



Rapport Projet Electronique



Varroa Killer

Le système de comptage des varroas

Binôme : Coste- -Faig et Lecuivre



SOMMAIRE

1. Présentation du projet

- I. Introduction
- II. Choix du projet
- III. But du projet

2. Solutions techniques

- I. Point de départ
- II. Le système de récolte
- III. Le système de détection

3. Réalisation du projet

- I. La détection par image
- II. Le tapis roulant

4. Evolution future

- I. La détection
- II. Le tapis

5. Conclusion



Présentation du projet

I. Introduction

L'ensemble des projets proposés tournaient autour du thème de la ruche connectée.

Une ruche connectée est une ruche équipée de capteurs et de technologies modernes pour surveiller la santé des abeilles et les conditions environnementales dans lesquelles elles évoluent. Grâce à cette technologie, les apiculteurs peuvent surveiller à distance la température, l'humidité, la qualité de l'air et d'autres facteurs qui peuvent affecter la santé des abeilles et la production de miel.

Dans notre cas, les données collectées par les capteurs sont transmises en temps réel à un serveur, permettant à l'apiculteur de surveiller ses ruches à distance et de recevoir des alertes en cas de problèmes. Les ruches connectées sont également utiles aux apiculteurs afin de suivre la croissance de leur colonie, de suivre les périodes de floraison et de surveiller la production de miel.

A l'heure actuelle, les ruches connectées sont une solution efficace pour surveiller la santé des abeilles et réduire les pertes de colonies qui sont malheureusement très nombreuses. En utilisant des données précises et en surveillant les conditions environnementales, les apiculteurs peuvent prendre des mesures pour prévenir les maladies des abeilles et les protéger contre les facteurs environnementaux qui pourraient les affecter.

L'un des facteurs compliqués à détecter est la présence de parasites à l'intérieur de la ruche. L'un des exemples le plus révélateur sont les varroas.

II. Choix du projet

Plusieurs projets ont donc été proposés autour d'une ruche connectée, certains étaient déjà commencés l'année passée et d'autres totalement novateurs.

Nous avons choisi de prendre le projet de comptage des varroas morts car celui-ci correspondait au mieux à ce que nous voulions développer comme esprit de travail.

La force de ce projet est de pouvoir allier le côté électronique embarquée (software et hardware) avec de la mécanique notamment par la création d'un tout nouveau système.



Outre le côté technique, le choix d'un projet de comptage des varroas morts d'une ruche est important car il permet de surveiller efficacement les niveaux d'infestation de varroas dans la colonie d'abeilles. Les varroas sont l'un des principaux problèmes auxquels les apiculteurs sont confrontés, car ces parasites peuvent causer de graves dommages aux abeilles, réduire la production de miel et finir par tuer une colonie entière.



Figure n°1 : Image d'une abeille infestée par des varroas

Le comptage des varroas morts après le traitement d'une ruche permet à l'apiculteur de suivre l'efficacité des traitements utilisés pour lutter contre ce parasite. En surveillant régulièrement les niveaux de varroas dans une colonie d'abeilles, l'apiculteur peut prendre des mesures pour réduire l'infestation et préserver la santé de ses abeilles.

En outre, le comptage des varroas morts peut également aider à déterminer le moment optimal pour appliquer des traitements, en évitant de sur-traiter les abeilles tout en réduisant ainsi les risques pour l'environnement et surtout la santé de ses abeilles.

En somme, un projet comme celui-ci est un moyen simple mais efficace pour surveiller les niveaux d'infestation de varroas dans une colonie d'abeilles, en prenant en compte l'efficacité des traitements et la santé globale des abeilles. Cela peut grandement aider l'apiculteur à prendre des décisions plus éclairées et à mieux protéger leurs abeilles contre ces parasites nuisibles.

III. But du projet

Le but de ce projet est de réaliser un système autonome qui permettra un comptage précis et automatisé des varroas morts tombant de la ruche. Dans un premier temps, le système sera équipé de caméras qui auront pour objectif de détecter la présence de varroas morts et de les comptabiliser.

Le système doit être conçu de manière à être facile d'installation et d'utilisation. Il sera doté d'une interface utilisateur qui permettra à l'apiculteur de visualiser facilement les données collectées.



Ce système autonome peut également contribuer à réduire les coûts associés à la surveillance de l'infestation de varroas. En permettant une surveillance plus précise et automatisée, le système peut aider à éviter les traitements excessifs et coûteux contre les varroas, tout en assurant la santé des abeilles.

En résumé, le développement d'un système autonome de comptage des varroas morts permettra à l'apiculteur de surveiller plus efficacement l'infestation de varroas dans une colonie d'abeilles, de réduire le temps et les efforts nécessaires à cette surveillance, de limiter les coûts de traitement et d'assurer une bonne santé des abeilles.

Solutions techniques

I. Point de départ

Ce projet est complètement novateur car aucun travail en amont n'avait été effectué. Les seules choses que nous possédions étant une carte intégrant une petite caméra avec un écran LCD ainsi que des pistes de développement données par notre tuteur de projet.

L'idée est dans un premier temps de pouvoir intégrer sous une ruche, un système de récolte des éléments qui chutent de celle-ci. Puis dans un second temps de pouvoir analyser tous ces résidus, différencier les varroas morts du reste et ainsi pouvoir les comptabiliser.

II. Le système de récolte

Pour pouvoir récupérer tout ce qu'il tombe de la ruche, il faut dans un premier temps travailler sur un système intégrable sous la ruche.

L'idée de ce système est de fabriquer un compartiment à partir d'une hausse de ruche qui serait installé sous le corps de ruche.

Dans cette hausse, on fabriquerait un système de tapis roulant qui permettrait de récupérer les chutes de la ruche. Et en sortie de celui-ci, on y placerait le système permettant le comptage.



Le tapis serait constitué d'une toile de couleur uniforme et résistante, tendu entre 2 rouleaux dont l'un serait motorisé pour pouvoir le mettre en rotation.

Ce moteur serait commandé par la carte qui superviserait le système de comptage afin d'optimiser son fonctionnement et les composants.

Pour pouvoir nettoyer ce tapis, un balai installé au-dessous, viendrait faire tomber les résidus sur le sol.

III. Le système de détection

La première idée de détection que nous avons retenue est la reconnaissance par image.

Elle est possible grâce au tapis roulant car les résidus de la ruche tombent sur celui-ci et sont ainsi repartis à plat et sur un fond uni.

L'idée est qu'en fin du tapis, une caméra installée au-dessus puisse prendre des images de ce qu'il s'y trouve, et grâce à une intelligence artificielle puisse visualiser et compter les varroas morts présents.

Pour des raisons de facilité et de premier développement, nous allons commencer par utiliser une carte intégrant une caméra dans son architecture pour la détection.

Réalisation du projet

I. La détection par image

1. La carte Maixduino

La première carte embarquée fournit pour ce projet, est un modèle dit « Maixduino ». Ce circuit est un microcontrôleur open-source basé sur l'architecture RISC-V et conçu spécifiquement pour les applications d'intelligence artificielle (IA).



➤ Caractéristiques :

La Maixduino, permet de capturer des images en temps réel à l'aide de la caméra intégrée ou d'une caméra externe, puis de les traiter directement sur la carte. Elle permet d'effectuer des tâches telles que la détection d'objets, la reconnaissance faciale, le suivi de mouvement, la segmentation d'images et bien plus encore. Grâce à la puissance de l'unité de traitement d'IA Kendryte K210.

En outre, la carte Maixduino est dotée de nombreux ports d'extension, ce qui permet aussi de connecter d'autres capteurs et périphériques pour créer des systèmes plus complexes et interconnectés avec d'autres cartes. Il est possible également de communiquer avec d'autres appareils via des interfaces telles que UART, I2C et SPI.

Elle possède un Esp-32 intégré permettant de communiquer les données mesurées ou traitées et de pouvoir les transmettre.

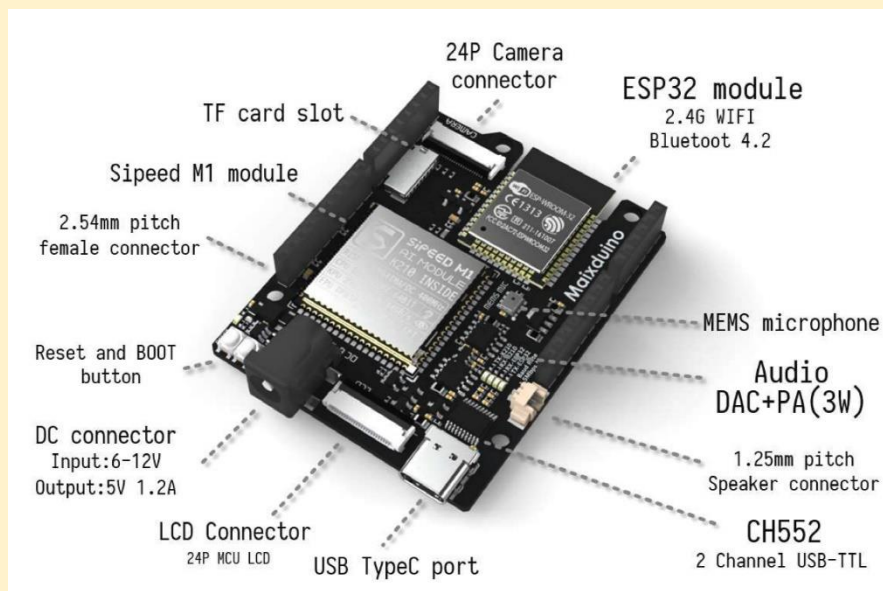


Figure n°2 : Illustration des éléments présents sur la carte Maixduino

➤ Méthode de programmation :

Cette carte peut se programmer de différentes façons, via l'IDE Arduino, de façon conventionnelle ou encore via un IDE spécifique : MaixPY, dont le langage de programmation est le python. Celui-ci permet de faire appel à plus de librairies, et bien plus de spécificités. Cet IDE nous permet par exemple avec un simple programme d'avoir un retour en direct de la caméra présente sur la carte Maixduino.



```
# Hello World Example
#
# Welcome to the MaixPy IDE!
# 1. Connect board to computer
# 2. Select board at the top of MaixPy IDE: 'tools->Select Board'
# 3. Click the connect button below to connect board
# 4. Click on the green run arrow button below to run the script!

import sensor, image, time, lcd

lcd.init(freg=15000000)
sensor.reset()

sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 2000)
clock = time.clock()

while(True):
    clock.tick()
    img = sensor.snapshot()
    lcd.display(img)
    print(clock.fps())
```

Tampon de frame désactivé - cliquez à nouveau sur le bouton de désactivation pour l'activer (en haut à droite)

Pas d'image

Figure n°3 : Screenshot de l'IDE MaixPY

Ce simple code, nous permet d'aborder les fonctions les plus importantes pour définir les capacités de la caméra.

En effet la fonction `.set_framesize` permet de donner la résolution voulue à la caméra. Qvga, Vga, Etc ... C'est un paramètre important car il nous permet d'augmenter la précision de la détection sur l'image.

Sur l'IDE Arduino, la carte Maixduino se programme comme tout autre microcontrôleur Arduino. Il suffit d'ajouter la carte dans le gestionnaire prévu à cet effet pour qu'elle soit reconnue afin d'utiliser les librairies correspondantes.



```
#include <Sipeed_ST7789.h>

SPIClass spi_(SPI0); // MUST be SPI0 for Maix series on board LCD
Sipeed_ST7789 lcd(320, 240, spi_);

void func()
{
    lcd.fillScreen(COLOR_WHITE);
    lcd.fillRect(65, 35, 198, 2, COLOR_BLACK);
    lcd.setTextSize(2);
    lcd.setTextColor(COLOR_BLACK);
    lcd.setCursor(65,20);
    lcd.println("COMPTEUR VARROA :");
    lcd.setTextSize(10);
    lcd.setTextColor(COLOR_BLACK);
    lcd.setCursor(160,120);
    lcd.println("1");
}

void func2()
{
    lcd.fillScreen(COLOR_RED);
    lcd.drawRect(20, 20, 50, 50, COLOR_WHITE);
    lcd.fillCircle(180, 50, 40, COLOR_WHITE);
    lcd.fillTriangle(10, 300, 200, 300, 200, 150, COLOR_WHITE);
    lcd.setTextSize(2);
    lcd.setTextColor(COLOR_WHITE);
    lcd.setCursor(1,100);
    lcd.println("hello Sipeed Maix");
}

void setup()
{
    lcd.begin(15000000, COLOR_RED);
    func();
}

void loop()
{
}
```

Figure n°4 : Illustration d'un programme d'affichage sur l'écran LCD

Ce code est un exemple de fonctionnement et de programmation de l'écran LCD intégré. Grâce à cet exemple nous avons pu réaliser une petite interface, qui nous permet d'afficher le nombre de varroas détectés.

➤ Travail effectué :

Pour commencer, nous avons effectué des tests de base afin de nous familiariser avec la programmation de la carte Maixduino. Nous avons réalisé des actions simples telles que l'allumage des LEDs et l'exécution de calculs élémentaires en utilisant l'IDE Arduino. Ces tests nous ont permis de nous remémorer les méthodes de programmation spécifiques à l'environnement Arduino.

Nous avons voulu par la suite programmer l'écran LCD de la carte Maixduino. Nous avons créé plusieurs menus pour afficher différents paramètres de la ruche, offrant ainsi un moyen pratique d'afficher et de consulter les informations pertinentes. Cela nous a permis de visualiser facilement les données de la ruche en temps réel.

Par la suite, nous avons entrepris la programmation de la caméra intégrée à la carte Maixduino en utilisant l'IDE Arduino. Cependant, nous avons rapidement rencontré une



première problématique : le module de la caméra n'était pas à jour avec le reste de la carte. Ceci nous a demandé un certain temps pour le diagnostic et résoudre le problème. Une fois cette difficulté résolue, nous avons réussi à afficher en direct ce que la caméra capturait sur l'écran LCD. Cela nous a ouvert de nouvelles possibilités pour la surveillance en temps réel ou la visualisation de scènes filmées par la caméra.

Dans notre recherche de bibliothèques Arduino permettant le traitement des images localement sur la carte Maixduino, nous avons malheureusement rencontré des difficultés supplémentaires. Tout d'abord la résolution de la caméra intégrée était très faible, ce qui rendait la détection de petit objets très complexe. De plus, nous avons découvert qu'il n'existait pas de bibliothèques disponibles permettant un traitement local des images sur la carte Maixduino, c'est-à-dire sans la relier à un ordinateur pour utiliser des bibliothèques telles qu'OpenCV. En effet celle-ci ne sont pas encore développées par le constructeur.

Afin d'améliorer nos capacités de vision, nous avons décidé d'ajouter une nouvelle caméra grand angle à la carte Maixduino. Cela nous a offert un champ de vision plus large, ce qui était bénéfique pour certains types d'applications. Cependant, il est important de noter que l'ajout de cette nouvelle caméra n'a pas entraîné une amélioration sur la qualité d'image, car la résolution restait inchangée.

Par conséquent, nous avons donc dû explorer d'autres solutions pour le traitement d'image. Nous voulions mettre en place un système de stockage des images en utilisant une carte micro SD, ce qui nous permettrait d'enregistrer les images capturées par la caméra pour un traitement ultérieur. Cependant, nous avons rencontré encore des difficultés supplémentaires, car la carte SD n'était jamais reconnue malgré des tentatives de nombreuses tentatives.

Nous avons également tenté de programmer le module ESP32 avec la carte Maixduino, mais nous n'avons malheureusement pas obtenu de résultat de bon fonctionnement. Après des investigations approfondies, nous avons réalisé qu'il était impossible de programmer l'ESP32 présent sur la carte Maixduino en raison d'une erreur de communication inconnue. Malgré nos efforts pour résoudre cette situation, nous n'avons pas pu établir une transmission fonctionnelle entre les deux microcontrôleurs.

Ensuite, nous avons décidé de programmer la caméra en utilisant MaixPy, une autre alternative à l'IDE Arduino pour la carte Maixduino. Cela nous a ouvert de nouvelles possibilités et nous a permis de voir le flux vidéo de la caméra directement sur notre ordinateur. Cependant, nous avons été confrontés à une limitation majeure : la résolution de la caméra était préconfigurée en QVGA (320x240 pixels).



Malgré nos efforts, nous n'avons pas réussi à augmenter suffisamment la résolution de la caméra. La mémoire de la carte étant très rapidement saturée, ce qui limitait la quantité de données d'image que nous pouvions traiter. Cela a rendu la détection d'objets encore plus complexe, car la faible résolution ne permettait pas de capturer suffisamment de détails pour une détection précise.

Cependant, nous avons réussi à mettre en œuvre la détection de visages et de cercles en utilisant des seuils pour ajuster la précision de la détection. Nous avons utilisé des techniques telles que la détection de contours et la recherche de formes géométriques pour identifier ces éléments dans le flux vidéo. Malheureusement, en raison de la résolution limitée, les formes détectées étaient environ 10 fois plus grandes que leur taille réelle. Par exemple, un varroa, qui mesure environ 1 millimètre, était détecté comme une forme d'environ 1 centimètre.

Notre objectif étant d'estimer approximativement le nombre de varroas présents dans la scène en comptant ces détections. Cependant, en raison de la limitation de résolution et de la taille des formes détectées, ces estimations n'étaient pas précises. Malgré ces contraintes, nous avons réussi à mettre en place un système de détection fonctionnel, ce qui nous a donné une idée générale de la présence de forme mais pas de varroa.

En conclusion, bien que nous ayons rencontré des défis techniques liés à la résolution limitée de la caméra et aux ressources de la carte Maixduino, nous avons pu explorer les capacités de traitement d'images en direct avec MaixPy. Ces expérimentations nous ont permis d'acquérir de précieuses connaissances sur les performances et les limitations de la carte Maixduino dans le domaine du traitement d'images en temps réel.

Malheureusement nous avons découvert que cette carte, bien qu'étant encore commercialisée, n'est plus suivie par ses développeurs, ce qui veut dire qu'il n'existe plus de mises à jour des micrologiciels. Ce qui peut s'avérer problématiques, dans notre cas avec un mauvais fonctionnement de la partie Esp32, le manque de bibliothèques et de datasheets.

À la suite de cette découverte nous nous sommes mis en quête de chercher une autre carte ou système offrant les mêmes possibilités mais avec un suivi de développement encore en cours.

2. La carte HuskyLens

Après plusieurs recherches nous avons trouvé une nouvelle solution pouvant remplacer la Maixduino. C'est la carte HuskyLens qui a retenu notre attention, qui est un module de vision artificielle compact conçu pour faciliter l'intégration de la technologie de vision par ordinateur dans les projets électroniques.



➤ Caractéristiques :

La carte HuskyLens est principalement utilisée pour la détection d'objets, la reconnaissance de formes et la communication avec d'autres dispositifs électroniques.

Elle est équipée d'une caméra et d'un processeur intégré capable de réaliser des tâches de traitement d'image en temps réel. Elle peut détecter et suivre des objets, reconnaître des formes prédéfinies et mesurer des distances avec précision.

L'interface utilisateur de la carte HuskyLens est conçue de manière intuitive, avec un écran directement intégré et des boutons de contrôle simple, ce qui permet une configuration et une prise en main facile. Elle peut également être contrôlée via une interface série ou I2C, offrant une flexibilité supplémentaire dans l'intégration avec d'autres microcontrôleurs comme une Arduino ou autre.

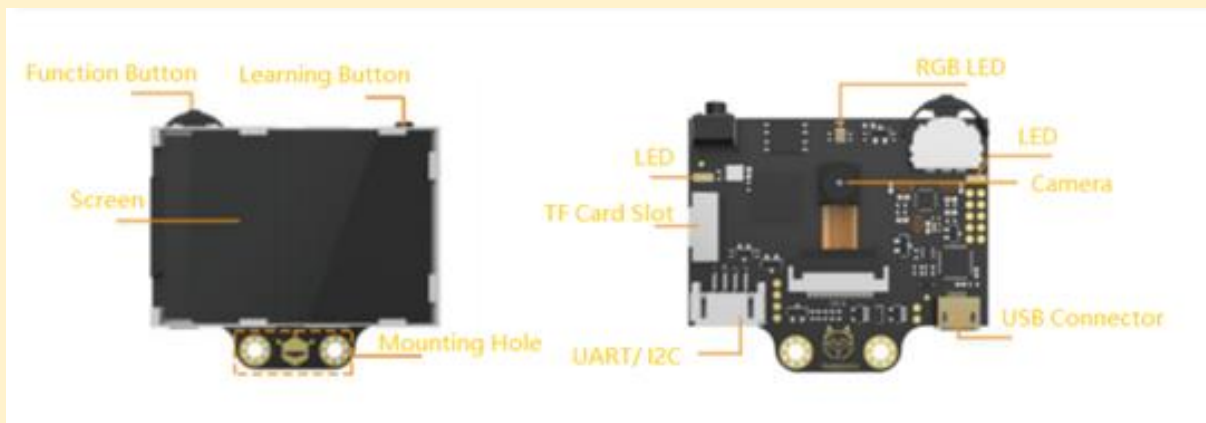


Figure n°5 : Illustration des éléments présents sur la carte Huskylens

Elle possède plusieurs modes de fonctionnement paramétrables directement via l'écran et les boutons de contrôles. En fonction du mode, on choisit de détecter des éléments en fonction de la couleur, de la forme, des visages, des lignes, ect...

La carte HuskyLens peut être utilisée dans beaucoup de projets électroniques qui ont besoin de la présence de camera ou de capteur de distance, tels que les robots autonomes, les systèmes de surveillance, les jeux interactifs et les dispositifs de réalité augmentée. Elle offre des fonctionnalités avancées de vision par ordinateur dans un format compact, ce qui la rend idéale pour les applications où l'espace et la facilité d'utilisation sont importants.

En résumé, la carte HuskyLens est un module de vision artificielle compact et convivial utilisé pour la détection d'objets, la reconnaissance de formes et la communication avec d'autres dispositifs électroniques. Elle offre en plus des fonctionnalités avancées de vision par ordinateur dans un format facile à utiliser, ce qui en fait un très bon choix pour notre projet.



➤ Méthode de programmation

Pour pouvoir communiquer avec cette carte, il faut la connecter via une autre carte de type Arduino en utilisant les protocoles de connexion série UART ou I2C. Pour ce faire il faut configurer le port série dans le code Arduino en spécifiant le baud rate approprié et initialiser la communication série, puis le Huskylens en envoyant une commande spécifique pour le préparer à la détection d'objets.

Pour lancer la détection d'objets il faut envoyer une commande appropriée, Huskylens analysera l'image de sa caméra intégrée et renverra les informations des blocs détectés. Pour lire les résultats de détection, il suffit d'utiliser la communication série pour récupérer les informations des blocs détectés.

Les informations récupérables sont le type de bloc, les coordonnées à l'écran, les dimensions et un ID unique pour chaque objet détecté. Ensuite, nous pourrions traiter les résultats en fonction des types de blocs détectés, en déclenchant des actions spécifiques ou en affichant des informations à l'écran. Enfin, pour faire une détection continue en temps réel, il faut utiliser la boucle principale d'Arduino pour le traitement et la détection.

➤ Travail effectué

Nous avons commencé par découvrir la carte Huskylens, en utilisant l'écran, les différents boutons et LEDs et surtout la caméra intégrée. Nous avons testé de nombreuses fonctionnalités, telles que la reconnaissance faciale, le suivi d'objets, la détection de couleurs et de lignes.

Cependant, nous avons rapidement identifié une première problématique : la difficulté de détecter plusieurs objets simultanément. Bien qu'il soit possible de savoir si les objets recherchés sont présents, il n'est pas possible de connaître leurs nombres exacts.

Pour évaluer les performances de la carte Huskylens, nous l'avons testée dans différents environnements de luminosité et sous différents angles de vue. Nous avons également varié les couleurs pour observer l'impact sur la détection. Nous avons constaté que la modification de la colorimétrie pouvait affecter la capacité de la caméra à détecter les objets, en fonction de la lumière reçue dans la pièce.



Figure n°6 : Screenshot du mode de reconnaissance par couleur

Nous avons également pris en compte la résolution de la caméra de la Huskylens et avons constaté qu'elle ne permettait pas de détecter des objets plus petits qu'un centimètre. Pour remédier à cette limitation, nous avons décidé de coupler la carte Huskylens avec une autre carte à microcontrôleur, la carte Heltec, similaire à une carte Arduino mais qui intègre un ESP32 et dispose d'un petit écran.

La communication entre la carte Huskylens et la carte Heltec peut se faire via une liaison série UART ou en utilisant le protocole I2C. Nous avons opté pour la liaison série, car elle est plus simple à mettre en place. Pour récupérer les informations de la caméra Huskylens, nous utilisons un algorithme qui nous fournit toutes les informations sur les blocs visibles à l'écran.

Dans un premier temps, la carte Heltec stocke et analyse les coordonnées pour s'assurer que cet objet n'a pas déjà été détecté. Ensuite, elle compare l'ID aux objets déjà détectés pour vérifier s'il s'agit bien de l'objet recherché.

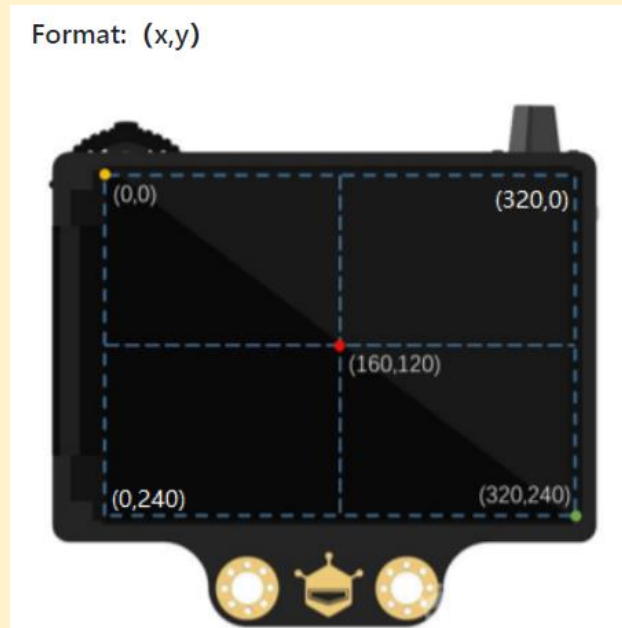


Figure n°7 : Illustration des coordonnées de la caméra sur l'écran Huskylens

À chaque nouvelle détection, une LED clignote pour alerter l'utilisateur. Nous supprimerons cette fonctionnalité ultérieurement, car elle n'est pas nécessaire lorsque la détection se fait de manière automatique.

Grace à cela, nous pouvons effectuer un comptage approximatif des objets présents dans le champ de vision de la caméra.

```
void printResult(MUSKYLENSResult result){  
  if (result.command == COMMAND_RETURN_BLOCK){  
    Serial.println(String() + F("Block:xCenter=") + result.xCenter + F(",yCenter=") + result.yCenter + F(",width=") + result.width + F(",height=") + result.height + F(",ID=") + result.ID);  
  }  
  else if (result.command == COMMAND_RETURN_ARROW){  
    Serial.println(String() + F("Arrow:Origin=") + result.xOrigin + F(",yOrigin=") + result.yOrigin + F(",xTarget=") + result.xTarget + F(",yTarget=") + result.yTarget + F(",ID=") + result.ID);  
  }  
  else{  
    Serial.println("Object unknown!");  
  }  
}
```

Figure n°8 : Illustration d'un programme d'affichage des coordonnées de détection d'un objet

Cette commande nous donne le type de bloc, les coordonnées à l'écran, les dimensions et un ID permettant de différencier les objets détectés. Nous avons développé un code qui utilise les exemples et tests que nous avons réalisés. Lorsqu'un objet appris par la Huskylens est détecté, toutes les informations nécessaires sont envoyées à la carte Heltec.

Les objets testés étaient des ronds rouges sur une feuille de papier uniforme, cela représente très sommairement des varroas sur le tapis roulant.

Avec cette carte-là, nous n'avons pas pu aller plus loin dans la démarche à cause du temps qu'il nous était réservé pour le projet.



II. Le tapis roulant

1. Solutions imaginées

Le but de ce système est qu'il soit intégré sous la ruche, mais surtout facile à transférer de l'une à l'autre tout en nécessitant très peu de maintenance.

Pour cela il est plus simple d'utiliser un gabarit de hausse de ruche car il viendrait se positionner directement sous le corps de la ruche sans difficulté et aux bonnes dimensions.

On viendrait faire des modifications pour pouvoir y intégrer un système de tapis roulant placer sous la grille de la ruche. Celui-ci permettrait de récupérer tous ce qui tombe à travers elle, ainsi les éléments qui chutent serait bien repartie sur la surface place du tapis pour permettre une meilleure qualité d'analyse.

Pour faire tourner le tapis, on utiliserait un moteur fixé en sortie de l'axe d'un des 2 rouleaux du tapis, commandé directement par la carte microcontrôleur choisie pour le comptage. Pour améliorer le rendement et réduire la consommation d'énergie, on ferait faire un tour de tapis avec des durées régulières et choisie à l'avance. Il est inutile de faire tourner le tapis en permanence, uniquement des échantillons sont utiles.

On agrandirait la hausse sur la partie arrière afin de pouvoir y intégrer le système de comptage avec caméra.

Pour réaliser ce comptage, il existe plusieurs possibilités en fonction des choix techniques et économiques sur le ou les caméras.

En effet, le cas le plus simple est si l'on arrivait à trouver une caméra permettant de capter l'entièreté de la largeur du tapis avec assez de précision pour détecter des varroas, ainsi on aurait besoin uniquement d'une rotation du tapis avec une caméra fixe placée au-dessus.

Pour le cas le plus couteux, on partirait sur le même type de système et de fonctionnement mais avec plusieurs caméras au lieu d'une si celle-ci ne couvrent pas entièrement la largeur du tapis.

Enfin la dernière option qui est sensiblement la plus économique, serait de partir du système couteux mais en utilisant qu'une seule caméra. Cela engrangerait donc un souci de lecture du tapis en entier, mais pour pallier cela, la caméra devrait être installer sur un système mobile pour que à chaque arrêt du tapis, celle-ci effectue un aller-retour en prenant plusieurs images afin de reconstruire toute la largeur du tapis et en faire l'analyse.



Ce choix ne peut donc se faire qu'après avoir cherché, analysé et testé des produits compatibles avec ce que l'on veut mettre en place.

2. Le travail effectué

La réalisation du tapis roulant ne peut se faire sans que la partie comptage fonctionne correctement en amont.

En effet, tant que l'on n'a pas décidé de choisir, le nombre de caméra, le type de système, la hauteur d'implantation des caméras, ect... ; il n'est pas possible de commencer la fabrication du système de tapis roulant et de fixation du projet complet.

Nous n'avons tout de même pas négligé cette partie, nous avons réfléchi sur une projection de ce que pourrait être ce système et travaillé dessus en réalisant une représentation en trois dimensions de ce que pourrait être la hausse de ruche modifiée.

Pour réaliser ces modèles 3D, nous avons utilisé le logiciel de modélisation SketchUp car c'est un des logiciels le plus simple et intuitif pour réaliser ce type de travail. Même avec sa version gratuite, il est possible de faire de belles modélisations.



Figure n°9 : Représentation 3d d'une ruche classique et de la ruche modifiée (vue arrière)



Figure n°10 : Représentation 3d d'une ruche classique et de la ruche modifiée (vue avant)

Sur ces représentations, on peut y voir le rajout à l'arrière de la ruche, avec un écran de visualisation pour le comptage.

Le système est complètement fermé afin de ne pas être perturbé par des éléments extérieurs et les intempéries. Une petite extension sur le côté contient la carte de commande ainsi que le moteur du tapis roulant. Un support permet de faire tenir les caméras (ici c'est l'exemple de 2 caméras fixes). Et puis le tapis roulant positionner entre 2 rouleaux permettant de le faire tourner.

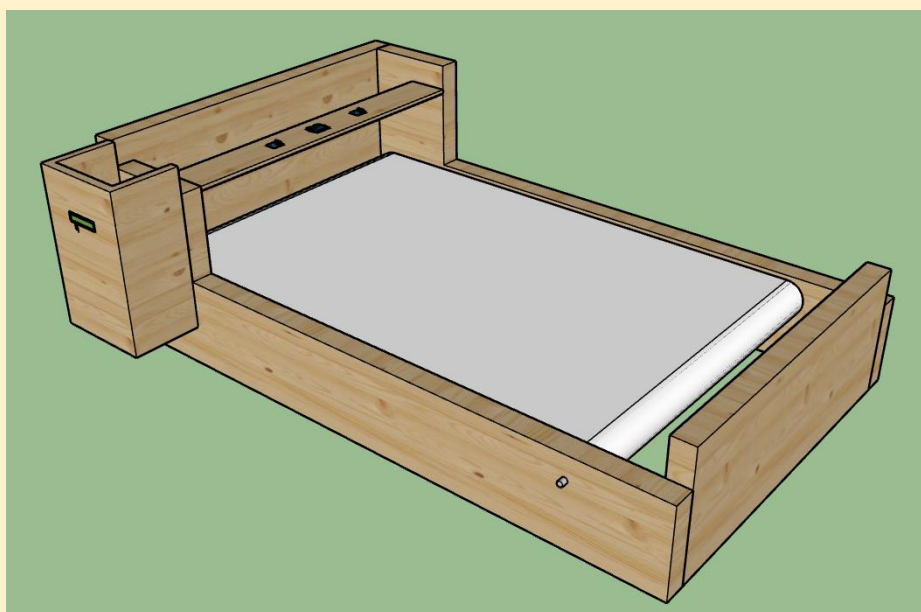


Figure n°11 : Représentation 3d du système sans la ruche



Evolution future

I. La détection

Pour continuer le projet, il faudrait travailler tout d'abord sur la partie caméra, en faisant le choix d'en avoir une ou plusieurs puis trouver le modèle avec la meilleure résolution pour pouvoir détecter les varroas sur le tapis.

Par la suite il faut travailler sur l'apprentissage de la détection des varroas et réussir à les compter. Partir de l'intelligence artificielle de la carte elle-même ou alors utiliser un outil extérieur en capacité de le faire.

Pour réussir à mieux les détecter, il serait peut-être intéressant de travailler sur l'éclairage du tapis avec une certaine couleur pour faire ressortir visuellement les parasites en contraste avec le reste.

II. Le tapis

Pour ce qui est du tapis roulant, il reste à le créer et le fabriquer en fonction des choix qui seront effectués pour la partie détection.



Conclusion

En conclusion, le projet de comptage de varroas morts représente une avancée significative dans la surveillance et la gestion des colonies d'abeilles. En développant un système autonome permettant de compter les varroas morts tombant de la ruche, l'apiculteur bénéficie d'un outil précieux pour évaluer l'infestation de varroas et prendre des mesures appropriées.

Ce projet a permis de débiter la conception et la mise en œuvre d'un système intégrable sous la ruche, comprenant un dispositif de récolte des éléments tombant de la ruche et un système de comptage des varroas morts. L'utilisation de la reconnaissance par image, associée à une caméra placée à la fin du dispositif de récolte, permet d'obtenir des résultats plutôt positifs pour un premier développement.

Il est à espérer que le commencement de ce projet servira de base pour de futures améliorations et développements de la surveillance des varroas et de la santé des abeilles. Grâce à des initiatives telles que celle-ci, nous pouvons travailler ensemble pour préserver ces précieuses pollinisatrices et maintenir l'équilibre écologique dont nous dépendons tous.

Ce fut un projet très intéressant à débiter malgré les nombreuses problématiques auxquelles nous avons dut faire face.

Nous remercions Mr Peter pour l'intérêt qu'il porte envers ses élèves ainsi que ses abeilles et espérons que cela pourra continuer le plus longtemps possible. La volonté de faire découvrir le travail de la nature avec l'apprentissage académique de l'ingénieur est quelque chose qui devrait être plus souvent pris en compte surtout dans le monde d'aujourd'hui.