

STUDIO SHODWE

INTERNET OF THINGS

A R R O I O T

MEMBRE DU GROUPE :



ARRO IOT

ABOUT US

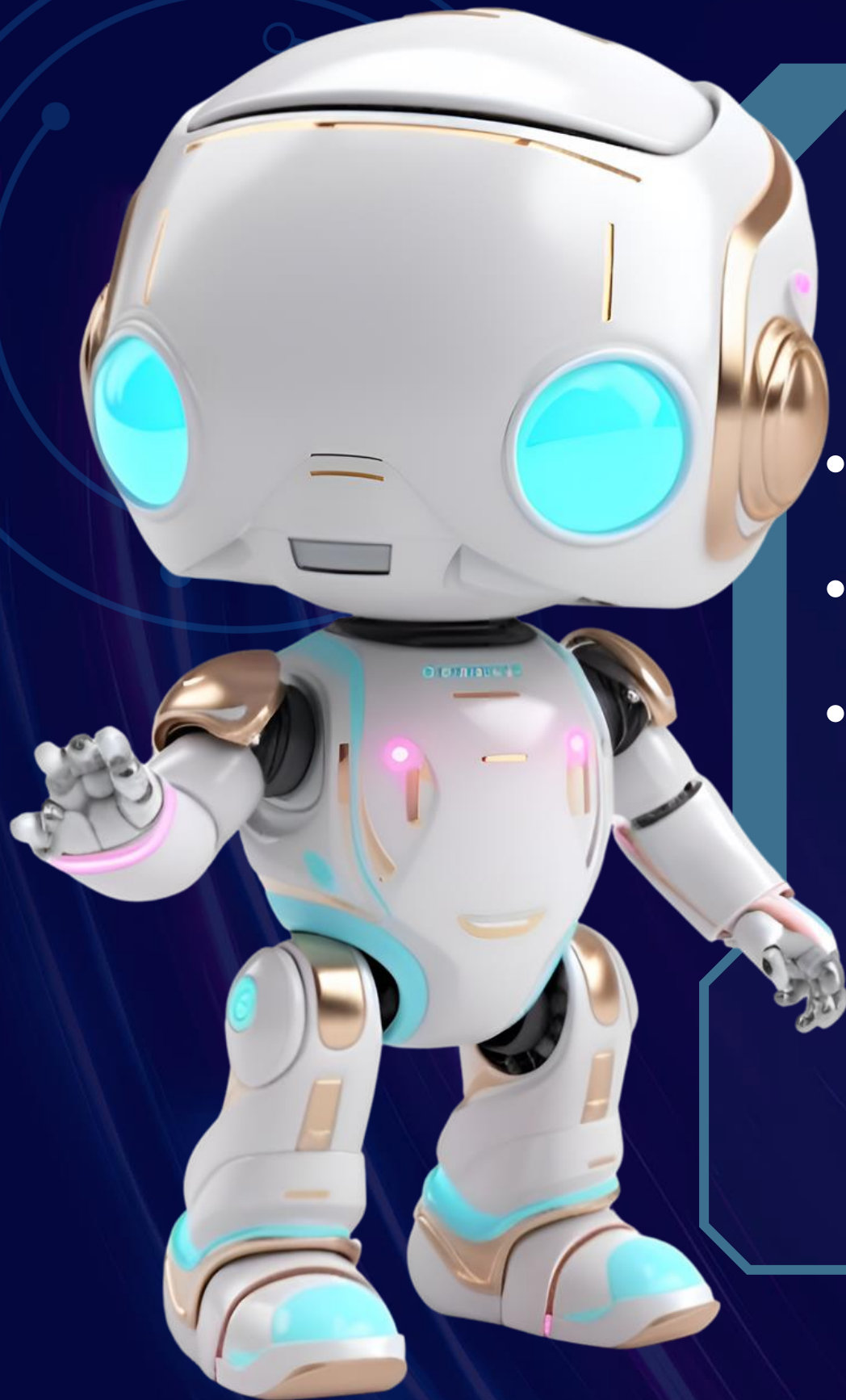
BIENVENUE CHEZ ARRO IOT, LÀ OÙ L'INNOVATION RENCONTRE LA DURABILITÉ. NOUS SOMMES UNE ENTREPRISE SPÉCIALISÉE DANS LE DÉVELOPPEMENT D'OBJETS CONNECTÉS DÉDIÉS AUX SYSTÈMES D'ARROSAGE INTELLIGENTS. NOTRE MISSION EST DE TRANSFORMER LA MANIÈRE DONT L'EAU EST UTILISÉE DANS L'AGRICULTURE ET LES ESPACES VERTS GRÂCE À LA TECHNOLOGIE IOT.



ARRO IOT

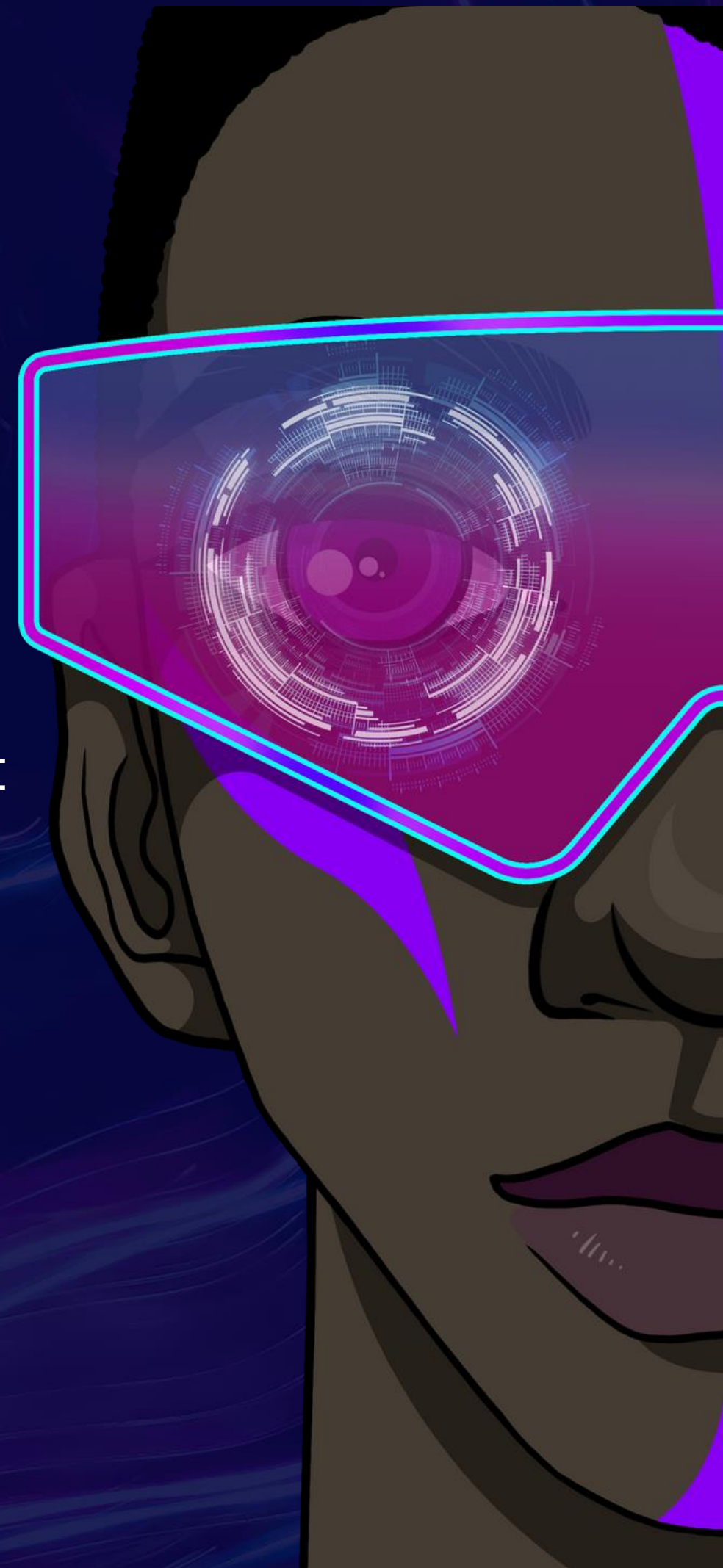
NOS ENDEUX ET OBJECTIFS

- INNOVER POUR L'AVENIR
- CONSERVATION DES RESSOURCES
- ACCESSIBILITÉ ET SIMPLICITÉ



PROBLÉMATIQUE

Comment concevoir et mettre en œuvre un système d'arrosage automatique intelligent qui utilise les technologies IoT et Cloud pour assurer une gestion optimale et durable de l'eau, tout en garantissant la sécurité et la fiabilité des données collectées et des composants matériels et logiciels utilisés ?



ARRO IOT

GESTION DE PROJET ET COMMUNICATION

- Tableau Trello : "Backlog", "À faire", "En cours", "En revue", "Terminé".
- Cartes
- Étiquettes



Suivi et Communication

- Réunions Hebdomadaires
- Communication Continue



ARRO IOT

GESTION DE PROJET

ORGANISATION DE L'ÉQUIPE

Rôle	Nom	Responsabilités
Développeur Principal	Nadine	Coordination globale, gestion des parties prenantes et suivi des progrès.
Responsable Matériel	Ahmed	Développement du firmware pour l'ESP32 et intégration des capteurs.
Responsable matériel	Salah	Sélection et configuration des composants matériels (DHT11, ESP32, écran LCD).
Ingénieur Logiciel	Thinhinane	Dev des logiciels naissaicaires y compris les bibliotheques et les scripts
Analyste de Données	Querdia	Collecte, analyse et visualisation des données.
Documentaliste	Sadok	Documentation du projet, y compris les rapports et les manuels d'utilisation.

PLANIFICATION DU PROJET AVEC LA MÉTHODOLOGIE AGILE

Sprint	Dates	Tâches
Sprint 1	01/07/2024 - 07/07/2024	Planification initiale, configuration des outils, définition des rôles
Sprint 2	08/07/2024 - 14/07/2024	Développement initial, configuration matérielle, tests des capteurs et de l'écran LCD
Sprint 3	15/07/2024 - 21/07/2024	Intégration du code et des composants, tests de fonctionnalité
Sprint 4	22/07/2024 - 28/07/2024	Optimisation du code, rédaction de la documentation utilisateur
Sprint 5	29/07/2024 - 04/08/2024	Finalisation des tests, préparation de la présentation et des livrables



GESTION DES RISQUES

Risques	Description	Stratégies
Risques Techniques	Problèmes matériels ou logiciels inattendus.	Tests réguliers, prototypage et validation itérative.
Risques de Planification	Délais non respectés.	Ajustements flexibles du calendrier, réévaluation des priorités.
Risques de Communication	Mauvaise communication entre les membres de l'équipe	Utilisation de Trello et Slack pour une communication claire et continue, réunions régulières.

GESTION DES INCIDENTS

Aspect	Objectifs	Mesures Mises en Place
Détection des Incidents	Mettre en place des mécanismes pour détecter rapidement les incidents de sécurité.	Système d'Alerte : Utiliser des systèmes d'alerte pour détecter et signaler les incidents de sécurité en temps réel.
Réponse aux Incidents	Établir des procédures de réponse aux incidents pour limiter les dommages et restaurer le système à son état normal.	Plan de Réponse aux Incidents : Développer un plan de réponse aux incidents détaillant les actions à entreprendre en cas d'incident de sécurité.
Enregistrement des Incidents	Documenter les incidents de sécurité pour analyser leur cause et améliorer les mesures de sécurité.	Analyse Post-Incident : Effectuer des analyses post-incident pour comprendre les causes des incidents et éviter qu'ils ne se reproduisent.

DIFFICULTÉ RENCONTRÉE



- PROBLÈME DE COMPATIBILITÉ AVEC ARDUINO CLOUD

CONSEQUENCES

- ABSENCE DE DONNÉES RÉELLES
- TESTS LIMITÉES

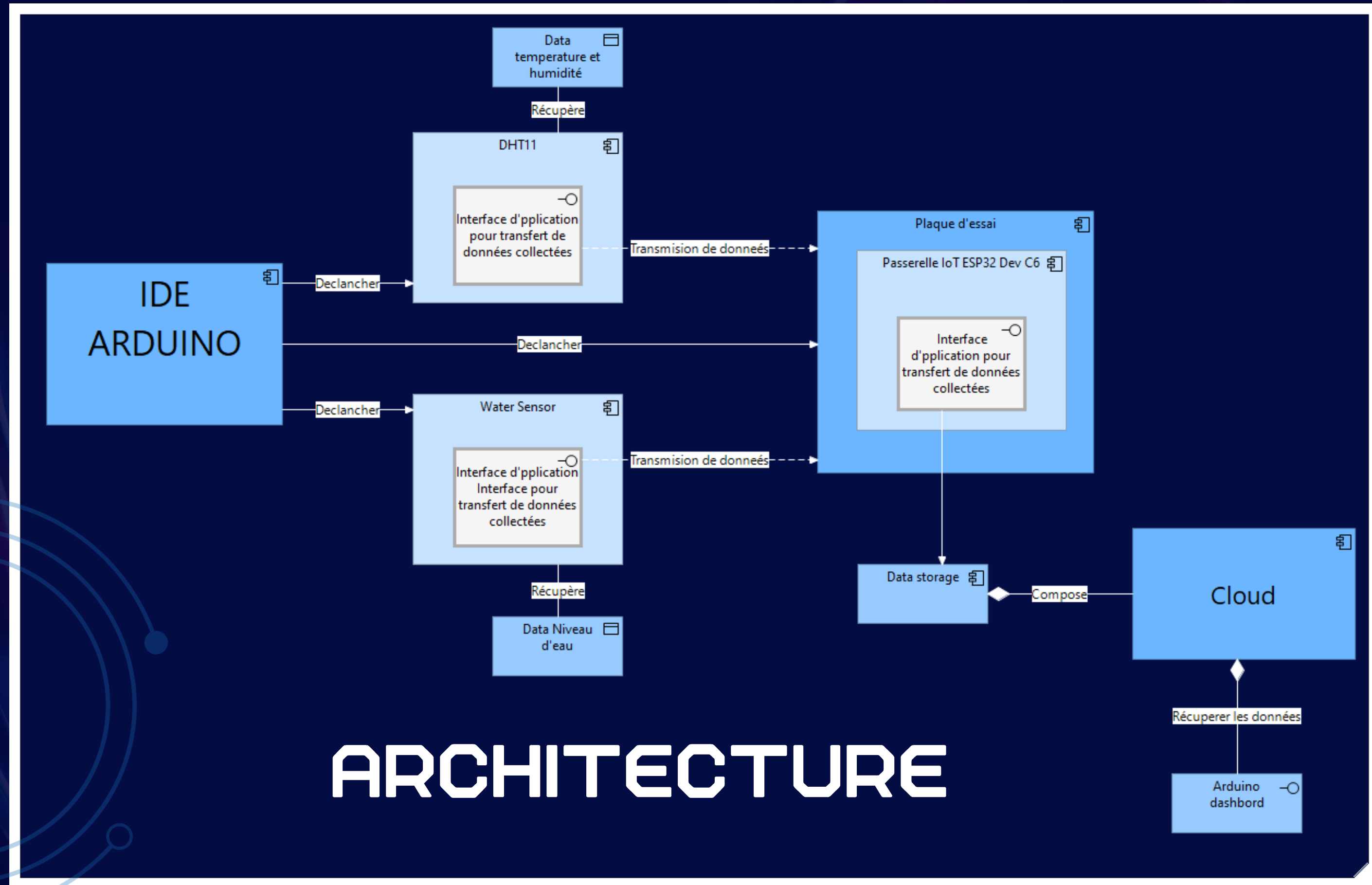
SOLUTIONS ENVISAGÉES

- PASSAGE À UNE VERSION SUPÉRIEURE
- Utilisation d'Autres Plateformes



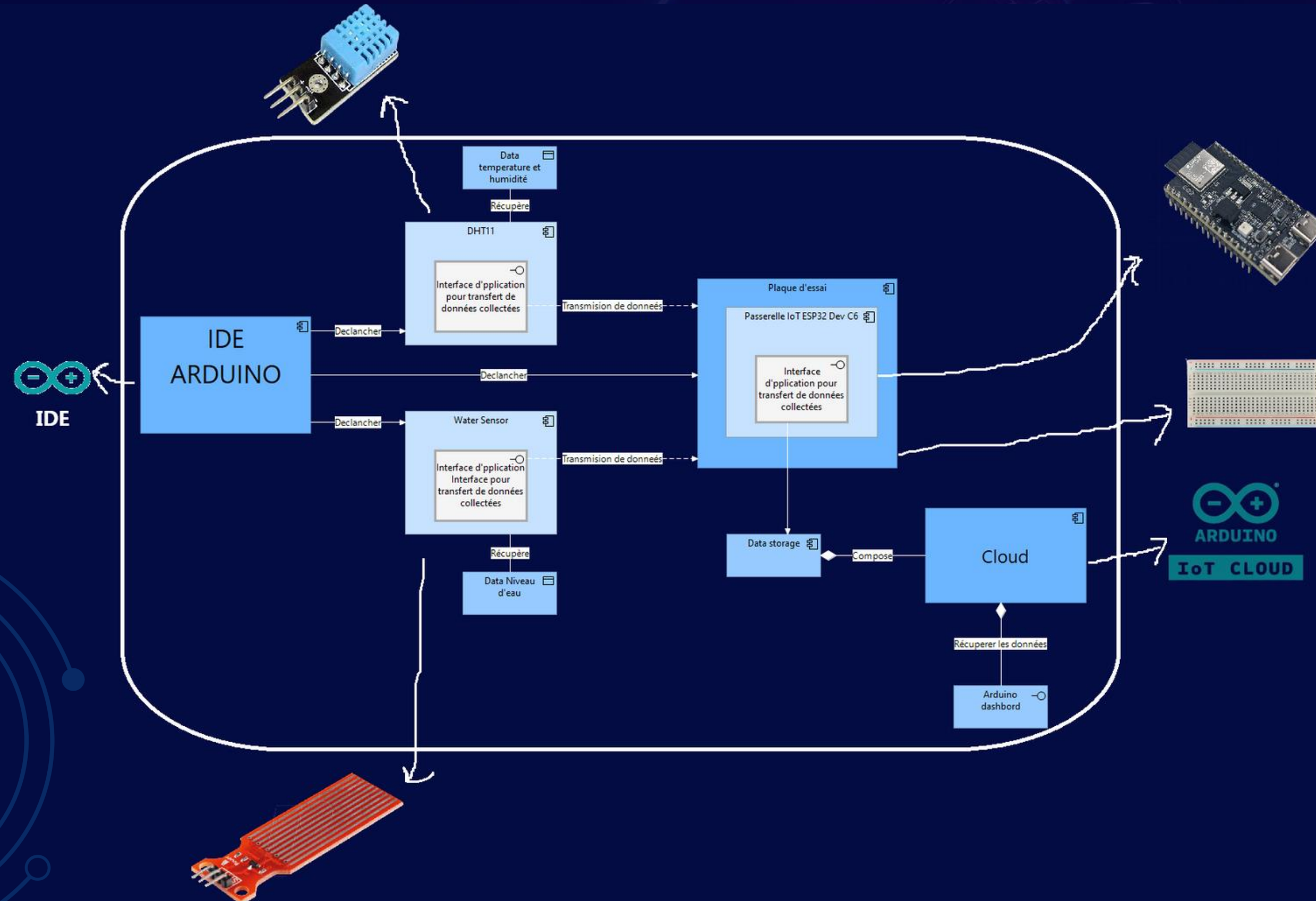
Aspect	Objectifs	Mesures Mises en Place
Sécurité des Données	<p>Confidentialité : Assurer que seules les personnes autorisées peuvent accéder aux données.</p> <p>Intégrité : Protéger les données contre toute modification ou altération non autorisée.</p> <p>Disponibilité : Assurer que les données sont accessibles aux utilisateurs autorisés quand ils en ont besoin.</p>	<p>Cryptage des Données : Utilisation de techniques de cryptage pour protéger les données pendant la transmission entre les capteurs et le système central.</p> <p>Authentification et Autorisation : Mise en place de mécanismes d'authentification pour vérifier l'identité des utilisateurs et contrôler l'accès aux données.</p> <p>Sauvegarde : Implémentation de procédures de sauvegarde régulières pour prévenir la perte de données en cas de défaillance du système.</p>
Sécurité Matérielle	<p>Protection Physique : Protéger les composants matériels contre les dommages physiques et les manipulations non autorisées.</p> <p>Surveillance et Maintenance : Mettre en place des procédures de surveillance et de maintenance régulières pour assurer le bon fonctionnement des composants matériels.</p>	<p>Boîtier de Protection : Utilisation de boîtiers pour protéger les capteurs et l'ESP32 contre les environnements hostiles (humidité, poussière, chocs).</p> <p>Câblage Sécurisé : S'assurer que les câbles sont correctement isolés et protégés contre les courts-circuits et les interférences électromagnétiques.</p> <p>Plan de Maintenance : Établir un plan de maintenance régulier pour vérifier l'état des composants et remplacer ceux qui sont défectueux ou usés.</p>
Sécurité Logicielle	<p>Robustesse du Code : Écrire du code robuste et résistant aux erreurs pour éviter les bugs et les vulnérabilités.</p> <p>Mises à Jour de Sécurité : Garder les bibliothèques et les logiciels utilisés à jour avec les derniers correctifs de sécurité.</p> <p>Tests et Vérifications : Effectuer des tests de sécurité réguliers pour identifier et corriger les vulnérabilités.</p>	<p>Revue de Code : Effectuer des revues de code régulières pour identifier les vulnérabilités potentielles et les corriger.</p> <p>Tests de Sécurité : Utiliser des outils de test de sécurité pour vérifier la résistance du logiciel aux attaques courantes (injections, débordements de mémoire, etc.).</p> <p>Environnement de Développement Sécurisé : S'assurer que l'environnement de développement est sécurisé et que seuls les membres autorisés ont accès au code source.</p>

ARCHITECTURE



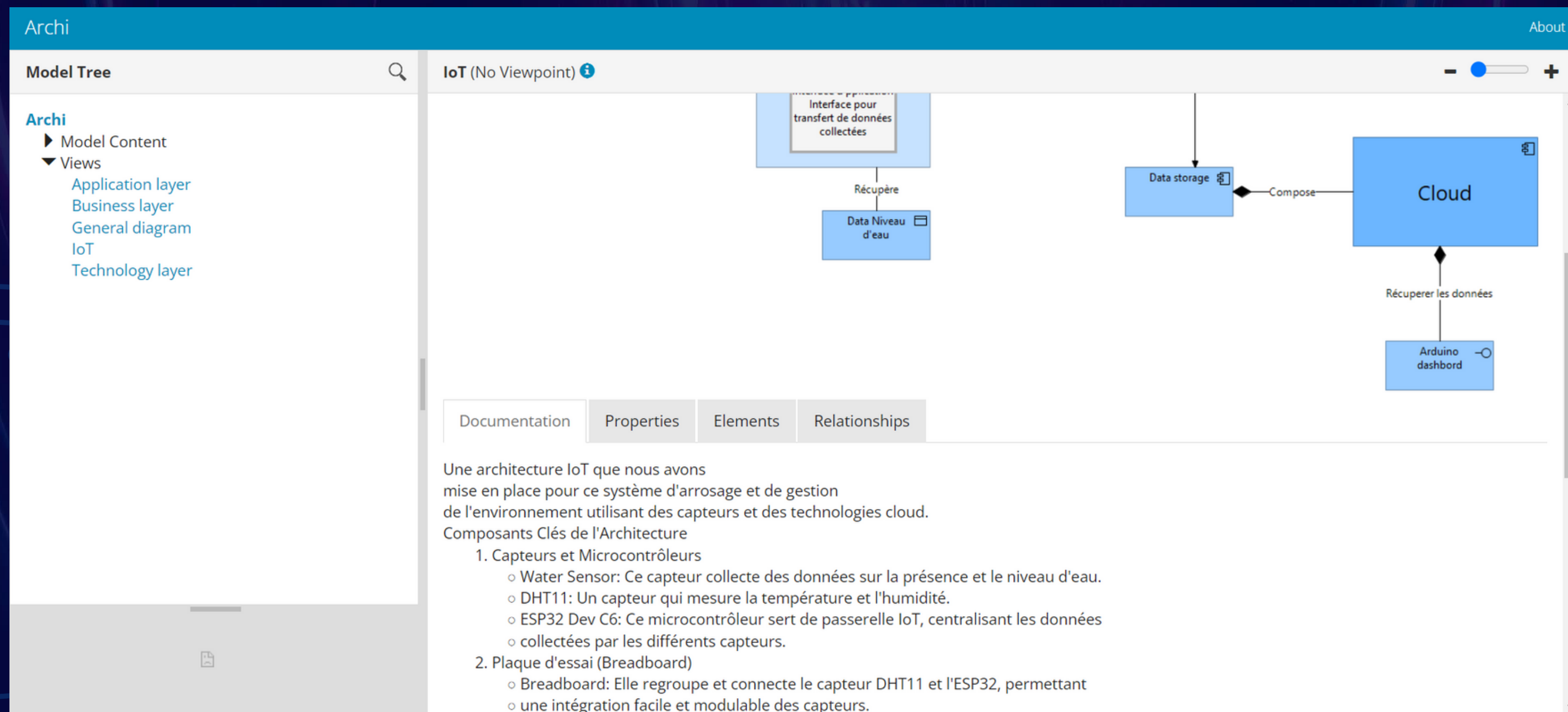
ARCHITECTURE

ARCHITECTURE AVEC COMPOSANT



LIEN GITHUB

HTTPS://GITHUB.COM/SALAHMDK/IOT



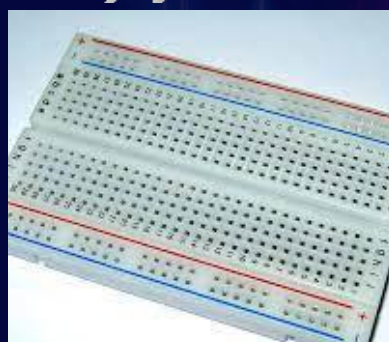
COMPOSANTS



ULTRASONIC



UNE LED



UNE PLAQUE
D'ESSAI



MINI MOTEUR



MODULE RELAIS



UN POTENTIOMÈTRE



KIT DE CABLE



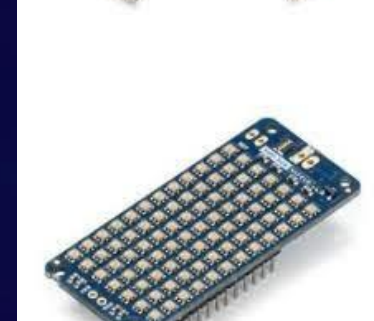
CAPTEUR NIVEAU D'EAU



MICRO
SERVOMOTEUR



CABLE MALE MALE



CARTED'EXTENSION



BOUTON



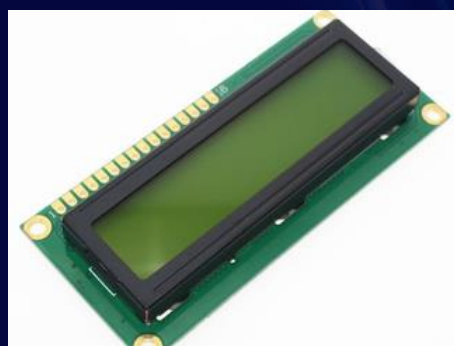
MOTEUR PAS A
PAS



CARTE ARDUINO



ESP 3 2



MODULE
D'AFFICHAGE
LCD



BARETTES DE
CONNEXION



JOYSTICK



BADGE D'ACCES



CARTE D'ACCES



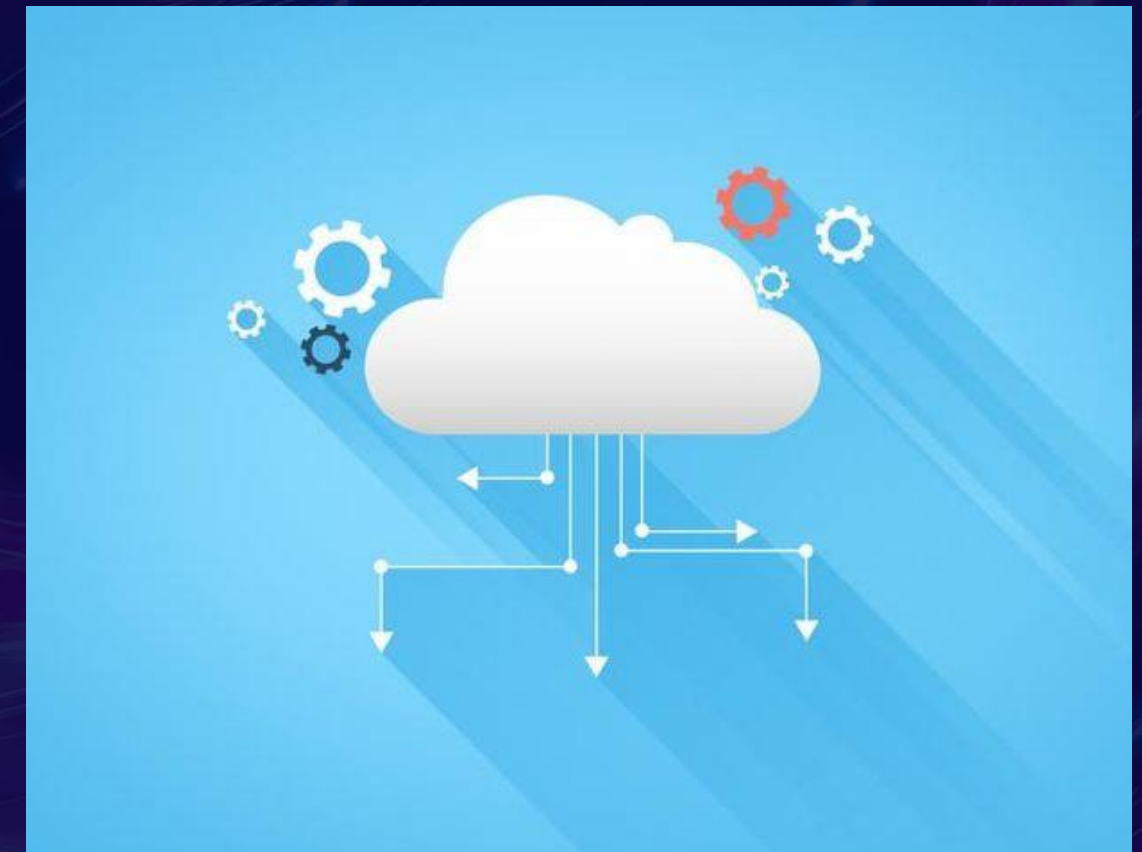
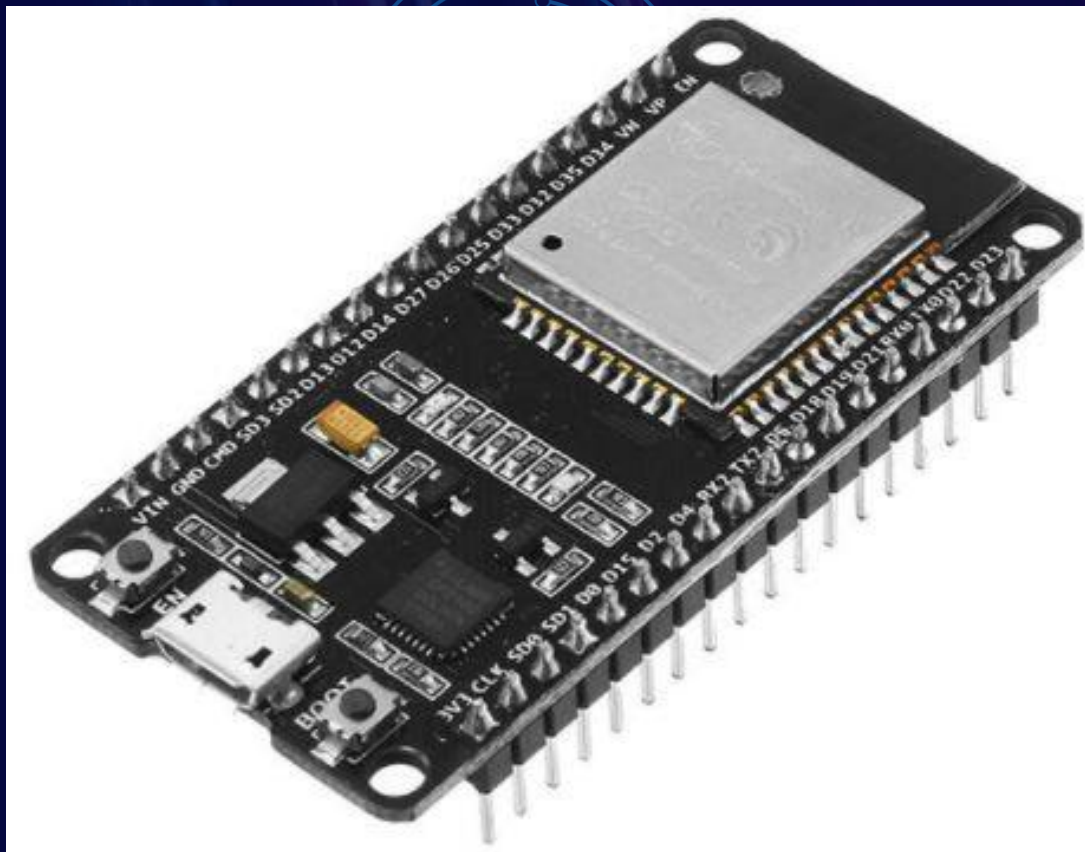
LECTEUR RFID

ARRO IOT RÉALISATION

Carte utilisée : ESP32

IDE Arduino

Arduino Cloud



PARAMÉTRAGE DE LA CARTE ESP32

- SÉLECTIONNER ET INSTALLER LES CARTES ESP32 SUR IDE ARDUINO
- SÉLECTIONNER LE PORT COM ASSOCIÉ
- CONNEXION DE LA CARTE AU RÉSEAU WI-FI ET VÉRIFIER LA CONNECTIVITÉ VIA UN SCRIPT
- ESSAYER LE FONCTIONNEMENT DE LA CARTE EN ESSAYANT DE FAIRE ALLUMER UNE LED





ESP32C6 Dev Module



sketch_jul18a.ino



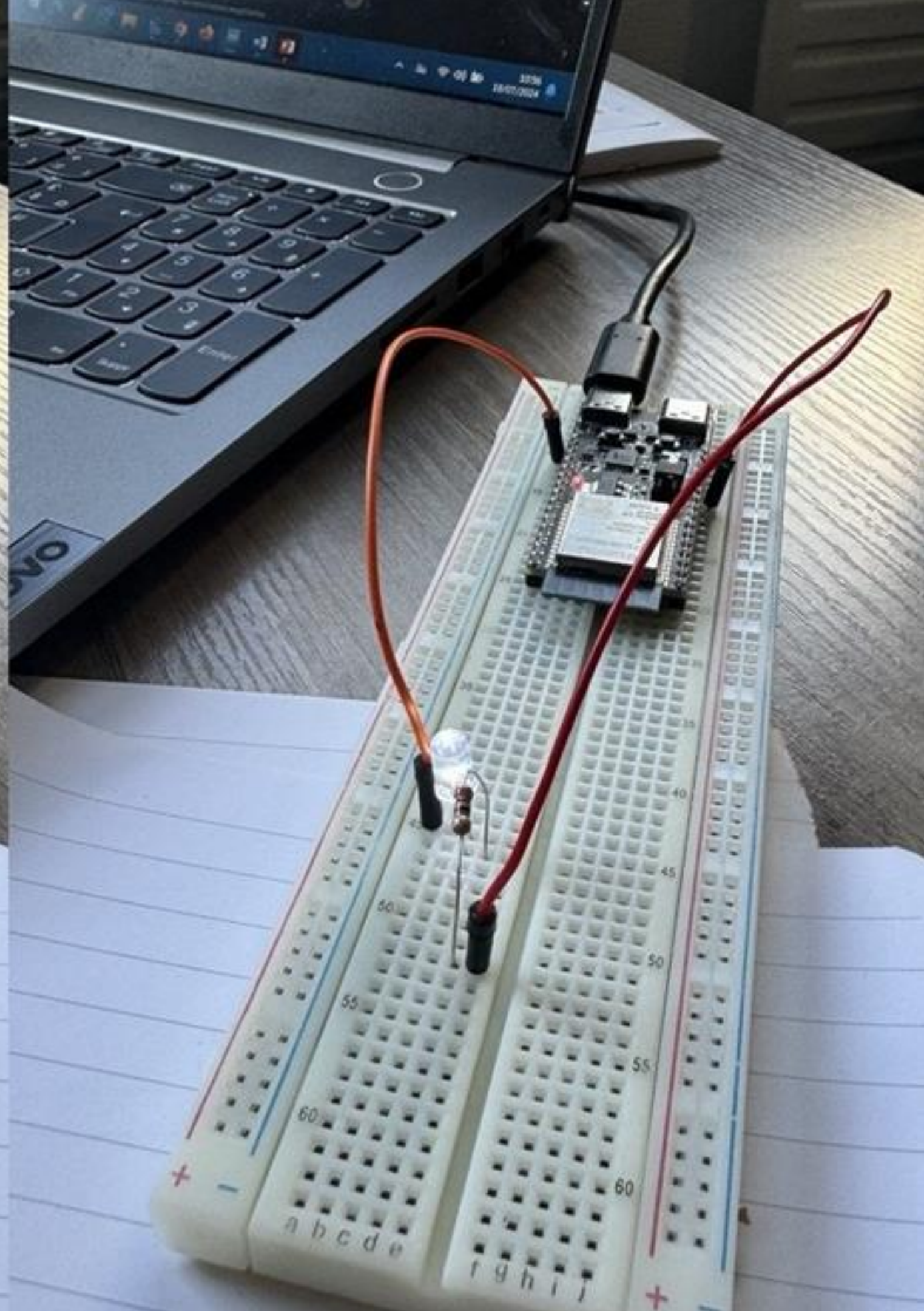
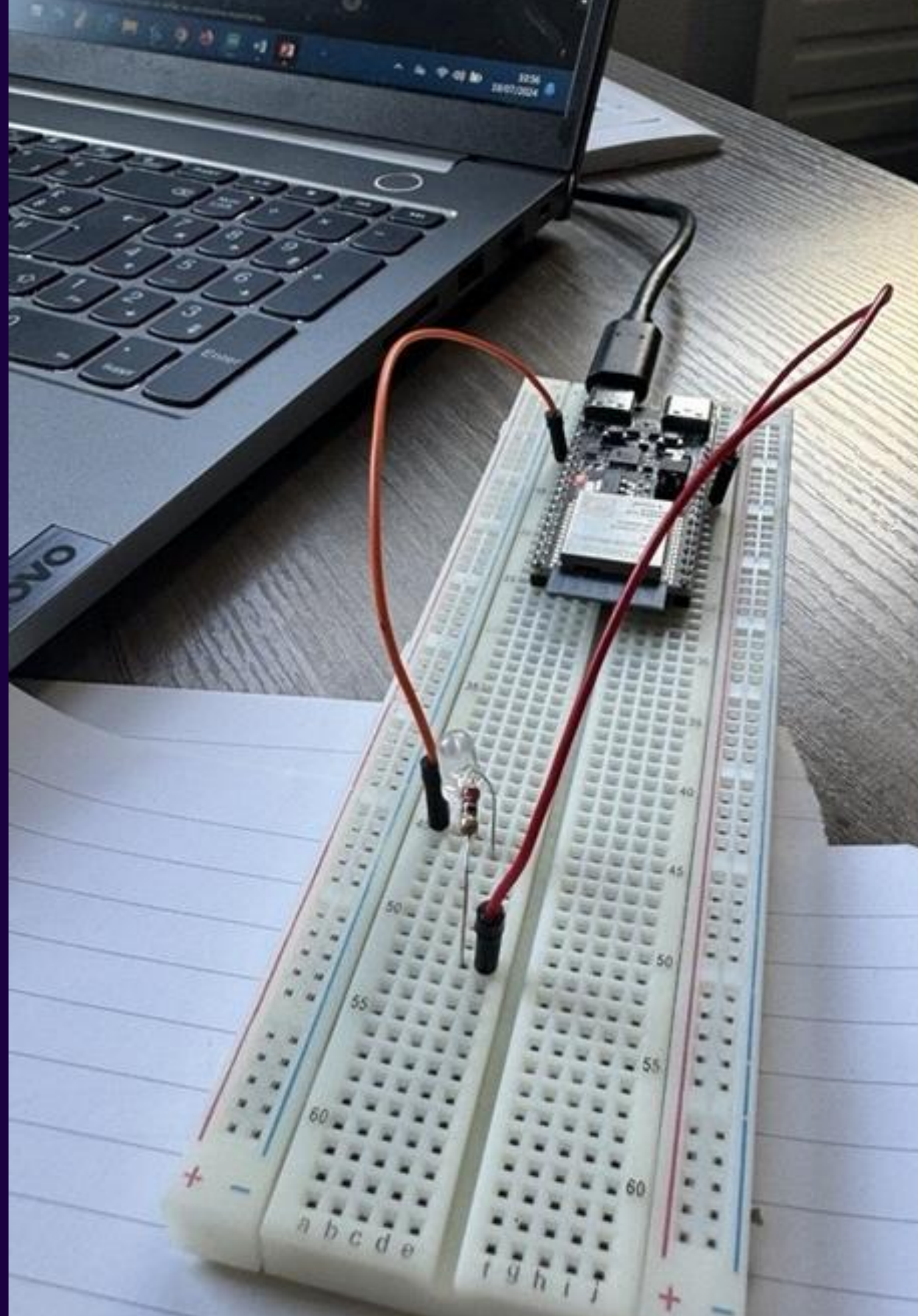
```
1  #include "WiFi.h"
2
3  const char* ssid = "Livebox-D7D0";
4
5  const char* pwd  = "MRo4MEqDrMt6aaDRm5";
6
7
8
9  void setup() {
10
11     Serial.begin(9600);
12
13     Serial.println("Configuration du WiFi");
14
15     WiFi.begin(ssid, pwd);
16
17
18
19     while (WiFi.status() != WL_CONNECTED) {
20
21         delay(500);
22
23         Serial.println("Connexion au WiFi..");
24
25     }
26
27
28
29     Serial.println("Connecté au réseau WiFi");
30
31 }
32
33
```

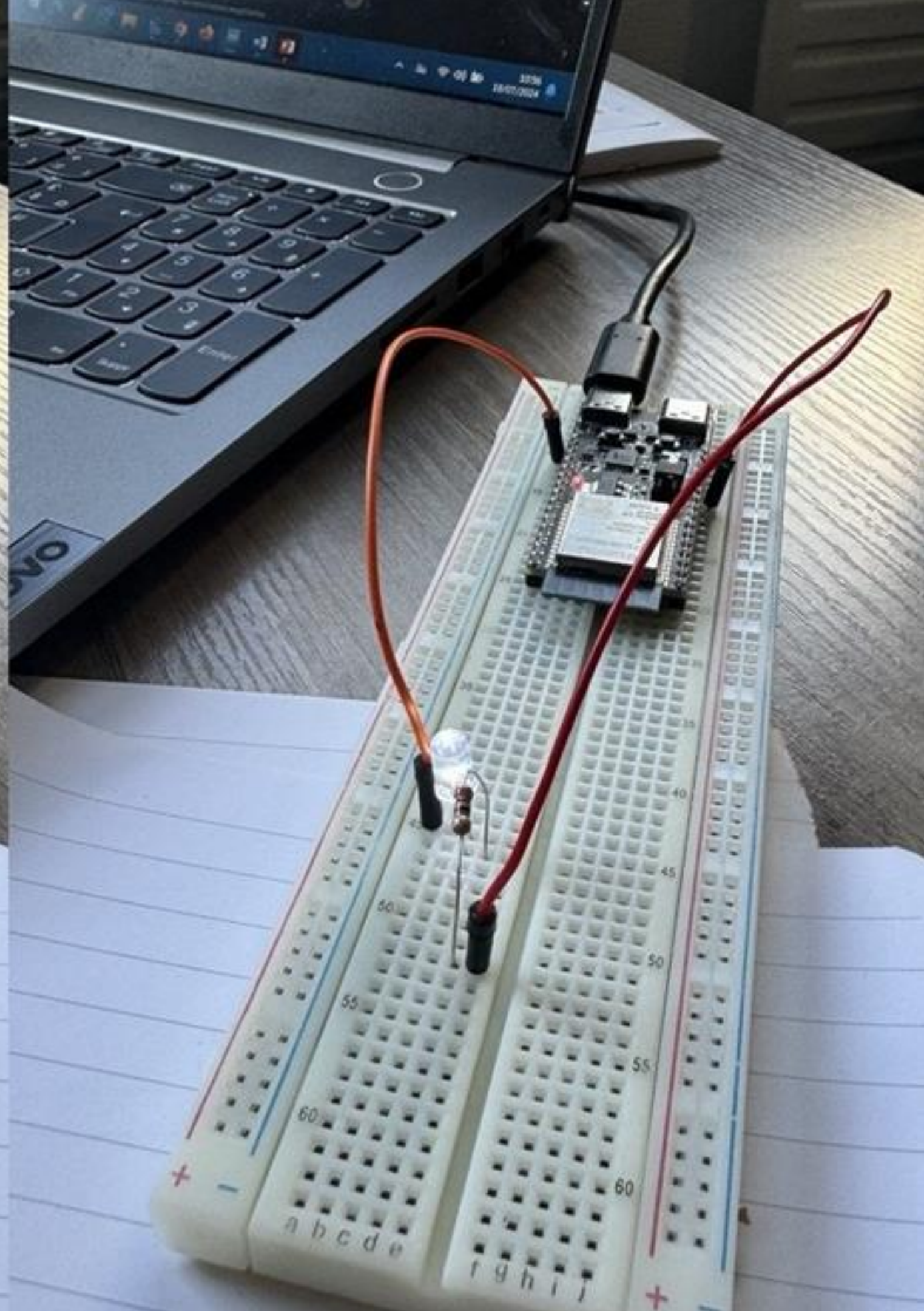
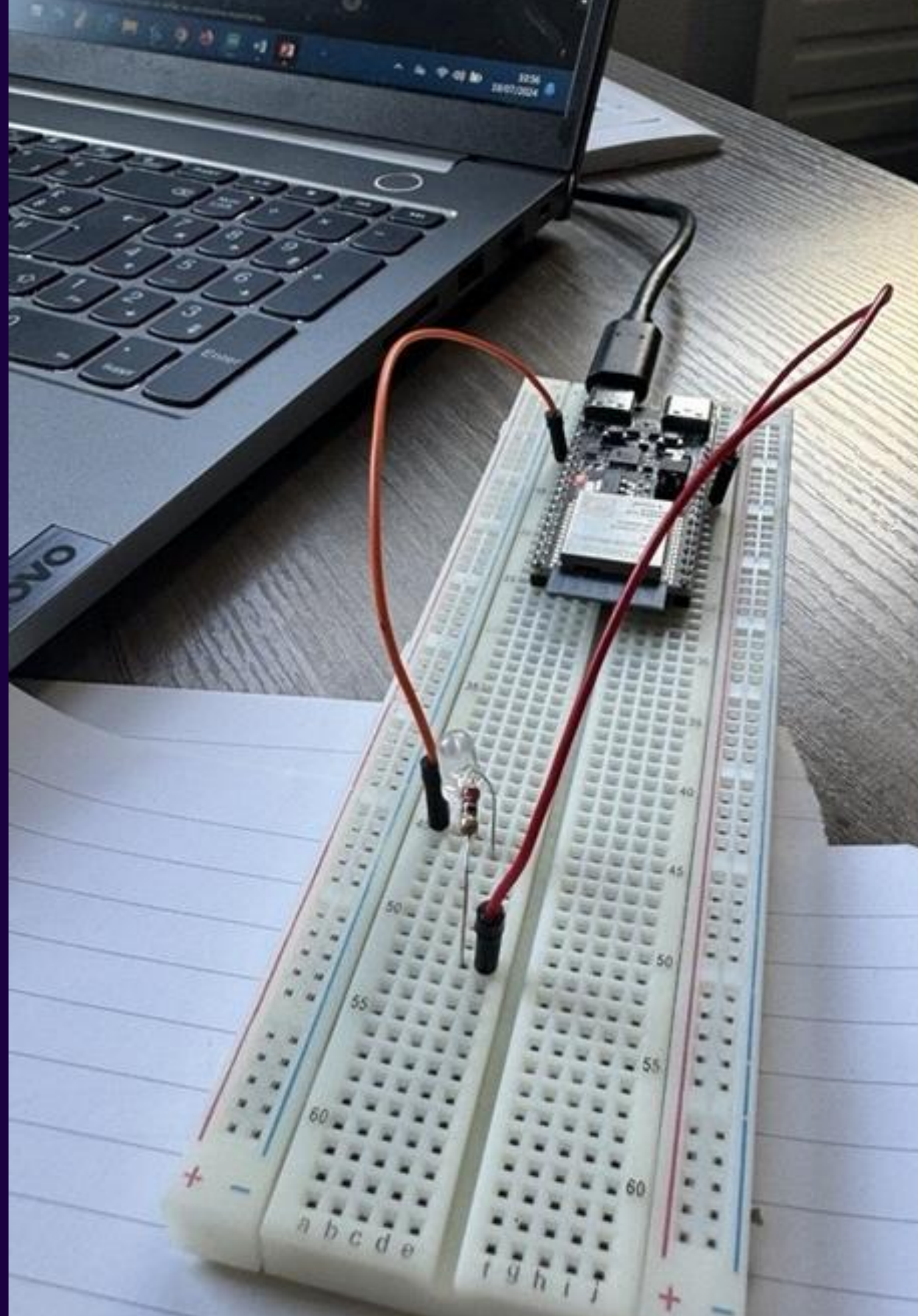
Output Serial Monitor

```
.....  
Writing at 0x000c9ec0... (85 %)  
Writing at 0x000d054f... (88 %)  
Writing at 0x000d63be... (91 %)  
Writing at 0x000dc04b... (94 %)  
Writing at 0x000e1afa... (97 %)  
Writing at 0x000e91ce... (100 %)  
Wrote 892240 bytes (558447 compressed) at 0x00010000 in 3.8 seconds (effective 1867.5 kbit/s)...  
Hash of data verified.  
  
Leaving...  
Hard resetting via RTS pin...
```

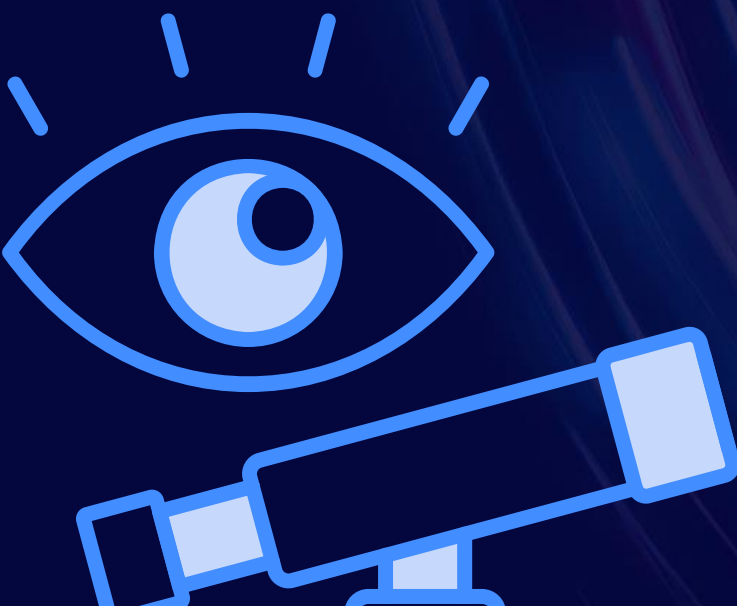
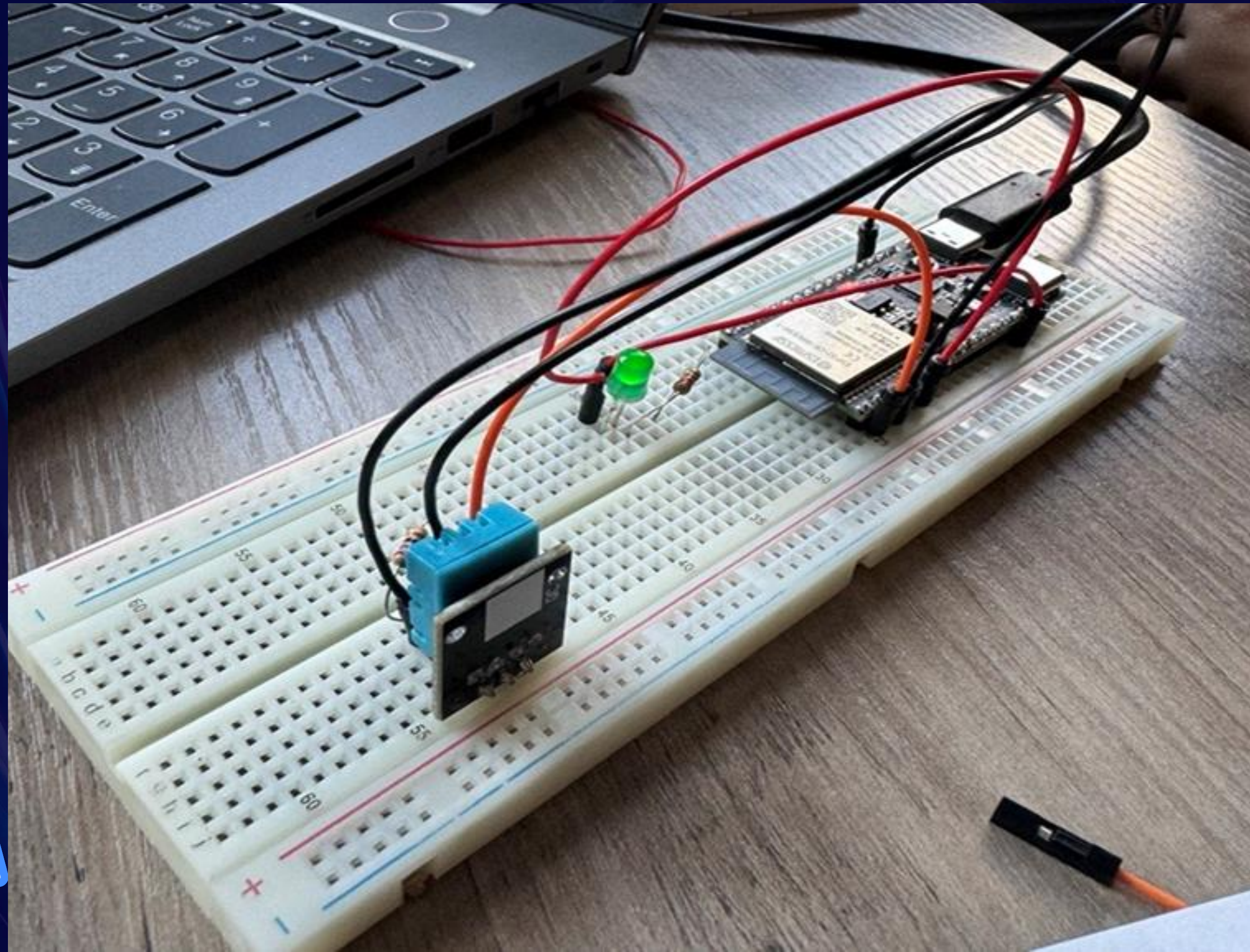



```
26 // Constantes
27 // Définir la broche de la LED
28 const int ledPin = 2; // Utilisez la broche GPIO 2 pour la LED
29
30 void setup() {
31     // Initialiser la broche de la LED comme une sortie
32     pinMode(ledPin, OUTPUT);
33 }
34
35 void loop() {
36     // Allumer la LED
37     digitalWrite(ledPin, HIGH);
38     delay(1000); // Attendre 1 seconde
39
40     // Éteindre la LED
41     digitalWrite(ledPin, LOW);
42     delay(1000); // Attendre 1 seconde
43 }
44
45
```



PARAMÉTRAGE CAPTEUR TEMPÉRATURE



CLOUD

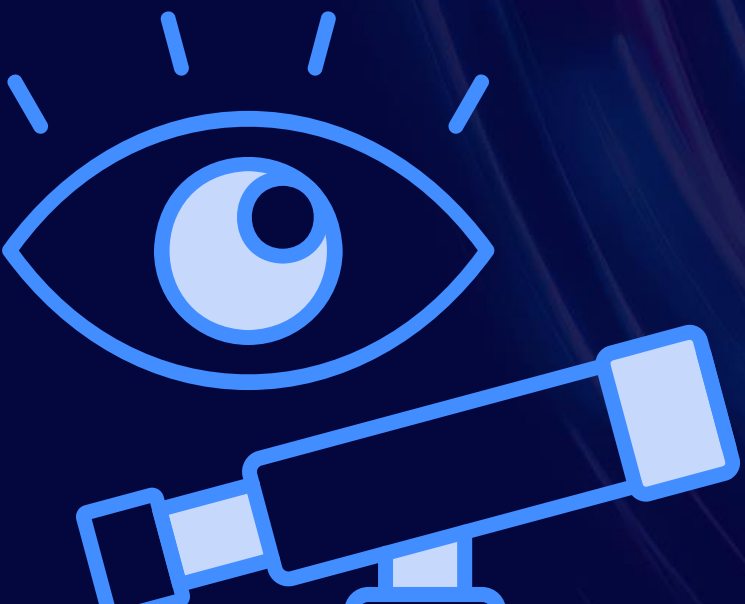
- CRÉER UN COMPTE SUR ARDUINO CLOUD IOT
- AJOUTER UN NOUVEL APPAREIL
- ON SUIT LES INSTRUCTIONS POUR GÉNÉRER UN DEVICE ID ET UNE DEVICE SECRET KEY

Device ID

7d22b87a-49ab-4432-bb0d-9a4fc8e4c3c4

Secret Key

T!3w61mBhTpKm?QTCw#2Tz2oQ



CLOUD

Configure network

×

Enter your network credentials to allow your device to connect to the Cloud.

Wi-Fi Name *

Livebox-D7D0

Password

MRo4MEqDrMt6aaDRm5

👁

Secret Key *

.....

👁

IMPORTANT: Remember to go to the **"Sketch"** tab and upload the sketch to load the credentials on the board.

SAVE

CLOUD

Add variable

Name

Temperature

Sync with other Things

Temperature Sensor (°C) eg. 1 °C

Declaration

CloudTemperatureSensor temperature ;

Variable Permission

Read & Write

Read Only

Variable Update Policy

CANCEL

ADD VARIABLE


CLOUD

Things > Untitled

SetupSketchMetadata


Cloud Variables

ADD


Name ↓	Last Value	Last Update
<input type="checkbox"/> humidity CloudRelativeHumidity humidity;	+	
<input type="checkbox"/> relay bool relay;	+	
<input type="checkbox"/> Temperature CloudTemperatureSensor temperature;	+	


Associated Device


ahmed

ID: 7d22b87a-49ab-4432-bb0d-9a4f...

Type: ESP32S3 Dev Module


Status:  Offline

 Change

 Detach


Network

Enter your network credentials to connect your device.

 Configure

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home



CLOUD

```
#include <DHT.h>

// Définir les pins
#define DHTPIN 4      // Pin du capteur DHT11
#define DHTTYPE DHT11 // Type de capteur DHT
#define LEDPIN 5      // Pin de la LED

DHT dht(DHTPIN, DHTTYPE);

void setup() {
    // Initialiser la communication série
    Serial.begin(115200);

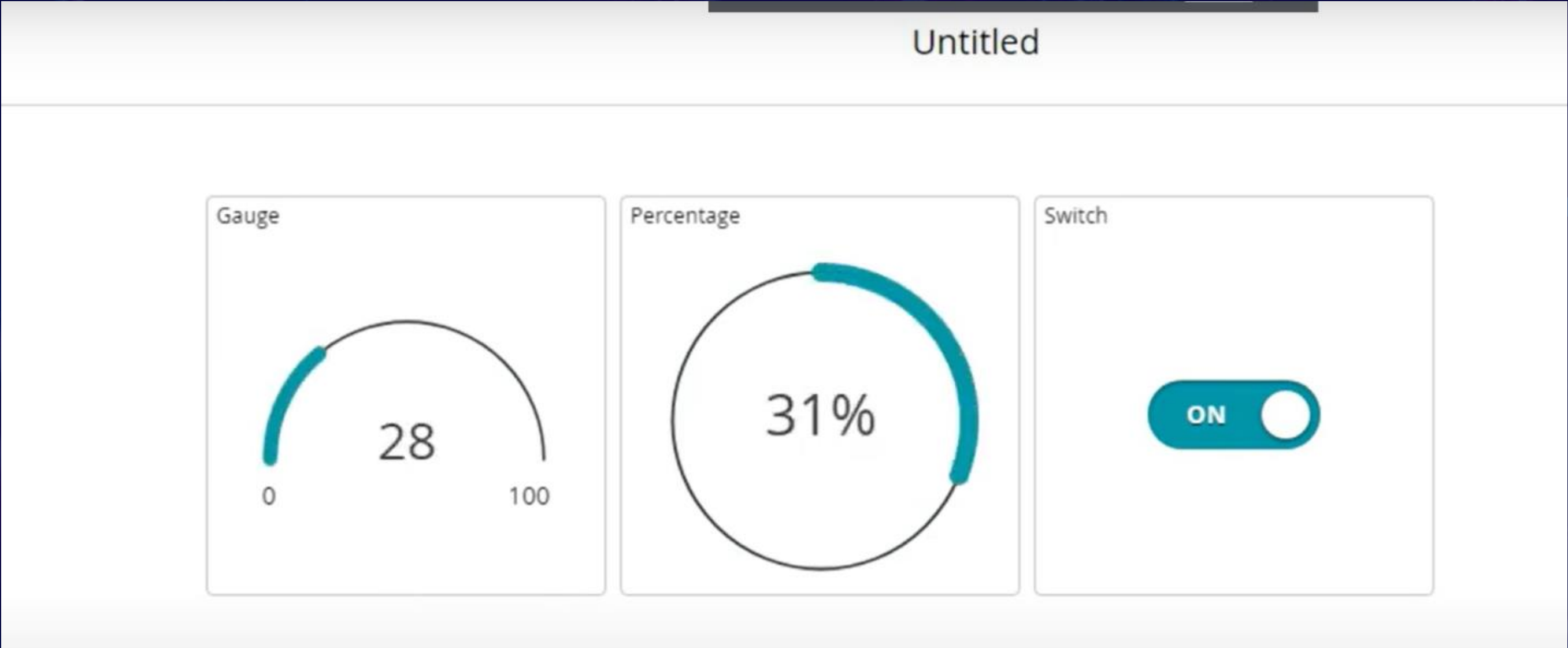
    // Initialiser le DHT11
    dht.begin();

    // Initialiser la pin LED comme sortie
    pinMode(LEDPIN, OUTPUT);

    // Éteindre la LED au démarrage
    digitalWrite(LEDPIN, LOW);
}

void loop() {
    // Lecture de la température et de l'humidité
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    // ...
```

CLOUD



PERSPECTIVES D'AMÉLIORATION

1. COMPATIBILITÉ CLOUD

- ARDUINO CLOUD PAYANT
OU AUTRES PLATEFORMES
IOT ESP32
- COLLECTE ET ANALYSE EN
TEMPS RÉEL

2. AMÉLIORATION DES CAPTEURS

- AJOUTER CAPTEURS
(TEMPÉRATURE, HUMIDITÉ,
PH, LUMINOSITÉ)
- PRÉCISION ACCRUE DE
L'ARROSAGE



PERSPECTIVES D'AMÉLIORATION

3. EXTENSION DES FONCTIONS

- Notifications en temps réel
- Contrôles à distance via une application mobile

4. DÉPLOIEMENT À GRANDE ÉCHELLE

- Tester dans jardins publics et exploitations agricoles
- Évaluer performance et recueillir retours



ARRO IOT

GET IN TOUCH WITH US



+123-456-7890



rrroIot@studioshodwe.com



www.Arroiot.com