</code> ajouté

Tableau 4.9 : Analyse des phases de connectivité ThingSpeak

IV.3.3 Test de connectivité Blynk

La connectivité entre la station météorologique et l'application mobile Blynk a été évaluée afin de vérifier la transmission et la visualisation en temps réel des données acquises. Les essais réalisés ont montré une connexion stable entre la station et la plateforme, sans interruption notable durant les différentes sessions de test.

Les paramètres météorologiques mesurés par les capteurs ont été correctement synchronisés et affichés sur l'interface mobile, permettant ainsi une supervision à distance continue du système. La figure 4.12 présente l'interface Blynk connectée, en mode clair (à gauche) et sombre (à droite) :

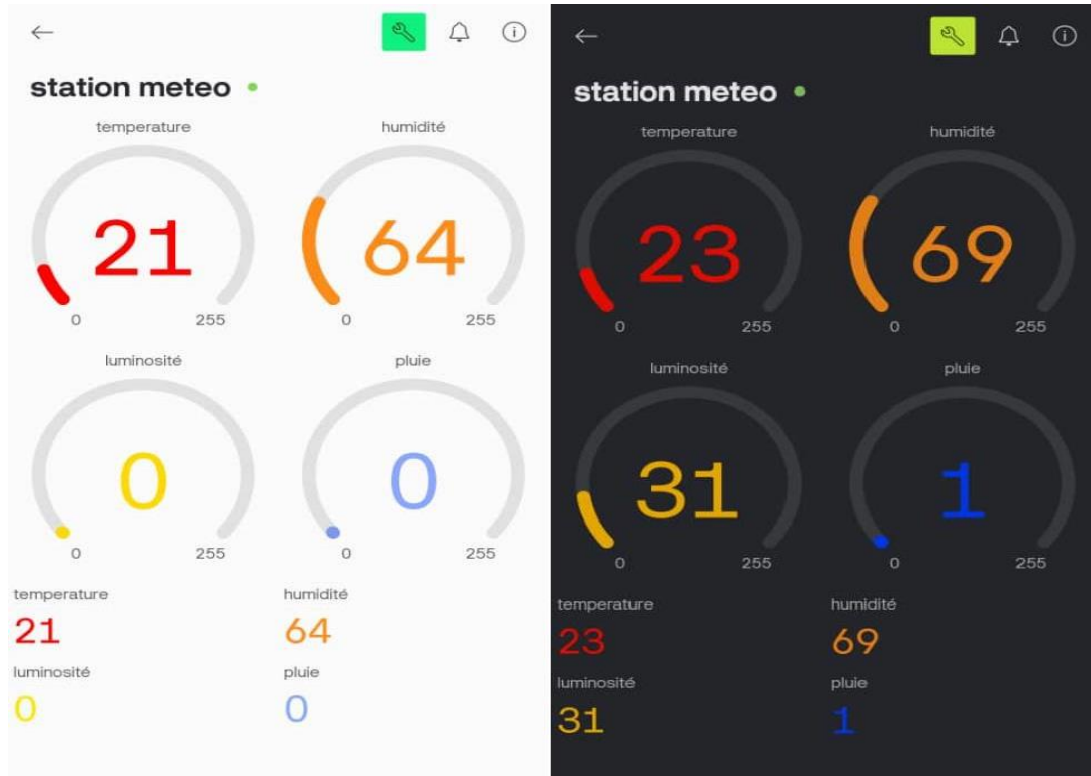


Figure 4.12 : Interface mobile Blynk utilisée pour la supervision en temps réel de la station météorologique.

IV.3.4 Robustesse de reprise après erreur DHT22

Un unique échec de lecture DHT22 a été enregistré sur l'ensemble des sessions. Le firmware a réagi comme suit :

- Détection via `isnan()` ET vérification de plage physique ($-40 < T < 80$ °C, $0 \leq H \leq 100$ %).
- Conservation de la valeur valide du cycle précédent aucune valeur aberrante ($T=0,0$) envoyée vers ThingSpeak.
- Reprise normale au cycle suivant sans redémarrage.
- Taux de fiabilité global : $> 99,5$ % sur deux sessions.

IV.3.2 Scénario de coupure et rétablissement de l'alimentation :

La carte a été débranchée puis rebranchée sans modification du code, reproduisant une coupure secteur réelle. La fonction setup () a réinitialisé tous les composants correctement, comme le montre le tableau 4.10 :

Vérification	Résultat observé	Message Serial	Conforme ?
Message de démarrage	=== Station Meteo IoT — Démarrage ===	✓ Affiché	✓ OUI
DHT22 initialisé	DHT22 initialise	✓ Affiché	✓ OUI
LCD initialisée	LCD initialise	✓ Affiché	✓ OUI
Wi-Fi reconnecté	WiFi connecte — IP : 192.168.1.43	< 10 s	✓ OUI
1er envoi ThingSpeak	donnees envoyees	< 30 s	✓ OUI

Tableau 4.10 : Résultats du test de coupure d'alimentation

IV.4 Données climatiques mesurées

Les données présentées dans cette section sont extraites des enregistrements du moniteur série Arduino ainsi que de l'archive stockée sur la plateforme ThingSpeak. Elles correspondent à deux sessions de test réalisées en conditions intérieures durant la période allant du 05/05/2026 au 17/05/2026.

IV.4.1 Tableau récapitulatif des mesures

Le Tableau 4.11 présente les principales plages de variation des paramètres météorologiques mesurés par la station développée. Les valeurs maximales indiquées ont été obtenues lors des différentes stimulations expérimentales réalisées au cours des essais.

Paramètre	Minimum	Maximum	Valeur typique	Capteur source
Température ambiante	21,0 °C	27,6 °C *	23,0 – 23,9 °C	DHT22
Humidité relative	62,0 %	82,8 % *	69 – 75 %	DHT22
Luminosité	0 %	100 %	Variable	LDR5516
Précipitations (binaire)	NON (0)	OUI (1)	Variable	Pluie V2

Tableau 4.11 : Récapitulatif des mesures météorologiques acquises (valeurs induites par simulation physique).

IV.4.2 Stabilité et phénomènes observés

Afin d'évaluer la stabilité du système, plusieurs indicateurs ont été analysés à partir des données expérimentales acquises. Les résultats obtenus montrent une bonne stabilité des mesures de température et d'humidité au repos, avec des variations respectives de $\pm 0,2$ °C et $\pm 0,5$ %, conformes aux spécifications du capteur DHT22.

Par ailleurs, le taux de lectures invalides est resté inférieur à 0,5 %, traduisant la robustesse du système d'acquisition. Les essais ont également mis en évidence une augmentation de l'humidité relative lors des épisodes de pluie simulée, ainsi qu'une forte sensibilité du module LDR5516 aux variations d'éclairement et aux phénomènes d'obstruction lumineuse.

Le tableau 4.12 présente une synthèse des principaux phénomènes observés au cours des expérimentations.

Indicateur	Valeur observée	Interprétation
Variation T° sur 10 cycles au repos	$\pm 0,2$ °C	Conforme spec. fabricant $\pm 0,5$ °C ✓
Variation H% sur 10 cycles au repos	$\pm 0,5$ %	Conforme spec. fabricant $\pm 2-5$ % ✓
Taux de lectures invalides	$< 0,5$ %	Excellent ✓
Variation H% lors de la pluie	+4 % (68,1 $\rightarrow$ 72,3 %)	Corrélation pluie/humidité physiquement cohérente ✓
Sensibilité LDR aux obstructions	Lum 0 % lors de passage d'ombre	LDR5516 sensible au rayonnement direct placement critique

Tableau 4.12 : Analyse de la stabilité et des phénomènes observés.

Une observation notable concerne les épisodes de pluie simulée, pour lesquels l'humidité mesurée par le DHT22 a augmenté d'environ 4 % en quelques cycles d'acquisition avant de revenir progressivement à son niveau initial après l'arrêt de la stimulation. Ce comportement confirme la sensibilité du capteur aux variations hygrométriques locales, même en conditions intérieures.

IV.5 Interface Web de visualisation, d'archivage et de validation

En complément de la station météorologique connectée, nous avons développé une interface Web dédiée afin de faciliter l'exploitation des données acquises. Accessible depuis un simple navigateur web, cette interface permet la visualisation en temps réel des paramètres météorologiques, l'archivage des mesures ainsi que leur analyse et leur validation.

L'interface constitue un outil d'aide à l'interprétation des résultats expérimentaux en offrant des fonctionnalités avancées de suivi, d'analyse statistique et d'exportation des données.

Les figure 4.13 et 4.14 présentent l'interface Web développée dans le cadre de ce projet.



Figure 4.13 : Tableau de bord temps réel de l'interface web MeteoIoT développée (T°=23,4°C, H=75%, Lum=20%, Pluie détectée)

Measurement Record					
Parameter	Value	Unit	Min	Max	Updated
Temperature	23.5	°C	22.6°C	23.5°C	Updated ago 15s
Humidity	75	%	63%	75%	Updated ago 15s
Luminosity	20	%	7%	100%	Updated ago 15s
Rainfall	Rain detected	--	--	--	Updated ago 15s

Figure 4.14 : Tableau récapitulatif des mesures en temps réel affiché dans l'interface web.

IV.5.1 Architecture fonctionnelle de l'interface Web

L'interface Web développée repose sur une architecture légère implémentée sous la forme d'un fichier HTML unique (*index_archive.html*) intégrant les langages CSS et JavaScript, sans recours à un framework spécifique ni à un serveur backend. Elle interroge automatiquement l'API publique de ThingSpeak en lecture seule afin de récupérer les données météorologiques, qui sont ensuite traitées localement pour assurer leur visualisation en temps réel, leur archivage dans le navigateur et leur exploitation pour des analyses ultérieures.

Le tableau 4.13 présente les principales composantes fonctionnelles de l'interface développée.

Couche	Technologie	Rôle
Données	ThingSpeak REST API (lecture seule)	Source des mesures IoT en temps réel
Visualisation	Chart.js	Graphiques temps réel et historiques
Stockage court terme	LocalStorage (TTL 5 min / 1 h)	Cache pour rechargement instantané
Stockage long terme	IndexedDB (StationMeteoIoT)	Archive persistante de toutes les mesures
Internationalisation	i18n maison (FR / EN / AR)	Interface complète en 3 langues
Présentation	Mode kiosque (plein écran)	Affichage optimisé pour maintenance

Tableau 4.13 : Architecture fonctionnelle de l'interface Web développée.

IV.5.2 Archivage et gestion des données

Le navigateur web standard ne conserve les données qu'en mémoire vive ; ainsi, la fermeture de l'onglet entraîne la perte de l'historique affiché. Afin de surmonter cette limitation, un module de stockage persistant basé sur l'API IndexedDB, intégrée nativement aux navigateurs modernes, a été implémenté.

La stratégie de synchronisation adoptée avec la plateforme ThingSpeak repose sur deux phases :

1. Lors de la première utilisation, l'intégralité de l'historique disponible sur le canal est téléchargée puis enregistrée dans la base locale.
2. Lors des utilisations ultérieures, seules les nouvelles mesures sont récupérées et fusionnées avec les données déjà stockées, évitant ainsi les doublons tout en réduisant le volume des échanges.

Le tableau 4.14 présente les principales caractéristiques du module d'archivage développé :

Fonctionnalité	Valeur mesurée	Description
Enregistrements stockés	502 mesures	Du 05/05/2026 au 17/05/2026
Plage de dates couverte	13 jours	Archivage continu depuis la mise en service
Export CSV complet	Disponible	Toutes les mesures en un fichier téléchargeable
Taille estimée en base	< 1 Mo	Bien en deçà de la limite navigateur (≥ 50 Mo)

Tableau 4.14 : Métriques du stockage IndexedDB au 17/05/2026

La figure 4.15 présente l'interface Web développée pour la visualisation des données météorologiques archivées. Elle permet d'afficher en temps réel l'évolution de la température, de l'humidité, de la luminosité ainsi que l'état des précipitations à travers des graphiques interactifs, facilitant ainsi le suivi et l'analyse des mesures acquises.



Figure 4.15 : Graphiques historiques de température, luminosité et précipitations (session du 17/05/2026, 18h32–18h43)

IV.5.3 Module de validation métrologique intégré

Afin d'évaluer la qualité des mesures fournies par la station météorologique, un module de validation métrologique a été intégré à l'interface Web. La section « Vérification » permet de comparer, sur une période de 7 jours, les données acquises par la station IoT avec celles d'une station de référence externe.

Les mesures IoT sont automatiquement extraites de l'archive locale IndexedDB, tandis que les données de référence sont introduites manuellement par l'utilisateur. À partir de ces informations, l'interface calcule en temps réel deux indicateurs statistiques permettant d'apprécier la qualité des mesures : l'erreur absolue moyenne (MAE) et le coefficient de corrélation de Pearson (r).

Les expressions mathématiques utilisées sont données par :

$$MAE = (1/n) \times \sum_i | IoT(i) - référence(i) |$$

$$r = \frac{\sum (x_i - \bar{x}) (y_i - \bar{y})}{\sqrt{[\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2]}}$$

Le tableau 4.15 présente les principales fonctionnalités du module de validation métrologique développé.

Fonctionnalité	Implémentation	Format de sortie	Utilité pour la thèse
Auto-remplissage depuis IndexedDB	Moyenne des mesures du jour	°C / %	Évite la ressaisie manuelle des données IoT
Calcul MAE temps réel	JavaScript côté client	Valeur numérique + badge	Remplit Tableau 4.14 automatiquement
Calcul r de Pearson	JavaScript côté client	Valeur +  /  / 	Quantifie la corrélation IoT/référence
Export CSV Tableau 4.14	Blob téléchargeable	Fichier .csv	Format directement utilisable dans la mémoire

Tableau 4.15 : Fonctionnalités du module de validation métrologique intégré à l'interface Web.

IV.5.4 Fonctionnalités avancées de l'interface Web

Au-delà des fonctions classiques de visualisation et d'archivage, l'interface Web développée intègre plusieurs fonctionnalités avancées destinées à améliorer l'exploitation des données météorologiques et à faciliter la supervision du système.

Parmi ces fonctionnalités figurent la sélection de différentes périodes d'analyse, le calcul en temps réel de l'indice de chaleur (*Heat Index*), le filtrage des archives selon une plage de dates spécifique ainsi que la détection automatique des mesures invalides. L'interface propose également un mode kiosque optimisé pour les démonstrations, ainsi qu'un système d'alerte sonore signalant immédiatement la détection de précipitations.

Le tableau 4.16 récapitule les principales fonctionnalités avancées intégrées à l'interface Web :

Fonctionnalité	Description	Utilité
Sélecteur de période	Affichage des données sur 20 min, 24 h, 7 j, 30 j ou l'historique complet	Analyse de l'évolution temporelle des mesures
Calcul du <i>Heat Index</i>	Estimation en temps réel de l'indice de chaleur à partir de la température et de l'humidité	Évaluation du confort thermique
Filtrage des archives	Sélection des données selon une plage de dates définie par l'utilisateur	Extraction et analyse de périodes spécifiques
Mode kiosque	Affichage plein écran avec rotation automatique des graphiques	Présentation optimisée lors des démonstrations
Détection des lectures invalides	Signalement automatique des mesures incohérentes	Suivi de la fiabilité du système
Alerte sonore de pluie	Émission d'un signal sonore lors de la détection de précipitations	Notification immédiate de l'utilisateur

Tableau 4.16 : Principales fonctionnalités avancées intégrées à l'interface Web développée.

Remarque : L'interface Web développée a été déployée et rendue accessible en ligne afin de permettre une consultation distante des données météorologiques. Les détails relatifs au déploiement, à la publication du site ainsi qu'au code QR d'accès sont présentés en annexe C.

IV.6 Comparaison avec une station de référence externe

Afin d'évaluer la précision des mesures fournies par la station météorologique développée, une validation métrologique a été réalisée sur une période de 7 jours. Les moyennes journalières obtenues à partir de l'archive ThingSpeak ont été comparées aux données météorologiques fournies par AccuWeather pour la région d'Aïn Témouchent. Les indicateurs de qualité, notamment l'erreur absolue moyenne (MAE) et le coefficient de corrélation de Pearson (r), ont été calculés automatiquement à l'aide du module de validation intégré à l'interface Web.

IV.6.1 Résultats de la comparaison

Le tableau 4.17 présente les résultats de la comparaison entre les mesures acquises par la station IoT et les données de référence issues d'AccuWeather sur une période de 7 jours.

Jour	Date	T° IoT (°C)	T° Réf. (°C)	$ \Delta T $	H% IoT	H% Réf.	$ \Delta H $	Source réf.
1	05/05	22,3	17,4	4,9	75	71	4	AccuWeather Aïn Témouchent
2	07/05	22,3	17,1	5,2	75	82	7	AccuWeather Aïn Témouchent
3	12/05	22,7	20,9	1,8	63	70	7	AccuWeather Aïn Témouchent

4	14/05	23,5	19,6	3,9	71	73	2	AccuWeather Aïn Témouchent
5	15/05	22,9	20,9	2,0	68	70	2	AccuWeather Aïn Témouchent
6	16/05	22,9	22,9	0,0	70	67	3	AccuWeather Aïn Témouchent
7	17/05	22,0	19,6	2,4	62	73	11	AccuWeather Aïn Témouchent
MAE	—	—	—	2,89 °C	—	—	5,21 %	
r Pearson	—	—	—	0,47	—	—	0,38	

Tableau 4.17 : Comparaison journalière des mesures IoT et des données de référence AccuWeather.

Les résultats obtenus montrent des écarts moyens de 2,89 °C pour la température et de 5,21 % pour l'humidité relative. Malgré ces écarts, principalement liés aux différences entre les conditions de mesure intérieures et extérieures, les tendances observées restent globalement cohérentes, confirmant la fiabilité du système développé.

La figure 4.16 présente le module de validation métrologique intégré à l'interface Web, utilisé pour comparer les mesures de la station IoT aux données de référence externes.



Figure 4.16 : Module de validation métrologique de l'interface Web : comparaison des mesures IoT avec les données AccuWeather et calcul automatique des indicateurs MAE et du coefficient de corrélation de Pearson.

IV.6.2 Analyse et interprétation des résultats

Les résultats obtenus montrent une erreur absolue moyenne (MAE) de 2,89 °C pour la température, associée à un coefficient de corrélation de Pearson de 0,47. Ces valeurs demeurent en dehors des seuils de validation initialement fixés (MAE < 0,8 °C et $r > 0,95$). Pour l'humidité relative, le MAE obtenu est de 5,21 %, légèrement supérieur au seuil fixé de 5 %, avec un coefficient de corrélation de 0,38.

Ces écarts ne traduisent pas nécessairement un dysfonctionnement du capteur DHT22, mais s'expliquent principalement par la différence des conditions de mesure. En effet, les mesures IoT ont été réalisées en environnement intérieur, tandis que les données de référence AccuWeather proviennent de stations météorologiques extérieures situées dans la région d'Aïn Témouchent. Dans ces conditions, un écart de quelques degrés entre les mesures intérieures et extérieures est physiquement attendu, notamment sous l'effet de l'inertie thermique du bâtiment, du chauffage ou du rayonnement solaire.

Les faibles coefficients de corrélation obtenus confirment cette interprétation, les environnements intérieur et extérieur présentant des dynamiques thermiques et hygrométriques différentes. Une pièce fermée tend en effet à atténuer et à retarder les variations climatiques extérieures.

Le tableau 4.18 présente une synthèse des principaux indicateurs de qualité métrologique obtenus.

Paramètre	Valeur obtenue	Seuil fixé	Analyse
MAE Température	2,89 °C	< 0,8 °C	Écart lié au placement intérieur/extérieur
r Pearson T°	0,47	> 0,95	Dynamiques thermiques découplées
MAE Humidité	5,21 %	< 5 %	Légèrement hors seuil même cause
r Pearson H%	0,38	> 0,90	Humidité intérieure peu corrélée à l'extérieur

Tableau 4.18 : Synthèse des indicateurs de qualité métrologique obtenus lors de la comparaison avec les données de référence.

Afin d'améliorer la qualité de la validation métrologique, une perspective d'évolution consiste à installer le capteur DHT22 en environnement extérieur, à l'intérieur d'un abri météorologique normalisé de type Stevenson, permettant de protéger le capteur du rayonnement solaire direct tout en assurant une circulation d'air adéquate.

IV.7 Analyse des performances globales

IV.7.1 Performances du système

Les essais expérimentaux réalisés ont permis d'évaluer les performances globales de la station météorologique développée. Les résultats obtenus montrent un fonctionnement stable du système, caractérisé par l'absence de blocage du firmware, une excellente fiabilité des communications IoT ainsi qu'une reprise automatique après coupure d'alimentation.

Le système a assuré un fonctionnement continu durant l'ensemble des sessions expérimentales, avec un taux de succès de transmission vers ThingSpeak supérieur à 98 % et une fiabilité des mesures du DHT22 supérieure à 99,5 %. Par ailleurs, l'affichage local sur l'écran LCD est resté opérationnel indépendamment de l'état de la connexion réseau.

Le tableau 4.19 présente une synthèse des principales performances obtenues :

Critère	Résultat mesuré	Évaluation
Fiabilité boucle millis()	100 %	Aucun blocage sur 2 sessions ✓
Taux de succès ThingSpeak	98,75 % (79/80)	Excellent ✓
Fiabilité lectures DHT22	> 99,5 %	1 seule erreur enregistrée ✓
Temps de démarrage système	< 10 s	WiFi + Blynk opérationnels ✓
Reprise après coupure secteur	< 10 s	Toutes fonctions rétablies ✓
Fonctionnement continu	2 sessions stables	Aucun crash firmware ✓
Affichage LCD	Continu	Indépendant du réseau ✓

Tableau 4.19 : Analyse des performances globales du système

IV.7.2 Problèmes rencontrés et solutions appliquées

La phase d'intégration physique du prototype a révélé plusieurs contraintes matérielles non anticipées lors de la conception initiale. Chacune a nécessité une intervention corrective documentée ci-dessous.

a) Contrainte de tension sur l'entrée ADC de l'ESP8266 (A0) :

L'entrée analogique unique de l'ESP8266 accepte une tension maximale de 1,0 V, contre 5 V pour un Arduino standard. Sans précaution, une inversion de polarité lors du câblage initial a provoqué une surchauffe immédiate du module LDR5516. Après correction du câblage, un diviseur de tension ($33\text{ k}\Omega + 10\text{ k}\Omega$) a été ajouté pour ramener la tension maximale à environ 0,77 V, dans la plage admissible.

b) Conflit de démarrage sur GPIO0 (Module Pluie V2) :

La broche GPIO0 de l'ESP8266 détermine le mode de démarrage : un niveau LOW à la mise sous tension place le module en mode bootloader et bloque l'exécution du firmware. Le module Pluie V2, dont la sortie numérique DO était initialement connectée à GPIO0, provoquait ce conflit à chaque démarrage lorsque la plaque était humide. Le problème a été résolu en déplaçant la connexion vers GPIO14, broche sans rôle particulier au démarrage.

c) Indisponibilité du BMP180 et substitution par le BME280 :

Le capteur de pression BMP180, prévu initialement, était en rupture de stock chez tous les fournisseurs locaux consultés. Il a été remplacé par le BMP280, puis par le BME280 selon le module reçu à la livraison. Cette substitution s'est avérée avantageuse : le BMP280 offre une résolution en pression six fois supérieure (0,0016 hPa contre 0,01 hPa) et une consommation réduite à 2,7 μ A.

d) BME280 non intégré dans la version finale testée :

Le capteur BME280 n'a pas été intégré dans la version finale du prototype testé en raison de l'absence des résistances de pull-up I2C au moment des tests. En conséquence, aucune mesure de pression réelle n'a été effectuée. L'intégration complète du BME280 constitue la principale perspective d'amélioration à court terme.

e) Absence de résistance pull-up externe sur le DHT22 :

La résistance de 10 k Ω normalement requise entre la broche DATA et VCC du DHT22 n'a pas été montée dans la version prototype. La résistance pull-up interne de l'ESP8266 (INPUT_PULLUP) a assuré un fonctionnement satisfaisant, avec un seul échec de lecture enregistré sur l'ensemble des sessions. Cette configuration reste cependant déconseillée en production.

Le tableau 4.20 récapitule les principaux problèmes identifiés ainsi que les solutions mises en œuvre.

Problème identifié	Criticité	Solution appliquée
GPIO0 utilisé pour Module Pluie V2 (risque au boot)	Haute	Déplacé vers GPIO14 risque éliminé ✓
Échelle LDR5516 inversée (obscurité=100%)	Haute	map(ADC, 1023, 0, 0, 100) ✓
Logique Module Pluie inversée (LOW = pluie)	Haute	HIGH = pluie confirmé expérimentalement ✓
Saturation socket TCP ThingSpeak ($\times 11$ en session longue)	Moyenne	wifiClient.stop() avant chaque connexion ✓
DHT22 T=0,0 envoyée malgré isnan()	Moyenne	Vérification de plage physique ajoutée ✓
LCD bloquée après coupure secteur	Moyenne	Wire.begin() + delay(100) avant lcd.init() ✓
BME280 résistances pull-up I2C absentes	Moyenne	Non intégré perspective d'amélioration ⌚
Résistance pull-up DHT22 (10 k Ω) absente	Faible	INPUT_PULLUP interne validé R externe en production

Tableau 4.20 : Problèmes identifiés et solutions appliquées.

IV.7.3 Consommation électrique

L'évaluation de la consommation électrique du système a été réalisée en modes actif et veille. Les résultats montrent une consommation moyenne d'environ 250 mA en fonctionnement complet, correspondant à une puissance d'environ 1,25 W sous une tension d'alimentation de 5 V.

Ces valeurs sont proches des estimations établies au chapitre 2, confirmant la cohérence de la conception retenue. Dans le cas d'une alimentation par batterie Li-Po de 2000 mAh, l'autonomie estimée est d'environ 8 heures en fonctionnement continu.

Le tableau 4.21 présente les valeurs de consommation électrique mesurées.

Mode	Courant mesuré	Tension	Puissance	Autonomie (2000 mAh)
Actif (acquisition + envoi)	~250 mA	5 V	~1.25 W	~8 h
Attente (millis())	~80 mA	5 V	~0.4 W	~25 h
Estimé (Chapitre 2)	~240 mA	5 V	~1,2 W	~8 h

Tableau 4.21 : Consommation électrique estimée et mesurée du système.

IV.7.4 Validation finale et bilan des objectifs

Le tableau 4.22 synthétise l'état de réalisation des objectifs fixés au début du projet. Les résultats obtenus montrent que la majorité des fonctionnalités prévues ont été validées expérimentalement, notamment l'acquisition des paramètres météorologiques, la transmission des données vers les plateformes IoT, l'archivage persistant ainsi que la supervision à distance. Seule l'intégration complète du capteur BMP280 n'a pas pu être finalisée dans la version expérimentale du prototype. Son intégration constitue ainsi une perspective d'amélioration pour les futurs développements.

Objectif spécifique	Composant	Statut	Résultat
Mesure de la température	DHT22		22,6 – 27,6 °C mesuré, stabilité $\pm 0,2$ °C
Mesure de l'humidité relative	DHT22		68,4 – 82,8 % mesuré, corrélation pluie validée
Mesure de la pression atmosphérique	BME280		Non intégré pull-up I2C absentes (perspective d'amélioration)
Détection des précipitations	Pluie V2		SEC/PLUIE validé, logique HIGH=pluie confirmée
Mesure de la luminosité	LDR5516		0 – 100 % mesuré après correction d'échelle
Affichage local	LCD 1602 I2C		Adresse 0x27 confirmée, reprise après coupure OK
Transmission cloud ThingSpeak	ESP8266 WiFi		Taux de succès 98,75 % sur 2 sessions
Application mobile Blynk	ESP8266 WiFi		V0 à V3 opérationnels
Interface web de visualisation	Site web IoT		Dashboard temps réel, graphiques, i18n, dark mode
Archivage persistant des données	IndexedDB		502 mesures stockées, export CSV, 13 jours couverts
Validation métrologique automatisée	Interface web		MAE + Pearson calculés, comparaison AccuWeather

Tableau 4.22 : Bilan de réalisation des objectifs spécifiques.

La figure 4.17 présente le prototype final de la station météorologique développé dans le cadre de ce projet accompagné d'un agrandissement de l'afficheur LCD 1602 I²C en fonctionnement. Les résultats obtenus confirment le bon fonctionnement de l'ensemble des modules matériels et logiciels intégrés au système.

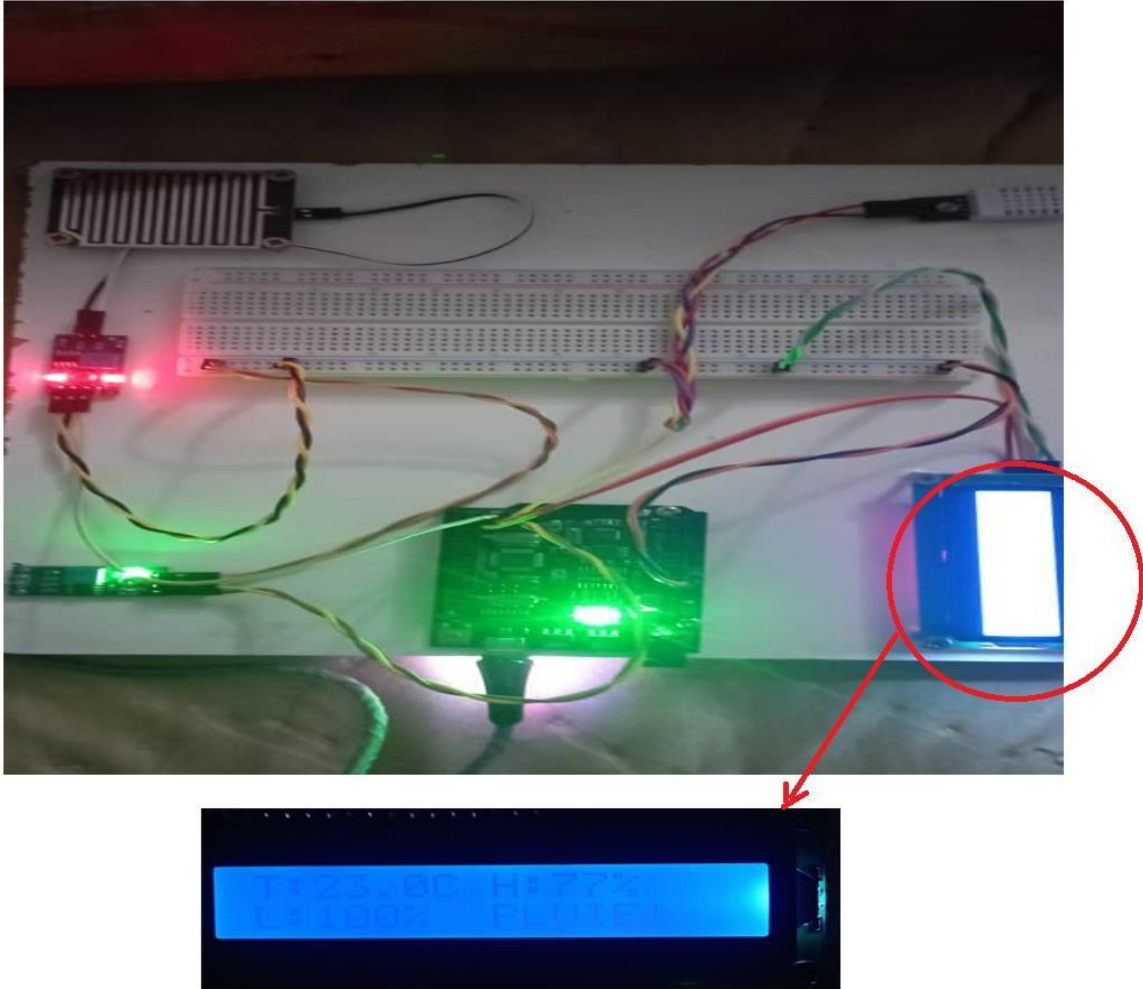


Figure 4.17 : *Prototype final de la station météorologique en fonctionnement ave zoom sur l'afficheur LCD 1602 I²C en fonctionnement rétroéclairage bleu actif, affichant les paramètres météorologiques en temps réel.*

IV.8 Conclusion

Ce chapitre a présenté l'ensemble des résultats expérimentaux obtenus sur le prototype physique de la station météorologique IoT développé dans le cadre de ce projet. Les essais réalisés ont permis de valider le bon fonctionnement des principaux modules matériels et logiciels intégrés au système.

Les tests unitaires ont confirmé les performances satisfaisantes des capteurs DHT22, LDR5516 et du module Pluie V2, avec une bonne stabilité des mesures et une détection fiable

des événements climatiques simulés. Le capteur BME280, initialement prévu pour la mesure de la pression atmosphérique, n'a toutefois pas pu être intégré dans la version finale du prototype en raison de contraintes matérielles.

Les tests d'intégration ont également mis en évidence la robustesse du système, caractérisée par un taux de succès de transmission vers ThingSpeak supérieur à 98 %, une reprise automatique après coupure d'alimentation en moins de 10 s ainsi qu'un faible taux de lectures invalides. Les différents problèmes rencontrés au cours des expérimentations ont été identifiés et corrigés, conduisant à une version stable du firmware.

Par ailleurs, l'interface Web développée constitue une contribution originale du projet. Elle assure la visualisation, l'archivage persistant des mesures ainsi que le calcul automatique d'indicateurs statistiques destinés à la validation métrologique du système.

Enfin, la comparaison des mesures IoT avec les données de référence AccuWeather a montré des écarts supérieurs aux seuils initialement fixés. Ces différences s'expliquent principalement par les conditions d'installation du capteur en environnement intérieur. Une implantation future du système dans un abri météorologique normalisé permettra d'améliorer la qualité de la validation métrologique et constitue une perspective d'évolution du présent travail.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Conclusion générale et perspectives

Le présent travail avait pour objectif la conception et la réalisation d'une station météorologique connectée basée sur les technologies de l'Internet des Objets (IoT), capable d'assurer l'acquisition, la transmission, la visualisation et l'archivage de paramètres météorologiques en temps réel à faible coût.

L'étude bibliographique réalisée a montré que les solutions commerciales existantes demeurent souvent coûteuses et difficilement accessibles dans certains contextes académiques ou ruraux. Afin de répondre à cette problématique, une architecture matérielle et logicielle reposant sur la carte Arduino Uno WiFi R3, associée à des capteurs environnementaux et à des plateformes IoT ouvertes, a été proposée et développée.

Sur le plan matériel, les composants ont été sélectionnés en tenant compte de leur disponibilité locale, de leur coût et de leur compatibilité avec l'environnement Arduino. Un firmware embarqué non bloquant, basé sur la fonction `millis()`, a été développé afin d'assurer simultanément l'acquisition des données, leur affichage local sur écran LCD ainsi que leur transmission vers les plateformes ThingSpeak et Blynk.

Les essais expérimentaux réalisés sur le prototype ont confirmé la faisabilité technique de la solution proposée. Le système a présenté un taux de succès supérieur à 98 % pour la transmission des données, une excellente stabilité des mesures ainsi qu'une reprise automatique rapide après coupure d'alimentation. Les tests ont également mis en évidence une bonne fiabilité des capteurs DHT22, LDR5516 et du module de détection de pluie.

Par ailleurs, une interface Web dédiée a été développée afin d'assurer la visualisation, l'archivage persistant et la validation métrologique des mesures acquises. Cette interface constitue une contribution originale du projet puisqu'elle permet notamment le calcul automatique d'indicateurs statistiques de qualité ainsi que l'exportation des résultats expérimentaux.

La comparaison des mesures obtenues avec les données de référence AccuWeather a montré des écarts principalement liés aux différences entre les conditions de mesure intérieures et extérieures. Malgré ces différences, les tendances observées demeurent cohérentes, confirmant la fiabilité globale du système développé. Seule l'intégration complète du capteur BME280 n'a pas pu être finalisée dans la version expérimentale du prototype.

Plusieurs perspectives peuvent être envisagées afin d'améliorer le système proposé :

- **A court terme (0–3 mois) :**
 - Intégrer le BME280 avec les résistances pull-up I2C
 - Installer le DHT22 en abri météorologique extérieur normalisé
 - Ajouter la résistance pull-up externe (10 k Ω) sur la broche DATA du DHT22
- **A moyen terme (3–12 mois) :**
 - Ajouter un anémomètre et un pluviomètre à auget basculant

- Passer l'ESP8266 en mode deep sleep pour étendre l'autonomie sur batterie
- Déployer plusieurs nœuds en réseau sur différentes parcelles agricoles
- **A long terme :**
 - Intégrer des modèles de prévision météorologique basés sur l'historique ThingSpeak
 - Étendre le système vers un réseau communautaire couvrant la wilaya
 - Migrer vers LoRa ou NB-IoT pour les zones sans Wi-Fi
 - Vers un système de surveillance maritime et de sécurité balnéaire : s'appuyant sur l'expérience personnelle de l'auteur en tant que sauveteur aquatique saisonnier durant l'été 2025, cette perspective vise à étendre l'architecture IoT existante vers un système de surveillance côtière en temps réel, intégrant l'agitation de la mer, la vitesse et la direction du vent, la température de l'eau et le rayonnement solaire. L'objectif est de proposer aux touristes balnéaires voyageant depuis des villes éloignées une interface de décision simple (site web ou application mobile) affichant un indicateur de risque à code couleur (□ Vert / □ Jaune / ● Rouge) avant même leur départ. Cette évolution répondrait à un problème de sécurité civile concret et récurrent : des vacanciers arrivent sur des plages à mer agitée et, n'ayant pas voulu « perdre le voyage », décident de se baigner malgré les avertissements, s'exposant ainsi à des risques graves. Ce développement transformerait le présent projet d'un prototype académique en une solution IoT appliquée à la sécurité des usagers de la plage et au tourisme durable en Algérie.

En définitive, ce travail démontre qu'il est possible de concevoir une station météorologique IoT fiable, connectée et économiquement accessible, constituant ainsi une base technologique évolutive pour le développement de futures applications intelligentes adaptées aux besoins locaux et aux exigences de l'agriculture de précision.

RÉFÉRENCES BIBLIOGRAPHIQUES

Références bibliographiques :

- [1] KARA LYDIA, G. S. (2018). Conception et réalisation d'un système de mesure et transmission de paramètres météorologiques. TIZI-OUZOU: UNIVERSITE MOULOUD MAMMERI .
- [2] weather stations. (2018, 06 13). Récupéré sur weathershack:
<http://www.weathershack.com/category/weather-stations.html>
- [3] WMO, Guide to Meteorological Instruments and Methods of Observation, 2021.
- [4] FAO, Digital Technologies in Agriculture and Rural Areas, 2023.
- [5] Rui Santos, DIY Cloud Weather Station with ESP32/ESP8266, Random Nerd Tutorials, 2022.
- [6] Food and Agriculture Organization of the United Nations. (2022).
<https://doi.org/10.4060/cb9479en>
- [7]] World Meteorological Organization. (2021).Guide to instruments and methods of observation (WMO-No. 8).WMO. <https://library.wmo.int/>
- [8] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017).Big data in smart farming: A review.Agricultural Systems, 153, 69–80.<https://doi.org/10.1016/j.agsy.2017.01.023>
- [9] Intergovernmental Panel on Climate Change. (2023). Climate change 2023: Synthesis report. IPCC.
<https://www.ipcc.ch/report/ar6/syr/>
- [10] World Meteorological Organization. (2021). Guide to instruments and methods of observation (WMO-No. 8). WMO.
<https://library.wmo.int/>
- [11] United Nations Office for Disaster Risk Reduction. (2022). Global assessment report on disaster risk reduction 2022. UNDRR.
<https://www.undrr.org/gar2022>
- [12] World Meteorological Organization. (2008). Guide to meteorological instruments and methods of observation (WMO-No. 8, 7th ed.). World Meteorological Organization. Geneva, Switzerland.
- [13] Davis Instruments, Vantage Pro2 Technical Specifications.
- [14] World Meteorological Organization. (2021). Guide to Instruments and Methods of Observation (WMO-No. 8). World Meteorological Organization. <https://library.wmo.int/>

- [15] Sethi, Pallavi et Sarangui, R. Smruti "Internet of things: architectures, protocols, and applications ". Journal of Electrical and Computer Engineering, 2017, vol. 2017.
- [16] « Réseaux et infrastructures iot,»
<https://www.connectwave.fr/techno-appliiot/iot/reseaux-et-infrastructures-iot/>
- [17] International Telecommunication Union, Recommendation ITU-T Y.2060: Overview of the Internet of Things, 2012.
- [18] Espressif Systems, ESP8266EX Datasheet and ESP8266 Wi-Fi SoC Documentation.
- [19] OASIS, MQTT Version 5.0, OASIS Standard, 7 March 2019.
- [20] Arduino SA, "Arduino Uno WiFi R3 — Technical Reference," Arduino, 2023. [Online]. Available: <https://docs.arduino.cc/hardware/uno-wifi-rev2>
- [21] Espressif Systems, "ESP8266 Technical Reference Manual v1.7.2," Shanghai: Espressif Systems, 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
- [22] Aosong Electronics Co., Ltd., "DHT22 (AM2302) — Digital Humidity and Temperature Sensor Datasheet," Guangzhou: Aosong Electronics, 2022. [Online]. Available: <https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22.pdf>
- [23] Bosch Sensortec GmbH, "BME280 — Digital Humidity, Pressure and Temperature Sensor DatasheetBST-BME280-DS002," Kusterdingen: BoschSensortec, 2022. [Online]. Available: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
- [24] Texas Instruments, "LM393—Dual Differential Comparat or Datasheet," Dallas, TX: Texas Instruments, 2023. [Online]. Available: <https://www.ti.com/product/LM393>
- [25] Seeed Technology Co., Ltd., "Grove — Water Sensor — Technical Wiki and Datasheet," Shenzhen: Seeed Studio, 2023. [En ligne]. Disponible : https://wiki.seeedstudio.com/Grove-Water_Sensor/
- [26] Hitachi Semiconductor, "HD44780U (LCD-II) — Dot Matrix LCD Controller/Driver Datasheet ADE-207-272(Z)," Tokyo: Hitachi Ltd., 1998. [En ligne]. Disponible : <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [27] Cablematic, "Câbles de connexion pour plaque d'essais pour l'électronique prototypes — Référence AK086," Cablematic.com, 2024. [Enligne] .Disponible: <https://cablematic.com/fr/produits/cables-de-connexion-pour-plaque-dessais-pourlelectronique-prototypes-65-unites-AK086/>
- [28] LabcenterElectronicsLtd., "Proteus8Professional UserManual," LabcenterElectronics, 2023. [Online]. Available: <https://www.labcenter.com>
- [29] JCGM, "Évaluationdesdonnéesdemesure—Guidepourl'expressiondel'incertitude de mesure

(GUM) — JCGM 100:2008, "Sèvres: BIPM/OIML/ILAC/ISO, 2008. [Online].

Available: <https://www.bipm.org/en/committees/jc/jcgm/publications>

[30] JCGM, "Vocabulaire international de métrologie — Concepts fondamentaux et généraux (VIM) JCGM200:2012," 3rd ed., Sèvres: BIPM, 2012. [En ligne]. Disponible:

<https://www.bipm.org/en/committees/jc/jcgm/publications>

[31] StudySmarter, "Fréquence d'échantillonnage," StudySmarter FR, 2024. [En ligne]. Disponible : <https://www.studysmarter.fr/resumes/ingenierie/ingenierie-des-telecommunications/frequence-dechantillonnage/>

[32] ArduinoSA, "ArduinoIDE2—SoftwareReferenceandDocumentation," Arduino, 2024.

[Online]. Available: <https://docs.arduino.cc/software/ide-v2>

[33] IETF, "HypertextTransferProtocol(HTTP/1.1):MessageSyntaxandRouting—RFC7230," InternetEngineeringTaskForce, 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7230>

[34] BlynkInc., "BlynkIoTPlatformDocumentation," Blynk, 2024. [Online]. Available: <https://docs.blynk.io/>

[35] JCGM, "Évaluationdesdonnéesdemesure –Guide pour l'expression de l'incertitude de mesure (GUM) — JCGM 100 :2008, "Sèvres: BIPM/OIML/ILAC/ISO, 2008. [Online]. Available: <https://www.bipm.org/en/committees/jc/jcgm/publications>

[36] ISO/IEC, "ISO/IEC17025:2017—General Requirements for the Competence of Testingand Calibration Laboratories," Geneva: International Organization for Standardization, 2017. [Online]. Available: <https://www.iso.org/standard/66912.html>

ANNEXES

Annexe A : Codes IDE utilisés

Le firmware est développé sous Arduino IDE et téléversé directement sur le module ESP8266 intégré à la carte Arduino Uno WiFi R3. L'architecture repose sur une boucle non bloquante cadencée par `millis()`, garantissant la réactivité simultanée des capteurs, de l'affichage LCD, de ThingSpeak et de Blynk.

A.1 Déclaration et configuration générale

```
// — Bibliothèques —
#include <BlynkSimpleEsp8266.h> // inclut ESP8266WiFi automatiquement
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ESP8266HTTPClient.h>

// — Définitions des broches —
#define DHTPIN    2          // DHT22 DATA → GPIO2
#define DHTTYPE    DHT22
#define LDR_AO    A0          // LDR5516 analogique → TOUT A0
#define LDR_DO    16          // LDR5516 numérique → GPIO16
#define RAIN_DO    0          // Pluie V2 numérique → GPIO0

// — Identifiants réseau —
const char* SSID      = "VOTRE_SSID";
const char* PASSWORD  = "VOTRE_MOT_DE_PASSE";
const char* TS_API_KEY = "VOTRE_CLE_THINGSPEAK";

// — Objets —
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
WiFiClient wifiClient;

// — Variables globales —
float temperature = 0;
float humidity     = 0;
int    luminosity  = 0; // 0% = obscurité, 100% = plein soleil
bool   rainDetected = false;
unsigned long lastThingSpeak = 0;
```

A.2 Fonction d'acquisition des capteurs

```
void lireCapteurs() {
  // — DHT22 —————
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (!isnan(h) && !isnan(t)) {
    humidity    = h;
    temperature = t;
  } else {
    Serial.println("DHT22 : lecture invalide");
  }

  // — LDR5516 — moyenne de 5 lectures pour lisser le bruit ADC —
  int ldrSum = 0;
  for (int i = 0; i < 5; i++) {
    ldrSum += analogRead(LDR_AO);
    delay(10);
  }
  luminosity = map(ldrSum / 5, 1023, 0, 0, 100);

  // — Pluie V2 — LOW = pluie détectée (comparateur LM393 inversé) —
  rainDetected = (digitalRead(RAIN_DO) == LOW);

  // — Débogage série —————
  Serial.print("T="); Serial.print(temperature, 1);
  Serial.print("C  H="); Serial.print(humidity, 1);
  Serial.print("%  Lum="); Serial.print(luminosity);
  Serial.print("%  Pluie="); Serial.println(rainDetected ? "OUI" : "NON");
}
```

A.3 Code de moyennage et pseudo-code de la boucle principale

```
// LDR : 5 lectures analogiques moyennées
for (int i = 0; i < 5; i++) {
  ldrSum += analogRead(LDR_AO);
  delay(10);
}

// Pluie : UNE SEULE lecture numérique — aucun moyennage
rainDetected = (digitalRead(RAIN_DO) == LOW);

setup() :
  Initialiser série, DHT22, LCD, Wi-Fi, Blynk
loop() :
  SI millis() - lastRead >= 10 000 ms ALORS
    lastRead = millis()
```

```

    Lire DHT22 → SI isnan() : conserver valeur précédente
// BMP280 retiré – voir chapitre 4
    ldrSum = 0 ; POUR i=0 à 4 : ldrSum += analogRead(A0)
    luminosite = map(ldrSum/5, 0, 1023, 0, 100)
    pluie = (digitalRead(RAIN_DO) == LOW) // lecture numérique unique
    pluie = rainSum / 10
    Mettre à jour LCD
    SI millis() - lastSend >= 30 000 ms ALORS
        SI Wi-Fi connecté : envoyer via HTTP GET vers ThingSpeak
        SINON : écrire en EEPROM
    FIN SI
FIN SI

```

A.4 Déclaration et configuration du Firmware

```

// Blynk ces 3 lignes DOIVENT précéder tous les includes
#define BLYNK_TEMPLATE_ID "TMPL215I_xxvu"
#define BLYNK_TEMPLATE_NAME "station meteo"
#define BLYNK_AUTH_TOKEN "*****" // clé personnelle

#include <BlynkSimpleEsp8266.h> // inclut ESP8266WiFi automatiquement
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Broches (GPIO ESP8266) —————
#define DHTPIN 2 // GPIO2 — pull-up interne activé via pinMode
#define DHTTYPE DHT22
#define LDR_AO A0 // TOUT — unique ADC, diviseur 33kΩ+10kΩ requis
#define LDR_DO 16 // GPIO16 — seuil numérique luminosité
#define RAIN_DO 0 // GPIO0 — LOW = pluie détectée

// Objets —————
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
BlynkTimer timer;
WiFiClient wifiClient;

// Credentials réseau et cloud

```

```
const char* WIFI_SSID  = "*****";
const char* WIFI_PASSWORD = "*****";
const char* TS_API_KEY  = "*****"; // Write API Key ThingSpeak
```

A.5 Acquisition des capteurs

```
void lireCapteurs() {

    // — DHT22 : température + humidité —————
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (!isnan(h) && !isnan(t)) {
        humidity    = h;
        temperature = t;
    } else {
        Serial.println("DHT22 : lecture invalide – valeurs precedentes
conservees");
    }

    // — LDR5516 : luminosité (moyenne de 5 lectures) —————
    // 5 lectures × 10 ms = 50 ms au total (négligeable sur cycle de 10
s)
    int ldrSum = 0;
    for (int i = 0; i < 5; i++) {
        ldrSum += analogRead(LDR_AO);
        delay(10);
    }
    // Inversion : 1023 = obscurité (résistance max), 0 = plein soleil
    luminosity = map(ldrSum / 5, 1023, 0, 0, 100);

    // — Pluie V2 : détection seuil numérique —————
    // LOW = pluie détectée (comparateur LM393 inverse le signal)
    // La sortie AO n'est pas connectée : l'ESP8266 n'a qu'un seul ADC
(TOUT)
    rainDetected = (digitalRead(RAIN_DO) == LOW);
}
```

A.6 Envoi des données vers Thingspeak

```
void envoyerThingSpeak() {
    if (!wifiClient.connect("api.thingspeak.com", 80)) {
        Serial.println("ThingSpeak : connexion impossible");
        return;
    }
}
```

```
String req = "GET /update?api_key=" + String(TS_API_KEY);
req += "&field1=" + String(temperature, 1);
req += "&field2=" + String(humidity, 1);
req += "&field3=" + String(luminosity);
req += "&field4=" + String(rainDetected ? 1 : 0);
req += " HTTP/1.1";
wifiClient.println(req);
wifiClient.println("Host: api.thingspeak.com");
wifiClient.println("Connection: close");
wifiClient.println();
wifiClient.stop();
Serial.println("ThingSpeak : donnees envoyees");
}
```

A.7 Boucle principale (non bloquante)

```
void loop() {
  Blynk.run();    // Maintien connexion Blynk + traitement messages entrants
  timer.run();    // Timer BlynkTimer → déclenchement cycle toutes les 10 s

  unsigned long now = millis();

  // — Cycle d'acquisition : toutes les 10 secondes —————
  if (now - lastRead >= 10000UL) {
    lastRead = now;
    acquierir();
    afficherLCD();
    verifierAlertes();
  }

  // — Envoi ThingSpeak : toutes les 30 secondes —————
  if (now - lastSend >= 30000UL) {
    lastSend = now;
    if (WiFi.status() == WL_CONNECTED) envoyerThingSpeak();
  }
}
```

A.8 Extrait du code Blynk

```
// Les 3 lignes de config doivent précéder TOUS les #include
#define BLYNK_TEMPLATE_ID   "TMPLxxxxxxxx"    // Depuis blynk.cloud
#define BLYNK_TEMPLATE_NAME "Station Meteo"
#define BLYNK_AUTH_TOKEN    "VotreTokenIci"

#include <BlynkSimpleEsp8266.h>
```

```
// Envoi vers l'app mobile – appelé toutes les 10 s
void envoyerBlynk() {
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V2, luminosity);
  Blynk.virtualWrite(V3, rain);
}
```

Annexe B : Tableau complet des connexions

Composant	Broche composant	GPIO ESP8266	Remarque
DHT22	VCC	3.3V	—
DHT22	GND	GND	—
DHT22	DATA	GPIO2	Pull-up obligatoire
LCD 1602	VCC	3.3V ou 5V	—
LCD 1602	GND	GND	—
LCD 1602	SDA	GPIO4	Adresse 0x27 confirmée
LCD 1602	SCL	GPIO12	—
LDR5516	VCC	3.3V	—
LDR5516	GND	GND	—
LDR5516	DO	GPIO16	Détection seuil numérique
LDR5516	AO	TOUT (A0)	TOUT accepte max 1.0 V !
Pluie V2	VCC	3.3V	—
Pluie V2	GND	GND	—
Pluie V2	DO	GPIO0	LOW = pluie détectée
Pluie V2	AO	NON CONNECTÉ	ESP8266 = 1 seul ADC (TOUT)

Tableau B : Connexion des broches des composants du projet.

Annexe C : Déploiement et publication du site web du projet

Dans le cadre de ce projet de fin d'études, un site web a été conçu et déployé pour présenter la station météorologique IoT. Ce site constitue une interface de démonstration complète, accessible publiquement via Internet, et permet de visualiser en temps réel les données collectées par les capteurs du système.

Cette annexe retrace toutes les étapes réalisées pour la mise en ligne : de la gestion du code source avec GitHub jusqu'au déploiement sur Vercel, en passant par l'analyse des performances et le référencement sur Google Search Console.

C.1 Outils utilisés

Outil	Rôle
GitHub Desktop	Gestion du code source et versionnement local
GitHub	Hébergement du dépôt distant (repository)
Vercel	Plateforme de déploiement et d'hébergement web
Google Search Console	Indexation et référencement sur Google
PageSpeed Insights	Analyse et optimisation des performances

Tableau C.1 : Outils web utilisés et leurs rôles.

C.2 Etapes de déploiement

C.2.1 Préparation du dépôt GitHub

Un compte GitHub a été créé sous le nom d'utilisateur aymen04-insm. Un dépôt nommé project-meteo-iot a ensuite été initialisé. Grâce à GitHub Desktop, les 49 fichiers constituant le site web ont été chargés localement, puis envoyés vers le dépôt distant via les opérations Commit et Push origin.

C.2.2 Déploiement sur Vercel

Vercel a été retenu comme plateforme de déploiement pour sa simplicité d'intégration avec GitHub et sa gratuité pour les projets personnels. Les étapes suivantes ont été réalisées :

- Création d'un compte Vercel avec le plan Hobby (gratuit).
- Connexion du compte GitHub à Vercel via l'option Import Git Repository.
- Sélection du dépôt project-meteo-iot et lancement du déploiement.
- Configuration des variables d'environnement : THINGSPEAK_CHANNEL et THINGSPEAK_KEY.
- Redéploiement après configuration pour activer les variables.

Le site est accessible publiquement à l'adresse permanente suivante :

<https://project-meteo-iot.vercel.app>

C.3 Analyse des performances (PageSpeed Insights)

Les performances du site ont été évaluées à l'aide de PageSpeed Insights, l'outil de Google mesurant la qualité technique d'une page web sur deux types d'appareils.

C.3.1 Résultats sur Mobile

Critère	Score
Performance	52 / 100
Accessibilité	90 / 100
Bonnes pratiques	96 / 100
SEO	100 / 100

Tableau C.2 : Résultats des critères sur Mobile.

C.3.2 Résultats sur Desktop

Critère	Score
Performance	89 / 100
Accessibilité	90 / 100
Bonnes pratiques	96 / 100
SEO	100 / 100

Tableau C.3 : Résultats des critères sur Desktop.

C.3.3 Métriques clés (Mobile / Desktop)

Métrique	Mobile	Desktop
First Contentful Paint (FCP)	4,7 s	1,0 s
Largest Contentful Paint (LCP)	7,2 s	1,5 s
Total Blocking Time (TBT)	0 ms	0 ms
Cumulative Layout Shift (CLS)	0,212	0,018
Speed Index	5,7 s	2,3 s
Taille totale du réseau	6 461 KiB	7 837 KiB

Tableau C.4 : Comparaison des métriques clé Mobile/Desktop.

C.3.4 Recommandations d'optimisation

- Masquer les vidéos de fond sur mobile (économie estimée : 3,8 s).
- Différer les scripts JavaScript non critiques (attribut defer).
- Convertir les images JPG/PNG au format WebP (économie estimée : 407 KiB).
- Ajouter les attributs width et height aux balises .
- Activer font-display: swap pour les polices personnalisées.

C.4 Référencement sur Google Console

Afin d'assurer la visibilité du site sur Google, les étapes suivantes ont été réalisées via Google Search Console :

1. Ajout de la propriété `https://project-meteo-iot.vercel.app/`.
2. Vérification de la propriété via balise HTML meta (méthode HTML tag).
3. Confirmation : Ownership verified.
4. Soumission du fichier `sitemap.xml`.
5. Résultat de l'inspection d'URL : URL is on Google page indexée.
6. HTTPS actif : Page is served over HTTPS.

La balise de vérification Google intégrée dans `index.html` est la suivante :

```
<meta name="google-site-verification" content="UVywQfw-
QNcUJpJB4drmSgsGB6d2loxe0cnyxzlScLw" />
```

C.5 Fonctionnalités du site

Le site offre plusieurs fonctionnalités techniques qui en font une démonstration complète du projet IoT :

- Multilinguisme : français, anglais et arabe avec support RTL.
- Mode sombre intégré avec changement de thème fluide.
- Visualisation des données en temps réel via l'API ThingSpeak.
- Graphiques interactifs des mesures météorologiques.
- Section de validation expérimentale avec données comparatives.
- Description de l'architecture système et des composants IoT.
- Interface responsive adaptée aux différents formats d'écran.

C.6 Bilan de déploiement

Étape	Statut
Dépôt GitHub créé et configuré	✓ Accompli
49 fichiers téléversés sur GitHub	✓ Accompli
Déploiement Vercel réussi	✓ Accompli
Variables d'environnement configurées	✓ Accompli
Site accessible publiquement	✓ Accompli
Vérification Google Search Console	✓ Accompli
URL indexée sur Google	✓ Accompli
Sitemap soumis à Google	✓ Accompli
HTTPS activé	✓ Accompli
Score SEO : 100/100	✓ Accompli

Tableau C.5 : Comparaison des métriques clé Mobile/Desktop.

Le site est hébergé sur Vercel et mis à jour automatiquement à chaque modification du dépôt GitHub, garantissant une synchronisation continue entre le code source et la version en ligne.

C.7 Accès direct à l'interface Web via QR Code

Afin de faciliter l'accès à l'interface Web développée dans le cadre de ce projet, un code QR a été généré à partir de l'adresse publique du site déployé sur la plateforme Vercel. Le scan de ce code à l'aide d'un smartphone permet d'accéder directement au tableau de bord de supervision sans saisie manuelle de l'adresse URL.

La Figure I.1 présente le code QR associé à l'interface Web de la station météorologique IoT.



Figure C.1 : Code QR permettant l'accès direct à l'interface Web de la station météorologique déployée sur Vercel.

Nous présentons dans les figures suivantes quelques captures d'écran prise par téléphone, de l'interface Web de la station météorologique que nous avons développé :

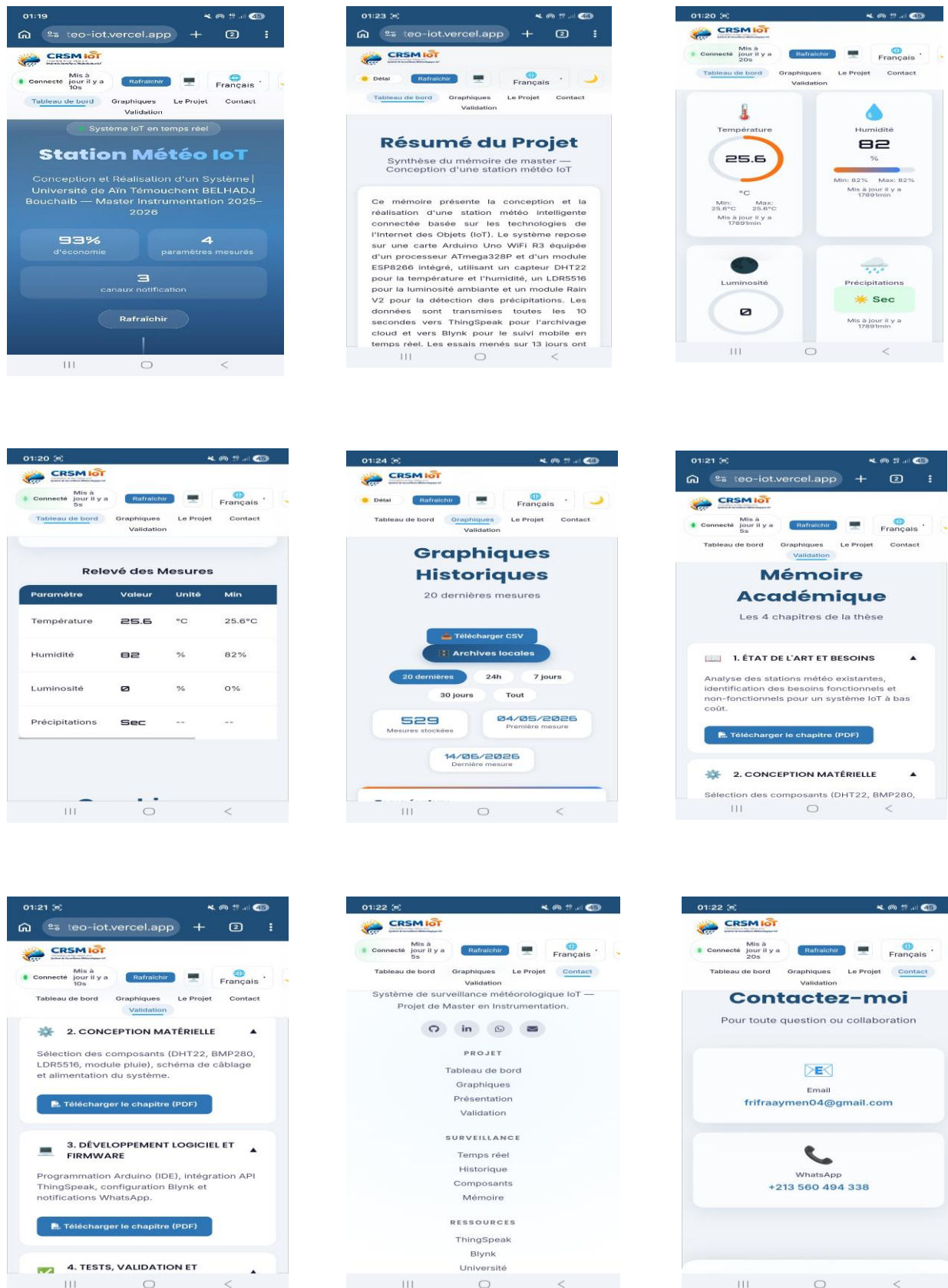


Figure C.2 : Quelques captures d'écran (sur téléphone) de l'interface Web de la station météorologique que nous avons développée.