GNG1503

Manuel d'Utilisateur du Projet de Conception

Une jardinière automatique

Soumis par:

[EXAF-7 (A7)]

[Camille Kulinda-Olibrice, 300117032]

[Asish Aumdra, 300208586]

[Alexandre Beaudette, 300177531]

[Juan Roque Sanchez, 300183904]

[Anthony Abou Fayssal, 300167325]

Le 9 décembre 2020

Université d'Ottawa

Résumé

Pour le cours GNG 1503 nous avons la tâche de créer un système automatique pour le Bureau du développement à l'Université d'Ottawa. Son gestionnaire du développement durable, Monsieur Jonathan Rausseo, dans le cadre de son travail, veut rendre l'Université d'Ottawa plus 'vert' en installant plusieurs planteurs autour du campus. Bien que ceux-ci aient un but environnementaliste, l'administration ne peut pas dépenser trop d'argent pour leur entretien. Par conséquent, Monsieur Jonathan Rausseo a besoin d'une jardinière automatique qui répond à tous les besoins d'entretien et d'ameublement.

La jardinière doit être écologique, facile à transporter, diversifiable, et facile à contrôler. Le coût total du projet ne pouvait pas dépasser 100$. Pour arriver à concevoir tel produit, nous avons suivi le processus de la pensée conceptuelle, qui est comporté par 5 étapes et qui met l'accent sur l'empathie et le prototypage rapide et itératif.

Table des matières

Résumé ii

Table des matières iii

Liste de figures iv

Liste de tables v

1 Introduction 1

1.1 Identification des besoins et contraintes 1

1.1.1 Enoncé du problème 1

1.1.2 Besoins du client 1,2

1.1.3 Étalonnage 3

1.1.4 Spécifications et contraintes 5

1.2 Conception de sous-systèmes 6

1.2.1 Système de réservoir 6

2 Comment le prototype est construit 16

2.1 Prototype I 16

2.2 Prototype II 17, 18, 19,20,21

2.3 Prototype III 21,22

2.4 Solution finale 22,23

2.4.1 LDM (Liste des Matériaux) 23,24

2.4.2 Liste d'équipements 24

3 Comment utiliser le prototype 24,25

4 Comment maintenir le prototype 25,26

5 Conclusions et recommandations pour les travaux futurs 26

6 Bibliographie 27

APPENDICES 29

APPENDICE I: Fichiers de conception 29

Liste de figures

Figure 1 - Conception de système de réservoir
Figure 2 - Conception de système de réservoir
Figure 3 - Conception de système de réservoir
Figure 4 - Conception de système de réservoir
Figure 5 - Conception de système de réservoir
Figure 6 - Conception du système d'arrosage
Figure 7 - Conception du système d'arrosage
Figure 8 - Conception du système d'arrosage
Figure 9 - Conception du système d'arrosage
Figure 10 - Conception du système d'arrosage
Figure 11 - Conception du système d'humidité
Figure 12 - Conception du système d'humidité
Figure 13 - Conception du système d'humidité
Figure 14 - Conception du système d'humidité
Figure 15 - Concept final
Figure 16 - Diagramme de Gantt Trello
Figure 17 - Diagramme de Gantt Trello
Figure 18 - Prototype 1 OnShape
Figure 19 - Prototype 1 physique
Figure 20 - Circuit électrique TinkerCAD
Figure 21 - Prototype 2 physique
Figure 22 - Code Arduino
Figure 23 - Code Arduino
Figure 24 - Code Arduino
Figure 25 - Code Arduino
Figure 26 - Prototype 3 physique
Figure 27 - Prototype 3 fonctionnel
Figure 28 - Produit final sans étui
Figure 29 - Produit final

Liste de tables

Table 1 Besoin du client
Table 2 étalonnage de trois jardinières automatiques existantes
Table 3 Quantification de l'étalonnage des trois jardinières automatiques existantes
Table 4 Spécifications de Conception Technique

1 Introduction

Depuis le début du temps, la race humaine a comme but principal de survivre. Une des choses les plus importantes est l'alimentation. Il a toujours été important pour l'humain d'avoir quelque chose à manger pour donner au corps l'énergie nécessaire pour compléter des tâches. L'une des sources de nourriture les plus importantes est la végétation. Celle-ci a pu permettre à l'être humain de manger sans chasser, de créer des produits ou épices… De nos jours, la végétation n'est pas la seule utilité des plantes. C'est maintenant très commun d'avoir une plante simplement pour son aspect visuel ou pour remplir une espace vide, ce qu'on appelle "ajouter du vert à la salle". L'un des plus grands problèmes avec ceci est l'entretien de ces plantes. Il est très commun d'oublier d'arroser une ou plusieurs plantes et quand ceci se passe, ça peut arriver que la plante meurt. Pour empêcher ceci, notre équipe a conçu un modèle de planteur complètement automatique.

L'université d'Ottawa avait ce problème, de ne pas avoir assez de main d'œuvre pour être capable d'arroser toutes les plantes sur le campus. Alors, ils sont venu voir notre classe de génie à la recherche d'une solution efficace, avec quelques contraintes ou demandes mis en place par le client, la conception a pu commencer. Après de nombreux concepts, essais et prototypes, nous, l'équipe EXAF-7 sommes finalement arrivées à une bonne solution pour ce problème. Dans les pages suivantes, le processus de conception est expliqué et démontré par les nombreux efforts mis par notre équipe.

1.1 Identification des besoins et contraintes du produit

1.1.1 Énoncé du problème

L'objectif de ce projet de conception est de concevoir une jardinière avec un système d'arrosage automatique qui est efficace, sécuritaire, pas trop coûteux et moderne. Nous ferons cela tout en pensant au côté environnemental et écologique.

1.1.2 Besoins du client

Les besoins du client ont changé au cours du projet. Nous présentons donc seulement ses derniers besoins. En fonction de ces besoins, nous avons élaboré des critères de conception.

Table 1 Besoin du client

Numéro	Besoins	Critères de conception
1	Le planteur a un système d'arrosage automatique.	Duration de la batterie (heures) Coût ($) Volume d'eau (Litres)

2	Le planteur a un système de détection d'humidité.	Détection/contrôle d'humidité par un capteur Duration de la batterie (heures) Cout ($)
3	Le coût des planteurs et du système de la jardinière n'est pas élevé	Matériaux Coût ($)
4	Le planteur peut être résistant au feu	Matériaux
5	Le système informe les personnes d'entretiens sur le besoin d'eau ainsi que la santé d'une telle plante dans un tel édifice.	Efficacité/simple à interpréter Coût ($) Duration de la batterie (heures)
6	Le planteur est environnemental et utilise une quantité d'eau minime.	Volume réservoir (Litres) Coût ($)
7	Le planteur n'est pas spécifique à une plante.	Grandeur (m) Coût ($)
8	Le planteur est transportable et difficile à voler	Poids (Kg) Matériaux Fixation au sol
9	Le planteur a un style moderne	Matériaux Coût ($) Forme Estéthique
10	Le planteur est facile à nettoyer pour éviter la moisissures et les insectes	Forme Coût ($) Matériaux
11	Le jardinier utilise des plantes faciles à entretenir comme les plantes serpents ou le pothos.	Coût ($) Beauté/visuel
12	Les planteurs se trouvent partout sur le campus commençant par les endroits publics notamment les édifices Desmarais, le Carrefour d'apprentissage et SITE.	Taille (m) Coût ($) Estéthique

En plus de quelques autres besoins qui ne sont pas nécessairement prescrits par le client, ces besoins peuvent être regroupés sous trois catégories.

- Exigences fonctionnelles
    - Système d'arrosage
    - Contrôle d'humidité
- Exigences non fonctionnelles
    - Sécurité
    - Écologique
    - Matériaux
- Contraintes
    - Poids
    - Duration de la batterie
    - Coût
    - Taille

1.1.3 Étalonnage

Afin de se faire une idée de ce qu'il existe déjà sur le marché, nous avons procédé à un étalonnage de différents produits afin d'identifier les forces et les faiblesses. Étant donné que le client a modifié ses exigences, les besoins qui font l'objet d'étalonnage au début du projet diffère légèrement des besoins utilisés pour le produit final.

Table 2 étalonnage de trois jardinières automatiques existantes

Spécifications	**CLICK AND GROW**	**CONTINENTAL AWS-10**	**ESOTEC KIT COMPLET D'ARROSAGE**
Image			
Compagnie	Click & Grow	Continental	Esotec
Poids	1.2kg	4.54 g	1.3 kg

Matériaux	Plastique ABS	Plastique	Plastique ABS
Taille	Hauteur : 21cm Largeur: 30cm Profondeurs: 12cm	34.93 x 8.26 x 20.96 cm	22cm x 19cm x 16cm
Coût	99.99$	164.95$	114.15$
Système d'arrosage	Surveille le niveau d'eau	Par chronomètre	Deux fois par jours, 15 gouttes par fois
Système d'humidité	Surveille le niveau de nutriments	AUCUNE	AUCUNE
Sécurité	AUCUNE	AUCUNE	Fixé sur le pot de plante
Écologique	Plastique recyclable	Plastique recyclable	Recyclable, énergie solaire
Duration de la batterie	Électrique, 100-240 V c.a., environ 50/80 Hz	4 x AA batteries	Batteries NiMH, duration dépend d'utilisateur

Table 3 Quantification de l'étalonnage des trois jardinières automatiques existantes

Spécifications:	**Importance (Poids)**	**CLICK AND GROW**	**CONTINENTAL AWS-10**	**ESOTEC KIT COMPLET D'ARROSAGE**
Compagnie:		Click and Grow	Continental	Esotec
Poids	2	(2)2 = 4	(2)3 = 6	(2)1 = 2
Matériaux	4	(4)2 = 8	(4)1 = 4	(4)2 = 8
Taille	2	(2)1 = 2	(2)3 = 6	(2)2 = 4
Coût	5	(5)3 = 15	(5)1 = 5	(5)2 = 10
Système d'arrosage	5	(5)3 = 15	(5)2 = 10	(5)1 = 5
Système d'humidité	4	(5)2 = 10	(5)1 = 5	(5)1 = 5

Sécurité	3	(5)1 = 5	(5)1 = 5	(5)2 = 10
Écologique	4	(5)2 = 10	(5)2 = 10	(5)3 = 15
Duration de la batterie	5	(5)3 = 15	(5)2 = 10	(5)1 = 5
Total		84	61	69

Une fois l'étalonnage fait, nous avons déterminé que le système de jardinières 1 'CLICK AND GROW'. Nous avons étudié le système et copié quelques idées, mais notre jardinière finale était quand même très différente.

1.1.4 Spécifications/contraintes

Afin de nous guider, nous avons élaboré des spécifications cibles afin de dicter la performance attendue de chaque composante.

Table 4 Spécifications de Conception Technique

	Critères de conception	**Relation (=, >, <)**	**Valeur**	**Unités**	**Méthode de vérification**
	Exigences fonctionnelles				
1	Système d'arrosage	=	oui	s.o	Essai
2	Contrôle d'humidité	=	oui	s.o	Essai
	Contraintes				
3	Poids	<	8	kg	Analyse
4	Duration de la batterie	>	14	jours	Analyse, essai et estimation
5	Coût	<	100	$	Estimation, vérification finale
6	Taille	<	4	L	Estimation, essai
	Exigences non-fonctionnelles				

7	Sécurité	=	oui	s.o	Analyse, essai
8	Écologique	=	oui	s.o	Estimation, analyse
9	Matériaux				

1.2 Conceptions des sous-systèmes

1.2.1 Système de réservoir

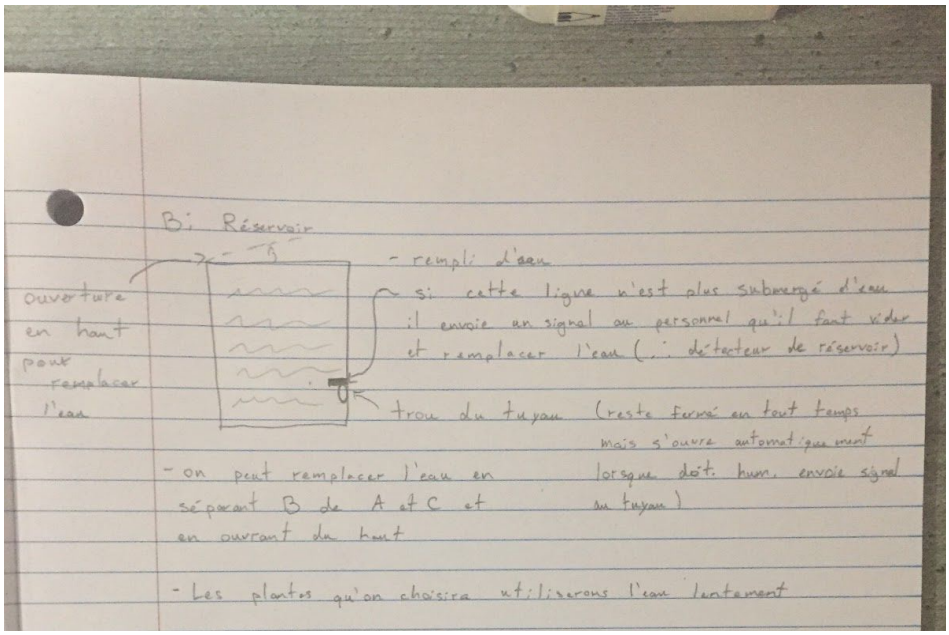

Figure 1

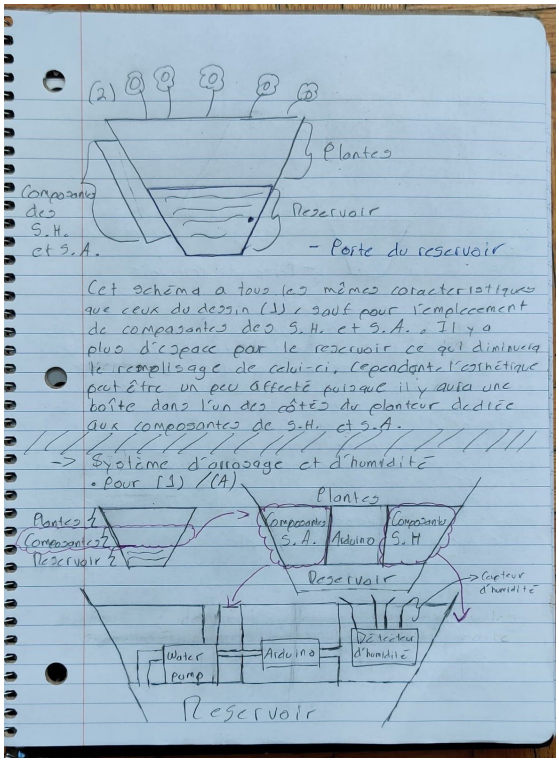

Figure 2

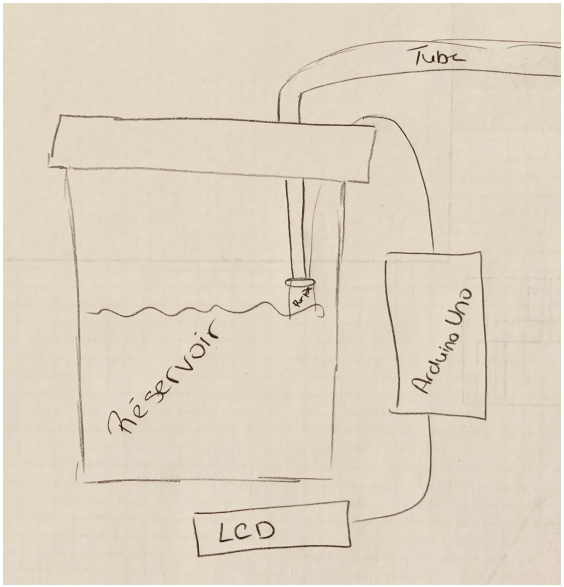

Figure 3

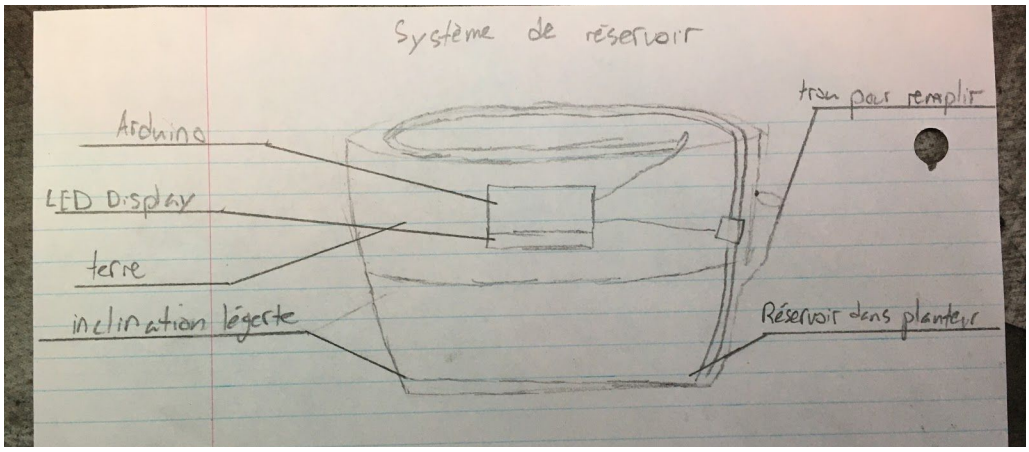

Figure 4

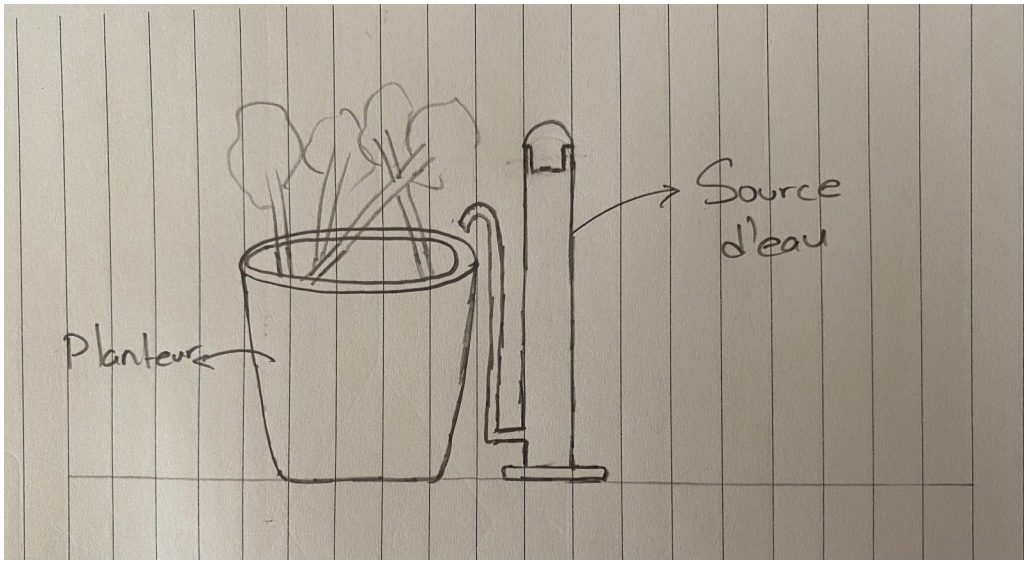

Figure 5

Voici tous les concepts de notre équipe. Après le partage des concepts et la rencontre d'équipe, nous avons pu constater quelque chose, des similarités d'un autre sens. Premièrement, une idée commune était de mettre le réservoir dans le même pot que la plante elle-même. Cette idée fonctionne car la plante ne prend pas beaucoup de place et notre idée était d'avoir un planteur d'environ deux pieds. Donc, avec cette grandeur en tête, nous avons décidé que ce plan était plus esthétique et efficace. De plus, une autre idée commune était un détection de niveau d'eau dans le réservoir. Ceci permet à l'utilisateur de savoir quand vient le temps de remplir le réservoir. Finalement, nous avons décidé que pour remplir le réservoir, il y aurait un petit trou sur le côté pour être capable de facilement remplir le réservoir.

1.2.2 Système d'arrosage

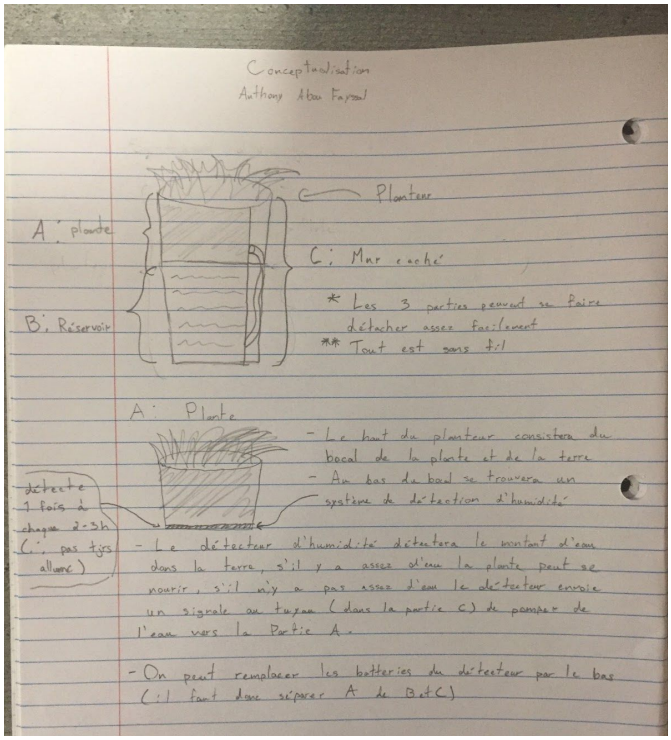

Figure 6

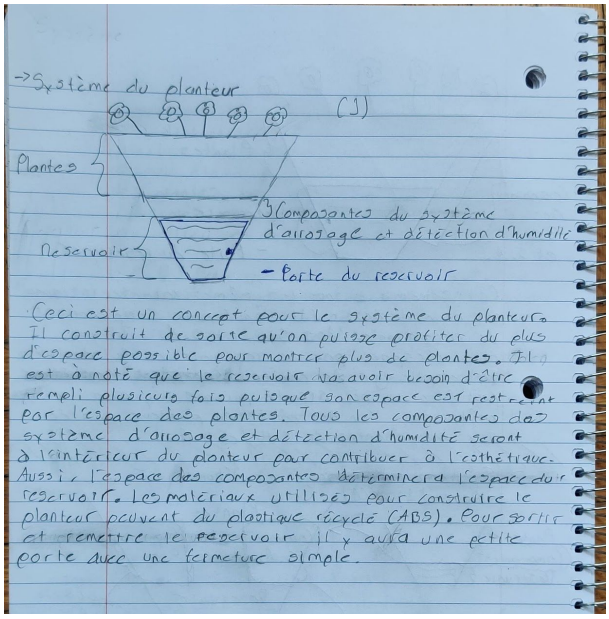

Figure 7

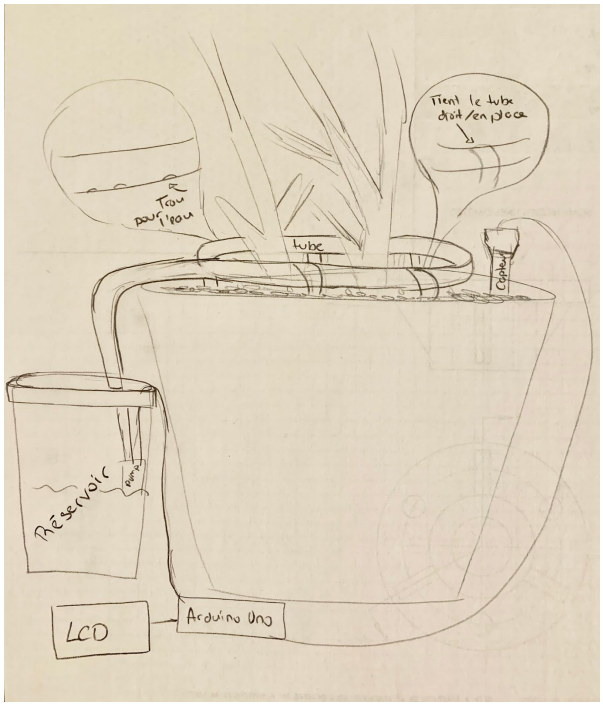

Figure 8

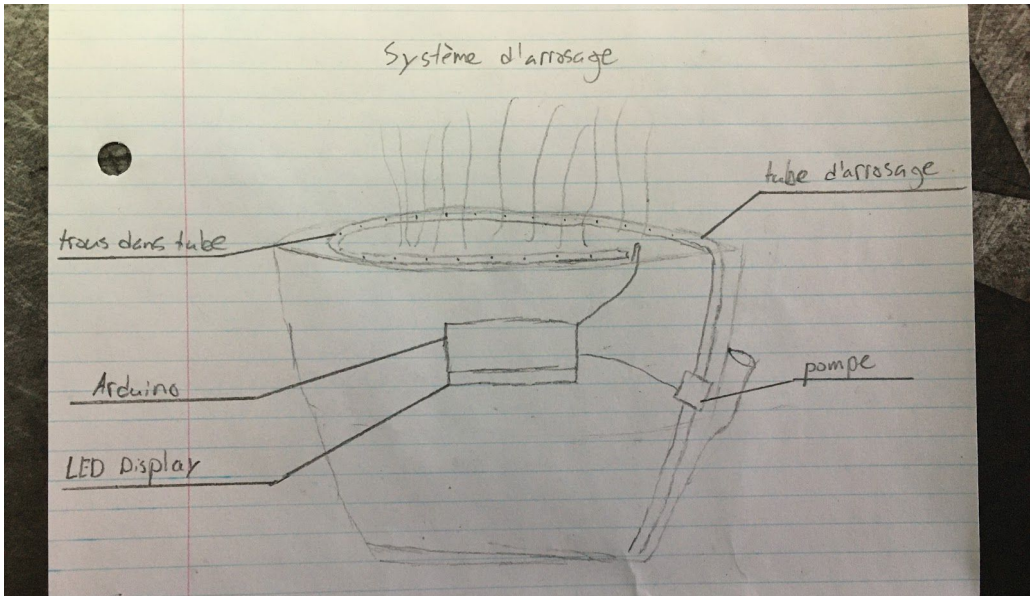

Figure 9

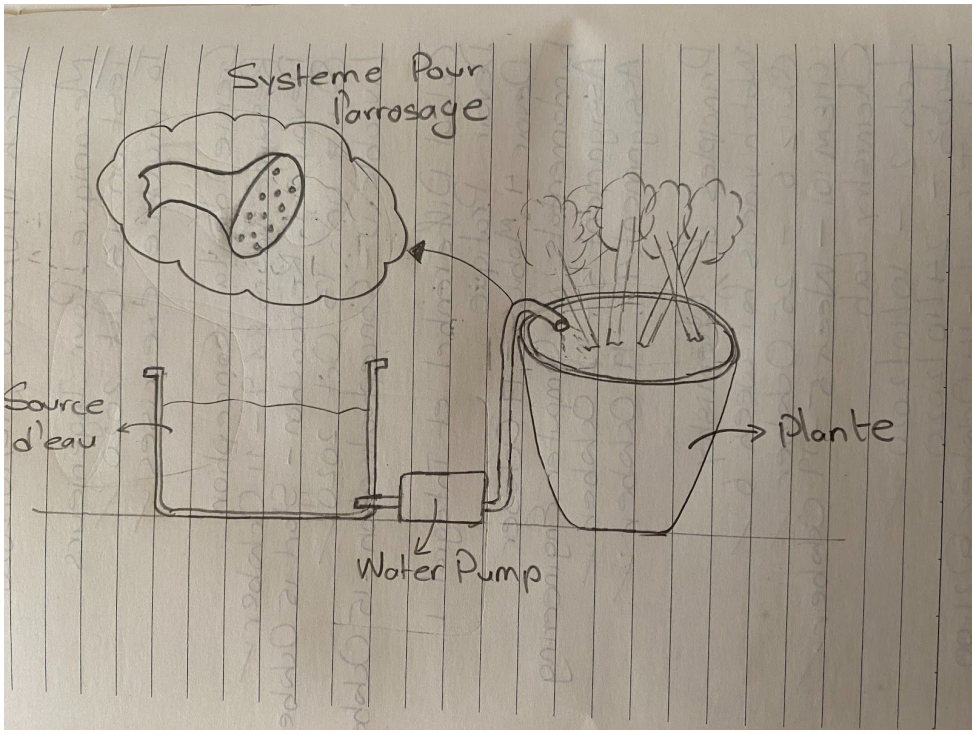

Figure 10

En groupe, nous étions tous en accord que le système d'arrosage était très important. Donc, avec ceci en tête, nous avons passé beaucoup de temps à essayer de trouver la meilleure solution. Premièrement, tous les concepts utilisent une pompe pour amener l'eau du réservoir à la plante. Notre équipe a choisi d'élaborer sur cette idée et d'utiliser un tube pour amener cette eau du réservoir, qui se retrouve sous la plante, jusqu'à la plante. Un autre aspect à réaliser est que nous avons décidé de faire passer le tube tout autour de la plante pour être capable de maximiser son arrosage. Finalement, nous avons décidé de faire un compartiment sur le côté du planteur pour contenir le arduino, les fils et le reste des composantes électriques. Finalement, notre concept final pour ce sous-système était une pompe dans le réservoir qui pompe de l'eau par un petit tube, jusqu'à la plante, qui fait le tour de celle-ci pour mieux l'arroser.

1.2.3 Système de détection d'humidité

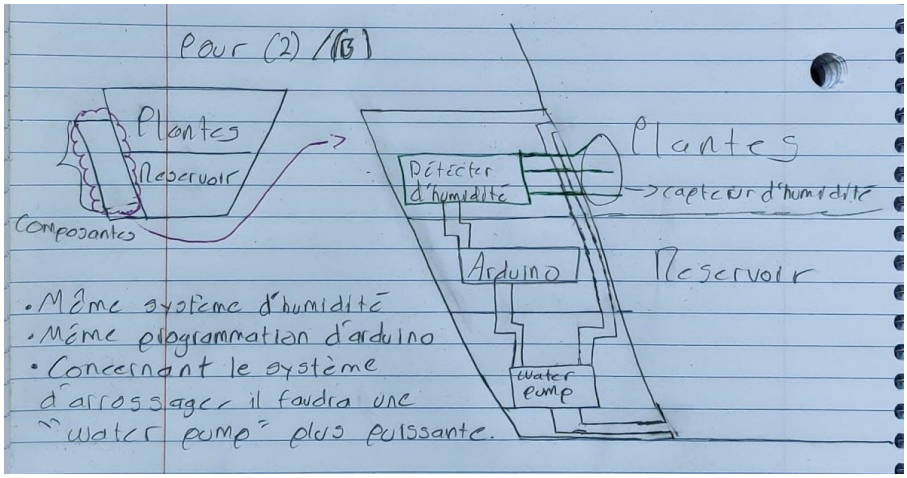

Figure 11

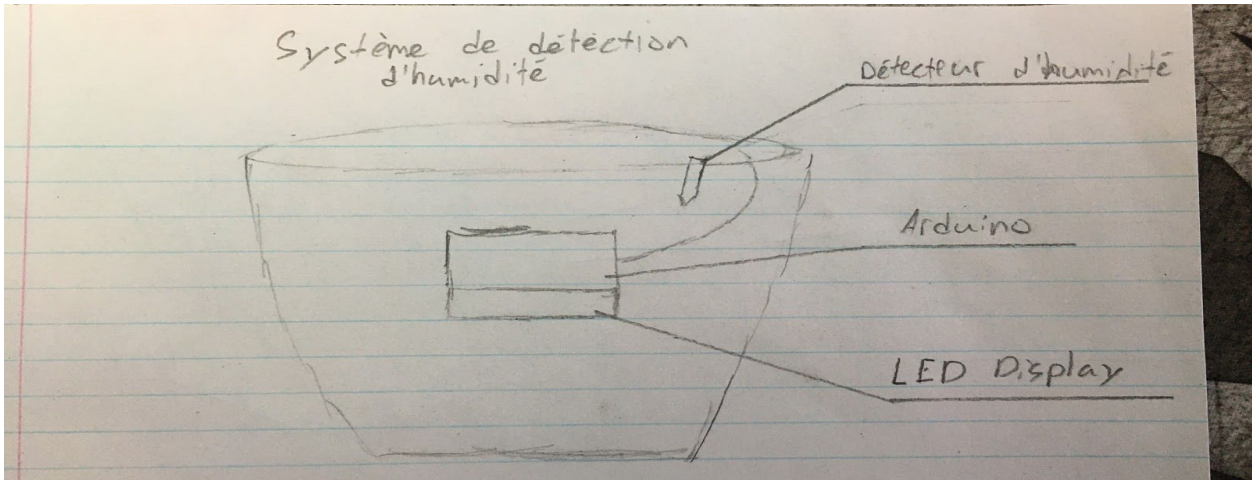

Figure 12

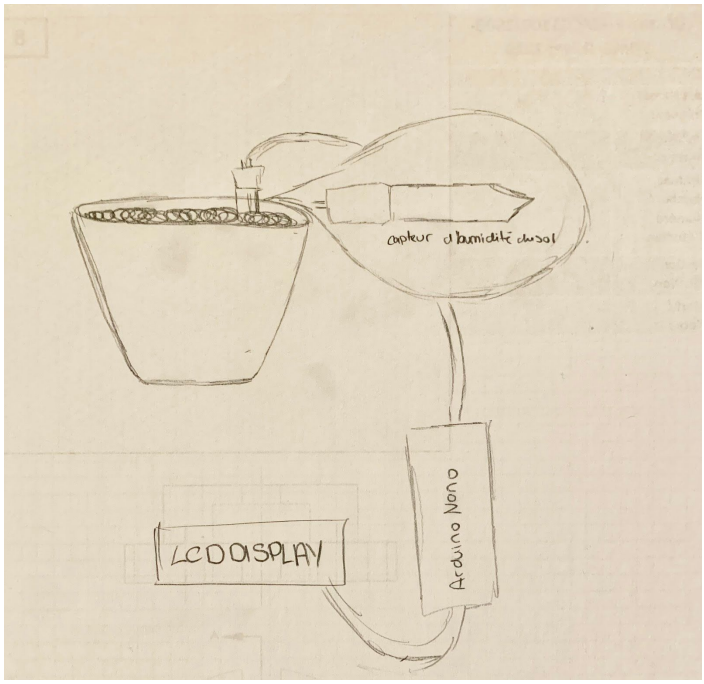

Figure 13

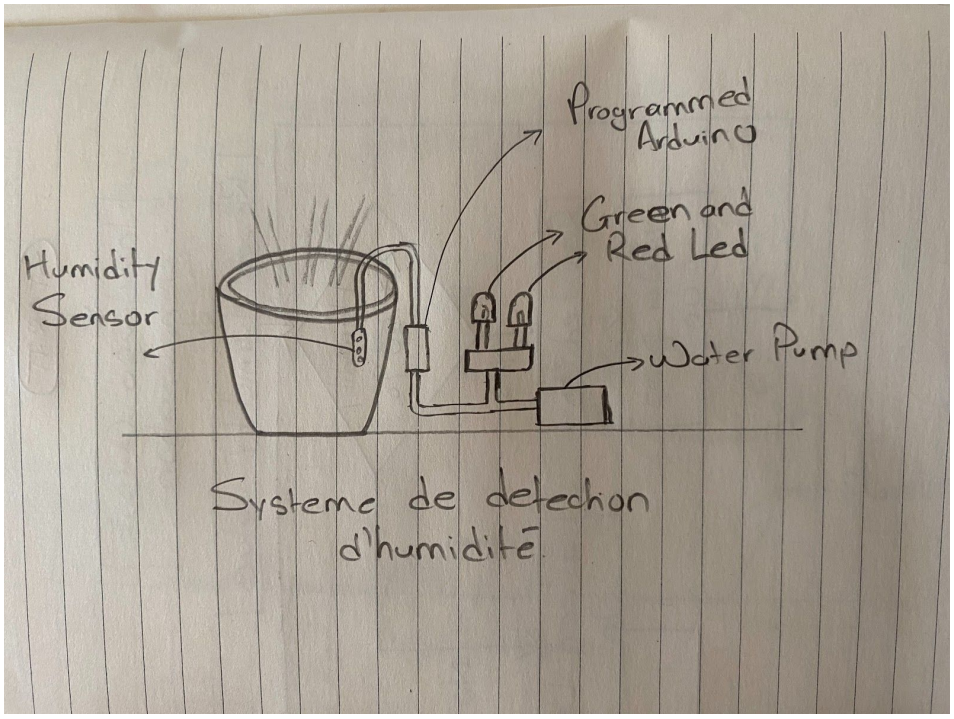

Figure 14

Pour le sous système de détection d'humidité, l'idée principale était simplement d'avoir un petit détecteur d'humidité dans la terre de la plante et que ce détecteur soit connecté au arduino dans le

petit compartiment sur le côté du pot. Après avoir discuté de tous nos différentes idées de ce concept, nous sommes venu à une idée finale, qui est simplement de placer le détecteur dans la terre et que quand la terre est trop sèche, le détecteur envoie un signale au arduino, qui envoie un signal à la pompe pour arroser la plante.

1.3 Concept final

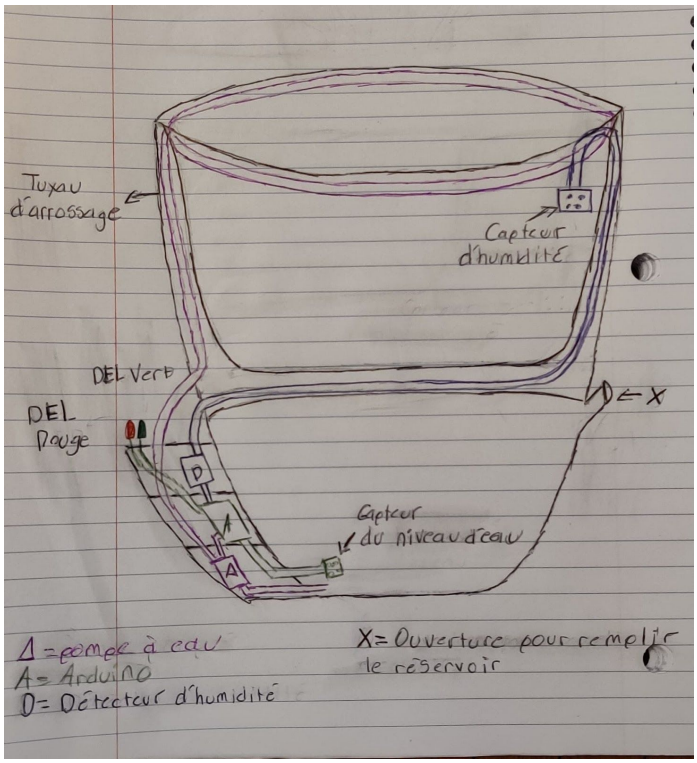

Figure 15

Après les nombreuses discussions et tous les concepts de l'équipe, nous sommes arrivés à une solution qui faisait plaisir à tous. Pour débuter, le système de réservoir choisi est le réservoir dans le bas du pot de plante, pour un réservoir compact et efficace. De plus, le trou "x" est le petit trou pour remplir le réservoir. Pour le système d'arrosage, nous sommes arrivées à l'idée de la pompe dans le réservoir, qui pompe l'eau par un tube jusqu'à la plante, où elle sera arrosée de tous les angles. De plus, ce système fonctionne avec le système de détection d'humidité car quand les détecteurs envoient un signal à l'arduino pour lui dire que la planète terre est sèche, la pompe est activée et la plante est arrosée. Enfin, pour le système de détection d'humidité, nous avons simplement choisi de mettre des détecteurs d'humidité dans la terre, qui vont envoyer un signal à l'arduino quand ce sera nécessaire. Finalement, toutes les composantes électroniques (fils,

arduino) seront stockées dans le compartiment sur le côté du pot pour donner au planteur un bon aspect esthétique.

1.4 Plan du projet suivi

Avant de construire les trois prototypes du produit final, nous nous sommes rendu compte qu'il fallait faire un plan pour que nous puissions compléter le projet avant le 3 décembre 2020.Avec ce plan, nous avions une meilleure organisation et idée sur ce qui allait se passer tout au long de la conception. Par conséquent, nous avons créé une diagramme et assigné différentes tâches aux membres de l'équipe.

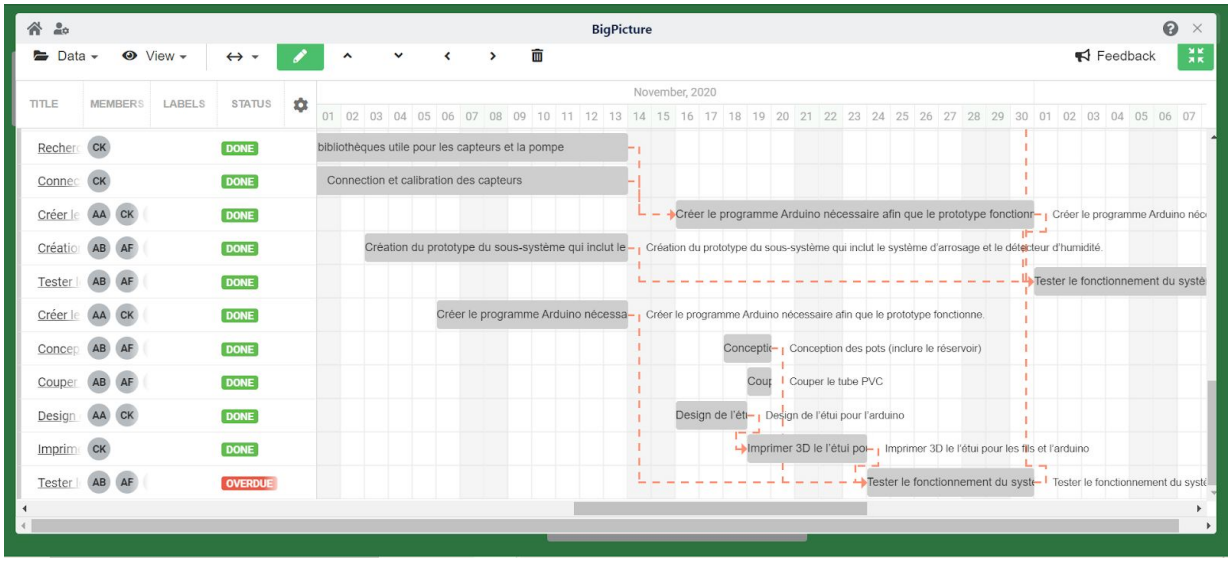

Figure 16

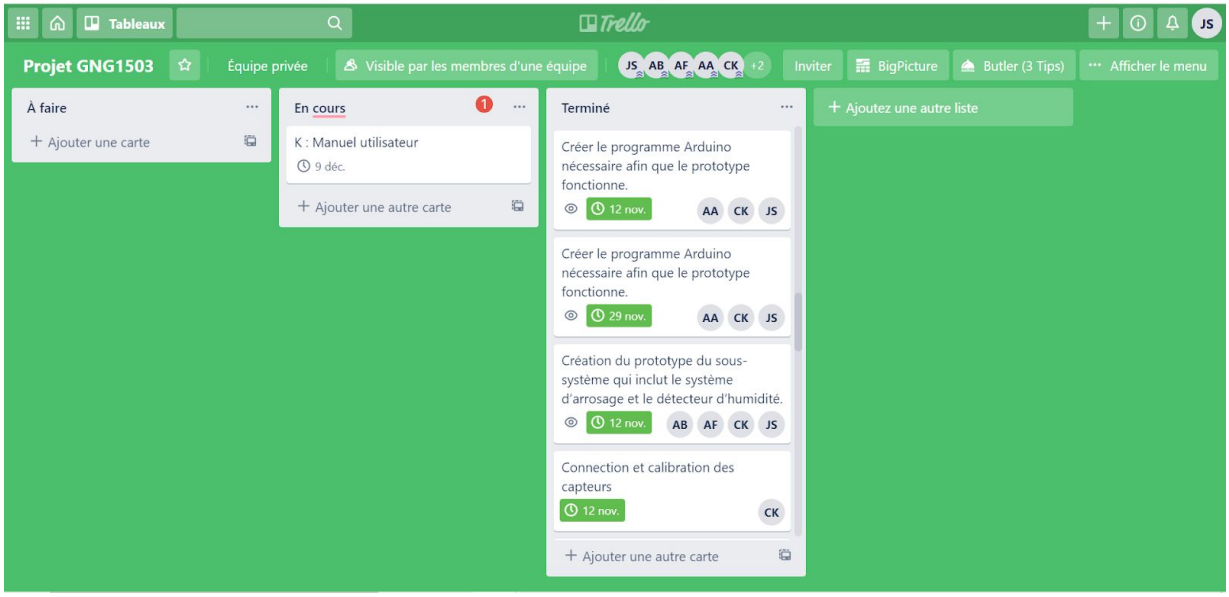

Figure 17

2 Comment le prototype est construit

Durant le processus de design, nous avons pu créer trois prototypes distincts qui nous ont permis de développer notre produit final. Notre premier prototype était une estimation de la taille et l'emplacement des sous-systèmes. Notre deuxième prototype est concentré sur la conception du code Arduino nécessaire pour l'arrosage automatique des plantes. Notre troisième prototype est concentré sur l'assemblage des trois sous-systèmes principaux dans le planteur. Finalement notre produit final est très similaire à notre troisième prototype mis à part quelques modifications à l'apparence.

2.1 Prototype I

Pour le prototype 1 nous avions conçu un prototype physique ainsi qu'un prototype sur OnShape. Le prototype OnShape fut conçu grâce à notre expérience du laboratoire 2: Conception Assisté par Ordinateur. Ce laboratoire nous avait permis de nous familiariser avec les CAO. Après ce laboratoire et une recherche rapide afin de voir comment dessiner la forme spécifique que nous voulions dessiner, nous avons pu créer notre prototype 1 virtuel, qui nous servira à visualiser la forme dont nous aurions besoin. Voici une image de ce prototype 1 virtuel:

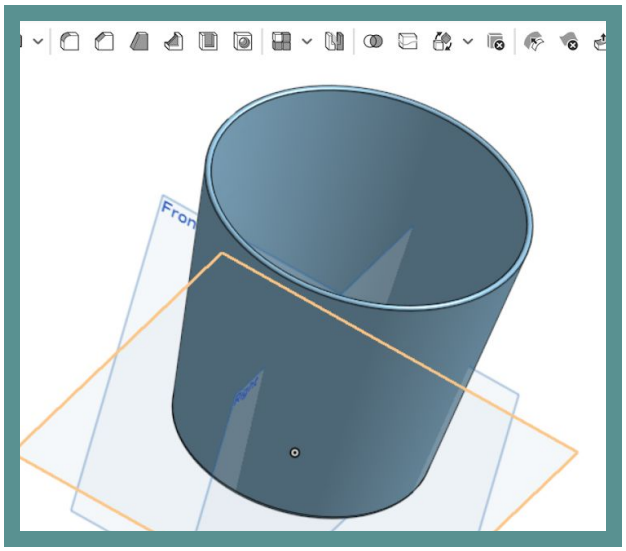

Figure 18

Bien évidemment, ce prototype servait simplement à visionner la forme que prendrait notre produit final. Pour cela nous avons créé un prototype physique qui démontreront à quoi auront l'air les sous-systèmes tous installés sur notre produit final. Pour ce prototype, il fallait s'assurer de rester fidèle à notre objectif qui était seulement de créer un prototype non-fonctionnel servant de visuel pour notre produit final. À cause de cela, les matériaux et la technologie utilisée ne dérange pas. Nous avons donc opté d'utiliser n'importe quoi qu'on pourrait retrouver dans notre

maison afin de représenter les différents aspects du produit. Voici une image de ce prototype physique:

Figure 19

Le sceau et le bol étaient interposés l'un par dessus l'autre afin de représenter les deux compartiments du seau (le bol pour la plante et le sceau pour le réservoir). Le lacet mauve sert à représenter le tuyau de la pompe qui arrosera la plante. Le fil gris représente le fil du capteur d'humidité qui sera connecté au Arduino. Le rectangle brun sert à représenter l'Arduino qui sera attaché au côté du sceau.

2.2 Prototype II

Pour le deuxième prototype, nous voulons créer les circuits ainsi que le code Arduino nécessaire pour faire arroser les plantes. Tout d'abord, à l'aide de TinkerCAD on a créé le circuit. La figure ci-dessous démontre le circuit :

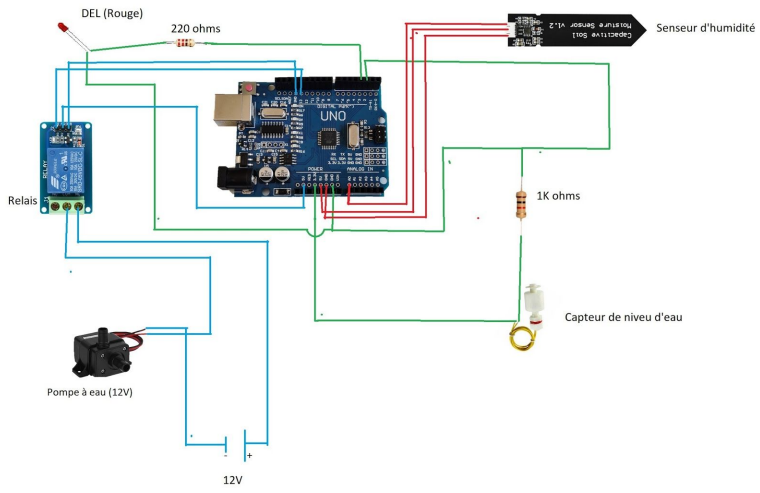

Figure 20

On a tout d'abord tester toutes les composantes séparément pour vérifier leur fonctionnement tout d'abord. Après ceci, on a fait des recherches sur chacune des composantes et leur fonctionnement. On a développé le code final en partie. Avec l'aide du circuit ci-dessus, on construit le circuit physique avec l'aide de toutes les composantes présentes dans l'image ci-dessus.

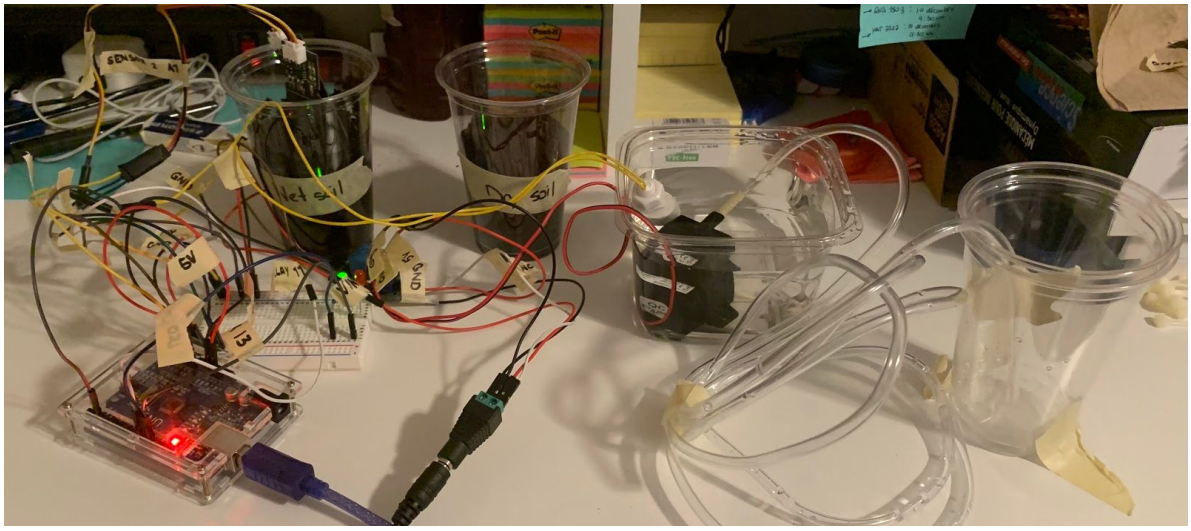

Figure 21

La figure ci-dessus représente notre deuxième prototype. Les deux capteurs d'humidité se trouvent dans le sol et sont connectés à l'Arduino, le float (capteurs de niveau d'eau) se trouve dans le contenant d'eau. La lumière verte est allumée puisque le niveau d'eau est suffisant. La pompe à eau se trouve dans l'eau et est attachée au tube qui mène à une autre contenant ou l'eau sortira. La pompe et l'Arduino sont capables d'être alimentées par le chargeur laptop. Ce dernier fournit assez de pouvoir pour le circuit en entier ainsi que limiter le pouvoir de la pompe à eau. Puisque sinon ils pompent l'eau trop fort.

Chaque composantes doit travailler ensemble donc à la fin on a combiné les codes et le fonctionnement des composantes pour créer le code finale que tu peux voir ci-dessous :

soil_sensor_water_pump_led_water_level §

```
#define Soil_Moisture_Sensor_1 A0       //Soil sensor 1 is connected to A0
#define Soil_Moisture_Sensor_2 A1       //Soil sensor 2 is connected to A1
#define Water_Level_Sensor 2            //Vertial float is connected to pin 2
int relay = 11;                         //Relay is connneced to pin 11
int soilMoisture1 = 0;                  //Variable that stores value of sensor 1
int soilMoisture2 = 0;                  //Variable that stores value of sensor 2
bool watered = false;                   //Variable that contains state of the soil
int redLED = 13;                        //Red LED is connneced to pin 13
int greenLED = 12;                      //Green LED is connneced to pin 12
int waterPumpCounter = 0;               //Counter for the # of pumps it takes to empty reservoir
int waterPumpMax = 0;                   //Stores the # of pumps it takes to empty the reservoir for future use

void setup()
{
  pinMode(redLED, OUTPUT);                          //Initialize red LED pin as output
  pinMode(greenLED, OUTPUT);                        //Initialize green LED pin as output
  pinMode(relay, OUTPUT);                           //Initialize relay pin as output
  digitalWrite(relay, LOW);                         //Set the relay to OFF
  pinMode(Soil_Moisture_Sensor_1, INPUT);           //Initialize sensor 1 as input
  pinMode(Soil_Moisture_Sensor_2, INPUT);           //Initialize sensor 2 as input
  pinMode(Water_Level_Sensor, INPUT_PULLUP);        //Initialize vertical float as input pullup (acts like a button)
  Serial.begin(9600);                               //Initialize serial monitor
}

void loop() {
  ledStatus();                                     //Runs void below names ledStatus to adjust light status
  if (digitalRead(Water_Level_Sensor) == LOW) {    //This code will run if float sensor is LOW (when reservoir is EMPTY)
```

```
    waterPumpMax = waterPumpCounter;              //Stores # of pumps required to empty reservoir for future use
    waterPumpCounter = 0;                         //Resets value of waterPumpCounter to 0 for the next watering
    ledStatus();                                  //Runs void below names ledStatus to adjust light status
  }
  ledStatus();                                    //Runs void below named ledStatus to adjust light status
  Serial.print("Sensor 1 : "); Serial.println(analogRead(Soil_Moisture_Sensor_1)); //Prints value of sensor 1 to the serial monitor
  Serial.print("Sensor 2 : "); Serial.println(analogRead(Soil_Moisture_Sensor_2)); //Prints value of sensor 2 to the serial monitor
  delay(100);                                                                      //Delay

  if (soilMoisture1 > 300 and soilMoisture2 > 300 and watered == false and digitalRead(Water_Level_Sensor) == HIGH and waterPumpCounter <= waterPumpMax) {
  //This code will run if value of sensor 1 & 2 are above 300, watered is false (plant not watered), if the water reservois is not empty and if the number of
  //pumps dose not exceed the amount used previously to prevent from running the pump without water and damaging it
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
    //pumpON();                                   //Runs void below named pumpOn to turn on the water pump
    delay(3000);                                  //Runs void below named ledStatus to adjust light status
    pumpOFF();                                    //Runs void below named pumpOff to turn off the water pump
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
  } else {                                        //This code runs if one or more than one of the conditions above are false
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
    pumpOFF();                                    //Runs void below named pumpOff to turn off the water pump
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
  }
  if (watered == true) {                          //This code runs if the plant has just been watered
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
    delay(10000);                                 //10 secondes delay (to be adjusted) as a waiting period before checking if the plant has been watered enough
    ledStatus();                                  //Runs void below named ledStatus to adjust light status
    soilMoisture1 = analogRead(Soil_Moisture_Sensor_1); //Verifies the value of sensor 1
```

```
    soilMoisture2 = analogRead(Soil_Moisture_Sensor_2); //Verifies the value of sensor 2
    delay(100);                                         //Delay
    watered = false;                                    //Set value of watered to false to allow plant to be watered again when needed
    ledStatus();                                        //Runs void below named ledStatus to adjust light status
  } else {                                              //This code runs if the plant has not been watered
    ledStatus();                                        //Runs void below named ledStatus to adjust light status
    soilMoisture1 = analogRead(Soil_Moisture_Sensor_1); //Verifies the value of sensor 1
    soilMoisture2 = analogRead(Soil_Moisture_Sensor_2); //Verifies the value of sensor 2
    delay(100);                                         //Delay
    ledStatus();                                        //Runs void below named ledStatus to adjust light status
  }
}

void ledStatus () {
  if (digitalRead(Water_Level_Sensor) == LOW) { //This code runs if the reservoir is empty
    digitalWrite(redLED, HIGH);                 //Turns red LED ON
    digitalWrite(greenLED, LOW);                //Turns green LED OFF
  } else {                                      //This code runs if the reservois is not empty
    digitalWrite(greenLED, HIGH);               //Turns greeen LED ON
    digitalWrite(redLED, LOW);                  //Turns red LED OFF
  }
}

void pumpON() {                                 //This code is in charge of turning the pump ON
  ledStatus();                                  //Runs void below named ledStatus to adjust light status
  digitalWrite(relay, HIGH);                    //Turns relay ON
  watered = true;                               //Set value of watered to true to prevent plant from being watered again
```

soil_sensor_water_pump_led_water_level §

```
    ledStatus();                                //Runs void below named ledStatus to adjust light status
  }
}

void ledStatus () {
  if (digitalRead(Water_Level_Sensor) == LOW) { //This code runs if the reservoir is empty
    digitalWrite(redLED, HIGH);                 //Turns red LED ON
    digitalWrite(greenLED, LOW);                //Turns green LED OFF
  } else {                                      //This code runs if the reservois is not empty
    digitalWrite(greenLED, HIGH);               //Turns greeen LED ON
    digitalWrite(redLED, LOW);                  //Turns red LED OFF
  }
}

void pumpON() {                                 //This code is in charge of turning the pump ON
  ledStatus();                                  //Runs void below named ledStatus to adjust light status
  digitalWrite(relay, HIGH);                    //Turns relay ON
  watered = true;                               //Set value of watered to true to prevent plant from being watered again
  waterPumpCounter = waterPumpCounter + 1;      //Everytime the pump is turned the variable +1 and stores the value
  ledStatus();                                  //Runs void below named ledStatus to adjust light status
}

void pumpOFF() {                                //This code is in charge of turning the pump OFF
  ledStatus();                                  //Runs void below named ledStatus to adjust light status
  digitalWrite(relay, LOW);                     //Turns relay OFF
  ledStatus();                                  //Runs void below named ledStatus to adjust light status
}
```

Figure 22-25

Le code ci-dessus est long mais simple. Le code commence par déclarer les pin et leur fonction (INPUT, OUTPUT). Ensuite, le code vérifie si l'état du niveau d'eau est assez haut pour arroser la plante et fait une appelle à la fonction ledStatus() qui s'occupent de mettre à jour les deux LED selon le niveau d'eau. Le code continue par vérifier si le sol doit être arrosé à l'aide des deux capteurs de niveau d'eau. Ensuite il y a un gros if else statement qui permet à la plante de s'arroser si le sol est sec et sinon il ne fait rien. Il y a aussi une fonction if else qui est permet un delay pour que le sol absorbe l'eau et vérifie l'humidité après ce dernier pour voir si le sol est assez humide. Finalement, à la fin du code se trouve les deux fonctions qui contrôlent la pompe.

La fonction pumpON() allume la pompe à l'aide du relay et met à jour les deux LED et la fonction pumpOFF() fait la même chose mais au lieu de l'allumer il l'éteint. Et la fonction ledStatus() mentionnée se trouve à la toute fin.

2.3 Prototype III

Pour le troisième prototype, on s'est concentré sur la construction du réservoir et planteur physique. Avec ce prototype, on voulait finaliser l'emplacement de toutes les composantes et les placer dans leur positionnement final. Pour ce prototype, on a coupé un seau et on le place sur l'autre ceci créer la séparation entre les plantes et le réservoir. On a placé la pompe et le float dans l'eau et on a fait passer les fils par un trou dans le côté du seau. La pompe est fixée au fond pour qu'il ne bouge pas.

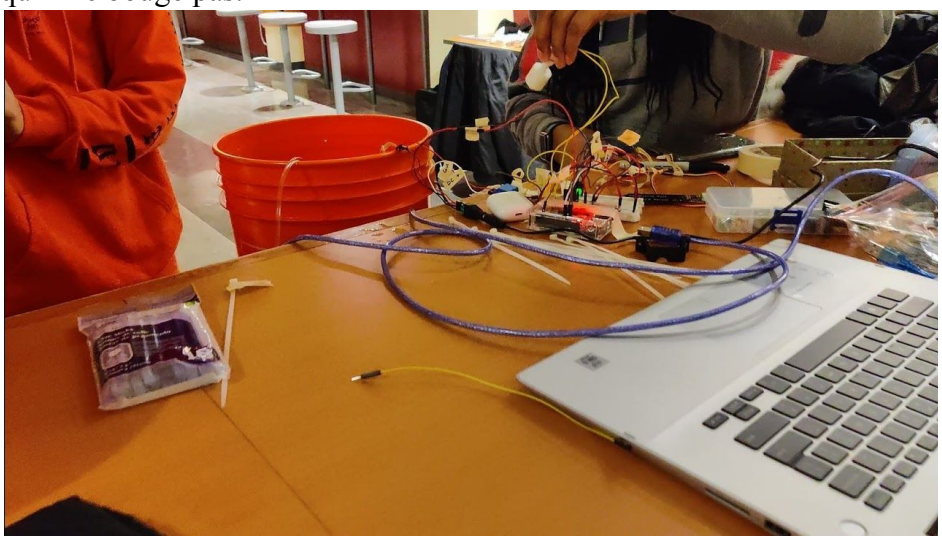

Figure 26

Ensuite on a fixé le tube autour du rebord du seau pour le planteur. Et quand on passe le tube dans le réservoir on coupe l'excès pour ne pas avoir trop de tube pour faire passer l'eau. Ceci augmente la vitesse de l'arrosage de l'eau.

Figure 27

On a testé le code développé au prototype deux pour vérifier qu'il fonctionne avec le nouvel environnement. On a testé le code à chaque étape de la construction pour assurer le fonctionnement du prototype à la fin de la construction.

2.4 Solution finale

Notre produit final est très similaire à notre troisième prototype. Notre troisième prototype est un prototype fonctionnel donc on voulait juste améliorer l'apparence physique du prototype, l'organisation des fils, finaliser l'étui pour l'Arduino L'image à droite (figure 28) est notre produit final sans l'étui Arduino :

Figure 28

On y a ajouté la couleur noir pour qu'il ne sort pas de l'ordinaire dans les édifices de l'université. Ensuite, on a décidé de souder les fils à une PCB board pour qu'ils soient plus organisés et pour éliminer le breadboard qu'on utilisait auparavant. Ceci a rendu nos fils plus compact et organisé et nous on permis de placer l'arduino avec toutes ses fils dans l'étui qui se trouvent à la droite dans le figure 29

Figure 29

Le prototype final est complètement fonctionnel est sous budget et à une belle apparence physique. Notre produit répond à tous les besoins du client efficacement.

2.4.1 LDM (Liste des Matériaux)

Nomenclature des matériaux					
No	**Description du composant**	**Quantité**	**Prix unitaire**	**Montant TTC (CDN$)**	**Fournisseur**
1	Lot Résistances/LED	1	12.99	12.99	Amazon
2	Lot de fils mâles et femelles	1	11.95	11.95	Amazon
3	Capteur de niveau d'eau	1	11.99	11.99	Amazon

4	Capteur d'humidité du sol	1	12.88	12.88	Amazon
5	Pompe à eau	1	14.28	14.28	Amazon
6	Tube (8ft)	1	2.23	2.23	Amazon
8	Pot	2	4.85	8.97	Home Depot
			Total	75.29$	

2.4.2 Liste d'équipements

Énumérez tout l'équipement qui est nécessaire pour construire cette partie.

- Microcontrôleur Arduino
- Soudure
- Ruban
- Ziptie
- Imprimante 3D

3 Comment utiliser le prototype

Vu que notre système est entièrement automatisé, il n'y pas grand chose à faire pour l'utilisation sauf de connecter le système à une prise électrique et de mettre de l'eau dans le réservoir et le planteur est prêt à se faire fonctionner.

Remplir le réservoir

1. Enlevez le bouchon.

2. Ajoutez l'eau au réservoir.

3. Remettez le bouchon.

À noter

• Ne dépassez pas 4 L, vous pouvez assurer cela en regardant quand le DEL s'allume vert.

• Si nécessaire, utilisez un entonnoir pour éviter les dégâts.

• L’eau peut être ajoutée à n’importe quel moment.

• Il est recommandé d’arrêter la pompe si elle est en marche avant de remplir le réservoir.

4 Comment maintenir le prototype

Laver la pompe

1. Ajoutez de l’eau au réservoir.
2. Déconnecter le tuyau de la pompe qui transmet de l’eau en haut du planteur.
3. Mettre la pompe en marche.
4. Ajoutez maintenant un liquide lavant et désinfectant au réservoir.
5. Remettez la pompe en marche.
6. Si le liquide lavant et désinfectant fait des bulles, ajoutez de l’eau encore une fois au réservoir et mettez la pompe en marche.

Laver le reservoir

1. Retirez le planteur du haut doucement pour ne rien casser.
2. Vous pouvez maintenant accéder au réservoir du bas et vous pouvez nettoyer comme vous le souhaitez.

Dépannage

Si l’Arduino et la pompe ne se connectent pas, essayez à nouveau. Si l’Arduino et la pompe ne se connectent toujours pas, vérifiez que l’Arduino ou la pompe est activé. S’il y a des fuites, arrêtez immédiatement la pompe et la débrancher du prise. Envoyez un courriel au producteur qui va réparer la pompe pour vous.

Questions ou commentaires?

Pour toutes questions et commentaires, écrivez à ckuli037@uottawa.ca ou jroqu009@uottawa.ca.

5 Conclusions et recommandations pour les travaux futurs

Ce projet de conception nous a permis d'apprendre plusieurs leçons qui nous permettront de devenir de meilleurs ingénieurs:

- L'utilisation de la pensée conceptuelle afin de créer divers concepts et d'ensuite choisir les aspects de chacun.
- Travailler en équipe à distance.
- Appris à utiliser Arduino de manière efficace afin de concevoir des produits.

Pour nos futurs travaux, il serait bien de se familiariser avec d'autres technologies et logiciels d'ingénierie, d'utiliser des matériaux et outils avec une meilleure qualité et passer plus de temps à créer plusieurs prototypes afin de pouvoir comparer l'efficacité de chacun.

6 Bibliographie

Amazon. "*CONTINENTAL AWS- 10 Système d'arrosage automatique pour conteneurs"*. Consulté le 4 octobre 2020, à:

https://www.amazon.ca/CONTINENTAL-AWS-10-Automatic-Watering-containers/dp/B007NHRYH0/ref=asc_df_B007NHRYH0/?tag=googleshopc0c-20&linkCode=df0&hvadid=292994535539&hvpos=&hvnetw=g&hvrand=10085592282199130865&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9000668&hvtargid=pla-314799926864&psc=1

Amazon. "*Elegoo Lot de 120 fils multicolores Dupont 40 broches mâles vers femelles, 40 broches mâles vers mâles, 40 broches femelles vers femelles pour arduino"*. Consulté le 20 octobre 2020, à:

https://www.amazon.ca/Haitronic-Arduino-component-starter-resistors/dp/B06XX3VRM6/ref=sr_1_77?dchild=1&keywords=arduino+wire+kit&qid=1603382410&sr=8-77

Amazon. "*Esotec 101100 Kit complet d'arrosage/ Système d'irrigation solaire complet avec 15 goutte-à-goutte."* Consulté le 4 octobre 2020, à:

https://www.amazon.fr/Esotec-Solar-Bew%C3%A4sserungs-Set-6mm-16-Schlauchl%C3%A4nge/dp/B01HTWM52C?ref_=Oct_s9_apbd_omwf_hd_bw_b4jc7pn&pf_rd_r=FBKJQDBFBXDTSXMB3CHJ&pf_rd_p=7fbd1195-844e-571a-a23c-8c812d7d00e8&pf_rd_s=merchandised-search-10&pf_rd_t=BROWSE&pf_rd_i=4338553031

Amazon. *"Gikfun capacitif Soil Moisture Sensor résistant à la corrosion pour Arduino Détection de l'humidité Jardin d'arrosage DIY (lot de 2 pcs) Ek1940"*. Consulté le 20 octobre 2020, à:

https://www.amazon.ca/Gikfun-Capacitive-Corrosion-Resistant-Detection/dp/B07H3P1NRM/ref=sr_1_13?crid=1CATUQ3SULWC4&dchild=1&keywords=soil+humidity+arduino&qid=1603379879&sprefix=soil+hum%2Caps%2C201&sr=8-13

Amazon. "*Kit de démarrage Haitronic 254 en 1 composante Arduino avec résistance (10 sortes: 10R-1M), LEDs(5 couleurs), commutateur, Potentiomètre B50K, Embase de broche pour Arduino UNO R3 Leonardo, MEGA2560, Nano, Rasberry Pi"*. Consulté le 20 octobre 2020, à:

https://www.amazon.ca/Haitronic-Arduino-component-starter-resistors/dp/B06XX3VRM6/ref=sr_1_77?dchild=1&keywords=arduino+wire+kit&qid=1603382410&sr=8-77

Amazon. "*LEDGLE Mini pompe submersible 12 V - Pompe à eau électrique sans balais - Ultra silencieuse - IP68 - Design double usage - Pour aquarium, hydroponie, aquaponie, fontaines, bassins et aquariums statuaires*". Consulté le 20 octobre 2020, à:

https://www.amazon.ca/LEDGLE-Submersible-Ultra-Quiet-Dual-Purpose-Hydroponics/dp/B07G74MDLM/ref=sr_1_3?dchild=1&keywords=Arduino+Water+Pump&qid=1603398216&sr=8-3

Amazon. *"Penn plax - Tuyau flexible 2.5m"*. Consulté le 20 octobre 2020, à :

https://www.amazon.ca/Penn-Plax-Standard-Airline-Tubing-Accessories/dp/B0002563MM/ref=sr_1_19?dchild=1&keywords=PVC%2Btubing&qid=1603397383&sr=8-19&th=1

Amazon. "*Urbest Aquarium filaire liquide Capteur de niveau d'eau Interrupteur à flotteur"*. Consulté le 20 octobre 2020, à:

https://www.amazon.ca/URBEST-Aquarium-Liquid-Sensor-Switch/dp/B00NW7RX0S/ref=sr_1_8?dchild=1&keywords=water+level+sensor+arduino&qid=1603387429&sr=8-8

Best Buy Canada. "*Jardin intérieur intelligent/capsule de basilic de Click & Grow."* Consulté le 4 octobre 2020, à:

https://www.bestbuy.ca/fr-ca/produit/jardin-interieur-intelligent-capsule-de-basilic-de-click-grow-sgs1us-paquet-de-3-blanc/10660703?cmp=knc-s-71700000055311646&gclid=CjwKCAjwiOv7BRBREiwAXHbv3FcFpPydDoFzhNnAa27J6apVWKwTAcP2KFHDERaHmweQtWyZMijt-RoCVFUQAvD_BwE&gclsrc=aw.ds

The Home Depot. "*The Home Depot Orange Home Depot Logo Bucket, 19L"*. Consulté le 20 octobre 2020, à:

https://www.homedepot.ca/product/the-home-depot-orange-home-depot-logo-bucket-19-l/1000143871

APPENDICES

APPENDICE I: Fichiers de conception.

Pour accéder à tous les fichiers de conception, veuillez suivre le lien suivant: https://makerepo.com/JuanRoque/gng1503a7jardinire-automatique-