Rapport du projet Smartinage

Projet réalisé par des élèves volontaires de 4ème, 5ème et 6ème secondaire de « de l'Autre côté de l'Ecole », école à pédagogie Freinet. Encadré par leurs professeurs de physique et mathématiques : Jérôme Carette et Christophe Moreels.

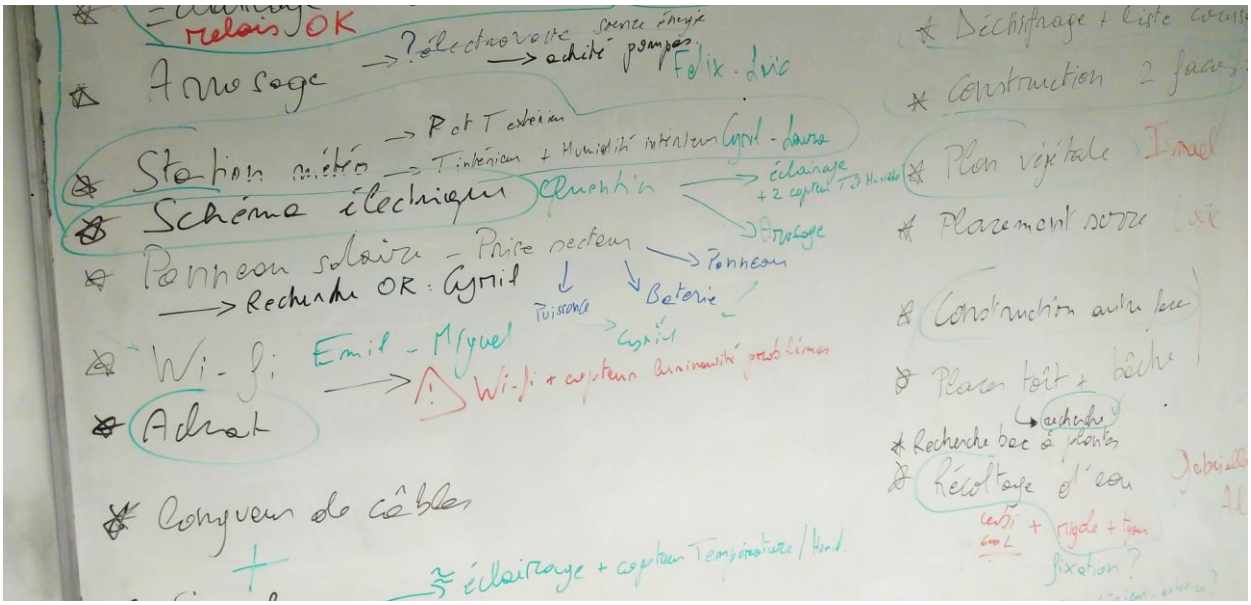
1. Organisation du projet

1.1. Réunions du mardi midi

Les réunions ont lieu les mardis midi entre 13h et 14h au laboratoire de physique ou en salle informatique.

1.2. Journées Robotique

Des rencontres auront lieu ponctuellement durant des journées entières (journées pédagogiques par exemple). Deux de ces journées ont été encadrées par des étudiants de l'ULB mis à disposition par l'expérimentarium de physique. Ces journées ont été l'occasion de donner des gros coups d'accélérateur dans l'avancement du projet.



1.3. Préambule

N'ayant pu travailler qu'une semaine sur deux, nous n'avons pas pu aller jusqu'au bout du projet. Finalement, la serre est totalement opérationnelle (des tomates ont poussé cette année qui fut pourtant très compliquée pour tous les légumes). Le système de récolte des eaux de pluies et d'arrosage est en place. Tout le matériel est acheté. Tous les capteurs sont déjà soudés aux fils correspondants, mais pas encore reliés à l'Arduino. Tous les codes indépendants sont fonctionnels, mais pas encore le code général permettant à toutes les fonctionnalités en même temps. Et finalement, nous n'avons pas encore pu installer le système d'automatisation dans la serre en fin d'année. Cela sera donc effectué au cours de cette année-ci.

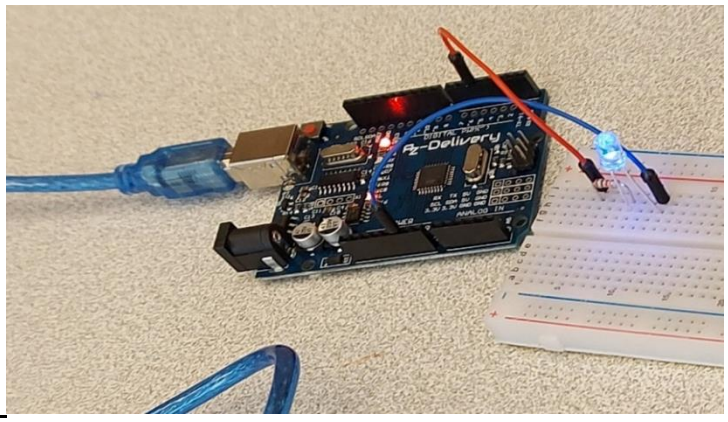
2. Construction de la station météo

2.1. Soudures *(Florian)*

Soudure initiales des différents capteurs afin de pouvoir opérer des branchements et des tests.

2.2. Première rencontre avec Arduino *(Loïc F)*

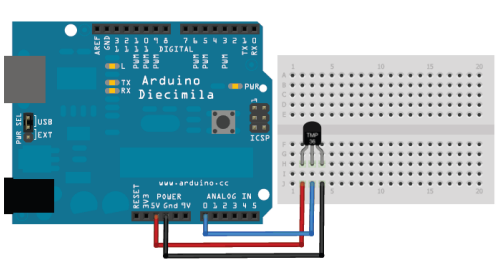
Objectif : allumer un simple led bleu et le faire clignoter

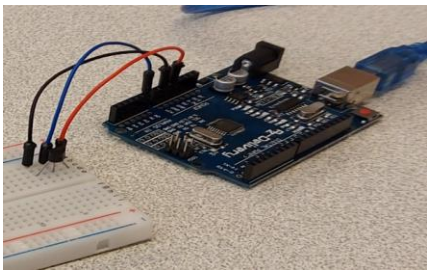



```
void setup() {
  pinMode(5, OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH);
  delay(500);
  digitalWrite(5, LOW);
  delay(500);
}
```

2.3. Capteur de température *(Loic F)*

Le capteur TMP36 nous donne la température (très précis).

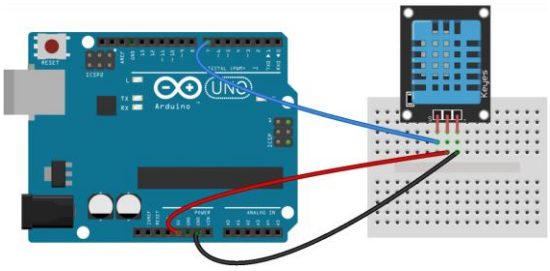


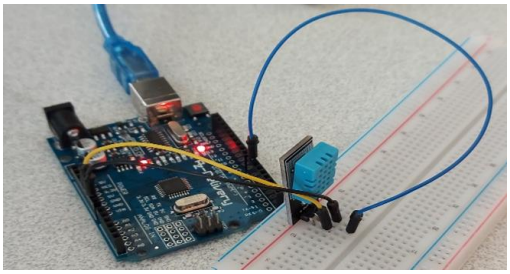


```
int sensorPin = 0;
void setup()
{
 Serial.begin(9600);
}

void loop()
{
 int reading = analogRead(sensorPin);
 float voltage = reading * 5.0;
 voltage /= 1024.0;
 Serial.print(voltage); Serial.println(" volts");
 float temperatureC = (voltage - 0.5) * 100 ;
 Serial.print(temperatureC); Serial.println(" degrees C");
 delay(5000);
}
```

2.4. Capteur d'humidité et température *(Loic F)*

Le capteur nous donne l'humidité dans l'air (précis) et la température (moins précis que l'autre capteur de température TMP36). Attention à ne pas oublier la librairie dans le code.





```
#include <dht.h>

dht DHT;

#define DHT11_PIN 7 // A remplacer par le pin de votre choix

void setup(){
  Serial.begin(9600);
}
void loop(){
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);
  delay(2000);
}
```

2.5. Capteur d'humidité du sol *(Léa, Apolline, Gabrielle)*

Voir schémas de montage et code de la section 3.3. L'idée générale est de mesurer l'humidité dans le sol à l'aide de capteurs résistifs. Lorsque la résistance électrique du sol est supérieure à un certain seuil calibré à l'avance (en fonction du type de sol par exemple), alors l'arrosage se met en route.

2.6. Capteur de luminosité *(Miguel, Laura, Emil)*

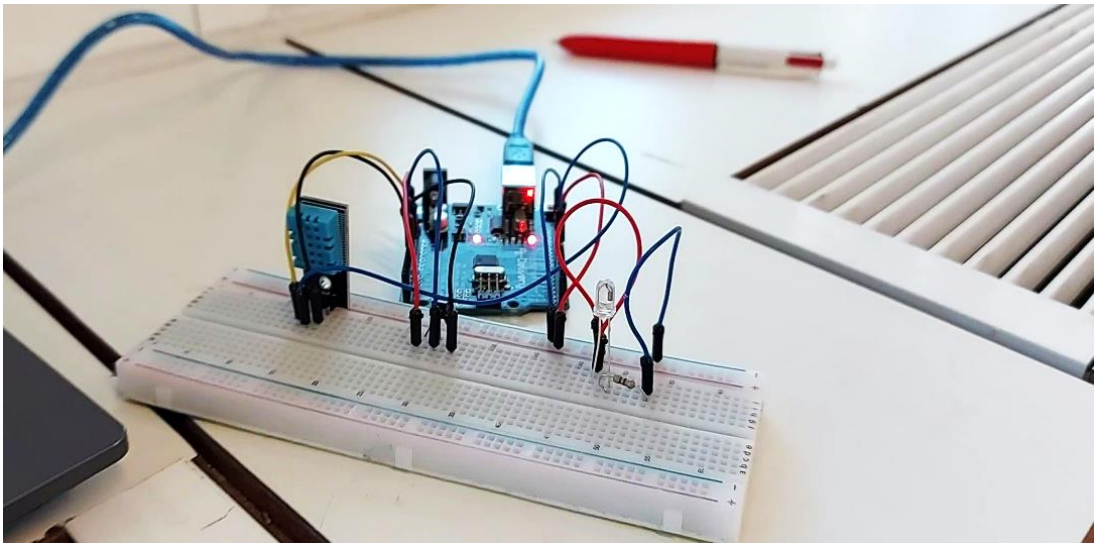
Voir schémas de montage et code de la section 4.1. Il s'agit d'un capteur GY-302. Tout comme pour l'humidité du sol, l'idée est ici d'allumer automatiquement des panneaux LED (achetés tels quels) lorsque la luminosité n'est pas suffisante. On s'assurera que l'allumage ne soit possible que pendant les heures de la journée.

2.7. Capteur de pression *(Quentin)*

Le capteur de pression n'a pas d'autre utilité à ce stade que pour obtenir l'information en temps réel de la pression atmosphérique. A terme, on pourrait réfléchir à voir ce que cela peut laisser prédire en terme de précipitations à venir.

2.8. Mise en commun des capteurs *(Loïc F, Quentin)*

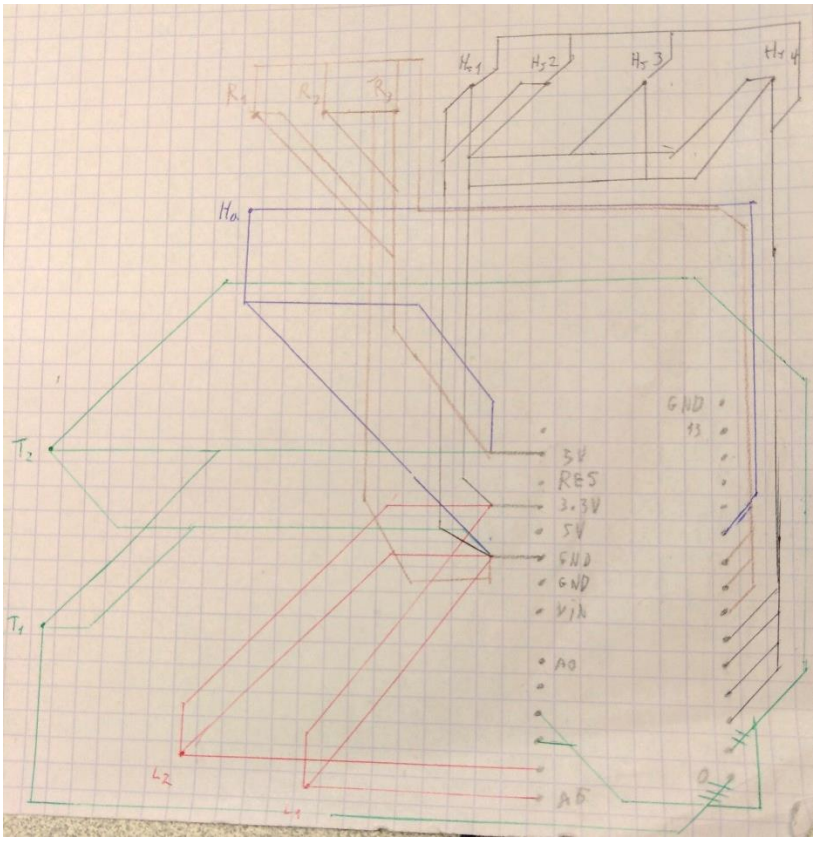
Au tout début, la mise en commun portait sur les capteurs de température et d'humidité. Une lampe LED s'allume à chaque fois qu'on reçoit une donnée.



```
#include <dht.h>

dht DHT;
int sensorTEMPPin = A0;

#define DHT11_PIN 7 // A remplacer par le pin de votre choix
#define LEDPin 10 // A remplacer par le pin de votre choix

void setup(){
  Serial.begin(9600);
  pinMode(LEDPin, OUTPUT);
}

void loop(){
    digitalWrite(LEDPin, HIGH);

  // ICI LE CODE DU CAPTEUR D'HUMIDITE (+T)
  int chk = DHT.read11(DHT11_PIN);
```

```
 Serial.print("Temperature = ");
 Serial.println(DHT.temperature);
 Serial.print("Humidity = ");
 Serial.println(DHT.humidity);

 // ICI LE CODE DU CAPTEUR TMP36 (TEMPERATURE)
 int reading = analogRead(sensorTEMPPin);

float voltage = reading * 5.0;
voltage /= 1024.0;

Serial.print(voltage); Serial.println(" volts");

float temperatureC = (voltage - 0.5) * 100 ;
Serial.print(temperatureC); Serial.println(" degrees C");

   digitalWrite(LEDPin, LOW);
  delay(5000);
}
```

Ensuite, lorsque les différents capteurs ont tous fonctionnés indépendamment, il a fallu réaliser un schéma de câblage avant de procéder à la soudure. Concrètement, il faut savoir combien de capteurs d'humidité et de luminosité sont nécessaires, la longueur des fils à prévoir, ainsi que la disponibilité suffisante de ports sur la carte arduino.



2.9. Dimensionnement de la batterie

Le projet d'alimenter le projet à l'aide d'une batterie est rapidement abandonné au profit d'un système d'alimentation électrique apporté directement à la serre depuis le réseau électrique de l'école.

2.10. Conception et réalisation d'un support étanche *(Romain)*

Un boitier étanche a été modélisé en 3D. Il contient des espaces pour le branchement de la carte Arduino, de quelques capteurs ainsi que de la batterie, et permet l'évacuation de cables. Il contient également un capuchon vissable avec le logo suivant en relief. Le tout est imprimé en 3D.

3. Réalisation de l'arrosage automatique

3.1. Conception d'un système de récupération de l'eau de pluie

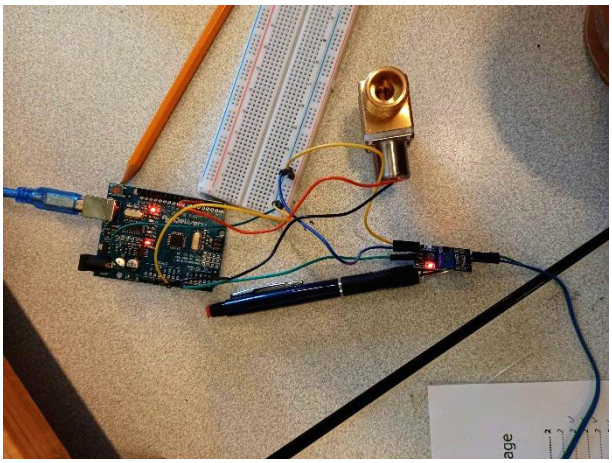
Une cuve de récupération de l'eau de pluie est prévue. Elle permet la récolte d'une grande quantité d'eau pendant les périodes pluvieuses pour éventuellement couvrir les sécheresses à venir. Les calculs basés sur les statistiques récentes indiquent, pour les dimensions de la serre prévue, que 600L sont plus que nécessaires pour subvenir à nos besoins, dans le sens que nous ne devrions pas récolter davantage étant donné la faible surface du toit. Néanmoins, 1000L permettent également de profiter des plus grandes périodes de pluviosité auxquelles l'on pourrait s'attendre dans le futur.

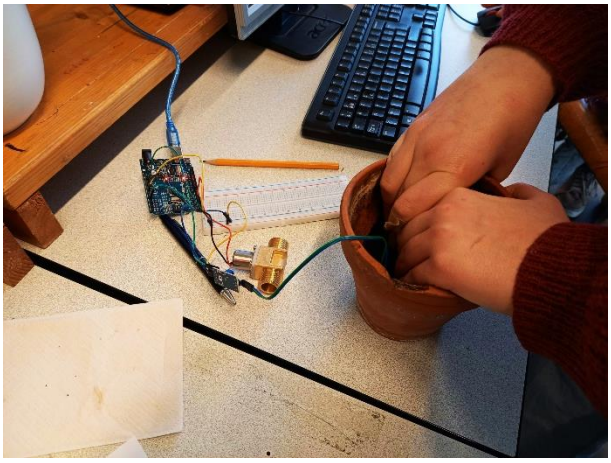
3.2. Achat du matériel

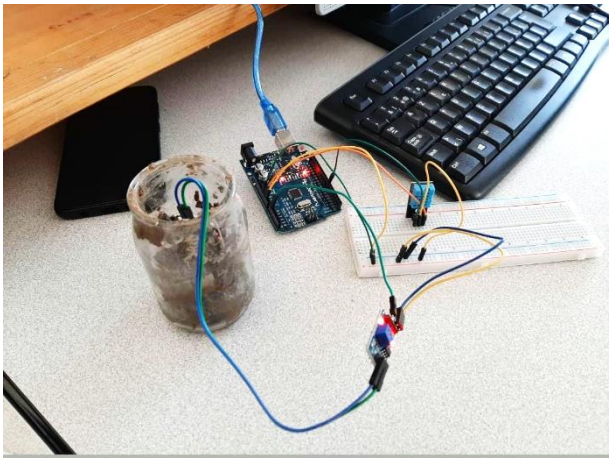
La cuve de 1000L est achetée chez ecocuves.be. Les pompes à débit moyen, tuyaux d'arrosage, rigoles et toitures sont achetés chez Brico (disponible à proximité de l'école).

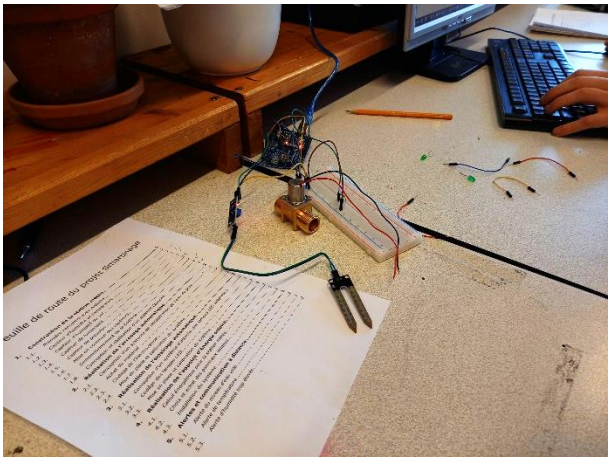
3.3. Codage du système d'arrosage automatique *(Léa, Apolline, Gabrielle)*

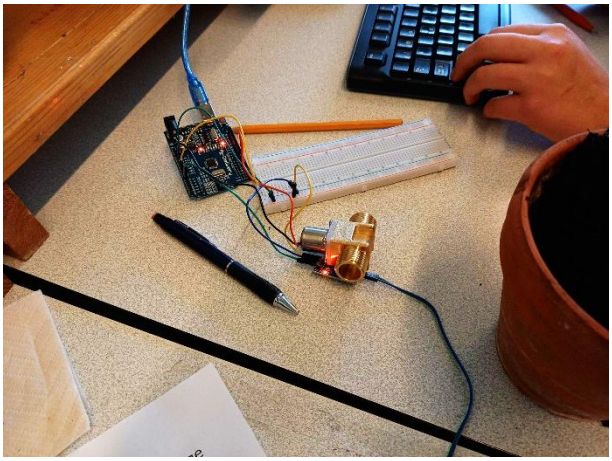
On a compris ce qu'était le codage avec la lecture du code du capteur de l'humidité du sol et on a appris à superposer 2 codes. Ensuite, on a fait des expériences pour savoir quand l'humidité du sol était parfaite pour une plante « normale », ce qui correspondait à une résistivité affichée de 600. Sans le code, l'électrovanne ne s'active donc que pour des résistivités inférieures à 600.





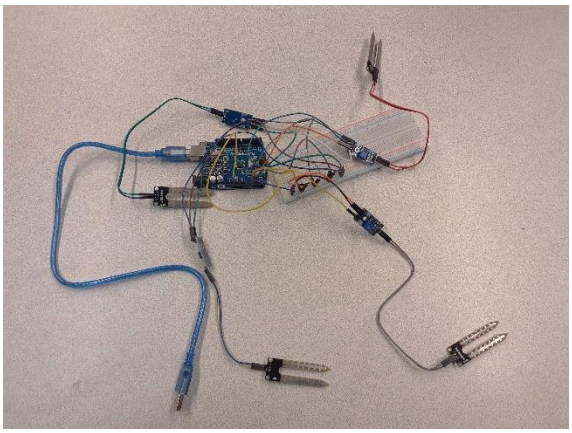


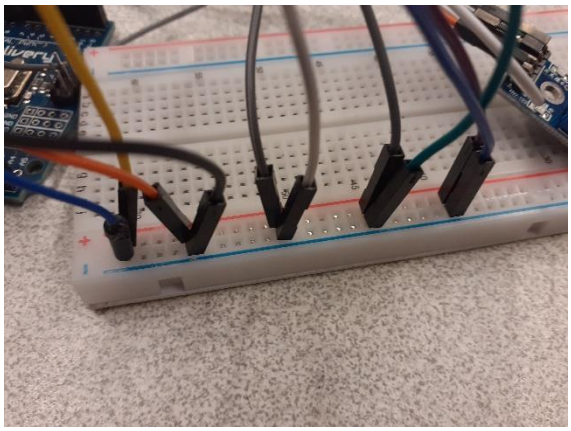




```
int sensorPin = A0;
int sensorValue;
int elec_vanne = 4;

void setup() {
 Serial.begin(9600);
}

void loop() {

 sensorValue = analogRead(sensorPin);
 Serial.println("Analog Value : ");
 Serial.println(sensorValue);

 delay(1000);
 if(sensorValue>600){
  digitalWrite(elec_vanne, HIGH);}
 if(sensorValue<600){
  digitalWrite(elec_vanne, LOW);}
}
```

Initialement, nous avons choisi d'installer 4 capteurs d'humidité du sol : 2 capteurs d'humidité dans chacun de 2 bacs indépendants. Pour des raisons de temps et de budget, nous avons finalement décidé de n'installer qu'une seule pompe, et donc nous n'avons besoin que de 2 capteurs, ce qui est suffisant pour effectuer une moyenne globale.





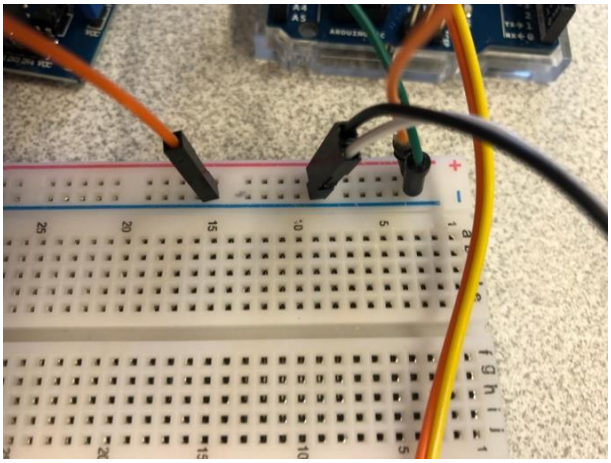
3.4. Mise en place et calibration du système

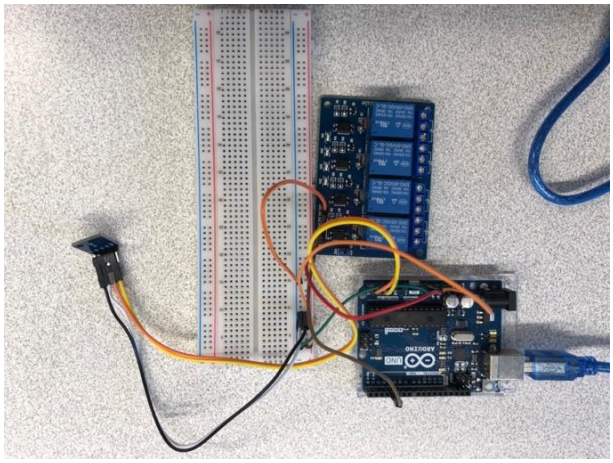
Pour des raisons de temps, l'installation (et donc la calibration du système) n'a pas pu avoir lieu. Néanmoins, la cuve à eau, les rigoles, les tuyaux et la pompe sont opérationnels. Ils nécessitent à ce stade une activation manuelle. La partie robotisation est également prête, mais nécessite à ce stade une installation sur le site.

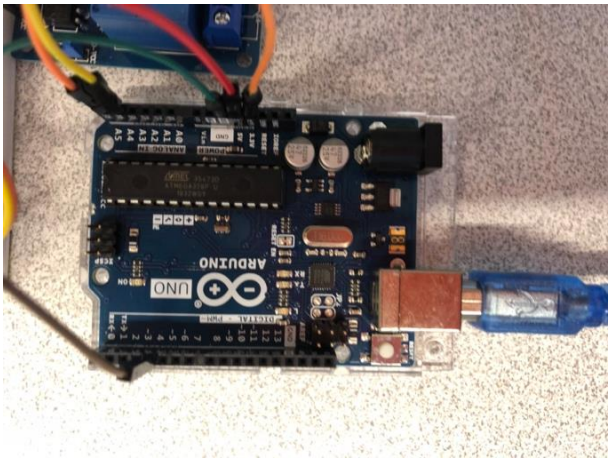
4. Réalisation de l'éclairage automatique

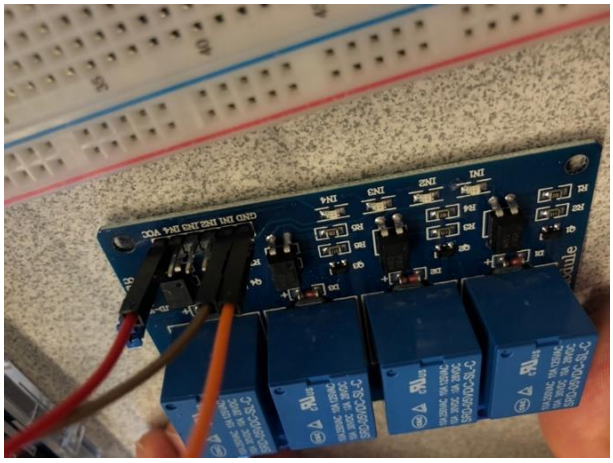
4.1. Codage du système d'éclairage LED *(Miguel, Laura, Emil)*

L'éclairage automatique est programmé pour détecter une quantité de lumière (LUX). Quand le capteur capte une lumière, il l'écrit sur le moniteur série. La lumière est interprétée de manière à ce que quand le niveau de lumière est inférieur à 800 lux, on allume les LED, et s'il est supérieur à 1000, on les éteint. Le système ne fonctionne que de 08H00 à 20H00, hors de ces plages horaires, les LED sont éteintes. Pour définir la date et l'heure : https://www.epochconverter.com/ (mettre un « T » devant le nombre décadécimal)









```
/* Branchements :
 *  BH1750 VCC -> 3.3V; GND -> GND; SCL -> A5; SDA -> A4
 *  Relai VCC -> 5V; IN1 -> D4; GND -> GND
 */
#include <Wire.h>
#include <BH1750.h>
#include <TimeLib.h>

#define TIME_HEADER  "T"   // Header tag for serial time sync message
#define TIME_REQUEST  7    // ASCII bell character requests a time sync message
```

```
BH1750 lightMeter;

int Allummer = 800;
int Eteindre = 1000;
bool Statut = false;
int execution = 4;


void setup(){

  Serial.begin(9600);
  Wire.begin();

  lightMeter.begin(BH1750::ONE_TIME_HIGH_RES_MODE);

  Serial.println(F("BH1750 One-Time Test"));

// TEMPS
  Serial.begin(9600);
  while (!Serial) ; // Needed for Leonardo only
  pinMode(13, OUTPUT);
  setSyncProvider( requestSync);  //set function to call when sync required
  Serial.println("Waiting for sync message");

}
void ChangementDeStatut() {

   int lux = lightMeter.readLightLevel();
 int h = hour();
if(h >= 8 && h <= 19) {

  if(Statut == false && lux <= Allummer) {

      Statut = true;
     Serial.println("Allummer");
     digitalWrite(execution, HIGH);
    } else if (Statut == true && lux >= Eteindre) {

     Statut = false;
     Serial.println("Eteindre");
     digitalWrite(execution, LOW);
    }else {
      Serial.println("rien faire");
    }
}else if( Statut == true) {
  Statut = false;
     Serial.println("Eteindre");
     digitalWrite(execution, LOW);
}

}

void loop() {

  while (!lightMeter.measurementReady(true)) {
    yield();
  }
  if (Serial.available()) {
    processSyncMessage();
  }
//Affichage sur le port série
 int lux2 = lightMeter.readLightLevel();
  Serial.print("Lumiere: ");
  Serial.print(lux2);
  Serial.println(" lx");
  Serial.print("Statut = ");
  Serial.println(Statut);
  ChangementDeStatut();
  digitalClockDisplay();
  Serial.println("_____________");
```

```
  Serial.println();


  lightMeter.configure(BH1750::ONE_TIME_HIGH_RES_MODE);
  delay(1000);
}
void digitalClockDisplay(){
  //Ecrire l'heure
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.print(" ");
  Serial.print(day());
  Serial.print(" ");
  Serial.print(month());
  Serial.print(" ");
  Serial.print(year());
  Serial.println();
}


void printDigits(int digits){
  // Extraction de l'heure (du 10-digits)
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}


void processSyncMessage() {
 //Attendre la réception de la config
  unsigned long pctime;
  const unsigned long DEFAULT_TIME = 1357041600; // Jan 1 2013

  if(Serial.find(TIME_HEADER)) {
     pctime = Serial.parseInt();
     if( pctime >= DEFAULT_TIME) { // check the integer is a valid time (greater than Jan 1 2013)
       setTime(pctime); // Sync Arduino clock to the time received on the serial port
     }
  }
}

time_t requestSync()
{
  Serial.write(TIME_REQUEST);
  return 0; // the time will be sent later in response to serial mesg
}
```

4.2. Relais *(Miguel, Laura, Emil, Florian)*

Après avoir compris le fonctionnement des relais, nous avons réfléchi au nombre de relais nécessaires. Il en faut un pour l'alimentation des LED, et un pour l'alimentation de la pompe. Nous avions initialement acheté des groupes de 4 relais, mais utiliserons finalement plutôt deux relais simples indépendants.

5. Réalisation de l'appoint d'énergie solaire

Cette partie du projet est annulée pour des raisons de temps disponible, mais n'est pas totalement rejetée pour des projets futurs.

6. Alertes et communication à distance

En théorie, tout est mis en place afin de pouvoir, via le wifi de l'école et une carte arduino munie d'une carte wi-fi, pour activer ou désactiver tout capteur à distance. Nous avons testé cela, et avons réussi à activer les LED à distance via nos smartphones. Il reste encore à mettre en place un système d'alerte dès qu'un niveau d'eau trop bas de la cuve advient, ou tout autre problème nécessitant une intervention sur le site.

7. Construction de la serre

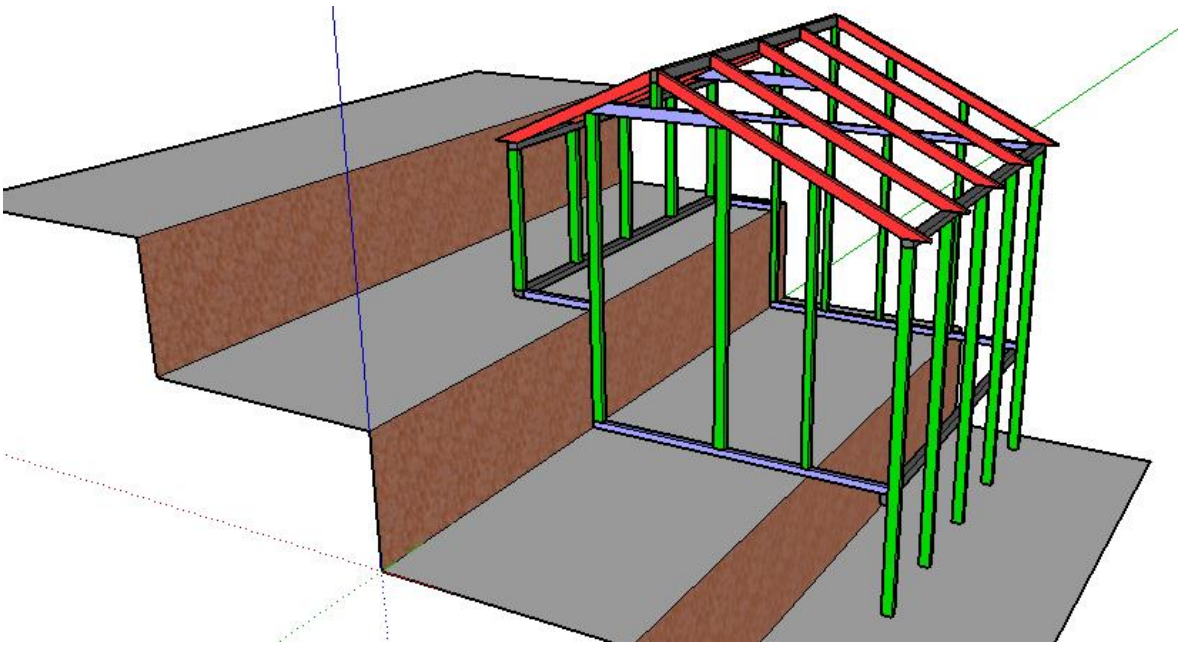
7.1. Conception

Le terrain sur lequel nous sommes amenés à réaliser la serre est sous forme de longues terrasses d'1,5m de large. Ces terrasses sont constituées de terre, et les murs de soutènement sont réalisés en bois. Nous allons donc tirer profit de ce terrain pour :

- réaliser des bacs à plantes naturels (en pleine terre) sur le flanc « gauche » de la serre ;
- réaliser un plancher pour positionner des bacs sur le flanc « droit » de la serre. Cela permet un abri sous le plancher pour stocker du matériel à l'abri de la pluie ;
- fixer la serre aux murs de soutènement existants afin de renforcer la stabilité des talus et de la serre.

De cette manière, les dimensions de la serre permettent d'avoir une bonne surface pour la récupération des eaux de pluie ainsi qu'une surface suffisante pour planter.

La concetion est réaliser sur Sketchup sur base de différentes sources trouvées sur internet ainsi que les connaissances de dimensionnement d'un professeur de l'école ingénieur civil architecte.



7.2. Achat des matériaux

Tous les matériaux pour réaliser la serre (structure, toiture, baches, vis) sont achetés chez Brico. Les outils sont fournis par l'école ou empruntés aux professeurs ou élèves. Les planchers ainsi que quelques renforts métalliques sont fournis par des restes de matériel appartenant à des anciens chantiers de l'école.

7.3. Réalisation du chantier