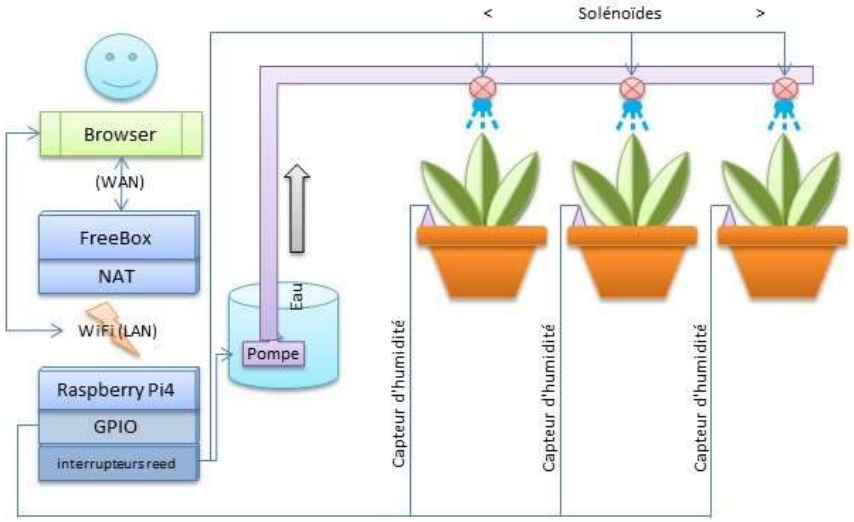
Système d'arrosage intellectuel

Introduction générale

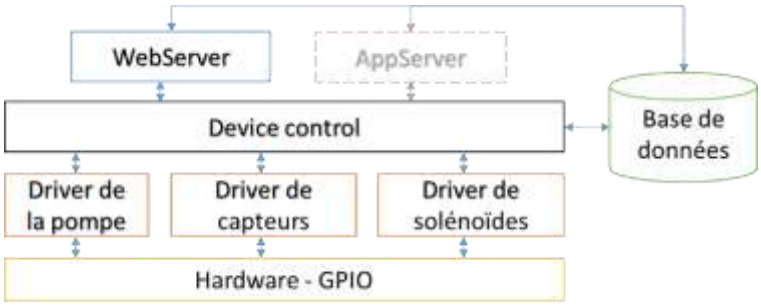
Mon idée du projet c'est **un système surveillant l'arrosage en tenant compte de besoins de chaque plante** (et non selon un horaire fixe quelle que soit la météo). Ce système serait très utile pour tout le monde qui aime la beauté de nature dans sa demeure, mais qui 1) n'a pas la possibilité de surveiller ses plantes, 2) part en vacances en période de canicules et en rentrant retrouve les restes incinérés de ses plantes ou 3) n'a pas le talent de jardinier. Je mûris cette idée depuis longtemps. J'ai déjà acquis quelques matériels nécessaires pour ce projet et je suis en train de développer le schéma de circuit électrique (voir bloquée). Cependant, je n'ai pas encore commencé à faire le software. Ce serait formidable de faire ce projet maintenant, quand il y a une opportunité de demander des conseils et d'échanger des idées. Je suis en train de chercher le binôme.

Fonctionnalité et matériels

En mode automatique le contrôleur (Raspberry Pi4) interrogera les capteurs tôt le matin (à une certaine heure). En se basant sur les données d'humidité du sol, le système n'arrose que les pots où c'est nécessaire et avec autant d'eau que son plante en a besoin. Pour le faire, il est censé maintenir une base de données de plantes connectées au système, où le degré de leurs besoins d'eau sera fixé. Par exemple, les cactus nécessitent moins d'eau par arrosage que les roses. Si les capteurs de certains pots indiquent une humidité du sol insuffisante, RaspberryPi enverra un signal à la pompe et aux solénoïdes appropriés, ils ouvriront le flux d'eau vers ces pots et l'arrosage se produira.



L'utilisateur pourra consulter les logs via l'application et/ou sur le site afin de savoir si le système fonctionne correctement en son absence. Pour cela, il faut déployer un serveur web sur RaspberryPi et/ou écrire une application d'Android. En cas de nécessité l'utilisateur pourra passer de mode automatique à mode manuel et effectuera l'arrosage à distance. ***L'extension possible du projet :*** possibilité de brancher à distance une caméra afin de voir en direct sur le site et/ou dans l'appli qu'est-ce qui se passe dans jardin.



Objectifs

Partie programmation à faire (en C++)

Les **drivers** 1. Gestion d'ouverture/fermeture de solénoïdes 2. Interprétation de données arrivant de capteurs 3. Gestion de la pompe	**Base de données** contenante : 4. Besoins d'eau de plantes connectées 5. Données de capteurs (numéro de port, caractéristiques techniques)
Device control 6. Contrôle de materiel externe 7. Access à la base de données 8. Décisions d'irrigation automatique 9. Fournit les interfaces pour web serveur afin d'accéder au hardware (en mode manuel) 10. Effectue la collecte de statistique	**Web serveur** (**appli serveur** en option) qui fait : 11. Affichage des logs de N dernières sessions 12. Ajouter/enlever des plantes du système 13. Passer en mode manuel pour effectuer l'arrosage à distance 14. Session visio (en option, voir l'extension)

Calendrier

Reste du novembre	~~Choix du reste du matériel (interrupteurs, convertisseurs)~~, montage du circuit électrique ~~Elaboration de specification~~
Mi-décembre	Drivers de capteurs (2) Lancement de web-serveur (partie représentation, sans fonctionnalité) + BD (4,5)
Janvier	Drivers de la pompe, et solénoïdes ; première version de Device Control (1, 3, 6) Web-server : ajout/suppression de plante (7, 12)
Février	Gestion automatique, gestion manuel (8, 9, 10) Web-server : logs et rapports sur le site (11), gestion manuel
Mars	Apprentissage
Avril	Session visio (extension du projet, si on arrive à faire les taches précédentes sans retard) (14)

Apprentissage

Vu que la fiabilité des capteurs peut se varie, surtout puisqu'ils travaillent dans un environnement corrosif (de l'eau, des sels et des autres agents chimiquement actifs composants les terrains), la défaillance des capteurs est le plus probable parmi tous parties du système. Alors, il est nécessaire de prévoir le moyen de correction des erreurs liées à la réception de données incorrectes provenant de capteurs.

On a besoin d'un modèle illustrant la diminution d'humidité dans les pots en fonction de la consommation des plantes, de la température et des précipitations.

Approche 1

Les cas où on applique le modèle de prédiction (prenons les valeurs du modèle au lieu des valeurs des capteurs) :

1. Erreur de communication avec le capteur (n'importe quelle défaillance du matériel)
2. Valeurs arrivantes de capteur sont hors des limites acceptables, qui ont été trouvés lors du calibrage des capteurs.
3. Si les valeurs du capteur dépassent la différence du modèle (par exemple de 25 %)

Approche 2

On met deux capteurs (C et C') dans chaque pot (P). Cela nous permettra à fixer le désaccord possible entre ces deux.

Axiomes :

1. Tous les pots/plantes connectées sont dans les mêmes conditions météorologiques (sous ou sans abri, à l'ombre ou au soleil etc.)
2. Données arrivant de C = données arrivant de C' (mod degré de précision des capteurs).
3. Taux d'humidité augmente après l'arrosage, h(i) > h(i+1)
4. Taux d'humidité n'augmente pas sans arrosage, h(i) >= h(i+1)
5. L'humidité change entre les jours à peu près de même % pour tous les pots. C'est-à-dire que s'il pleuvait, alors l'humidité augmentera partout de à peu près de même %, s'il n'y avait pas de pluie, alors elle diminuera à peu près de même % (mod capacité d'absorption des plantes). Plus compliqué à traiter

Les modèles possibles :

La plus facile - régression linéaire avec les coefficients qui se basent sur :

- la consommation des plantes
- la statistique des température et l'humidité moyennes par mois dans une région donnée.

Tests

	A quoi s'agit-il ?	Tests
Tests des équipements électroniques Tests de drivers de capteurs Tests de drivers de solénoïdes Tests de drivers de la pompe	Normalement, il faut créer l'assemble de stands des capteurs liés aux Raspberries, varier les paramètres extérieurs (la température, d'humidité, etc.) et vérifier que les capteurs et l'ensemble du système se comportent correctement. Il n'y a aucun moyen d'automatiser avec notre durée du projet et budget actuel.	En option, on peut créer un ensemble de tests de base : - capteur dans un endroit sec - capteur dans un endroit humide Allumer le système, le laisser faire arrosage. Mesurer l'eau versée. Comparer avec la quantité de l'eau attendu. Car le but est de verser la bonne quantité d'eau.

Tests unitaires	Il s'agit de tester les méthodes d'une classe du programme indépendamment du reste du programme.	
Tests d'intégration	Les tests entre modules, ce sont des tests API entre différentes parties - test API de contrôle de périphérique - connexion avec la base de données - connexion avec le serveur Web	Tests API de contrôle de périphérique :
Tests système	Tests d'ensemble du système : ses performances, sa résistance, sa stabilité de fonctionnement à long terme (fuites de mémoire, crashs, fiabilité fonctionnelle)	

Specification :

https://docs.google.com/document/d/1AQ92oxA7R-2l6DDeZUX3vQeg0GsiFC95DZXq7_qI9EQ/edit?usp=sharing