

Rapport de Projet d'évaluation Soft Embarqué

Projet : The Smart Farm



Réalisé par : ZAIM OUSSAMA

ECHCHIDMI MOHAMED

Supervisé par :

M. MANSOURI ANAS

Année universitaire : 2022-2023

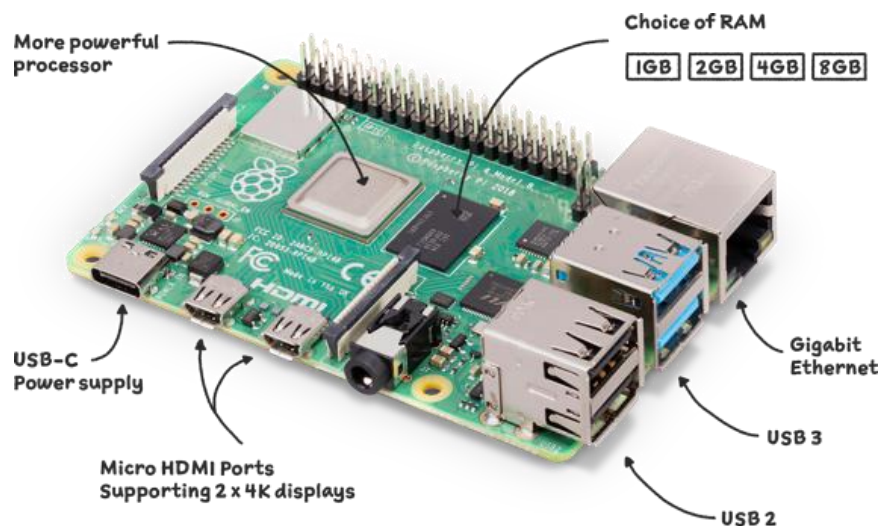




Chapitre 2 : Conception et modélisation du projet

a) Environnement technique du projet

Raspberry Pi 3 : C'est une carte de développement à usage général qui fonctionne comme un serveur dans notre projet. Elle est basée sur une architecture ARM et offre des capacités de traitement et de connectivité. Nous utilisons Raspberry Pi 3 pour héberger les codes de gestion des périphériques et capteurs.



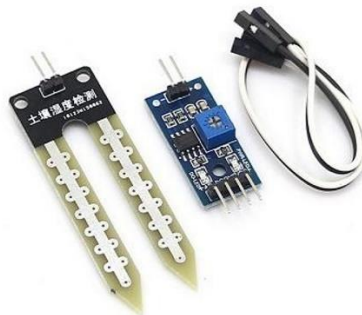
Arduino Uno : C'est une carte de microcontrôleur utilisée dans le projet. Arduino Uno est programmable à l'aide d'une version simplifiée du langage de programmation C/C++. Nous l'utilisons pour interagir avec certains capteurs et actionneurs du système.



Capteur de température DS18B20 : Ce capteur permet de mesurer la température ambiante avec une précision numérique. Il utilise la technologie OneWire pour la communication avec le microcontrôleur. Nous l'utilisons pour acquérir les données de température dans notre Smart Farm.



Capteur de niveau d'humidité YL-69 : Ce capteur est utilisé pour mesurer le niveau d'humidité du sol. Il est généralement associé à un module de capteur d'humidité pour convertir les variations de résistance en données de niveau d'humidité. Cela nous permet de surveiller l'humidité du sol dans notre Smart Farm.



Capteur de pression : Ce capteur est utilisé pour mesurer la pression atmosphérique. Il nous fournit des informations sur les variations de pression, ce qui peut être utile pour la surveillance des conditions météorologiques dans notre Smart Farm.



Capteur de niveau d'eau : Ce capteur est utilisé pour mesurer le niveau d'eau dans un réservoir ou une cuve. Il peut fonctionner en utilisant différents principes de détection, tels que la pression, la flottabilité ou la conductivité. Dans notre Smart Farm, ce capteur nous permet de surveiller le niveau d'eau disponible pour l'irrigation des cultures.



Le capteur de vitesse de vent est généralement constitué de plusieurs composants. Il comprend une hélice ou un anémomètre qui tourne en fonction de la force du vent. Cette rotation est convertie en un signal électrique qui est ensuite traité pour obtenir la vitesse du vent.



Pompe à eau DC : La pompe à eau DC est un dispositif utilisé pour le pompage de l'eau à partir d'une source vers une destination spécifique. Elle est généralement alimentée par un courant continu (DC) et peut être contrôlée pour activer ou désactiver l'écoulement de l'eau. Dans le contexte de votre projet, la pompe à eau DC peut être utilisée pour l'irrigation automatique des terres agricoles. Elle peut être activée en fonction des données mesurées, telles que le niveau d'humidité du sol, afin de fournir la quantité d'eau nécessaire aux cultures.



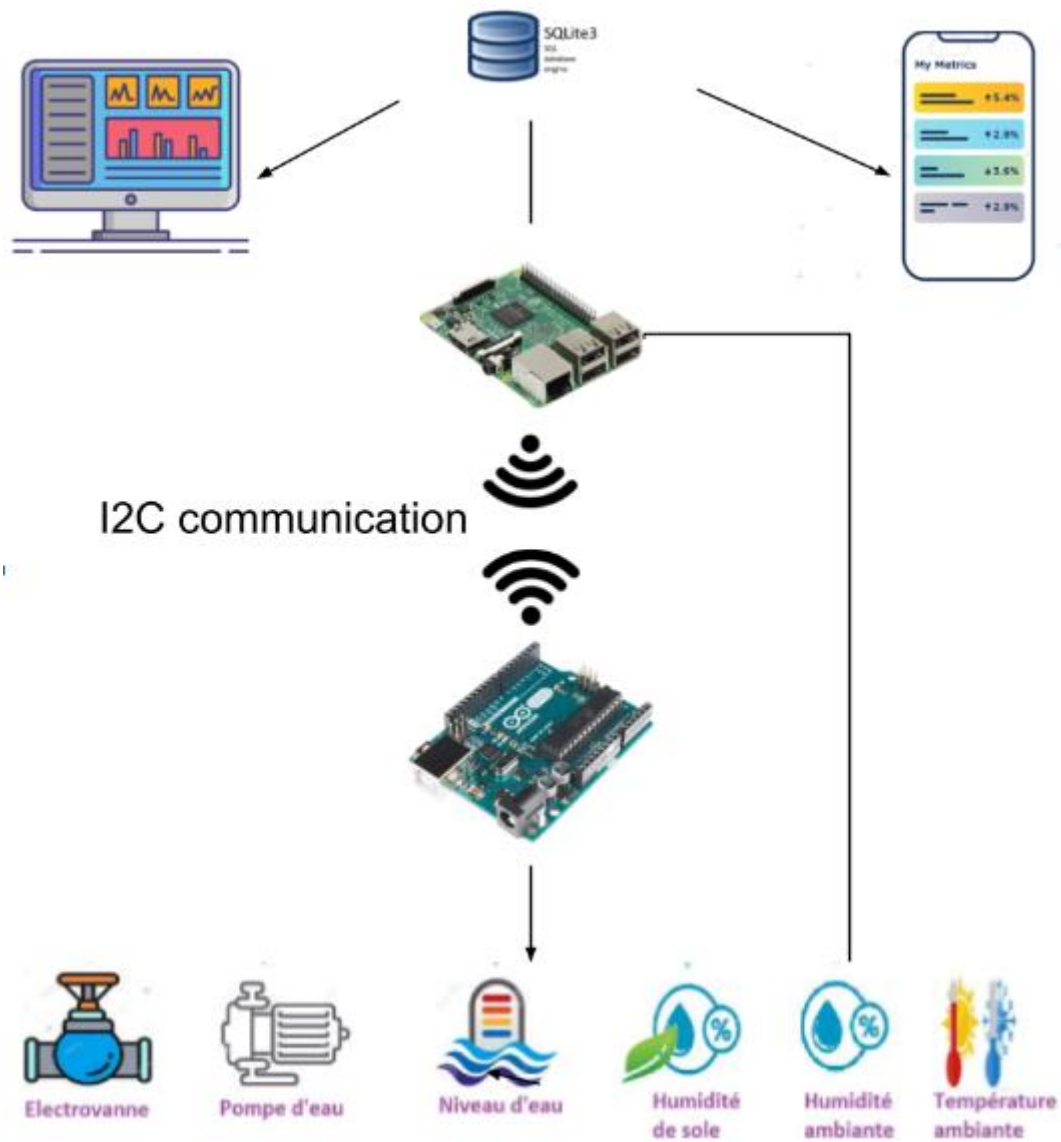
Servo-moteur : Le servo-moteur est un dispositif électromécanique qui permet de contrôler précisément l'angle de rotation d'un arbre. Il est couramment utilisé pour des applications de positionnement ou de contrôle de mouvement. Dans votre projet, le servo-moteur peut être utilisé pour ouvrir ou fermer le bash, en fonction des décisions prises par le système en se basant sur les données environnementales et les prédictions du modèle d'intelligence artificielle. Par exemple, si les conditions météorologiques indiquent une pluie imminente, le servo-moteur peut fermer le bash pour protéger les cultures de l'excès d'humidité.



En ce qui concerne les langages et l'environnement de développement, voici ce que nous utilisons :

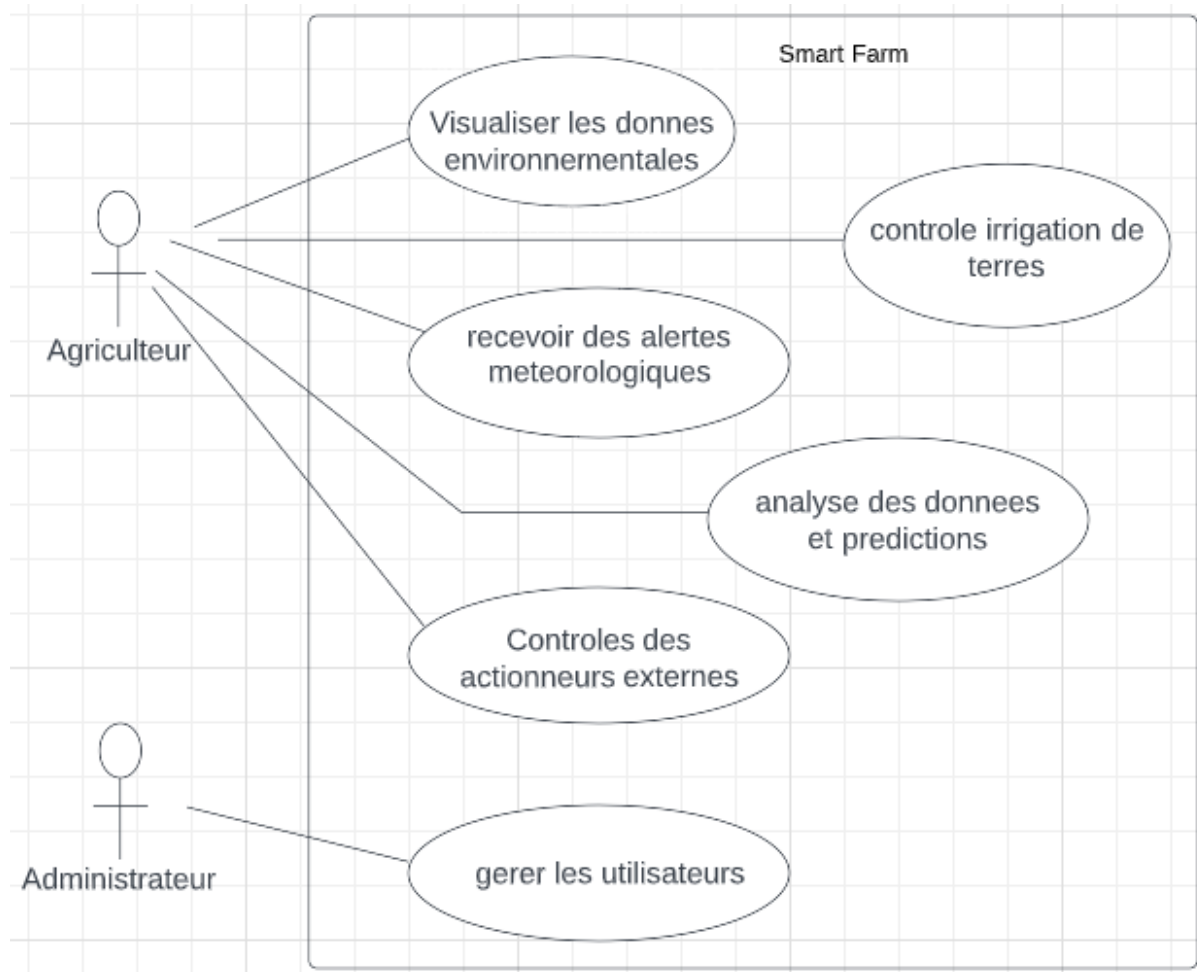
Langages de programmation : Nous utilisons principalement le langage C pour le développement des drivers et des codes liés à l'acquisition de données à partir des capteurs. Nous utilisons également le langage Python pour certaines parties du projet, comme le traitement des données et l'intégration avec l'IA.

Environnement de développement : Nous utilisons Visual Studio Code (VS Code) avec l'extension Remote SSH pour la programmation à distance sur la Raspberry Pi 3. Cela nous permet de développer, tester et déployer les codes directement sur la carte Raspberry Pi 3.



c) Conception détaillée du projet

-diagramme de cas d'utilisation :



Chapitre 3 : Réalisation et test du projet

a) Réalisation du projet

La réalisation du projet a été effectuée en utilisant différentes technologies et langages pour chaque partie de l'application. Voici un aperçu des langages et des technologies utilisés :

1) Partie Serveur (Raspberry Pi 3) :

- Langages : Python, C
- Technologies : Raspberry Pi OS, GPIO (General Purpose Input/Output), protocole I2C

2) Partie Client (Interface Web) :

- Langages : HTML, CSS, PHP
- Technologies : Framework web (le cas échéant), JavaScript (éventuellement pour des fonctionnalités interactives)

3) Base de données :

- Langage : SQL
- Technologies : MySQL, SQLite (selon les besoins)

4) Modèle d'intelligence artificielle :

- Langage : Python
- Bibliothèque : scikit-learn (ou toute autre bibliothèque adaptée pour le modèle de prédiction)

Les différentes parties de l'application ont été développées en utilisant ces langages et technologies, permettant ainsi une intégration harmonieuse entre les différentes fonctionnalités.

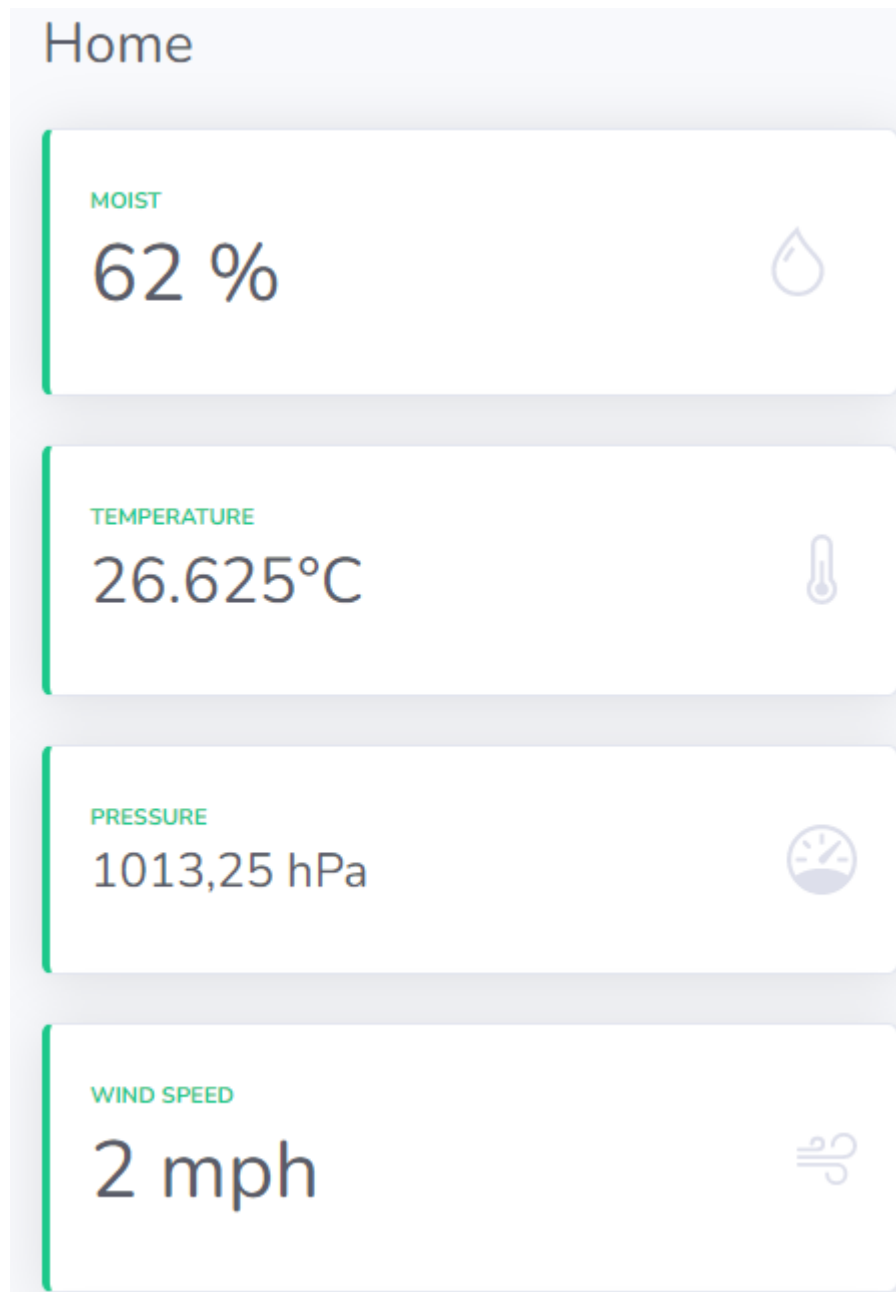
b) Test de l'application

Les tests de l'application ont été réalisés pour valider et vérifier le bon fonctionnement de chaque fonctionnalité. Différents scénarios ont été mis en place, couvrant les différentes fonctionnalités clés de l'application. Voici quelques-uns des scénarios de test :

1) Acquisition des données des capteurs :

- Vérification de la lecture précise des données de température à partir du capteur DS18B.

- Vérification de la lecture précise des données d'humidité à partir du capteur YL_69.
- Vérification de la lecture précise des données de pression à partir du capteur de pression.
- Vérification de la lecture précise des données de niveau d'eau à partir du capteur de niveau d'eau.
- Vérification de la détection précise de la vitesse du vent à partir du capteur de vitesse de vent.



2) Contrôle des actionneurs :

- Vérification du fonctionnement correct de la pompe à eau DC en démarrant et en arrêtant l'irrigation.
- Vérification du fonctionnement correct du servo-moteur pour l'ouverture et la fermeture du bash.

3) Prédiction des changements climatiques :

- Vérification de la précision de la prédiction des changements climatiques en utilisant le modèle d'intelligence artificielle.
- Comparaison des prédictions avec les données météorologiques réelles pour évaluer l'efficacité du modèle.

Prediction

26.562

62

2

1100

la sortie = 0

Bash opened

Des captures d'écran ont été prises à chaque étape des tests pour illustrer les résultats obtenus. Les captures d'écran sont accompagnées de commentaires détaillés expliquant les résultats et les performances de l'application.

Ces tests ont permis de valider la fiabilité et la précision de l'application, en s'assurant que les capteurs fonctionnent correctement, que les actionneurs sont contrôlés de manière adéquate et que le modèle d'intelligence artificielle fournit des prédictions précises.

Conclusion générale

En résumé, le projet de Smart Farm consiste en la mise en place d'un système de surveillance et de contrôle environnemental utilisant divers capteurs pour mesurer la température, l'humidité, le niveau d'eau, la pression et la vitesse du vent. Le système est géré par une Raspberry Pi, qui communique avec les capteurs et les actionneurs tels que les pompes à eau et les servomoteurs pour le contrôle des conditions agricoles. L'application a été développée en utilisant des langages tels que Python, Flask, HTML, CSS et JavaScript. Les tests réalisés ont validé la performance et la stabilité du système. En conclusion, ce projet a permis de mettre en pratique des compétences en développement et en surveillance environnementale pour une agriculture intelligente et durable.

