

ARROSAGE INTELLIGENT

CAHIER DES CHARGES

Projet IOT MASTER 2 S2I



UNIVERSITÀ DI CORSICA

PASQUALE PAOLI

Table des matières

1Présentation.....	3
1.1Contexte.....	3
1.2Existant.....	3
2Expression des besoins fonctionnels.....	3
2.1Version 1.....	3
2.2Version 2.....	4
2.3Version 3.....	4
2.4Version 4.....	4
3Expression des besoins non fonctionnels.....	4
3.1Hardware.....	4
3.1.1Capteur d'humidité.....	4
3.1.2Capteur dans la cuve de récupération.....	4
3.1.3Capteur de débit d'eau.....	4
3.1.4Contrôleur.....	5
3.1.5Electrovanne.....	5
3.2Communication.....	5
3.2.1Capteurs → arduino.....	5
3.2.2Arduino → Internet.....	5
3.2.3Arduino → Electrovanne.....	5
3.3Alimentation.....	5
4Contraintes.....	6
4.1Contraintes de délai.....	6
4.2Contraintes liées à la nature « aquatique » du projet.....	6
5Déroulement du projet.....	6
5.1Développement de la version 1.....	6
5.2Développement de la version 2.....	6
5.3Développement de la version 3.....	6
5.4Développement de la version 4.....	6
6Evaluation de la difficulté.....	7

1 Présentation

1.1 Contexte

Dans un contexte d'économie des ressources naturelles, on veut proposer un système d'arrosage automatique qui optimisera la dépense en eau. Les paramètres d'entrée envisagés pour le système d'optimisation sont :

- le taux d'humidité du sol,
- les prévisions météorologiques au lieu d'arrosage,
- le niveau de remplissage de la cuve de récupération des eaux de pluie.

On pourra également déclencher l'arrosage manuellement via une application Android.

1.2 Existant

Les systèmes d'arrosage programmés (arrosage journalier, hebdomadaire, ...) sont courants.

Les systèmes qui s'adaptent aux conditions climatiques émergent, comme les systèmes Smart Water de Hunter (Solar Sync EvapoTranspiration sensor).

2 Expression des besoins fonctionnels

Les exigences fonctionnelles (notées REQ-xx-Vy) sont réparties en 3 versions incrémentales, intégrant des améliorations après validation de la version précédente.

2.1 Version 1

REQ-01-V1

Le système doit prendre en compte le taux d'humidité du sol et les prévisions météorologiques pour déclencher l'arrosage selon les règles suivantes :

- si $\text{taux} > \text{taux_humidité_2}$, pas d'arrosage
- si $\text{taux_humidité_1} < \text{taux} < \text{taux_humidité_2}$ et pluie prévue dans les prochaines 48h, pas d'arrosage
- si $\text{taux} < \text{taux_humidité_1}$, arrosage

REQ-02-V1

Le système doit prendre en compte le taux d'humidité du sol pour arrêter l'arrosage.

- si $\text{taux} > \text{taux_humidité_3}$, fin de l'arrosage

REQ-03-V1

L'application mobile permet de paramétrer le lieu pour lequel on requiert les prévisions météorologiques.

REQ-04-V1

Le site web et l'application mobile permettent de visualiser quand l'arrosage a fonctionné.

2.2 Version 2

REQ-05-V2

Le capteur d'humidité du sol doit pouvoir être déporté et déconnecté du système de commande.

2.3 Version 3

REQ-06-V3

Le site web et l'application mobile permettent de visualiser quelle quantité d'eau de ville a été consommée.

2.4 Version 4

REQ-07-V4

Tant que la cuve d'eau de récupération est suffisamment pleine, elle sera utilisée de façon préférentielle.

- si la cuve est suffisamment pleine, la cuve est sélectionnée,
- sinon l'eau de ville est sélectionnée.

3 Expression des besoins non fonctionnels

On répartit les besoins non-fonctionnels en 3 catégories :

- Les besoins en hardware : capteurs et contrôleurs,
- Les besoins de communication : entre le(les) capteur(s) et l'arduino contrôleur, entre l'arduino et internet, entre l'arduino et l'électrovanne,
- Les besoins d'alimentation des capteurs, de l'arduino et de électrovanne.

3.1 Hardware

3.1.1 Capteur d'humidité

Le taux d'humidité du sol sera mesurer grâce à un capteur d'humidité des sols.

3.1.2 Capteur dans la cuve de récupération

Le niveau de remplissage de la cuve pourra être mesuré avec un capteur à ultrasons ou bien un capteur d'immersion qui serait fixé à une hauteur donnée.

3.1.3 Capteur de débit d'eau

Dans un but de suivi du coût de l'arrosage, on utilisera un capteur de débit d'eau sur l'arrivée d'eau

de ville.

3.1.4 Contrôleur

Le système sera contrôlé par un Arduino YUN.

3.1.5 Electrovanne

Une electrovanne permettra d'ouvrir/fermer l'arrivée d'eau.

Dans la version 3, une seconde electrovanne serait nécessaire pour exploiter la cuve de récupération d'eau de pluie.

3.2 Communication

3.2.1 Capteurs → arduino

On peut envisager 2 types de communication :

- filaire : les capteurs sont directement branchés sur les entrées analogiques de l'arduino,
- sans-fil : la communication doit être économe en énergie, (ZigBee entre le capteur et l'arduino ou SigFox entre le capteur et le serveur de données SigFox).

Il semble nécessaire de pouvoir déporter le capteur d'humidité à distance du contrôleur.

On peut imaginer que l'arrivée d'eau de ville, la cuve et le contrôleur seront dans un périmètre restreint et que par conséquent des connections filaires entre le contrôleur et les capteurs de débit d'eau et de niveau de la cuve sont acceptables afin de limiter le nombre de batteries utiles pour le système.

3.2.2 Arduino → Internet

Différents types de communication peuvent être envisagés, le plus simple à mettre en œuvre étant le WiFi. Une connection de type SigFox sera à étudier.

3.2.3 Arduino → Electrovanne

Il semble raisonnable que l'arduino soit connecté en filaire à l'electrovanne pour une commande directe à l'aide des sorties digitales de l'arduino.

3.3 Alimentation

Si l'electrovanne est branchée sur l'Arduino, il semble cohérent que les 2 matériels aient le même mode d'alimentation. Selon le modèle d'électrovanne disponible, on choisira donc une batterie ou le secteur.

Le système n'ayant pas besoin d'une haute réactivité, il faudra étudier de quelle façon il est possible de n'allumer l'Arduino que sur une période limitée chaque jour.

L'alimentation par un panneau solaire sera à étudier.

L'alimentation du capteur d'humidité déporté sera à étudier.

4 Contraintes

4.1 Contraintes de délai

Le temps disponible pour la réalisation de ce projet étant restreint.

→ On a choisi de définir **plusieurs versions du projet**, intégrant les difficultés petit à petit, afin d'assurer qu'une version minimale (V1) soit effectivement opérationnelle en décembre.

4.2 Contraintes liées à la nature « aquatique » du projet

La nature même du projet rend les tests assez difficiles à réaliser : en effet, même si on dispose d'un pot de terre pour simuler un jardin, une fois le pot arrosé, il n'est plus possible de refaire un test avec une situation de sécheresse.

Par ailleurs le capteur même ne sera disponible que fin octobre.

→ On utilisera dans un premier temps un capteur de température pour valider le principe du système (**REQ-01-V1** et **REQ-02-V1**)

De même, il n'est pas facile de tester l'utilisation de l'électrovanne.

→ On utilisera dans un premier temps une LED pour simuler l'ouverture/fermeture de l'électrovanne. (**REQ-01-V1** et **REQ-02-V1**).

Aucune solution de contournement ne sont actuellement envisagées pour les versions 3 et 4.

5 Déroulement du projet

On développera le système de façon incrémentale.

5.1 Développement de la version 1

Développement de l'application mobile et du site web.

Validation avec capteur de température et LED, puis avec capteur d'humidité et électrovanne.

5.2 Développement de la version 2

Déport du capteur d'humidité, communication ZigBee.

5.3 Développement de la version 3

Ajout du capteur de débit d'eau, visualisation du volume d'eau consommée sur l'application mobile et le site web.

5.4 Développement de la version 4

Ajout du capteur de niveau de la cuve et commande d'une deuxième électrovanne.

6 Evaluation de la difficulté

ID	Label	v1	v2	v3	v4
01	Composition matérielle	3	3	4	5
02	Mode de communication M2M	4	6	6	6
03	Utilisation de Web Service	3	3	3	3
04	Développement d'une application mobile	4	4	4	4
05	Développement d'un site web	3	3	3	3
06	Base de données	1	1	1	1
07	Traitement des données	3	3	3	4
08	Gestion énergétique	4	4	4	4
09	Interaction avec l'environnement	5	5	5	5
10	Déplacement				
11	Intelligence Artificielle				
12	Utilisation d'un gestionnaire de version	1	1	1	1
13	Qualité du code	3	3	3	3
14	Documentation	3	3	3	3
15	Etude de marché	1	1	1	1
16	Choix technologiques	5	5	5	5
17	Identification				
18	Optimisation	1	3	3	3
	Difficulté totale	44	48	49	51