

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté Génie Electrique et Informatique
Département d'Informatique
Master SII



PROJET

Module : Intelligence Artificielle des Objets
Thématiques de projet : Serre Intelligente

Membre de groupe :

- LEMDANI Kocela
- SMAILI Maya
- AIT MESSAOUD Lina
- OULD TALEB Dyhia

Année Universitaire : 2023 - 2024

Sommaire

1. Introduction	1
2. Serre intelligente	1
3. Situation actuelle de la serre, défis et opportunités d'amélioration	1
4. Avantages des serres intelligentes	2
5. Architecture technique	2
5.1. Matériel nécessaire	2
5.2. Configuration du logiciel	3
5.3. Fonctionnement du système	3
6. Réalisation technique	4
6.1. Montage du matériel	4
6.2. Programmation de la carte ESP32	6
6.3. Configuration de l'environnement	7
7. Gestion de projet	10
7.1. Métriques logiciels	13
7.2. Méthodologie et outils utilisée pour développer le projet	13
8. Conclusion	13

1. Introduction

L'agriculture sous serre est de plus en plus populaire pour faire pousser des fruits et des légumes tout au long de l'année dans des conditions contrôlées. Cependant, la surveillance des conditions environnementales telles que la température, l'humidité du sol et la luminosité peut être un défi important pour les agriculteurs. Pour relever ce défi, les technologies de l'Internet des objets (IoT) peuvent être utilisées pour développer des applications de serre intelligentes pour la surveillance et le contrôle de ses conditions.

Pour assurer la croissance des plantes et optimiser la production agricole, la serre intelligente est munie principalement de capteurs (de température, d'humidité, de lumière..) et d'actionneur (pompe à eau, moteur..).

Pour la réalisation de notre projet, nous avons utilisé :

- Un IDE Arduino.
- Une carte ESP32.
- Les capteurs nécessaires pour concevoir une serre intelligente.
- Node-RED pour gérer les flux de données et créer un tableau de bord pour visualiser les informations collectées.
- InfluxDB qui est une base de données qui nous permet de stocker et d'analyser les données collectées.

L'objectif de cette application de serre intelligente est de fournir une solution pour la surveillance et le contrôle des conditions environnementales dans les serres, ce qui permettra d'améliorer la qualité et la quantité de la production des produits agricoles et réduire les coûts liés à la main-d'œuvre. Le reste de ce rapport décrit en détail la mise en œuvre de cette application, les technologies utilisées et les avantages potentiels pour les producteurs agricoles.

2. Serre intelligente

Une serre intelligente est une serre qui utilise une combinaison des technologies avancées pour surveiller et contrôler les conditions environnementales à l'intérieur de la serre. Les serres intelligentes peuvent être contrôlées à distance à l'aide d'une application ou d'un ordinateur n'importe où dans le monde, ce qui permet une plus grande flexibilité et une meilleure surveillance des conditions environnementales. Elles peuvent également être programmées pour s'adapter aux différentes conditions de croissance pour chaque type de plante, ce qui augmentera le taux de réussite de la culture.

3. Situation actuelle de la serre, défis et opportunités d'amélioration

Les serres traditionnelles nécessitent une surveillance constante pour maintenir les conditions optimales pour la croissance des plantes. Ce qui représente un défi pour les agriculteurs. Le projet de serre intelligente vise à fournir une solution plus accessible et plus facile à utiliser pour les agriculteurs.

4. Avantages des serres intelligentes

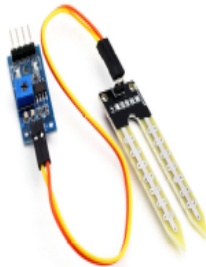
- Protégez les plantes contre les températures extrêmes en fournissant un environnement de croissance stable pour les cultures agricoles ce qui permet une production plus élevée grâce à la possibilité de contrôler les conditions climatiques de la culture et une favorisation de la production pendant toutes les saisons.
- Protégez les plantes contre les maladies. Le système de contrôle automatique permet de surveiller en temps réel tous les changements environnementaux.
- Surveiller à distance la serre et rester au courant des conditions climatiques qui s'y trouvent. Le système de contrôle automatique permet une surveillance continue à distance en fournissant les moyens permettant ainsi d'agir rapidement et en temps réel.

5. Architecture technique

5.1. Matériel nécessaire



Carte ESP32
avec 38 pins



Capteur
d'humidité de sol



Capteur de
température
et d'humidité
DHT11



Capteur de
Lumière
(photorésistance)



Les LED



Les résistances



Les câbles



Breadbord (lab
d'essai)



Un pc



Écran LCD I2C



Clavier



Servomoteur



Pompe à eau



Plaque en bois



Plaques de verre



plante

5.2. Configuration du logiciel

- Utiliser l'IDE Arduino afin de programmer la carte ESP32 pour qu'elle puisse collecter les données des capteurs et les envoyer à Node-RED.
- Utiliser Node-RED pour créer un flux qui traite les données des capteurs et commande la pompe à eau en fonction des seuils définis pour l'humidité, commande aussi l'ouverture des fenêtres d'aération selon les données de prévision météorologique en temps réel des plateformes externes comme l'API Open WeatherMap disponible sous node-red.
- Configurer Node-RED pour stocker les données des capteurs dans la base de données influxDB.
- Utiliser Node-RED pour créer un Tableau de bord Node-RED (dashboard) qui affiche les données des capteurs en temps réel, ainsi que l'état de la pompe, de la fenêtre et la météo extérieure.

5.3. Fonctionnement du système

- La serre est divisée en quatre zones. Dans chaque zone, il ne peut y avoir qu'un seul type de plante. Les informations sur les quatre types de plantes à l'intérieur de la serre sont stockées dans notre base de données sous influxDB.
- Les capteurs utilisés (détecteur de température et d'humidité de l'air (dht11), détecteur d'humidité de sol, détecteur de luminosité (LDR)) collectent les données de la serre.
- Après avoir téléversé le code de arduino dans la carte ESP32, cette dernière traite les données des capteurs et les envoie à Node-RED.
- Node-RED utilise les données des capteurs pour commander les tâches des actionneurs :
 - système d'irrigation dans le cas de manque d'eau dans le sol (cette tâche est effectuée par une pompe à eau et une pile de 9 volts).
 - Ouverture ou fermeture du dôme de protection (dans notre cas ouverture de la fenêtre), en fonction des conditions météorologiques du moment.
 - Thermostat(dans notre cas LED Rouge) en fonction des besoins des plantes.
 - La lumière artificielle(Dans notre cas LEDs Blanches) en fonction des besoins des plantes.
- Notre dashboard affiche les données des capteurs en temps réel, permettant ainsi à l'utilisateur de surveiller l'environnement de la serre et d'ajuster les paramètres au besoin.

6. Réalisation technique

6.1. Montage du matériel

→ construction de la serre

Plaque en bois : 50cm*50cm

La surface de la base en verre : 25cm*35cm

La hauteur : 30 cm

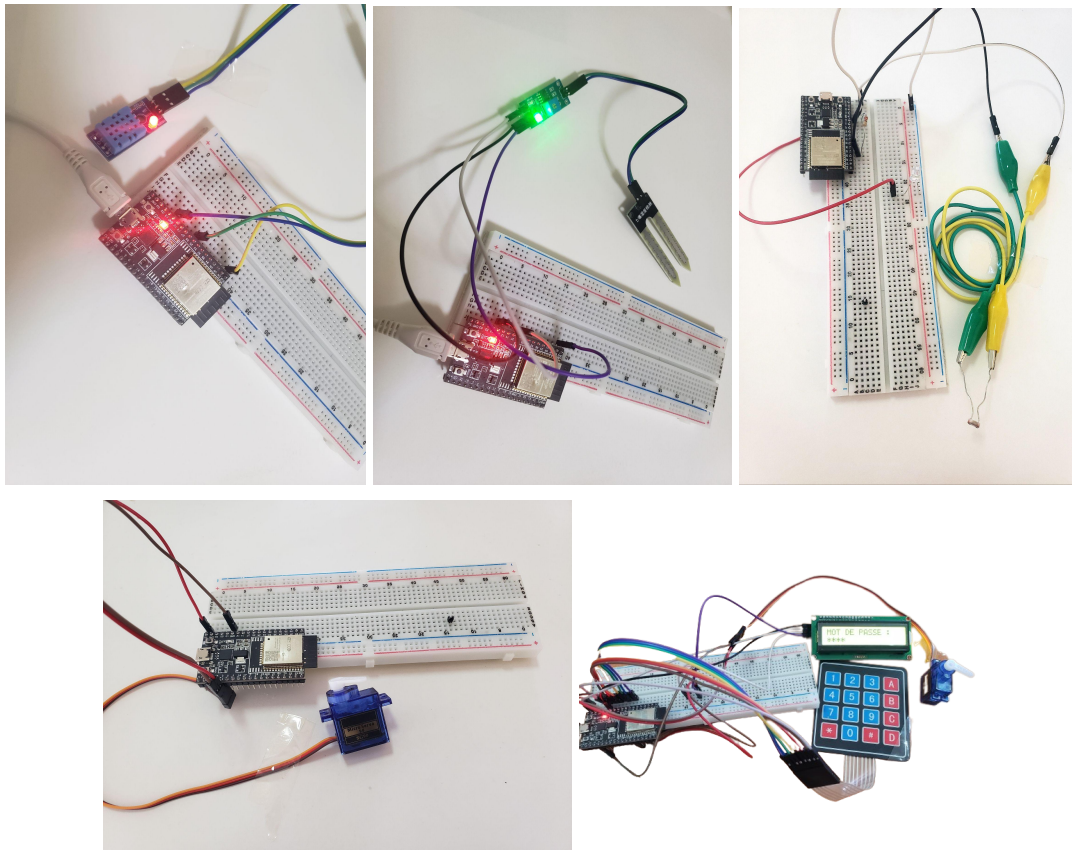


→ Connexion des composants à l'esp32

Connexion de tous les capteurs, de la pompe à eau, de l'écran LCD et du clavier à la carte ESP32 à l'aide de breadboard et des fils de branchement appropriés.

Système	Pin système	Pin esp32/système
LDR (photorésistance)	Patte 1	3.3V
	Patte 2	Brancher à une patte de la résistance
Résistance	Patte 1	brancher à LDR
	Patte 2	GND
LED	VCC	4
	GND	GND
Micro Servo de la fenêtre	VCC	5V
	GND	GND
	DATA	15
Micro Servo de la porte	VCC	5V
	GND	GND
	DATA	2
LCD I2C	VCC	5V
	GND	GND
	SCL	22
	SDA	21
Capteur d'humidité de sol	VCC	3.3V
	GND	GND
	AO	VP
Clavier 4x4	1	32
	2	33
	3	25

Clavier 4x4 (suite)	4	26
	5	27
	6	14
	7	12
	8	13
DHT11	VCC	5V
	GND	GND
	DATA	18
Pompe	+	ON (Relais)
	-	Fil noir de la pile(-)
Pile	+	ON (Relais)
	-	File noir de la pompe(-)
Relais	+	3.3V
	-	GND
	S	16
	ON	Au fil rouge de la pile (+)
	ON	Au fil rouge de la pompe (+)



6.2. Programmation de la carte ESP32

Après l'installation de l'IDE Arduino, nous avons créé un nouveau projet et ouvert un nouveau fichier nommé *serre intelligente* pour notre programme. Ensuite, on a programmé la carte ESP32 pour qu'elle collecte les données des capteurs et les envoie à Node-RED en utilisant les bibliothèques appropriées tel que : DHT.h, ESP32 Servo.h, pour les capteurs. Enfin, on a testé si le code est fonctionnel et qu'il est correctement téléversé sur la carte ESP32.

les bibliothèques qu'on a utilisé :

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <ESP32Servo.h>
```

un fragment de code de notre fonction setup

```
void setup() {
  pinMode(LIGHTPIN, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(pumpPin, OUTPUT);
  pinMode(waterLevel, INPUT);
  myservoF.attach(15); // attaches the servo on pin 15 to the servo object
  Serial.begin(9600); // Initialisation du port série à 9600 bauds
  dht.begin(); // Initialisation du capteur DHT
```

un fragment de code de notre fonction loop

```
void loop() {

    float temperature = dht.readTemperature(); // Lecture de la température
    float humidity = dht.readHumidity(); // Lecture de l'humidité
    int lightValue = analogRead(LIGHTPIN); // Lecture de la valeur du capteur de lumière
    int sensor_Sol_Value = analogRead(sensorPin Sol); // Lit la valeur du capteur

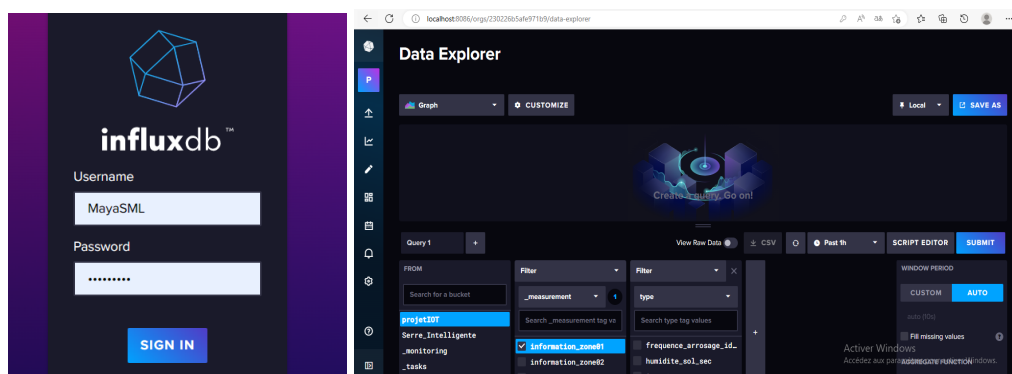
    //lecture de l'entrée du capteur de niveau d'eau
    waterLevel = analogRead(A0);
    Serial.println(waterLevel);
    //si le niveau d'eau est bas, activer la pompe
    if (waterLevel >3000) {
        digitalWrite(pumpPin, HIGH);

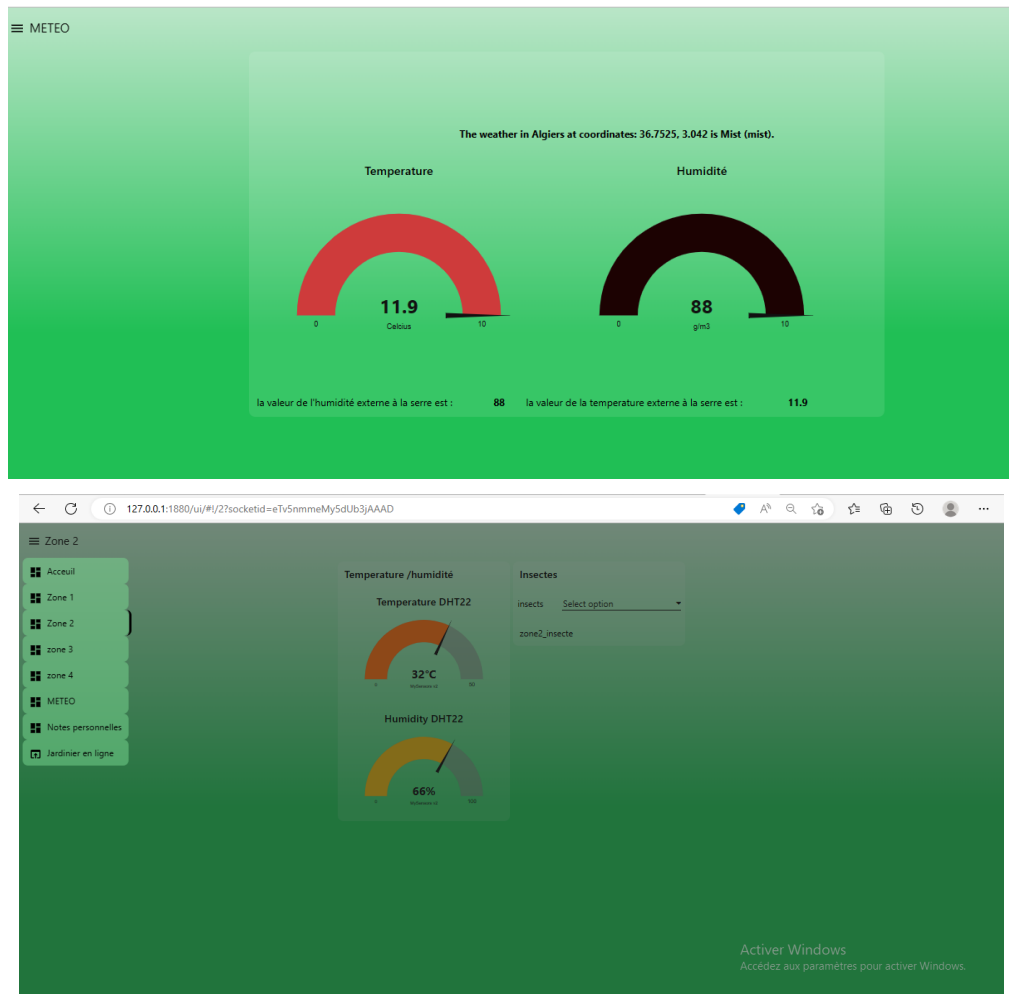
    } else {
        digitalWrite(pumpPin, LOW);
    }

    // Envoi des valeurs sur le port série
    Serial.print(" Temperature: ");
    Serial.print(temperature);
    Serial.print(" C Humidite: ");
    Serial.print(humidity);
    Serial.print(" % Valeur de lumiere: ");
    Serial.print(lightValue);
    Serial.print(" Valeur de l'humidite de sol : ");
    Serial.println(sensor_Sol_Value); // Affiche la valeur lue sur le port série
    if (lightValue < 100) { // si la lumière est faible
        digitalWrite(ledPin, HIGH); // allume la LED
    } else { // sinon
        digitalWrite(ledPin, LOW); // éteint la LED
    }
}
```

6.3. Configuration de l'environnement

- On a installé node-RED, arduino, InfluxDB et les paquets nécessaires sur notre ordinateur local pour configurer l'environnement de l'application.





- On a utilisé un Contrôle automatique qui signifie Configurer des règles dans Node-RED pour contrôler automatiquement les équipements de la serre, tel que l'ouverture de la fenêtre en fonction des conditions météorologiques externes, en récupérant de node red la valeur de weathermap en utilisant le node Openweathermap et le node fonction ainsi qu'un serial out connecté à notre port.

voici le code de notre fonction

```
var tempc = msg.payload.tempc;
if (tempc < 20) {
    var command = "MOVE_SERVO_180"; // Replace this with the appropriate command for your setup
    msg.payload = command; // Set the payload to the command
    return msg; // Pass the message to the output
} else { // If temperature is greater than or equal to 10, pass the message to the output without sending a command to move the servomotor
    return msg;}
```

Et sur arduino

```
#include <ESP32Servo.h>
Servo myservo; // create servo object to control a servo

void setup() {
    Serial.begin(9600); // initialize serial communication at 9600 bits per second
    myservo.attach(15); // attaches the servo on pin 15 to the servo object
}

void loop() {
    if (Serial.available() > 0) {
        String input = Serial.readStringUntil('\n'); // read the incoming serial command
```

```

if (input == "MOVE_SERVO_180") {
  myservo.write(180); // move the servo to the 180 degrees position
  // wait for 1 second to allow the servo to move
}
else{
  myservo.write(0); // move the servo back to the 0 degrees position
}
}
}

```

- Accès intelligent à l'aide des capteurs, sécurisé par un code d'entrée affiché sur un écran LCD I2C et automatisé par les actions programmées tel que ouverture de la porte si le code est correct, éteindre les leds si l'on a une lumière suffisante et les allumées dans le cas contraire.

7. Gestion de projet

Une gestion de projet efficace pour notre projet d'application de serre intelligente nécessite une planification claire, une affectation efficace des tâches, des rôles clairement définis pour les membres de l'équipe, un suivi et un contrôle rigoureux, ainsi qu'un processus de test et de déploiement bien planifié.

- La première partie de ce travail a fait l'objet d'achat de matériels, les capteurs, les leds, les résistances, les fils, la pompe, le relais, les servomoteurs ainsi que le vers et le bois. Et l'installation de logiciels nécessaires.
- La deuxième partie, c'est le branchement des capteurs à l'esp 32 et l'écriture des codes arduino et teste des résultats de chacun à part. Cette partie était un succès.

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for opening files, saving, compiling, and uploading. The main editor window displays the code for 'DHT11.ino'. The code includes a setup function that initializes the serial port at 9600 baud and the DHT11 sensor. The loop function reads humidity and temperature data from the sensor, prints them to the serial monitor, and delays for 10000 milliseconds. The Serial Monitor window at the bottom shows the output of the code, displaying humidity and temperature readings in both percentage and degrees Celsius.

```

DHT11.ino
7
8 void setup() {
9   Serial.begin(9600);
10  dht.begin();
11 }
12
13 void loop() {
14   delay(10000);
15
16   float humidity = dht.readHumidity();
17   float temperature = dht.readTemperature();
18
19   Serial.print("Humidite : ");
20   Serial.print(humidity);
21   Serial.print("% - Temperature : ");
22   Serial.print(temperature);
23   Serial.println(" C");
24 }

```

Output Serial Monitor

Message (Enter to send message to 'ESP32 Dev Module' on 'COM7')

```

Humidite : 51.00% - Temperature : 25.00 C
Humidite : 52.10% - Temperature : 25.50 C
Humidite : 52.20% - Temperature : 25.50 C
Humidite : 52.40% - Temperature : 25.30 C
Humidite : 65.90% - Temperature : 25.90 C
Humidite : 72.20% - Temperature : 27.40 C
Humidite : 76.20% - Temperature : 28.90 C
Humidite : 78.10% - Temperature : 30.20 C
Humidite : 78.90% - Temperature : 31.10 C

```

- La troisième partie a fait l'objectif de l'installation des modules sur node red et on a commencé le travail sur les flows node red et sur la base de données influxdb.

The screenshot shows the Data Explorer interface. It displays a table with sensor data. The table has columns for 'table', 'measurement', 'field', 'value', 'time', 'type', and 'value'. The data is organized into rows, each representing a different sensor measurement. The table is titled 'Data Explorer' and has a 'CUSTOMIZE' button in the top right corner.

table	measurement	field	value	time	type	value
information_sensors	information_sensors	value	0	2023-02-23T03:10:00.000Z	frequency_average_label_jour	1
information_sensors	information_sensors	value	0	2023-02-23T03:10:00.000Z	humidity_min_max	30
information_sensors	information_sensors	value	1	2023-02-23T03:10:00.000Z	insects	Les pucerons
information_sensors	information_sensors	value	2	2023-02-23T03:10:00.000Z	insects	Les acariens
information_sensors	information_sensors	value	2	2023-02-23T03:10:00.000Z	insects	Les thrips (thysanoptera)
information_sensors	information_sensors	value	0	2023-02-23T03:10:00.000Z	humidity_label_k_j	14
information_sensors	information_sensors	value	0	2023-02-23T03:10:00.000Z	mm	epinard
information_sensors	information_sensors	value	0	2023-02-23T03:10:00.000Z	temperature_label	15

- Dans quatrième partie, on a regroupé tous les codes dans un seul code arduino puis on a débuté avec le dashboard node red et la récupération des données des capteurs.

The screenshot shows the Arduino IDE interface. The top menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The 'Tools' dropdown menu is open, showing 'ESP32 Dev Module'. The main editor displays the code for 'serre_intelligente.ino'.

```

1 #include <DHT.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <Keypad.h>
5 #include <ESP32Servo.h>
6
7 #define DHTPIN 18 // Broche à laquelle est connecté le capteur DHT11
8 #define DHTTYPE DHT11 // Type de capteur DHT
9 #define LIGHTPIN 34 // Broche à laquelle est connecté le capteur de lumière
10 #define ledPin 4 // broche de la LED

```

Below the code editor, the 'Serial Monitor' is open, displaying a series of sensor readings. A yellow banner at the top of the Serial Monitor says 'Not connected. Select a board and a port to connect automatically.'

```

Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 4 Valeur de l'humidite de sol : 3840
3837
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 1 Valeur de l'humidite de sol : 3835
3843
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 0 Valeur de l'humidite de sol : 3839
3838
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 18 Valeur de l'humidite de sol : 3834
3830
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 57 Valeur de l'humidite de sol : 3829
3827
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 64 Valeur de l'humidite de sol : 3826
3834
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 39 Valeur de l'humidite de sol : 3837
3835
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 15 Valeur de l'humidite de sol : 3822
3835
Temperature: 23.90 C Humidite: 62.00 % Valeur de lumiere: 0 Valeur de l'humidite de sol : 3836
3835

```

- La dernière étape, ou on a effectué la réalisation physique de notre maquette Serre intelligente.



7.1. Métriques logiciels

- Le taux d'humidité dans notre serre assure que les plantes reçoivent les niveaux d'humidité appropriés pour leur croissance. Dans notre cas, le capteur d'humidité de sol actionne le démarrage d'une pompe à eau si le niveau de l'humidité est inférieur aux conditions idéales pour la plante.
- La température à l'intérieur de la serre est mesurée pour assurer qu'elle est maintenue dans les limites acceptable pour les plantes.
- notre système répond au temps réel par les actions souhaitées sous arduino.

7.2. Méthodologie et outils utilisée pour développer le projet

Pour développer le projet de serre intelligente, nous avons utilisé des capteurs de température, d'humidité et de lumière. Ces capteurs sont connectés à une carte esp32 pour collecter des données en temps réel sur les conditions de la serre. Les données collectées sont ensuite transmises à Node-RED pour le traitement et le contrôle des équipements de la serre. Ce dernier déclenche des actions en conséquence, telles que l'allumage de led, commandement de la pompe, l'allumage des servo moteur

8. Conclusion

La réalisation d'un projet de serre intelligente à l'aide d'un IDE Arduino, d'une carte ESP32 et de capteurs nécessaires est une tâche complexe mais réalisable avec les bonnes compétences et les outils appropriés. L'utilisation de Node-RED pour la gestion des flux de données et la création d'un tableau de bord pour visualiser les informations collectées permet une surveillance et une gestion efficaces des conditions environnementales dans la serre. L'intégration d'une base de données InfluxDB permet également de stocker et d'analyser les données collectées pour prendre des décisions éclairées sur l'optimisation des conditions de croissance des plantes.

Cependant, pour la réussite du projet, il est important de suivre une méthodologie de gestion de projet adaptée, d'attribuer des tâches spécifiques à chaque membre de l'équipe, de définir les rôles de chaque membre de l'équipe et de suivre un calendrier réaliste pour respecter les délais et les budgets. En fin de compte, la serre intelligente peut offrir de nombreux avantages, notamment une production de plantes plus efficace et durable, une économie d'eau et une réduction des coûts énergétiques.

En conclusion, ce projet nous a permis d'approfondir et d'enrichir nos connaissances théoriques. Il nous a aussi permis d'avoir une expérience au niveau de la réalisation pratique. Nous avons eu l'occasion d'étudier, de concevoir et d'utiliser une diversité de matériels et logiciels.