|  |
| --- |
| Gestion d'Arrosage Automatique |

Date de création – le 2 mai 2025

Date de fin – le 7 mai 2025

Réalisateur du projet - Sulejmani Mefail-CIN4B-ETML.

Chef de projet Monsieur Costa Lopes.

Expert 1 : Monsieur Bertino

Expert 2 : Monsieur Albert

Table des matières

[1 Analyse préliminaire 4](#_Toc197506364)

[1.1 Introduction 4](#_Toc197506365)

[*1.2* Objectifs 4](#_Toc197506366)

[*1.3* Planification initiale 5](#_Toc197506367)

[1.4 Méthodologie du projet 7](#_Toc197506368)

[2 Analyse / Conception 8](#_Toc197506369)

[*2.1* Concept 8](#_Toc197506370)

[2.2 Stratégie de test 8](#_Toc197506371)

[2.3 Risques techniques 8](#_Toc197506372)

[2.4 Matériel et logiciels à disposition 8](#_Toc197506373)

[*2.5* Risques techniques 8](#_Toc197506374)

[2.6 ESP32-WROOM 9](#_Toc197506375)

[2.7 Ecran OLED 9](#_Toc197506376)

[2.8 Json 9](#_Toc197506377)

[2.9 CRUD 9](#_Toc197506378)

[2.10 Arduino IDE 9](#_Toc197506379)

[2.11 GIT 9](#_Toc197506380)

[2.12 Maquette 9](#_Toc197506381)

[*2.13* Dossier de conception 9](#_Toc197506382)

[3 Réalisation 9](#_Toc197506383)

[*3.1* Dossier de réalisation 9](#_Toc197506384)

[3.2 Liaison du matériel 10](#_Toc197506385)

[3.3 Connexion sur le Wifi IOT 10](#_Toc197506386)

[3.4 Création serveur http 10](#_Toc197506387)

[3.5 Mesure de l’humidité 10](#_Toc197506388)

[3.6 Interface web 10](#_Toc197506389)

[3.7 Stockage des données 10](#_Toc197506390)

[*3.8* Description des tests effectués 10](#_Toc197506391)

[*3.9* Erreurs restantes 10](#_Toc197506392)

[*3.10* Liste des documents fournis 10](#_Toc197506393)

[4 Conclusions 11](#_Toc197506394)

[4.1 Objectifs attaints 12](#_Toc197506395)

[4.2 Difficultés rencontres 12](#_Toc197506396)

[4.3 Suites du projet 12](#_Toc197506397)

[5 Annexes 12](#_Toc197506398)

[*5.1* Résumé du rapport du TPI / version succincte de la documentation 12](#_Toc197506399)

[*5.2* Sources – Bibliographie 12](#_Toc197506400)

[*5.3* Journal de travail 12](#_Toc197506401)

[*5.4* Manuel d'Installation 12](#_Toc197506402)

[*5.5* Manuel d'Utilisation 12](#_Toc197506403)

[*5.6* Archives du projet 12](#_Toc197506404)

**Tableau des légendes**

# Analyse préliminaire

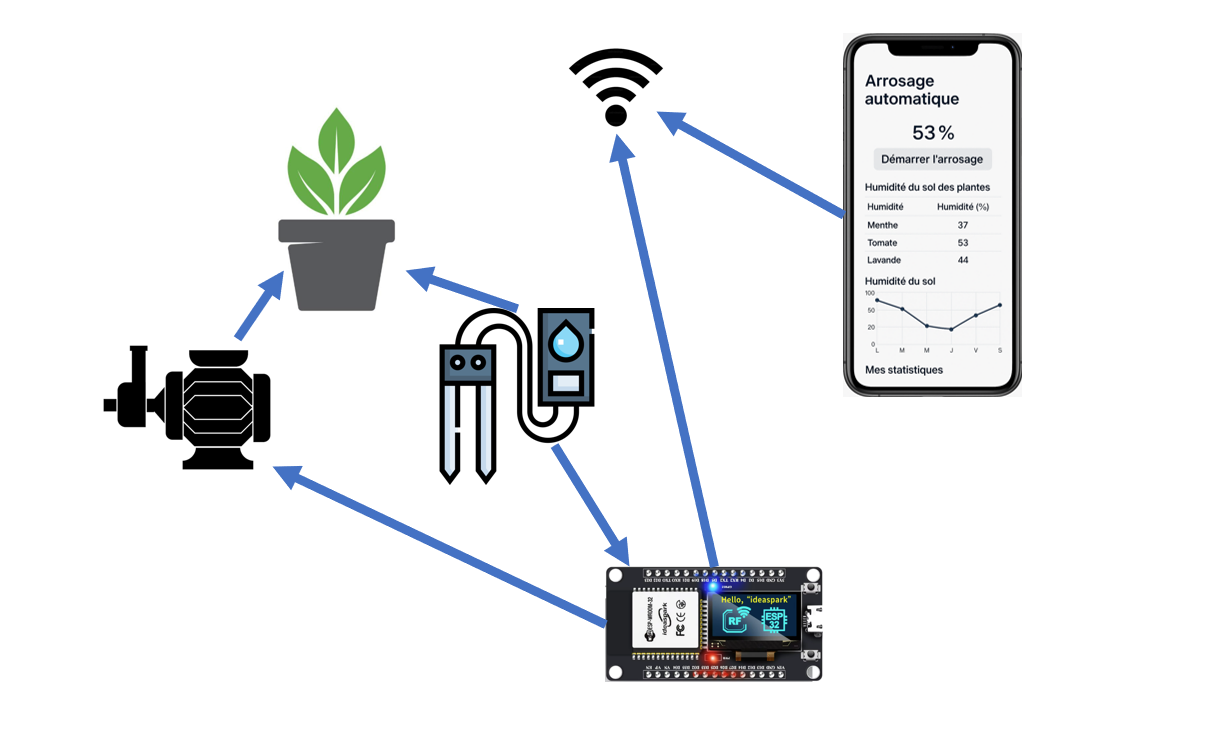
La recherche préliminaire établit les bases sur lesquelles le projet est fondé et ses objectifs principaux.

L'objectif de ce projet est de créer un système d’arrosage automatique ou manuelle gère via une interface Web.

## Introduction

Développer un système d’arrosage automatique basé sur un ESP32, un capteur d’humidité du sol et un relais, avec une plateforme de gestion permettant à un utilisateur non technique de configurer les paramètres d’arrosage via une interface web.

Ce projet a pour objectif la prise en main d’un ESP32-WROOM équipé d’un écran OLED intégré. L’ESP32 se connectera au réseau Wi-Fi IoT afin d’héberger un serveur web. Un capteur d’humidité est installé dans une plante pour mesurer son taux d’humidité. Les données relevées seront envoyées à l’ESP32, qui pourra alors déclencher automatiquement une pompe pour arroser la plante si nécessaire. Il sera également possible de lancer l’arrosage manuellement via l’interface web, qui servira de système de contrôle. Enfin, l’historique des arrosages sera enregistré dans la mémoire de l’ESP32 pour assurer un suivi complet.

Ici, on peut voir un graphique simple montrant le résultat que je dois obtenir à la fin de ce projet**

## Objectifs

La grille d’évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, …).

En plus de cela, le travail sera évalué sur les 14 points spécifiques suivants

1. Connecter automatiquement l’ESP32 au réseau Wi-Fi local (réseau IOT dédié).
2. Développer une interface web locale pour la consultation et la configuration du système d’arrosage.
3. Lire la valeur d’humidité du sol depuis un capteur analogique, avec affichage en valeur brute ou en pourcentage.
4. Implémenter un contrôle automatique de l’arrosage basé sur un seuil d’humidité configurable.
5. Déclencher l’arrosage si la valeur mesurée est inférieure au seuil défini par l’utilisateur.
6. Permettre l’activation manuelle de l’arrosage via un bouton dans l’interface web.
7. Permettre le réglage de la durée d’arrosage (par exemple de 5 à 60 secondes).
8. Sauvegarder les paramètres de configuration dans une mémoire persistante (EEPROM émulée ou fichier JSON via LittleFS/SPIFFS).
9. Enregistrer l’historique des arrosages avec les dates, durées et seuils utilisés.
10. Mettre en place un serveur HTTP sur l’ESP32 répondant aux requêtes suivantes :
11. GET /data : renvoie les données actuelles du système (état, humidité, paramètres).
12. Implémenter la commande POST /manual pour activer manuellement l’arrosage via l’interface web.
13. Implémenter la commande POST /add pour ajouter une plante avec ses paramètres (nom, type, seuil).
14. Héberger les fichiers HTML et CSS de l’interface web dans le système de fichiers embarqué (SPIFFS ou LittleFS).

## Planification initiale

La planification initiale a pour but d'estimer le temps à consacrer à chaque phase du projet afin d’organiser efficacement le travail. Elle permet également, une fois le projet terminé, de comparer les estimations avec le temps réellement utilisé et d’analyser les écarts. Cette réflexion contribue à améliorer la gestion du temps et la planification pour de futurs projets.

Le temps prévu pour la réalisation de ce projet est de 89 heures et 55 minutes. Le projet débute le 2 mai 2025 et se termine le 2 juin 2025. Les tâches prévues dans cette planification sont basées sur le cahier des charges fourni par le chef de projet.

Voici les tâches génériques qui englobent les étapes fondamentales du projet, notamment les quatre plus importantes :

**Analyse** : cette tâche repose sur une phase de recherche, de réflexion, de vision du produit, d’imagination, de premiers tests et de compréhension du fonctionnement global.

**Implémentation** : cette tâche consiste à développer les fonctionnalités demandées dans le cahier des charges.

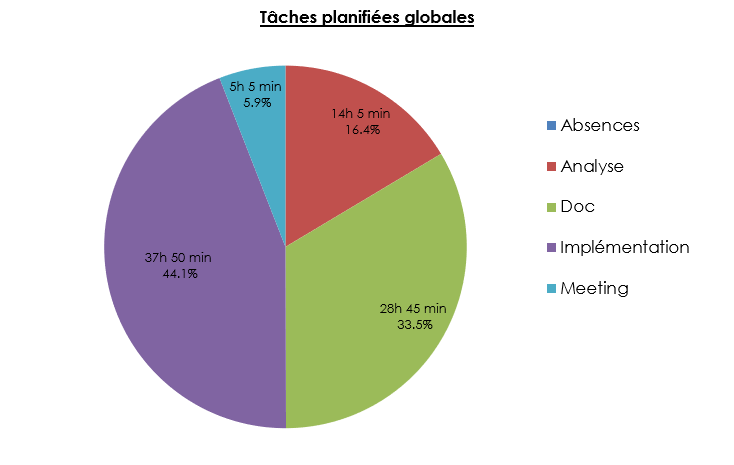
**Doc** : cette tâche comprend la rédaction du rapport, du journal de travail et des annexes.

**Meetings** : cette tâche concerne les échanges avec les experts et le chef de projet pour faire le point sur l’avancement et valider les choix techniques.

|  |
| --- |
| Absences - Imprévus |
| Analyse ‑ Cahier des charges  Analyse - Mise en place de la méthode de la gestion du projet  Analyse - Planification initiale  Analyse - Recherche  Analyse ‑ Composants  Analyse ‑ Tests |
| Implémentation - Commentaires code  Implémentation - Connexion réseau IOT  Implémentation - ESP32  Implémentation - GIT  Implémentation - Interface web  Implémentation - Mise en place des branches ESP32  Implémentation - Stockage |
| Doc - Journal de travail  Doc - Rapport |
| Meeting - Chef du projet  Meeting - Expert 1  Meeting - Expert 2 |

La différence entre le pourcentage de temps de planification que j’avais prévu et celui indiqué dans le CDC s’explique par mon estimation personnelle ; en raison de mes difficultés en français, je compte donc consacrer davantage de temps à la rédaction du rapport, comme l’illustre l’image ci‑dessus.





## Méthodologie du projet

Pour le déroulement de ce projet sur le ESP32, j’ai choisi d’utiliser la méthode de 6 pas : Informer, Planifier, Décider, Réaliser, Contrôler, Évaluer.

La principale raison de ce choix pour le projet est que je suis seul à le réaliser et aussi une meilleure réflexion sur chaque étape de mon point de vue.

La méthode des 6 étapes me permettra de planifier étape par étape mon projet de construction. Elle m’aidera à rester concentré et à ne rien oublier. Ce travail de fond m'aidera à éviter certains soucis et erreurs dans mon projet de construction, dans mon cas, me permet aussi de décomposer mes étapes hardware avec l’ESP32 et les components pour permettre l’arrosage des plantes, développement web, gestion du réseau et stockage etc.

Les 6 étapes organisées dans mon projet :

* **Informer** : Je me suis documenté sur l’ensemble des éléments nécessaires au bon déroulement du projet, notamment sur le matériel (ESP32, capteurs d’humidité, relais), les contraintes du cahier des charges, les outils de développement (Arduino IDE, Git, SQLite, JSON) et les fonctionnalités attendues comme le CRUD et l’interface web statique. Cette phase m’a permis de bien cerner les objectifs et le contexte global du projet.
* **Planifier** : J’ai établi une planification structurée, en répartissant les tâches selon quatre axes : analyse, implémentation, documentation et finalisation. Cette organisation m’a permis de mieux gérer mon temps et de suivre l’avancement du projet de manière efficace.
* **Décider** : J’ai fait des choix techniques en fonction des besoins et des contraintes : utilisation d’un fichier JSON pour le stockage des données, implémentation d’un serveur HTTP pour gérer la communication entre l’ESP32 et l’interface, conception de l’interface en HTML/CSS, et adoption de la méthode des 6 étapes pour structurer l’ensemble du projet.
* **Réaliser** : J’ai développé le code progressivement, en finalisant chaque fonctionnalité indépendamment (lecture des capteurs, contrôle du relais, gestion des données, etc.) avant de les intégrer dans l’ensemble du système. Cela m’a permis de tester chaque partie de manière isolée avant de les réunir.
* **Contrôler** : J’ai effectué des tests manuels pour vérifier le bon fonctionnement de chaque fonctionnalité (lecture de l’humidité, déclenchement automatique ou manuel de l’arrosage, ajout de plantes, etc.). J’ai également testé tous les composants électroniques pour m’assurer de leur fiabilité.
* **Évaluer** : En fin de projet, j’ai comparé les résultats obtenus avec les objectifs fixés initialement. Grâce aux tests réalisés dès les premières étapes, j’ai pu identifier les points forts et les éventuelles améliorations à envisager pour une future version du système.

Pourquoi je n’ai pas choisi les autres méthodes ?

Méthode Agile : adaptée aux projets évolutifs avec des changements fréquents, ce qui n’est pas le cas ici car mon projet a un objectif bien défini aux départs.

Cycle en V : plus utilisé pour des projets complexes nécessitant de longues phases de validation et de tests stricts.

# Analyse / Conception

## Concept

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*
* *…*

## Stratégie de test

## Risques techniques

## Matériel et logiciels à disposition

* 1 microcontroleur ESP32(ex. ESP32-WROOM-32)
* Ecran OLED 0,96 128×64 px (driver SSD1306, bus I²C)
* Capteur d'humidite du sol hw-103 v0.1
* Relais 5V SRD-05VDC-SL-C pour contrôler une pompe ou électrovanne
* 1 ordinateur type PC Windows
* Câble Dupont
* Mini planche 45,72x35,56mm.
* Cable USB-A à micro USB
* Environnement de développement Arduino IDE pour programmer l'ESP32 en C
* SQLite (ou fichier JSON pour simplicite) pour la plateforme de gestion
* Logiciel de contrôle de version Git Desktop
* Microsoft Office pour la documentation
* Driver pour la reconnaissance de l’ESP32 CH340
* Connexion Wifi local (IOT)
* Maquette Figma

## ESP32-WROOM

## Ecran OLED

## Module de capteur d’humidité du sol

## SPIFFS

## JSON

## CRUD

## Arduino IDE

Explique pourquoi ce choix

## GIT

## Maquette

## Dossier de conception

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

## Diagrammes fonctionnels

# Réalisation

## Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

### Liaison du matériel

### Connexion sur le Wifi IOT

### Mise en œuvre du serveur HTTP

### Mesure de l’humidité

### Interface web

### Stockage des données

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

## Bilan des fonctionnalités

## Objectifs attaints

## Difficultés rencontres

## Bilan personnelle

## Résultat de stratégie de test

## Suites du projet

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

### Installations

### Tutoriels

System d’arrosage Automatique:[ESP32 - Automatic Irrigation System | ESP32 Tutorial](https://esp32io.com/tutorials/esp32-automatic-irrigation-system)

### Librairies

### Achats

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Lexique

## Utilisation de l’intelligence artificielle

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*