

République du Cameroun

Paix-Travail-Patrie

Ecole National Supérieure des
Postes Télécommunication et
des technologies de
l'information et de la
communication



Republic of Cameroon

Peace-Work-Fatherland

National Advanced School of
Posts, Telecommunication and
Information and Communication
Technologies

RAPPORT DE TRAVAUX **PRATIQUES**

Cours : Programmation des applications réseaux

**Intitulé du TP : Conception et réalisation d'une plateforme
de monitoring IoT en temps réel.**

Membres du groupe :

- **ELIMBI EYANGO**
- **HAOUA BASSIROU**
- **HAROUNA SOULEY**
- **SINGUE AYISSI**

Supervisé par : Dr NZEBOP

ANNEE ACADEMIQUE:

2025-2026

INTRODUCTION

L'essor de l'Internet des Objets (IoT) transforme progressivement notre manière d'interagir avec les environnements physiques. Aujourd'hui, la capacité de mesurer, transmettre et analyser des données en temps réel constitue un levier majeur dans les domaines de la sécurité, de l'automatisation et de la supervision intelligente.

C'est dans cette perspective que s'inscrit le présent travail pratique, dont l'objectif principal est la conception d'une **plateforme de monitoring IoT interactive**, capable de collecter des données environnementales via des capteurs électroniques, de les transmettre à un serveur et de les restituer graphiquement à travers une interface web accessible en réseau local.

Ce projet repose sur une **double synergie technologique** :

- Une couche **matérielle et embarquée** assurée par Arduino et les capteurs,
- Une couche **logicielle et réseau** assurée par un serveur Node.js et une interface web dynamique.

Ainsi, ce travail illustre concrètement la convergence entre électronique, programmation réseau et technologies web modernes.

PARTIE I : SYSTÈME IoT ET ACQUISITION DES DONNÉES

Cette section présente l'aspect matériel du projet. Elle concerne la connexion des différents capteurs à l'Arduino et la mesure des grandeurs physiques réelles, telles que la distance, la température et l'humidité. Ces capteurs permettent de collecter des données fiables et précises sur l'environnement. Ces informations serviront de fondation pour le traitement ultérieur et pour l'affichage sur la plateforme web, garantissant ainsi que le système reflète fidèlement les conditions réelles et facilite la prise de décision basée sur des mesures concrètes.

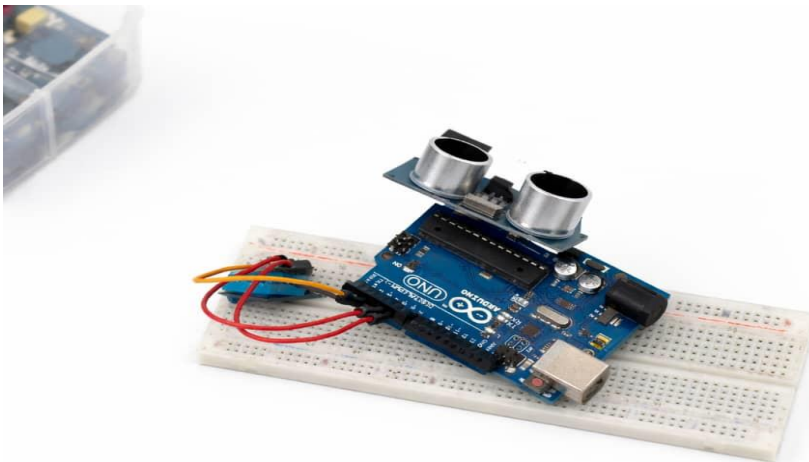
➤ Architecture Matérielle

Le cœur du système repose sur une carte **Arduino**, jouant le rôle de microcontrôleur central.

Elle assure la lecture périodique des capteurs, le traitement des données et leur transmission vers le serveur.

Le montage comprend:

- Une carte Arduino,
- Un capteur ultrason HC-SR04,
- Un capteur DHT11 (température et humidité),
- Des connexions filaires,
- Une alimentation USB.



➤ Capteurs et Mesures Physiques

- **Capteur Ultrason – HC-SR04**

Ce capteur mesure la distance sans contact grâce à l'émission d'ondes ultrasonores. Il permet la détection d'obstacles et le déclenchement d'alertes de proximité, ce qui introduit une dimension de **surveillance active** dans le système.

- **Capteur Température – DHT11**

Il fournit la température ambiante en degrés Celsius. Son intégration permet une observation thermique continue de l'environnement surveillé.

- **Capteur Humidité – DHT11**

Associé au capteur thermique, il mesure l'humidité relative en pourcentage, offrant ainsi une vision globale des conditions atmosphériques.

➤ Programmation Embarquée

Le programme Arduino assure :

- La lecture cyclique des capteurs,
- La conversion des données en format numérique,
- L'encapsulation des valeurs sous forme **JSON**,
- L'envoi vers le serveur via le port série.

PARTIE II : PLATEFORME WEB ET COMMUNICATION RÉSEAU

Cette partie présente l'intelligence du système : comment les données collectées par les capteurs sont transmises, traitées et affichées en temps réel sur la plateforme web. L'objectif est de rendre les informations accessibles à distance, fiables et exploitables pour l'utilisateur.

➤ Déploiement Serveur

Le serveur, développé en **Node.js**, agit comme intermédiaire entre le monde physique et l'interface utilisateur.

Il reçoit les données série, les traite et les diffuse instantanément aux clients connectés. Technologies mobilisées:

- Node.js
- Express.js
- WebSocket
- HTML / CSS / JavaScript
- SerialPort
-

➤ Mise en Réseau et Accessibilité

✓ Accès à l'interface web du système

Le système développé intègre une plateforme web permettant la supervision et le contrôle des équipements connectés.

Lorsque les appareils sont reliés au même réseau local (Wi-Fi ou câble Ethernet), l'utilisateur peut accéder à l'interface en entrant l'adresse suivante dans un navigateur web : **http://192.168.1.252:3000/login**

- **192.168.1.252** : représente l'adresse IP locale du serveur hébergeant l'application web.
- **3000** : correspond au port de communication utilisé par le serveur Node.js pour écouter les requêtes.
- **/login** : indique la page d'authentification où l'utilisateur doit saisir son identifiant et son mot de passe.

Cette interface permet:

- La visualisation des données des capteurs en temps réel,
- administration du système,
- La sécurisation de l'accès grâce à une authentification.

L'intégration d'une interface web transforme un simple montage électronique en une solution intelligente et connectée. Elle offre non seulement un suivi en temps réel, mais aussi une expérience utilisateur moderne, démontrant la synergie entre l'électronique embarquée et le développement web. Ce lien constitue donc le cœur opérationnel du système, garantissant accessibilité, sécurité et efficacité.

✓ Interfaces Graphiques et Expérience Utilisateur

La plateforme propose une interface sombre, moderne et ergonomique, organisée autour de trois composantes majeures :

✓ Interface de Connexion

Elle assure la sécurité du système grâce à un mot de passe, empêchant les accès non autorisés.

La plateforme intègre :

- authentification par mot de passe,
- gestion de sessions,
- protection contre accès non autorisé.



✓ Tableau de Bord

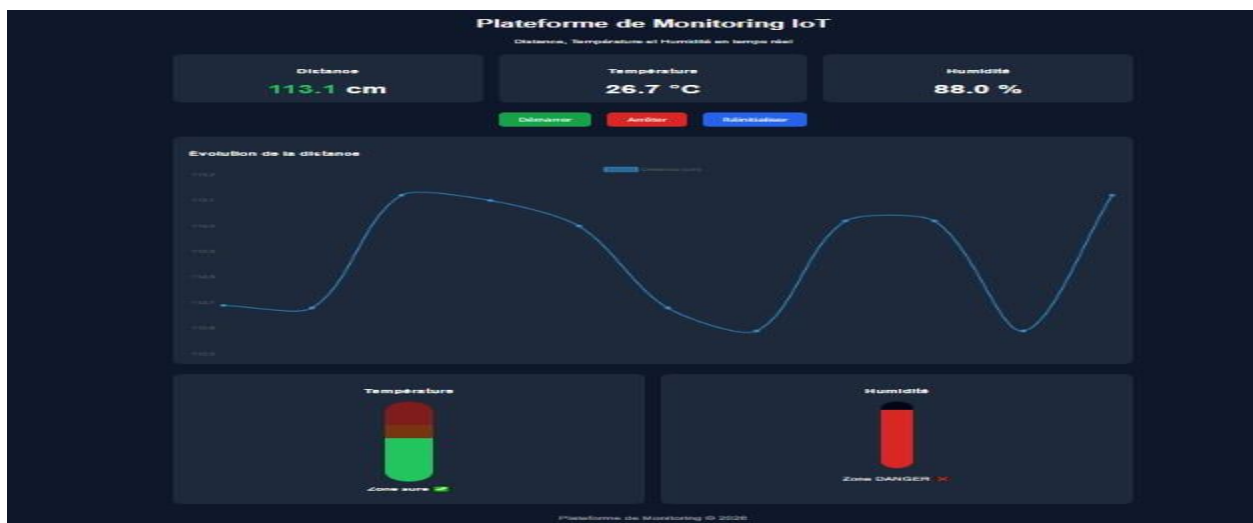
Cœur visuel de la plateforme, il affiche :

- Distance en cm,
- Température en °C,
- Humidité en %,
- Boutons de contrôle,
- Graphique d'évolution,
- Jauges colorées intuitives.



✓ Visualisation Dynamique

Les graphiques évolutifs permettent d'observer les tendances et de détecter immédiatement toute anomalie environnementale.



CONCLUSION

Au terme de ce projet, il apparaît clairement que nous n'avons pas seulement assemblé des capteurs et écrit quelques lignes de code : nous avons donné naissance à un **système vivant**, capable de ressentir son environnement, de transmettre l'information et d'interagir avec l'utilisateur en temps réel. La combinaison entre la partie **Arduino IoT** et la **plateforme web** transforme un simple montage électronique en une solution complète, moderne et intelligente. Les données ne restent plus enfermées dans un microcontrôleur : elles voyagent, s'affichent, se sécurisent et deviennent utiles.

L'accès via le réseau local et l'interface de connexion montre que le projet ne se limite pas à la technique brute, mais s'ouvre vers une logique professionnelle où la **sécurité, l'ergonomie et la supervision à distance** prennent toute leur importance. Chaque capteur joue son rôle comme un sens humain, tandis que l'interface web agit comme le cerveau qui observe, analyse et décide.

Ce travail illustre parfaitement l'esprit de l'Internet des Objets : un univers où les objets ne sont plus passifs, mais **connectés, communicants et intelligents**. Au-delà de l'aspect académique, ce projet représente une passerelle vers des applications réelles maisons intelligentes, systèmes de surveillance, gestion d'énergie ou solutions industrielles.

En somme, ce n'est pas seulement un rapport de travaux pratiques, c'est la démonstration qu'avec de la logique, de la rigueur et un peu de créativité, la technologie peut devenir un outil puissant capable de relier le monde physique au monde numérique avec élégance et efficacité.

ANNEXES : Explication des codes

➤ Annexe A : Code Arduino

partie du code	Extrait de code	Rôle / Explication
Bibliothèque DHT	<code>#include "DHT.h"</code>	Permet d'utiliser le capteur de température et d'humidité.
Déclaration DHT	<code>#define DHTPIN A0</code> <code>#define DHTTYPE DHT11</code>	Indique la broche utilisée et le type de capteur.
Déclaration Ultrason	<code>#define TRIG_PIN A3</code> <code>#define ECHO_PIN A4</code>	Définit les broches pour envoyer et recevoir l'onde ultrasonore.
Alimentation capteur	<code>VCC_PIN A2</code> <code>GND_PIN A5</code>	Permet d'alimenter le capteur directement via l'Arduino.
Setup	<code>Serial.begin(9600);</code>	Initialise la communication série avec l'ordinateur.
Mesure distance	<code>pulseIn(ECHO_PIN, HIGH)</code>	Calcule le temps de retour de l'onde pour obtenir la distance.
Lecture Temp/Hum	<code>dht.readHumidity()</code> <code>dht.readTemperature()</code>	Lit les données du capteur DHT.
Format JSON	<code>Serial.print("{\"distance\":");</code>	Transforme les données en format JSON compréhensible par le serveur.
Delay	<code>delay(1000);</code>	Rafraîchissement toutes les 1 seconde.

➤ Annexe B : Serveur Node.js

Partie	Extrait	Explication
Import modules	<code>require("express")</code>	Framework serveur web.
SerialPort	<code>new SerialPort({ path:"COM3" })</code>	Récupère les données Arduino via USB.
Parser	<code>ReadlineParser</code>	Lit les données ligne par ligne.
JSON réception	<code>JSON.parse(line)</code>	Convertit les données texte en objet exploitable.
WebSocket	<code>wss.clients.forEach</code>	Envoie les données en temps réel aux navigateurs.
Port serveur	<code>server.listen(3000)</code>	Rend le site accessible sur le réseau local.
Session	<code>express-session</code>	Gère la connexion utilisateur.
Mot de passe	<code>bcrypt.hashSync</code>	Sécurise le mot de passe par hachage.
Route login	<code>/login</code>	Page de connexion.
Route monitoring	<code>/monitoring</code>	Accès au tableau de bord.
Logout	<code>/logout</code>	Déconnexion sécurisée.

➤ Annexe C: Interface Web

Partie	Élément	Explication
Chart.js	<code><script src="chart.js"></code>	Bibliothèque graphique pour tracer la courbe de distance.
Cards	Distance / Temp / Humidité	Affichage numérique en temps réel.
Boutons	Start / Stop / Reset	Contrôle de la connexion WebSocket.
Jauge température	temp-fill	Barre verticale colorée selon seuils.
Jauge humidité	humBar	Indicateur de taux d'humidité.
WebSocket	<code>new WebSocket(ws://...)</code>	Réception des données du serveur.
Alerte sonore	alert.mp3	Signal sonore si distance trop faible.
Graphique	<code>Chart(type:"line")</code>	Historique de distance.
Zones couleurs	Vert / Orange / Rouge	Indication d'état sécurisé ou dangereux.

➤ Annexe D: Login

Élément	Code	Rôle
Formulaire	<code><form method="POST"></code>	Envoie le mot de passe au serveur.
Champ password	<code><input type="password"></code>	Masque le mot de passe.
Bouton	Se connecter	Déclenche la vérification serveur.
CSS sombre	<code>background #0f172a</code>	Interface moderne et professionnelle.
Sécurité	POST /login	Évite l'exposition du mot de passe dans l'URL.