Wegrzyn Nicolas

IMA 4

Rapport de projet IMA 4

Hydroponie

Année 2015-2016

Tuteur : Alexandre Boé

Sommaire

1. Introduction

L'hydroponie est une technique de culture qui consiste à cultiver des végétaux hors de la terre. Ce domaine se répand dans de plus en plus de domaines d'application. On retrouve cette technique pour des utilisations domestiques ou encore professionnelles. Le but principal étant de pouvoir contrôler la croissance de la plante.

L'objectif ici est de rendre autonome en énergie ce type de structure. Il s'agit alors de créer un système qui permettra par la suite de ne plus consommer d'électricité par le réseau.

Pour ce faire, nous avons évalué x points importants à réaliser afin de répondre au cahier des charges :

- La gestion de l'humidité et de la température au sein du caisson. Il s'agit de conserver la plante dans les meilleures conditions possibles.
- Le contrôle de l'éclairage au sein de la culture. En effet, étant donné que nous nous situons dans le cadre d'une culture d'intérieur, il est nécessaire de recréer des cycles d'éclairage mais aussi de choisir une couleur de lumière pour la plante.
- La connaissance du niveau du réservoir ainsi que l'apprivoisement en eau de la plante.
- L'alimentation en énergie du système. Elle devra se faire par un panneau solaire qui délivrera de l'électricité.

Enfin la gestion simultanée de tous ces évènements se fera via un Arduino Uno.

Ce rapport rend compte de la démarche et des réalisations effectuées dans le cadre du projet de 4^{e} année dans le département Informatique-Microélectronique-Automatique. Il permet de mettre en évidence les compétences acquises mais aussi les solutions apportées aux problèmes rencontrés durant le déroulement du projet.

2. Présentation du projet

2.1. Objectifs

L'objectif de ce projet est de réaliser un caisson hydroponique autonome pour le fonctionnement. La plante devra pouvoir être alimentée et éclairée seule mais devra également se trouver dans de bonnes dispositions.

Premièrement pour réguler la température ainsi que l'humidité au sein du système, on placera un capteur de température et d'humidité de type DHT11. Celui-ci permet de renvoyer à l'Arduino les informations nécessaires. En cas de dépassement du seuil tolérable définit par l'utilisateur, on mettra en marche deux ventilateur (un par paramètre) afin que la chaleur s'échappe ou que l'air ambiant devienne plus sec. Enfin, on insèrera une résistance de puissance afin de garder une chaleur ambiante dans le caisson.

En ce qui concerne l'éclairage, on choisira des LEDs RGB de forte puissance assurant une bonne luminosité. Il est intéressant d'utiliser ce type de LED pour contrôler le type de couleur que l'on souhaite imposer à la plante. On se penchera sur le moyen de choisir cette fréquence lumineuse et comment introduire des cycles d'éclairage afin d'assurer des périodes de « nuit » à la plante.

Pour l'alimentation en eau du système, il s'agira de contrôler une pompe à eau afin d'effectuer un goutte à goutte sur la plante. Il faudra également renvoyer une information à l'utilisateur via l'Arduino sur l'état du niveau de réservoir. Pour cela, on utilisera un potentiomètre relié à un flotteur pour créer un réglage de niveaux hauts et bas.

Pour finir, on s'occupera d'alimenter le système global par un panneau solaire afin que le projet soit totalement autonome en énergie. Celui-ci devra fournir assez de puissance pour alimenter les ventilateurs, l'Arduino, le buzzer ainsi que la résistance de puissance chauffante. Il faudra certainement insérer un MPPT (maximum power point tracker) afin de récupérer le maximum d'énergie possible.

Afin de faire fonctionner ces parties ensembles, il faudra programmer notre Arduino en conséquence. On utilisera l'IDE, plus intuitif et simple d'utilisation, disponible en salle de projet.

2.2. Tâches à réaliser

- Savoir utiliser le DHT11 : Récupérer les informations envoyées par le capteur et les traduire avec l'Arduino et permettre de les fournir à l'utilisateur si besoin sur un port série.

- Pouvoir commander les ventilateurs en fonction d'un seuil d'humidité ou de température délivré par le capteur DHT11.

- Savoir piloter les LEDs RGB de manière à obtenir la nuance de rouge, vert et bleu souhaitée. Créer un programme permettant de créer des cycles d'éclairage et des cycles de « nuit ».

- Recevoir l'information de l'état du niveau du réservoir en permettre de prévenir l'utilisateur en cas de problème.

- Contrôler la pompe à eau de manière à obtenir un arrosage constant selon un débit souhaité pour la plante en question.

- Savoir récupérer l'énergie fournie par le panneau solaire et la transmettre au système.

- Programmer l'Arduino de telle sorte que chacun des programmes fonctionne simultanément avec les autres sans conflit et de manière continue.

Pour cela, il est nécessaire de faire un bilan du matériel pour la commande de chaque partie. En ce qui concerne l'alimentation, on commencera par simuler à partir des alimentations disponibles en salle de TP pour étudier la demande en électricité.

3. Déroulement du projet

3.1. Gestion de la température et de l'humidité

Etude du capteur DHT11 :

Le capteur DHT11 permet de fournir une information en température et en humidité de l'air ambiant, ce qui est tout à fait adapté à notre problématique. Ce composant possède 4 pins dont un pour l'alimentation, un pour la masse et deux pour transmettre les messages. Ceux-ci existent sous formes de messages binaires envoyés sur les pins de l'Arduino. Par exemple, si la température est de 23°c, le DHT11 enverra sous forme binaire la suite : 0001 0111 (signifiant 23 en binaire). En réalité il fournit 5 messages de 8 bits comprenant les messages nécessaires mais aussi un « checksum » par exemple. Le DHT11 fonctionne sous 5V, on peut donc directement utiliser l'Arduino pour l'alimenter.

La résistance chauffante WH25 :

On utilise une résistance de puissance afin de fournir de la chaleur au sein du caisson hydroponique. En l'occurrence, c'est une résistance qui permet de dissiper 25W, ce qui suffisant dans le cadre de notre application. Pour obtenir de telles conditions, on l'alimentera directement sur une alimentation de 24V afin qu'elle fournisse environ 3A, autrement, la résistance ne chaufferait pas correctement. On prendra soin de la coller via une pâte chauffante à un dissipateur de chaleur afin de créer un radiateur

Les ventilateurs :

Pour ventiler le système, j'ai choisi deux ventilateurs de PC. Ce sont des ventilateurs alimentés sous 12V et ils devront être commandés en tout ou rien. Soit le seuil fixé est atteint et ils fonctionnent soit ils sont éteints. Il s'agira de placer celui de l'humidité en haut du caisson afin d'éliminer celle-ci qui se condensera en hauteur. A l'inverse, on créera une convection de chaleur en bas du caisson en fixant le ventilateur au dissipateur (avec la résistance). Dans notre cas, on utilise une alimentation externe pour faire fonctionner les ventilateurs.

Le montage :

Voici finalement le montage que l'on obtient. A noter que le radiateur n'est pas représenté et que les moteurs symbolisent les ventilateurs.

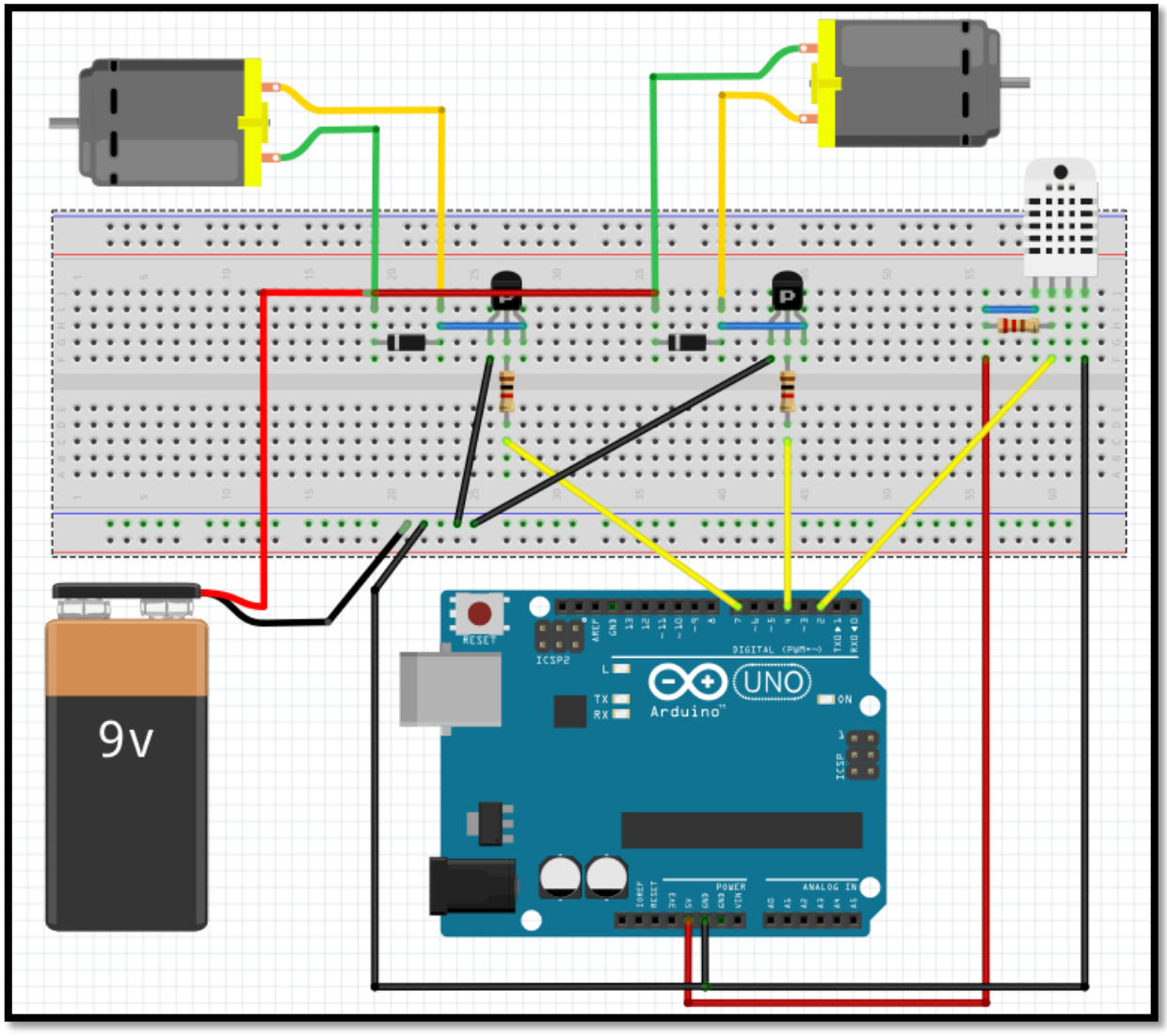

Figure 1: montage du contrôle des ventilateurs

On utilise deux transistors afin d'effectuer le contrôle des ventilateurs. On insère également une diode de protection pour contrer le retour de courant.

3.2. Eclairage du système

Les LEDs RGB :

On utilise dans notre cas, une série de LEDs RGB afin de faire varier la couleur de celles-ci. Malheureusement les LEDs utilisées sont des CMS (composants montés en surface) et non pas des LEDs classiques avec des « pattes ». Il n'a pas été possible de réaliser un ruban de LEDs complet mais quelques une suffisent pour notre application. Chaque LED peut dissiper une centaine de mW. En réalisant le ruban complet, il est possible d'atteindre la dizaine de Watts. De plus, le composant demande un courant de 20mA, ce qui est adapté avec ce que peut fournir l'Arduino. Voici ce que peut fournir chaque LED :

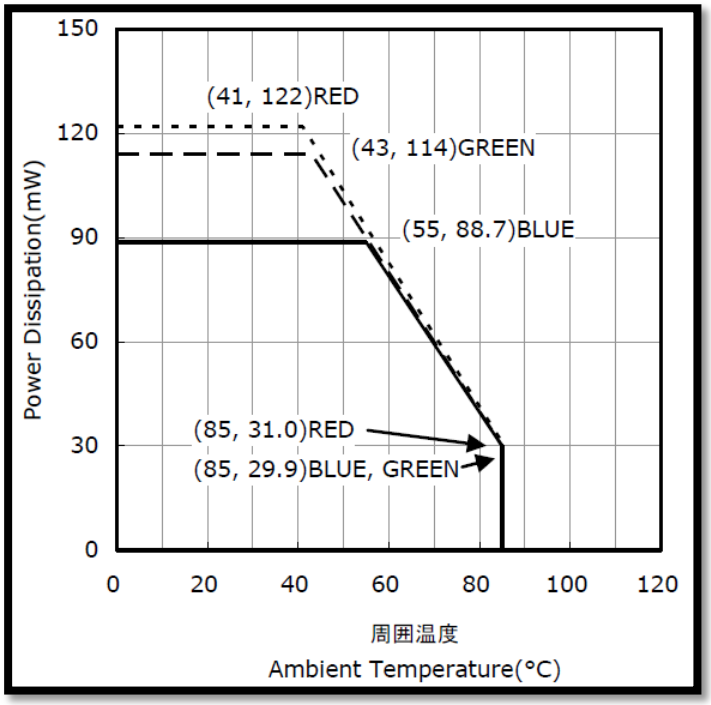

Figure 2: Puissance dissipée par les LEDs

D'autre part, pour régler notre intensité lumineuse, il faut insérer des résistances variables reliées chacune sur un port analogique de l'Arduino, celui-ci lira la valeur et transcrira un signal à appliquer sur la LED en question (une résistance influe sur une couleur).

Le montage :

Voici une représentation du montage effectué. Pour simplifier, une seule LED est représentée mais en réalité il suffirait de mettre chaque « patte » en parallèle avec la correspondante.

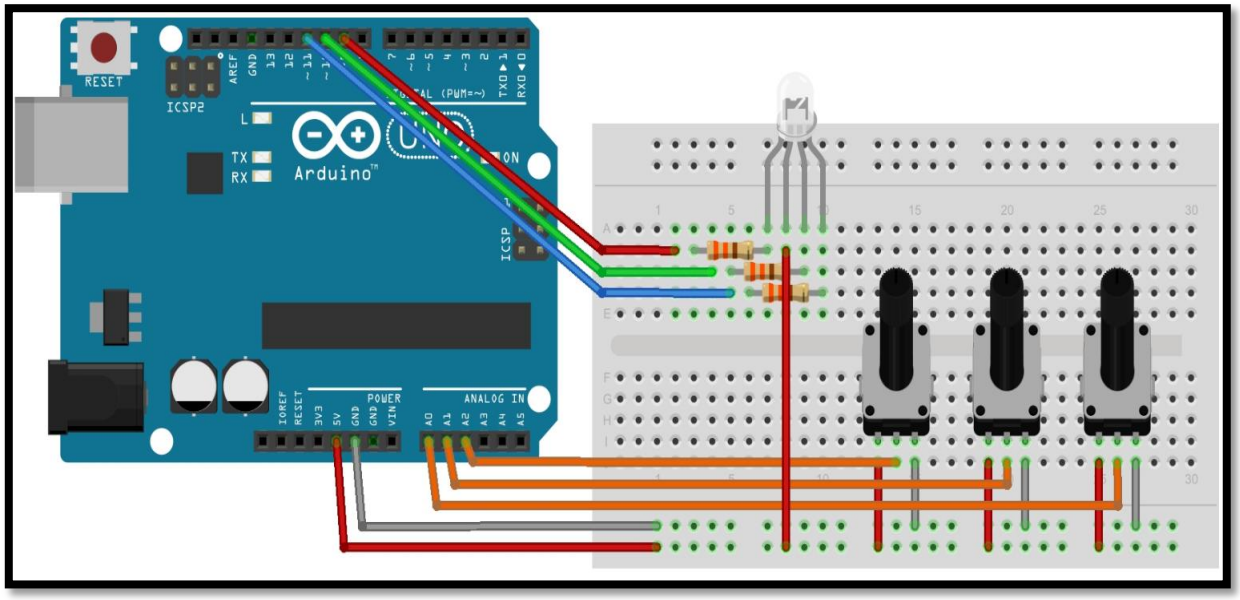

Figure 3: Montage de contrôle des LEDs

On applique trois résistances de protection afin de limiter le courant dans chacune des LEDs internes. Il faut également prendre soin de vérifier le courant que peuvent tolérer les LEDs, il est différent selon la couleur.

3.3. Alimentation en eau et gestion du niveau du réservoir

Pour cette partie, on s'intéresse au contrôle de la pompe du système ainsi qu'à la connaissance du liquide restant dans le réservoir.

Pour cela, on utilise dans un premier temps un potentiomètre relié à un flotteur comme suit :

Figure 4 : Potentiomètre et flotteur

Il est nécessaire d'utiliser un potentiomètre de grande résistance. En effet, nous ne cherchons pas à dissiper de l'énergie mais bien à simplement recueillir l'information de l'angle du potentiomètre. C'est pour cela que celui-ci est un 2MΩ. L'information lue par l'Arduino sera la plus précise possible. Il suffit de placer notre potentiomètre à l'horizontal et grâce au flotteur, on peut déterminer une valeur pour laquelle le niveau d'eau est haut et une autre pour laquelle le niveau est bas. Dans les deux cas, on active un buzzer pour prévenir l'utilisateur.

D'autre part, on gère un moteur à courant continu de 9V (symbolisant la pompe) via le même système que pour les ventilateurs. C'est-à-dire, un transistor PNP en sortie d'Arduino qui permet de passer d'un état bloqué à un état passant le moteur. Cela nous permet de créer un goutte à goutte, en effet il s'agit, selon nos besoin, de décider à quel moment le moteur est alimenté et à quel moment il ne l'est pas en utilisant un délai sur le programme principal.

3.4. Programmation

Après avoir testé le programme Arduino correspondant à chaque montage, il est nécessaire de rassembler chaque code et de vérifier que chaque programme fonctionne en symbiose avec les autres. On utilise l'IDE disponible et qui plus est la version 1.0.3 puisque les précédentes ne prennent pas en compte la librairie nécessaire pour utiliser le DHT11. Pour le programme final, on définit toutes les variables dont nous avons besoin ainsi qu'une fonction d'initialisation.

```
#include "DHT.h"

#define DHTPIN 2      // pin du capteur
#define DHTTYPE DHT11
#define fan4 4 // pin du ventilateur d'humidité
#define fan7 7 // pin du ventilateur de température

int pinPot=0; // pin du potentiomètre

int const MOTEUR = 5 ; // pin du moteur (equivalent à la pompe)

int redPin = 9; // pin de la led rouge
int greenPin = 10; // pin de la led verte
int bluePin = 11; // in de la led bleue

int buzzer=12; // pin du buzzer

int maxHum = 28; // seuil d'humidité (en %)
int maxTemp = 23; // seuil de température (en °c)

int valPot=0; // variable pour récupérer la tension aux bornes du potentiomètre traduite par le CAN . On l'in

int potPin_red = A0; // pin de la résistance variable de la led rouge
int potPin_green = A1; // pin de la résistance variable de la led verte
int potPin_blue = A2; // pin de la résistance variable de la led bleue

int readValue_red; // variable de stockage de la valeur lue à la résistance variable qui gère la led rouge
int readValue_green; // variable de stockage de la valeur lue à la résistance variable qui gère la led verte
int readValue_blue; // variable de stockage de la valeur lue à la résistance variable qui gère la led bleue

int writeValue_red; // variable d'envoi vers la led rouge
int writeValue_green; // variable d'envoi vers la led verte
int writeValue_blue; // variable d'envoi vers la led bleue

DHT dht(DHTPIN, DHTTYPE);
```

Figure 5 : Déclaration des variables

```
void setup() {

  pinMode(potPin_red, INPUT); // déclare la résisatnce variable de la led rouge comme une entrée
  pinMode(potPin_green, INPUT); // déclare la résisatnce variable de la verte rouge comme une entrée
  pinMode(potPin_blue, INPUT); // déclare la résisatnce variable de la led bleue comme une entrée

  pinMode(redPin,OUTPUT); // déclare le pin de la led rouge comme une sortie
  pinMode(bluePin,OUTPUT); // déclare le pin de la led bleue comme une sortie
  pinMode(greenPin, OUTPUT); // déclare le pin de la led verte comme une sortie

  pinMode(MOTEUR, OUTPUT);

  pinMode(buzzer, OUTPUT);

  pinMode(fan4, OUTPUT);
  pinMode(fan7, OUTPUT);

  Serial.begin(9600); // initialisation de la communication avec la console

  dht.begin();
}
```

Figure 6 : Initialisation

Enfin le programme se déroule en boucle et effectue les actions suivantes :

- Il stocke dans des flottants la température et l'humidité reçues par le DHT11
- Il lit la valeur du potentiomètre
- Il lit les valeurs de résistances variables et permet ainsi d'écrire sur les pins qui alimentent les LEDs.
- Il existe une fonction qui permettrait de lire sur un port série la température, l'humidité ainsi que la valeur du potentiomètre.
- Enfin il effectue ses 4 fonctions principales comme suit :

```
/* Détection des niveaux et bas du réservoir */
  if (valPot < 20 || valPot > 1000)
{
  digitalWrite(buzzer, HIGH);
  delay(1);
  digitalWrite(buzzer, LOW);
  delay(1);
}
```

Figure7: Alerte des niveaux hauts et bas

```
/* Arrosage plante */

  digitalWrite(MOTEUR, HIGH);   // faire tourner le moteur
  delay(5000);                  // attendre 5 secondes
  digitalWrite(MOTEUR, LOW);    // arrêter le moteur
  delay(3000);                  // attendre 3 secondes
```

Figure 7: Contrôle de la pompe pour l'arrosage

```
/* Activation du ventilateur d'humidité */
if(h > maxHum) {
    digitalWrite(fan4, HIGH);
} else {
   digitalWrite(fan4, LOW);
}
```

Figure 8: Gestion de l'humidité

```
/* Activation du ventilateur de température */
if(t > maxTemp) {
    digitalWrite(fan7, HIGH);
} else {
   digitalWrite(fan7, LOW);
}
```

Figure10: Gestion de la température

4. Difficultés rencontrées

Durant le déroulement de ce projet, voici les difficultés qui ont pu être rencontrées :

- Les LEDs RGB n'auraient pas du être des CMS. Il a fallu utiliser des LEDs disponibles à l'école qui donc nécessitaient un PCB pour pouvoir être soudées. Pour contrer cela, il a fallu créer à partir d'une plaque possédant des pistes droites, des nouvelles pistes en découpant au cutter deux cotés distincts. De ce fait, on obtenait 4 pistes par LEDs qui pouvaient chacune être reliées à la masse ou à un couleur.

- La disponibilité des transistors souhaités étant limitée, pour l'utilisation du moteur de la pompe, il a fallu effectuer un montage de transistor. Celui souhaité et disposant du bon gain nécessaire n'étant plus disponible, il suffisait de créer un montage Darlington entre deux transistors. Le montage repose sur le principe de placer l'émetteur du premier PNP sur la base du deuxième afin de créer un transistor avec un plus grand gain.

- Durant une des premières semaines, le montage réalisé permettant de contrôler les ventilateurs ne fonctionnait plus. Il a fallu tester tous les composants qui marchaient pourtant correctement. Quelques jours plus tard après remontage, le système fonctionnait de nouveau mais cela a occasionné une perte de temps importante.

- La gestion du temps a été difficile du fait d'effectuer le travail seul. Le temps passé sur un possible PCB aurait en plus retardé l'avancement global du projet.

- La soudure de certains composants a été rendue compliquée (notamment pour les LEDs) du fait du manque de PCB mais s'avère tout de même fonctionner avec des brasures directes qui étaient forcément obligatoires pour des composants tels que le capteur qui se trouve dans l'enceinte du caisson.

- La phase de recherche des composants et des solutions à été trop longue et aurait due être raccourcie afin de laisser place à la réalisation.

5. Propositions d'améliorations

Le projet est arrivé à son terme en matière de fonctionnement autonome, la régulation se fait correctement ainsi que l'éclairage et l'arrosage. En revanche, certaines améliorations peuvent être apportées pour rendre le projet plus performant :

- Il serait intéressant et plus ergonomique de créer un PCB regroupant tous les composant nécessaires.

- Il n'a pas été possible d'aborder l'alimentation en énergie par panneau solaire, il faudrait l'insérer dans le système.

- Dans le futur, une deuxième pompe serait appréciable afin de remplir le bac réservoir. A l'heure actuelle, c'est l'utilisateur qui remplit le bac. Il serait également intéressant de penser à insérer un système de communication à distance avec l'utilisateur.

6. Conclusion

Au terme de ce projet, nous pouvons dire que certains objectifs ont été remplis mais que le sujet aurait pu être poussé plus loin encore. L'objectif de gérer chaque paramètre via un Arduino a pu être mené à terme. En revanche, la question de l'alimentation par panneau solaire n'a pu qu'être abordée. La gestion du temps a été mal entreprise mais ce qui est fonctionnel a été réalisé au mieux. Une autre personne aurait été souhaitable afin de mener à bien le projet.

Néanmoins, ce projet a permis de mettre en pratique des connaissances de la formation telles que la programmation de l'Arduino ou encore les montages sur les transistors. En outre, un projet de plus ample envergure tel que celui-ci, permet d'aborder une phase de recherche plus conséquente que d'ordinaire. Dans le cas présent, l'aspect de recherche sur les besoins des plantes qu'il a fallu adapter électroniquement a été intéressant. Peut-être aurait-il fallu un cahier des charges pour une plante spécifique avec des besoins précis afin de se concentrer directement sur la partie électronique.

Quoi qu'il en soit, ce projet permet d'effectuer un travail de plus envergure, de pouvoir effectuer une réalisation en autonomie et d'avoir des interactions avec les encadrants afin de discuter de solutions techniques pour aboutir le projet. Ce lui-ci a donc été formateur en terme de gestion et de réflexion.

7. Remerciements

Je tenais à remercier le département Informatique-Microélectronique-Informatique pour la diversité, la qualité et l'originalité des sujets proposés ainsi que l'équipe pédagogique.

Je remercie également le tuteur de ce projet, Mr Alexandre Boé, pour les entretiens accordés au projet ainsi que Mme Emmanuelle Pichonat pour les conseils et l'aide apportée.

Je remercie d'autre part Mr Thierry Flamen pour ses réponses et conseils vis-à-vis des solutions proposées durant le projet.

Je remercie enfin Mr Laurent Engels pour la réalisation de la vidéo de présentation et pour le montage de celle-ci