

Rapport Projet Séance 2 - comptage de varroa

Objectif de la séance :

L'objectif de cette séance était d'explorer et de comprendre le fonctionnement du code Arduino de l'ESP32-CAM. Plus spécifiquement, il s'agissait de trouver une méthode permettant de capturer une image avec l'ESP32 et de pouvoir la transmettre en local, potentiellement via le port série, étant donné qu'on n'a pas le droit d'utiliser de serveur et donc le wifi.

Travaux réalisés :

- Analyse du code Arduino de l'ESP32 – CAM
 - Compréhension des bibliothèques et fonctions fournies par Arduino avec l'ESP32 – CAM, pour la capture d'image (i.e. : format, qualité, ...).
- Communication série
 - J'ai essayé de comprendre comment l'ESP32 CAM MB, pouvait communiquer via le port série.
- Développement d'un script Python de réception
 - Dans le but de pouvoir visualiser les images transmises, j'ai essayé de créer un script Python capable de lire le flux série et de reconstruire l'image avec la bibliothèque Pyseries.

Résultats obtenus :

- J'ai réussi à identifier dans le code Arduino, le buffer qui permet la communication série des octets de l'image qui a été prise, et comment prendre une photo avec l'ESP32 – CAM.
- Les codes Arduino et Python, que j'ai réalisés ne sont pas encore concluants, je n'ai pas encore réussi à récupérer tous les octets de données envoyés par l'ESP32 – CAM, ce qui fait que je n'ai pas encore d'image reconstruite non plus.

Possible difficultés :

- Il y aura sûrement un problème de débit concernant la transmission de données via le port série de l'ESP32, donc on devra sûrement réduire la qualité de l'image, donc possible être amené à faire du filtrage avant de compter les varroas

Prochaines étapes :

- Faire fonctionner les liaisons série, et obtenir une image pour pouvoir un peu mieux visualiser avec quoi on devra travailler pour pouvoir effectuer le comptage de varroas, et aussi si possible trouver le meilleur compromis entre qualité et débit pour que le transfert ne prenne pas une éternité et que la qualité d'image soit traitable par la suite.