

GNG 1503
Manuel d'Utilisateur du Projet de Conception

La Jardinière Autonome

Soumis par:

[**SUCRE&POTAGER** - A10]

[Danika Longtin 1, 300132771]

[Kassoum Sere Hamdan 2, 300209975]

[Lina Oubia 3, 300156514]

[Wadie Sbai 4, 300191800]



uOttawa

L'Université canadienne
Canada's university

9 Décembre, 2020

Résumé

Ce manuel permet à l'utilisateur le maintien, la reproduction ainsi que l'amélioration correcte et complète de notre concept de jardinière à arrosage automatique. L'utilisateur trouvera par la suite une liste de figures lui permettant de clarifier et de concrétiser son idée sur la jardinière, sa structure, sa forme ainsi que son style. Ensuite, Il trouvera les instructions nécessaires et suffisantes de l'utilisation et le contrôle de la jardinière globalement et de ses sous-systèmes précisément. Ces derniers seront aussi décortiqués, et leurs fonctionnements seront bien expliqués afin de faciliter la tâche de l'utilisation. Et concernant la reproduction de la jardinière, ce manuel offre à l'utilisateur une liste assez détaillée de matériaux et composants électroniques nécessaires, leurs quantités, ainsi que des liens où il pourra les acheter, en lui précisant aussi le prix moyen de chaque composant.

Pour le sous-système de pompage et de détection d'humidité, ce manuel offre à l'utilisateur le code dont il aura besoin, le logiciel où insérer le code ainsi qu'un schéma de câblage accompagné d'une explication simple que l'utilisateur doit poursuivre pour accomplir un bon câblage qui lui permettra la détection de l'humidité du sol et la commande de la pompe pour pomper l'eau et arroser les plantes.

Ainsi pour l'interface Web, l'utilisateur pourra bénéficier d'un accès global au système d'arrosage, dans ce manuel, il pourra avoir accès à toutes les informations qui ont permis de mettre sa mise en place, donc il aura la possibilité de l'améliorer; du point de vue du design de la page web, ou il peut aussi inclure plus d'icônes ou même améliorer le code afin de faire afficher plus d'informations etc...

En gros, ce manuel permet à l'utilisateur la concrétisation du produit ainsi que son amélioration.

Table des matières

1	Introduction	5
1.1	Organisation et classification des besoins	5
1.1.1	Problème de conception	6
1.1.2	Leçons et conclusions	6
1.2	Critères de conceptions:	7
1.2.1	Étalonnage	8
1.2.2	Tableau d'importance	9
1.2.3	Spécifications	10
1.2.4	Leçons et conclusions	11
1.3	- Concepts préliminaires	12

1.3.1 Concepts de sous-systèmes	12
1.3.1.1 Concepts de Danika	12
1.3.1.2 Concepts de Kassoum	13
1.3.1.3 Concepts de Wadie	15
1.3.1.4 Concepts de Lina	16
1.3.2 Concepts des systèmes complets	17
1.3.2.1 Le système fonctionnel # 1	17
1.3.2.2 Système fonctionnel # 2	17
1.3.2.3 Système fonctionnel # 3	18
1.3.3 Analyse et évaluation des systèmes	18
1.3.4 Meilleure solution	19
1.3.5 Leçons et conclusions	19
1.4 Prototypes #1	20
1.4.1 Tableau du plan de test des prototypes	20
1.4.2 Présentation des tests et des prototypes	24
1.4.2.1 Représentation 3D du prototype à base de matériaux de maison - Danika	24
1.4.2.2 Schémas de Câblage du concept de base - Kassoum	26
1.4.2.3 Schémas de Pompage de l'eau - Lina	27
1.4.2.4 Représentation de l'interface WEB - Wadie	29
1.4.3 Leçons et conclusions	29
1.5 Prototypes #2	30
1.5.1 Tableau du plan de test des prototypes	30
1.5.2 Présentation des tests et des prototypes	33
1.5.2.1 Représentation 3D du prototype à base de matériaux de maison - Danika	33
1.5.2.2 Schémas de Câblage du concept de base - Kassoum	35
1.5.2.3 Schémas de Pompage de l'eau 3D et Concept du capteur de niveau d'eau- Lina:	41
1.5.2.4 Représentation de l'interface WEB - Wadie	42
1.5.3 Leçons et conclusions	44
1.6 Prototypes #3	44
1.6.1 Tableau du plan de test des prototypes	45
1.6.2 Résultats des tests	47
1.6.3 Rétroaction du client	50
1.6.4 Leçons et conclusions	50
2 Comment le prototype est construit	51
2.1 Catégorie	51
2.1.1 LDM (Liste des Matériaux)	51
2.1.2 Liste d'équipements	52
2.1.3 Instructions	53
3 Comment utiliser le prototype:	55

4	Comment maintenir le prototype:	55
5	Conclusions et recommandations pour les travaux futurs	57
6	Bibliographie	57

Liste de figures

Image A
Image B
Image C
Image D
Image E
Image F
Image G
Image H
Image I
Image J
Image K
Image L
Image M
Image N
Image O
Image P
Image Q
Image R
Image S

Liste de tables

Tableau 1
Tableau 2
Tableau 3
Tableau 4
Tableau 5
Tableau 6
Tableau 7
Tableau 8
Tableau 9
Tableau 10
Tableau 11
Tableau 12

Liste des acronymes:

Les acronymes et les définitions sont présentés ci-dessous pour établir les notions courantes pour les utilisateurs de ce manuel.

Acronyme	Définition
cm	Centimètre, unité de longueur
kg	Kilogramme, unité de masse
m	Mètre, unité de longueur
nodeMCU	Ulate-forme open source IoT , matérielle et logicielle ^{4,5} , basée sur un SoC Wi-Fi ESP8266 ESP-12 fabriqué par Espressif Systems
l	Longueur
m²	mètre carré, unité d'aire
m³	Mètre cube, unité de volume
h	heure, unité de temps
min	Minute, unité de temps
s	Seconde, unité de temps

kg/m³	Unité de masse volumique
\$	dollar canadien, unité monétaire
L	litres, Unité de volume
v	volume
API	Automate programmable industriel
V	Voltes, unité de tension
N	Newton, unité de poids
HTML	<i>HyperText Markup Language</i> , un langage de balisage conçu pour représenter les pages web.

1 Introduction

Il y a un manque de verdure sur le campus de l'université d'Ottawa. La verdure augmente le taux de concentration et la durée de focus d'un étudiant lorsqu'il ou elle étudie. C'est pourquoi il est important d'augmenter le niveau de présence de verdure dans les édifices du campus. L'utilisateur aurait besoin d'une jardinière autonome qui contient des plantes et les abreuve. L'entretien de la Jardinière doit être minime pour les responsables de l'entretien. La jardinière doit être facile d'utilisation, facile à laver et le remplissage doit se faire sans causer de dégât. Notre produit est un atout compétitif sur le marché. L'interface web facilite son utilisation, sa conception minimise l'utilisation d'eau, elle assure son bon fonctionnement et la santé des

plantes. La Jardinière est un produit qui va permettre d'ajouter de la verdure sur le campus. Elle arrose les plantes à l'aide d'une pompe et l'arrosage se fait en fonction du pourcentage d'humidité de la terre détecté par une sonde d'humidité. L'eau en excès est filtrée et renvoyée au réservoir, le bouchon de remplissage est facile à ouvrir et donne directement accès au réservoir. Le couvercle de la jardinière s'ouvre, ce qui donne accès à l'utilisateur toutes les composantes intérieures.

1.1 Organisation et classification des besoins

Légende:

1	2	3	4	5
non nécessaire	non important	Non-nécessaire mais important	Nécessaire mais pas important	Nécessaire et important

Classification	Besoins	Importance
Aspects économiques	1. La jardinière coûte au plus 100 \$	5
	2. La jardinière est autonome et faible d'entretien	5
	3. Les matériaux utilisés pour la jardinière sont durable, de bonne qualité et à bon prix	4
Aspects environmental	4. Les matériaux utilisés pour la jardinière sont bon pour l'environnement	3
	5. La jardinière limite sa consommation en énergie	3
Aspects sécuritaires	6. La jardinière n'attire pas d'insect	4
	7. La jardinière est sans moisissures	4
	8. La jardinière est solide	4
Aspects d'adaptation	9. La jardinière est mobile	5
	10. La jardinière s'adapte à plusieurs types de plantes	3
	11. La jardinière est apte à être installé dans différent édifice du campus	4

Aspects physiques	12. La jardinière as un allure professionnel, futuriste et attirante	3
Aspects fonctionnelles	13. La jardinière détecte l'humidité du sol	5
	14. La jardinière arrose les plantes	5
	15. La jardinière alerte le client lorsque le réservoir d'eau est vide	3
	16. La jardinière est facile à utiliser	4

1.1.1 Problème de conception

Le représentant du bureau du développement durable à l'université d'Ottawa, Jonathan Rausseo a besoin d'une jardinière intérieure pour les édifices du campus, facile d'utilisation, qui détecte l'humidité du sol et arrose les plantes. Ce produit doit être sécuritaire, attrayant, mobile et bon pour l'environnement tout en étant sous le budget de 100\$.

1.1.2 Leçons et conclusions

L'interaction avec le client lors de l'entrevue du 24 septembre a permis d'avoir une perception plus claire des besoins, problèmes, limitations, frustrations et désirs du client. Ce en faisant preuve d'empathie. Une question qui n'a pas été abordée lors de la réunion initiale du client est sa préférence de format. Par exemple, si le client veut sa jardinière en forme de boîte, cylindrique, sphère, debout sur des pattes, accrochées au mur, etc. Un besoin intéressant à inclure serait une application qui indique par exemple l'humidité, le temps d'arrosage et la quantité d'eau restante dans le réservoir. Ce projet est un défi dont la résolution va devoir impliquer le talent et le domaine d'étude de chaque membre de l'équipe. Hiérarchiser les besoins du client lors de la conception est un élément important pour assurer la satisfaction de celui-ci. La jardinière aussi appelée "un planteur" apportera de la verdure sur le campus ce qui invite les élèves à utiliser davantage les espaces communs du campus pour étudier. C'est ce qu'on désire lorsque la pandémie sera terminée.

1.2 Critères de conceptions:

Légende

1	2	3	4	5
Indésirable	Pas important	Important mais non nécessaire	Important et nécessaire	Critique

Critères de conception		Type de critère:	Répond au besoin (énumérés dans l'annexe 1)	Importance Relative
Un Système d'arrosage intégré		fonctionnelle	2	5
1.	Type de système d'arrosage	Fonctionnelle	14	4
2.	Coût de système d'arrosage	Contrainte / Métrique	1	5
Système de détection d'humidité intégré		Fonctionnelle	13,16,7	5
1.	Type de système de détection d'humidité	Fonctionnelle	5,16,13	4
2.	Coût de système de détection d'humidité	Contrainte /Métrique	1	5
Esthétique		Non-fonctionnelle	12	3
1.	Allure futuriste et atrayente	Non-fonctionnelle	12	3
2.	Forme de la Jardinière	Non-fonctionnelle	12,9	4
Matériaux utilisés pour la jardinière		Contrainte / Non-fonctionnelle	8,12	4
1.	Provenance des matériaux (Ecologie)	Non-fonctionnelle	4	3
2.	Qualité des matériaux	Non-fonctionnelle	8	4
3.	Coût des matériaux	Contrainte / Métrique	1,3	5
Le coût total de la Jardinière		Contrainte / Métrique	1	5
Le poids de la Jardinière		Métrique	11,16,8	4
Consommation énergétique de la Jardinière		Contrainte / Métrique	5	5
		Fonctionnelle	10,11,14	5

Volume total de la Jardinière				
1.	Volume du bac de terre	Fonctionnelle	10	3
2.	Volume du réservoir d'eau	Fonctionnelle	17,2	4
Système de détection de la quantité d'eau restante dans le réservoir		Fonctionnelle	2,16	5
Système de commande à distance		Fonctionnelle	16	4

Tableau 1-Critères de conceptions

1.2.1 Étalonnage

Données d'analyse	ESOTEC 101100	Gardena flex 1890	Jeteven JETEVENFRHOCORA789
couts	70.99\$	139.99\$	29.99\$
Consommation énergétique	Alimentation par énergie solaire si absence de piles	1 pile alcaline 9V(Électrique)	électrique
Niveau Écologique	Peut se passer d'électricité avec l'énergie solaire	Arrêt et mise en marche automatique pour économiser de l'eau et du carburant	Permet d'économiser 70 % d'eau par rapport à un arrosage manuel
Facilité de transport	Léger et dimensions considérables	Très petit et léger	Lourd et grandes dimensions
Autonomie et fréquence d'arrosage	Automatique , fréquence réglable entre 1h et 7 jours	Automatique , fréquence réglable entre 4h et 72h	Automatique
Poids	1.3kg	0.6 kg	1.2 kg
Taille	22 x 19 x 16 cm	2.54 x 2.54 x 2.54 cm	36 x 23 x 2 cm
Durabilité	A tester	5 à 8 ans	A tester
Système de réglage d'arrosage	inclus	inclus	inclus
Résistance de la	Métal résistant au soleil	Résistant aux	Plastique ABS résistant

matière de fabrication		intempéries	
Type d'arrosage	Goutte à goutte	Arrosage rotatif / goutte à goutte	Goutte à goutte
Système de detection d'humidite	Absent	inclus	Absent
Type de systeme de detection d'humidite	-----	Sonde d'humidite	-----
Esthétique	Insatisfaisant	satisfaisant	Insatisfaisant

Tableau 2- Etalonnage

1.2.2 Tableau d'importance

Le tableau d'importance permet de trouver le meilleur des trois produits établis lors de l'étalonnage. Premièrement, le niveau de qualité de chaque produit en fonction de chaque donnée d'analyse est établi sur une échelle de 1 à 3, 3 étant le meilleur produit selon le critère. Ensuite, ce niveau de qualité est multiplié par l'importance de chaque critère de conception. Toutes les valeurs pour un produit sont additionnées. Le produit avec la valeur finale la plus élevée désigne le produit qui répond le plus aux attentes de notre client.

Critères	Importance	ESOTEX 10100	Gardena flex 1890	Jeteven JETEVENFRH OCORA789
Couts	5	2	1	3
Consommation énergétique	5	3	2	2
Niveau écologique	3	3	2	2
Facilité de transport	4	2	3	1
Autonomie et fréquence d'arrosage	4	3	2	1
poids	4	1	3	2
taille	5	1	3	2

durabilite	4	1	3	1
Résistance de la matière de fabrication	4	3	3	3
Système d'arrosage	5	3	3	3
Type d'arrosage	4	2	3	2
Système de détection d'humidité	5	1	3	1
Type de système de détection d'humidité	4	1	3	1
Esthétique	3	2	3	2
total	----	117	155	111

Tableau 3 - Sélection du meilleur produit

1.2.3 Spécifications

	Spécifications de conception	Relation (=, < or >)	Valeurs	Unité	Méthode de vérification
Exigences fonctionnelles					
1	réservoir	-	-	litres	Essai
2	eau	>	5	litres	Collecte d'eau de pluie
3	Pompe	>	5	L/min	Essai
4	Contrôle de l'humidité	>	6	C	Sonde
Exigences non-fonctionnelles					
1	motifs(logo uottawa)	-	-	-	Essai
2	Plantes différentes	-	-	-	Essaie
3	Simplicité	-	-	-	Essaie
4	fiabilité	-	-	-	Essaie
Contrainte					

1. 1	Matériel utilisé	-	Arduino Pompe a eau Capteur d'humidité terminaison d'un arroseur Module raspberry pi Node MCU	-	Essaie
2	coût	<	100	\$	budget
3	poids	<	25	kg	Analyse
4	longuer*largeur	=	60*40	cm	Analyse
5	hauteur	<	150	cm	Analyse

Tableau X - Liste de spécifications cibles

1.2.4 Leçons et conclusions

Les besoins du client ont pu permettre d'établir une liste claire et précise des critères nécessaires à la conception de la jardinière. La plupart des critères répondent aux besoins interprétés lors du livrable B. Les besoins sont énumérés de 1 à 17 dans l'annexe 1. Les besoins associés à chacun des critères ont été spécifiés dans le Tableau 1. Sur le mauvais côté des choses, les besoins en lien avec la sécurité n'ont pas tous été touchés par les critères, comme le besoin 6 et 7. La liste de critères de conception a permis aux membres de l'équipe d'établir des données d'analyse pour l'échantillonnage des produits existants. L'analyse de chacune des données d'analyse des produits a permis d'établir le meilleur produit à l'aide de la méthode décrite au Tableau 3. L'étalonnage des trois prototypes ciblés ainsi que le tableau d'importance ont permis de trouver le système d'arrosage le plus performant. Les membres de l'équipe souhaitent que ce produit puisse satisfaire et même dépasser ses attentes du client. En conclusion, notre objectif est de concevoir un produit compétitif au Gardena avec nos spécifications cibles établies

1.3 - Concepts préliminaires

Plusieurs concepts ont été générés pour différents sous-systèmes. Des exemples sont le système d'arrosage automatique, de détection d'humidité, d'entreposage d'eau, d'alimentation énergétique, de communication entre les systèmes, de détection de problèmes techniques, des systèmes de transportation de l'eau et même un système pour l'interface web. C'est sous-système ont été regroupés pour former les produits fonctionnels 1, 2 et 3. Une méthode d'analyse permet de conclure que le produit 3 est le plus performant, le plus compétitif et répond le plus aux attentes du client.

1.3.1 Concepts de sous-systèmes

1.3.1.1 Concepts de Danika

Système de détection d'humidité: Pour mon système de détection d'humidité, j'ai décidé d'utiliser la pesanteur de la terre pour trouver le pourcentage d'eau de celle-ci. Le contenant de terre et plantes reposerait sur une balance. On peut donc trouver la masse de la terre en soustrayant le poids du contenant et des plantes au poid total. Si la terre sèche as une masse volumique de 700 kg/m^3 et que le volume de terre est de V , alors le pourcentage d'humidité de la terre est représenté par:

$$\frac{(\text{Masse de terre}) - (700 \text{ kg/m}^3)(V)}{(700 \text{ kg/m}^3)(V)} \cdot 100$$

Système de communication: Pour communiquer entre les différentes composantes du système, on pourrait utiliser un arduino. L'Objectif serait que l'arduino connecté à la balance fasse le calcul du pourcentage d'humidité de la terre. Si l'humidité détectée est inférieure à au

pourcentage voulu, un signal de sortie est envoyé au système d'arrosage. L'arduino, la balance et la pompe à eau seraient alimenté par une pile rechargeable comme ceux de nos téléphones portables.

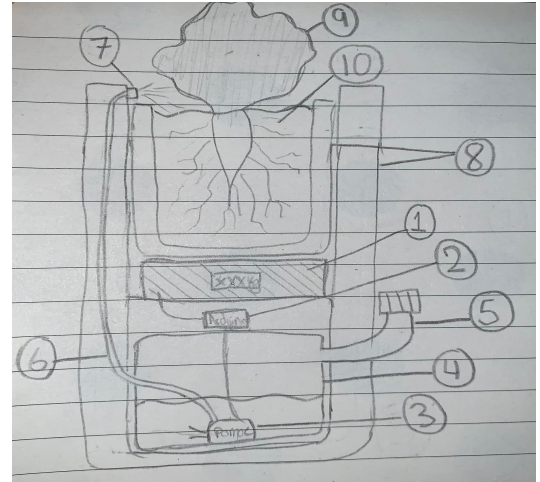


Image A

Système de d'arrosage automatique: La pompe d'eau serait aussi connectée au système de communication. Lorsque la pompe d'eau reçoit un signal du arduino, elle pompe l'eau du réservoir par le tuyau de transportation jusqu'au bouchon de sortie. Le bouchon de sortie permet à l'eau de sortir en forme de petite particule comme l'image suivante.

Lorsque la terre atteint le pourcentage d'humidité voulu, la balance envoie le signal au arduino qui fait ensuite arrêter la pompe.

Image B



1	Balance (Digital Body Weight) 20\$	6	Tuyau de transportation de l'eau 6\$
2	Arduino	7	Bouchon des sortie

3	Pompe d'eau (Mini pompe à eau submersible) 8\$	8	Pot extérieur et pot intérieur 20\$ - 40\$
4	Réservoir d'eau (water container) 5\$ - 20\$ dépendamment du type	9	Plante 0\$
5	Ouverture pour le remplissage 0\$-5\$	10	Terre 10\$

Tableau 4 - Composantes des sous-systèmes de Danika

@@

Système de stockage d'eau: L'eau sera stockée dans un réservoir. Connecter l'appareil à une source d'eau diminuerait l'entretien du produit mais ferait en sorte qu'il perdrait sa mobilité, ce qui ne répondrait pas aux attentes du client. L'emplacement et la forme du réservoir varient selon la forme du produit, le type d'arrosage et le type de système de communication. Bien sûr, plus le volume du réservoir est grand, moins la jardinière aura besoin d'entretien. Quelques exemples sont présentés dans la figure suivante.



Image C

1.3.1.2 Concepts de Kassoum

Sous-système de détection et avertissement en cas de panne general :

L'objectif du client est d'obtenir un arroseur automatique doté de nouvelles fonctions qui lui permettront de prendre soin de son jardin avec plus de facilité. Mais à quoi servirait un système automatique si celui-ci n'est pas en mesure de détecter ses pannes et d'aviser le personnel compétent pour un dépannage imminent? C'est dans ce cadre que je propose de doter notre arroseur automatique de capteur qui vérifieront son fonctionnement normal. L'automate programmable ou l'arduino sera équipé de lignes de codes qui lui permettront de vérifier ceci. Le processus se déroulera comme suite:

A la réception de valeurs insuffisantes concernant le pourcentage d'humidité d'une portion de terre, l'automate va commander l'arrosage de cette portion de terre. Des lignes de codes seront rajoutées afin de vérifier s'il y a changement au niveau de la valeur du pourcentage d'humidité qui a entraîné l'arrosage de la portion de terre par notre dispositif. Ainsi chaque 1 minute suivant la commande d'arrosage d'une portion de terre fixe par l'arduino, celui-ci va vérifier si la valeur insuffisante du pourcentage d'humidité est inchangée. Cette vérification se fera trois fois de suite. Si elle reste inchangée, l'arduino ordonnera à son module GSM connecté d'envoyer un message au personnel qualifié pour un dépannage imminent et d'émettre un signal sonore grâce à un buzzer. Sinon si la valeur est changée, alors il ne se passera rien. Ces lignes de codes usant du arduino et du capteur d'humidité permettront de détecter n'importe quel type de panne fonctionnement. Qu'il soit dû aux capteurs d'humidité ou à la pompe.

Sous-système de Spray :

Le Sous Système de Spray a pour but de veiller à la bonne santé des plantes. A quoi servirait un système automate qui s'occupe de l'arrosage des plantes si celui-ci ne peut s'occuper de leur défense contre les insectes. Une pochette sera attachée à notre système. À l'intérieur, on y trouvera du gaz à vaporiser. Ce gaz luttera contre l'envahissement du Jardin par les insectes. Afin de rendre le système entièrement automatique, la vaporisation aussi se fera tout seul chaque 72 heures. Alors, lors de la mise en marche de notre système, l'arduino ordonnera à la pompe à l'intérieur de la poche de gaz d'insecticide de vaporiser les plantes durant 3 minutes. Je juge cet intervalle de temps suffisant à vaporiser toute la superficie desservie par notre prototype. Après cette première vaporisation, l'arduino va compter 72 heures avant de vaporiser pour une seconde fois. Soit un delay 259.200.000 millisecondes. Le système se comportera ainsi alors jusqu'à son arrêt total par le superviseur. Il protégera les plantes en les vaporisant pendant 3 minutes toutes les 72 heures. Ces lignes de code se trouveront dans une boucle infinie.

Sous-système 3: Système d'automatisation et de Contrôle à distance

La partie d'automatisation sera divisée en deux parties: l'automatisation à partir d'une interface WEB avec contrôle et l'automatisation simple

Automatisation simple:

Dans cette partie les données recueillies par notre capteur d'humidité seront fréquemment envoyées à un NODE MCU qui assurera la communication sans fil avec l'arduino. Mais le NODE MCU

et le capteur d'humidité sont quant à eux reliés entre eux. Ainsi l'information recueillie par le NODE MCU sera envoyée à l'arduino qui sera chargé de la traiter. Chaque information sur le pourcentage d'humidité correspond à une portion de terre précise et dans le système le pourcentage d'humidité pour que les plantes de cette portion de terre grandissent dans les meilleures conditions y sont spécifiés. Le arduino alors après réception des informations va vérifier si le pourcentage reçu par portion de terre correspond au pourcentage entre lors de la configuration. Si OUI, il ne se passera rien. Sinon, il va commander l'arrosage à la portion de terre spécifique en envoyant une tension de 5V vers le PIN de la pompe correspondant à la portion de terre cible . l'arrosage se fera parallèlement à la mesure du pourcentage d'humidité . lorsque le pourcentage recherché sera atteint, l'arduino va par la suite envoyer 0V et la pompe va s'arrêter. Cette manière de fonctionner est préférable car elle évite le gaspillage d'eau et aussi chaque plante sera arrosée selon son besoin en eau afin qu'elle grandisse dans les bonnes conditions, car nous le savons, le manque ou l'excès tue.

Automatisation de l'interface:

La deuxième partie est l'automatisation à partir de l'interface WEB . Ici nous allons relier une interface WEB a notre Systeme . Ainsi le NODE MCU lorsqu'il recueille les informations sur le pourcentage d'humidité des différentes plantes va les transmettre à un Arduino puis à un serveur Raspberry PI. C'est le raspberry qui sera chargé de l'hébergement de notre application WEB. nous y aurons accès à un résumé du pourcentage d'humidité de toutes les portions de terre ou ont été plantées nos capteurs d'humidité et les responsables pourront ainsi vérifier l'état du jardin à chaque moment et ils pourront aussi actionner ou stopper l'arrosage de portions de terres spécifiques pour à leur guise à travers l'application Web même lorsqu'ils ne sont pas dans les environs.

Système d'alimentations:

Afin d'assurer une alimentation continue au système , je propose que celui-ci soit connecté au secteur. Il recevra ainsi une alimentation de 220 volts. Cette alimentation est supérieure à celle que doit normalement recevoir le arduino et la pompe.nous allons y remédier à travers un diviseur de tension pour s'assurer de délivrer 9 volts au arduino et 12 volts à la pompes.

1.3.1.3 Concepts de Wadie

Système de détection de l'humidité :

En ce qui concerne ce sous système , on peut utiliser un capteur d'humidité du sol de 12V provenant de XCSOURCE; un système comprenant la capacité de détecter une humidité inférieure à l'humidité définie et donc cela signifie que lorsque l'humidité du sol est inférieure à l'humidité définie, le relais tire automatiquement et commence à pulvériser de l'eau. La pompe effectue l'humidification jusqu'à ce que le réglage d'humidité soit automatiquement arrêté. Il peut contrôler une variété d'humidité du sol et le seuil correspondant peut être contrôlé par le potentiomètre.(type de résistance). Le produit est compatible avec arduino et donc peut être lié aux autres composantes du systèmes d'arrosage .Cela dit le produit dispose d'une fonction de délai avec un délai de 3 à 5 secondes, c'est à dire que lorsque l'humidité est détectée comme étant critique, le relais ne clignote pas et ne scintille pas.



Image D

Système de récupérateur d'eau de pluie:

Le récupérateur d'eau de pluie permet de stocker les eaux pluviales. En fonction des usages, par exemple pour l'arrosage des plantes, la capacité d'un récupérateur peut aller de 100 à 150 litres selon les modèles. La pluie qui tombe sur le toit de la maison est canalisée par les gouttières, qui sont directement reliées à la cuve grâce à un collecteur d'eau. Il s'agit d'installer un collecteur dans la descente, qui fera la connexion entre celle-ci et votre réservoir. Il vaut mieux opter pour un collecteur muni d'un filtre, afin de récupérer une eau la plus propre possible. En ce qui concerne l'opération d'alimentation en eau du réservoir, on utilisera une pompe électrique qui permet d'acheminer l'eau du récupérateur directement dans la jardinière sans effort.

Système d'arrosage :

Nous pouvons opter pour un arrosage rotatif à jet réglable, qui présente des avantages comme son bas prix (environ 25 dollars), il peut couvrir de grandes surfaces allant jusqu'à 20 m de diamètre, et en plus il permet d'effectuer des économies d'eau par la possibilité de régler le jet d'eau en débit et en portée, il existe aussi des modèles s'inscrivant dans toutes ces caractéristiques très facile à porter (ex: GARDENA MAMBO COMFORT). De plus, on le branche directement sur un tuyau d'arrosage ordinaire. Très simple d'utilisation, il est doté de deux ou trois bras de forme horizontale, peut être aussi posés sur le sol. Il est aussi conçu en plastique (8 dollars) ou en métal inoxydable et donc plus résistant et plus coûteux (58 dollars).

1.3.1.4 Concepts de Lina

Système de détection d'humidité :

On pourrait utiliser un arduino ou une automate programmable afin de permettre une communication avec le capteur d'humidité : Si le capteur détecte une valeur inférieure à la valeur moyenne, un signal de sortie est envoyé à la pompe d'eau.

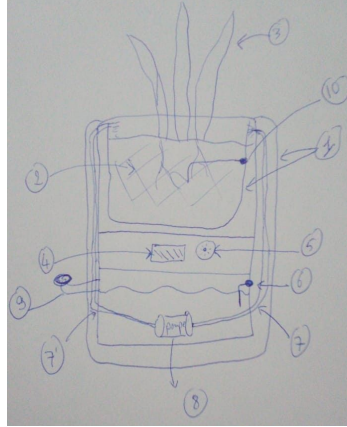


Image E

Système de détection du niveau de l'eau dans le réservoir et système d'alarme :

On pourrait utiliser une automate programmable (un arduino) afin de permettre une communication avec le capteur 2 de niveau de l'eau dans le réservoir : si le capteur 2 détecte un niveau d'eau assez bas, une led rouge à la surface du jardinière commencera à clignoter ou une alarme sonore .

Système d'arrosage automatique:

Le composant primordial pour cette fonction est la pompe d'eau qui sera connectée à l'API (automate progr

ammable industriel). Une fois que la pompe d'eau reçoit un signal, elle pompe une quantité d'eau bien définie du réservoir par 2 tuyaux de transportation jusqu'aux bouchons de sortie .

Système de d'arrosage en goutte à goutte:

Ceci est un principe simple et très connu, en aura besoin d'un tuyau troué , une pompe si le réservoir se situe au dessous du sol , et si le réservoir est au dessus nous pouvons utiliser qu'une électrovanne qui arrêtera le déroulement de l'eau une fois que le capteur détecte que le sol est assez humide.

Système de communication avec l'utilisateur :

Un petit écran à la surface de la jardinière permettant à l'utilisateur d'avoir des informations sur le niveau actuel de l'eau dans le réservoir et l'humidité du sol .

1	Pots extérieure et intérieure	6	Capteur de niveau d'eau (Capteur choisis) 20€73
2	sol	7	Tuyaux
3	Plante	8	Pompe (Pompe choisis) 9€99
4	API	9	Bouchon de remplissage
5	Led Rouge	10	Capteur d'humidité (Capteur choisi) 3,49 €

Tableau 5 - Composantes des sous-systèmes de lina

Système d'alimentation à base de pile

L'alimentation à l'aide des piles de 4 volts peut être utilisée afin d'assurer la mobilité du système. En utilisant ces piles nous allons permettre au arduino et à la pompe d'être à disposition de la bonne plage de tension permettant leur bon fonctionnement.

1.3.2 Concepts des systèmes complets

1.3.2.1 Le système fonctionnel # 1

Le système d'arrosage numéro un que nous pouvons réaliser en combinant nos différentes idées est celui-ci. Il pourrait être selon la forme 3.1 des formes d'arroseurs automatiques définis plus haut. Il devrait être muni d'un système de détection de niveau d'eau. Le capteur à flotteur sera utilisé afin de déterminer le niveau d'eau du réservoir et lorsque celui-ci aura atteint sa valeur critique, une alerte sera émise par le système pour alerter le superviseur. Le système sera commandé par un arduino avec fil qui recevra en entrée les différentes données des capteurs d'humidité et des capteurs de niveau d'eau et enverra en sortie des signaux électriques vers la pompe à eau ou l'alarme de signalisation de la baisse critique de niveau d'eau. La détection du pourcentage d'humidité du sol sera assurée par le capteur de détection d'humidité d'arduino qui est un capteur capacitif moins chère mais a une précision faible. Quand à l'alimentation du système en eau, il sera assuré par notre système d'entreposage de l'eau de pluie. Des tuyaux installés à travers toute l'université serviront à véhiculer l'eau de pluie vers le réservoir de notre prototype et ainsi le client pourra économiser sur la consommation en eau. Cette eau sera pompée par une pompe de 9 volts vers des tuyaux minces munies de spray au bout pour assurer l'arrosage du jardin.

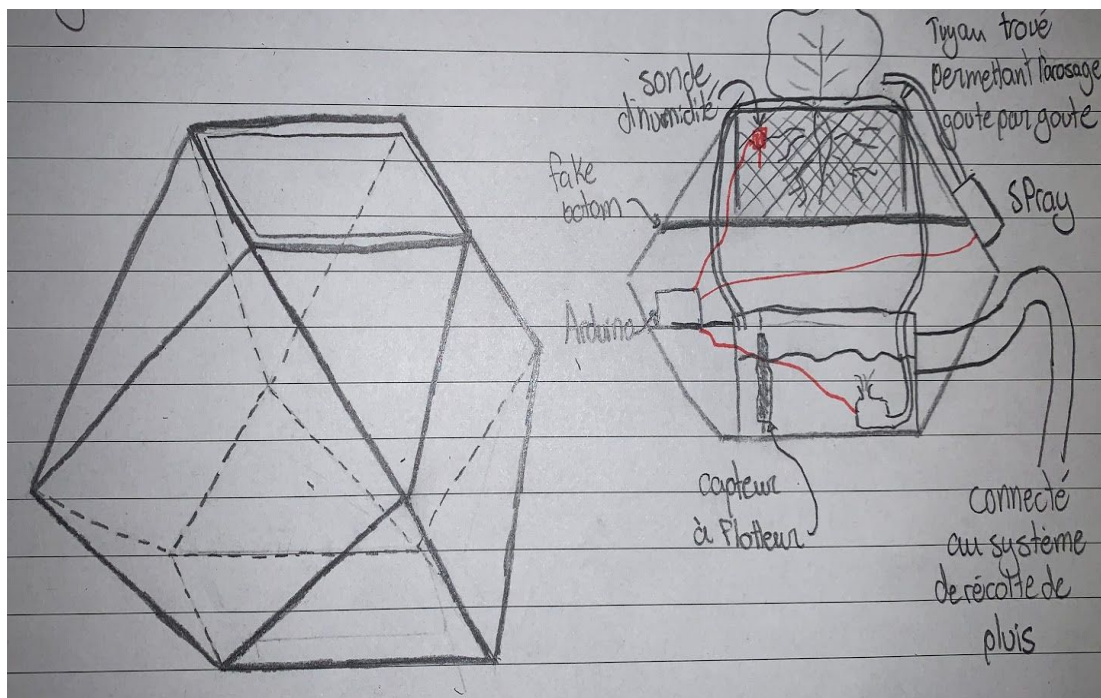


Image F

1.3.2.2 Système fonctionnel # 2

Le deuxième prototype qui pourrait être conçu à travers nos idées ci-dessus pourrait prendre la forme 2.2 dessiné plus haut. Il serait aussi muni d'un système d'entreposage d'eau de pluie comme le premier ainsi que d'une pompe de 9 volts pour pomper l'eau vers le tuyau d'arrosage à spray.

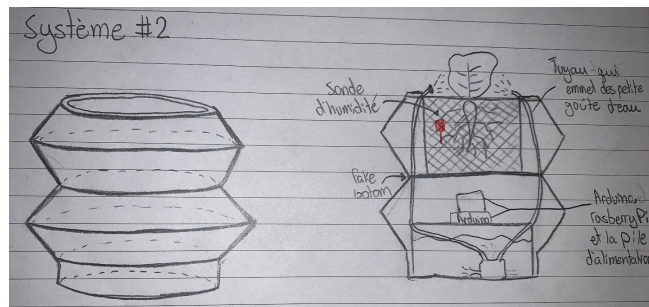


Image G

L'arrosage à spray permettrait de couvrir tout le jardin car ils seraient placés des deux côtés du jardin et pourraient le couvrir en entier. Le pourcentage d'humidité serait détecté à l'aide du même capteur d'humidité arduino que le premier. Mais la différence se trouve sur le système de commande sans fil et l'interface web que présenterait ce prototype. Le système de commande sans fil géré par l'arduino et le NODE MCU nous permettront d'avoir une allure futuriste et d'éviter l'usage des fils qui pourraient créer des incidents. L'interface WEB quand à lui nous permettrait d'obtenir les informations concernant le pourcentage d'humidité de chaque portion de terre à distance et le commandement aussi de l'arrosage des plantes pourraient être fait à distance par le superviseur si celui-ci juge que le pourcentage d'humidité des portions de terre contenant ces plantes est suffisant.

1.3.2.3 Système fonctionnel # 3

Le système trois qui pourrait être conçu par la mise en phase de nos idées afin de répondre aux exigences du client pourrait présenter les caractéristiques suivantes. Il aurait contrairement aux deux autres un réservoir d'entreposage d'eau au fond remplissable par le superviseur. En fonction de la consommation journalière des plantes qui serait calculée et de la taille du réservoir, le superviseur aura un intervalle de temps après lequel il devra remplir le réservoir d'alimentation en eau de notre réservoir automatique. L'alimentation en énergie électrique serait assurée par deux piles de 3 piles de 4 volets ainsi

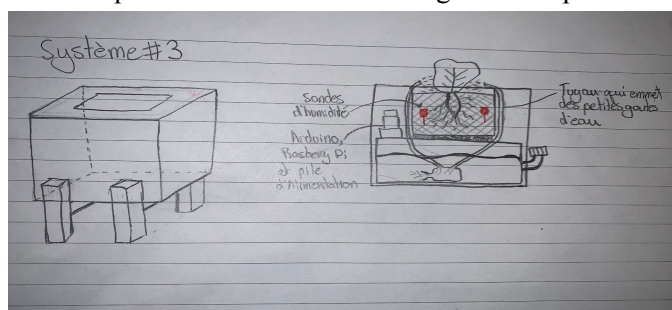


Image H

on aura plus a utiliser de diviseur de tension pour éviter la surtension causée par le secteur . Les 12 volts délivrés seront suffisants pour la mise en marche de la pompe et du arduino . sans oublier que l'utilisation de piles permettra à notre système d'être facilement mobile. Le dispositif aura la forme 1.1 qui a une forme futuriste et sera apte au déplacement . l'arrosage se fera par des tuyaux a spray connecte au tuyau de la pompe. Ainsi ils recevront l'eau provenant de la pompe et serviront correctement tout le jardin. Le pourcentage d'humidité dans ce cas sera déterminé à l'aide d'une sonde d'humidité du sol de 12V provenant de XCSOURCE. Il communiquera avec un système de commande sans fil doté d'une interface web. Le système de commande sans fil lorsqu'il recevrait des valeurs d'humidité inférieur aux valeurs prédéfinies dans les lignes de code du arduino, il enverra un signal électrique à la pompe qui pompera l'eau du réservoir vers les tuyaux a spray. Ainsi lorsque le capteur détectera un pourcentage d'humidité supérieur à celui défini dans les lignes de code du arduino, celui-ci ordonnera l'arrêt du pompage d'eau à la pompe.

L'interface web quand a lui servira au client de suivre l'évolution de son potager à distance et parallèlement commander son arrosage . Un système de détection de panne sera ajouté afin de rendre le système plus autonome et celui-ci pourrait alerter les superviseurs sur l'interface WEB ou à l'aide d'un message sur leur mobile.

1.3.3 Analyse et évaluation des systèmes

Critères	Importance	Système #1	Système #2	Système #3
Couts	5	1	2	3
Consommation énergétique	5	2	3	3
Niveau écologique	3	1	3	3
Facilité de transport	4	2	2	2
Autonomie et fréquence d'entretien	4	3	2	2
poids	4	3	3	3
taille	5	3	3	3
durabilite	4	3	2	2
Résistance de la matière de fabrication	4	2	2	2
Systeme d'arrosage	5	3	3	3
Type d'arrosage	4	1	2	2
Système de detection	5	3	3	3

d'humidite				
Type de système de détection d'humidité	4	1	2	3
Esthétique	3	3	2	2
Volume du réservoir d'eau	4	3	1	1
total	----	144	149	158

Tableau 6 - Analyse des trois systèmes fonctionnelles

1.3.4 Meilleur solution

La meilleure solution proposée est le système fonctionnelle #3 en raison de ses composantes efficaces. Celui-ci est économique en énergie et aussi est constitué de composants durables et précis comme la sonde de détection du niveau d'humidité qui est plus précis que le capteur d'humidité arduino. L'alimentation à l'aide des piles lui permet d'être mobile et il annule l'utilisation de diviseurs de tension pour éviter la surtension. Il utilise le réservoir d'eau qui favorise sa mobilité. le système trois a comme système de commande l' ARDUINO et le NODE MCU ,lui permettant d'être facilement maniable et de transférer les données du jardin vers une interface WEB rendant son control et sa commande plus facile a distance. Un système de détection de panne à défaut d'un capteur de niveau d'eau permettrait de détecter tous types de pannes qui empêcherait l'arrosage automatique du jardin et la bonne humidité du jardin

1.3.5 Leçons et conclusions

Pour conclure, toutes les idées de sous-systèmes des membres de l'équipe ont permis d'établir trois produits complexes. Une méthode d'analyse permet de conclure que le produit 3 est le plus performant, le plus compétitif et répond le plus aux attentes du client. Le meilleur système va probablement encore avoir besoin d'amélioration. L'entretien de celui-ci est moindre lorsque sa vient au réservoir d'eau tandis que l'entretien des piles pour alimenter le système d'interface, la Arduino et le Raspberry Pi peut augmenter les besoins en entretien. Le type de pile choisi va avoir un effet direct sur l'entretien. Le coût du système ne devrait pas être une contrainte mais il faut être prudent lors de l'achat des matériaux pour ne pas dépenser tout le budget sur certaines composantes.

1.4 Prototypes #1

Puisque qu'il existe plusieurs types d'objectifs lors du prototypage, le plan de nos tests de prototypage permet de clarifier l'objectif de chacun de nos tests. Les tests seront exécutés tout au long du livrable F. Les petits tests indépendants de chaque aspect des sous-systèmes vont nous permettre de réduire l'incertitude et les risques de mauvais fonctionnement global du produit. Il faut toujours commencer quelque part, donc les premiers prototypes seront des schémas de sous-système, des schémas de câblage, de constitution de base, des tests de base ou des représentations 3D à base de matériaux trouvés à la maison. Un tableau de plan de test des prototypages sera présenté pour expliquer le pourquoi, le quoi, le

comment et le quand de chaque test relié à chaque prototypes. Plus tard les tests et les prototypes seront présentés ainsi que la rétroaction du client et possiblement des utilisateurs de ceux-ci.

1.4.1 Tableau du plan de test des prototypes

N° de Test	Objectif du test(Pourquoi)	Description du prototype utilisé et de la méthode de test de base (Quoi)	Description des résultats à documenter et comment ces résultats seront utilisés (Comment)	Durée estimée du test et date prévue du début du test (Quand)
Tests relié à la structure - Danika				
1	Tester la faisabilité et la stabilité de la structure	Le prototype utilisé est le prototype 3D fait à base de carton, de pots de yaourts, de colle chaude et d'une vraie plante.	Les résultats seront positifs ou négatifs dépendant de la réussite du prototype et de sa stabilité. Ces critères définissent la faisabilité de la structure.	Le rassemblement des matériaux ainsi que la construction du prototype en carton peut prendre une journée ou deux maximum.
2	Trouver les dimensions nécessaire au produit final et a ses compartiments	Le prototype utilisé est le prototype 3D fait à base de carton. Les dimensions de celle-ci seront multiplié par un facteur commun pour conclure de dimensions réels	Les résultats seront des mesures approximatives et retranscrites en dessins. Les dimensions serviront à prédire les dimensions dans l'espace des autres composantes pour assurer que le produit final soit coordonné.	Le dessin qui inclut les mesures peut prendre une heure donc le tout sera d'une durée approximative de 2 heures.
3	Trouver le volume nécessaire de terre	Basé sur les dimension réel du compartiment du pot de terre, la quantité de terre peut être conclue.	Le volume de terre en m ³ est à documenter. Ces résultats seront utilisés pour remplir le pot avec la plante et la terre et aussi pour l'achat de la terre.	Quelque minutes puisque les résultats du volume du pot de terre sont definit par le test 2
4	Tester la quantité d'eau nécessaire par semaine pour une plante pathos	L'entretien de la jardinière va dépendre de l'utilisation en eau par semaine pour arroser une plante pathos. Une plante dans son pot de terre et un pichet d'eau seront utilisés.	Il va falloir documenter la quantité d'eau par semaine pour garder la plante en vie. Ces résultats seront utilisés pour estimer la durée d'autonomie de la jardinière en connaissant la capacité d'eau du réservoir.	Ce test pourrait durer plusieurs mois. Par contre, on peut le faire en une semaine en vérifiant les résultats d'une recherche internet de la quantité d'eau nécessaire à cette plante
Tests relié au câblage du concept de base - Kassoum				
5	test sur le branchement	À l'aide du logiciel proteus , un test sur le branchement entre le	Les résultats obtenues lors de nos test seront répertoriés et comparés à d'autres	Ces tests seront effectués au maximum 5 fois . Par contre, il

	(capteur d'humidité , node MCU,)	capteur d'humidité et le nodeMCU sera effectué afin de vérifier si ce schémas de câblage nous permettrait de capter le pourcentage d'humidité. C'est ainsi que nous allons y reproduire le même schémas effectué sur fritzing accompagné de certaines ligne de code et vérifier si le schémas répond à nos attentes	résultats expérimentaux tirés d'internet . de cette comparaison nous saurons ci nodeMCU et capteur d'humidité peuvent être couplés afin de prelever correctement le pourcentage d'humidite du sol.Au cas ou ces deux résultats seront en déphasages nous allons essayer de calibrer nos résultats expérimentaux à travers des équations que nous allons insérer dans les ligne de code du NODEMCU	faudra prendre en compte la possibilité d'une déconnexion entre les câbles lors de la réalisation. C'est ainsi que toutes les dispositions (soudage à l'étain) seront prises afin d'assurer la communication entre nos différents composants.
6	test d'envoi et réception d'informations ou commande (Nodemcu plate forme Web)	le nodeMCU peut jouer le rôle d'héberger d'une plateforme web , nous allons alors par la suite tester cette aptitude de notre nodeMCU réellement ainsi que celle d'affichage d'information sur la plateforme web ou de réception de commande à exécuter venant de la plateforme web	Les résultats obtenus lors de ces tests seront binaires . c'est-à- dire réussite ou échec. Les test seront effectués réellement et les bugs lors de la communication entre la plateforme web et le nodeMCU seront notés ainsi que les retards durant la communication et les actions.Après avoir identifiés les différents bugs, des recherches seront menées afin d'y trouver des solutions adéquates et rendre la communication entre les deux parties plus fluide .	Ces tests seront effectués plusieurs fois .Après le déploiement du système , des tests continueront à être effectués afin de rendre la communication entre la plateforme web et le nodemcu plus rapide ainsi que l'exécution des commandes.
7	test sur le branchement Nodemcu arduino	encore sur proteus, nous allons y répéter le schémas de câblage fritzing ci dessous entre l'arduino et le NODEMCU. Pour vérifier si celui-ci pourrait transmettre les informations recueillies à l'arduino .	Les résultats de ce test nous permettront de vérifier si le schemas de cablage est correcte et si les informations venant du nodeMCU sont correctement transmises au arduino sans être déformées par l'un des deux.	Ces tests seront effectués au maximum 5 fois . Par contre, il faudra prendre en compte la possibilité d'une déconnexion entre les câbles lors de la réalisation. C'est ainsi que toutes les dispositions (soudage à l'étain) seront prises afin d'assurer la communication entre nos différents composants.
8	Test sur le branchement	sur proteus à partir d'arduino , nous allons	L'arduino constitue ici notre centre de commandement .	Quant à ces tests, ils seront effectués

	Arduino , Buzzer, Pompe	essayer de commander un buzzer et une pompe pour vérifier le schéma de câblage	Ainsi, il enverra des impulsions électriques aux actionneurs tout au long du cycle. Les résultats des tests permettront de s'assurer que ces impulsions sont bien envoyées et les actions correspondantes correctement exécutées par les différents composants	plusieurs fois car l'arduino représente le centre de commande de notre système et tout bug doit être évité. C'est ainsi que plusieurs solutions seront explorées afin de ne pas surcharger l'arduino et permettre à celui-ci de commander aisément les actionneurs qui effectueront correctement leurs tâches
Tests reliés au pompage de l'eau - Lina				
9	Tester quel diamètre du tube sera nécessaire pour la force de pompage de la pompe	Un prototype de pompe réelle + des types de tuyaux seront utilisés. Le tuyau dont le diamètre est le plus conforme à la pompe sera choisi.	Les résultats seront en cm. Le diamètre du tuyau doit être un peu plus grand que celui de l'échappement de la pompe .	Ce test ne dure pas longtemps : 30 min sera assez suffisante pour le choix du diamètre convenable.
10	Tester la quantité d'eau pompée selon les différents matériaux des tuyaux	La pompe de l'eau et un chronomètre seront utilisés. La plus grande quantité d'eau pompée en 1 min avec différents tuyaux de matériaux différents sera calculée.	Les résultats seront en ml, le tuyau permettant le pompage de la plus grande quantité d'eau (suffisante et nécessaire pour la plante) durant 1 min sera choisi.	Ce test dure à peu près 15 min pour le calcul du volume d'eau transmis par chaque tuyau.
11	Tester les différentes formes du réservoir	La forme du réservoir doit être conforme à la forme du pot (qui est le contenant)... Et doit être capable de réserver la plus grande quantité d'eau possible . Ce test sera basé sur des mini et différents contenants en plastique en forme cylindrique, parallélépipède carré et rectangle droit .	Les résultats seront en m ³ (volume d'eau maximale que peut contenir chaque réservoir)+ Un schéma de la forme du réservoir choisi et ses dimensions (diamètre et hauteur pour le cylindrique /longueur, largeur et hauteur pour le parallélépipède) .	Ce test dure à peu près 30 min , puisque le calcul des dimensions réelles du réservoir doit être précis.
Tests reliés à l'interface web - Wadie				
12	Test de la communication des informations sur le taux d'humidité présent dans la structure	Les informations sur le taux d'humidité doivent être envoyées à la plateforme web . Ainsi après son design, celle-ci sera configurée de telle	Les résultats obtenus lors des différents tests seront tout d'abord sous une forme binaire . la réussite, ou l'échec à établir une communication entre le	Ces tests seront effectués plusieurs fois et continueront durant même l'exploitation du système car à travers les recherches, des

		<p>sorte qu'elle soit hébergée par le NODEMCU. A travers les différentes lignes de codes qui seront télévisées dans le nodeMCU , nous aurons pour tâche de tester la communication de ces différentes informations entre ces deux entités.</p>	<p>NODEMCU et la plateforme web ou sera affiche les informations concernant le pourcentage d'humidité de la terre.</p>	<p>fonctions pourraient etre utilises dans les lignes de code afin de ,moins surcharger les composants électriques et permettre à ceux-ci de travailler efficacement</p>
13	<p>Test de la rapidité de l'acquisition des informations</p>	<p>Après l'envoi de l'information, celle-ci doit être reçue et retranscrite en un langage compréhensible par les êtres humains . Le temps mis pour l'affichage de ces informations dépendra des lignes de codes qui s'assureront du traitement des informations. Ainsi, plusieurs procédés seront étudiés et testés afin de trouver le plus approprié</p>	<p>Ces résultats concernent l'efficience de la communication entre la plateforme web et le nodemcu. C'est-à- dire le temps mis à transmettre ces informations et l'exactitude de cette information . les test seront effectués et le temps de réponse sur la plateforme sera répertorié au fur et à mesure jusqu'à ce que nous arrivions à un interval d'affichage des données raisonnable</p>	<p>Ces tests aussi seront effectués à plusieurs reprises pour trouver des moyens plus efficaces de transcrire les signaux électriques en langage machine et du langage machine au langage humain</p>
14	<p>Test d'envoi de commandes au système de commande fixe (arduino)</p>	<p>Après réception des différentes informations, l'opérateur pourra envoyer des commandes à effectuer par le système de commande fixe. Des tests seront effectués afin de rendre l'envoi et l'exécution des commandes possible et dans un délai raisonnable.</p>	<p>Dans cette partie , les résultats vont aussi s'étaler sur deux parties. La première concerne la mise en place ou non d'une communication de l'interface web vers l'arduino qui sera suivi d'une réponse au niveau des actionneurs. La deuxième partie portera aussi sur l'efficience de cette communication. C'est à dire le temps mis pour recevoir et retranscrire les informations venant de la plateforme web en commande et sans oublier les résultats sur l'effectivité des actions qui seront effectuer</p>	<p>Étant toujours à la quête d'amélioration , nous allons effectuer ces test plusieurs fois afin de réussir plus efficacement à transcrire le langage machine en signaux électriques.</p>

Tableau 8 - Plan de test

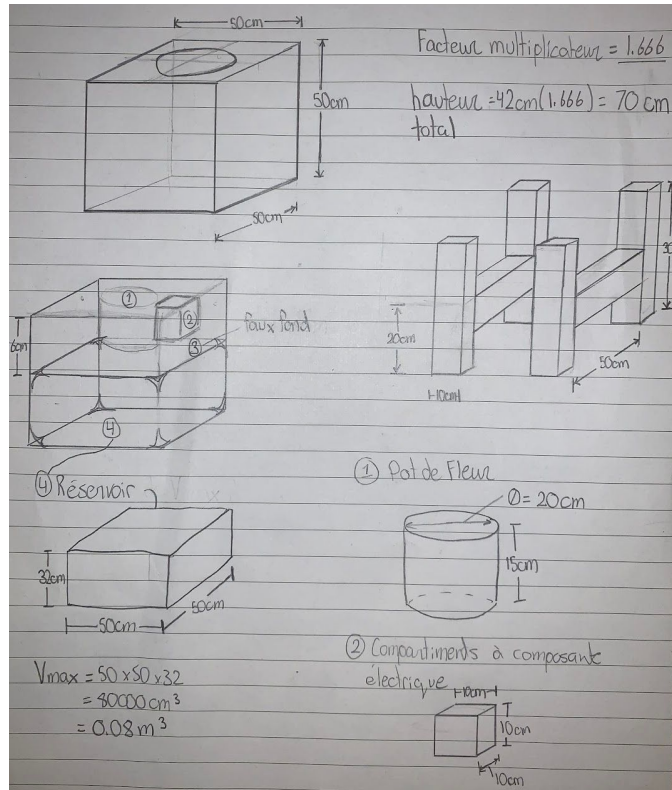
1.4.2 Présentation des tests et des prototypes

1.4.2.1 Représentation 3D du prototype à base de matériaux de maison - Danika

Le **premier test** avait pour but de tester la faisabilité et la stabilité de la structure. Lors de la construction du prototype, il était impossible de prévoir la coordination des différentes composantes comme le réservoir et l'emplacement du bouchon de remplissage. Mon prototype ne passe pas le test de l'esthétique mais il permet de conclure positif sur la faisabilité ainsi que sur la stabilité de la jardinière. Le prototype est stable, le centre de gravité est centré et les pattes sont au même niveau. Ce sont toutes des caractéristiques que va devoir avoir notre produit final.

Présentation: Le prototype est un boîtier avec 4 pattes formé de pots de yaourt. Un réservoir avec un bouchon d'entrée y est inclus, par contre celui-ci ne maximise pas l'espace. Dans le boîtier est inséré deux pailles, normalement connectées à une pompe, qui émergent du réservoir jusqu'à la plante. Il y a un faux fond avec des trous pour le passage des deux tuyaux ainsi qu'un compartiments pour les composantes électroniques.





Images I

Le deuxième test avait comme objectif d'approximer les dimensions nécessaires au produit final et à ses compartiments. Les dimensions du prototype du test 1 seront multipliées par un facteur commun pour conclure de dimensions réelles. Ces mesures incluent les dimensions du boîtier, le volume du pot de terre ainsi que le volume du réservoir. Ces mesures vont pouvoir être utilisées lors de la conception de la jardinière sur SolidWorks ainsi que pour assurer la coordination du produit final.

Le troisième test avait comme objectif de trouver le volume nécessaire de terre en fonction des dimensions approximatives établies au deuxième test. Le pot inclus de la jardinière avait comme diamètre 20cm et comme hauteur 15 cm. On peut alors conclure que le volume du pot est:

$$V = 0.004712 \text{ m}^3 \\ = 4.712 \text{ L}$$

Le volume de terre doit toujours être inférieur au volume du pot pour éviter que l'eau déborde alors un volume approximatif serait:

$$V = 4.2 \text{ L de terre}$$

Le prix pour 5.4L de terres au Dollarama est de 1.25\$ seulement!

Le quatrième test avait pour but de trouver la quantité d'eau nécessaire par semaine pour une plante pathos. Ce type de plante a été suggérée par le client puisqu'elle nécessite peu d'entretien. Sur le site de [La Résidence](#), par Léon et George, on suscite que pour maximiser la santé d'une plante pathos côté arrosage, il faut:

“Laissez sécher votre Pothos entre chaque arrosage. Arroser abondamment lorsque la terre est sèche, habituellement une fois par semaine. Ne laissez pas d'eau stagner au fond du cache-pot. [...] Notez que la fréquence d'arrosage doit être moins fréquente si la plante n'a pas beaucoup de lumière.”

Ceci signifie que le taux d'humidité nécessaire pour signaler le produit de procéder à l'arrosage doit être très faible, de 0 à 5%. Par contre, les arrosages doivent être abondants. Avec mes observations, la plante pathos que j'appartiens est en bonne santé et continue de grandir ce avec un arrosage abondant à chaque semaine. Les plantes sur le campus tout comme celles chez moi sont intérieures donc nécessitent moins d'eau en raison de la faible quantité de soleil. Mon approximation serait un arrosage au semaine. L'idéal serait de ne pas laisser d'eau stagner au fond du pot et aussi d'avoir la certitude que l'humidité obtenue par la sonde soit représentative. Ce qui réglerait ces deux inconvénient serait d'avoir un filtre au fond du pot de terre et que l'excès d'eau soit évacué et retourné dans le réservoir. Cela maximise l'utilisation de l'eau et minimise l'entretien de la jardinière. Ceci sera inclus dans le prochain prototype. Ce test était supposé calculer l'utilisation en eau par semaine consommer par une pathos. On connaît la durée de cycle d'arrosage mais si on inclut un pot/filtre, il est plutôt question de trouver le maximum d'eau qui peut être retenue par la sol. Dans la recherche, [le sol est une éponge](#), “la masse de terre humide vaut 100 g et la masse de terre sèche 90 g, la teneur absolue du sol en eau sera alors de 10 g.” Donc selon mes calculs:

100g terre humide
90g terre sèche = 10g d'eau
100g - 90g = 10g d'eau
10g / 1000g/L = 0.01L
0.01L * 100 = 1L

Image J

La teneur en eau d'un sol saturé d'eau est de 0.125 fois le volume de terre, V.

Pour conclure, le taux d'humidité pour déclencher l'arrosage est de 0% à 5%. Pour un arrosage abondant hebdomadaire qui sature 4.2L de sol en eau, la consommation d'eau est d'environ 0.525L par semaine. Si la capacité d'eau du réservoir est de 0.08m³ soit 80 L alors la durée d'autonomie de la jardinière est de 152.38 semaine soit 2.93 années. Il est donc possible de conclure que pour notre produit, le remplissage d'eau ne va pas être un facteur qui augmente son entretien.

1.4.2.2 Schémas de Cablage du concept de base - Kassoum

Le schéma électrique ci-dessous représente le branchement expérimental de notre système. Il fut réalisé à l'aide du logiciel fritzing après plusieurs recherches sur la connexion adéquate entre les différents composants de notre système. Le nodeMCU de forme rectangulaire à gauche se chargera de récolter les informations sur le pourcentage d'humidité du sol. Ces informations seront transmises par le capteur d'humidité en rouge. Le nodeMCU va par la suite effectuer la communication de ces informations au centre de commande portable (plate-forme WEB) et au centre de commande fixe (arduino) .

Le nodeMCU sera à nouveau sollicitée pour héberger la plateforme web ou sera afficher en temps réel les pourcentages d'humidité du sol . Des onglet y seront designe afin de pouvoir commander l'action ou l'inaction de la pompe en jaune. Afin de pouvoir effectuer ces commandes venant de la plateforme web, le branchement fut fait de tel sorte que lorsque l'action ou l'inaction du moteur est demandée , le nodeMCU envoi l'impulsion électrique correspondante au Arduino en bleu. Celui-ci est le centre de commande fixe. Il reçoit aussi les informations sur le pourcentage d'humidité du sol. Lorsque le pourcentage est inférieur au seuil, il envoie un signal électrique à la pompe en jaune pour que celle-ci assure l'arrosage de la portion de terre. L'arrosage de la portion de terre continue à être effectué jusqu'à ce que le pourcentage d'humidité que l'arduino lira atteint le seuil d'arrêt d'arrosage intégré dans les lignes de codes.

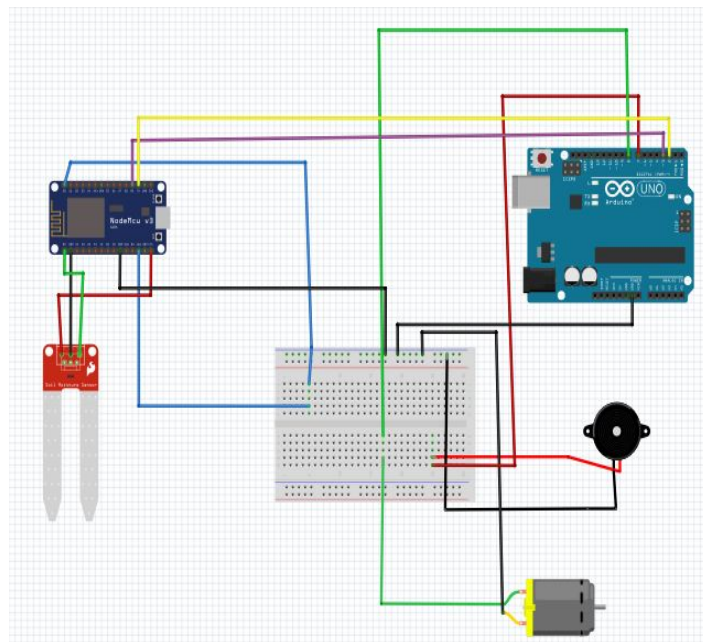


Image K

Le composant en noir est un buzzer qui sera charge de donner des alertes de non fonctionnement du systeme. Lorsque le pourcentage d'humidité reçu est inférieur au seuil normal(différent du seuil d'arrêt d'arrosage) , l'arduino enverra un signal pour assurer l'arrosage. Un test sera ainsi effectué toutes les deux minutes ,à cinq reprises, afin de vérifier si le pourcentage d'humidité reste inchangé. Si celui-ci reste inchangé, alors un signal électrique sera envoyé au buzzer afin qu'il alerte le responsable d'une possible défaillance au niveau des composants électroniques ou d'un manque d'eau au niveau du réservoir .

1.4.2.3 Schémas de Pompage de l'eau - Lina

Le schéma de pompage réel se compose de : une pompe +moteur+ un tuyau(avec ou sans spray)+ un réservoir avec bouchon de remplissage +brides de fixations. En effet, le moteur électrique entraîne l'axe de la pompe sur lequel est logée la roue. L'eau qui entre dans la roue axialement pas la bride d'aspiration et le goulot d'aspiration est déviée par les aubes de la roue dans un mouvement radial .les

forces centrifuges qui affectent chaque particule de fluide provoque une augmentation de la vitesse et de la pression lorsque l'eau s'écoule au travers de la zone des aubes.

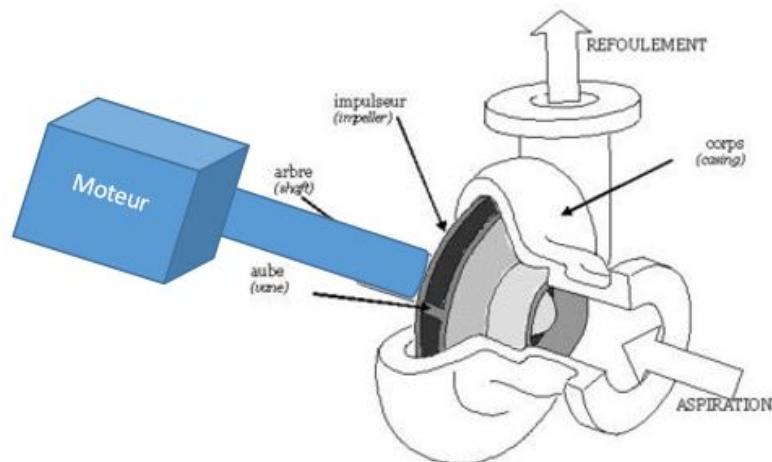
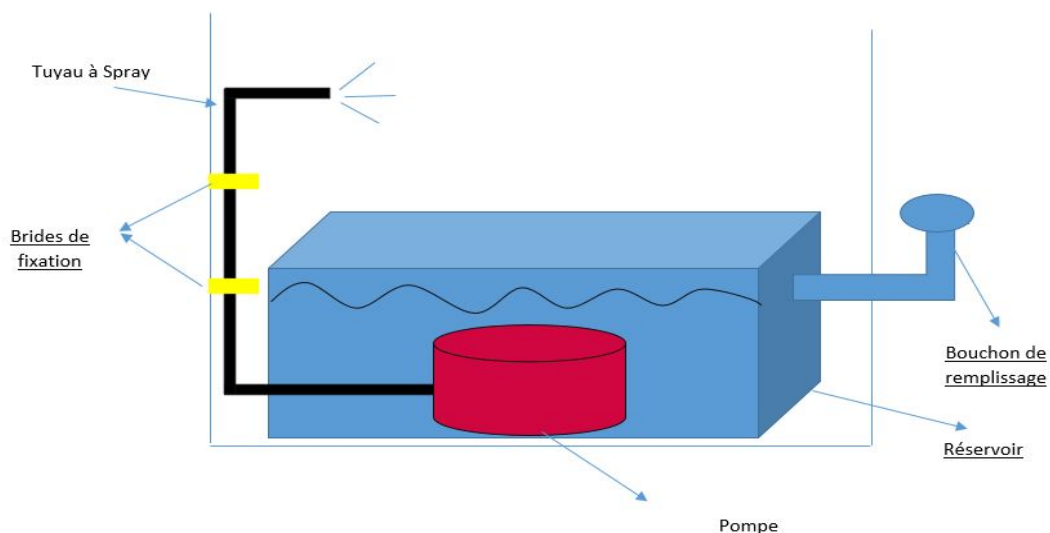


Image L

Schéma d'une pompe motorisée centrifuge - Source Wikipédia

En réalité l'intégration de ce sous-système au sein du système globale va être comme ceci :



Image

NB: la pompe sera câblée et donc commandée par un arduino qui lui aussi exécute les commandes de l'interface web.

En effet pour effectuer le prototype premier de ce sous-système on utilisera une pompe , un tuyau, un réservoir, et de l'eau :



Ce prototype nous permet l'évaluation du fonctionnement d'une vraie pompe au sein d'un réservoir (pour le réservoir j'ai juste utilisé un contenant d'eau normal) ainsi que la transmission de l'eau par le tuyau et l'échappement de l'eau.

Les objectifs primordiaux de cet essai sont le test du fonctionnement de la pompe et la transmission de l'eau par le tuyau dont le diamètre doit être conforme à celle de l'échappement de la pompe afin d'éviter l'écoulement de l'eau et garantir une transmission complète et bonne, ainsi que le choix du réservoir qui pourra nous procurer assez d'eau tout en contenant la pompe submersible. Une fois que ces buts sont atteints, l'essai du prototype sera achevé. D'autre part, ce prototype nous permet aussi le test du bruit de la pompe car nous ne voulons surtout pas que la pompe cause un énorme bruit.

Image M

1.4.2.4 Representation de l'interface WEB - Wadie

L'interface web se présente comme ci-dessous.

- les pourcentage d'humidité recueillis à partir du nodemcu seront regroupés dans les tableaux en fonction du tableau correspondant à leur portion de terre.
- Les onglets on et off serviront à contrôler la pompe d'arrosage
- L'onglet login servira à entrer ces informations afin d'avoir accès à ma plateforme Image
- Enfin l'onglet help servira à obtenir des informations quand à l'utilisation de la plateforme web
-

GARDEN CONTROL



HUMIDITY %

Part1	Part2	Part3	Part4
%	%	%	%

Image N

1.4.3 Leçons et conclusions

Ce livrable repose sur le premier prototypage des sous systèmes qui compose notre conception de la jardinière . Danika a pris en charge la présentation en 3D du prototype en se servant des matériels de maisons (carton , gobelets...) ,ceci nous a permis la concrétisation de notre idée conceptuelle ainsi que d'effectuer plusieurs tests tels que préciser la quantité de sol nécessaire et la stabilité et les dimensions finales du produit. Elle a aussi fait de la recherche pour trouver les spécifications à la plante pathos. D'autre part, Kassoum s'est intéressé à la partie électronique, au câblage du concept de base. Il a généré des tests pour concrétiser les schémas de câblage, des capteurs, d'arduino incluant la pompe, le buzzer, ainsi que la communication avec l'interface web. Cette partie est assez délicate mais primordiale puisqu'elle est la base de commande et sans elle le système ne pourrait pas fonctionner. Du coup, les tests de Kassoum reposent plus sur le branchement de ces différents outils qui nous permettrons la commande de notre concept de base. Ensuite, Lina a prototyper le système de pompage en présentant, dans un schéma 2D, son intégration au sein du système. La représentation d'un système réel d'une pompe permet la compréhension de son fonctionnement et le test de l'écoulement de l'eau dans un tuyau permet de choisir le diamètre convenable à celui de l'échappement de la pompe. Enfin, Wadie nous a présenté l'interface web qui sera l'outil de communication entre l'utilisateur et le produit. Grâce à cette interface l'utilisateur pourra faire fonctionner la pompe en cliquant sur le bouton ON, et la faire arrêter en cliquant sur le bouton OFF. En plus, l'interface web pourra afficher toutes les informations concernant les grandeurs captées par les capteurs comme le taux d'humidité du sol capté. Tous ces prototypes diminuent de plus en plus l'incertitude concernant le produit final. Aucune rétroaction n'as été fournie puisque le client n'as pas pu nous rencontrer lors de la rencontre client #2. Heureusement, nous allons obtenir une rétroaction de sa part lors de la rencontre du 10 novembre.

1.5 Prototypes #2

Le livrable G est le prototype 2 de la jardinière. L'objectif du prototype 2 est de tester les systèmes d'avantage avant de commencer le prototypage final. Les objectifs de chaque test sont présentés au tableau du plan de test des prototypes. On y retrouve aussi la description des prototypes utilisés, la méthode de test de base, la description des résultats à documenter, comment ceux-ci vont pouvoir être utilisés et la durée estimée de chaque test. Dans ce livrable on présente aussi la conception du prototype sur le programme solidworks. Le code pour faire fonctionner le système de communication est présenté suivi de la conception du capteur de niveau d'eau et de l'interface WEB.

1.5.1 Tableau du plan de test des prototypes

N° de Test	Objectif du test(Pourquoi)	Description du prototype utilisé et de la méthode de test de base (Quoi)	Description des résultats à documenter et comment ces résultats seront utilisés (Comment)	Durée estimée du test et date prévue du début du test (Quand)
Tests relié à la structure - Danika				
1	Pour clarifier l'apparence de la jardinière, la grandeur des mesures, pour faciliter les explications au clients ainsi que pour faire la conception des petit détails	Nous avons fait une représentation de la jardinière sur solidwork	Les résultats visuels permettent une rétroaction plus spécifique sur les composantes de la jardinière.	La conception de ce prototype va prendre de trois à quatre jours.
2	Pour tester la méthode du pot sous un entonnoir pour accumuler l'eau et la renvoyer au réservoir.	Le prototype utilisé sera un pot de fleur extérieur avec deux couches de trous, une entonnoir, de la terre et de l'eau.	Les résultats seront Négatifs: si l'eau s'échappe partout et la terre ne filtre pas l'eau. Positifs: si la terre filtre l'eau et celle-ci tombe dans l'entonnoir et récupère dans un autre pot.	Le test et la préparation vont prendre une journée.
3	Pour tester la force due au poids du réservoir sur les pattes de bois.	Le prototype utilisé sera le prototype de la conception de la jardinière sur solidwork	Avec la fonction de simulation de forces appliquées.	Le temps d'Apprendre comment la simulation solidworks fonctionne, une heure ou deux.
Tests relié au code du concept de base - Kassoum				

4	Test d'hébergement de la plateforme web sur le nodeMCU à travers le code	Le code a été saisi sur l'IDE de l'arduino et celui-ci servira à héberger l'interface de commande WEB. après le design de celui-ci, et le téléchargement du code final sur le nodeMCU , nous allons effectuer des test de commande d'allumage ou d'inspection afin de vérifier l'exactitude du protocole de communication	Après avoir télécharger du code sur le nodemcu tout en y ajoutant l'adresse du serveur web les résultats des différents tests nous permettront de voir si la plateforme web s'affiche effectivement et si les données y sont transmises avec exactitude.	Le test se déroulera en deux temps . premièrement nous allons héberger la plateforme web sur le nodemcu et l'observer durant 1 mois pour s'assurer de son effectivité. Par la suite durant l'utilisation de la plateforme nous allons effectuer des tests fréquemment pour s'assurer du bon fonctionnement.
5	Test du code de contrôle de la pompe sur un breadboard avec un arduino	Le code de contrôle pompe arduino fut désigné sur l'IDE d'arduino. A travers un breadboard nous allons connecter l'arduino , la pompe et les alimenter afin de tester le système de commande	A travers le code et le branchement, le test du bon fonctionnement de la pompe sera effectué et nous déduirons a travers les résultats si le code traduit correctement les commandes souhaitées	Le test se déroule en une semaine au maximum, le temps de s'assurer de l'exactitude des commandes que effectuerons notre système en se basant sur nos lignes de code.
6	Test du code de connection capteur d'humidité nodemcu	Le code fut désigné sur l'IDE d'arduino et par la suite téléchargé sur le NODEMCU . Le test sera réalisé en temps réel. Nous allons les connecter sur un breadboard et par la suite plonger le capteur dans de l'eau afin de voir s'il y aura un changement du pourcentage d'humidité sur le serial monitor.	Les résultats que nous obtiendrons tout d'abord sur le serial monitor nous permettrons de nous renseigner sur l'exactitude de notre branchement et de notre code. Ainsi des modifications pourront y être apporté afin de rendre le procédé de communication entre les deux plus fluides	Le test se déroule tout d'abord en une semaine pour s'assurer de l'exactitude des valeurs renvoyées . par la suite des tests seront effectués mensuellement pour s'assurer du bon fonctionnement.
7	Test du code de la connection serial node mcu arduino uno	Un code afin d'assurer la connexion serial entre le nodeMCU et l'arduino fut désigné. Il a été téléchargé sur les deux en fonction de l'envoyeur et du receveur. Par la suite il seront branchés sur	Le test nous permettra de recueillir les valeurs du capteur d'humidité au niveau du arduino et de contrôler le pompage de l'eau en fonction de ces valeurs. Ainsi notre branchement nous permettras d'obtenir certains résultats quant à la	Le test se déroulera en une semaine. Étant une connection serial , il suffit uniquement de vérifier que le protocole de communication en se basant sur notre code est exact . Le reste des tests se déroulent au niveau du branchement.

		un breadboard afin de vérifier si l'envoi et la réception du pourcentage d'humidité est effective.	commande à distance et la commande fixe	
Tests relié au pompage de l'eau - Lina				
8	Tester la faisabilité du capteur de niveau d'eau	Ce prototype en 3D est basé sur un gobelet en plastique, une paille en plastique, et un fil d'étanchéité.	Les mesures seront en cm : une comparaison entre le niveau d'eau dans le gobelet et celui dans la paille. Comparé si les deux niveaux sont exactement égaux ou s'il existe un rapport entre eux.	Ce test dure à peu près 15min pour le calcul du rapport de niveau de l'eau entre le gobelet et la paille.
9	Tester l'efficacité des joints d'étanchéité : le fil d'étanchéité et le pistolet à cire.	Dans ce test on utilise un fil d'étanchéité, le pistolet à cire et bien évidemment le prototype en 3D.	On choisira à la fin le composant qui permet une très haute étanchéité. Et qui résistera aussi longtemps.	Ce test dure 30 minutes. 15 min pour le test du fil d'étanchéité , et 15 pour le pistolet à cire
Tests relié à l'interface web - Wadie				
10	Tester l'exactitude de la base de l'interface Web	Grace au langage informatique HTML , nous allons pouvoir réaliser la base de notre interface Web sans les fonctionnalités additionnelles	Le résultat attendu est l'ouverture d'une page HTML via un navigateur web ,contenant les informations codés sur le logiciel "sublimetext"	Ce test peut prendre 3 jours , dépend des lignes de code à traiter et à révéifier.
13	Tester la conformité du design de la page	Grace au langage informatique CSS relié au HTML nous allons pouvoir optimiser le design de la plateforme Web pour qu'il favorise la manipulation	Nous devons nous mettre à la place de l'utilisateur et prendre la position neutre , puis juger de si l'interface est assez agréable ou pas à manipuler	Ce test peut durer pendant quelques jour le temps de créer le design nécessaire et de se construire un avis définitif
14				

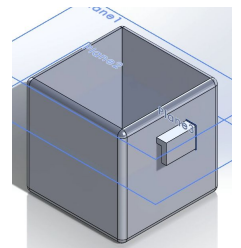
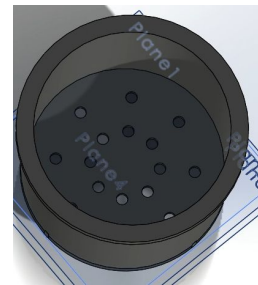
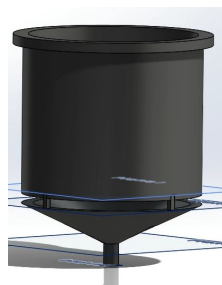
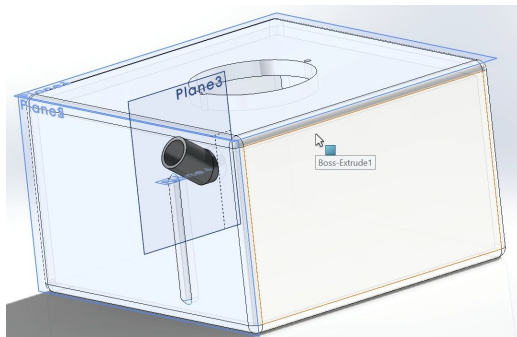
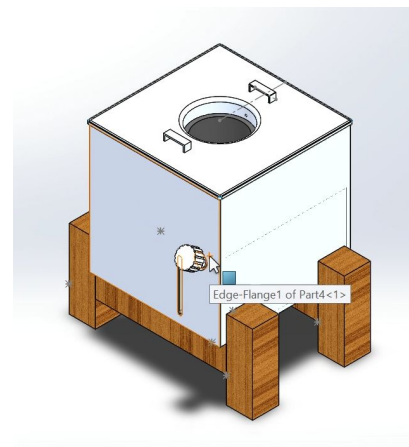
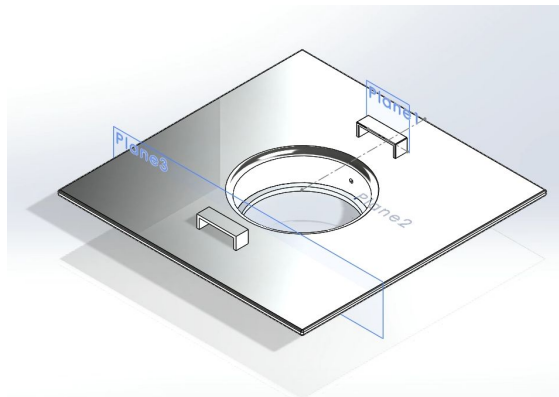
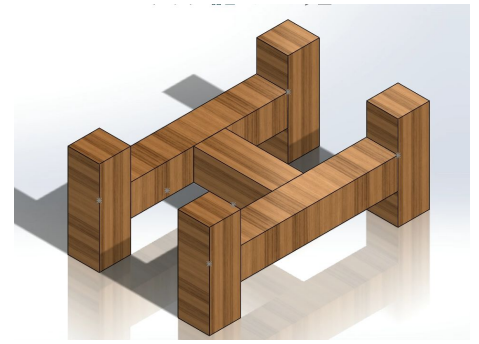
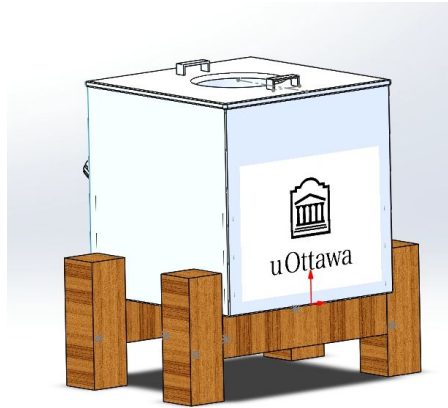
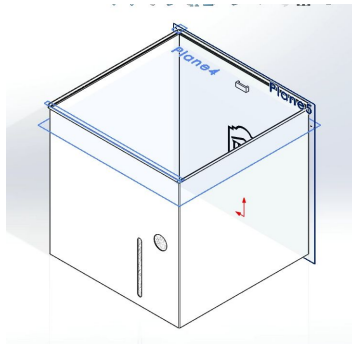
Tableau 9 - plan de test 2

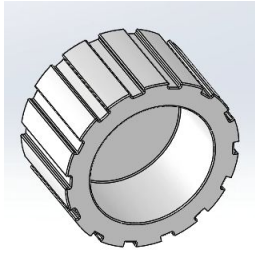
1.5.2 Présentation des tests et des prototypes

1.5.2.1 Représentation 3D du prototype à base de matériaux de maison - Danika

Le premier test était la conception de la jardinière sur solidworks. Il avait comme objectif de clarifier l'apparence de la jardinière, la grandeur des mesures, faciliter les explications au client ainsi que

faire la conception des petits détails. Ce prototype a permis l'établissement de plusieurs détails incluant l'entonnoir, les poignées pour soulever la pièce du dessus, les matériaux, l'emplacement du compartiment pour le système de communication, les pattes de bois, le logo de l'université et plusieurs autres détails importants.





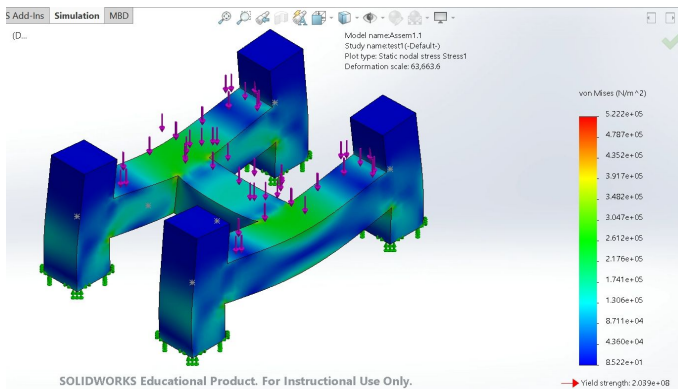
Images O

https://youtu.be/ckFvIV6C_Ao Voici lorsqu'on anime la reconstruction de la jardinière sur solidwork.

Le deuxième test avait comme objectif de tester la méthode du pot sous un entonnoir pour accumuler l'eau et la renvoyer au réservoir. Le prototype utilisé sera un pot de fleur extérieur avec deux couches de trous, une entonnoir, de la terre et de l'eau. Le prototype qui a été utilisé est un petit pot de fleur avec deux couches de trous, une entonnoir, de la terre et de l'eau. L'eau passe bien dans les petits trous et la terre reste belle et bien saturée d'eau. Une observation est qu'il y a eu une grosse accumulation d'eau au commencement. Donc ça prend un instant avant que l'eau traverse le sol. L'eau est tombée directement dans le petit pot donc les résultats de ce test étaient positifs.



Images P



Le troisième test avait comme objectif de tester la résistance des pattes de bois à l'aide du prototype de la conception de la jardinière sur solidwork avec la fonction de simulation solidwork. J'ai fixé les quatre pattes de l'assemblage et appliqué une force supérieure à la force approximative. Cette force est exercée sur les pièces où repose le boîtier de la jardinière. Avec la masse du boîtier et toutes les composantes à l'intérieur plus le réservoir d'eau

remplie, j'ai pu trouver un poids approximatif de 1000 Newton. Donc, 1000 newton est appliqué dans la simulation. La pièce n'a pas de rouge ce qui veut dire que la simulation ne démontre aucune fracture. C'est la preuve que notre jardinière est stable et forte.

<https://youtu.be/7f4rJSbS4JI> Voici la simulation de la force exercée sur les pattes de notre jardinière. Je crois par contre avoir augmenté l'élasticité du bois.

1.5.2.2 Schemas de Cablage du concept de base - Kassoum

Code communication serial entre le Arduino et le Node MCU. le nodeMCU sera chargé d'envoyer les différentes informations concernant le pourcentage d'humidité du sol à l'arduino pour que celui-ci commande le pompage de pompe.



```
code_arduino
String humidity;
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
}
void loop() { // run over and over
  if (Serial.available()) {
    humidity = Serial.read();
  }
  // allumage de la pompe a l'aide des valeur du capteur d'humidite
  if (humidity.toInt() <5||humidity = "LED1 on")
  {
    digitalWrite(10,HIGH);
  }
  if (humidity.toInt() >50|| humidity="LED1 off")
  {
    digitalWrite(10,LOW);
  }
}

*****
#include <SoftwareSerial.h>
int sense_Pin = 0; // sensor input at Analog pin A0
String str;
int value = 0;
void setup() {
  Serial.begin(115200);
  delay(2000);
}
void loop() {

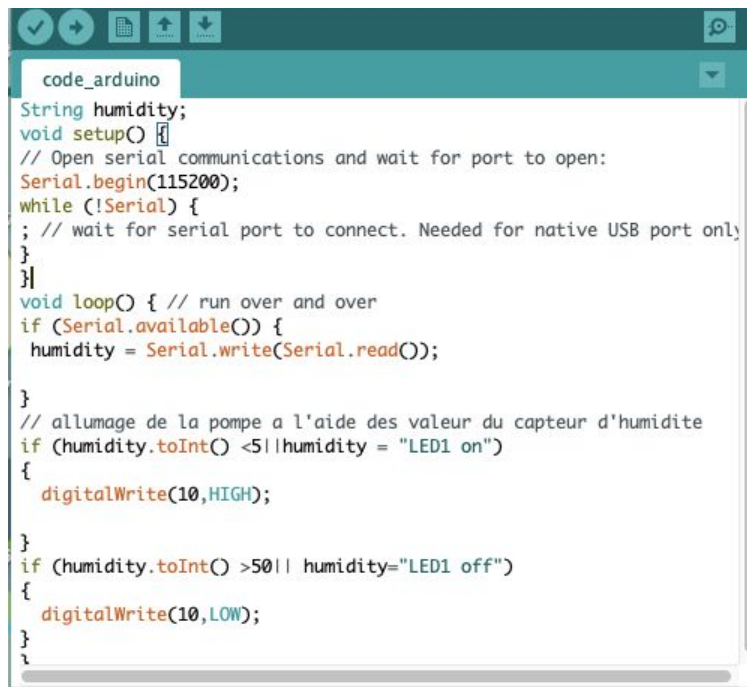
  Serial.print("MOISTURE LEVEL : ");
  value= analogRead(sense_Pin);
  value= value/10;
  Serial.println(value);
  Serial.println("la valeur du pourcentages d'humidite est de ");
  Serial.println(value);
```

```

Serial.println("%");
str =String("coming from ESP8266: ")+String("H= ")+String(value);
Serial.println(str);
delay(1000);
}

```

Code de la connexion entre le nodeMCU et l'arduino ainsi que celui de commande de la pompe. Ce code permettra au arduino de recevoir les informations venant du nodemcu . Par la suite, il commandera la pompe en fonction de ces résultats . Dans ce code nous avons aussi ajouter des lignes qui serviront à la commande à distance. Ainsi après avoir vérifié que humidity = LED ON ou LED OFF, il pourra commander l'allumage ou l'inspection à distance de la pompe.



```

code_arduino
String humidity;
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
}
void loop() { // run over and over
  if (Serial.available()) {
    humidity = Serial.read();
  }
  // allumage de la pompe a l'aide des valeur du capteur d'humidite
  if (humidity.toInt() <5||humidity = "LED1 on")
  {
    digitalWrite(10,HIGH);
  }
  if (humidity.toInt() >50|| humidity="LED1 off")
  {
    digitalWrite(10,LOW);
  }
}

```

```

String humidity;
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
}
void loop() { // run over and over
  if (Serial.available()) {
    humidity = Serial.read();
  }
}

```

```

}
// allumage de la pompe a l'aide des valeur du capteur d'humidite
if (humidity.toInt() <5||humidity = "LED1 on")
{
    digitalWrite(10,HIGH);

}

if (humidity.toInt() >50|| humidity="LED1 off")
{
    digitalWrite(10,LOW);
}
}
}
*****

```

Code d'hébergement de la plateforme web sur le mode MCU.

Ce code servira à utiliser le NODEMCU comme hébergeur, et celui-ci pourra ainsi recevoir un pourcentage d'humidité . Le maintenancier pourra par la suite en se connectant sur le même réseau que le NODEMCU envoyer des commandes au système pour allumer ou éteindre la pompe et il aura également accès aux pourcentages d'humidité. La commande à distance nécessite pour le moment une utilisation d'un réseau wifi identique. Une nouvelle connexion sera mise en place par la suite se basant sur un réseau MQTT tout en utilisant les brokers publiques MQTT.

```

kassoum
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

/*Put your SSID & Password*/
const char* ssid = "OUEDRAOGO"; // Enter SSID here
const char* password = "ouedraogo410"; //Enter Password here

ESP8266WebServer server(80);

uint8_t LED1pin = D7;
bool LED1status = LOW;

uint8_t LED2pin = D6;
bool LED2status = LOW;

void setup() {
    Serial.begin(115200);
    delay(1000);
    pinMode(LED1pin, OUTPUT);
    pinMode(LED2pin, OUTPUT);

    Serial.println("Connecting to ");
    Serial.println(ssid);

    void handle_led1off() {
        LED1status = LOW;
        Serial.println("GPIO7 Status: OFF");
        server.send(200, "text/html", SendHTML(false,LED2status));
    }

    void handle_led2on() {
        LED2status = HIGH;
        Serial.println("GPIO6 Status: ON");
        server.send(200, "text/html", SendHTML(LED1status,true));
    }

    void handle_led2off() {
        LED2status = LOW;
        Serial.println("GPIO6 Status: OFF");
        server.send(200, "text/html", SendHTML(LED1status,false));
    }

    void handle_NotFound(){
        server.send(404, "text/plain", "Not found");
    }

    String SendHTML(uint8 + led1stat,uint8 + led2stat){

```

```

kassoum
//connect to your local wi-fi network
WiFi.begin(ssid, password);

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());

server.on("/", handle_OnConnect);
server.on("/led1on", handle_led1on);
server.on("/led1off", handle_led1off);
server.on("/led2on", handle_led2on);
server.on("/led2off", handle_led2off);
server.onNotFound(handle_NotFound);

server.begin();
Serial.println("HTTP server started");

id loop() {
    server.handleClient();
}

```

Images Q

```
kassoum
server.handleClient();
if(LED1status)
{digitalWrite(LED1pin, HIGH);}
else
{digitalWrite(LED1pin, LOW);}

if(LED2status)
{digitalWrite(LED2pin, HIGH);}
else
{digitalWrite(LED2pin, LOW);}

id handle_OnConnect() {
LED1status = LOW;
LED2status = LOW;
Serial.println("GPIO7 Status: OFF | GPIO6 Status: OFF");
server.send(200, "text/html", SendHTML(LED1status,LED2status));

id handle_led1on() {
LED1status = HIGH;
Serial.println("GPIO7 Status: ON");
server.send(200, "text/html", SendHTML(true,LED2status));
```

```
kassoum

String SendHTML(uint8_t led1stat,uint8_t led2stat){
String ptr = "<!DOCTYPE html> <html>\n";
ptr += "<head><meta name=\"viewport\" content=\"width=device-width, height=device-height\">\n";
ptr += "<title>Pomp Control</title>\n";
ptr += "<style>html { font-family: Helvetica; display: inline-block; vertical-align: middle; margin-top: 50px; width: 100%; text-align: center; } body { margin-top: 50px; } h1 { color: #444444; margin: 50px auto; width: 80%; text-align: center; } .button { display: block; width: 80px; background-color: #1abc9c; padding: 5px 10px; margin: 5px auto; } .button-on { background-color: #1abc9c; } .button-on:active { background-color: #16a085; } .button-off { background-color: #34495e; } .button-off:active { background-color: #2c3e50; } p { font-size: 14px; color: #888; margin-bottom: 10px; }</style>\n";
ptr += "<body>\n";
ptr += "<h1>ESP8266 Web Server</h1>\n";
ptr += "<h3>Humidity : x%</h3>\n";

if(led1stat)
{ptr += "<p>POMP Status: ON</p><a class=\"button button-off\" href=\"/\">OFF</a>\n";
else
{ptr += "<p>POMP Status: OFF</p><a class=\"button button-on\" href=\"/\">ON</a>\n";
```

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

/*Put your SSID & Password*/

const char* ssid = "OUEDRAOGO"; // Enter SSID here

const char* password = "ouedraogo410"; //Enter Password here

ESP8266WebServer server(80);

uint8_t LED1pin = D7;

bool LED1status = LOW;

uint8_t LED2pin = D6;

bool LED2status = LOW;

void setup() {

Serial.begin(115200);

delay(100);

pinMode(LED1pin, OUTPUT);

pinMode(LED2pin, OUTPUT);

Serial.println("Connecting to ");

Serial.println(ssid);

//connect to your local wi-fi network

WiFi.begin(ssid, password);

```

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());

server.on("/", handle_OnConnect);
server.on("/led1on", handle_led1on);
server.on("/led1off", handle_led1off);
server.on("/led2on", handle_led2on);
server.on("/led2off", handle_led2off);
server.onNotFound(handle_NotFound);

server.begin();
Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
  if(LED1status)
  {digitalWrite(LED1pin, HIGH);}
  else
  {digitalWrite(LED1pin, LOW);}

  if(LED2status)
  {digitalWrite(LED2pin, HIGH);}
  else
  {digitalWrite(LED2pin, LOW);}
}

void handle_OnConnect() {
  LED1status = LOW;
  LED2status = LOW;
  Serial.println("GPIO7 Status: OFF | GPIO6 Status: OFF");
  server.send(200, "text/html", SendHTML(LED1status,LED2status));
}

void handle_led1on() {
  LED1status = HIGH;
  Serial.println("GPIO7 Status: ON");
  server.send(200, "text/html", SendHTML(true,LED2status));
}

void handle_led1off() {
  LED1status = LOW;
  Serial.println("GPIO7 Status: OFF");
  server.send(200, "text/html", SendHTML(false,LED2status));
}

void handle_led2on() {
  LED2status = HIGH;
  Serial.println("GPIO6 Status: ON");
}

```

```

server.send(200, "text/html", SendHTML(LED1status,true));
}

void handle_led2off() {
  LED2status = LOW;
  Serial.println("GPIO6 Status: OFF");
  server.send(200, "text/html", SendHTML(LED1status,false));
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}

String SendHTML(uint8_t led1stat,uint8_t led2stat){
  String ptr = "<!DOCTYPE html> <html>\n";
  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
  ptr +="<title>Pomp Control</title>\n";
  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";
  ptr +=".button {display: block;width: 80px;background-color: #1abc9c;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";
  ptr +=".button-on {background-color: #1abc9c;}\n";
  ptr +=".button-on:active {background-color: #16a085;}\n";
  ptr +=".button-off {background-color: #34495e;}\n";
  ptr +=".button-off:active {background-color: #2c3e50;}\n";
  ptr +="p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
  ptr +="</style>\n";
  ptr +="</head>\n";
  ptr +="<body>\n";
  ptr +="<h1>ESP8266 Web Server</h1>\n";
  ptr +="<h3>Humidity : x%</h3>\n";

  if(led1stat)
  {ptr +="<p>POMP Status: ON</p><a class=\"button button-off\" href=\"/led1off\">OFF</a>\n";}
  else
  {ptr +="<p>POMP Status: OFF</p><a class=\"button button-on\" href=\"/led1on\">ON</a>\n";}

  ptr +="</body>\n";
  ptr +="</html>\n";
  return ptr;
}
*****

```

1.5.2.3 Schémas de Pompage de l'eau 3D et Concept du capteur de niveau d'eau-Lina:



Ce prototype représente le concept de base que nous allons adopter pour capter le niveau de l'eau dans le réservoir; il se compose d'un gobelet en plastique, une paille en plastique, et un fil d'étanchéité.

Le gobelet représente le réservoir dont les dimensions réelles sont les suivantes: 48cm*32cm

Le Test1 est l'étude de l'emplacement exacte du tube dans le réservoir, pour le gobelet de 8 cm de hauteur la longueur de la tige idéal était celle de 6cm : qui se situe au centre en laissant une marge de 2cm en haut et en bas du gobelet ce qui fait un rapport de **3/4**.

Et donc , pour notre réservoir dont la hauteur sera 32cm ; la longueur du tube devra être **$32 * \frac{3}{4} = 24$ cm**; avec une marge de **4cm** en bas et en haut du réservoir.



NB: Le tube sera en plastique rigide ,de couleur transparente afin de permettre la distinction de l'eau , et il sera gradué ... la graduation finale sera la limite d'eau qu'il ne faut pas excéder afin que l'eau ne déborde pas du réservoir.

Le Test 2 dans ce test nous avons étudié l'efficacité des deux joints d'étanchéité:

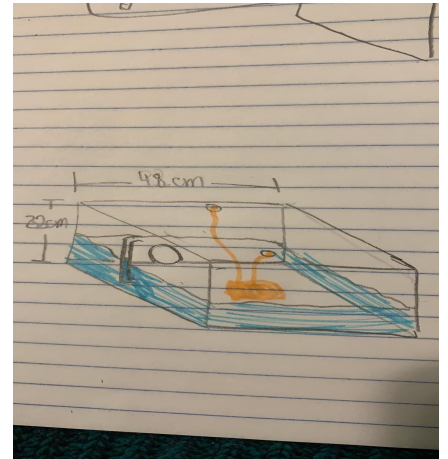
1-le fil qu'on fait rouler autour de la paille (réellement le tube) afin d'arrêter l'écoulement de l'eau.

2- le pistolet à cire : qui fait fondre de la cire qu'on met tout autour de la paille et généralement n'importe où nous voulons arrêter l'écoulement de l'eau.



En effet le fil est assez bon mais risque de s'abîmer rapidement avec le temps et sachant que notre réservoir est assez grand et donc nous ne pouvons pas prendre le risque.Néanmoins, le pistolet à cire a fait preuve d'une grande efficacité et il pourra résister un longtemps et déjà le réservoir sera fabriqué par du plastique assez rigide et donc la cire chaude ne causera pas de dégâts. Enfin, et pour augmenter l'efficacité de l'étanchéité, nous avons décidé de faire une combinaison entre les deux. Nous commencerons par

enrouler le fil autour du tube et après nous ajouterons une bonne couche de cire chaude qui assurera le maintien du fil et donc augmentera le niveau d'étanchéité.



Images R

1.5.2.4 Representation de l'interface WEB - Wadie

L'interface web n'est pas une tâche simple puisqu'elle requiert l'exploitation de plusieurs langages informatiques, pour la première étape et le premier test, il s'agit d'abord de créer la base de l'interface Web, en utilisant le langage HTML sur le logiciel sublimetext:

Une ligne de code sur ce langage commence toujours par <html> et finit par la même accolade. Il y a le <body> qui comprend le développement (des paragraphes) et avant des titres (appelé titres ou h) dans le <head>.



```
1 <!DOCTYPE html>
2 <html>
3 <head></head>
4   <title>GARDEN CONTROL</title>
5 </head>
6 <body>
7 LOGIN
8 </body>
9 </html>
```

Puis petit à petit il s'agit d'ajouter les éléments de design (les éléments interactifs ainsi que le serveur HTTP seront déterminés prochainement après). Les balises permettent de rentrer plusieurs fonctionnalités comme les images, les sauts de lignes, les éléments classés numérotés etc. il existe plusieurs balises dans le langage HTML.


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>GARDEN CONTROL</title>
5   <meta charset="utf-8" />
6   <h1>GARDEN CONTROL</h1>
7 </head>
8 <body>
9   <h1>Vous pouvez voir le niveau d'humidité ici</h1>
10  
11
12
13  <ol>
14    <li>Veuillez vous connecter.</li>
15    <li>Veuillez cliquer sur le bouton help pour de l'aide </li>
16  </ol>
17
18
19
20
21
22 </body>
23 </html>

```

Le test consistera alors d'ouvrir le fichier untitled.html via Google chrome et de visualiser si les informations sont correctes et si le format correspond .

Le deuxième test lui concerne le langage CSS , qui peut etre introduit dans le HTML pour le design , on ajoute alors la balise <style> pour personnaliser la couleur des chaînes de caractères , par exemple si on veut caractériser un text en bleu on tape le nom de la balise et cette chaîne de code avec l'ouverture de la balise Style synonyme de l'utilisation d'une commande css.

```

<!DOCTYPE html>
<html>
<head>
  <title>GARDEN CONTROL</title>
  <meta charset="utf-8" />
  <h1>GARDEN CONTROL</h1>
</head>
<style>
  title
  {
    color: blue;
  }
</style>
<body>
  <h1>Vous pouvez voir le niveau d'humidité ici</h1>
  

  <ol>
    <li>Veuillez vous connecter.</li>
    <li>Veuillez cliquer sur le bouton help pour de l'aide </li>
  </ol>

```

Images S

Puis après avoir créer le design qu'on pense satisfaisant , il ne nous reste plus qu'à essayer de se mettre à la place de l'utilisateur et faire les modifications nécessaires

1.5.3 Lessons et conclusions

La conception du prototype sur le programme solidworks a permis de comprendre la représentation de la personne en charge de la structure de la jardinière. Le code pour faire fonctionner le système de communication a permis d'en tester le fonctionnement. La conception du capteur de niveau d'eau a permis de satisfaire les attentes du client établie lors de la rétroaction de la rencontre client 2. Cette rétroaction était concernant le personnel à l'entretien de l'université qui remplissait le réservoir sans penser. Le capteur de niveau d'eau va permettre la sécurité de tous et des composantes électroniques du système de communication. La conception de l'interface WEB surprend le client et elle permet d'en assurer son fonctionnement ou de progresser vers cet objectif. L'interface web n'est pas une tâche facile. Pour conclure, ce livrable nous donne confiance en la faisabilité du prototype du produit final. Il ne manque qu'à réunir les composantes nécessaires pour la construction de celui-ci.

1.6 Prototypes #3

Les prototypes 1 et 2 nous ont permis de communiquer nos idées et de tester la faisabilité des systèmes. Concernant la structure, la représentation 3D en carton et sur solidworks du prototype nous ont permis la concrétisation de notre idée conceptuelle. Le prototypage du câblage et du système de communication et la rédaction du code nous permettent d'avoir une certitude sur le fonctionnement de la jardinière. Le prototypage du capteur de niveau d'eau et du système de pompage a permis de l'améliorer pour permettre de satisfaire les attentes du client. La création de l'interface web nous permet de dépasser les attentes et de faciliter l'entretien de la jardinière. Le prototypage et la rétroaction de la jardinière et de ses systèmes nous ont donc permis d'améliorer notre conception et d'en vérifier la faisabilité. Le prototype 3 donne l'opportunité d'améliorer notre solution au problème de conception.

1.6.1 Tableau du plan de test des prototypes

N° de Test	Objectif du test(Pourquoi)	Description du prototype utilisé et de la méthode de test de base (Quoi)	Description des résultats à documenter et comment ces résultats seront utilisés (Comment)	Durée estimée du test et date prévue du début du test (Quand)
Tests du fonctionnement du système				
1	Tester le branchement de la pompe à l'arduino	Plusieurs composantes ont été utilisées tels que les relais, les transistors, l'arduino et bien évidemment, la pompe.	Dans cette dernière partie des Test , nous nous sommes plus basés sur le courant nécessaire pour que nos actionneurs répondent correctement à nos ordres . Puisque l'arduino délivre du 5v	La durée estimée de cette tâche est de 1h, considérant les problèmes techniques qui pourraient arriver. Si tout va bien, pas plus de 10 minutes.

			la pompe doit être compatible. Les résultats de ce test sont soit positifs si la tension délivrée est suffisante au fonctionnement de la pompe et négatifs si non. Plusieurs recherches ont été menées sur les différents procédés d'amplification de tension	
2	Tester le branchement de la sonde d'humidité au NodeMCU	Le nodeMCU est utilisé pour séparer le système de commande mobile du système de commande central. D'où le terme employé jardinière sans fil. Les tests ont tout d'abord été effectués sur la connection effective entre la sonde capacitive et le NodeMCU afin que celui-ci puisse rediriger ces informations vers le système de contrôle et de commande mobile	Si le branchement fonctionne, les résultats de ce test seront positifs et on remarquera une variation du pourcentage d'humidité au niveau du serial monitor en fonction du milieu où la sonde sera placée.	La durée estimée de cette tâche est de 1h, considérant les problèmes techniques qui pourraient arriver. Si tout va bien, pas plus de 10 minutes.
3	Tester le branchement NodeMCU Arduino afin d'assurer la connection Serial	L'arduino, le NodeMCU, le breadboard, ainsi que des fils de connections mâles-mâles furent utilisés.	Si le branchement fonctionne, la connection serial sera effective et on remarquera un flux d'information dans les deux sens selon les lignes de codes que nous avons saisies	La méthode de branchement est déjà définie dans notre schéma de câblage donc mettre tout en œuvre et tester son fonctionnement pourrait prendre de 10 à 20 minutes si tout fonctionne comme prévu.
4	Tester le fonctionnement du code	Le prototype utilisé est constitué de la sonde d'humidité, l'arduino ainsi que la pompe.	Si l'arduino reçoit bien l'information de la sonde d'humidité et que la pompe est mise en marche lorsque le pourcentage d'humidité est inférieur à 7%, c'est positif.	Le code est difficile à écrire et une petite erreur dans celui-ci peut faire en sorte que rien ne fonctionne. Le raffinement du code est très important et faire certain que l'information passe bien de la sonde, l'arduino et ensuite la pompe prendra

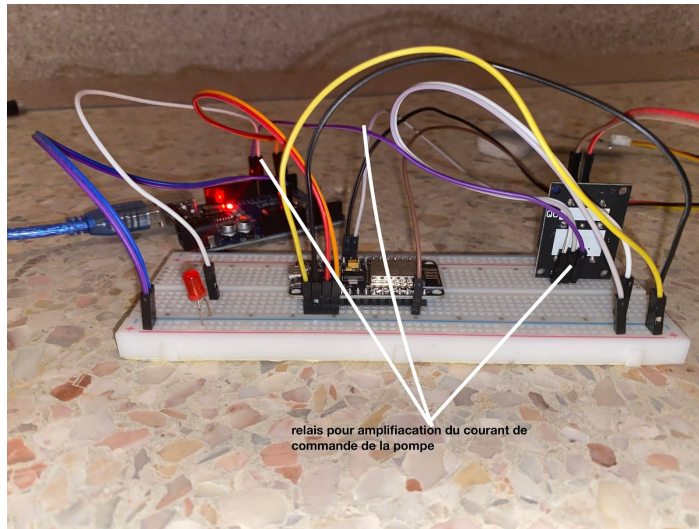
				approximativement une journée.
5	Tester le fonctionnement du système d'alerte en cas de problème	Le système d'alerte défini au livrable précédent se présente comme étant formé de plusieurs lignes de code qui vont jouer le rôle de chien de garde. Afin de tester le fonctionnement du système d'alerte, nous allons utiliser une Led pour l'instant à la place d'un buzzer vu qu'ils sont tous les deux polarisés de la même manière ainsi qu'un arduino.	Alors le arduino, en fonction de certains événements vérifiés ou non vérifiés enverra une impulsion électrique à la LED qui s'allumera pour nous signaler un problème. Ces résultats permettent de vérifier le branchement et aussi l'exactitude du code.	Ce test peut prendre une journée mais se fait de 30 à 40 minutes.
6	Tester le fonctionnement du NodeMCU en temps qu'hébergeur d'une interface web	Un nodeMCU possède plusieurs qualités dont celle d'hébergement d'une interface web. Alors nous allons à partir d'un navigateur, du nodeMCU, de l'adresse IP de celui-ci et des lignes de codes qui y ont été téléversés nous allons diriger un batterie de test pour s'assurer de l'hébergement de la plateforme web.	Dans cette batterie de tests, plusieurs aspects sont traités. Tout d'abord le test d'ouverture de la plateforme web lorsque l'adresse ip du NODEMCU est saisi dans un navigateur, suivi de l'aspect design de cette plateforme ainsi que la présence de tous les onglets nécessaires au bon fonctionnement de notre prototype.	La durée estimée de cette tâche est de 1h, considérant les problèmes techniques qui pourraient arriver. Si tout va bien, pas plus de 10 minutes.
7	Tester le fonctionnement de l'interface	Toutes les composantes du système de communication ainsi que les composantes du système d'arrosage font partie du prototype nécessaire pour ce test.	Les données qui seront reçues sur l'interface ainsi que les données qui seront envoyés aux autres systèmes de commandes définissent l'issue de ces tests	La durée estimée de cette tâche est d'une semaine, considérant les problèmes techniques qui pourraient arriver. Si tout va bien, pas plus de deux jours.

Tableau 10-plan de test 3

1.6.2 Résultats des tests

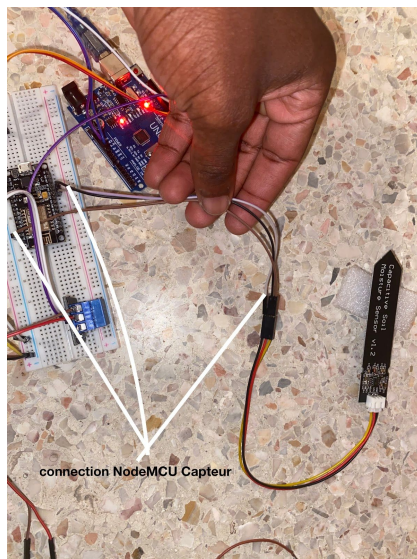
Test 1:

Le premier test avait comme objectif de vérifier la fonctionnalité du branchement de la pompe à l'arduino. Lors du test, nous avons tout d'abord essayé d'alimenter la pompe à partir des 5v par l'arduino, ce qui fut sans succès. Alors nous avons par la suite fait usage de relais jqc-3ff-s-z. Ceux-ci prennent en entrée les 5 volts de l'arduino et nous délivrent en sortie X volts au niveau en fonction de la tension du courant auquel nous l'avons relié. C'est ainsi que nous avons utilisé des piles de 9 volts pour effectuer nos tests. Le résultat fut concluant. Nous fûmes en capacité de commander la pompe à partir du relais, de l'arduino et des piles de 9V.



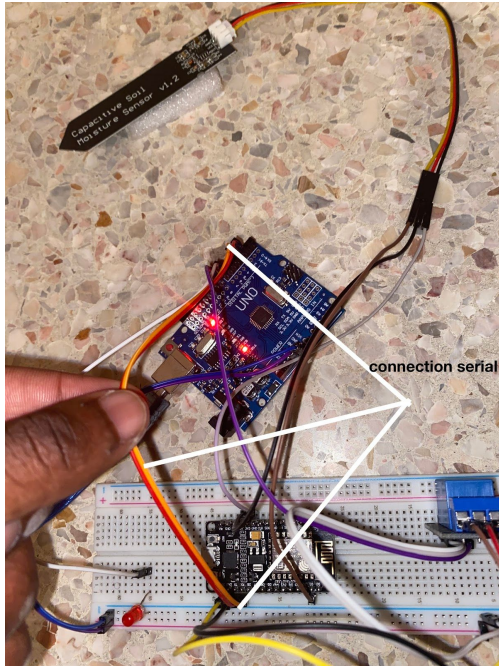
Test 2:

Le deuxième test avait comme objectif de vérifier la fonctionnalité du branchement de la sonde d'humidité au NodeMCU. Lors du test, le monitor serial affichait effectivement les valeurs capacitives du pourcentage d'humidité du sol ou notre sonde était placée et ces résultats étaient concluants. La prochaine étape est de directement transmettre ces résultats reçus à l'interface web et à l'arduino.



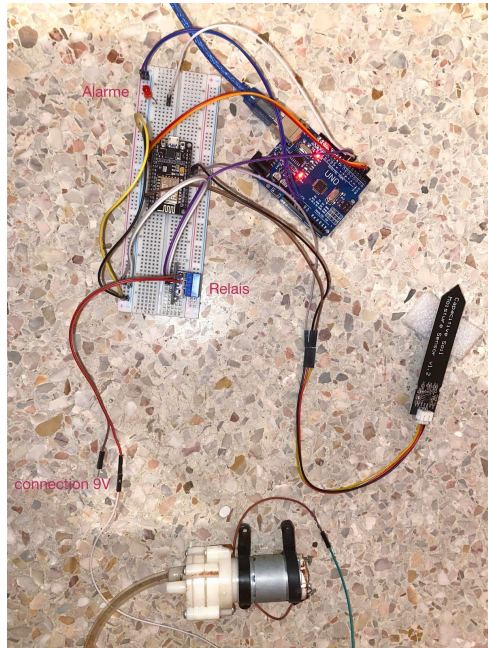
Test 3:

Le troisième test avait comme objectif de vérifier la fonctionnalité du branchement entre le NodeMCU et le arduino. La connexion serial dans les deux sens devrait être vérifiée. Ainsi nous avons effectué le branchement sur le breadboard et procède à un envoie d'information du Nodemcu vers l'arduino et Vise Versa. Lorsque l'envoi est effectif, le serial moniteur devrait nous le confirmer à travers un message. Ce test son tour fut concluant.



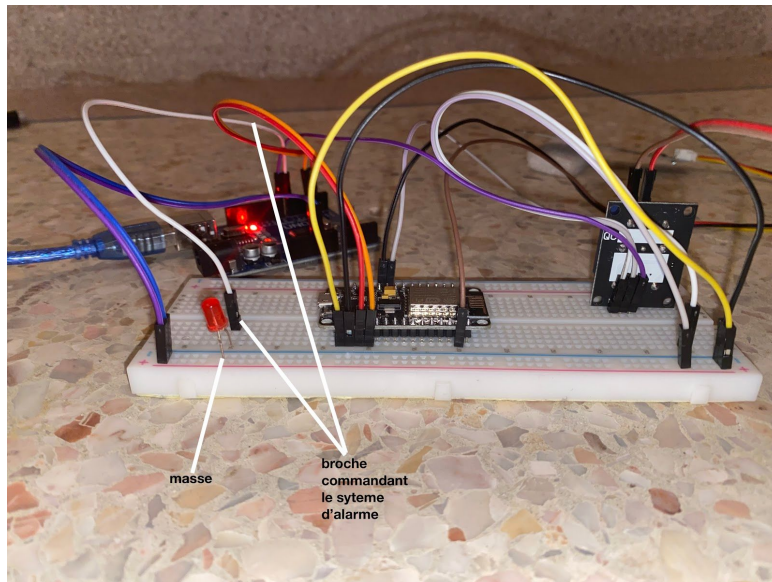
Test 4:

Le quatrième test avait comme objectif de vérifier la fonctionnalité du code. Tous ces tests sont clôturés par le cablage complet de notre prototype afin de vérifier le fonctionnement en parallèle des différentes parties.



Test 5:

Le cinquième test avait comme objectif de vérifier la fonctionnalité du système d'alerte en cas de problème. L'arduino envoyait ainsi des impulsions à notre Led en fonction de plusieurs événements qu'il aurait vérifié. Les résultats sont concluants et on s'assurera du bon branchement entre l'arduino et la Led qui remplace le buzzer en attendant que nous puissions nous en procurer une. L'effectivité de cette partie du code fut aussi testée parallèlement.



Images S

Test 6:

Le sixième test avait comme objectif de vérifier la fonctionnalité du NodeMCU et la connexion du système de communication avec l'interface. La génération de l'interface fut effective et celui-ci possédait tous les onglets dont nous avons besoins. Ces onglets aussi réalisaient correctement les tâches auxquels ils étaient assignés tels que l'envoi d'une impulsion électrique à l'arduino lorsque l'utilisateur appui sur le bouton ON. cette impulsion servira à commander la mise en marche de la pompe.

Test 7:

Le septième test avait comme objectif de vérifier la fonctionnalité de la communication Hardware Software. Le test fut effectué avec succès car nous arrivons à recevoir les pourcentages d'humidité sur l'interface et aussi nous arrivons à envoyer des commandes depuis celui-ci en se servant des onglets.

Tous ces tests sont clôturés par le câblage complet de notre prototype afin de vérifier le fonctionnement en parallèle des différentes parties.

1.6.3 Rétroaction du client

Le 09 novembre 2020, notre équipe a eu la chance de rencontrer le client et de lui présenter son second prototype en lui expliquant le fonctionnement général de la jardinière ainsi que le fonctionnement et l'intégrité des sous-parties dans le système au complet.

En effet, nous avons commencé par présenter la structure extérieure de la jardinière ainsi que les différents matériaux qui seront utilisés pour ça. Ensuite, nous avons passer à évoquer le système de pompage et d'arrosage incluant le type de pompe qui sera utilisé et la disposition des tuyaux et de la pompe au sein du réservoir dont le matériel à été aussi précisé. Puis, nous avons expliqué d'une façon simpliste le schéma de câblage du concept de base ainsi que toutes les pièces électroniques qui seront incluses, et enfin la présentation de l'idée de l'interface WEB, des informations qu'elle va pouvoir affichée à l'utilisateur ainsi que le design de cette dernière.

La rétroaction du client a été globalement positive et motivante. Il a apprécié le fait que le système peut être contrôlé à distance via l'interface web, ainsi que le système de la maximisation de l'utilisation de l'eau par le filtrage de l'excès de l'eau et son retour au réservoir. Néanmoins, la seule remarque qui nous a été faites c'est l'amélioration d'un système qui capte le niveau de l'eau dans le réservoir afin de ne pas causer un débordement de l'eau au sein du système, ainsi qu'il avait demandé l'ajout de deux couvercles un pour le pot extérieur et l'autre pour le réservoir afin de permettre le lavage plus ou moins mensuel du système, ainsi que permettre le réglage de la pompe et de toutes les composantes électroniques qui vont être incluses dans un petit boîtier au sein du réservoir. Notre équipe a pris toutes ses notes en considération et a amélioré son prototype pour être de plus en plus conforme aux besoins et désirs du client, comme est bien détaillé tout au long de ce livrable.

1.6.4 Lessons et conclusions

Ce dernier livrable regroupe les tests finals du produits et a vraiment permis de vérifier le fonctionnement du système de notre jardinière. Le prototype 3 a permis l'amélioration de notre solution au problème de conception. La liste des matériaux finalise les utilisations en termes de composantes et de

matériaux de construction. La rétroaction du client a permis à l'équipe d'améliorer les prototypes pour avoir un produit final plus commode et fonctionnel. Le processus de la conception de cette jardinière a permis une évolution des concepts et des objectifs tout en gardant en tête la définition du problème. Nous croyons alors que notre jardinière est une bonne solution au problème de Jonathan, du manque de verdure sur le campus de l'université d'Ottawa.

2 Comment le prototype est construit

Construction du système d'arrosage: Ce système bien évidemment se compose principalement d'une pompe submersible vu que cette dernière doit être au fond du réservoir pour absorber l'eau et puis le pomper et non pas à l'extérieur afin de garder notre produit de plus en plus compact; Et on passe au réservoir qu'on a choisi qu'il soit en plastique rigide afin d'éviter le rouillement si on aurait utilisé d'autres matériaux, on pourrait utiliser l'inox par exemple mais c'est plus cher; Ce réservoir contiendra aussi un bouchon de remplissage où l'utilisateur pourra le remplir une fois que le volume d'eau atteint le minimum; Et pour capter le niveau d'eau nous n'avons pas utilisé un capteur électronique mais plutôt un qui est plus ou moins manuel, une sorte de fissure dans le boîtier qui permet de voir le réservoir transparent, donc le niveau d'eau! Et bien sûr nous avons ajouté une petite marque pour préciser que voici le niveau maximal du réservoir que celui qui remplit ne doit surtout pas dépasser pour éviter le débordement de l'eau au sein de la jardinière, même si le réservoir contient déjà un couvercle pour éviter ça! Or, la pompe ne peut pas pomper l'eau sans les tuyaux, pour notre prototype nous avons utilisé deux tuyaux à spray afin d'élargir l'air d'arrosage et garantir que chaque partie du sol soit assez humide et donc la santé des plantes sera protégée.

Construction du système de communication:Le système de communication a pour composant principal le nodeMCU. Il est utilisé comme passerelle entre la partie commande à distance et celle fixe. Il héberge l'interface de commande à distance et sera chargé de la transmission des commandes à distance de l'interface vers la partie de commande fixe qui est l'arduino. C'est dans ce cadre que nous avons mis en place une communication serial entre les deux composants(arduino et nodeMCU).Il existe aussi une communication entre le nodeMCU et le capteur d'humidité pour permettre au nodeMCU d'intercepter les différents pourcentages d'humidité du sol, les télécharger sur l'interface web et effectuer le partage avec l'arduino à travers la communication serial. Nous avons utilisé des câbles mal-femelle afin d'établir la communication entre l'arduino et le nodeMCU en branchant deux broches de l'arduino aux broches RX et TX du nodeMCU. La communication entre le capteur d'humidité et le nodeMCU fut établie en reliant leur masse à la masse commune, ainsi que le Vcc du capteur capacitif à celui du nodeMCU pour assurer l'alimentation. La broche de valeur analogique du capteur d'humidité est reliée à la broche D3 du nodeMCU. En effet, la communication avec l'interface de commande à distance est principalement basée sur des lignes de codes qui seront expliquées plus bas. Ensuite, la communication buzzer où pompe est suivie en reliant les différentes masses des composants à la masse commune et les différents vcc à la broche de l'arduino chargée de leur commande distinct. N'oublions pas qu'avant d'arriver à la pompe, le signal venant de l'arduino sera amplifié à l'aide d'un relais.C'est différents câblages ont tout d'abord été désignés

sur le logiciel fritzing, par la suite testé séparément et enfin relié ensemble pour construire le prototype final.

Construction de l'interface: L'interface Web se fait à partir du Web server ESP8266 , elle est accessible à partir de n'importe quel appareil connecté au local network , Tout d'abord on installe l'application Arduino IDE pour créer un serveur WEB , nous avons pu compter sur internet pour apprendre comment réaliser cette tâche , à travers nos recherches , nous avons suivis les instructions du circuit du serveur WEB comme la fabrication du code qui a été copié dans l' IDE Arduino et téléchargé sur le NodeMCU et qui était constitué de différents langages informatiques comme le langage html , consistant à créer la base de l'interface , le CSS consistant à désigner la page web . Puis Ensuite pour l'ouverture il suffit d'ouvrir le moniteur série et attendre que NodeMCU soit connecté au réseau WiFi, le serveur Web a lancé et affiche l'adresse IP et en plus Lorsqu'un utilisateur non autorisé souhaite accéder ou que les informations d'identification saisies sont erronées, le message «Échec de l'authentification» s'affiche dans le navigateur.

Construction de la structure: Bien sûr, le prototype de la structure a été fait sur solidworks partie par partie pour ensuite en faire l'assemblage. Ce prototype est important puisqu'il nous a permis de vérifier plusieurs aspects de la faisabilité de notre conception. L'aluminium est le matériel choisi pour les composantes comme le boîtier, son couvercle et le compartiment à composantes électroniques. Ce choix n'était pas aléatoire. L'aluminium est le métal le plus recyclé du monde, plus de 75% de l'aluminium qui circule à cet instant même est recyclé. Le client donnait une grande importance au niveau écologique du produit et c'est pourquoi ce fut un choix évident pour la construction de la jardinière. Ce matériel doit résister à la corrosion et à la rouille parce qu'il y a des chances qu'il entre en contact avec de l'eau. Le bois des pattes a été choisi plutôt pour son esthétique et sa force. Le plastique du réservoir est plus facile à laver que du bois par exemple et retient des liquides sans fuites.

2.1 Catégorie

2.1.1 LDM

	Matériaux et composantes	Valeurs	unité	Prix total
Structure extérieure	Bois : 2x2x8 Épinette Select EP 228 PREM	2.99\$	x5	14.95\$
	Aluminium : Tôle pliable	12.99\$	x3	38.97\$
	Poignées : Poignée fonctionnelle en plastique 3 3/4 in (96 mm) CàC - Noir	2.02\$	x2	4.04\$
	Pompe : Ultra Quiet Mini DC 12V Lift 5M 800L/H Brushless Motor Submersible Water	18.33\$	x1	18.33\$

Système d'arrosage et de communication	Pump			
	Tuyaux : Tuyau principal en vinyle, 1/2 po × 50 pi	17.60\$	x1	17.60\$
	Arduino : Microcontrôleur Arduino Uno R3 USB (inclu dans le kit des laboratoire)	26.95\$	x1	0\$
	Fils : Fils Connecteurs pour Arduino (65pk) (inclu dans le kit des laboratoire)	6.27\$	x15	0\$
	Résistances (inclu dans le kit des laboratoire)	-	2x	0\$
	Breadboard (inclu dans le kit des laboratoire)	-	x1	0\$
	Battery : Support de batterie ElecFreaks 2xAA avec interrupteur et couvercle pour micro: bit	1.84\$	1x	1.84\$
Réservoir avec bouchon	Bidon à essence : Bidon à essence Scepter, 20L		1x	0\$

Tableau 11-Liste des matériaux

2.1.2 Liste d'équipements

- ***Équipement pour construction du boîtier:***

No	Équipement
1	Découpe laser, Epilog Laser Mini
2	Cisaille à pédale
3	Plieuse à tôle
4	Pinces
5	Une perceuse, des vis et une boulonneuse à percussion

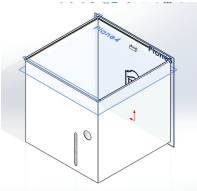
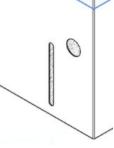
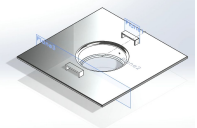
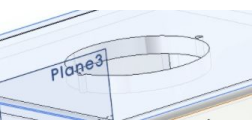
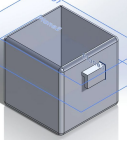
- ***Équipement pour construction du système de communication***

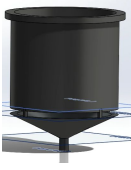
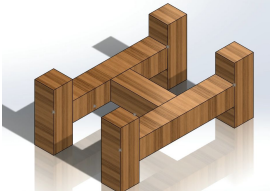
No	Équipement
1	Capteur d'humidite et nodeMCU
2	NodeMCU et Arduino
3	Arduino et Pompe 9V accompagné d'un amplificateur de tension

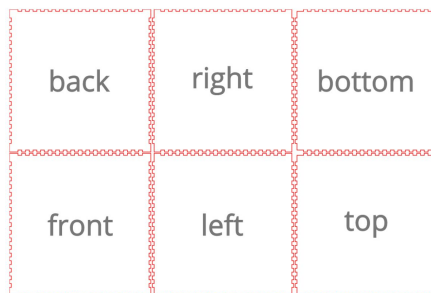
4	Arduino et Buzzer
---	-------------------

2.1.3 Instructions

- Étapes de construction du boîtier

Étapes	Explications	
1	Couper la tôle avec le laser cutter pour créer la forme du boîtier ainsi que son couvercle. Les dimensions de la boîte seront de 50x50x50cm.	
2	Couper la fissure pour le visuel du niveau d'eau ainsi que le trou pour faire passer la bouche de remplissage	
3	Couper un trou centré au centre du couvercle de diamètre de 20cm ou en fonction du diamètre de votre pot de plante.	
4	Couper un trou centré au centre du réservoir en fonction du diamètre de l'entonnoir	
5	Avec une perceuse à trous, percer des trous de même diamètre que vos tubes de transportation d'eau. Ceux-ci sont situés, un chaque côté du trou de plante sur le couvercle et un sur chaque côté du trou d'entonnoir sur le réservoir.	
6	Couper la tôle avec le laser cutter pour créer la forme du boîtier à composante électronique. Les dimensions de la boîte seront de 10x10x10cm.	

7	Coller l'entonnoir bien centré au pot filtrés de la plante	
8	Couper les 2x2 en 4 morceaux de longueur de 30cm, 2 morceaux de longueur de 50cm et un autre de longueur de 20cm.	
9	 <p>Les 50cm relie deux pattes ensemble et le 20cm relie les 50cm ensemble. Pour l'assemblage, vissez les composantes ensembles avec une perceuse, des vis et une boulonneuse à percussion.</p>	
10	Optionnelle: Vous pouvez peindre le logo de votre université sur le boîtier de la jardinière	



Dessin des boîtiers pour le Couper au Laser (Voir Appendice II pour dessins d'assemblage)

- **Étapes de construction du système de communication**

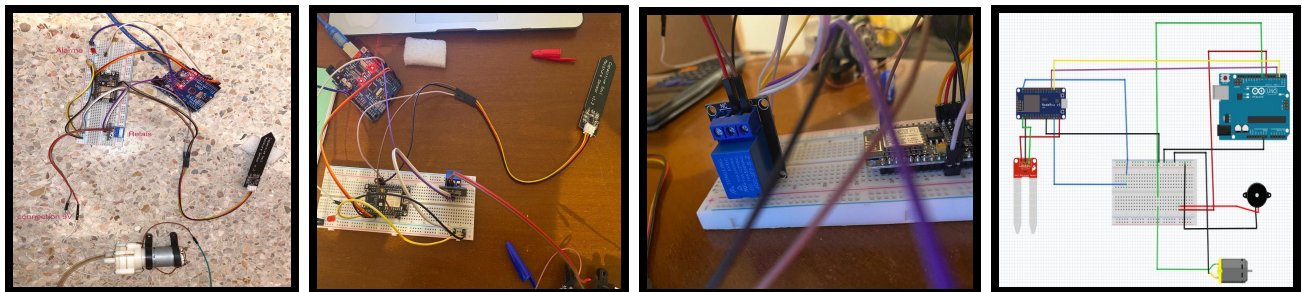
No	Étapes	Explications
1	Schemas de Cablage	<p>Capteur d'humidité NodeMCU: Le Gnd et le Vcc des deux composants ont etetes relies afin d'assurer l'alimentation en courant électrique.les données capacitifs du pourcentage d'humidite sont receuillit et envoyés de la broche OUT vers la broche D01 du NODEMCU</p> <p>NODEMCU-ARDUINO: Connection Serial entre les deux composants assure par les broches TX et RX du nodeMCU connecté aux broches 03 et 04 de l'arduino. l'Arduino et le NodeMCU sont alimentes tous les deux par du 5V</p>

		<p>ARDUINO POMPE La pompe n'est pas polarise ainsi c'elle-ci sera alimente par la broche 7 de l'arduino . l'impulsion de 5 volts qui lui est delivre sera amplifie grace a un relai . le courant quittant l'arduino par la broche 7 entre dans le relais par ca broche S. celui ci est relie a la masse et a l'alimentation commune du système par son GND et son VCC</p> <p>Arduino buzzer L'alarme (le buzzer ou la l'aide) est un composant polarise. Celui-ci a son VCC qui est relie a la broche 7 du arduino. Son GND quand a la est relie a la masse commune du systeme de communication</p>
2	Conception du protocole de communication serial Node MCU -Arduino	<p>Notre prototype presentaot un besoin de partage d'information du NODEMCU vers l'arduino. Celui-ci devait partager ces informations avec la plateforme de commande a distance quelle heberge ainsi qu'avec l'arduino ou le système de commande fixe. Nous avons donc, a l'aide de code que vous trouverez dans nos fichier .INO installer cette connection serial entre l'arduino et le NODE MCU. une partie du code fut upload dans le NODEMCU fin de permettre a celui-ci d'envoyer des informations vers une autre entite.</p> <p>Une seconde partie du code fut uplod dans l'Arduino pour permettre a celui-ci d'intercepter les informations venat du node MCU .</p> <p>Alors , en fonction des informations qui seront recuent par l'arduino , si le pourcentage d'humidite est inferieur a 50% ou si l'onglet de mise en marche de la pompe est actif , alors celui-ci va directement commencer le pom[age de l'eau.</p> <p>L'action inverse deroules lorsque l'onglet est desactive ou le pourcentage d'humidite est >50%</p>
3	Conception du code d'hébergement de l'interface sur le NODEMCU	<p>Le code d'hebergement de l'interface web fut upload dans le nodeMCU car cest lui qui jouera le role d'hébergeur afin de minimiser les couts de la mise en place d'une plateforme web . ainsi dans notre code, nous devons y regle le SSID du wifi ainsi que son motr de passe car il sera utilise par le nodeMCU pour etablir la connection wifi (partie put your password and SSID et partie check if Wifi connected). le responsable lui aussi doit etre sur le meme reseau wifi et va dans son navigateur saisir l'adresse IP du nodeMCU qui sera afficher dans le serial MONITOR lors des tests. Ainsi a la longue il n'aura qua se connecter par cette adresse IP et recevra les informations concernant le pourcentage d'humidite du sol car une partie du code fut désignée à cet effet. L'autre partie fut designe pour</p>

		envoyer des impulsions electriques vers l'arduino lorsque les ongetlets mise en marche ou arret du moteur sont actives.
4	Transcription du langage machine venant de l'interface web en impulsion électrique	Dan sle code upload dans le node MCU, nous avons une partie lorsque l'ongle mise en mar he est active (handle_led1on) et lorsque celle-ci aussi est desactive(handle_led1off). Ceci entraine l'allumage (handle_led2on)ou l'instection(handle_led2off) d'une lampe sur notre interface. Cest etats definissent la mise en marche ou l'arret de la pompe. Alors lorsque la lampe est mise en marche, cette informatiosn est transferer a l'arduino a travers la connection serial . le code arduino se chargera de traiter ces informations. Lorsquil recevra dans sa boucle if, "led1 on" ou "led2 off", il allumera ou eteindra la pompe en lui envoyant une impulsion electrique.

Tableau 12-Étapes d'instructions

Concrétisation par images :

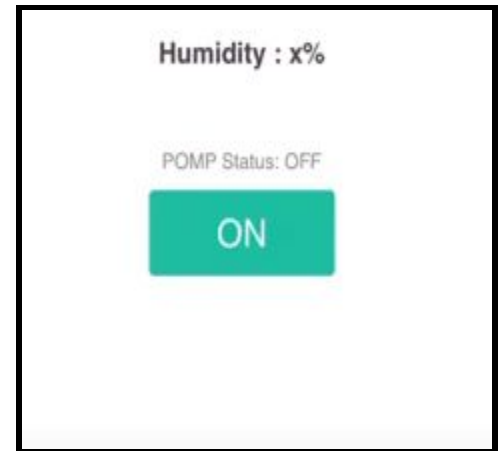
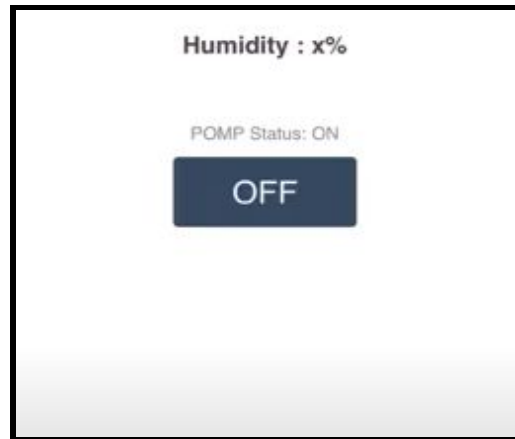
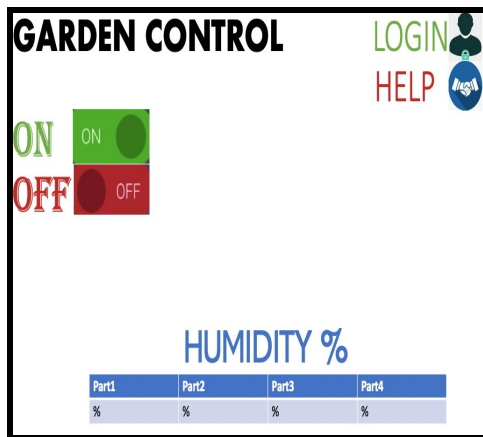


3 Comment utiliser le prototype:

Utilisation de la Jardinière Autonome

- La Jardinière est un produit autonome qui arrose les plantes selon le pourcentage d'humidité détecté par les sondes d'humidité.
- Un remplissage du réservoir est nécessaire une fois par mois, ce pour garder le réservoir plein en tout temps. (Système antivol)
- L'utilisation de l'interface est simple et est conçue pour faciliter l'utilisation de la jardinière. L'interface donne accès à l'information du pourcentage d'humidité détecté en tout temps. L'interface offre une commande à distance de la pompe ainsi qu'un confidentialité impeccable
- Pour assurer une utilisation sécuritaire de la jardinière, soyez certain que les composantes électroniques sont à leur place lors du remplissage du réservoir. Cela évite un contact de l'eau avec les fils électriques et donc une électrocution.

- Pour une l'installation de la jardinière, choisissez un endroit éliminer pour que la plante puisse faire la photosynthèse. L'endroit ne devrait pas être trop passant puisque le risque de collision avec la jardinière augmenterait. Choisissez un endroit calme où les gens qui étudient peuvent en admirer la beauté et la verdure surtout. La jardinière est solide donc l'endroit d'installation n'est jamais coulé dans le béton.



4 Comment maintenir le prototype:

Maintenance régulière:

Plusieurs tests ont été effectués pour la validation de la conception finale. Ceux-ci sont décrites dans la section de ce manuel des prototype #1, #2 et #3 ainsi que les résultats applicables. Pour une utilisation durable, il est important de prendre soin des composantes intérieures lors du lavage du réservoir.

Lavage du réservoir: Annuel

1. À l'aide des poignées situées sur le dessus de la Jardinière, soulever le couvercle pour accéder aux composantes intérieures.
2. Avant de commencer le lavage des composantes, déconnectez la sonde d'humidité et les tuyaux d'eau du pot de terre et mettez la plante et le couvercle en lieu sûr.
3. Ensuite, soulever la boîte qui contient les composantes du système de communication, mettez aussi les composantes électroniques ainsi que la pompe en lieux sûr.
4. Vous devriez être en mesure de soulever les réservoirs et de le laver.
5. Utiliser de l'eau et du savon organique pour rincer l'intérieur du réservoir pour assurer la bonne santé des plantes.
6. Refermer le couvercle et assurez-vous que toutes les composantes sont en place.

Changement des batteries: Hebdomadaire

1. Les batteries les plus performantes doivent être changées une fois par semaine.
2. À l'aide des poignées situées sur le dessus de la Jardinière, soulever le couvercle pour accéder aux composantes intérieures.
3. Les batteries se trouve dans le conteneur à batterie, ouvrez le et simplement changer les piles
4. N'oubliez pas de bien positionner les piles, les côtés avec les côtés positifs, ainsi que de bien refermer le porteur de batterie.
5. Refermer le couvercle et assurez-vous que toutes les composantes sont en place.

Remplissage du réservoir: Une fois au trois mois

1. Ouvrir le bouchon de remplissage dans le sens antihoraire pour accéder au réservoir.
2. Remplir le réservoir jusqu'à ce que le niveau d'eau atteigne la ligne noir. Celle-ci indique le niveau maximal d'eau pour la sécurité du produit et de ses composantes.
3. Refermez le bouchon de remplissage de façon ferme pour que celle-ci ne s'ouvre pas par accident.

Pièces avec haut niveau de risque de défaillance

1. Le porteur de piles est en plastique et son fonctionnement
2. Les connexions de fils du système de communication
3. Le bouchon de remplissage
4. La pompe et la connexion des tuyau à celle-ci jusqu'à la plante

5 Conclusions et recommandations pour les travaux futurs

Tout d'abord la demande de la réalisation d'un projet est une chose très avantageuse surtout durant les 1ere années du parcours universitaire vu que la plupart des étudiants gardent un peu l'esprit lycéen qui se base plus sur la théorie que sur la pratique, et donc la réalisation des projets nous permet un certain contact entre les études et le monde réel et concret du travail.

Sur ce, le génie de la conception en général et ce projet de jardinière à arrosage automatique et commande à distance précisément nous ont permis d'acquérir une bonne et large expérience que ça soit sur l'esprit personnel ou celui de l'équipe. Dans le cadre personnel, chaque membre de notre équipe a appris comment bien gérer son temps, améliorer encore plus son organisation, forger sa personnalité en menant des entrevues avec le client et en présentant devant notre professeur et nos collègues. Les membres de l'équipe ont pu tester leurs capacité de recevoir la rétroaction de l'autre, soit positive ou négative, ils ont pu répondre aux questions sans hésitations. Les membres ont pu faire des recherches scientifiques en étudiant la santé des plantes et l'électronique, primordiale pour la commande à distance de notre jardinière.

D'autre part, toujours dans les éléments positifs de notre projet, on se permettra tout d'abord de mentionner la cohésion du groupe. Personne n'est resté en retrait, les tâches ont été équitablement réparties, non seulement au niveau de la quantité de travail qu'au niveau des compétences et des centres d'intérêts de chacun. Les grandes lignes ont été approuvées par tous les membres du groupe et aucune altercation n'est survenue tout au long de l'année. Le travail en équipe a donc été très fructueux. Bien sûr, nos avis n'ont pas toujours été communs mais des compromis ont été trouvés pour contenter tout le monde; En fait, malgré la situation récente dont on fait face et que chacun de notre équipe était dans un pays différent: Maroc, Burkina Faso, Canada, grâce à la bonne communication, la motivation, l'enthousiasme, la créativité, ainsi que le respect mutuel entre les membres de l'équipe nous avons pu surmonter ce challenge et mettre en réalité ce projet de jardinière automatique.

Toutefois, si ce projet devrait être réalisé ou même complété, on recommande à l'utilisateur l'amélioration de l'interface web en raffinant le design et pourquoi pas permettre l'accès à encore plus d'informations par exemple l'humidité du sol de la face extérieure et intérieure; L'ajout d'un système d'alarme en cas de défaillance; l'utilisation de matériaux de plus haute qualité pour éviter le rouille du boîtier et des couvercles, L'utilisation d'un capteur d'humidité électronique au lieu du manuel afin de permettre au utilisateur de connaître le niveau de l'eau récent dans le réservoir tout simplement en accédant à l'interface web; l'utilisation d'un matériel à très grande étanchéité pour le mini boîtier qui contient les composantes électroniques afin d'éviter le contact avec l'eau et donc leurs dysfonctionnements. D'autre part, ça peut être optimiser esthétiquement aussi en trouvant un design encore plus futuriste et beau que celui présenté dans les images ci-dessus.

Pour clore, l'équipe **sucre&potager** souhaite fortement que la conception de la jardinière automatique vous plaise et bonne chance si vous avez l'intention d'améliorer le concept.

6 Bibliographie

[Le sol est une éponge](#)

[Santé des plantes](#)

[Gardena](#)

[Plante pothos](#)

[pompage](#)

APPENDICES I: Fichiers de conception

Lien pour makerrepo: <https://makerepo.com/Danika/gng1503-a10-self-watering-planter>

Fichiers CAD: Les fichiers de conception sur solidworks comprennent la conception de chaque pièces. Ceux-ci peuvent vous aider à comprendre le fonctionnement du prototype et les matériaux utilisés pour chaque pièce. On y trouve le fichier d'assemblage de la base en bois et le type de bois utilisé ainsi que l'assemblage final du produit. Les simulations de forces y sont incluses ainsi que les vues explosées et animées de l'assemblage.

Fichier Code: les fichiers .INO sont divisés en trois. Deux parties ont été jumelés afin d'être uploader dans le NodeMCU mais ces parties sont concurrentes séparément car elles traitent deux aspects différents. Une partie est basée sur la conception de l'interface ainsi que le traitement des informations de l'interface. L'autre partie est basée sur le recueil d'information venant du capteur d'humidité ainsi qu'une partie du code d'abolition de la connexion serial arduino et NodeMCU. La troisième partie du code représente le code du système de commande fixe ou de l'arduino. Il a été conçu afin de commander la pompe et le buzzer en fonction des différentes données du capteur d'humidité. Il constitue un pilier aussi pour la commande a distance car il reçoit des informations en provenance de la plateforme web et assure la mise en place des actions souhaitées. Il assure aussi la gestion des erreurs de fonctionnement lorsque la pompe refuse de pomper pour cause de manque d'eau par exemple, la défaillance du capteur d'humidité ou un bug au niveau de la plateforme web.

APPENDICES II: Dessins d'assemblage

