RAPPORT DE SÉANCE 6 :

Durant la **6ième séance** nous avons travaillé principalement sur l'**intégration matérielle** de la **carte électronique** (**PCB**) du projet ainsi que sur les **premiers tests** de **fonctionnement du système**.

Dans un premier temps, nous avons continué le **code Arduino** afin que la puissance de chauffe s'**adapte à la puissance du panneau solaire**.

La **puissance mesurée power_mW** est **convertie en valeur de PWM** (entre **0 et 255**) grâce à une **règle de proportion** :

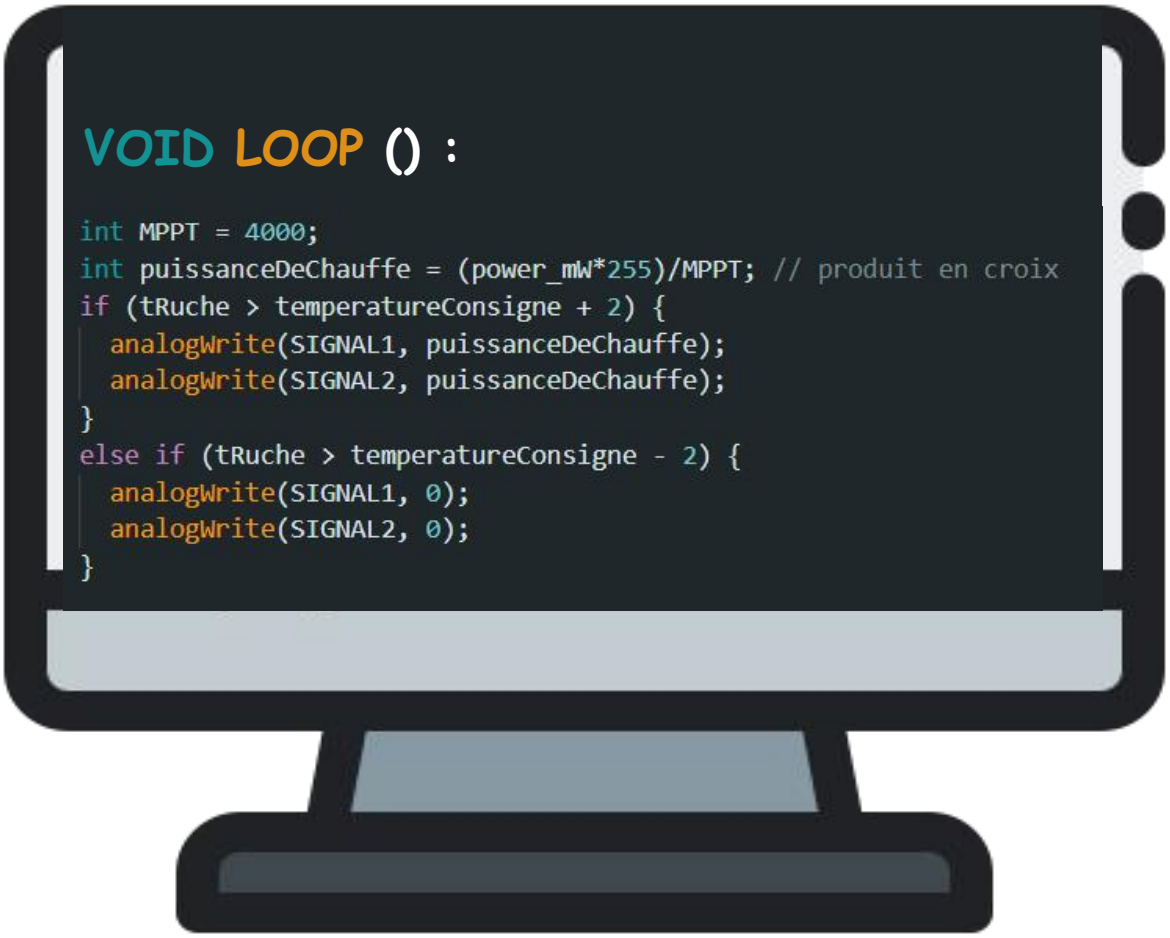
int puissanceDeChauffe = (power_mW * 255) / MPPT;

où **MPPT** représente approximativement la **puissance maximale attendue du panneau solaire** (*à mesurer*).

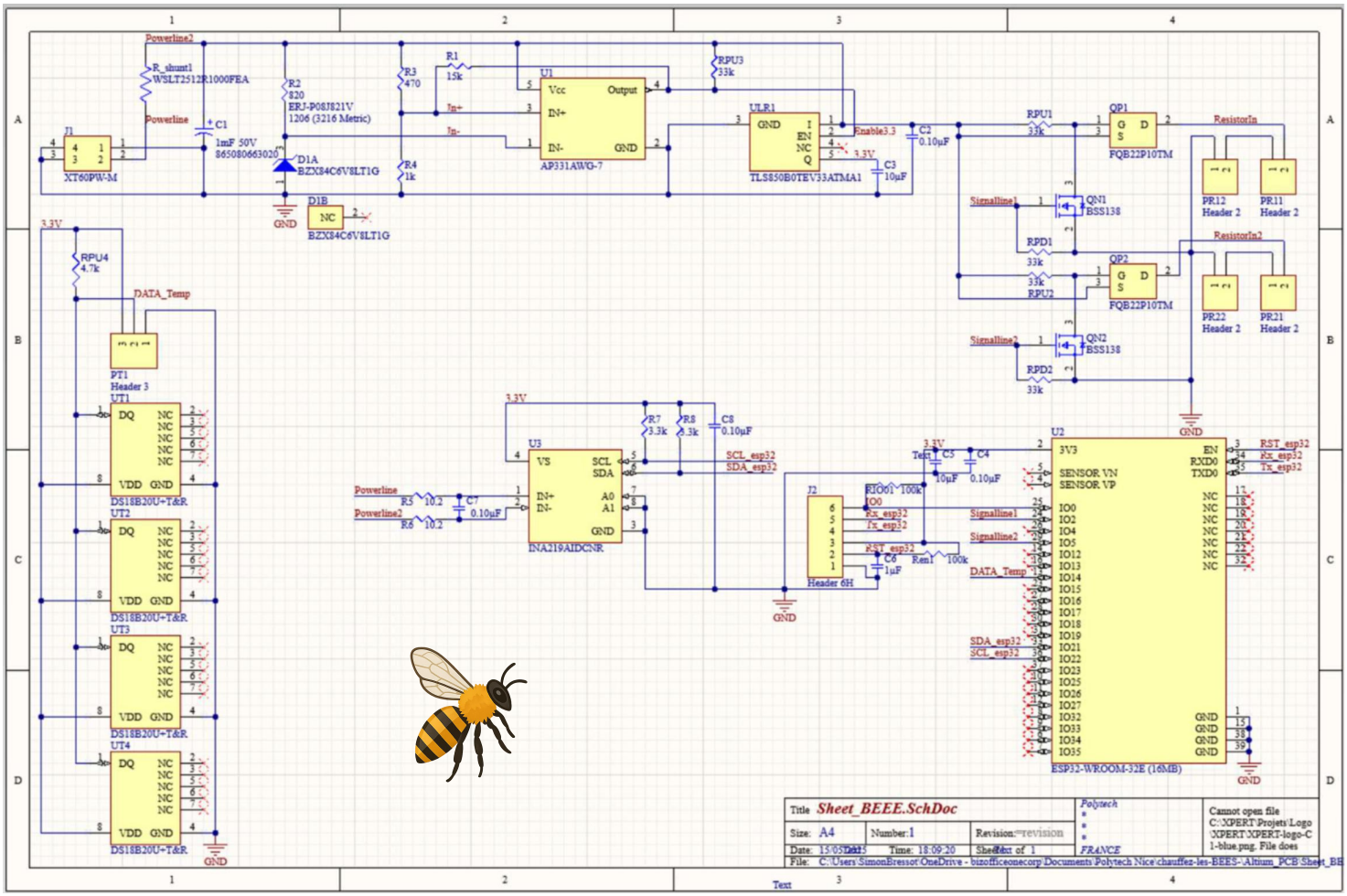
Ainsi, plus la **puissance disponible** est **élevée**, plus les **résistances peuvent chauffer**, et à l'inverse, lorsque le **panneau produit peu**, la **chauffe est naturellement limitée pour ne pas provoquer d'effondrement de tension**.

Ce principe permet une **adaptation automatique** et **continue** de la **puissance de chauffe**, s'intégrant parfaitement à un système alimenté exclusivement par **énergie solaire**.

Pour **tester les résistances**, nous avons utilisé «**digitalWrite**(signalline1/2, **HIGH**);».



Ensuite, **Hussein** m'a **expliqué en détail l'architecture globale de la PCB** à partir du **schéma électronique suivant**.



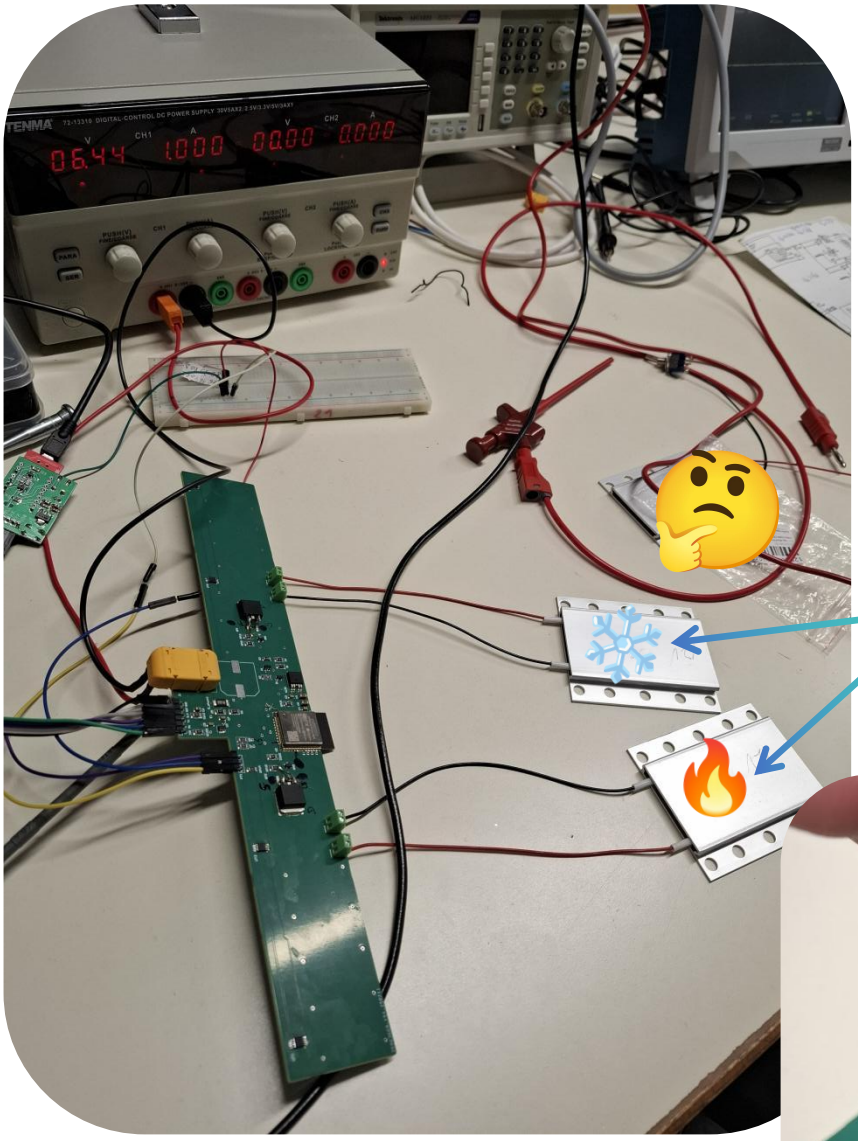
La **partie située en haut** du schéma correspond à l'**alimentation du système par panneau solaire**. Cette section comprend une **diode Zener** ainsi qu'un **comparateur de tension (U1)**, dont le rôle est de **fixer une tension seuil** à partir de laquelle le **système** est **autorisé à s'activer.**

Lorsque la **tension fournie par le panneau solaire est suffisante**, le **comparateur autorise le fonctionnement du reste de la carte**.

Cette **alimentation** est ensuite **régulée** par un **régulateur de tension (ULR1)**, qui **délivre une tension stable de 3,3 V**. Cette tension est **distribuée à l'ensemble des composants nécessitant cette alimentation**, notamment les **capteurs de température** et la carte **ESP32**.

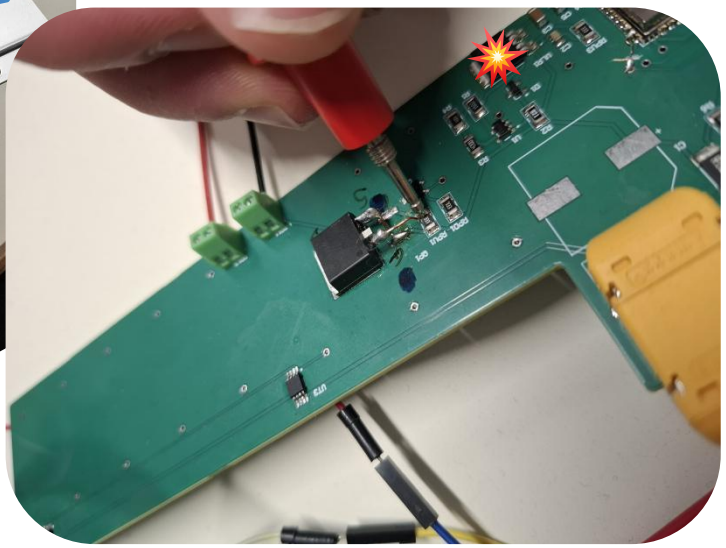
La carte **ESP32** est représentée par le **composant U2**. Elle constitue le **cœur du système** et assure à la fois l'**acquisition des données capteurs**, la **communication (Bluetooth)**, ainsi que le **pilotage des résistances chauffantes**.

Au centre du schéma, on retrouve également le **capteur INA219**, utilisé pour **mesurer la tension et le courant fournis par le panneau solaire**. Ces mesures permettent d'**estimer la puissance disponible** et **d'optimiser la puissance de chauffe des résistances** afin de s'adapter aux variations d'ensoleillement.
Sur la **partie gauche** de la carte se trouvent les **capteurs de température**. **4 capteurs** sont directement **intégrés sur la PCB (UT1 à UT4)**, et une **prise** permet de **connecter deux capteurs de température externes** destinés à mesurer la **température de la ruche (PT1)**. Tous ces capteurs **communiquent avec l'ESP32** via **un bus One Wire**, en **partageant le même fil de données nommé *DATA_Temp***. Chaque capteur est **identifié par une adresse unique**, ce qui permet à l'**ESP32 de les distinguer**.
Enfin, la **partie située en haut à droite** du schéma correspond au **circuit de commande des résistances chauffantes**. Celles-ci sont **pilotées par des signaux PWM** (***SignalLine1*** *et* ***SignalLine2***) **générés par l'ESP32**. La **commande de puissance** est réalisée à l'aide d'un montage comprenant un **transistor NMOS** suivi d'un **transistor PMOS**, permettant de **commuter efficacement les résistances**.

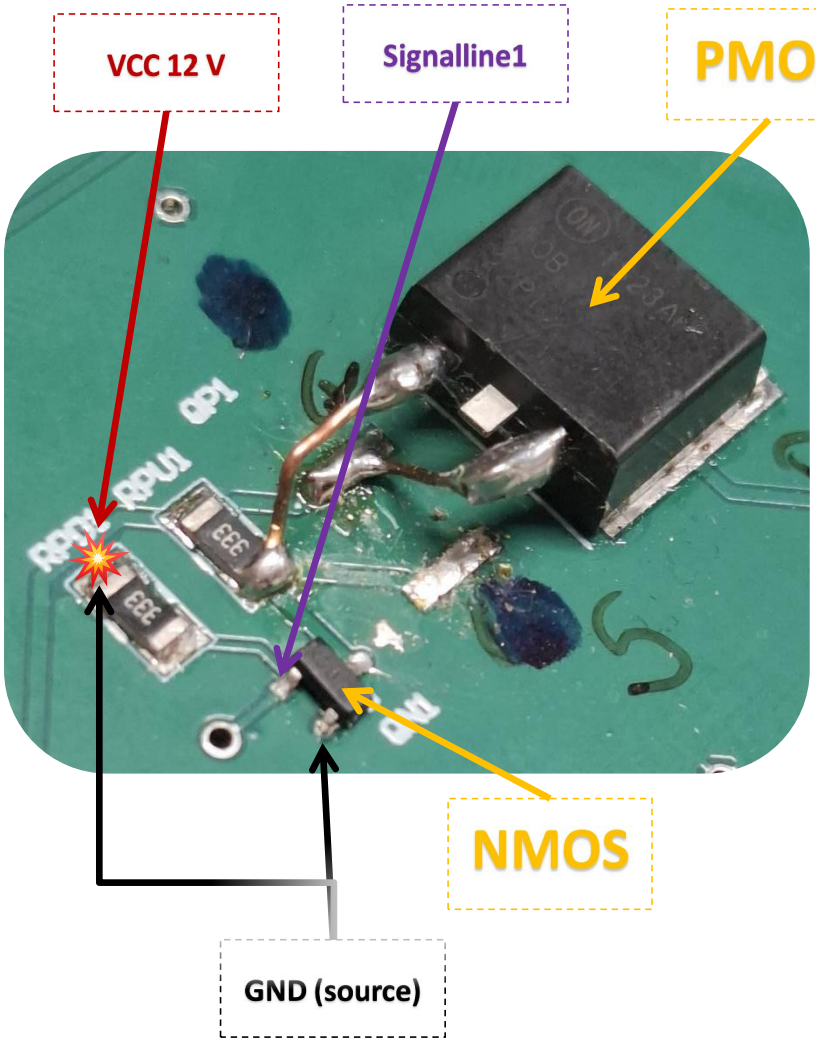


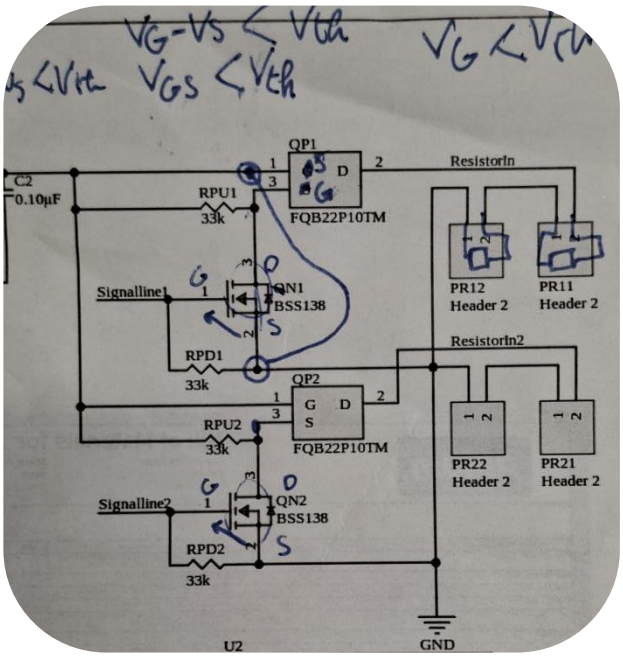
Une fois les résistances branchées, nous avons procédé à des tests de chauffe.
Lors de ces essais, nous avons constaté qu'une des résistances ne chauffait pas.
Afin de vérifier si le problème provenait de la résistance elle-même, nous avons inversé les deux résistances.
Cependant, le problème persistait : la même voie ne fonctionnait toujours pas.

Résistances



Nous en avons alors déduit que le **dysfonctionnement provenait de la PCB**, plus précisément de la **ligne de commande associée à SignalLine1**.
Par la suite, nous avons **constaté un incident majeur** : le **régulateur de tension** avait **fondu**, ce qui a entraîné l'**endommagement de la carte ESP32**.
La PCB est donc devenue inutilisable !





Une **phase d'investigation** a été menée afin d'**identifier l'origine de la défaillance**. **Après observation** et **mesures**, nous avons relevé que la **résistance RPD1** se trouvait à un **potentiel de 12 V**, alors qu'elle aurait dû être **reliée à la masse (0 V)**.
Cette anomalie nous amène à penser qu'**un court-circuit s'est produit**, probablement dû à un problème de routage ou de **placement de composants sur la PCB**.
En particulier, la **résistance RPD1** semble avoir été **placée trop près de la ligne 12 V,** ce qui aurait **provoqué un contact parasite ou une fuite de potentiel**.
Ce défaut aurait entraîné la **destruction du régulateur de tension**, puis de **l'ESP32**.