

Projet Electronique

Sujet du projet : Traitement de CO₂ dans une ruche.

Introduction :

L'un des plus gros dangers dans une ruche est la présence de varroas. Les varroas sont des petits acariens venant s'agripper et se nourrissant du corps gras des abeilles. Cela va provoquer une diminution de l'espérance assez important chez ces dernières. Le parasite influence la capacité d'apprentissage des butineuses ce qui influe sur le comportement de vol, la direction ainsi que le succès de retour à la ruche. Certaines abeilles ne meurent pas à l'extérieur mais se trompent de ruche, participant au phénomène de dérive des butineuses. Cette dérive a pour effet d'augmenter les transmissions de varroas dans les ruches voisines.

Les varroas sont plus sensibles au CO₂ que les abeilles, c'est pour cela que nous allons réguler une certaine quantité de CO₂ dans la ruche.

Réalisation :

Nous avons entrepris des tests sur une carte préalablement conçue par un autre groupe. Cette carte, contrôlée par un ESP32, est équipée de trois capteurs de CO₂ de référence CCS811. L'objectif de cette carte est de mesurer le niveau de CO₂ à différents emplacements dans la ruche en disposant la carte verticalement. Cette disposition est cruciale étant donné que la densité de l'air peut entraîner des variations significatives du taux de CO₂ à différents niveaux de la ruche.

Lors des tests initiaux des capteurs de CO₂ (CCS811) sur la première version de la carte, nous avons rencontré des difficultés. Les mesures obtenues n'étaient pas satisfaisantes, ce qui a nécessité des améliorations. C'est ainsi que nous avons procédé au développement d'une deuxième version de la carte, intégrant de nouveaux modules.

Cette nouvelle version de la carte combine deux capteurs sur une même carte. Tout d'abord, l'ajout d'un ENS160 offre une solution plus complète. Ce module est capable de mesurer non seulement le taux de CO₂, mais également la concentration totale de composés organiques volatils (TVOC) ainsi que la qualité générale de l'air. Ensuite, nous avons intégré un capteur ATH20, qui permet de mesurer la température et l'humidité.

L'un des avantages majeurs de ces modules est leur facilité d'intégration avec l'ESP32. Chaque module peut utiliser soit le protocole I2C, soit le SPI pour communiquer avec l'ESP32, offrant ainsi une flexibilité dans la configuration. De plus, chaque module est équipé de huit broches, comprenant principalement des broches d'alimentation et de communication, ce qui simplifie grandement le processus de connexion et d'intégration dans le système global.

Pin d'alimentation :

- VIN - il s'agit de la broche d'alimentation, comme les capteur de gaz et de température peuvent utiliser 3V, nous avons inclus un régulateur de tension à bord qui prendra 3-5VDC et le convertira en toute sécurité.

- 3V3 - c'est la sortie 3.3V du régulateur de tension, vous pouvez y prélever jusqu'à 100mA si vous le souhaitez
- GND - Masse commune pour l'alimentation et la logique.

Pin de communication :

- CS : cette pin Chip Select permet de sélectionner le protocole de communication, si elle au niveau haut les capteurs vont communiquer en I2C si ils sont en niveau bas en bascule en SPI.
- MOSI/SDA : c'est une pin entrée/sortie en mode SPI c'est Master Output Slave Input et en mode I2C c'est Bus Bidirectionnel data
- SCLK/SCL : c'est une pin entrée SPI Serial Clock / I2C Bus Serial Clock Input
- MISO/ADDR : c'est une pin entrée / sortie SPI Master Input Slave Output / I²C Address Select
 - I2C ADDR pin lorsque elle a niveau haut l'adresse I2C est 0x53 /
 - I2C ADDR pin lorsque elle a niveau haut l'adresse I2C est 0x52

Pin d'interruption :

- INIT : Pin d'interruption



Figure 1 : Le nouveau module utilisé (ENS160+AHT20)

Sur cette version de la carte, nous avons rencontré d'importants défis lors de la détection et de l'intégration des capteurs. Ces capteurs utilisent à la fois les protocoles de communication I2C et SPI, ce qui ajoute une complexité supplémentaire à leur utilisation. Une broche cruciale de connexion, appelée "CS", est utilisée pour définir le mode de communication du capteur : lorsque cette broche est maintenue à un niveau bas, le capteur fonctionne en mode SPI, tandis que lorsqu'elle est à un niveau haut, le capteur communique en mode I2C. De plus, chaque capteur possède une adresse I2C qui doit être unique pour permettre une communication correcte avec le microcontrôleur.

Cependant, nous avons constaté une limitation significative : ces capteurs ne permettent que deux adresses I2C différentes, généralement 0X52 ou 0X53, en fonction du niveau logique de la broche ADDR. Cette contrainte a posé un défi majeur lors de l'intégration des capteurs, car nous avons besoin de trois mesures simultanées pour obtenir des données précises sur le niveau de CO2, la qualité de l'air, le TVOC et la température.

Malgré ces défis, nous avons réussi à obtenir des mesures cohérentes en utilisant la bibliothèque Arduino Adafruit ENS160 pour tester les capteurs de CO₂, la qualité de l'air et le TVOC. Cependant, la limitation des adresses I2C nous a confrontés à une contrainte majeure : nous n'avons pu obtenir que deux mesures simultanées sur les deux capteurs ENS160 disponibles. Une mesure était obtenue à partir de l'adresse 0X52 et une autre à partir de l'adresse 0X53. Cela signifie que nous ne pouvions pas récupérer les données du troisième capteur de CO₂ ni obtenir simultanément la mesure de température avec le taux de CO₂.

Pour la mesure de la température, nous avons utilisé la bibliothèque DFRobot_AHT20. Dans notre quête pour obtenir des mesures complètes, nous avons envisagé une solution alternative. L'idée était d'effectuer deux mesures simultanées avec deux capteurs, puis d'éteindre l'un des capteurs et d'allumer le troisième pour obtenir la troisième mesure. Cependant, cette approche a rencontré des obstacles. Le principal défi résidait dans le temps de préchauffage nécessaire pour obtenir des mesures précises. Ce temps était d'au moins 60 secondes, ce qui rendait cette solution inefficace pour une acquisition de données en temps réel.

Un peu de détails sur les bibliothèques utilisées pour effectuer les mesures :

1. Adafruit ENS160 : Cette bibliothèque est développée par Adafruit et est spécifiquement conçue pour les capteurs environnementaux. Elle prend en charge la communication avec des capteurs tels que le CCS811 et l'ENS160. Avec cette bibliothèque, vous pouvez obtenir des mesures de CO₂, de TVOC (composés organiques volatils totaux) et de qualité de l'air. Elle offre une interface conviviale et des fonctions faciles à utiliser pour récupérer les données des capteurs. De plus, elle est bien documentée, ce qui facilite son utilisation pour les développeurs.

Lien : <https://www.arduino.cc/reference/en/libraries/ens160-adafruit-fork/>

2. DFRobot_AHT20 : Cette bibliothèque est développée par DFRobot et est spécifiquement conçue pour le capteur de température et d'humidité AHT20. Elle permet de récupérer facilement les mesures de température et d'humidité à partir du capteur AHT20. Comme la bibliothèque Adafruit ENS160, elle offre une interface simple et des fonctions prêtes à l'emploi pour interagir avec le capteur. Elle est également bien documentée, ce qui facilite son intégration dans les projets Arduino.

Lien : https://github.com/DFRobot/DFRobot_AHT20

```
16:47:03.834 -> Concentration of total volatile organic compounds : 21 ppb
16:47:03.834 -> Carbon dioxide equivalent concentration : 400 ppm
16:47:03.834 ->
16:47:04.815 -> Sensor operating status : 2
16:47:04.815 -> Air quality index : 2
16:47:04.815 -> Concentration of total volatile organic compounds : 69 ppb
16:47:04.860 -> Carbon dioxide equivalent concentration : 494 ppm
16:47:04.860 ->
16:47:05.857 -> Sensor operating status : 2
16:47:05.857 -> Air quality index : 4
16:47:05.857 -> Concentration of total volatile organic compounds : 1028 ppb
16:47:05.857 -> Carbon dioxide equivalent concentration : 1009 ppm
16:47:05.857 ->
16:47:06.873 -> Sensor operating status : 2
16:47:06.873 -> Air quality index : 3
16:47:06.873 -> Concentration of total volatile organic compounds : 437 ppb
16:47:06.873 -> Carbon dioxide equivalent concentration : 847 ppm
16:47:06.873 ->
16:47:07.846 -> Sensor operating status : 2
16:47:07.885 -> Air quality index : 3
16:47:07.885 -> Concentration of total volatile organic compounds : 328 ppb
16:47:07.885 -> Carbon dioxide equivalent concentration : 811 ppm
16:47:07.885 ->
16:47:08.869 -> Sensor operating status : 2
16:47:08.869 -> Air quality index : 3
16:47:08.869 -> Concentration of total volatile organic compounds : 272 ppb
16:47:08.869 -> Carbon dioxide equivalent concentration : 775 ppm
16:47:08.915 ->
```

Figure 2 : Exemple de mesures du CO2 + qualité de l'air + TVOC

Dans notre processus d'amélioration du système, nous avons entrepris la conception d'une troisième version, intégrant non seulement les nouveaux capteurs, mais également un commutateur multiplexeur I2C de référence PCA9548A. Cette évolution visait à résoudre le défi rencontré précédemment, celui de gérer efficacement la communication avec plusieurs capteurs à l'aide d'un unique microcontrôleur ESP32.

Le PCA9548A est un composant essentiel dans cette conception. Il s'agit d'un commutateur multiplexeur I2C offrant huit canaux distincts. Son rôle principal est de permettre l'extension des capacités de communication I2C d'un microcontrôleur en autorisant la connexion simultanée de plusieurs périphériques sur un même bus I2C. L'aspect crucial de ce commutateur réside dans sa capacité à sélectionner dynamiquement l'un des huit canaux disponibles, et ainsi à diriger le flux de données I2C vers le périphérique connecté à ce canal spécifique.

Concrètement, cela signifie que chaque capteur peut être connecté à l'un des huit canaux du PCA9548A. Ensuite, l'ESP32 peut sélectionner le canal correspondant au capteur dont il souhaite récupérer les données, en écrivant simplement les commandes appropriées via le bus I2C. Cette approche offre une solution élégante pour gérer plusieurs capteurs sur un seul bus I2C, en permettant au microcontrôleur de communiquer sélectivement avec chaque capteur selon les besoins du système.

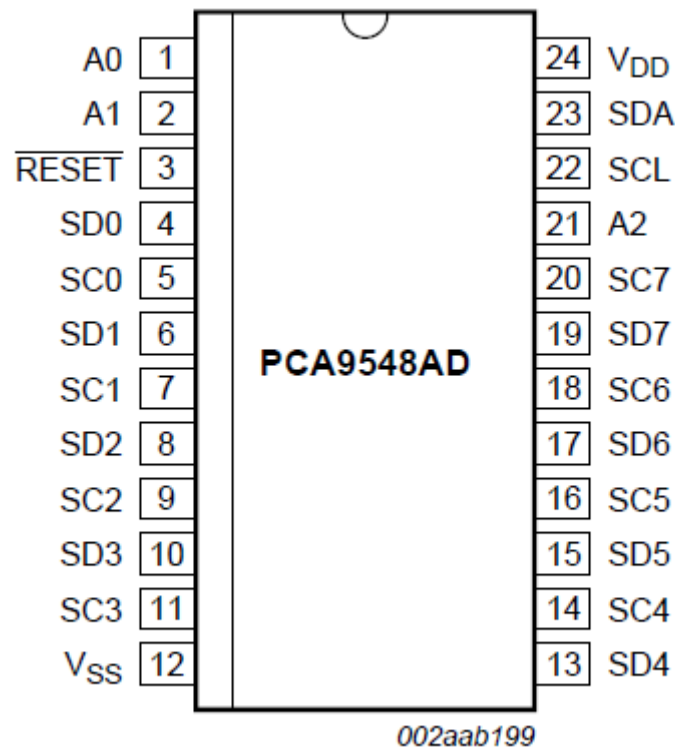


Figure 3 : MUX PCA9548A

Pour commencer le processus de connexion, nous avons pris soin de connecter la broche d'alimentation du PCA9548A, ainsi que les broches SDA (données), SCL (horloge) et RST (réinitialisation) au niveau bas, assurant ainsi un état initial stable pour le commutateur multiplexeur. Ensuite, nous avons établi les connexions entre les broches de commande (A0, A1 et A2) du PCA9548A et les broches correspondantes de l'ESP32, permettant à ce dernier de contrôler sélectivement les canaux du MUX. Une fois la configuration de base effectuée, nous avons procédé à la connexion des broches de sortie du MUX (SD0, SC0) au premier capteur. Cette étape consistait à acheminer les signaux de données (SD) et d'horloge (SC) du canal sélectionné vers les broches correspondantes du capteur. Ce processus a été répété pour les capteurs suivants, avec les broches SD1, SC1 et SD2, SC2 connectées aux capteurs 2 et 3 respectivement. Chaque étape de connexion a été soigneusement réalisée en suivant les spécifications du schéma de câblage et en vérifiant la conformité avec les besoins de communication des capteurs. Cette approche méthodique garantit une configuration fiable et cohérente du système, assurant ainsi un fonctionnement optimal lors de l'acquisition de données par l'ESP32 à partir des différents capteurs.

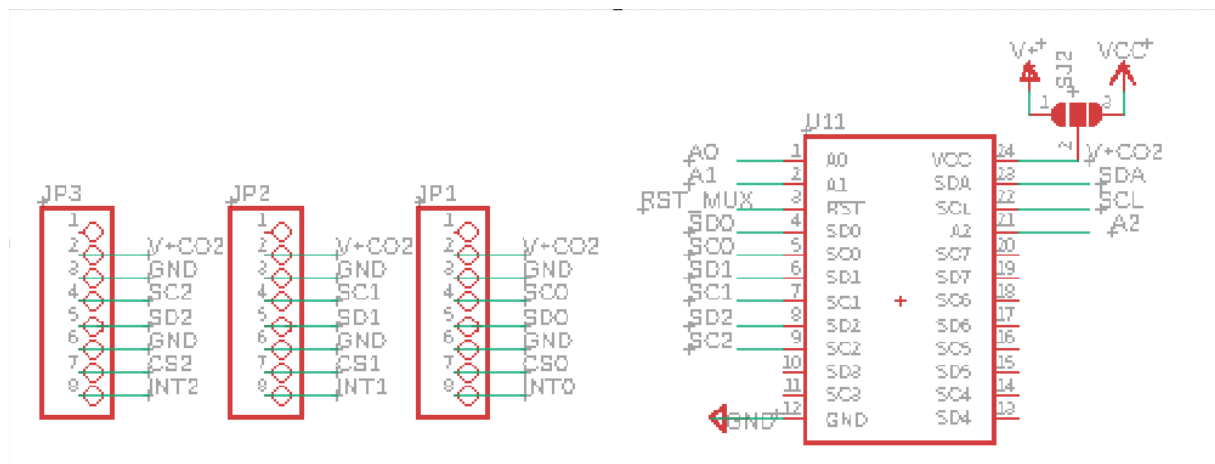


Figure 4 : Nouveau schématique pour le MUX et les 3 capteurs

Après avoir terminé les connexions et faire le schématique des composants essentiels, nous avons entamé le processus de routage de la carte. Cette phase cruciale du développement consiste à concevoir les chemins de connexion sur le circuit imprimé afin d'assurer une communication efficace et fiable entre tous les composants. "Cette phase du développement a été particulièrement exigeante et a demandé un investissement important en termes de temps et d'apprentissage. En effet, nous avons dû nous familiariser avec le logiciel de conception de circuits imprimés (CAD) Eagle, tout en apprenant les tenants et aboutissants de la conception de circuits imprimés.

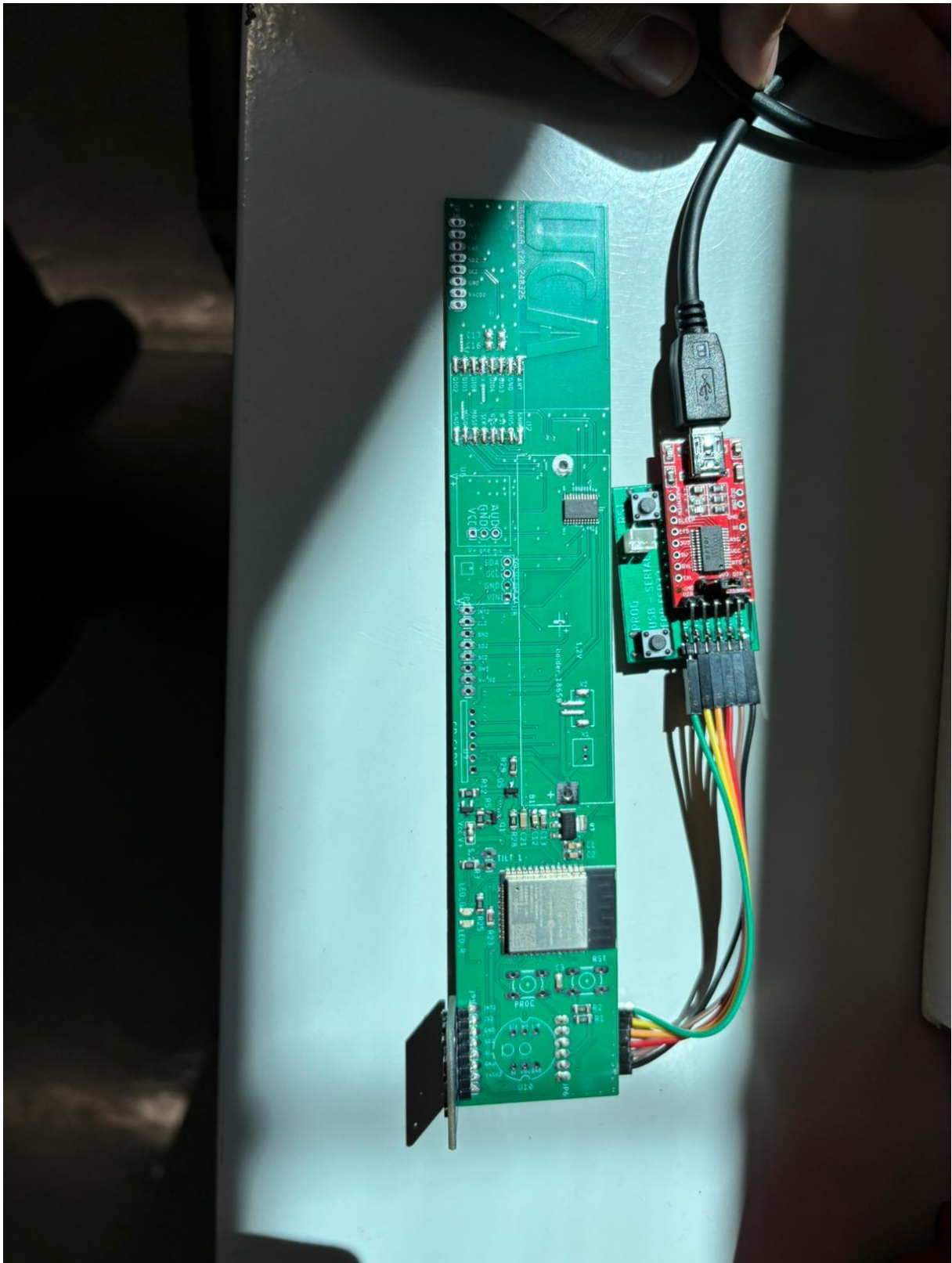
Nous avons également dû nous conformer à des normes rigoureuses, telles que les espacements entre les vias et les pistes, qui jouent un rôle crucial dans la fiabilité et les performances du circuit imprimé. Par exemple, nous avons appris que les espaces entre les vias et les pistes doivent respecter des dimensions spécifiques, avec un espacement de 12 pour les connexions et de 32 pour les pistes d'alimentation, conformément aux normes de l'industrie.

Ce processus d'apprentissage et de mise en pratique des normes de conception a été essentiel pour garantir la qualité et la robustesse de notre circuit imprimé final. Bien que cela ait représenté un défi, cela nous a également permis d'acquérir de nouvelles compétences et une expertise précieuse dans le domaine de la conception de circuits imprimés, renforçant ainsi notre capacité à réaliser des projets futurs avec succès.

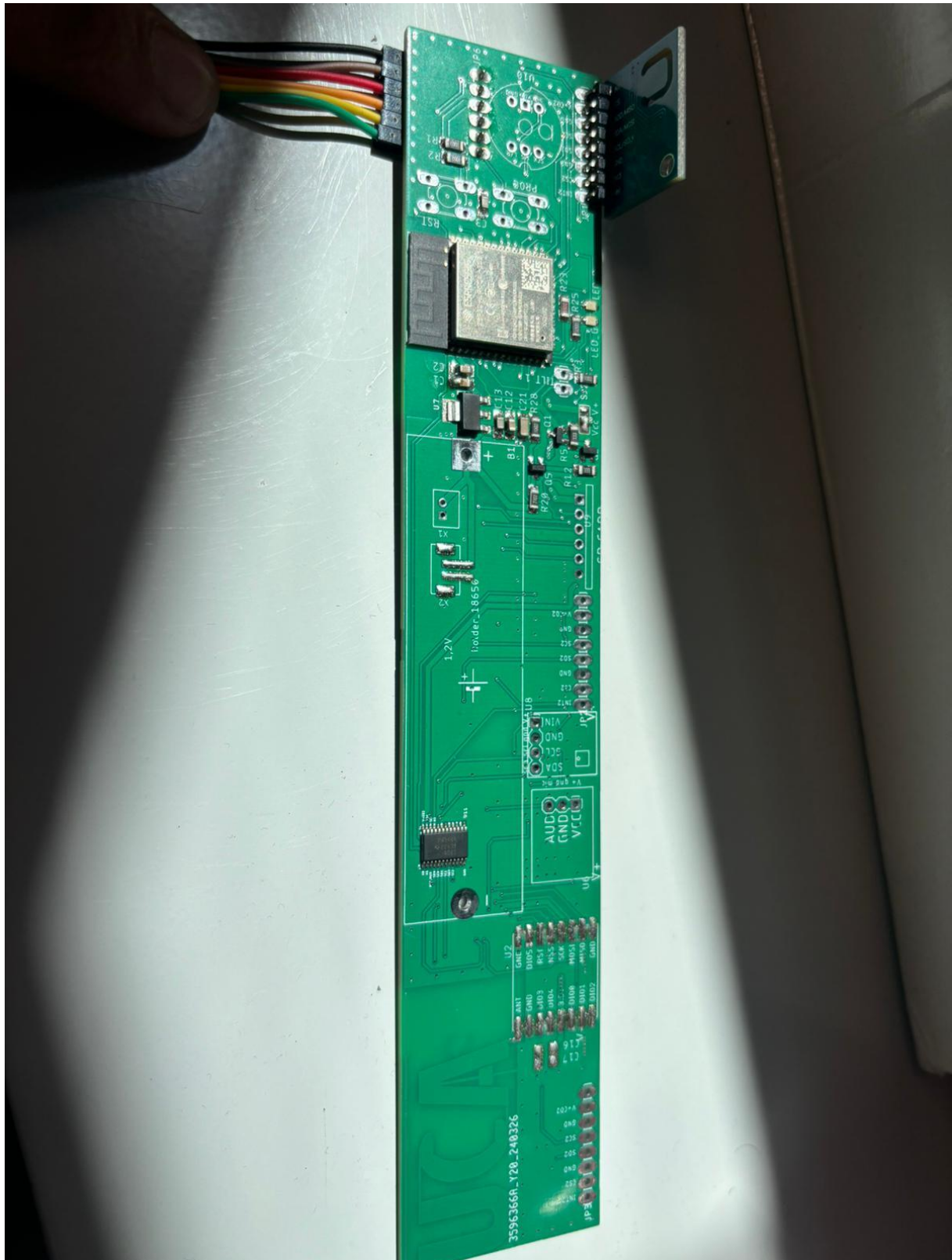
Après réception de la carte du fabricant, nous avons entrepris le soudage des composants, ainsi que la connexion d'un seul capteur pour effectuer des tests. Initialement, la carte semblait fonctionner correctement au niveau des LED et des connexions. Cependant, malheureusement, nous avons rencontré un obstacle majeur : nous n'avons pas réussi à détecter l'adresse I2C de ce capteur à l'aide de notre sketch scanner I2C.

Malgré nos efforts pour configurer correctement la pin MEASURE en mode HIGH et la pin RST du MUX en niveau haut, ainsi que nos tentatives de toutes les combinaisons possibles avec les broches de commande A0, A1 et A2, le capteur restait introuvable.

ZANTOUR Elyes
CARNEVALI Nicolas
ELSE4



ZANTOUR Elyes
CARNEVALI Nicolas
ELSE4



Conclusion

En conclusion, bien que nous ayons franchi plusieurs étapes importantes dans ce projet électronique, nous avons rencontré des défis inattendus lors des tests initiaux. Malgré nos efforts pour configurer correctement la carte et détecter l'adresse I2C du capteur, nous avons été confrontés à des difficultés techniques. Cependant, ces obstacles ne représentent pas un arrêt, mais plutôt une opportunité d'apprentissage et d'amélioration.

Les prochaines étapes consisteront à approfondir l'analyse du problème en testant le capteur à l'aide d'un oscilloscope, ce qui permettra de mieux comprendre les signaux et les interactions sur le bus I2C. En parallèle, il faut adapter le code pour intégrer la fonctionnalité du MUX, afin de permettre la sélection sélective des capteurs. Une fois ces défis techniques surmontés, le prochain groupe va poursuivre les tests en conditions réelles, en installant le système dans une ruche pour commencer à collecter des données précieuses sur l'environnement intérieur.