

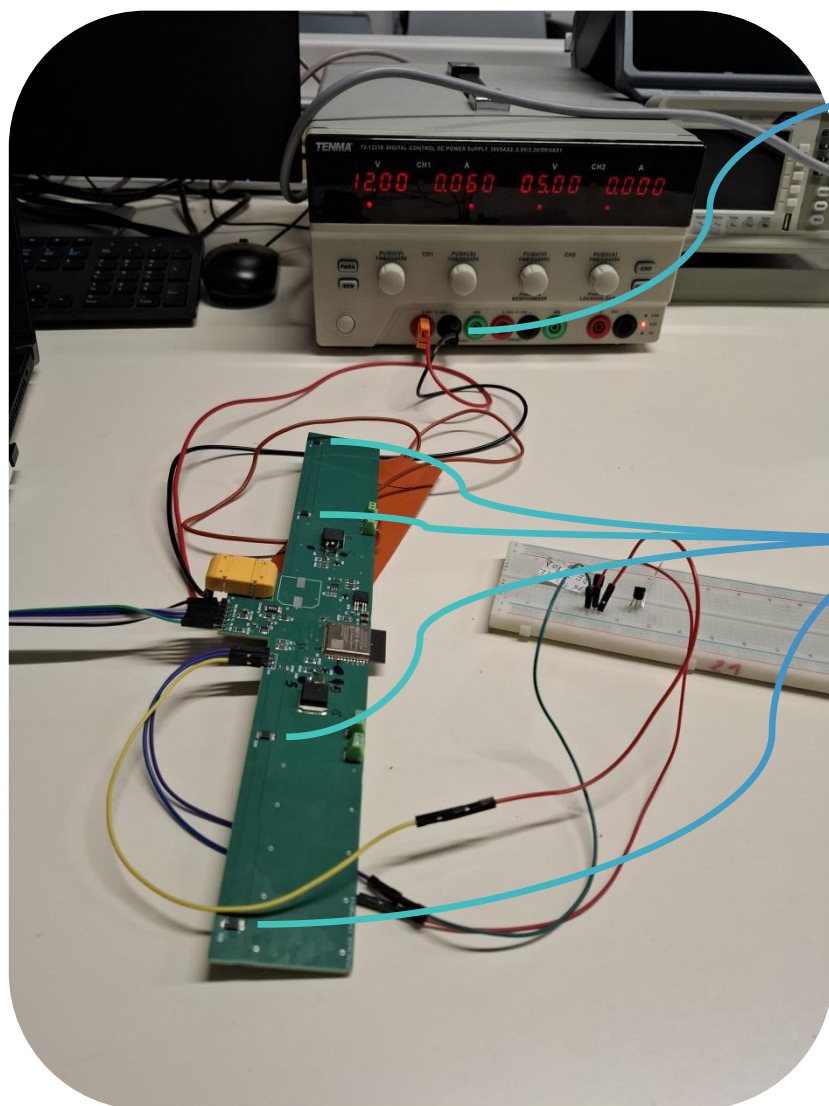
RAPPORT DE SÉANCE 4 :



Durant la 4^{ème} séance nous avons continué le **code Arduino**. Dans un premier temps, nous avons **récupéré les températures de la ruche** et de la **végétaline** provenant des **4 capteurs de température** sur la **PCB (végétaline)** et de **un seul capteur pour la ruche** afin de pouvoir ensuite **tester l'envoi** et **l'affichage** de celles-ci sur l'**application**.

Jusqu'à la **séance précédente**, les **températures** de la **ruche** et de la **végétaline** étaient simplement **générées aléatoirement** afin de pouvoir tester la communication **Bluetooth** entre l'**ESP32** et l'**application**.

Dans cette séance, nous avons donc **remplacé** ces **valeurs simulées** par les **mesures réelles** des **capteurs DS18B20** présents sur la **PCB**. Nous avons ainsi mis en place la lecture des capteurs. Voici ci-dessous, le **câblage effectué** pour le test :



La **PCB** est alimentée en **12V**, par la suite elle sera **alimentée** par le **panneau solaire**.

Les **quatre capteurs de température** dédiés à la **mesure de la végétaline** sont câblés sur un **bus OneWire**, ce qui signifie qu'ils **partagent tous la même ligne de données**. Ils sont ainsi tous reliés à l'**ESP32** via un **seul fil** connecté à la broche **GPIO 14**, en plus de l'**alimentation 3,3 V** et de la **masse**.



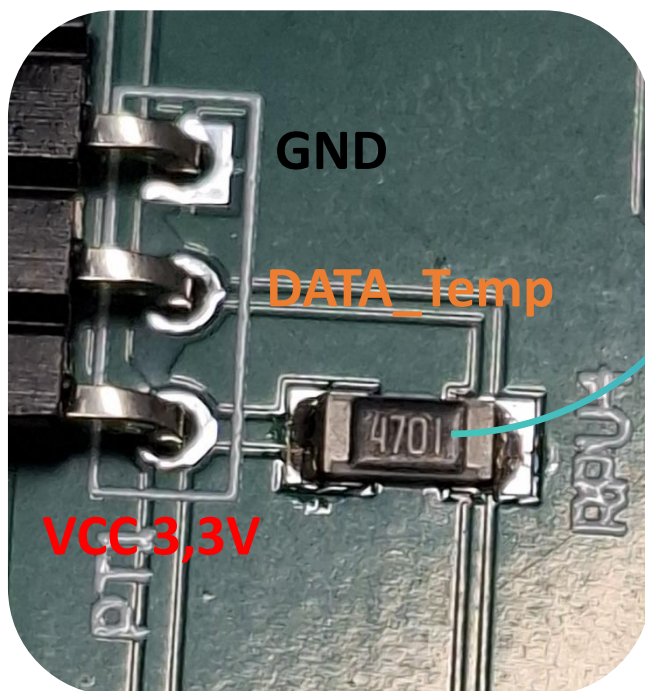
Grâce au **protocole OneWire**, chaque capteur possède une **adresse unique** que l'ESP32 peut **détecter au démarrage**. Cela permet de **lire individuellement** leurs valeurs même s'ils sont **connectés sur la même ligne**.

La PCB intègre donc **quatre capteurs DS18B20** destinés à **mesurer la température de la végétaline**.

Un **cinquième capteur**, dédié à la **mesure de la température au cœur de la ruche**, est quant à lui **déporté à l'extérieur** de la carte et raccordé via une **prise 3 pins** comprenant :

- **VCC (alimentation du capteur)**
- **Signal (ligne de données OneWire)**
- **GND (référence de masse)**

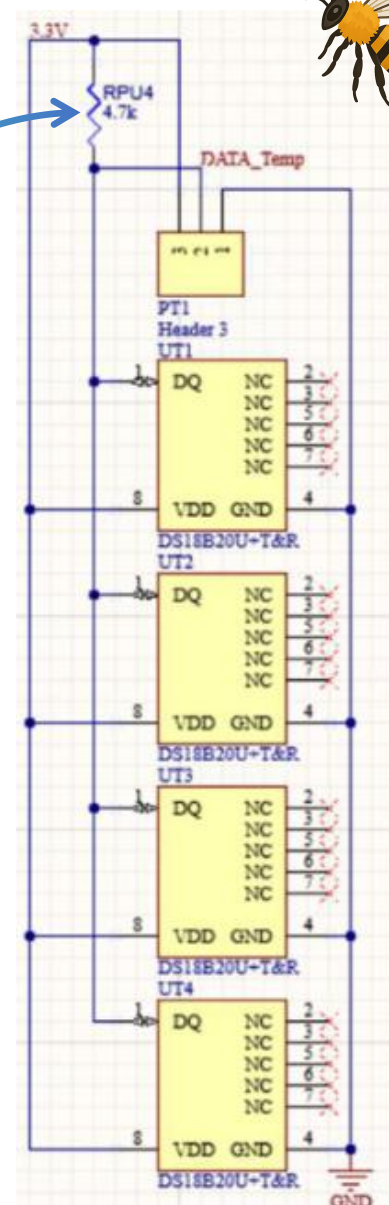
Cette disposition permet de **placer le capteur ruche directement dans l'environnement intérieur de la ruche** tout en laissant tout l'électronique sur la PCB principale. Grâce au **protocole OneWire**, ce capteur externe **partage lui aussi la même ligne de communication** que les autres capteurs, tout en conservant une **adresse unique** permettant de l'**identifier facilement**.

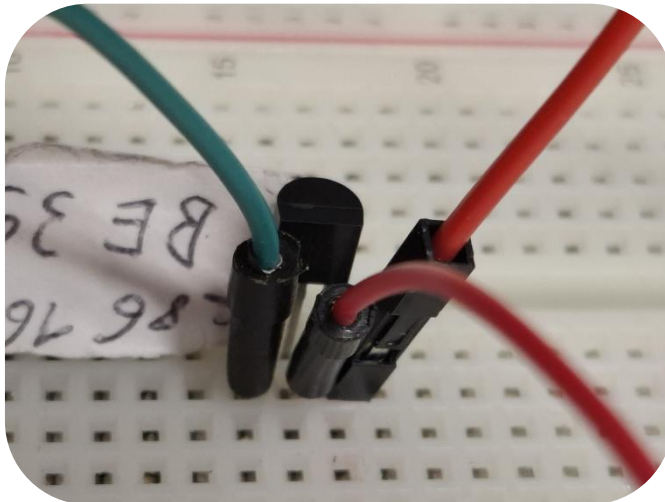


Sur le **connecteur de la PCB** pour le **capteur de la ruche** nous avons pu **identifier les pins** correspondant à l'aide du placement de la **résistance de 4,7kOhms**.

Ainsi nous avons de **haut en bas** :

GND / Signal OneWire (DATA_Temp) / VCC (3,3V)





Nous avons ensuite connecté le capteur de température de la ruche sur une «breadboard» de la façon suivante (de gauche à droite avec le côté plat vers nous) :
GND / Signal OneWire / VCC (3,3V)

Le code Arduino parcourt ensuite l'ensemble des capteurs détectés, récupère leurs températures respectives, puis calcule la moyenne des quatre mesures afin d'obtenir une valeur représentative de la température de la végétaline.

Nous avons affiché les adresses des capteurs détectés. Cela nous a permis de vérifier que chaque adresse correspondait bien à un capteur précis.

En particulier, nous avons identifié quelle adresse correspondait au capteur placé dans la ruche. Nous avons donc adapté le code pour que, si l'adresse détectée correspond à celle du capteur de la ruche, la température soit stockée dans la variable tRuche.

L'adresse du capteur dans la ruche est donc stockée avec :
DeviceAddress capteurRuche =
{ 0x28, 0x61, 0x64, 0x08, 0xEB, 0xE3, 0x98, 0x0D };



Nous avons utilisé **OneWire** et **DallasTemperature** : pour communiquer avec les capteurs de température DS18B20.

Ensuite, nous avons déclaré les variables et constantes :

- **tVeg** et **tRuche** : pour stocker les températures moyennes de la végétaline et de la ruche.
- **temperatureConsigne** : température cible pour le chauffage de la ruche.
- **TEMP_SENSOR_PIN** : broche à laquelle sont connectés les capteurs DS18B20.
- **capteurRuche** : adresse spécifique du capteur dans la ruche.

Dans le `setup()`, on initialise les capteurs **DS18B20** et on compte combien sont connectés, puis on affiche les adresses de tous les capteurs pour identifier le capteur de la ruche avec :

- `sensors.begin();`
- `numberOfDevices = sensors.getDeviceCount();`
- `getAddress(tempDeviceAddress, i)` qui récupère l'adresse de chaque capteur
- `void printAddress(DeviceAddress deviceAddress)`

Ensuite, dans le `loop()`, on lit les températures avec «`requestTemperatures(tempDeviceAddress)`» qui demande à tous les capteurs de mesurer leur température.

On trie les données :

On parcourt tous les capteurs détectés.

Pour chaque capteur, on récupère sa température avec `sensors.getTempC()`.

Puis, «`memcmp`» compare deux tableaux d'octets :

- `tempDeviceAddress` → adresse du capteur qu'on est en train de lire
- `capteurRuche` → adresse du capteur de la ruche
- `8` → car une adresse **DS18B20** contient **8 octets**



Enfin, on calcule la moyenne pour `tVeg` : on divise par le nombre de capteurs de végétaline, pour avoir une valeur représentative.

```
// Initialisation du bus OneWire et du capteur
OneWire oneWire(TEMP_SENSOR_PIN);
DallasTemperature sensors(&oneWire);
int numberOfDevices; // Number of temperature devices found
int numberOfCapteurRuche = 1;

Adafruit_INA219 ina219;
#define SIGNAL1 2
#define SIGNAL2 5 // resistances
bool heating = false;

#define SDA_PIN 21 // pin ina219 serial data
#define SCL_PIN 22 // pin ina219 serial clock

DeviceAddress capteurRuche = { 0x28, 0x61, 0x64, 0x08, 0xEB, 0xE3, 0x98, 0x0D };
```


VOID SETUP () :

```
// Start up the library for temp sensor
sensors.begin();
// Grab a count of devices on the wire
numberOfDevices = sensors.getDeviceCount();
// Loop through each device, print out address
for(int i=0;i<numberOfDevices; i++) {
  // Search the wire for address
  if(sensors.getAddress(tempDeviceAddress, i)) {
    Serial.print("SENSOR found n° ");
    Serial.print(i, DEC);
    Serial.println();
  }
  else {
    Serial.print("Ghost device at ");
    Serial.print(i, DEC);
    Serial.println(" but could not detect address. Check power and cabling...");
  }
}
//end temp config
delay(200);
```

VOID LOOP () :

```
// Lire la valeur du capteur de temperature
sensors.requestTemperatures(); // Send the command to get temperatures
// Loop through each device, print out temperature data
tVeg = 0.0;
tRuche = 0.0;
for(int i=0;i<numberOfDevices; i++) {
  // Search the wire for address
  if(sensors.getAddress(tempDeviceAddress, i)){
    Serial.print("Capteur: ");
    Serial.println(i,DEC);
    float tempC = sensors.getTempC(tempDeviceAddress);
    if (tempC == DEVICE_DISCONNECTED_C) {
      Serial.println("Capteur de temp deconnecte !");
      return;
    }
    if (memcmp(tempDeviceAddress, capteurRuche, 8) == 0) {
      tRuche = tRuche + tempC;
    }
    else {
      tVeg = tVeg + tempC;
    }
    Serial.print(tempC);
    Serial.println("°C");
  }
}
tVeg = tVeg/(numberOfDevices-numberOfCapteurRuche); // moyenne des mesures de température pour la végétaline
```

