

Livrable G: Prototype II et rétroaction du client

Équipe A15

Yahya Mostafa

Eden Kindja Nehema

Xavier Riva

Maysa Yadaas

Le 22 Octobre 2020

Table des matières

Introduction	2
En quoi consiste ce prototype	2
Résultats de tests précédents	2
Liste de spécifications	3
Spécification des matériaux	3
Fidélité et faisabilité du prototype	4
Fidélité	4
Faisabilité	5
Modèle numérique/expérimental	9
Rétroaction du client par rapport au prototype 1	12
Conclusion	13

Introduction

Dans ce livrable , on se focalise sur notre deuxième prototype qui est un sous-système critique qui est la surveillance de notre planteur . Pour établir cette tâche, on a eu recours à un capteur de niveau d'eau Arduino et on a représenté ci-dessous les informations nécessaires sur le produit et son fonctionnement et les tests qu'on a établis avec le capteur utilisé .

En quoi consiste ce prototype

Pour le prototype 2, nous avons décidé de traiter du système de surveillance puisque c'est un sous système critique. En effet, nous planifions tout finaliser les électroniques du système ainsi que le code qui les contrôlent.

Nous imaginons que le système de surveillance consistera en un capteur de niveau d'eau et de trois DEL indicatrices de couleurs rouge, vert et jaune. Le tout sera connecté à une carte arduino qui s'occupera du contrôle. Les DELs affichent une couleur différente selon la valeur du niveau d'eau.

Résultats de tests précédents

Le test de trajectoire du livrable précédent a démontré que la trajectoire était en effet le problème. La corde était mouillée partout mais une flaque d'eau s'est créée au bas de la parabole. Il faut donc adapter la trajectoire à une trajectoire plus verticale pour le prototype 3 pour éviter des pertes majeures en eau. De plus, la plante à tout consommé l'eau dans le réservoir (150mL) et ce en 3 jours. Cette information nous donne une idée de la taille de réservoir qu'il faudra pour le prochain prototype.



Liste de spécifications

Spécification des matériaux

Composante	Description	Quantité	Coût unitaire (\$)	Quantité x Coût
Arduino Uno	Un micro-contrôleur qui est capable de contrôler plusieurs composantes électroniques à l'aide de code. L'Arduino sera utilisé dans ce prototype pour contrôler un capteur de niveau d'eau et une lumière DEL.	1	0	0
Fils électriques	Les fils électriques sont nécessaires pour connecter toutes les composantes électriques.	10	0	0
Senseur de niveau d'eau	Le capteur de niveau d'eau détecte le niveau d'eau. Par exemple, plus l'eau couvre le capteur, plus les valeurs numériques seront élevées.	2	2,00	4
Lumière DEL	La lumière DEL est nécessaire pour notifier aux utilisateurs que le niveau d'eau est bas.	1	0	0
Eau	L'eau est utilisée pour tester le capteur de niveau d'eau.	2 verres	0	0

Fidélité et faisabilité du prototype

Fidélité

Caractéristique	Prototype 2	Produit final	Fidélité
Taille	62mmx20mmx8mm	62mmx20mmx8mm	Le produit acheté est de petite dimension donc ça convient à notre réservoir
Matériel	Mélange (métal, plastique)	Mélange (métal, plastique)	La composition du capteur de niveau d'eau ne constitue aucun problème quand il est dans le réservoir (Il est conçu pour être dans l'eau donc il n'y a pas de risque De rouiller ou de moisissure et le capteur ne dégage aucun produit ou matière nocive qui peut nuire à l'eau et affecter la plante.
Poids	0.2 kg	0.2 kg	Ce poids est idéal pour ne pas prendre de l'espace dans le réservoir et donc pour avoir plus d'eau et moins de perte de temps à remplir chaque fois le réservoir quand il ya un manque
Fonctionnalités	Surveillance du niveau d'eau dans le planteur en	Surveillance du niveau d'eau dans le planteur en	C'est le meilleur capteur qu'on ait trouvé pour notre

	<p>faisant expose une série de fils parallèles à des gouttes d'eau afin de le déterminer par La conversion d'eau en signal analogique et donc les valeurs obtenues seront lues sur la carte de Arduino pour obtenir un signal d'alarme (quand le réservoir est vide , le LED s'Allume pour prévenir qu'il faut remplir pour que la plante puisse absorber de nouveau de l'eau , aussi il s'allume quand le réservoir est presque vide pour nous rappeler d'ajouter de l'eau et finalement quand on est entrain de remplir il s'allume quand le réservoir est rempli à sa capacité maximale pour que l'eau ne déborde pas)</p>	<p>faisant expose une série de fils parallèles à des gouttes d'eau afin de le déterminer par La conversion d'eau en signal analogique et donc les valeurs obtenues seront lues sur la carte de Arduino pour obtenir un signal d'alarme (quand le réservoir est vide , le LED s'Allume pour prévenir qu'il faut remplir pour que la plante puisse absorber de nouveau de l'eau , aussi il s'allume quand le réservoir est presque vide pour nous rappeler d'ajouter de l'eau et finalement quand on est entrain de remplir il s'allume quand le réservoir est rempli à sa capacité maximale pour que l'eau ne déborde pas)</p>	<p>prototype 2 et prototype final. On a pu créer et essayer le code qu'on va utiliser pour surveiller la quantité d'eau dans le réservoir</p>
--	---	---	---

Faisabilité

La faisabilité de notre prototype peut être établie à l'aide des facteurs de TELOP. Alors quelles sont les forces et faiblesses de notre prototype en suivant les facteurs TELOP (Technique, Économique, Légal, Opérationnel, Planification)?

- Technique: Étant donné que nous sommes des étudiants, nous n'avons pas assez d'expertise dans le domaine mais nous travaillons avec ce que nous savons et apprenons. La réalisation de ce prototype a aussi été basée sur la recherche des ressources techniques.
- Économique: Ce prototype n'a pas été coûteux d'où le respect du budget planifié pour le prototype 2.
- Légal: Il n'y a aucune fraude, aucun problème juridique et aucun danger sur la structure, la conception et l'utilisation du prototype.
- Opérationnel: Il y a une bonne cohésion d'équipe et tout le monde est apte à réaliser sa tâche pour la réalisation du prototype.
- Planification : Le prototype est censé être prêt avant le 12/11/2020 à 23h59. Ce qui est faisable.

Plan d'essai

N° de test	Objectif de 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)
1	Le premier test à pour but de vérifier si les composantes du système de surveillance que nous envisageons fonctionne correctement.	Il faut assembler les composantes électriques sur une plaque de test et les alimenter avec une pile externe. Ceci permettra de voir si les composantes fonctionnent ensemble correctement. Ce test doit être fait en parallèle au test 3 puisque les composantes électriques et le code doivent être combinées pour fonctionner.	Les composantes du système de surveillance fonctionnent bien sauf que nous devons trouver un moyen pour empêcher le contact avec l'eau des parties qui ne sont pas waterproof. Nous envisageons d'utiliser du Flex Seal Liquid pour résoudre ce problème. Il s'agit de tremper la section à rendre waterproof dans le produit et attendre que ça sèche.	10 Novembre au 12 Novembre (3 jours)
2	Le deuxième test à pour but de tester la durée de vie du système de surveillance.	Le test consiste à alimenter le circuit avec une pile 9V et de mesurer expérimentalement le temps avant que la pile s'épuise. Ce test devrait être fait plusieurs fois idéalement pour avoir un résultat fiable.	Après une rétroaction avec les professeurs du laboratoire, on a réalisé que le client aimait plus l'idée d'alimenter les composantes électroniques avec une branche au lieu d'une batterie. Évidemment, les batteries doivent être remplacées à chaque X journées mais ceci rend la jardinière portable. Cependant, d'un point de vue global, ça vaut mieux d'utiliser une branche et de limiter la portabilité que de remplacer des centaines de batteries à travers le campus. De plus, les batteries coûteraient très cher si on cherche à	10 Novembre au 12 Novembre (3 jours)

			avoir une autonomie raisonnable.	
3	Le troisième test à pour but de vérifier le fonctionnement du code destiné à être utilisé avec les composantes électriques.	Il faut écrire le code et ensuite le faire rouler sur la carte arduino avec les composantes (test 1) pour tester si les entrées des composantes donnent les sorties correctes. Nous pouvons tremper le capteur de niveau d'eau dans un verre d'eau et vérifier si les LED correspondantes s'allument.	Le code a été écrit et il fonctionne bien. De ce fait, nous avons décidé d'utiliser deux capteurs de niveau d'eau contrairement au plan original d'en utiliser seulement un, cela permet d'avoir une précision plus accrue sur la lecture du niveau d'eau dans le réservoir. En effet, nous avons découvert que les capteurs de niveau d'eau ne sont pas très fiables donc en avoir plusieurs permet de limiter leur output à 1 ou 0 versus une gamme de valeur. Ceux-ci sont installés(l'un en bas et l'autre au milieu du réservoir), chaque capteur devra détecter le niveau d'eau indépendamment de l'autre et communiquer une valeur binaire(1 ou 0). En commençant par le résultat du capteur du bas suivi de l'autre, si la carte Arduino reçoit 1 1, la LED s'éteint; si elle reçoit 1 0, elle s'allume, et si elle reçoit 0 0 elle clignote.	10 Novembre au 12 Novembre (3 jours)

			0: Pas captage d'eau 1:Captage d'eau	
--	--	--	---	--

Critère d'arrêt

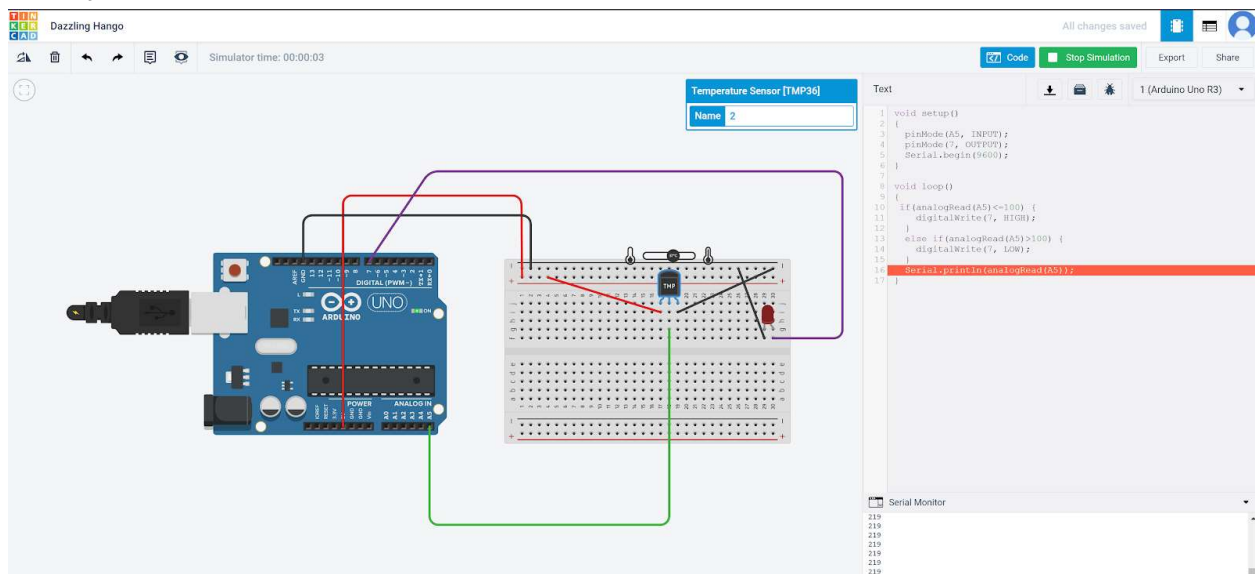
- Pour le test n°1: Lorsqu'on aura assemblé les composantes électroniques et qu'on est sûr qu'elles fonctionnent correctement entre-elles
- Pour le test n°2: Lorsqu'on aura fait le test d'autonomie deux fois
- Pour le test n°3: Lorsque le code fonctionnera correctement avec les composantes

Critère d'arrêt globale:

Quand on sera sûr que le prototype est efficace et qu'on aura trouver les erreurs majeures à améliorer dans la prochaine itération.

Modèle numérique/expérimental

Avant la création du prototype, on a fait une création digitale sur TinkerCad. Malheureusement, TinkerCad n'avait pas un capteur de niveau d'eau mais un capteur de température est un substitut excellent. Par exemple, lorsque le capteur détecte une haute température, une haute valeur dans le "Serial Monitor" est affichée. En comparaison, lorsque le capteur de niveau d'eau détecte un haut niveau d'eau, une haute valeur dans le "Serial Monitor" est ainsi affichée. Sur ce, le code, le nombre de composants utilisés et le placement de ses composants seront les mêmes si on avait utilisé un capteur de niveau d'eau. Lorsque les composants seront livrés, la création physique prendra très peu de temps. Voici le modèle TinkerCad créé pour ce prototype:



La lumière DEL s'allume lorsque les valeurs sont plus petit que 100 dans le "Serial Monitor". Une valeur de 100 ou moins est affiché lorsque la température détecté est de -1 °C ou moins. Voici le code utilisé:

```
void setup()
{
  pinMode(A5, INPUT);
  pinMode(7, OUTPUT);
  Serial.begin(9600);
}

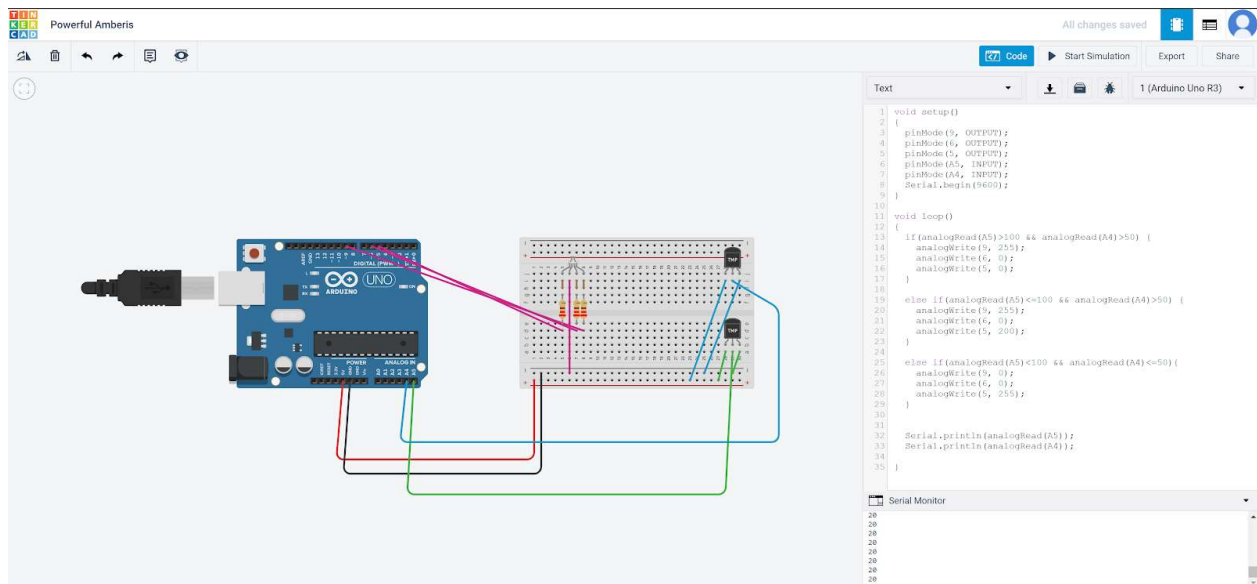
void loop()
{
  if(analogRead(A5)<=100) {
    digitalWrite(7, HIGH);
  }
}
```

```

}
else if(analogRead(A5)>100) {
  digitalWrite(7, LOW);
}
Serial.println(analogRead(A5));
}

```

Dans le modèle ci-dessus, un seul senseur de niveau d'eau n'est pas assez pour nous avertir qu'il n'y a pas assez d'eau (car il est de 60mm de hauteur). Donc on a décidé d'opter pour un système avec deux senseurs placé à deux hauteurs différentes dans le réservoir. Cette modèle est brièvement représenter sur la création TinkerCad ci-dessous:



Encore une fois, on a utilisé des senseurs à température comme substitut. Imaginons qu'ils sont des senseurs à niveau d'eau et que les deux senseurs sont placés à deux hauteurs différentes. Notez que le premier senseur est plus haut que le deuxième. Si le premier senseur affiche une valeur plus grande que 100 (par exemple), le DEL RGB est vert. Si le premier senseur affiche une valeur plus petit que 100 mais le deuxième est plus grand que 50, le DEL RGB est jaune. Finalement, si le deuxième senseur affiche une valeur plus petite que 50, le DEL RGB est rouge. Voici le code utilisé:

```

void setup()
{
  pinMode(9, OUTPUT);
  pinMode(6, OUTPUT);
}

```

```

pinMode(5, OUTPUT);
pinMode(A5, INPUT);
pinMode(A4, INPUT);
Serial.begin(9600);
}

void loop()
{
  if(analogRead(A5)>100) {
    analogWrite(9, 255);
    analogWrite(6, 0);
    analogWrite(5, 0);
  }

  else if(analogRead(A5)<=100 && analogRead(A4)>=50) {
    analogWrite(9, 255);
    analogWrite(6, 0);
    analogWrite(5, 200);
  }

  else if(analogRead(A4)<50){
    analogWrite(9, 0);
    analogWrite(6, 0);
    analogWrite(5, 255);
  }

  Serial.println(analogRead(A5));
  Serial.println(analogRead(A4));

}

```

Voir l'annexe pour la création physique du modèle ci-dessus. Note bien qu'on avait pas un DEL RGB donc au lieu on a utilisé un DEL bleu qui s'éteint lorsque le réservoir est rempli (senseur supérieur détecte de l'eau), qui s'allume lorsque le réservoir est partiellement rempli (senseur supérieur ne détecte pas l'eau mais senseur inférieur en détecte) et qui clignote lorsque le réservoir n'est pas rempli (deux senseurs ne détectent pas l'eau).

Rétroaction du client par rapport au prototype 1

Après la troisième rencontre avec le client, il est retenu:

- Il serait préférable d'inclure un trou dans le réservoir pour le remplissage. Car ça serait compliqué à chaque fois d'ouvrir la structure pour remplir le réservoir.
- Que la structure et le design ne devraient pas être changés car ils sont beaux et très simples.

Conclusion

En conclusion , ce livrable nous a aidé à mieux nous focaliser sur le système de surveillance et à mieux comprendre comment utiliser et gérer ce prototype pour pouvoir améliorer ce sous-système pour notre produit final . En d'autres termes , on a utilisé nos connaissances qu'on a apprises pendant le lab Arduino pour créer un code et effectuer différents tests sur le prototype .

Annexe

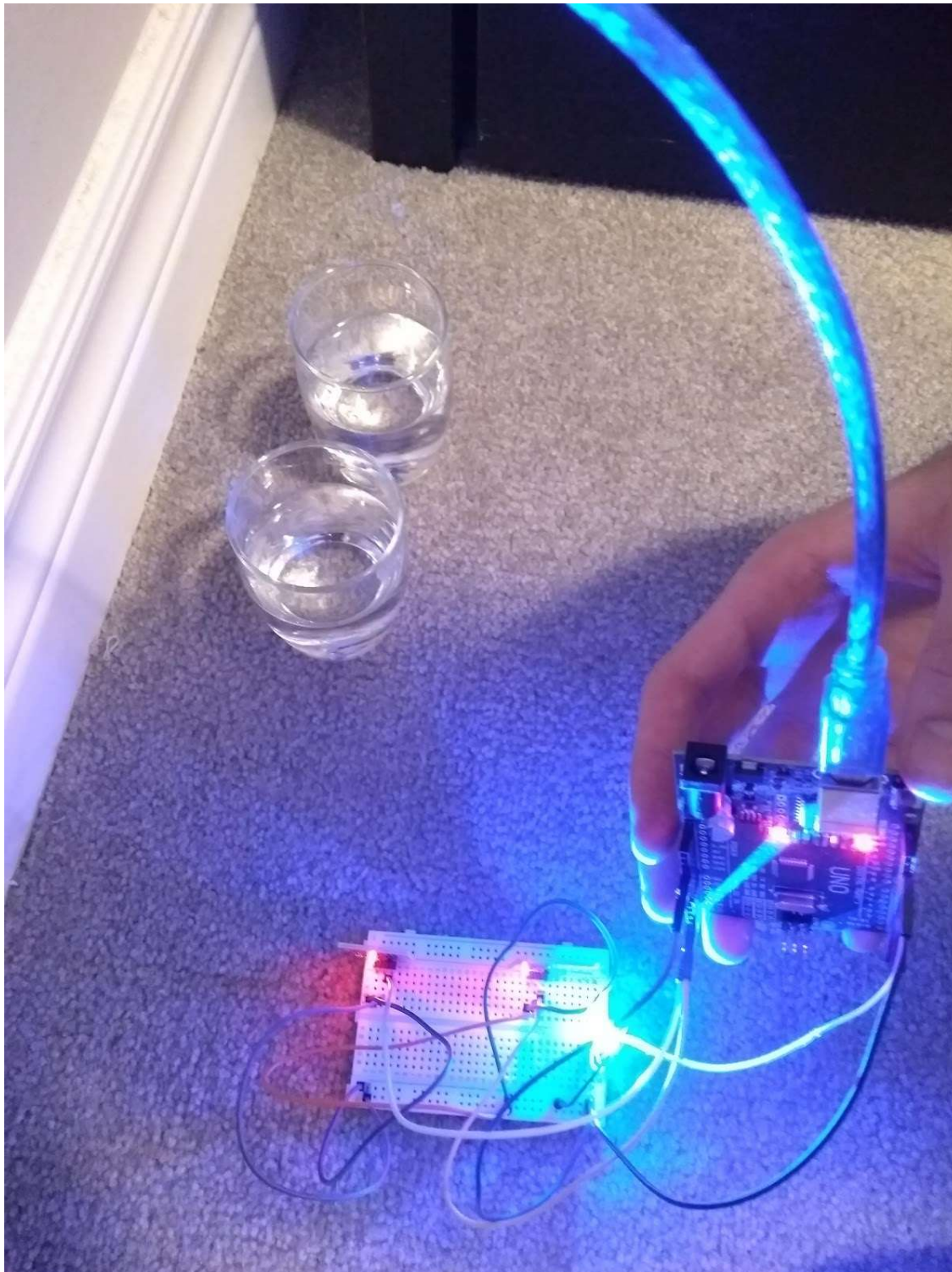


Figure 1 - Photo lorsque les senseurs ne détectent pas de l'eau (donc il faut remplir le réservoir!) Note que la lumière DEL clignote pendant cette phase.

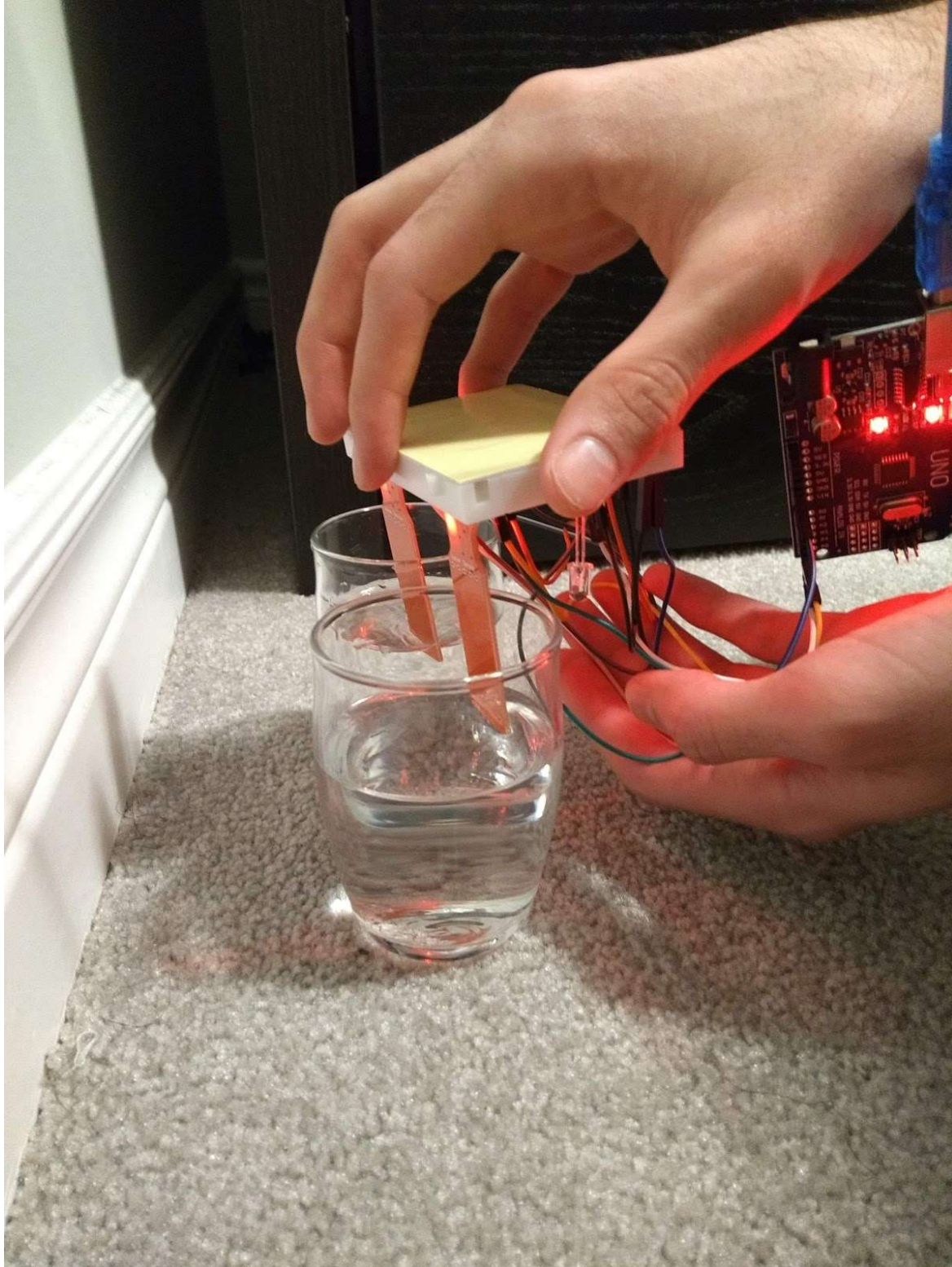
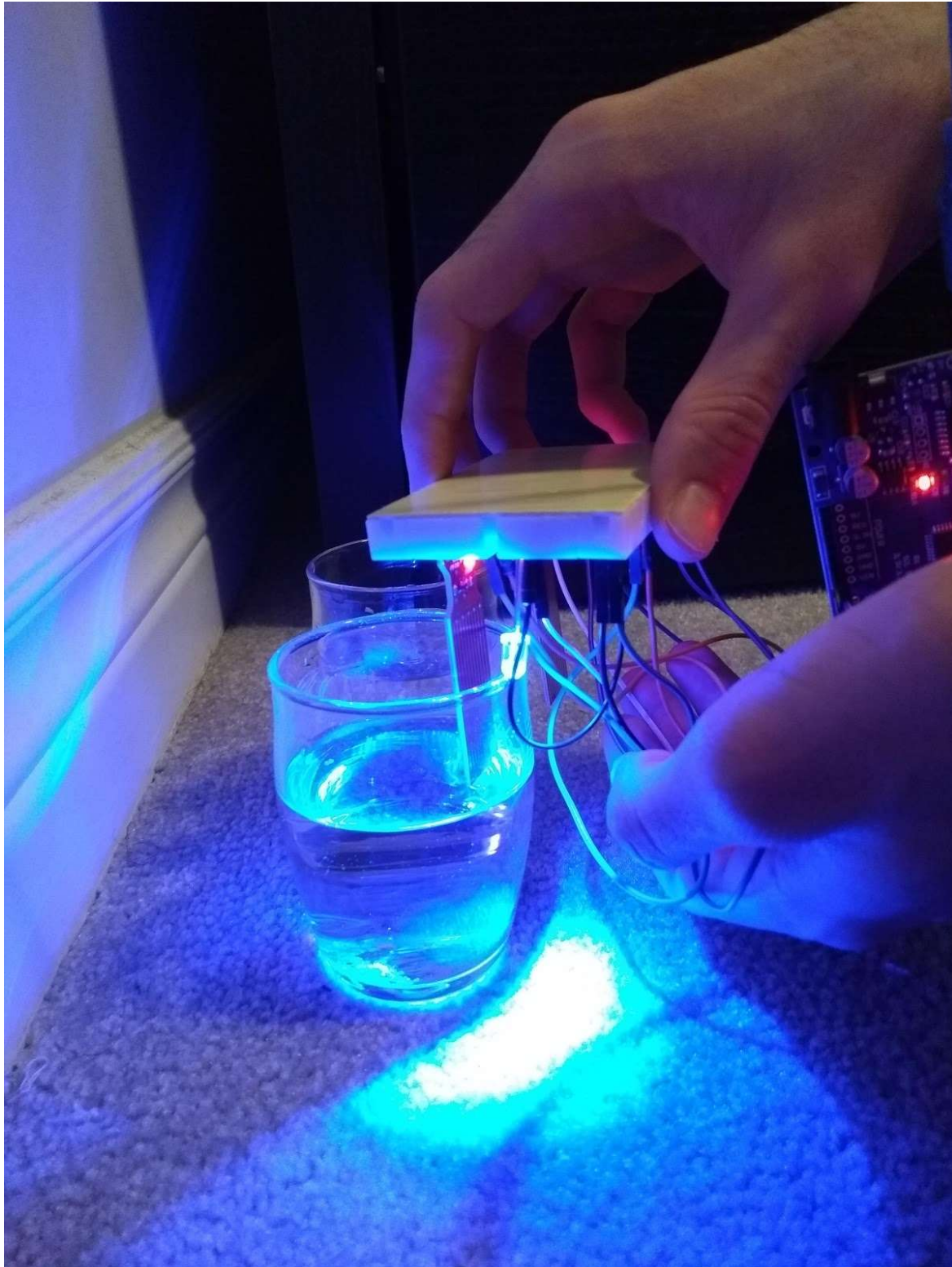


Figure 2 - Photo lorsque les deux senseurs détectent de l'eau (réservoir est rempli avec suffisamment d'eau) Note que la lumière est éteinte pendant cette phase.



**Figure 3 - Photo lorsque le capteur inférieur détecte de l'eau mais le capteur supérieur ne détecte pas de l'eau (réservoir est partiellement rempli)
Note que la lumière est allumée pendant cette phase.**



Figure 4 - Photo du système branché à une prise électrique.

Code utilisé pour les tests:

```
void setup() {
  pinMode(A5, INPUT);
  pinMode(A4, INPUT);
  pinMode(8, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int sensorUp = analogRead(A5);
  int sensorDown = analogRead(A4);

  if(sensorUp>0) {
    digitalWrite(8, LOW);
    Serial.println("Full");
  }

  else if(sensorUp==0 && sensorDown>0) {
    digitalWrite(8, HIGH);
    Serial.println("Half Full");
  }

  else if(sensorUp==0 && sensorDown==0) {
    digitalWrite(8, HIGH);
    delay(500);
    digitalWrite(8, LOW);
    delay(500);
    Serial.println("Not Full");
  }
}
```