

GNG1503

Manuel d'Utilisateur du Projet de Conception

Surveillance de Jardinières: Green Tide

Soumis par:

Groupe A06	
Abdel Hamid Abdoul Kadiri	300211823
Meghan Brown	300039235
Tyler Byrne	300039234
Danick Lamarche-Tardif	300169247

Le 9 décembre 2020

Université d'Ottawa

Remerciements

Nous remercions notre professeur, Emmanuel Bouendeu et notre client, M. Jonathan Rausseo pour cette belle opportunité.

Nous aimerions également remercier nos AEs Eric et Yassine qui nous ont guidés tout au long de ce processus.

Résumé

Dans le cadre du cours GNG1503 (Génie de la conception), nous avons relevé le défi de concevoir une jardinière d'intérieur afin de décorer les espaces communs de l'Université d'Ottawa pour notre client Jonathan Rausseo, gestionnaire du développement durable. Il aimerait que la jardinière soit autosuffisante, puisque l'université ne veut pas dépenser trop d'argent sur l'emploi d'entretien.

Jonathan Rausseo veut que la jardinière requiert un minimum d'entretien. Pour minimiser l'entretien, on a choisi d'ajouter un système d'arrosage automatique et un système de surveillance du sol. Aussi pour plaire à tout le monde, la jardinière devrait avoir une esthétique moderne. De plus, monsieur Rausseo nous a donné un budget de 100\$CAD. Puisque la jardinière sera localisée dans un environnement public, elle doit donner l'impression qu'elle est difficile à voler pour ne pas que les gens essaient de la voler mais devrait pouvoir être déplacés assez facilement au besoin.

Pour qu'on puisse répondre à tous ses besoins, nous avons suivi le processus de conception ce qui nous permet d'empathiser avec le client. Ceci nous a permis de créer un prototype fonctionnel selon les contraintes établies par le client.

Table des matières

Remerciements	1
Résumé	2
Table des matières	3
Liste de figures	5
Liste de tables	6
Liste d'acronymes	7
Introduction	8
Identification des besoins et processus de spécification du produit	8
Énoncé du problème	8
Besoins	8
Étalonnage	9
Spécifications cibles	11
Conceptualisation	12
Sous-système 1 : Le planteur	12
Concept 1 : Planteur Carré	12
Concept 2 : Planteur carré avec porte	13
Concept 3 : Planteur rectangulaire courte	13
Sous-système 2 : Le système d'arrosage	14
Concept 1 : Pompe	14
Concept 2 : Gravité	14
Concept 3 : Système d'eau de la ville	15
Sous-système 3 : Le système de surveillance du sol	16
Concept 1 : Capteur d'humidité du sol	16
Les trois systèmes	16
Étalonnage et choix de concept	18
Plan du projet	20
Plan du projet	20
LDM	22
Liste d'équipements	22
Prototypage, essais et validation du client	23
Prototype I	23

Prototype II	24
Prototype III	26
Solution Finale	28
Analyse	31
Nomenclature des matériaux	31
Choix des composantes électriques	31
Capteur d'humidité du sol	31
Servo	31
Capteur de débit	32
Circuit électrique	32
Code du microcontrôleur (Arduino)	33
Partie I : Capteur d'humidité du sol	33
Partie II : Capteur de débit	34
Partie III : Servo	34
Partie IV: Code Final	35
Tour d'eau	37
Planteur	37
Conclusions et recommandations	38
Bibliographie	39
APPENDICES	40
APPENDICE I: Manuel de l'utilisateur	40
Fonctions et capacités du produit	40
Remplir le réservoir	40
Utilisation du planteur	40
Utilisation de l'Arduino	41
Guide de sécurité	41
Dépannage	41
Questions ou commentaires?	42
APPENDICE II: Fichiers de conception	43

Liste de figures

Figure 1 : Planteur carré	12
Figure 2 : Planteur carré avec porte	13
Figure 3 : Planteur rectangulaire courte	13
Figure 4 : Système d'arrosage à l'aide d'une pompe	14
Figure 5 : Système d'arrosage à l'aide de la gravité	15
Figure 6 : Système d'arrosage attachés au système d'eau de la ville	15
Figure 7 : Surveillance du sol à l'aide d'un capteur d'humidité du sol	16
Figure 8 : Concept I	17
Figure 9 : Concept II	17
Figure 10 : Concept III	18
Figure 11 : Concept choisi	20
Figure 12 : List des tâches	21
Figure 13 : Diagramme de Gantt	21
Figure 14 : Prototype I	23
Figure 15 : Prototype II	24
Figure 16 : Code de prototype II	25
Figure 17 : Prototype II en action	25
Figure 18 : Prototype III	26
Figure 19 : Code complet	27
Figure 20 : Vue de face du produit final	28
Figure 21 : Vue oblique du produit final	29
Figure 22 : La connection du réservoir d'eau	29
Figure 23 : Planteur	30
Figure 24 : Boîte électronique	30
Figure 25 : Circuit électrique	32
Figure 26 : Tour d'eau	37
Figure 27 : Vue du dessus du planteur	38
Figure 28 : Dépannage d'Arduino	42

Liste de tables

Table 1 : Besoin du client	8
Table 2 : Etalonnage de trois planteur	9
Table 3 : Quantification de l'étalonnage des trois planteurs existantes	10
Table 4 : Spécifications cibles	11
Table 5 : Etalonnage des trois concepts	18
Table 6 : Quantification de l'étalonnage des trois concepts	19
Table 7 : Nomenclature des matériaux	22
Table 8 : Nomenclature des équipements	23
Table 9 : Fonctions des composante électrique	40
Table 10 : Dimension de la jardinière	40

Liste d'acronymes

Acronyme	Définition
\$CAN	Dollar canadien
=	Égal
<	Plus petit
>	Plus grand
AE	Assistant-enseignant
cm	Centimètre
ft	Pieds
kg	Kilogramme
L	Litre
lbs	Livres
LDM	Liste des matériaux
LED	Light-emitting diode (Diode électro-luminescente)
MPa	MégaPascal
N°	Numéro
N/A	Non applicable
PCB	Printed circuit board (carte de circuit imprimé)
PLA	Acide polylactique ou polylactide
po	Pouces
Servo	Servomoteur
V	Volt

1 Introduction

En octobre 2012, le pavillon des sciences sociales de l'Université d'Ottawa a ouvert ses portes au public.¹ Ce fut le premier bâtiment à avoir un "mur vivant". Depuis l'ajout de ce mur, les étudiants et les facultés souhaitent voir plus de verdure autour du campus. Cependant, l'ajout de plantes autour du campus pourrait être très coûteux si l'université devait embaucher une entreprise extérieure pour s'occuper régulièrement de ces plantes. La pandémie de covid-19 nous a appris que même si l'université peut engager une entreprise pour s'occuper des plantes, il y aura des moments où l'entreprise ne sera pas en mesure de s'occuper des plantes.

Dans le cadre du cours GNG1503 (Génie de la conception), nous avons relevé le défi de concevoir une jardinière d'intérieur pour les espaces communs de l'Université d'Ottawa pour notre client Jonathan Rausseo, gestionnaire du développement durable. Monsieur Rausseo nous a expliqué qu'il y a un manque de verdure sur le campus et que les êtres humains sont biophiles, qui ont donc tendance à interagir avec la nature.² Il aimerait apporter plus de verdure dans les espaces communs à l'aide de jardinières autosuffisantes, qui peuvent se maintenir sans aide extérieure.³

2 Identification des besoins et processus de spécification du produit

2.1 Énoncé du problème

L'Université d'Ottawa veut mettre des jardinières, autosuffisantes, nécessitant un entretien minimal et ayant un bel esthétique, dans les espaces communs pour ajouter de la verdure au campus et pour aider à améliorer la moralité des étudiants et des professeurs.

2.2 Besoins

Lors de la première rencontre avec le client, il nous a exprimé ce dont la jardinière avait besoin. Voici la liste des besoins que nous avons créés. En fonction de ces besoins, nous avons élaboré des critères de conception.

Table 1 : Besoins du client

N°	Besoins	Critères
1	Les planteurs comprennent un système d'arrosage automatique et/ou un système de surveillance du sol.	Système d'arrosage, Système de surveillance du sol

¹ "Social Sciences Building." Faculty of Social Sciences, 2012, socialsciences.uottawa.ca/about-faculty/fss-building. Consulté le 30 novembre 2020.

² "Biophilia." Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/biophilia>. Consulté le 30 novembre 2020.

³ "Self-sufficient." Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/self-sufficient>. Consulté le 30 novembre 2020.

2	Les planteurs devraient être d'un coût raisonnable.	Coût (\$CAN)
3	Les planteurs devraient être faciles à nettoyer.	Matériaux
4	Les planteurs devraient être d'une taille raisonnable pour un espace commun.	Dimensions (ft)
5	Les planteurs devraient avoir une esthétique simple et moderne.	Style
6	Les planteurs devraient donner l'impression qu'ils sont difficiles à voler pour ne pas que les gens pensent à essayer de les déplacer ou de les voler mais ils devraient être déplacés assez facilement.	Poids (kg)
7	Les planteurs devraient être résistants au feu et durables.	Matériaux

Voici d'autres besoins prescrits et autres pas nécessairement prescrits par le client, regroupés sous trois catégories.

- Exigences fonctionnelles
 - Système d'arrosage
 - Système de surveillance du sol
 - Système pour le déplacer
 - Temps d'entretien
- Exigences non fonctionnelles
 - Style
 - Matériaux
 - Durée de vie
 - Fiabilité
- Contraintes
 - Coût
 - Poids
 - Dimensions

2.3 Étalonnage

Afin d'avoir une idée de ce qui est sur le marché, nous avons effectué un étalonnage de trois produits. Cet étalonnage nous aide à identifier les forces et les faiblesses des produits.

Table 2 : Étalonnage de trois planteur

Jardinière d'intérieur




Spécifications	Jardinière Lechuza Classico Color 21	AeroGarden Harvest	Grapevine Urban Garden Planter
Compagnie	Lechuza 	AeroGarden 	The Home Depot 
Coût	67.34 \$CAD	229.95 \$CAD	119.00 \$CAD
Matériaux	Plastique PP	acier inoxydable	Palettes de pin recyclées écologiques et l'acier peint par poudrage noir
Dimensions	9 x 9 x 8 po	11 x 8 x 15 cm	37 x 12.6 x 15.35 po
Système d'arrosage	Système d'auto-arrosage + indicateur de niveau d'eau	Système d'auto-arrosage	N/A
Système de surveillance du sol	N/A	N/A	N/A
Style	Simple, propre	Simple, propre	Classique, plein air
Poids	2.25 lbs	5.99 lbs	24.52 lbs

Table 3 : Quantification de l'étalonnage des trois planteurs existantes

Spécifications	Importance	Jardinière Lechuza Classico Color 21	AeroGarden Harvest	Grapevine Urban Garden Planter
Compagnie		Lechuza	AeroGarden	The Home Depot
Coût	4	3	1	2
Matériaux	4	2	3	1
Dimension	3	1	2	3
Système d'arrosage	5	3	2	1
Système de surveillance du sol	5	1	1	1

sol				
Style	4	3	1	2
Pois	4	1	2	3
Total		59	49	51

Une fois que l'étalonnage est fait, nous avons déterminé que le planteur de Lechuza est le meilleur. Même si ce planteur avait le plus de points, il n'accomplit pas toutes les exigences de notre client. Nous avons étudié la jardinière et pris en note quelques idées, mais notre jardinière finale est très différente.

2.4 Spécifications cibles

Afin de nous aider à créer notre jardinière, nous avons déterminé des spécifications cibles.

Table 4 : Spécifications cibles

Critères de conception	Relation (=,<,>)	Valeur	Unités	Méthode de vérification
Exigences fonctionnelles				
1. Système d'arrosage	=	Oui	N/A	Essai
2. Système de surveillance du sol	=	Oui	N/A	Essai
3. Système pour le déplacer	=	Oui	N/A	Essai, Analyse
4. Temps d'entretien	<	10	Minutes	Essai
Exigences non fonctionnelles				
1. Style	=	Oui	N/A	Essai
2. Matériaux	=	Oui	N/A	Essai
3. Durée de vie	>	5	Années	Essai
4. Fiabilité	=	Oui	N/A	Essai
Contraintes				
1. Coût	<	100	\$CAD	Estimation, vérification finale

2. Poids	<	50	lbs	Analyse
3. Dimensions	<	3 x 4 x 3	ft	Analyse

3 Conceptualisation

Une fois que nous avons déterminé le problème, les besoins et les spécifications cibles, il est maintenant le temps de produire des concepts. Nous avons aussi créé des sous-systèmes pour notre jardinière finale. Ses sous-systèmes ont été créés à l'aide des besoins du client. Les sous-systèmes sont le planteur, le système d'arrosage et le système de surveillance du sol. Au cours de notre projet, certains de nos sous-systèmes ont évolué de leur concept préliminaire.

3.1 Sous-système 1 : Le planteur

Le premier sous-système que nous avons choisi est le planteur. Le planteur est l'endroit où les plantes sont situées. Lors de la création du planteur, nous devons garder en tête le style du planteur, car c'est important pour le client. Il aimerait que le planteur ait un style plus simple et moderne pour plaire au plus grand nombre de personnes.

3.1.1 Concept 1 : Planteur Carré

Ce planteur est de forme carrée de dimension 4x6x4ft. Ce planteur serait construit en bois. La forme et le matériel lui donneront une esthétique simplistique. Ce design nous permettra de mettre les composantes électrique et un réservoir d'eau sous les plantes. Voir *Figure 1*.

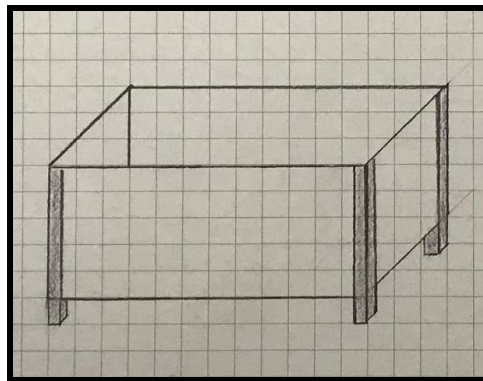


Figure 1 : Planteur carré

Les avantages de ce planteur sont que l'on peut cacher des composantes sous le planteur et il correspond au style souhaité par le client. Par contre, puisqu'il sera fait de bois, il a la possibilité d'écrouissage car il sera en contact avec un sol humide. De plus, il n'y a pas de solutions pour accéder aux choses sous les plantes.

3.1.2 Concept 2 : Planteur carré avec porte

Ce planteur est de forme carrée de dimension 4x6x4ft. Lui aussi serait fait de bois. En dessous des plantes, il y a un entrepôt pour cacher les composantes électriques et d'autres choses pour prendre soin des plantes. La forme et le matériel lui donneront une esthétique moderne. Voir *Figure 2*.

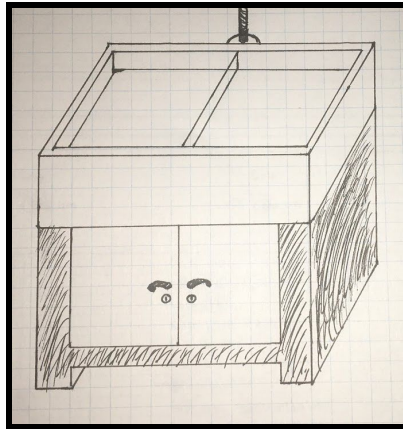


Figure 2 : Planteur carré avec porte

Les avantages de ce planteur sont que l'on peut cacher des choses sous le planteur et il correspond au style souhaité par le client. Par contre, puisqu'il sera fait de bois, il y a la possibilité d'écume car il sera en contact avec un sol humide.

3.1.3 Concept 3 : Planteur rectangulaire courte

Ce planteur est de forme rectangulaire de dimension 4x6x2ft. Lui aussi serait fait de bois. La forme et le matériel lui donneront une esthétique rustique. Voir *Figure 3*.

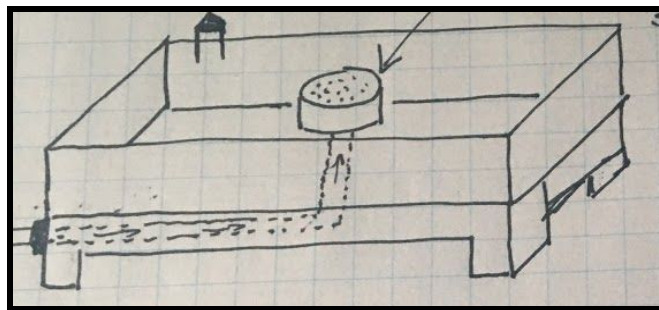


Figure 3 : Planteur rectangulaire courte

Les avantages de ce planteur est qu'il n'est pas très haut. Par contre, puisqu'il sera fait de bois, il y a la possibilité d'écume car il sera en contact avec un sol humide. De plus, il n'y a pas d'entrepôt pour cacher des composantes et n'est pas dans le style que le client voulait.

3.2 Sous-système 2 : Le système d'arrosage

Le deuxième sous-système que nous avons choisi est le système d'arrosage des plantes. Ce sous-système arrosera les plantes automatiquement. Lors de la création du système d'arrosage, nous devons garder en tête la fréquence d'arrosage des plantes et s'il y a un réservoir d'eau, la fréquence à laquelle le réservoir doit être rempli, car c'est important pour le client.

3.2.1 Concept 1 : Pompe

La pompe est un sous-système qui pompe l'eau d'un réservoir à travers un tuyau jusqu'aux plantes. La section de tuyau qui sera sous les plantes sera percée pour arroser les plantes. La pompe sera connectée à un Arduino qui allumera la pompe lorsque le sol est trop sec. Voir *Figure 4*.

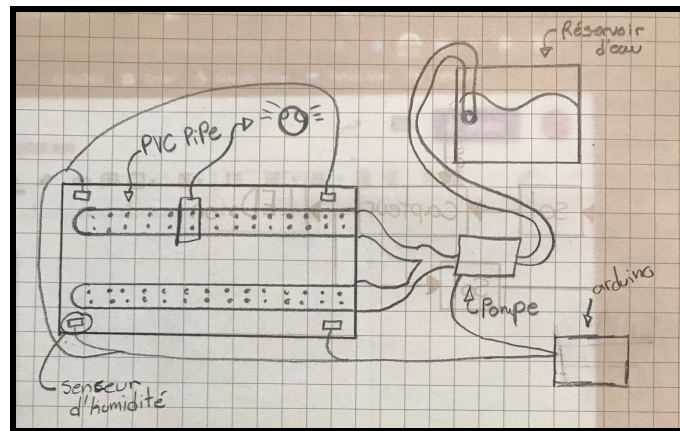


Figure 4 : Système d'arrosage à l'aide d'une pompe

Les avantages de la pompe est qu'elle peut fonctionner avec un Arduino ce qui rendrait le système d'arrosage automatique. Par contre, les pompes peuvent être très dispendieuses.

3.2.2 Concept 2 : Gravité

Dans ce concept, le réservoir d'eau serait situé plus haut que les plantes pour que l'eau puisse couler à l'aide de la gravité. À l'aide de l'Arduino, on utilise un interrupteur électronique pour laisser l'eau couler lorsque le sol est sec. Lorsque l'interrupteur est ouvert l'eau pourrait couler à travers du tuyau jusqu'aux plantes à l'aide de la gravité. La section de tuyau qui sera sous les plantes sera percée pour arroser les plantes. Voir *Figure 5*.



Figure 5 : Système d'arrosage à l'aide de la gravité

Les avantages d'un système qui coule par gravité sont que ça consomme moins d'électricité, la tour peut être déplacée et elle est éducative. De plus, la tour pourrait être incorporée à des plantes grimpantes. Cependant, la tour peut être assez massive.

3.2.3 Concept 3 : Système d'eau de la ville

Dans ce concept, il n'y a pas de réservoir d'eau. Ce système se compose d'un boyau relié au système d'eau du bâtiment. À l'aide de l'Arduino on utilise un interrupteur électronique pour laisser l'eau couler lorsque le sol est sec. Lorsque l'interrupteur est ouvert, l'eau pourrait couler à travers du boyau jusqu'aux plantes. Les plantes seront arrosées par un arrosoir au centre du planteur. Voir Figure 6.

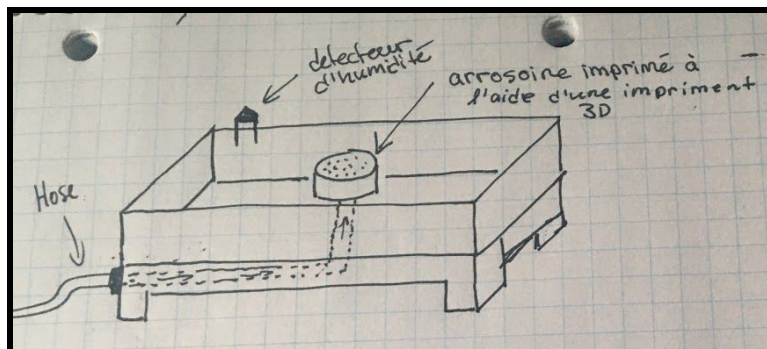


Figure 6 : Système d'arrosage attachés au système d'eau de la ville

Les avantages du système sont qu'il ne contient pas de réservoir d'eau, donc personne n'a besoin de remplir constamment un réservoir d'eau. Mais, il sera difficile de trouver une source d'eau dans un air commune pour chaque planteur.

3.3 Sous-système 3 : Le système de surveillance du sol

Le troisième sous-système que nous avons choisi est le système de surveillance du sol. Ce sous-système vérifie l'humidité du sol automatiquement. Lors de la création du système de surveillance du sol, nous devons garder en tête que différentes plantes ont besoin de différents niveaux d'humidité.

3.3.1 Concept 1 : Capteur d'humidité du sol

Le capteur d'humidité du sol est un sous-système qui détectera si le sol est sec ou non. Ce système fonctionne à l'aide d'un Arduino qui lira les données du capteur. Si le capteur sent que le sol est trop sec, il activerait le système d'arrosage du sol. Voir *Figure 7*.

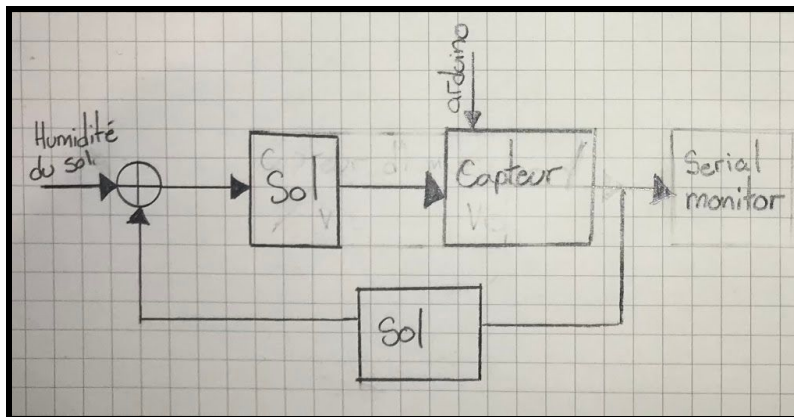


Figure 7 : Surveillance du sol à l'aide d'un capteur d'humidité du sol

Les avantages de ce système est qu'il peut être modifié selon les besoins des plantes.

3.4 Les trois systèmes

Voici les trois systèmes. Le premier système est une combinaison du planteur carré, de la pompe et du capteur d'humidité du sol. Le deuxième système est une combinaison du planteur carré avec porte, le système par gravité et du capteur d'humidité du sol. Le troisième système est une combinaison du planteur rectangulaire court, du système d'eau de ville et du capteur d'humidité du sol. Voir *Figures 8-9-10*.

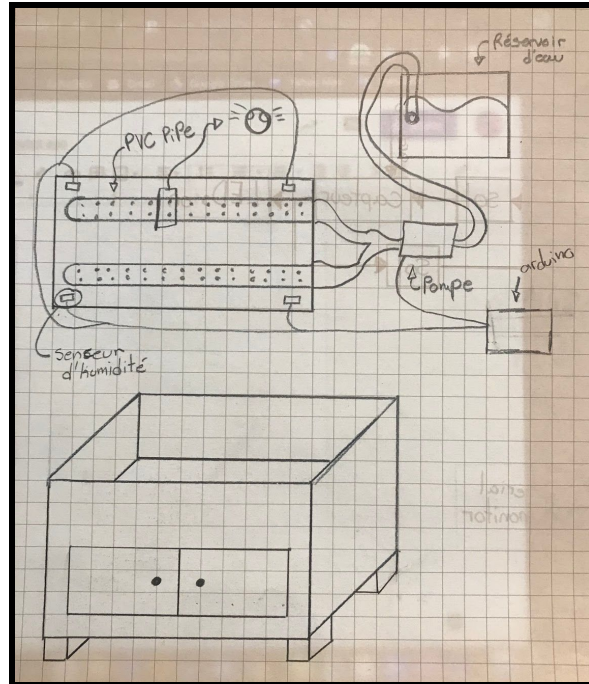


Figure 8 : Concept I

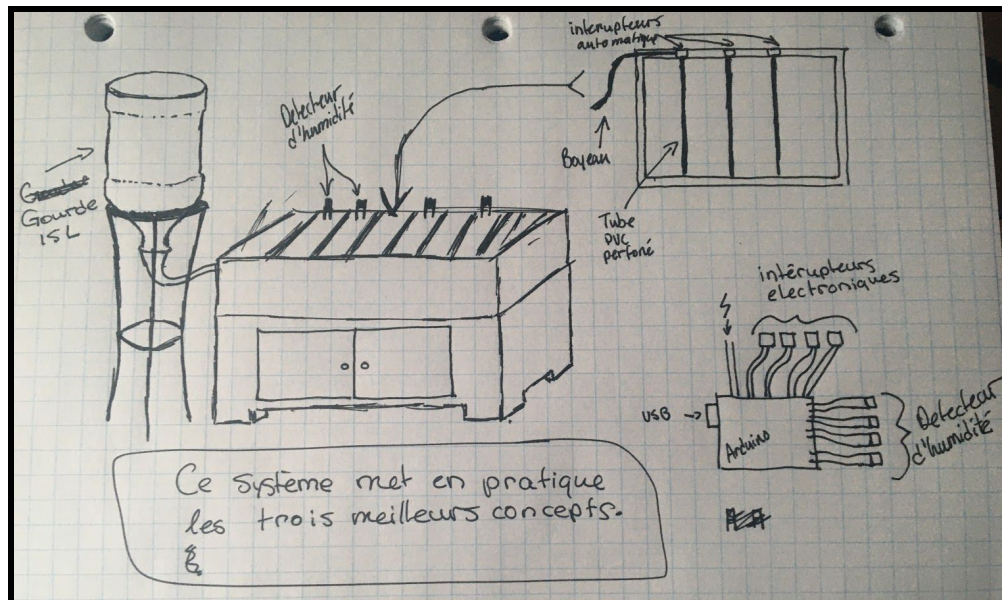
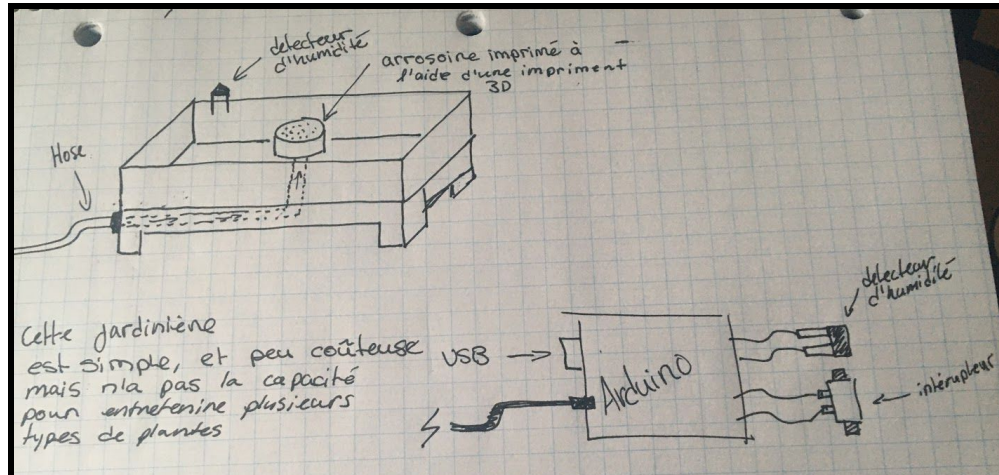


Figure 9 : Concept II



3.5 Étalonnage et choix de concept

Une fois que nous avons généré trois systèmes, nous avons effectué un étalonnage pour déterminer quel est le meilleur système.

Table 5 : Étalonnage des trois concepts

Descriptions des concepts de jardinières d'intérieur			
Spécifications	Systèmes I	Systèmes II	Systèmes III
Estimation relative des Matériaux	On a besoin de: <ul style="list-style-type: none"> - Bois - PVC - Arduino - Interrupteurs d'eau - Détecteur d'humidité - Fils électrique - Pompe à eau - Tank - Tarp 	On a besoin de: <ul style="list-style-type: none"> - Bois - PVC - Arduino - Interrupteurs d'eau - Détecteur d'humidité - Fils électrique - Gourde - Flex seal - Filament PLA 	On a besoin de: <ul style="list-style-type: none"> - Bois - PVC - Arduino - Interrupteurs d'eau - Détecteur d'humidité - Fils électrique - Tarp - Filament PLA
Estimation relative du Coût	200\$	100\$	150\$
Estimation relative des Dimensions	4x6x4 ft	4x6x4 ft	4x6x2 ft

Système d'arrosage	Pompe	Gravité	Système d'eau de la ville
Système de surveillance du sol	Détecteur d'humidité	Détecteur d'humidité	Détecteur d'humidité
Style	Simplistique	Moderne	Rustique
Estimation relative du Poids	150-250 lbs	250-300 lbs	100-150 lbs

Table 6 : Quantification de l'étalonnage des trois concepts

Spécifications	Avantages de chaque ensemble de systèmes			
	Importance	Systèmes I	Systèmes II	Systèmes III
Estimation relative des Matériaux	4	2	3	1
Estimation relative du Coût	4	1	3	2
Estimation relative des Dimensions	3	3	3	3
Système d'arrosage	5	2	3	1
Système de surveillance du sol	5	3	3	3
Style	4	2	3	1
Estimation relative du Poids	4	3	2	1
Total		66	83	49

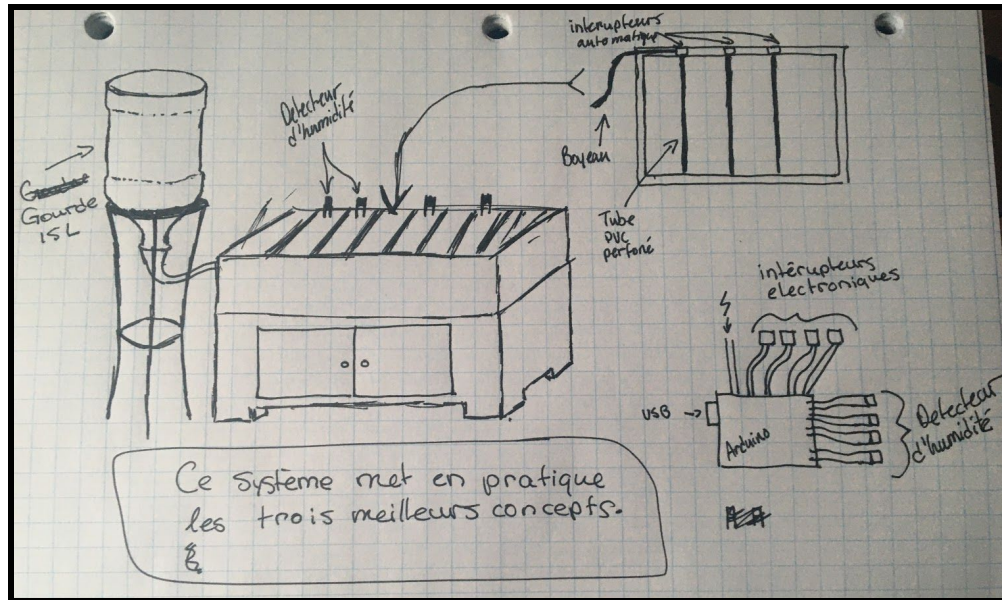


Figure 11 : Concept choisi

Finally, the best system was system II. This system is the best because it contains all the most convenient systems for the user, it does not need much maintenance, and it is cheap. In addition, its automatic system makes it possible to leave it for long periods of time without supervision. Its irrigation system ensures that all the regions of the garden bed are watered in an equivalent way. The soil monitoring system is the key to independence. With a little programming, the Arduino will always know how much water should be distributed so that the plants remain beautiful and healthy. Throughout our project, this concept evolved during the prototyping phase.

4 Plan du projet

4.1 Plan du projet

Once we had decided on the concept we were going to implement, it was time to create a task plan. This plan helps us to organize and complete all our tasks before their due date. We built this Gantt diagram in addition to a list of tasks.



Figure 12 : Liste des tâches

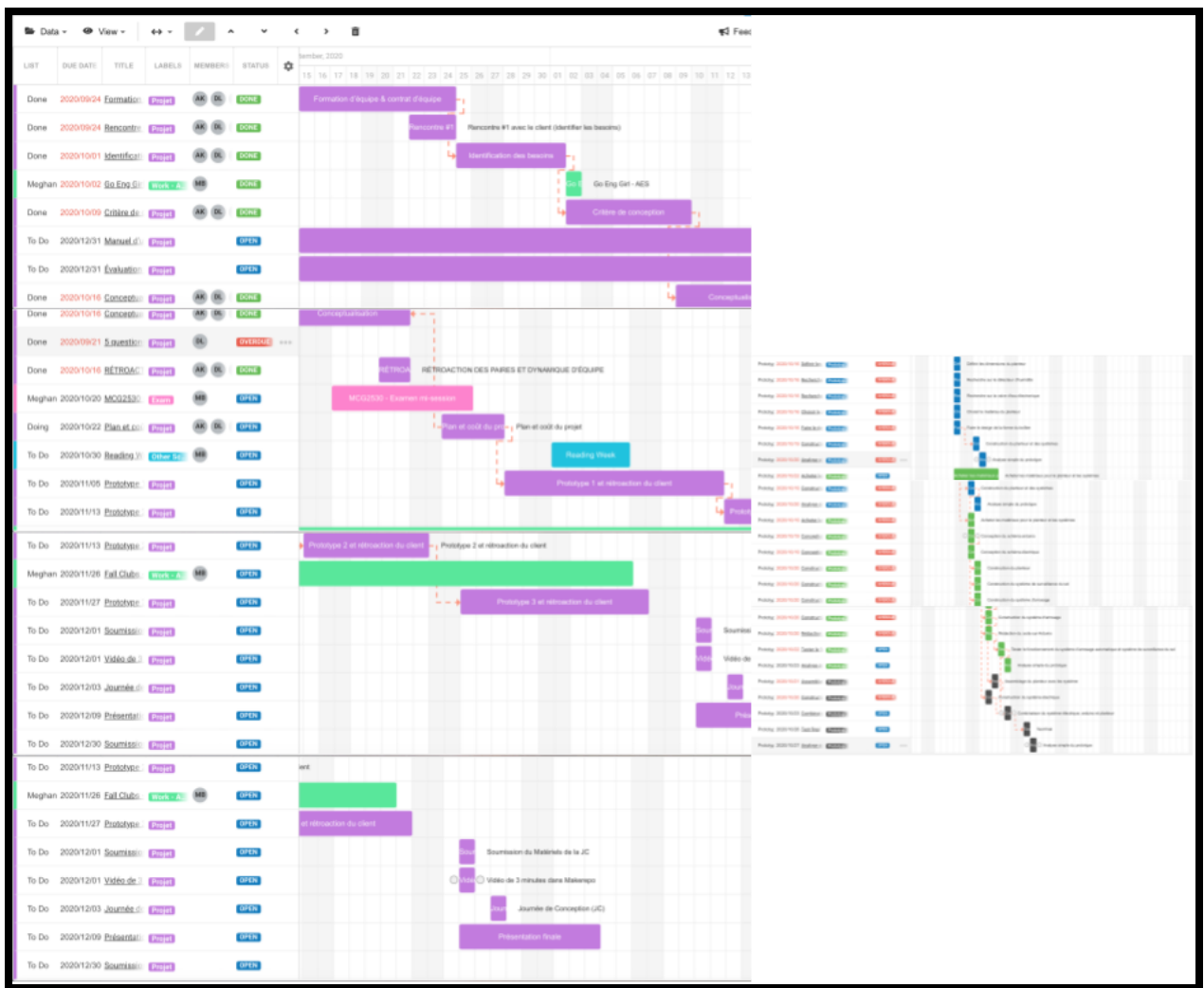


Figure 13 : Diagramme de Gantt

4.2 LDM

Voici une liste des matériaux que nous allons utiliser pour ce projet. Au cours de ce projet, cette liste a changé pour qu'on puisse s'adapter une fois qu'un problème est arrivé. Vous trouverez la liste des matériaux finaux ci-dessous.

Table 7 : Nomenclature des matériaux

Nomenclature des matériaux					
N°	Description du composant	Quantité	Prix unitaire	Montant TTC (CDN\$)	Fournisseur
1	Capteur de débit d'eau	1	10.99\$	10.99\$	Amazon
2	Capteur d'humidité du sol	1	11.99\$	11.99\$	Amazon
3	Sac de culture	1	16.72\$	16.72\$	Amazon
4	Bois recyclé	N/A	0\$	0\$	Tyler
5	Tuyau d'irrigation	1	6.58\$	6.58\$	Home Depot
6	Pichet à eau de 15L	1	0\$	0\$	Tyler
7	½ laiton - ⅜ raccord de tube barbelé	3	2.51\$	7.53\$	Home Depot
8	½ collier de serrage	2	2.21\$	4.42\$	Home Depot
9	Arduino Uno	1	0\$	0\$	L'Université
10	Interrupteur d'eau	1	13.40\$	13.40\$	Home Depot
11	Servo	1	25.99\$	25.99\$	Amazon
Total				99.62\$	

4.3 Liste d'équipements

Voici une liste des équipements et logiciels que nous planifions utiliser pour créer nos prototypes.

Table 8 : Nomenclature des équipements

Nomenclature des équipements	
N°	Description de l'équipement
1	SolidWorks
2	Arduino
3	Soudure
4	Imprimante 3D
5	Outil divers (scie, tournevis, perceuse, etc.)

5 Prototypage, essais et validation du client

Avant de pouvoir arriver à notre produit final, nous devons construire plusieurs prototypes. Notre premier prototype se concentre sur l'agencement des sous-systèmes de notre planteur. Le deuxième prototype se concentre sur la fonctionnalité du système de surveillance du sol. Le troisième prototype se concentre sur la fonctionnalité de tous les sous-systèmes critiques ensemble pour former une jardinière d'intérieur fonctionnelle.

5.1 Prototype I

L'objectif de ce premier prototype était de trouver l'agencement des sous-systèmes de notre planteur. Nous avons décidé de créer ce prototype à l'aide du logiciel SolidWorks. Ceci nous a permis de facilement faire des modifications des mesures et de déplacer les divers sous-systèmes. De plus, nous avons monté ce prototype au client pour qu'il puisse nous donner de la rétroaction. Voir *Figure 14*.

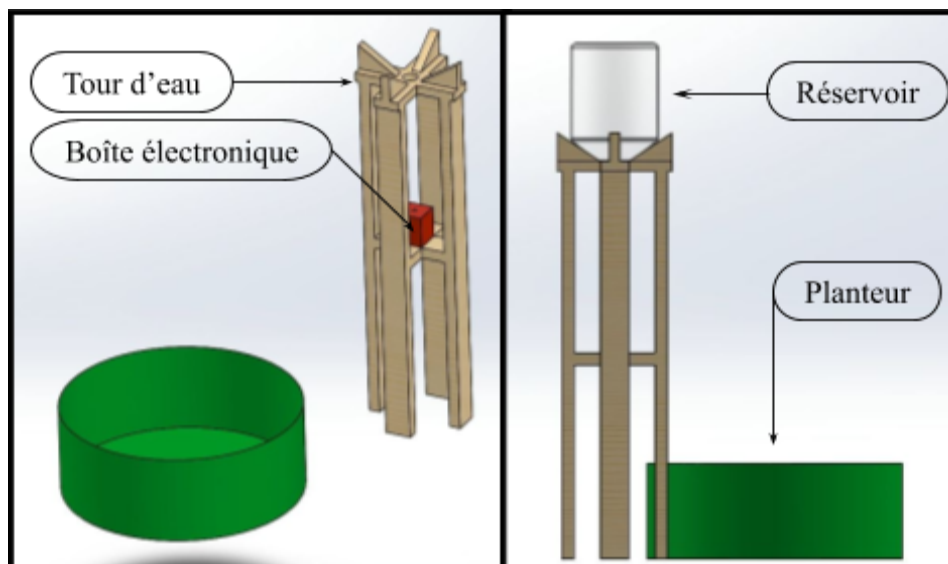


Figure 14 : Prototype I

Le client a vraiment aimé le prototype. Il a apprécié le fait que la jardinière puisse être programmée en fonction des besoins des plantes. De plus, il a aimé que le concept puisse devenir plus compact en déplaçant la tour d'eau à l'intérieur de la jardinière. La tour peut aussi être cachée par des plantes grimpantes. Par contre, il était concerné par la hauteur de la tour d'eau. Il voulait que l'on garde en considération les personnes qui rempliront le réservoir et leur grandeur.

5.2 Prototype II

L'objectif de ce deuxième prototype était de faire fonctionner le système de surveillance du sol. Nous avons décidé de créer ce prototype à l'aide d'un Arduino, un plaque de PCB, un capteur d'humidité et d'une soudeuse. Pour ce prototype nous avons créé un code qui dit à l'Arduino d'imprimer l'humidité du sol sur le moniteur en série. Voir Figures 15-16-17.

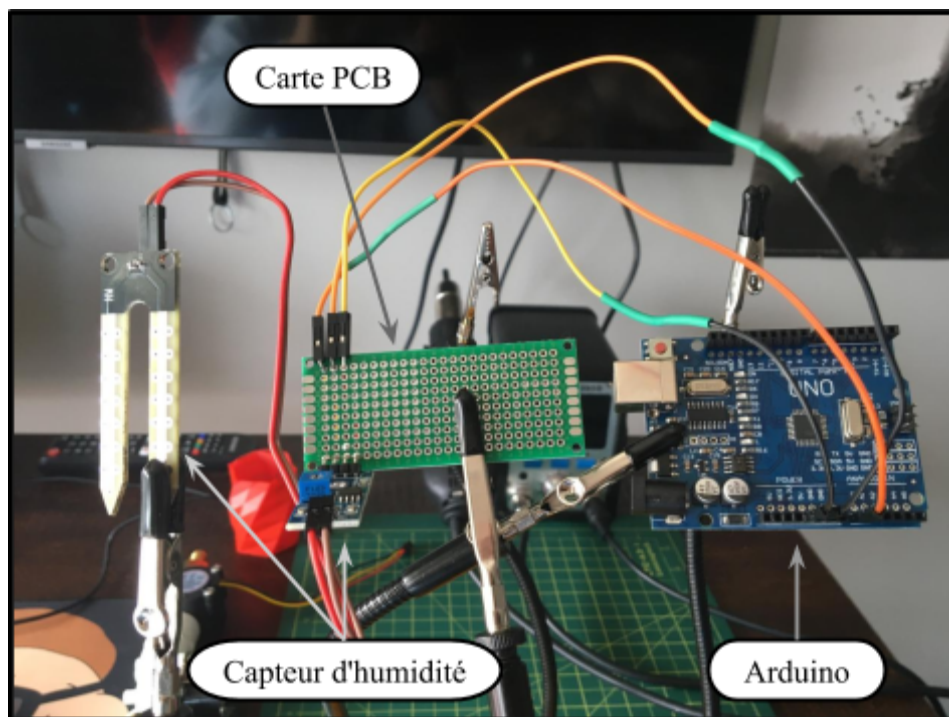


Figure 15 : Prototype II

```
Soil_Moisture_Sensor_1.0 | Arduino 1.8.13
Soil_Moisture_Sensor_1.0
//Soil Moisture Sensor
//NOTE: The lower the number is on the SM, the more the soil is moist

// Sensor set up
int sensorPin = A0;
int sensorValue;
int limit = 1000; //Test value

void setup()
{
  Serial.begin(9600); //Serial Monitor
  pinMode(13, OUTPUT); //LED
}

void loop()
{
  //Serial Monitor
  sensorValue = analogRead(sensorPin);
  Serial.println("Analog Value : ");
  Serial.println(sensorValue);

  if (sensorValue > limit) // If value of soil lower then limit
  {
    digitalWrite(13, HIGH); // LED high
  }
  else // If value of soil higher then limit
  {
    digitalWrite(13, LOW); // LED low
  }

  delay(1000);
}

Done Saving.

20 Arduino Uno on /dev/cu.usbserial-1410
```

Figure 16 : Code de prototype II

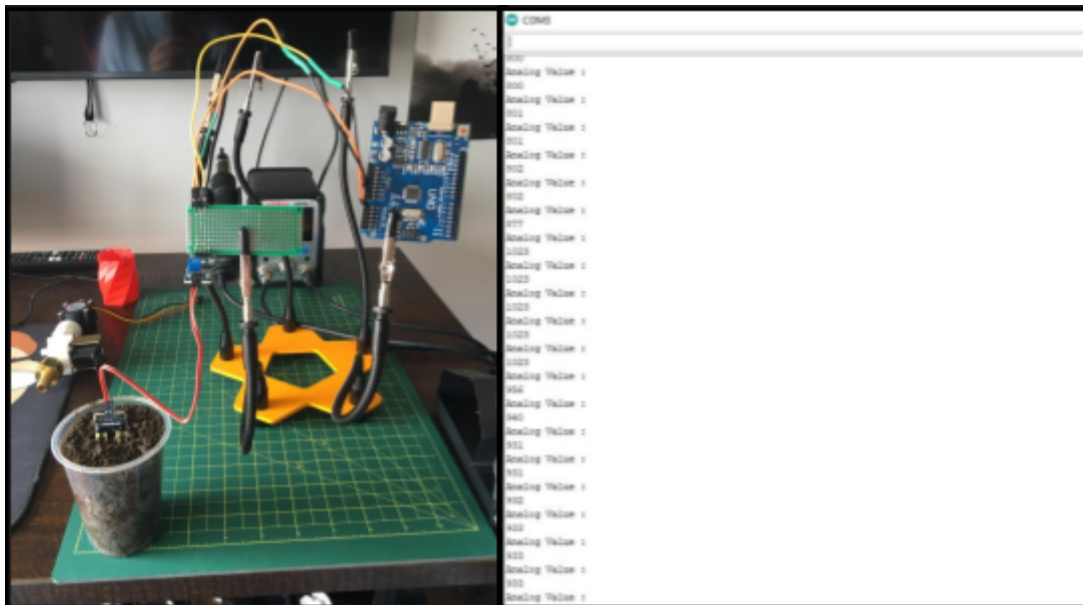


Figure 17 : Prototype II en action

Le code est simple et a bien fonctionné. Les données du capteur sont faciles à lire, plus la valeur est haute, plus le sol est sec. Le système d'électronique est très similaire au produit final. Par contre, les fils devront être allongés pour le produit final.

5.3 Prototype III

L'objectif de ce dernier prototype était de faire fonctionner tous les sous-systèmes critiques ensemble pour former une jardinière d'intérieur fonctionnelle. Nous avons décidé de créer ce prototype à l'aide d'un Arduino, un plaque de PCB, un capteur d'humidité, un servo, un interrupteur d'eau, un capteur de débit, du bois, un tuyau d'irrigation, un imprimante 3D et d'une soudeuse. Pour ce prototype nous avons créé un code qui dit à l'Arduino si le capteur d'humidité du sol détecte que l'humidité dans le sol est trop faible, l'interrupteur s'ouvre et laisse de l'eau couler à travers le capteur de débit et arrosera les plantes, une fois que le capteur de débit lit un certain débit, l'interrupteur se ferme. Voir *Figures 18-19*.

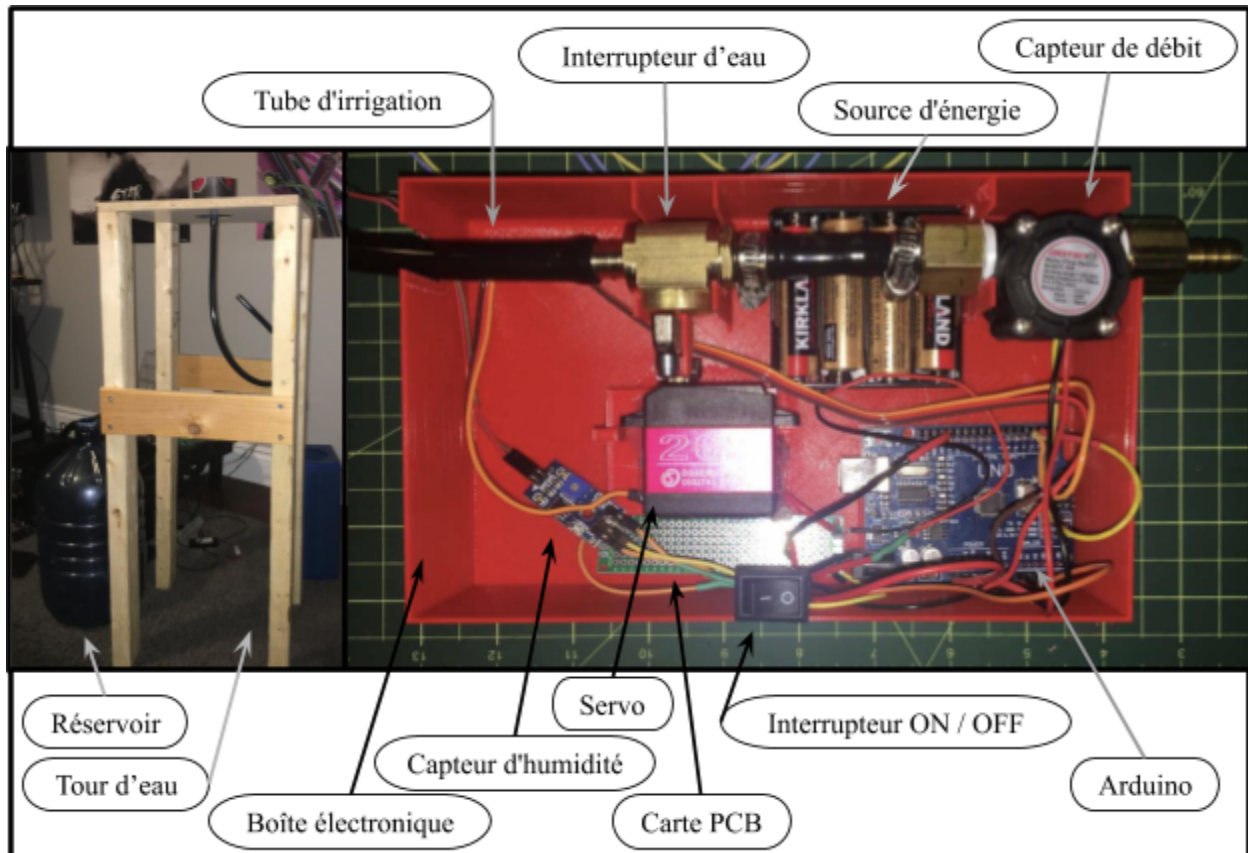


Figure 18 : Prototype III

```

The_Code | Arduino 1.8.13
The_Code
#include <Servo.h>

//Soil Moisture Sensor
int sensorPin = A0;
int moistureValue;
int limit = 500; //Test value. This is the value to change to set the min moisture level of the plants

//Servo
Servo servo; //Pin 4 digital

// Flow Sensor
int PulsePin = 3; //Arduino digital pin 3 is declared as Pulse
int led = 13; //led pin
int value = 250; //Test value. This is the value to change to set the max amount of water the plants need 500
volatile int pulseCount; //Variable to count number of pulses
volatile int i = 0; //Variable for led status

void setup()
{
  //Soil Moisture Sensor N/A

  //Servo
  servo.attach(4);
  servo.write(180);
  delay(2000);

  // Flow Sensor
  pinMode(PulsePin, INPUT_PULLUP); //Interrupt pin 3 declared as input and pull up resistor is enabled
  pinMode(led, OUTPUT); //led as output
  //Interrupt is attached to pin 3
  //ISR(Interrupt service routine) of interrupt is CountPulses
  //Interrupt is activated on HIGH to LOW transition

  //Serial Monitor
  Serial.begin(9600);
}

void loop()
{
  // Flow Sensor Skip
  pulseCount = 0; //Start counting from 0 again
  i = 0; //Reset led toggling variable
  digitalWrite(PulsePin, LOW); //Enables interrupt on arduino pin 3
  delay(2000); //Wait 1 second
  digitalWrite(PulsePin, HIGH); //Disable the interrupt

  //One second is over now and we have the number of pulses in variable
  //pulseCount

  //Calculating the water flow rate in Milli liters per minute
  double flowRate;
  flowRate = (pulseCount * 2.22 * 60);

  //THE CODE
  if (moistureValue == 1000) //If value of soil lower than limit
  {
    servo.write(0); //Switch Servo ON
    delay(2000);

    if (flowRate == value) //The amount of water we need to feed the plants
    {
      servo.write(180);
      delay(2000);
    }
  }
}

The_Code | Arduino 1.8.13
Sketch uses 5638 bytes (20% of program storage space. Maximum is 28672 bytes.
Global variables use 200 bytes (14% of dynamic memory, leaving 1700 bytes free).

The_Code
void setup()
{
  servo.attach(4);
  delay(2000);
}

void loop()
{
  // If 3f value of soil higher than limit
  {
    servo.write(180); //Switch Servo ON
    delay(2000);
  }

  //Serial Monitor
  //Soil Moisture Sensor
  moistureValue = analogRead(sensorPin);
  Serial.println("Soil Moisture Value:");
  // Flow Sensor
  Serial.println("Pulse count:");
  Serial.println(flowRate); //Print milli liters per minute on serial monitor
  Serial.println("led:");
}

//for the Flow Sensor
void CountPulses()
{
  pulseCount++; //Increment the variable in every pulse
  //On every pulse toggle the state of led.
  if (i % 2 == 0)
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);
}

The_Code | Arduino 1.8.13
Sketch uses 5638 bytes (20% of program storage space. Maximum is 28672 bytes.
Global variables use 200 bytes (14% of dynamic memory, leaving 1700 bytes free).

```

Figure 19 : Code complet

Durant ce prototype, nous avons rencontré un problème avec l'interrupteur électrique. Nous croyons qu'il était cassé quand nous l'avons obtenu. De plus, la pression de sortie de l'interrupteur n'était pas suffisante pour notre prototype. Nous avons remplacé cet interrupteur électrique par un interrupteur ordinaire et y avons attaché un servomoteur. Cela fonctionne exactement comme l'interrupteur électrique était censé fonctionner.

De plus, nous avons ajouté un capteur de débit à notre concept original. Après avoir discuté avec nos AEs, ils ont recommandé que nous ajoutons quelque chose pour rendre la quantité d'eau qui va aux plantes plus uniformes. Notre solution à cela était d'ajouter un capteur de débit. Ce capteur fait en sorte que les plantes reçoivent le même montant d'eau à chaque fois. Ceci a rendu notre système plus personnalisable pour les plantes et plus fiable.

Nous avons aussi créé une tour pour supporter la gourde d'eau. Elle a été construite uniquement en bois. En utilisant la gravité et un système électrique, l'eau sera facilement distribuée dans notre planteur. Pour attacher le pichet à eau de 15L à la tour, nous avons imprimé trois supports en plastique flexible à l'aide d'une imprimante 3D. Ce filament spécialisé est construit avec un plastique qui ressemble au caoutchouc, ceci fait en sorte que la pièce finale sera flexible et durable. Vous trouverez un lien vers le fichier pour la boîte d'électronique et celui des supports à l'Appendice II.

Ce prototype final nous donne une idée d'à quoi notre produit final va ressembler. Ce prototype est notre produit final semi-assemblé. Pour avoir notre produit final, nous devons peindre la tour d'eau pour lui donner une esthétique plus moderne. De plus, nous devons aussi combiner la tour complétée, le planteur et le système électronique.

6 Solution Finale

Voici notre produit final de Green Tide. Voir *Figures 20-24*.



Figure 20 : Vue de face du produit final



Figure 21 : Vue oblique du produit final



Figure 22 : La connection du réservoir d'eau



Figure 23 : Planteur

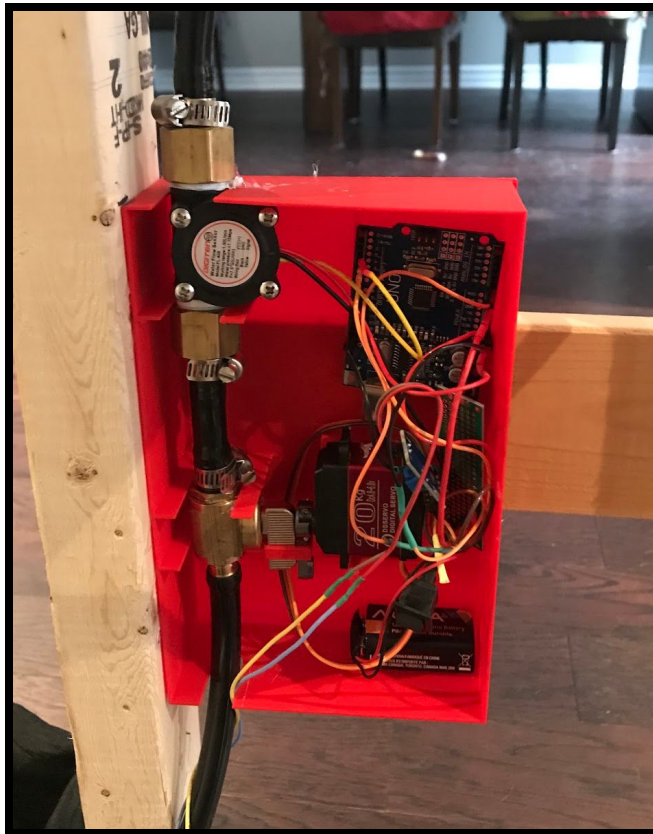


Figure 24 : Boîte électronique

7 Analyse

7.1 Nomenclature des matériaux

Voir *Table 7*.

7.2 Choix des composants électriques

7.2.1 Capteur d'humidité du sol

Un capteur d'humidité du sol détermine la quantité d'humidité dans le sol en mesurant la résistance entre les deux sondes métalliques qui sont insérées dans le sol.⁴ Nous avons choisi d'utiliser le capteur par KEEYEES. Ce capteur peut produire des valeurs digitales et analogiques et est compatible avec Arduino et Raspberry pi. Son voltage de fonctionnement est compris entre 3,3V et 5V.

Le capteur d'humidité de sol choisi est donc celui-ci : <https://www.amazon.ca/KeeYees>

7.2.2 Servo

Un servo est constitué d'un système en boucle fermée. Un servomoteur peut être utilisé pour faire tourner des pièces d'une machine avec un rendement élevé et avec une grande précision.⁵ Nous avons choisi d'utiliser le servo par DS SERVO. Ce servo peut tourner jusqu'à 180 degrés. Il est également étanche. Son voltage de fonctionnement est de 5V mais il peut être donné 6,8V pour un couple maximal. Ce servo peut produire des valeurs digitales et est compatible avec Arduino.

Le servo choisi est donc celui-ci : <https://www.amazon.ca/gp/product>

⁴ "Soil Moisture Sensor Module - Protosupplies". 2020. ProtoSupplies. <https://protosupplies.com/product/soil-moisture-sensor-module/#:~:text=The%20Soil%20Moisture%20Sensor%20Module%20determines%20the%20amount,of%20some%20type%20when%20a%20plant%20needs%20watering.> Consulté le 4 décembre 2020.

⁵ "What Is A Servo Motor And How It Works? | Realpars", 2018. Gastreich, W., PLC Programming Courses for Beginners | RealPars. <https://realpars.com/servo-motor/>. Consulté le 4 décembre 2020.

7.2.3 Capteur de débit

Un capteur de débit surveille la quantité d'eau fournie et utilisée, donc le débit d'eau qui doit être distribué.⁶ Nous avons choisi d'utiliser le capteur de débit par DIGITEN. Il utilise un capteur à effet Hall. Il utilise un champ magnétique externe pour lire les valeurs.⁷ Son voltage de fonctionnement est compris entre de 3.5V et 12V. Ce capteur peut produire des valeurs digitales et est compatible avec Arduino et Raspberry pi.

Le capteur de débit choisi est donc celui-ci : <https://www.amazon.ca/Sensor-Flowmeter>

7.3 Circuit électrique

Voici le schéma électrique. Voir *Figure 25*.

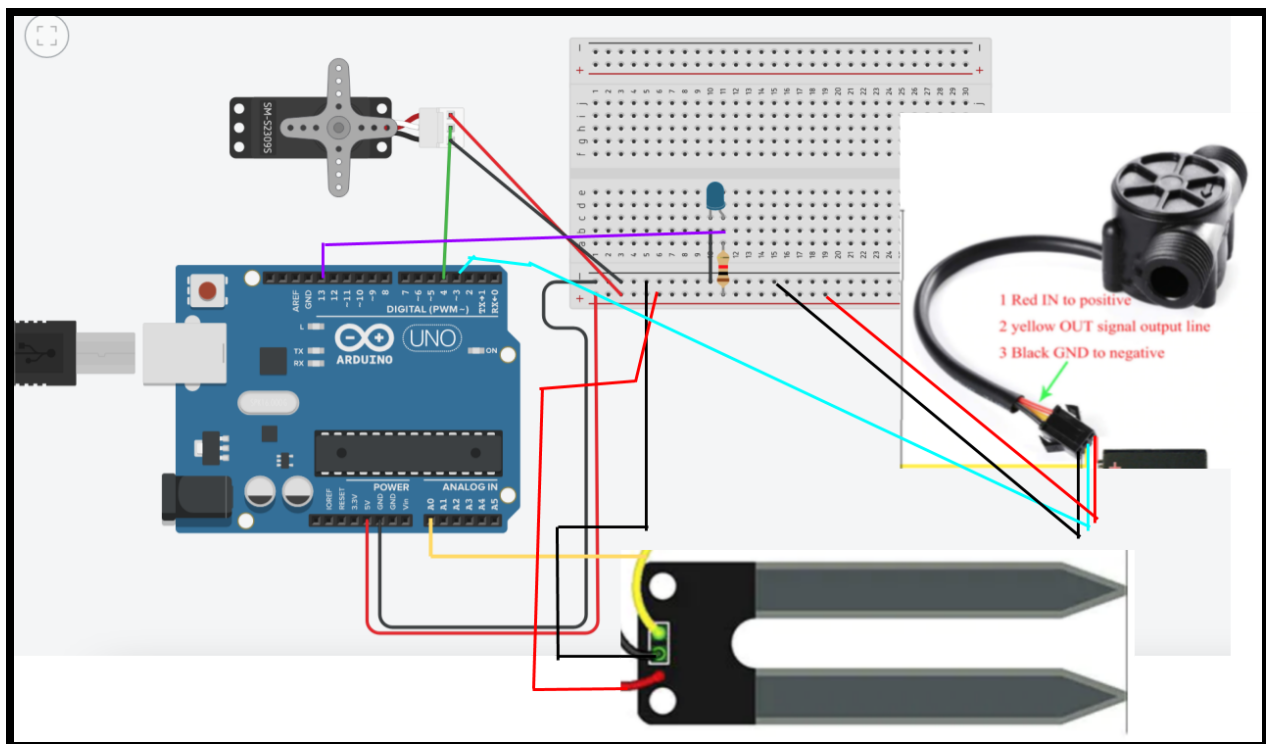


Figure 25 : Circuit électrique

⁶ “Water Flow Sensor - Working Principle , Applications & Examples.”. 2020. ElProCus - Electronic Projects for Engineering Students. <https://www.elprocus.com/a-memoir-on-water-flow-sensor/>. Consulté le 4 décembre 2020.

⁷ “Hall Effect Sensor”. 2020. <https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html>. Consulté le 4 décembre 2020.

7.4 Code du microcontrôleur (Arduino)

Étant donné que le circuit a quatre composantes de sortie, le code Arduino ne gèrera que deux broches: 3, 4, 13 et A0. Vous trouverez un lien vers le code final à l'*Appendice II*.

7.4.1 Partie I : Capteur d'humidité du sol

Le capteur d'humidité émet des valeurs analogiques. Il est branché dans la pin A0 et la LED est branchée dans la pin 13. Le code ci-dessous est un code qui teste la fonctionnalité du capteur d'humidité du sol. Le code dit si l'humidité du sol est plus basse que la limite, la LED va s'éteindre et si l'humidité du sol est plus haute que la limite, la LED va s'allumer. Aussi, il est programmé pour imprimer les valeurs du sol sur un moniteur en série. Vous trouverez un lien vers le code de test en l'*Appendice II*.

```
//Soil Moisture Sensor
//NOTE: The lower the number is on the SM, the more the soil is moist

// Sensor set up
int sensorPin = A0;
int sensorValue;
int limit = 1000; //Test value

void setup()
{
  Serial.begin(9600); //Serial Monitor
  pinMode(13, OUTPUT); //LED
}

void loop()
{
  //Serial Monitor
  sensorValue = analogRead(sensorPin);
  Serial.println("Analog Value : ");
  Serial.println(sensorValue);

  if (sensorValue > limit) // If value of soil lower then limit
  {
    digitalWrite(13, HIGH); // LED high
  }
  else // If value of soil higher then limit
  {
    digitalWrite(13, LOW); // LED low
  }

  delay(1000);
}
```

7.4.2 Partie II : Capteur de débit

Le capteur de débit émet des valeurs digitales. Il est branché dans la pin 3 et la LED est branchée dans la pin 13. Le code ci-dessous est un code qui teste la fonctionnalité du capteur de débit. Le code calcule le montant de débit à l'aide de l'équation ci-dessous, $flowRate=(pulsecount*2.22*60)$, (Débit=Constante*Unités de débit(ml / min)*Temps(secondes)). De plus, le code imprime les valeurs du débit sur un moniteur en série. Vous trouverez un lien vers le code de test à l'*Appendice II*.

```
// Flow Sensor
int Pulses = 3; //Arduino digital pin 3 is declared as Pulses
int led = 13; //Led pin
volatile int pulsecount; //Variable to count number of pulses
volatile int i = 0; //Variable for led status

void setup()
{
  pinMode(Pulses, INPUT_PULLUP); //Interrupt pin 3 declared as input and pull up resistor is enabled
  pinMode(led, OUTPUT); //Led as output
  //Interrupt is attached to pin 3
  //ISR(interuupt service routine) of interrupt is CountPulses
  //Interrupt is activated on HIGH to LOW transition
  attachInterrupt(digitalPinToInterrupt(Pulses), CountPulses, FALLING);
  Serial.begin(9600); //Starting serial monitor
}
void loop()
{
  pulsecount = 0; //Start counting from 0 again
  i = 0; //Reset led toggling variable
  interrupts(); //Enables interrupt on arduino pin 3
  delay (1000); //Wait 1 second
  noInterrupts(); //Disable the interrupt

  //One second is over now and we have the number of pulses in variable
  //'pulsecount'

  //Calculating the water flow rate in Milli Liters per minute
  double flowRate;
  flowRate = (pulsecount * 2.22 * 60);

  Serial.print("Flow rate=");
  Serial.print(flowRate); //Print milli liters per minute on serial monitor
  Serial.println("mL/minute");
}

void CountPulses()
{
  pulsecount++; //Increment the variable on every pulse
  //On every pulse toggle the state of led.
  if (i % 2 == 0)
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);
  i++;
}
```

7.4.3 Partie III : Servo

Le servo émet des valeurs digitales. Il est branché dans la pin 4. Le code ci-dessous est un code qui teste la fonctionnalité du servo. Le code dit à chaque 5 secondes de déplacer le servo

de 65 (interrupteur ouvert) et 160 (interrupteur fermé). Vous trouverez un lien vers le code de test à l'*Appendice II*.

```
// Include the Servo library : https://www.arduino.cc/reference/en/libraries/servo/
#include <Servo.h>

Servo servo;

void setup()
{
  servo.attach(4);
  servo.write(65); //off
  delay(2000);
}

void loop()
{
  servo.write(160); //on
  delay(5000);
  servo.write(65); //off
  delay(5000);
}
```

7.4.4 Partie IV: Code Final

Ci-dessous vous trouverez le code final. Il est composé des trois codes situés au-dessus. Le code dit que si l'humidité du sol est plus grande ou égale à la *limit*, le servo ouvrira l'interrupteur. Lorsque le capteur de débit est plus grand ou égale à *value*, le servo fermera. Les variable *limit* et *value* peut être modifiée selon les besoins de la plante.

```
#include <Servo.h>

//Soil Moisture Sensor
int sensorPin = A0;
int moistureValue;
int limit = 500; //Test value. This is the value to change to set the min moisture level of the plants

//Servo
Servo servo; //Pin 4 digital

// Flow Sensor
int Pulses = 3; //Arduino digital pin 3 is declared as Pulses
int led = 13; //Led pin
int value = 500; //Test value. This is the value to change to set the max amount of water the plants need
volatile int pulsecount; //Variable to count number of pulses
volatile int i = 0; //Variable for led status

void setup()
{
  //Soil Moisture Sensor N/A

  //Servo
  servo.attach(4);
  servo.write(160);
  delay(1000);

  // Flow Sensor
  pinMode(Pulses, INPUT_PULLUP); //Interrupt pin 3 declared as input and pull up resistor is enabled
  pinMode(led, OUTPUT); //Led as output
  //Interrupt is attached to pin 3
  //ISR(interrupt service routine) of interrupt is CountPulses
  //Interrupt is activated on HIGH to LOW transition
```

```

attachInterrupt(digitalPinToInterrupt(Pulses), CountPulses, FALLING);

//Serial monitor
Serial.begin(9600);
}

void loop()
{
  // Flow Sensor Shit
  pulsecount = 0; //Start counting from 0 again
  i = 0; //Reset led toggling variable
  interrupts(); //Enables interrupt on arduino pin 3
  delay (1000); //Wait 1 second
  noInterrupts(); //Disable the interrupt

  //One second is over now and we have the number of pulses in variable
  //'pulsecount'

  //Calculating the water flow rate in Milli Liters per minute
  double flowRate;
  flowRate = (pulsecount * 2.22 * 60);

  //THE CODE
  if (moistureValue >= limit) //If value of soil lower then limit
  {
    servo.write(65); //Switch Servo ON
    delay(1000);

    if (flowRate >= value) //The amount of water we need to feed the plants
    {
      servo.write(160);
      delav(1000):
    }
  }
  else
  {
    servo.write(65);
    delay(1000);
  }
}
else // If value of soil higher then limit
{
  servo.write(160); //Switch Solenoid OFF
  delay(1000);
}

//Serial Monitor
//Soil Moisture Sensor
moistureValue = analogRead(sensorPin);
Serial.println("Analog Value : ");
Serial.println(moistureValue);
// Flow Sensor
Serial.print("Flow rate =");
Serial.print(flowRate); //Print milli liters per minute on serial monitor
Serial.println("mL/minute");
}

//For the Flow Sensor
void CountPulses()
{
  pulsecount++; //Increment the variable on every pulse
  //On every pulse toggle the state of led.
  if (i % 2 == 0)
    digitalWrite(led, HIGH);
  else

    digitalWrite(led, LOW);
  i++;
}

```

7.5 Tour d'eau

Nous avons construit la tour d'eau en bois. Pour supporter la gourde d'eau, nous avons imprimé des supports en plastique flexible. Ceci aiderait avec le débit d'eau. Voir *Figures 22 et 26*.

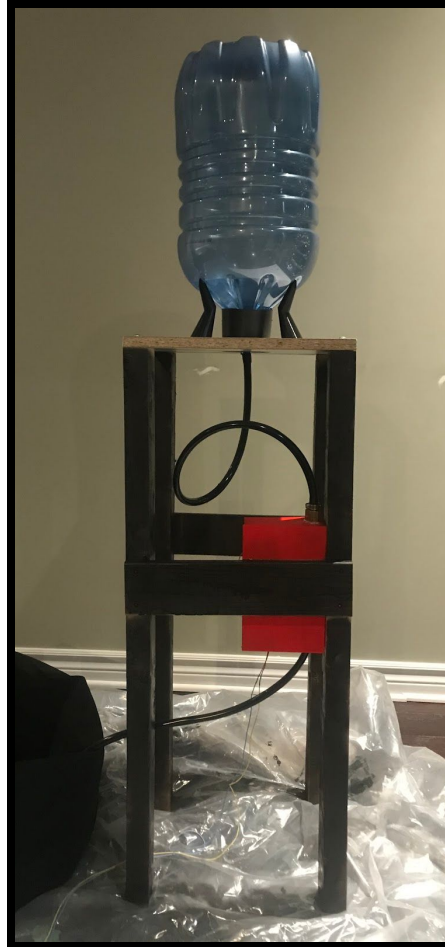


Figure 26 : Tour d'eau

7.6 Planteur

Le planteur est produit par CHEZAA. Ces dimensions sont de 80x30cm et sont faites de tissu feutré. Il est de forme circulaire. Voir *Figure 27*.

Le planteur choisi est donc celui-ci : <https://www.amazon.ca/Chezaa-Planter>



Figure 27 : Vue du dessus du planteur

8 Conclusions et recommandations

Tout au long de ce projet, nous avons appris à utiliser le processus de conception afin de faire preuve d'empathie avec le client. Nous avons également appris ce qu'il faut pour travailler en équipe et résoudre les conflits. De plus, nous avons aussi appris à utiliser diverses machines : imprimante 3D, soudure, et outils de construction. Nous avons également appris à travailler avec des microcontrôleurs, tel que Arduino.

Nous avons remarqué que plusieurs produits sur le marché n'avaient pas un système de surveillance du sol. Il était important pour nous que notre jardinière ait ce système. Avec l'aide d'un capteur d'humidité du sol, nous avons pu y parvenir. Notre système de surveillance du sol et le système d'arrosage travaillent ensemble pour rendre notre jardinière automatique.

Si nous devions refaire ou améliorer ce projet, il y aurait quelques changements à apporter.

Tout d'abord, on refera la jardinière. Nous aimerions construire une jardinière en bois ou en plastique. Ceci rendrait le planteur plus moderne.

De plus, on modifierait la tour. Nous aimerions raccourcir la longueur de la tour. Cela aiderait à diminuer nos dimensions.

Les prochaines étapes de ce projet consistent à choisir les plantes à planter dans la jardinière et à modifier le code afin qu'il reçoive la bonne quantité d'eau. Bref, nous avons créé Green Tide, la jardinière automatique, pour apporter plus de verdure au campus de l'Université.

9 Bibliographie

“Biophilia.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/biophilia>. Consulté le 30 novembre 2020.

“Hall Effect Sensor”. 2020. <https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html>. Consulté le 4 décembre 2020.

“Self-sufficient.” Merriam-Webster.com Dictionary, Merriam-Webster, <https://www.merriam-webster.com/dictionary/self-sufficient>. Consulté le 30 novembre 2020.

“Social Sciences Building.” Faculty of Social Sciences, 2012, <https://www.socialsciences.uottawa.ca/about-faculty/fss-building>. Consulté le 30 novembre 2020.

“Soil Moisture Sensor Module - Protosupplies”. 2020. ProtoSupplies. <https://protosupplies.com/product/soil-moisture-sensor-module/#:~:text=The%20Soil%20Moisture%20Sensor%20Module%20determines%20the%20amount,of%20some%20type%20when%20a%20plant%20needs%20watering>. Consulté le 4 décembre 2020.

“Water Flow Sensor - Working Principle , Applications & Examples.”. 2020. ElProCus - Electronic Projects for Engineering Students. <https://www.elprocus.com/a-memoir-on-water-flow-sensor/>. Consulté le 4 décembre 2020.

“What Is A Servo Motor And How It Works? | Realpars”, 2018. Gastreich, W., PLC Programming Courses for Beginners | RealPars. <https://realpars.com/servo-motor/>. Consulté le 4 décembre 2020.

APPENDICES

APPENDICE I: Manuel de l'utilisateur

Fonctions et capacités du produit

Table 9 : Fonctions des composantes électriques

Composantes électriques	Tension de fonctionnement	Pression de l'eau
Capteur d'humidité du sol	3.3V-5V	N/A
Capteur de débit	3.5V-12V	1.75MPa
Servo	4.8V-6.8V	N/A

Table 10 : Dimensions de la jardinière

Hauteur	54po
Largeur	24po
Longueur	48po
Volume du réservoir	15L
Longueur du tube	96po
Diamètre du tube	½po

Remplir le réservoir

1. Appuyer sur l'interrupteur pour fermer la jardinière
2. Retirer le réservoir et la pièce de connection
3. Retirer le réservoir de la pièce de connection
4. Remplir le réservoir
5. Remettre la pièce de connection sur le réservoir
6. Retourner rapidement le réservoir et replace le réservoir sur la tour

Utilisation du planteur

1. Mettre le tube au fond du planteur

2. Remplir le planteur de terre
3. Planter les plantes
4. Mettre le capteur d'humidité du sol dans le sol près du milieu du planteur

Utilisation de l'Arduino

1. Changer *limit* (valeur minimum de l'humidité du sol) et *value* (le montant d'eau qui doit passer à travers le capteur de débit, en ml/min) pour les valeurs appropriées pour les plantes
2. Téléverser le code à l'Arduino
3. Vérifier que la batterie de 9V est branchée
4. Appuyer sur l'interrupteur pour allumer la jardinière

Guide de sécurité

Ce jardin utilise de l'électricité. Attention à ne pas arroser les composants électroniques et ne mouillez pas l'Arduino, à risque de ruiner le système.

Dépannage

Si le code ne se télécharge pas sur l'Arduino, vérifiez que la carte indique *Arduino uno* et que le port est correct.

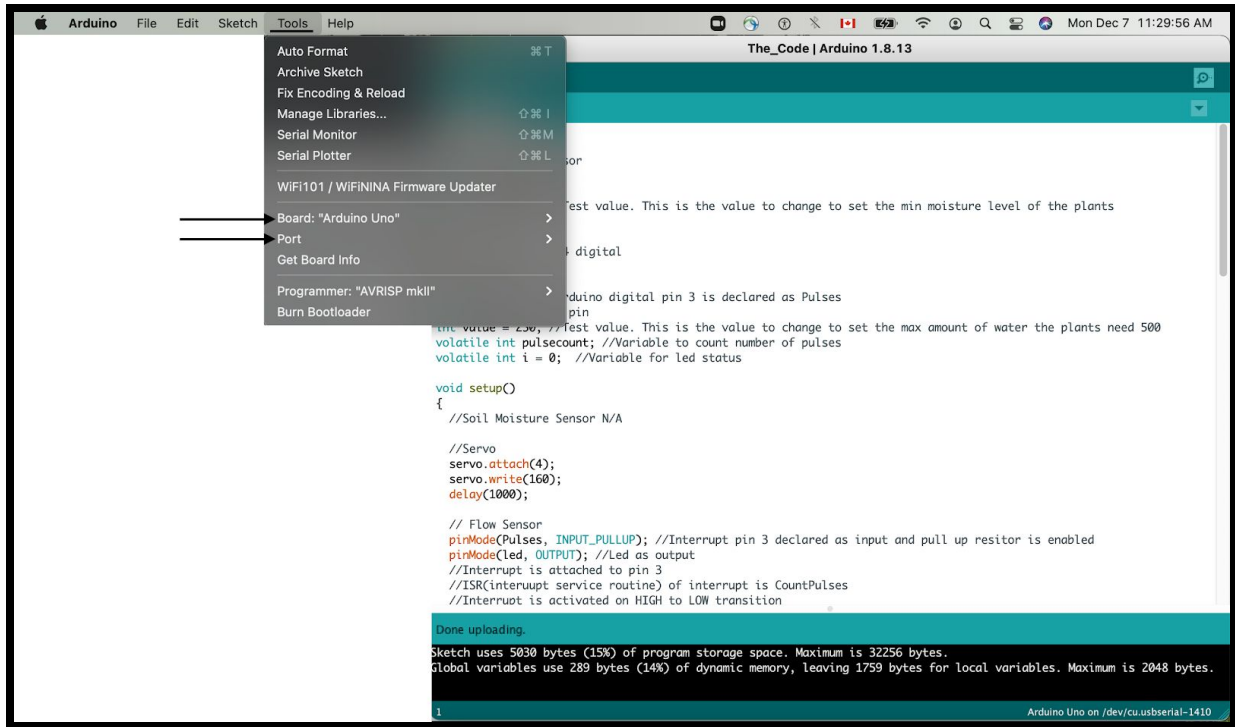


Figure 28 : Dépannage d'Arduino

Questions ou commentaires?

Si vous avez des questions ou des commentaires, écrivez à mbrow210@uottawa.ca.

APPENDICE II: Fichiers de conception

Pour accéder aux fichiers de conception, veuillez suivre le lien suivant :
<https://makerepo.com/BroMeg/green-tide-la-jardiniere-automatique-gng1503-groupe-a6>.

Vous trouverez les fichiers suivants dans la section *PROJECT FILES* :

- Le manuel d'utilisateur (.pdf) ;
- Prototype I (.pdf) ;
- Prototype II (.pdf) ;
- Prototype III (.pdf) ;
- Matériel pour la Journée de Conception (.pdf) ;
- Le code de test pour le servo (.ino) ;
- Le code de test pour le capteur d'humidité du sol (.ino) ;
- Le code de test pour le capteur de débit (.ino) ;
- Le code final de la jardinière (.ino) ;
- La boîte électronique (.stl) ;
- Les supports pour la gourde (.stl) ;