

## **PPE : Concevoir une ruche connectée**

---

### **Introduction :**

Tout d'abord, pour pouvoir créer notre ruche connectée, il faut définir la notion de "connectée". On cherche ici à créer un système capable d'être informé de l'état de la ruche en temps réel grâce à une communication bluetooth via téléphone à celle-ci dans laquelle seront placés divers capteurs permettant de connaître son bon fonctionnement. Dans ce projet, je me suis occupé de toute la programmation arduino des divers capteurs de températures, d'humidité ainsi que de l'électrovanne.

Pour ce projet, nous avons choisi un arduino pour utiliser les composants car c'était le plus simple, le plus basique et le plus utilisé par rapport au raspberry pi et il nous permettait de nous aider grâce aux divers forums en cas de problème majeur.

Pour mesurer la température intérieure j'ai utilisé un capteur de température basique KY-001.

Pour mesurer la température ainsi que l'humidité à l'extérieur, j'ai choisi un capteur qui peut mesurer la température et la teneur en eau de l'air à la fois : le capteur DHT-11. Le capteur DHT11 est lui capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes.

Pour distribuer du sirop à un certain intervalle de temps j'ai utilisé une électrovanne à commande directe alimentée par un générateur 12Volts avec un relais ce qui me permet de contrôler le flux de sirop débouchant par intermittence.

---

## **PRINCIPES DE FONCTIONNEMENTS**

## **CAPTEUR D'HUMIDITÉ GROVE**

Ce module capteur d'humidité compatible Grove permet de connaître la concentration d'eau dans la terre par exemple.

Le capteur délivre une valeur analogique en fonction de la teneur en eau grâce au principe de la conductivité.

Ce capteur mesure l'humidité du sol à partir des changements de conductivité électrique de la terre

(la résistance du sol augmente avec la sécheresse).

Une sortie digitale avec un seuil réglable par potentiomètre permet de déclencher une pompe d'arrosage ou une alarme par exemple.

Une seconde sortie analogique permet de suivre les fluctuations précises de l'humidité du sol.

---

## **RELAIS**

Un relais est un circuit électrique permettant de contrôler le flot de courant, comme un interrupteur pour les lumières.

La différence est qu'un interrupteur traditionnel fonctionne de façon mécanique, soit en l'activant physiquement avec votre doigt, alors qu'un relais fonctionne avec un courant électrique.

À l'intérieur, le processus reste mécanique. Le circuit de contrôle active un électroaimant qui tire une plaquette de métal qui active la connexion du circuit à contrôler. On peut d'ailleurs l'entendre très bien lorsque le relais s'active. L'avantage d'un relais est aussi qu'il isole complètement les deux circuits, permettant ainsi d'activer un circuit dans lequel circule un courant plus fort.

---

## **CAPTEUR DHT11 (humidité+température à l'extérieur)**

Le capteur numérique de température et d'humidité DHT11 est un capteur contenant une sortie numérique de la température et de l'humidité.

Application d'un module numérique dédié et la technologie de détection de la

température et de l'humidité, afin de garantir que le produit a une grande fiabilité et une excellente stabilité à long terme.

Le capteur comprend un capteur résistif de composants humides et un dispositif de mesure de la température.

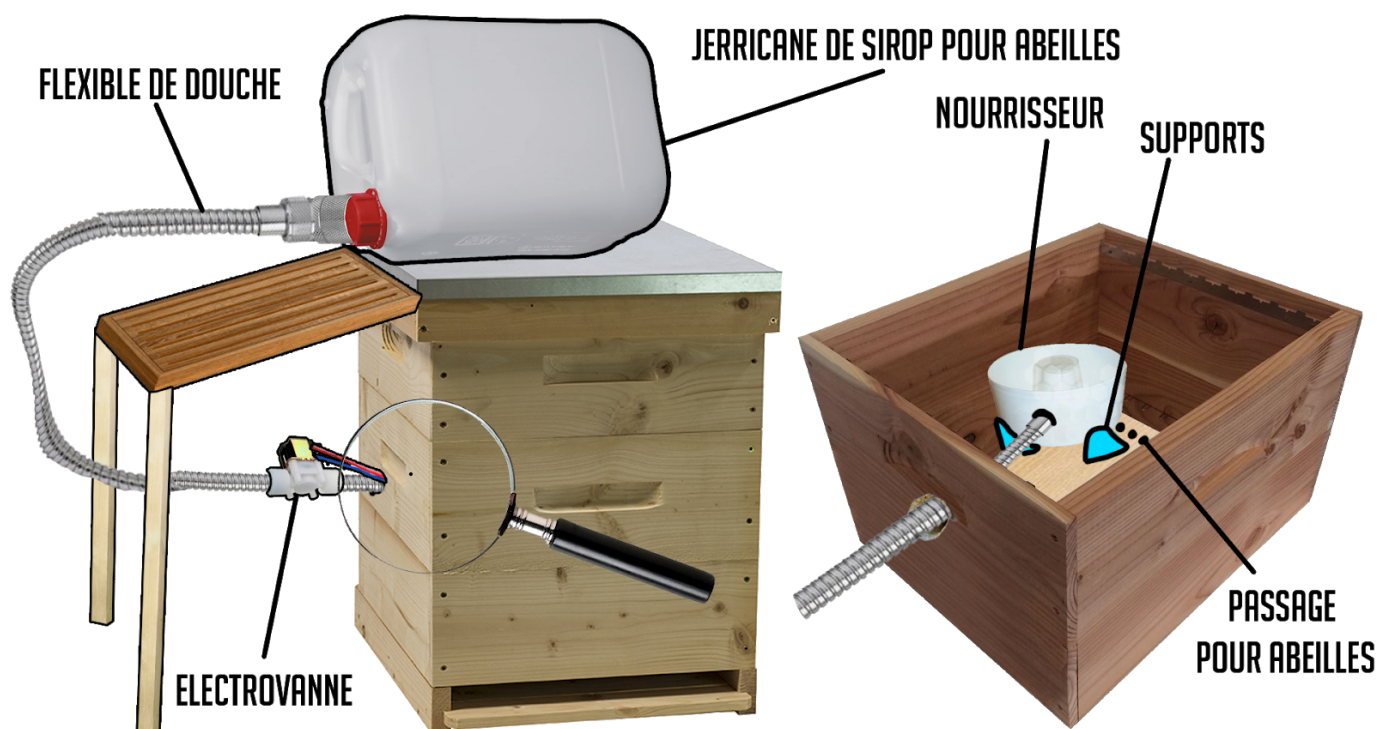
---

### **CAPTEUR KY001 (température à l'intérieur)**

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de +10mV.

---

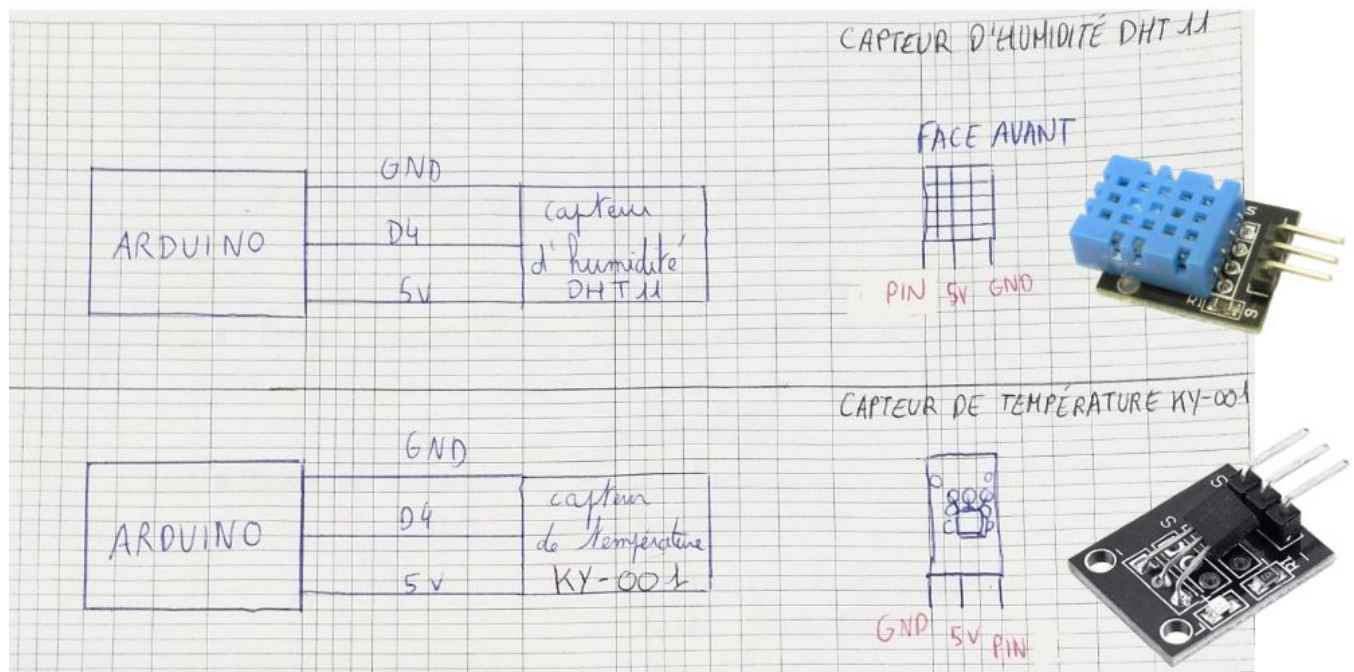
Voici une réalisation en perspective de ce qu'on aurait pu obtenir comme projet final si tout se serait passé correctement.



---

Voici tous les schémas de montage des divers composants que j'ai utilisé et programmé à l'aide de l'arduino

Tout d'abord les capteurs d'humidité/température et de température qui vont servir à mesurer cette donnée afin de la stocker et de l'envoyer en bluetooth au téléphone portable.



Voici mon programme arduino pour le capteur de température KY-001 :

```

2. #include <DallasTemperature.h>
3. #include <OneWire.h>
4.
5.                                     // déclaration de l'entrée auquel est relié le capteur de
    température
6. #define KY001_Signal_PIN 4
7.
8.                                     // configuration des bibliothèques pour notre utilisation
9. OneWire oneWire(KY001_Signal_PIN);
10. DallasTemperature sensors(&oneWire);
11.
12.
13. void setup() {
14.
15.                                     // initialisation de la sortie
16.     Serial.begin(9600);
17.     Serial.println("KY-001 temperature measurement");
18.
19.                                     // initialisation du capteur
20.     sensors.begin();
21. }
22.
23.                                     // boucle du programme principal
24. void loop()
25. {
26.                                     // la mesure de la température va commencer...
27.     sensors.requestTemperatures();
28.
29.                                     // ... et la température mesurée va être affichée
30.     Serial.print("Temperature: ");
31.     Serial.print(sensors.getTempCByIndex(0));
32.     Serial.println(" degrés mesurés");
33.
34.     delay(1000);                                     // 1 seconde d'intervalle entre les mesures
35. }

```

Voici mon programme arduino pour le capteur de température et d'humidité à la fois le DHT-11 :

```
1. #include "DHT.h"      // Librairie des capteurs DHT
2. #define DHTPIN 2      // Changer le pin sur lequel est branché le DHT
3. #define DHTTYPE DHT11
4. DHT dht(DHTPIN, DHTTYPE);
5.
6. void setup() {
7.   Serial.begin(9600);
8.   Serial.println("Test du composant DHT11 !");
9.
10.  dht.begin();
11. }
12. void loop() {
13.   // Délai de 2 secondes entre chaque mesure. La lecture prend 250 millisecondes
14.   delay(2000);
15.   // Lecture du taux d'humidité
16.   float h = dht.readHumidity();
17.   // Lecture de La température en Celcius
18.   float t = dht.readTemperature();
19.   // Pour Lire La température en Fahrenheit
20.   float f = dht.readTemperature(true);
21.
22.   // Stop Le programme et renvoi un message d'erreur si Le capteur ne renvoi aucune mesure
23.   if (isnan(h) || isnan(t) || isnan(f)) {
24.     Serial.println("Echec de lecture !");
25.     return;
26.   }
27.
28.   Serial.print("Humidite: ");
29.   Serial.print(h);
30.   Serial.print(" %\t");
31.   Serial.print("Temperature: ");
32.   Serial.print(t);
33.   Serial.println(" *C ");
34. }
```

Une fois ces capteurs utilisés, je passe à la distribution de la nourriture avec mon électrovanne, mon relais ainsi que l'alimentation 12Volts qui va nous servir à ouvrir ou fermer l'électrovanne par intermittence et ainsi laisser passer le sirop nécessaire à l'alimentation des abeilles dans la ruche.

Voici le programme de l'électrovanne par intermittence où l'on change le délai afin de laisser passer plus ou moins de sirop :



```

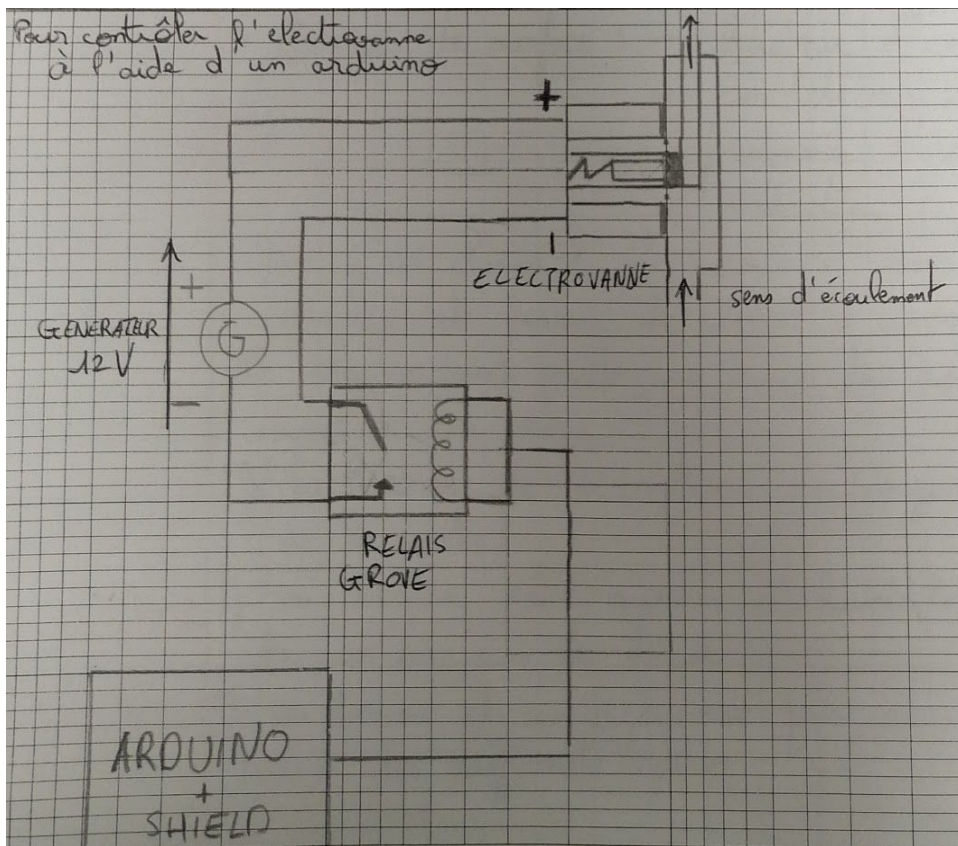
const int pinOut = 4;

void setup() {
  pinMode(pinOut, OUTPUT);
}

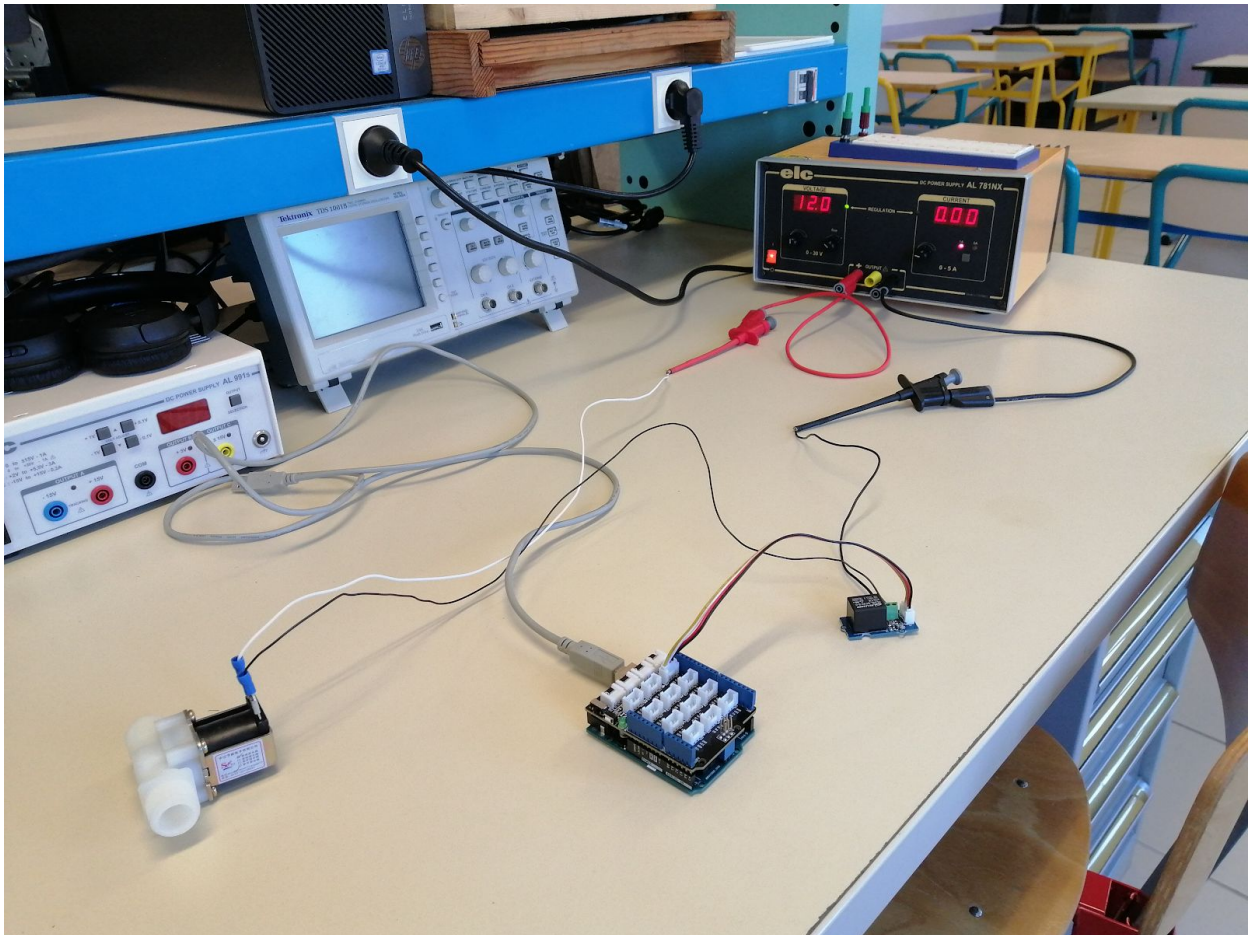
void loop() {
  Serial.begin(9600);
  digitalWrite(pinOut, 1);
  Serial.println("électrovanne ouverte!");
  delay(1500);
  digitalWrite(pinOut, 0);
  Serial.println("électrovanne fermée!");
  delay(1500);
}

```

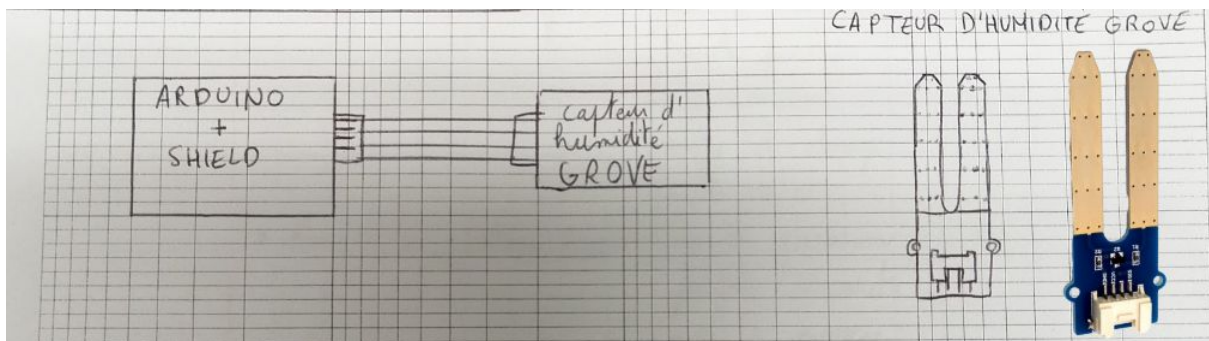
Le montage de l'électrovanne est schématisé ci-dessous:



Et le voici en condition réelles :



J'ai aussi programmé l'alerte en cas de réserve de sirop faible, pour cela, je place un capteur d'humidité grove au fond du réservoir et quand un seuil d'humidité descend, on alerte à l'apiculteur sur l'application que le réservoir est faible.



Voici ce que j'ai programmé en arduino afin d'alerter l'apiculteur du niveau de sirop restant :

```
1. int sensorPin = A0; //on definit La constante et nos variables
2. int sensorValue = 0;
3. int sensorValue2 = 0;
4.
5. void setup() {
6.     Serial.begin(9600); //on definit La transmission afin de lire Les données à 9600 bits/seconde
7. }
8. void loop() {
9.     sensorValue = analogRead(sensorPin); //on definit La variable sur le pin A0
10.    sensorValue2 = map(sensorValue,0,1024,0,100); //on convertit notre analogique qui va de 0 à 1024 en pourcentage c'est-à-dire
    de 0 à 100
11.    Serial.print("Humidité = " ); //on écrit La donnée qui est mesurée
12.    Serial.print(sensorValue2); //on affiche La mesure en pourcentage de l'humidité
13.    Serial.print(" ");
14.    Serial.println("%"); //on écrit l'unité de La donnée mesurée
15.    delay(1000); //on relance La mesure chaque seconde
16.
17.    if(sensorValue2 == 0){ //si l'humidité est de 0, cela signifie qu'il n'y a plus de sirop et donc on
    avertis l'utilisateur de ce manque
18.        Serial.println("On avertis l'utilisateur!");
19.        delay(1500);
20.    }
21. }
```

---