

# Rapport de projet

## LASER





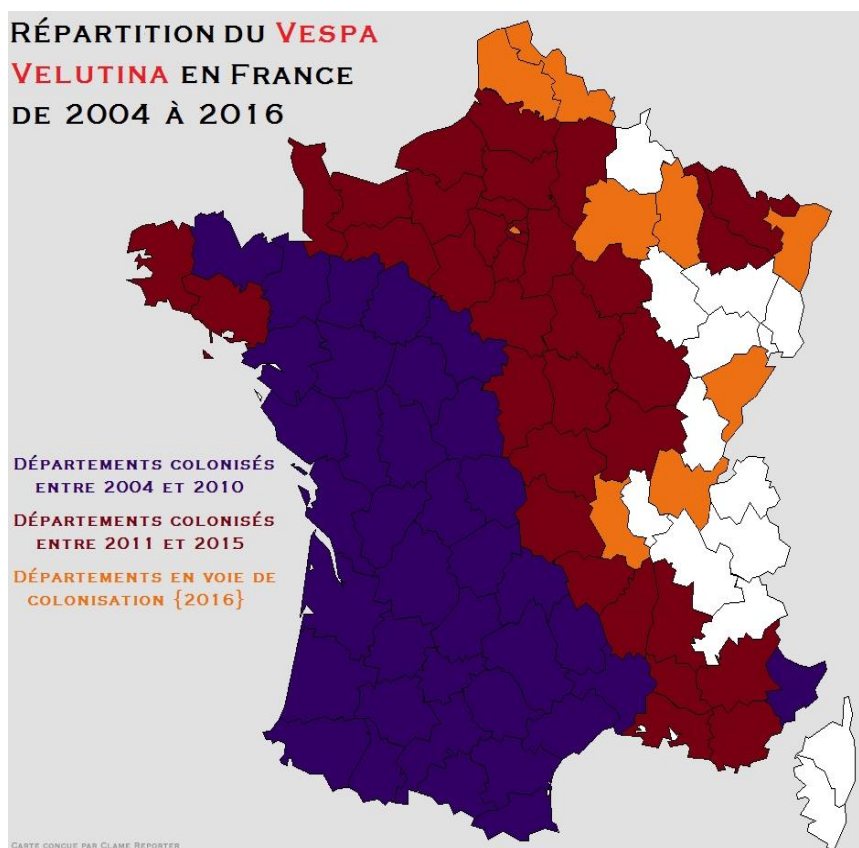
## Table des matières

- I. Contextualisation
  - 1. Objectif du projet
  - 2. Synopsis
  
- II. Détection
  - 1. Sonore
  - 2. Visuelle
  
- III. Mécanique
  - 1. Conversion image déplacement
  - 2. Mise en mouvement
  - 3. Asservissement de position
  
- IV. Laser
  - 1. Driver
  - 2. Diode Laser
  
- V. Carte électronique
  - 1. Choix des composants
  - 2. Schématique
  
- VI. Etat du projet
  - 1. Pistes non abouties
  - 2. Avancement global
  - 3. Conclusion personnelle

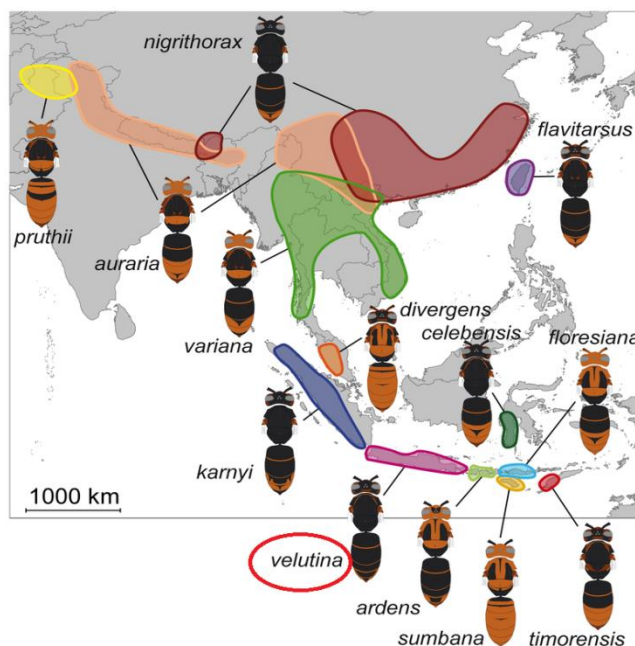
# I. Contextualisation

## 1. Objectif du projet

Depuis 2004, la France fait face à l'invasion d'une espèce importée par erreur : le frelon Asiatique (*Vespa Velutina*).



Comme son nom l'indique, cette espèce importée d'Asie du Sud-Est, plus précisément d'Indonésie, est un véritable fléau en Europe.



Contrairement à son cousin le frelon européen, le frelon Asiatique représente une grande menace pour les colonies d'abeilles, qui sont bien plus que de simples proies pour ce dernier.



Frelon européen



Frelon asiatique

En effet, le frelon Asiatique, pour nourrir ses larves, a besoin de chasser divers insectes. Les proies du frelon asiatique sont assez variées en milieu naturel ou agricole, mais sont majoritairement des abeilles dans les milieux plus urbanisés.

Ce prédateur cause beaucoup de dégâts dans les ruches des apiculteurs, si bien qu'il n'est pas rare que des colonies entières d'abeilles soient décimées.



Pour parvenir à ses fins, le frelon asiatique effectue un vol stationnaire devant l'entrée des ruches, et va attraper les abeilles qui lui passent à proximité.



Les abeilles sont des espèces fragiles qui subissent énormément l'activité humaine. Elles sont l'un des moteurs incontournables à la reproduction des végétaux, et sont aujourd'hui en déclin. Cette menace est un nouveau facteur d'affaiblissement des ruches contre lequel il faut lutter.

Contrairement à leurs cousines d'Asie, les abeilles d'Europe n'ont aucune défense contre le frelon asiatique. En Asie, la situation est différente, car les abeilles ont développé une technique de défense, qui consiste à tuer le frelon par hyperthermie.



Les abeilles d'Asie se regroupent autour du frelon et se mettent à vibrer pour faire augmenter la température.

Les abeilles sont naturellement plus résistantes à la chaleur que les frelons, ce qui permet, malgré quelques sacrifices, une défense efficace contre ce prédateur.

Aujourd'hui des solutions existent pour lutter contre le frelon asiatique. Certaines ont pour but de protéger les ruches sans tuer le frelon, comme les "muselières", qui permettent aux abeilles de sortir de la ruche, mais pas aux frelons d'y entrer, limitant les dégâts fait aux colonies d'abeilles :



Des pièges à frelons qui consistent à les attirer avec du liquide sucré par exemple, pour les faire tomber dans un entonnoir existent également.

Mais des méthodes plus naturelles existent également. Dans son habitat naturel en Asie, la *vespa velutina* a de nombreux prédateurs, la population est donc constamment régulée. A contrario, en Europe il y a peu de prédateur naturel du frelon asiatique. Parmi eux on compte par exemple certaines espèces de poules, qui vont pouvoir être élevées à proximité des ruches.

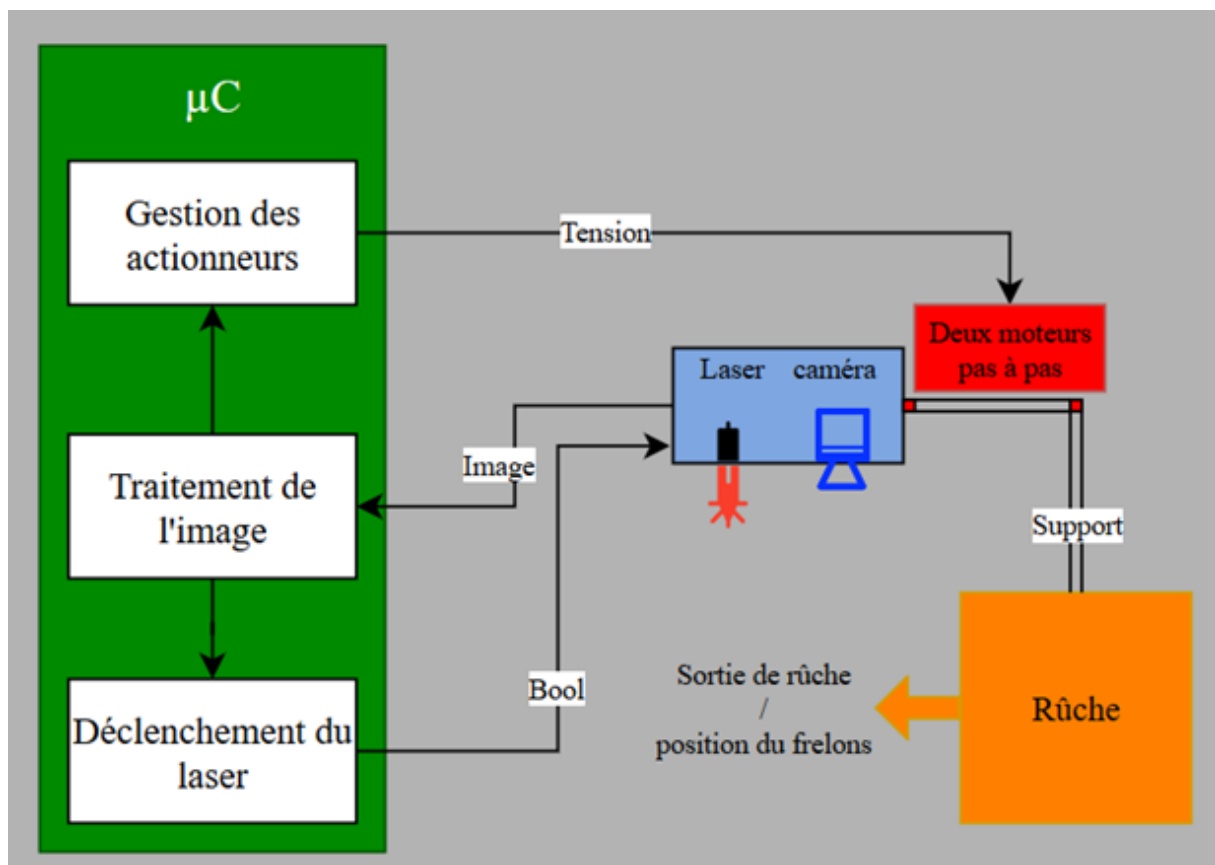
Bien que le moyen le plus efficace reste la destruction des nids de frelons, d'autres pistes restent à explorer.

Notre projet consiste donc à lutter contre le frelon asiatique de manière drastique, pour protéger les ruches et leurs abeilles.

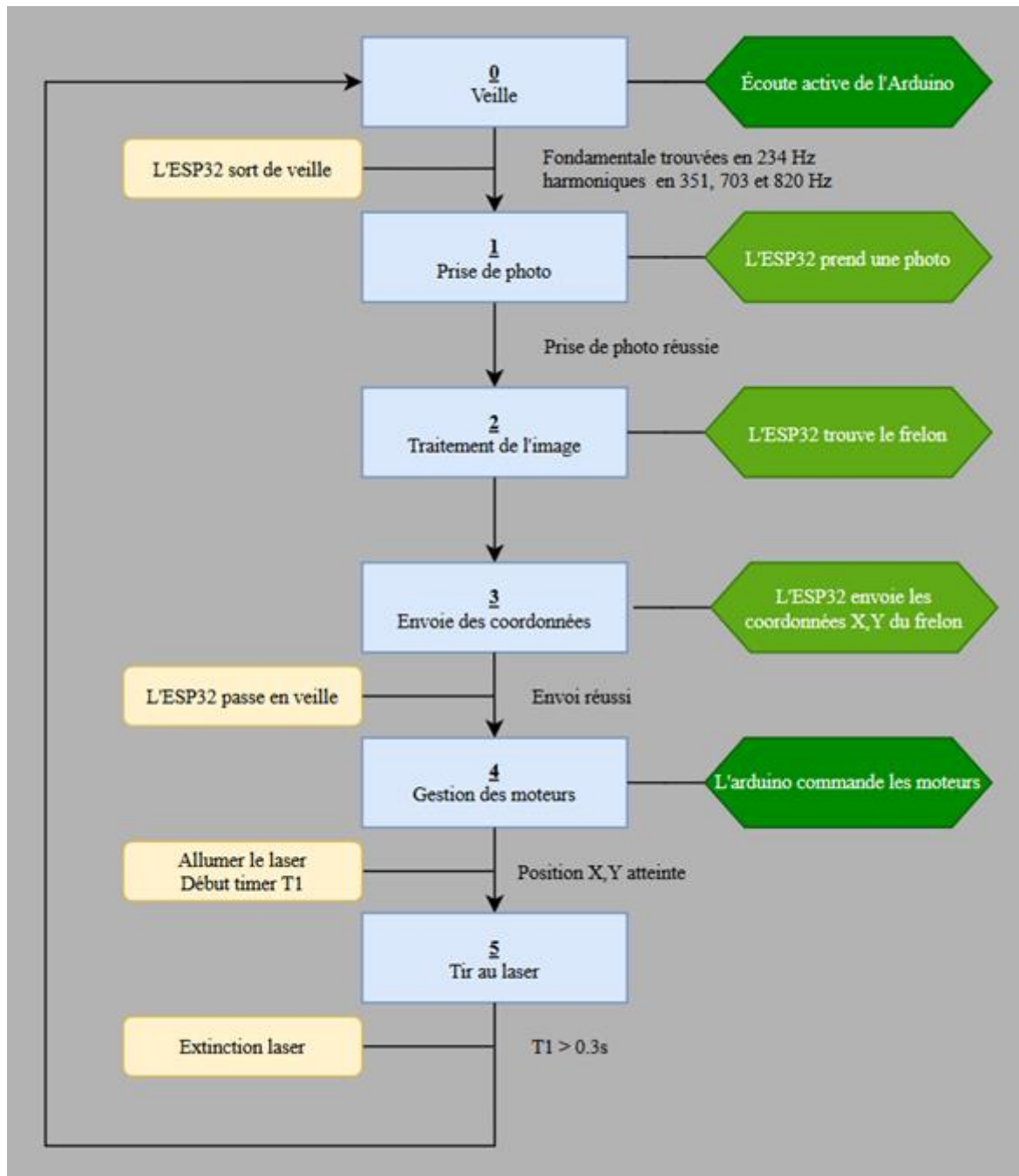
## 2. Synopsis

Dans la suite de ce rapport nous allons vous présenter le fonctionnement étapes par étapes de notre système en passant par la détection du frelon, la mécanique utilisée et le boîtier laser choisis ainsi que les solutions d'alimentations globales. Pour cela nous allons faire le détail des différents choix techniques, des tests que nous avons effectués ainsi que leurs résultats.

Nous terminerons ce rapport par une conclusion qui englobera l'état d'avancement du projet et ce qu'il reste à faire dessus pour qu'il soit opérationnel. Nous compléterons avec une conclusion personnelle du groupe pour mettre en avant ce que ce travail nous a apporté en termes de connaissances que ce soit techniques ou managériales.







## II. Détection

### 1. Sonore

Pour réduire la consommation d'énergie quand les abeilles ne sont pas attaquées par les frelons, le système est gardé en veille et réveillé seulement lorsque le microphone détecte le son d'un frelon.

L'algorithme de détection acoustique fonctionne en permanence et il écoute la présence des fréquences émises par un frelon en vol.

Pour capter le son émis par les frelons, nous avons utilisé un module microphone qui intègre le MAX9814. Avec cette solution on peut augmenter le gain du microphone au besoin.

Le signal amplifié est ensuite envoyé vers un Arduino Micro ou la conversion analogique numérique, puis la FFT sur le signal échantillonné s'effectuent. Suite à ce traitement, nous obtenons un tableau qui contient les amplitudes à différentes fréquences du signal. C'est ce qu'on appelle le spectre des raies en fréquence. Une fois les fréquences récupérées, nous pouvons isoler les raies du spectre qui nous intéressent, notamment le fondamental et les 3 harmoniques contenues dans le bourdonnement d'un frelon.



Module microphone

La prise d'écran ci-dessous montre l'algorithme implémenté pour détecter ces 4 fréquences. Celui-ci détecte si on a des pics centrés sur les fréquences intéressantes (234Hz, 375Hz, 703Hz, 820Hz) et ensuite vérifie que la somme des 3 harmoniques est supérieure au fondamental. Cette vérification permet d'isoler le bourdonnement du frelon du reste des sons présents à proximité de la ruche et donc d'avoir moins de fausses détections. Si la comparaison est vraie alors il y a présence du frelon.

```
// on cherche si le pic principal(fondamental) = 234Hz +- 10%
if (spectrum[3] > (0.66* (spectrum[2]+ spectrum[4]))) {
  // les valeurs en 5, 9 et 11 sont-elles des pics ?
  if (spectrum[5] > (0.66* (spectrum[4]+ spectrum[6]))) // pic en 375Hz en SAMPLES 128 (351Hz en SAMPLES 2048)
  {
    if (spectrum[9] > (0.66* (spectrum[8]+ spectrum[10]))) // pic en 675Hz en SAMPLES 128 (703Hz en SAMPLES 2048)
    {
      if (spectrum[11] > (0.66* (spectrum[10]+ spectrum[12]))) // pic en 825Hz en SAMPLES 128 (820Hz en SAMPLES 2048)
      {
        int somme_pics = spectrum[5]+spectrum[9]+spectrum[11];
        if(somme_pics > spectrum[3])
        {
          detect = true;
        }
      }
    }
  }
}
```

Implémentation de l'algorithme de détection sonore

Suite aux tests que nous avons faits, nous nous sommes aperçus que l'algorithme n'était pas assez sensible, c'est-à-dire qu'il fallait augmenter la fréquence des harmoniques de beaucoup avant qu'il y ait détection. Cela peut se corriger en rajoutant un coefficient devant la somme des harmoniques.

Un autre souci est la résolution du spectre qui est limité par la taille du buffer d'échantillonnage de l'Arduino Micro. Celui-ci est limité à 128 échantillons, ce qui nous laisse avec 64 raies en respectant le théorème de Shannon. On peut palier à ce problème en diminuant la fréquence d'échantillonnage à deux fois la fréquence maximum à détecter, c'est-à-dire  $825 \times 2 = 1650\text{Hz}$ .

Il faudrait aussi mettre en place un filtre passe-bas à l'entrée du système pour éviter le repliement des hautes fréquences dans la bande passante.

## 2. Visuelle

Une fois la présence d'un frelon asiatique confirmée par notre système de détection sonore, nous avons pour objectif de faire prendre une photo à notre microcontrôleur puis d'y identifier le frelon puis envoyer la position du dit frelon à l'Arduino s'occupant de la « visée ». Cette image est ensuite enregistrée sur une carte SD avec potentiellement les informations quant à la position du frelon dans le nom du fichier. Ex : '154\_258\_0001.jpg'

La caméra de l'ESP32-CAM est de suffisante pour prendre cette photo, nous n'avons pas la nécessité d'une qualité trop importante puisque ça ne ferait qu'allonger les temps de calculs ainsi qu'augmenter la place.

Nous basons notre détection sur trois choses :

- Nous placerons une surface blanche qui fera office de fond à notre image pour la détection. Le fond sera donc toujours blanc.
- Les frelons asiatiques sont plus foncés que les abeilles, on peut estimer une fourchette de niveau de gris dans laquelle les frelons seront compris, à préciser par la suite à l'aide d'une photorésistance.
- Lors de la chasse, les frelons asiatiques se positionnent en vol stationnaire devant la ruche, donc à une distance prévisible. On en déduit donc que la taille en pixel du frelon sur l'image sera comprise dans une fourchette qu'on peut mesurer.

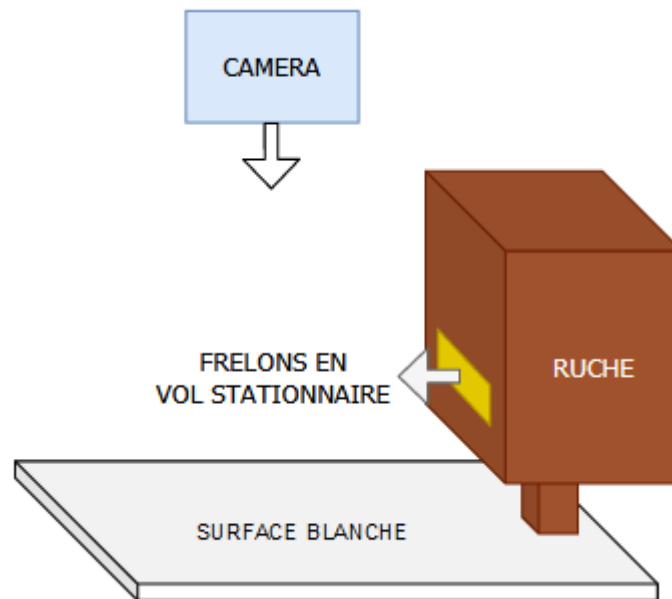
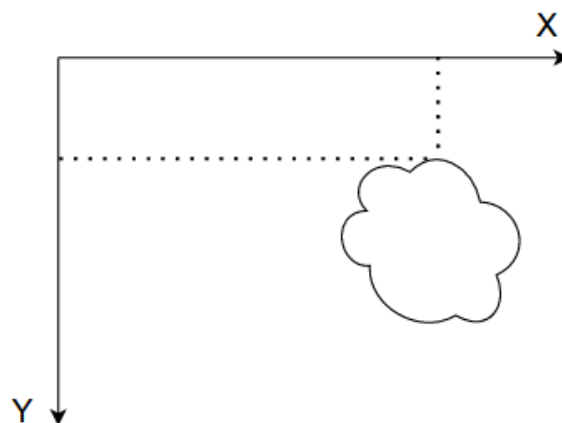


Schéma de l'installation

Nous prenons donc une photographie en niveaux de gris : 1 pixel = 1 octet, 0 : noir, 255 : blanc. Puis nous l'analysons tous les  $N$  pixels,  $N$  étant défini par l'utilisateur. Une fois « arrivé » sur un pixel d'un niveau de gris compris dans la fourchette définie précédemment, nous gardons en mémoire la position en  $X$  et  $Y$  et analysons les pixels voisins, puis ceux voisins de ceux-ci. Nous incrémentons un compteur des pixels constituant cette tâche de compter les pixels composant la tâche. Si la tâche a une taille comprise dans la fourchette de tailles, nous pouvons supposer que c'est notre frelon.

Nous ne pouvons qu'approximer le centre de cette tâche. Ex :

$$Taille\_tache = 300 ; Position\_x = 100 ; Position\_y = 150$$



On sait que  $Taille\_tache \approx \pi * rayon^2$  donc  $rayon = \sqrt{Taille\_tache / \pi}$

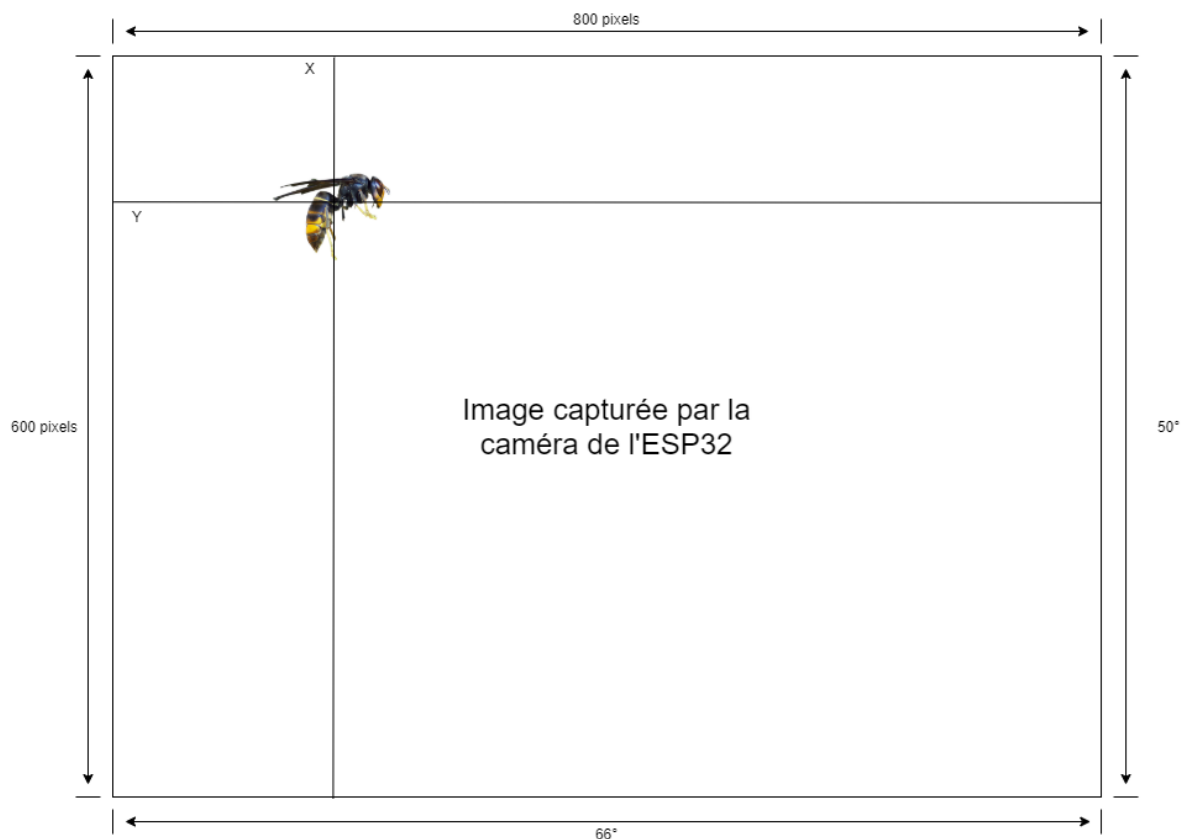
On estime le centre de la tache en  $X = Position_x = 100$  et  $Y = Position_y + rayon = 159$

Une fois cette position ( $X; Y$ ) trouvée, nous l'envoyons en UART à l'Arduino.

### III. Mécanique

#### 1. Conversion image déplacement

Lorsque les capteurs sonores et visuels ont rempli leurs tâches, une information est transmise au microcontrôleur en charge de la motricité, dans notre cas un simple Arduino Nano. Ce dernier reçoit une position en X et en Y dans une trame, qui lui indique où se situe le frelon asiatique dans le champ de vision de notre caméra (en considérant l'origine au coin supérieur gauche de l'image), elle est capable de prendre une image sur un angle de 50° en vertical et 66° en horizontal. On peut voir ci-dessous un schéma simplifié de la situation :



L'unité des informations reçues est le pixel, il faut donc convertir cette donnée en un angle qui va être plus facilement utilisable pour contrôler les différents actionneurs, nous avons donc calculé deux rapports, un par axe, nous permettant de convertir un nombre de pixel en un angle.

Axe X :

$$Rx = \frac{66^\circ}{800px} = 0.0825^\circ/px$$

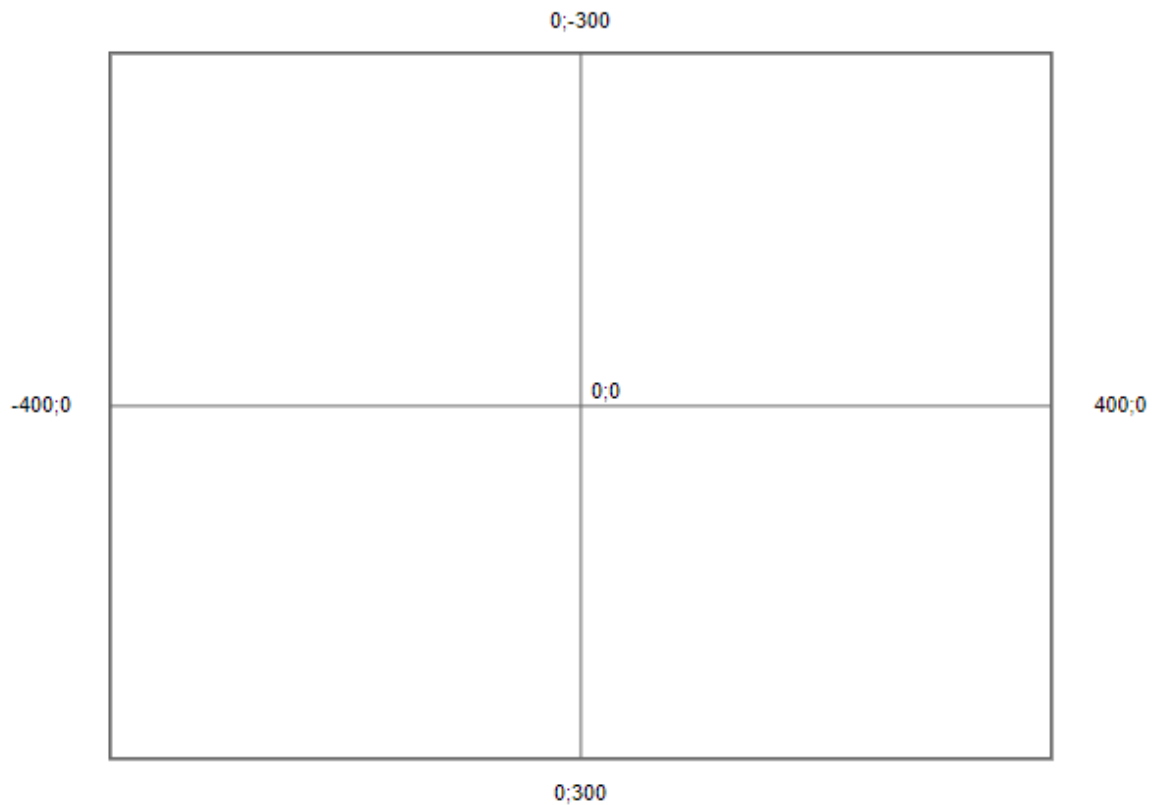
Axe Y :

$$Ry = \frac{50^\circ}{600px} = 0.08333^\circ/px$$



Nous pouvons observer que les deux rapports calculés sont différents, il va donc falloir traiter les axes indépendamment.

Lorsque la trame est reçue nous récupérons les deux positions, nous soustrayons à chacune la moitié du nombre de pixel total sur l'axe concerné pour obtenir une position par rapport au centre.



Puis en multipliant ces valeurs par nos rapports calculés précédemment, nous obtenons un angle utilisable plus facilement.

#### Exemple :

Si l'on reçoit dans notre trame [500][150] :

nous allons en déduire que notre frelon se trouve à la position [100][-150] par rapport au centre, ce qui va nous permettre d'obtenir des valeurs d'erreurs d'angles de

- En X :  $100 * 0.0825 = +8.25^\circ$
- En Y :  $-150 * 0.08333 = -12.4995^\circ$

Nous n'avons plus qu'à gérer un déplacement sur l'axe X de  $8.25^\circ$  et un déplacement sur l'axe Y de  $-12.4995^\circ$ .

## 2. Mise en mouvement

- Nous nous sommes premièrement mis d'accord sur un cahier des charges :
- Être capable de viser un point donné situé dans sur un plan en 2D
- Pouvoir supporter le poids du système
- Avoir une précision minimale de 5 mm à 1 m
- Posséder la capacité d'effectuer un demi-tour en moins de 200ms

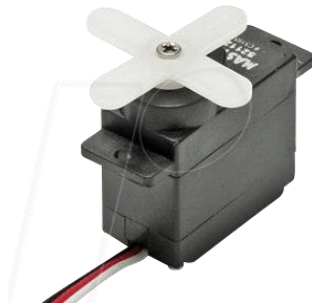
Dans le but de mettre en mouvement notre système Laser + caméra, un système de tourelle deux axes semble être le choix le plus pertinent, nous pouvons observer ce principe dans différents domaines, armement ou vidéosurveillance.



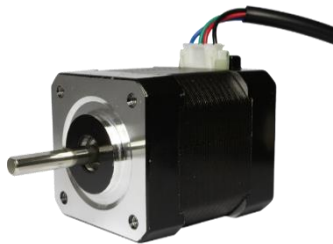
Aussi plusieurs options s'offraient à nous pour créer un mouvement rotatif à partir d'un système alimenté par une batterie (exclusions des moteurs à courant alternatif) :



Les moteurs à courant continu (MCC)



Les servos moteurs



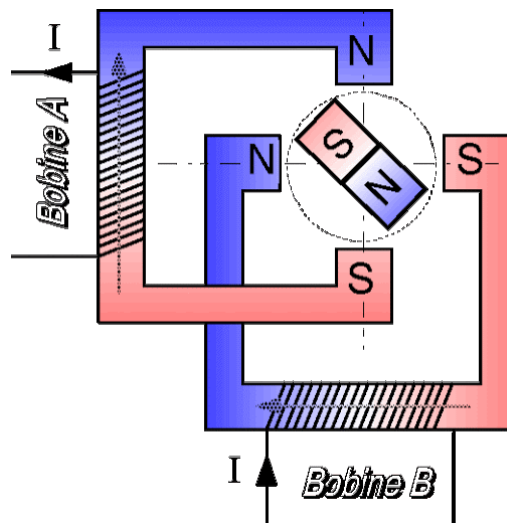
Les moteurs pas à pas

Tous ont leurs avantages et leurs inconvénients, les premiers sont les plus simples à mettre en œuvre, car une simple tension permet de les mettre en mouvement mais naturellement ils tournent à des vitesses très élevées et ont un couple très faible, un assemblage motoréducteur permet de pallier ce problème. Cependant on ne peut récupérer une information concernant leur mouvement qu'en ajoutant un encodeur sur l'axe moteur, qu'il va falloir connecter à notre microcontrôleur, ce choix n'est donc pas envisageable car la complexité du système est trop importante pour notre utilisation. Notons aussi qu'ils sont généralement utilisés pour des applications tirant profit de leur capacité à tourner à l'infini sans contrainte mécanique.

Les seconds sont eux limités dans leur rotation, le plus souvent à environ  $270^\circ$  mais cela dépend des modèles, aussi ils ont une consommation très faible, s'alimentent en 5V et une simple PWM (Pulse Width Modulation, modulation de largeur d'impulsion) permet de les faire se mouvoir. Un de leurs points positifs est qu'ils sont les seuls à avoir un couple résistant tel que sans alimentation ils restent dans leur position (sauf évidemment si le poids du système est trop important) mais cette caractéristique s'avère intéressante pour un système embarqué dont la faible consommation est un objectif visé. Cependant ils sont peu précis et leur utilisation est plutôt destinée à de l'hobbisme ou des jouets (modélisme) car ils ne sont pas très précis et leur fonctionnement se détériore au cours du temps, pour une même commande appliquée, le résultat va différer en fonction de son état.

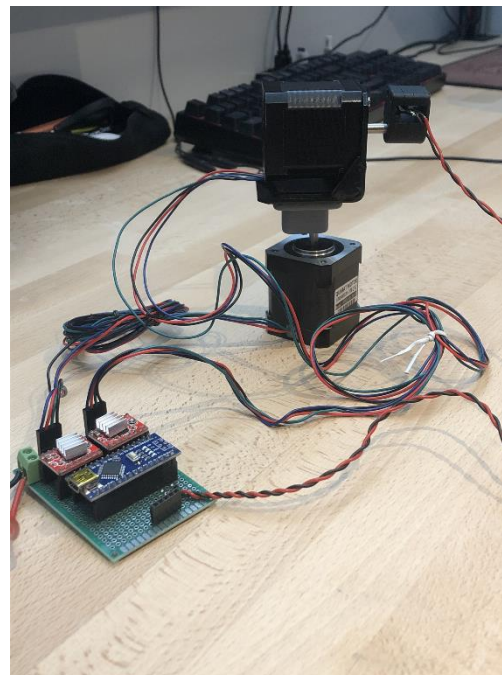
Le dernier mécanisme est le plus précis mais également le plus lourd à mettre en place, un driver doit être utilisé pour faire tourner un moteur pas-à-pas car ils disposent le plus souvent de 4 fils reliés à 2 bobines, dont il va falloir moduler les courants pour créer des champs magnétiques tournants et ainsi une rotation. Ils sont alimentés entre 8 et 15V et sont très souvent utilisés pour des machines nécessitant de la robustesse et de la précision telle

que de CNC ou des imprimantes 3D. Vous pouvez apprécier ci-dessous un schéma simplifié de notre mécanisme.



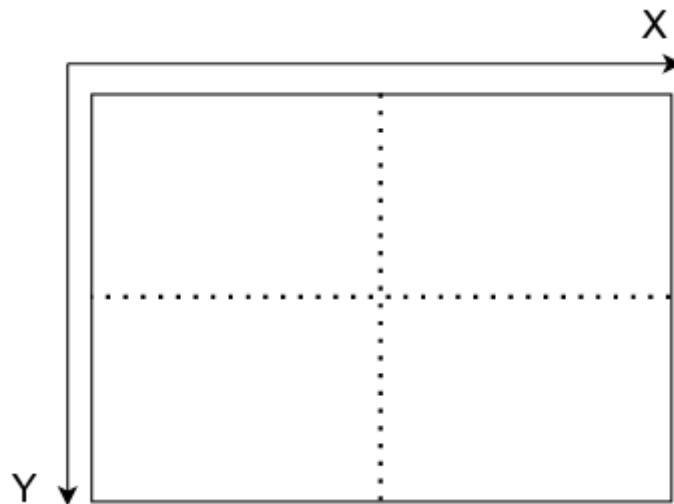
Notre attention a été retenue par les servomoteurs et les moteurs pas-à-pas car leur précision et leur couple important en font des actionneurs tout à fait intéressants pour notre système. Cependant la faible consommation des premiers et la durabilité des seconds constituent deux caractéristiques qui ne nous permettent pas de les départager sans observer le système réel en fonctionnement.

Nous avons donc construit 2 tourelles différentes grâce à ces technologies, la première constituée de deux MG 996R et l'autre comprenant deux QUIMAT 17HD48002H-22B. Vous pouvez observer que celle de droite est surmontée du laser final tandis que celle de gauche comprend la première diode utilisée dans nos tests.



### 3. Asservissement de position

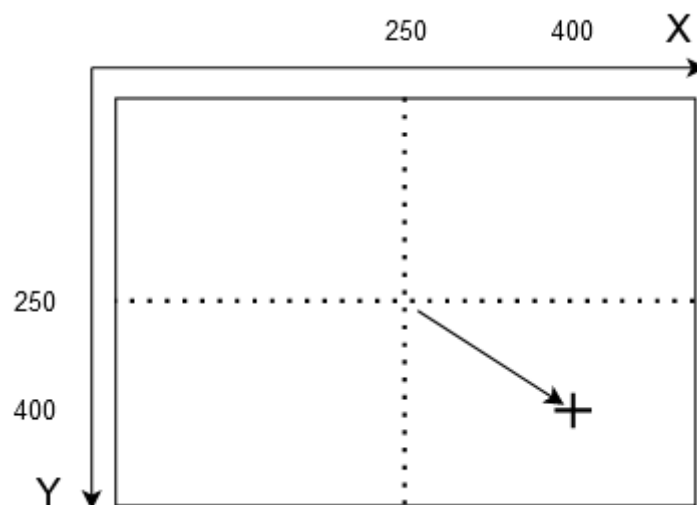
La caméra et le laser seront au plus proche l'un de l'autre, positionnés sur deux moteurs. Le laser pointe vers le centre de l'image. On peut se représenter les choses comme ci-dessous :



L'intersection des pointillés représente le centre de l'image ainsi que le point sur lequel pointe le laser. Les moteurs permettent de faire tourner l'appareil et le laser sur l'axe X pour l'un et Y pour l'autre.

Le centre de l'image a pour coordonnées la largeur et la longueur divisée par deux, si l'image fait 500x500 pixels, alors le centre sera en (250 ; 250).

Notre microcontrôleur reçoit de la part de l'ESP32-CAM les coordonnées du point sur lequel il nous fait centrer la caméra, on veut donc comparer ces coordonnées par rapport à celles du centre de l'image afin d'asservir la position des moteurs affluents sur les axes X et Y.

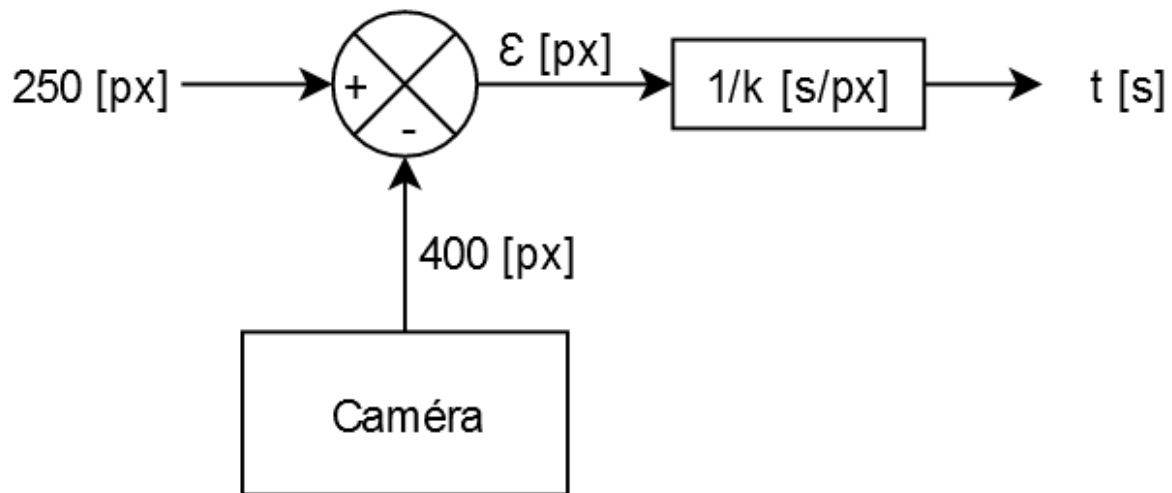




Une façon simple de répondre à cette problématique est de déduire, en fonction de la valeur du centre et des coordonnées visées, les directions dans lesquels doivent tourner les moteurs ainsi que le temps pendant lequel on doit les faire tourner, en considérant qu'on a une vitesse de rotation fixe.

On a comme informations le nombre de pixels par degrés sur l'image (nombre de pixels divisé par le champ de vision en degrés) ainsi que la vitesse de rotation en degrés par seconde du moteur. On déduit de ces deux informations la valeur de  $k$ , en pixels par secondes.

$$K \text{ [px/s]} = \text{Px/deg} * \text{deg/s}$$



- Axe X :
  - $\epsilon < 0$  : le frelon est sur la droite de l'image
  - $\epsilon > 0$  : le frelon est sur la gauche de l'image
- Axe Y :
  - $\epsilon < 0$  : le frelon est en bas de l'image
  - $\epsilon > 0$  : le frelon est en haut de l'image

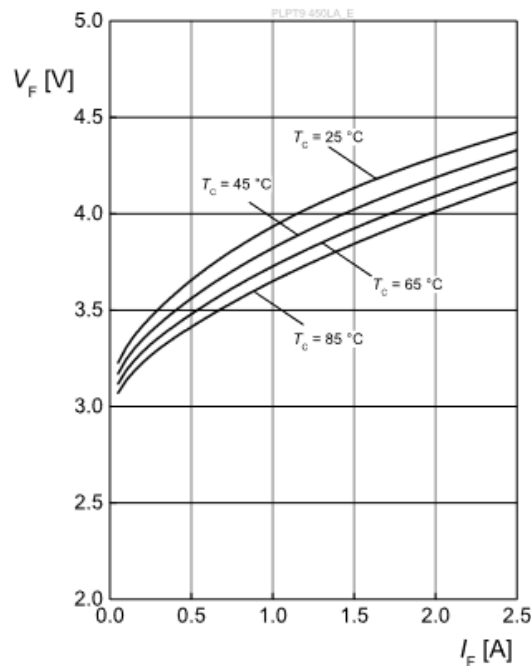
## IV. Laser

### 1. Driver

Le laser est très sensible aux variations de tension. En effet il se comporte beaucoup comme une diode. C'est-à-dire qu'une très petite variation de la tension d'alimentation fera varier le courant brutalement. Cette propriété s'observe sur le graphe ci-dessous, qui correspond à la caractéristique courant tension de la diode que nous avons choisie pour notre système. Pour cette raison, nous avons choisi de piloter le laser en courant et non pas en tension, limitant effectivement l'ampérage qui le traverse.

**Forward Voltage** 5), 6)

$$V_F = f(I_F)$$



Caractéristique courant-tension de la diode PLPT9

Pour le driver, nous avons opté pour le NCL30160. C'est un régulateur abaisseur de tension de type « Buck ». Il permet un réglage précis du courant dans la charge et il a été conçu spécialement pour le pilotage des DEL haute puissance, donc il est adapté pour notre utilisation. Ci-dessous un schéma du circuit utilisé.

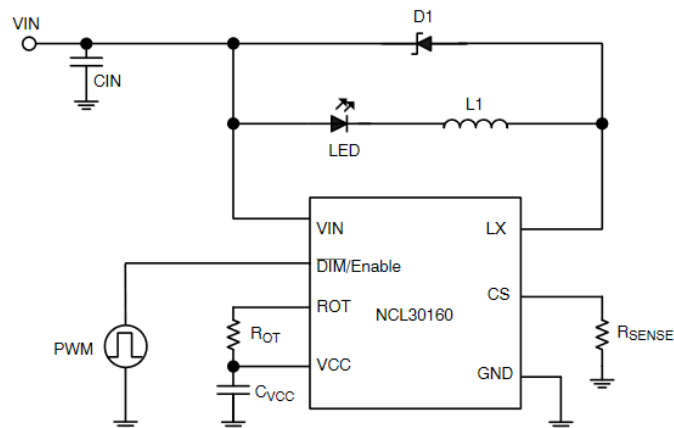


Schéma du circuit d'alimentation du laser

La résistance  $R_{sense}$  permet de savoir le courant circulant dans le laser pendant que le transistor du NCL est conduisant, et la résistance  $R_{ot}$  permet de fixer le temps d'extinction du transistor de puissance. En effet le circuit n'a pas de moyen de savoir le courant circulant dans le laser lorsque le transistor est éteint, mais nous pouvons fixer ce temps d'extinction en sachant que la vitesse de diminution du courant va être fixe par la valeur de l'inductance  $L1$ .

Nous avons dû faire plusieurs tests pour retrouver les bonnes valeurs de ces résistances. Pour cela, nous avons fait un calculateur sous Excel pour faciliter et accélérer les tests. Plus bas une capture d'écran de celui-ci.

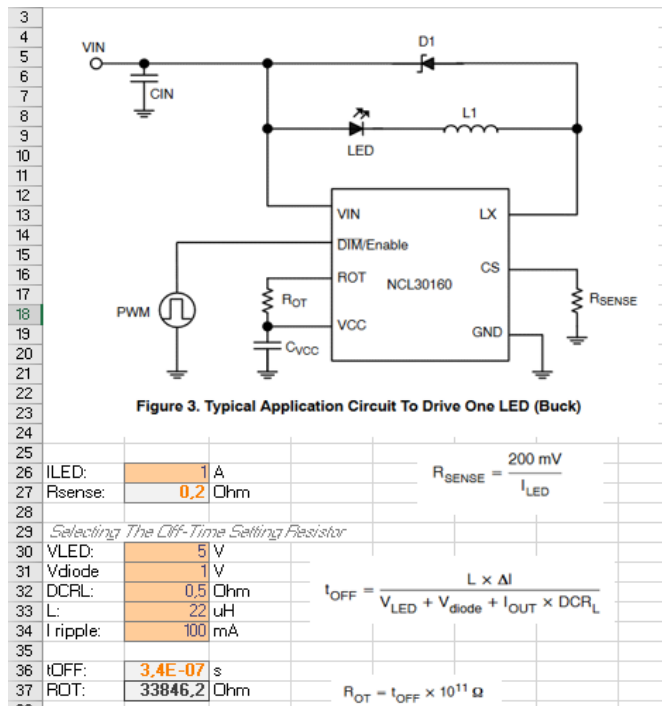


Illustration du calculateur de composants

Un autre point positif de cette puce est l'entrée DIM/Enable, qui permet de diminuer l'intensité du laser avec un signal PWM. Cela nous a permis de faire des tests à plus faible puissance et de savoir précisément la puissance requise pour éliminer les frelons.

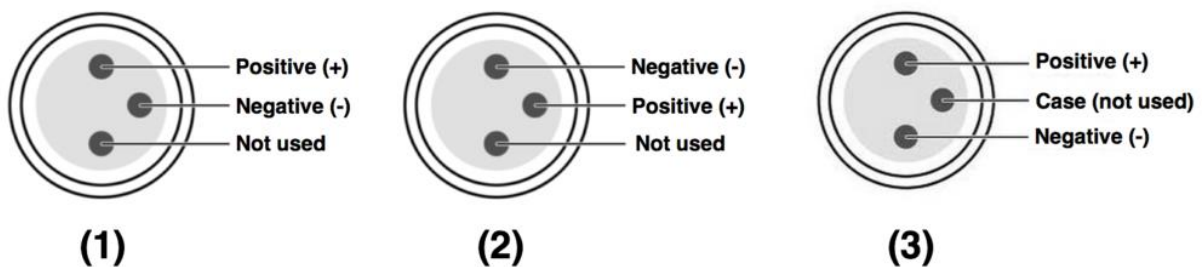
## 2. Diode Laser

### A) Premières expérimentations

Avant de nous lancer dans les recherches de quelle diode serait la plus adaptée pour notre projet nous avons dans un premier temps décidé de manipuler des diodes lasers banales afin de nous familiariser avec ce composant car personnes du groupe n'était à l'aise avec. Ces premières diodes étaient des diodes à lumières rouges de longueur d'onde 800 nm (celles que l'on retrouve dans les télécommandes, les lecteurs DVD, consoles de jeux-vidéos etc).



Nous nous sommes vite rendus comptes qu'il n'y avait pas une configuration universelle de celles-ci mais bien plusieurs comme le montre l'image ci-dessous.



Cette phase de tests primaire n'a donc pas été concluante pour plusieurs raisons. D'abord du fait des différents types de branchements existants, nous avons brûlés quelques diodes avant de réussir à en faire fonctionner correctement. Dans un second temps, la puissance émise et la température du faisceau lumineux de la diode étaient beaucoup trop faibles pour notre futur usage, et pour finir la lumière n'était pas collimée ce qui traduisait un manque de précision et une perte de puissance significative.

## B) Choix de la diode

Après donc cette première phase de test nous permettant de nous habituer à manipuler des diodes lasers, nous avons fait des recherches pour déterminer quelles étaient les caractéristiques nécessaires afin de venir à bout des frelons asiatiques. Nous avons donc pu déterminer que notre diode devait fournir une lumière bleue car elle s'avérait plus efficace et plus puissante que la lumière rouge, de plus la température du faisceau lumineux devait atteindre au minimum 47 °C car c'est la température létale minimum des frelons asiatiques.

D'après ces deux contraintes principales, notre choix s'est porté sur une diode de la marque OSRAM (référence en annexe) comportant ces caractéristiques :

- Lumière bleue de longueur d'onde 447 nm.
- Courant de fonctionnement : 2.5A.
- Tension de fonctionnement : 4.3V.
- Puissance de sortie max : 3W.
- Température de fonctionnement max : 85 °C.



Cette diode comporte donc toutes les caractéristiques nécessaires pour neutraliser les frelons asiatiques.

### C) Laser final

Maintenant que nous avons la diode, il nous reste à régler les problèmes de collimation et de gestion de la chaleur.

Dans un premier temps, pour régler la collimation nous décidons de partir sur un boîtier optique réglable composé de lentilles de verres. Ces lentilles vont nous permettre de focaliser notre faisceau lumineux en un seul point afin d'être le plus précis possible et de limiter la perte de puissance. Dans un second temps, pour pallier la gestion de la chaleur nous avons choisis de prendre un dissipateur thermique ce qui nous évitera que lorsque notre système est en fonctionnement, la diode soit très chaude et au contact de composant plastiques ce qui pourrait détériorer notre projet et donc le rendre inefficace.





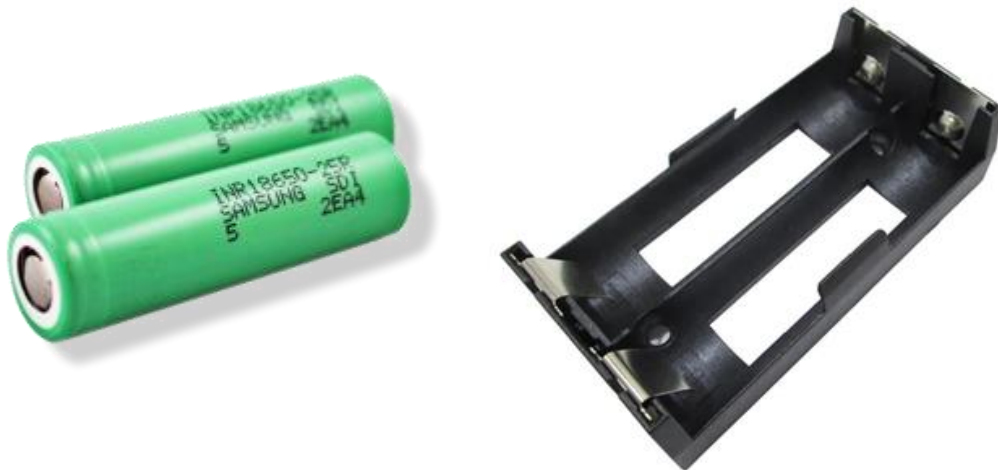
Une fois le tout monté nous avons pu passer à une deuxième phase de test en situation car nous avons comme cobayes des frelons asiatiques déjà morts. Ces tests ont été très concluants car nous avons réussi à faire brûler les frelons dans un temps qui nous était satisfaisant (l'ordre de quelques secondes).

## V. Carte électronique

### 1. Choix des composants

- Les composants électroniques se divisent en plusieurs familles :
- La gestion de l’Alimentation
- Les capteurs et la partie intelligente
- Les actionneurs

Pour les batteries, nous avons choisi des accumulateurs Li-Ion 18650, qui sont des batteries cylindriques de 18mm de diamètre et de 65mm de hauteur, nous les avons choisis car ils sont très communs, la technologie chimique est maîtrisée et de nombreux composants pour les charger ou les maintenir existent. Nous avons fait le choix de les maintenir avec des berceaux doubles, qui nous permettent si on les relie en série d’obtenir une tension de 7.4V nominale.



Des régulateurs de tension Buck (abaisseur) AP63300WU-7 sont utilisés pour obtenir les différentes tensions nécessaires 3.3V et 5V, il nous manquait cependant un régulateur Boost (rehausseur) pour obtenir le 12V nécessaire aux moteurs pas-à-pas.

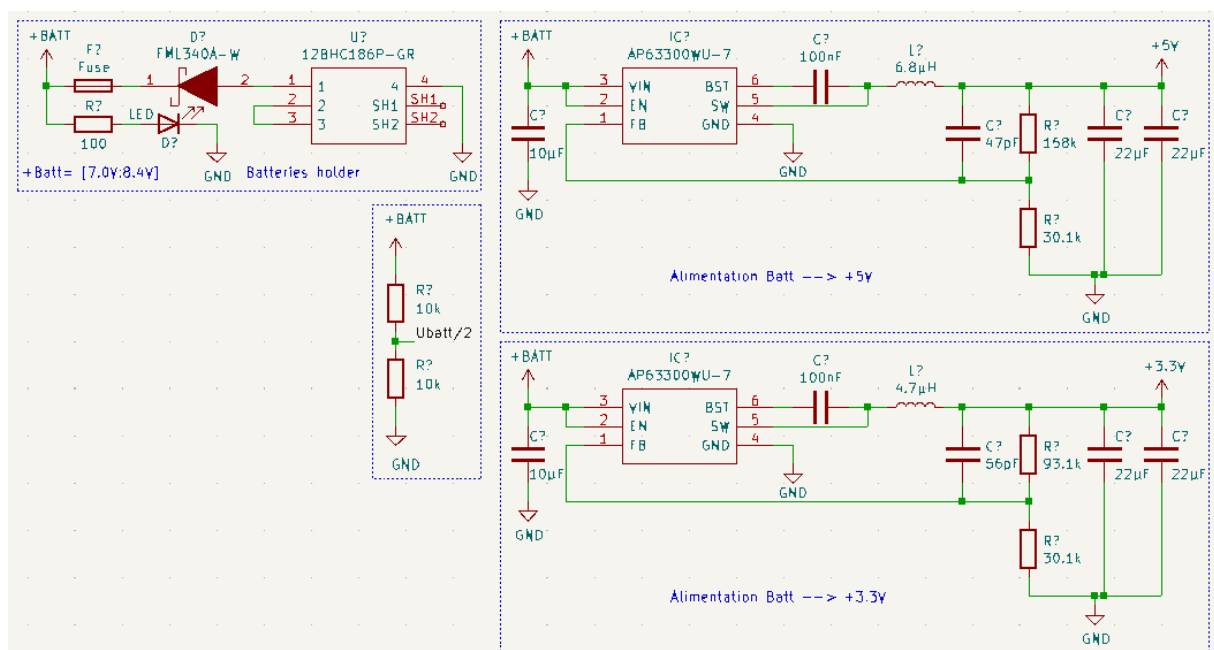
Deux microcontrôleurs sont utilisés, un Arduino Nano pour la partie motricité et un ESP32 Cam pour la partie détection d’image, une liaison UART (Universal Asynchronous Receiver Transmitter) assure la communication entre ces deux organes. Concernant les capteurs nous retrouvons, la caméra de l’ESP32, qui est reliée directement sans que l’on doive le faire, un micro qui est constitué d’une carte que l’on connecte grâce à un header 5 pins, de 2.54mm de pas, et il nous manquait dernièrement une photorésistance utile pour ajuster notre analyse d’image en fonction de la luminosité ambiante.

Enfin la partie actionneur comporte deux headers de 3 pins, de 2.54mm de pas, pour connecter les servomoteurs et deux cartes drivers A4988 servant à générer les signaux pour les moteurs pas-à-pas.

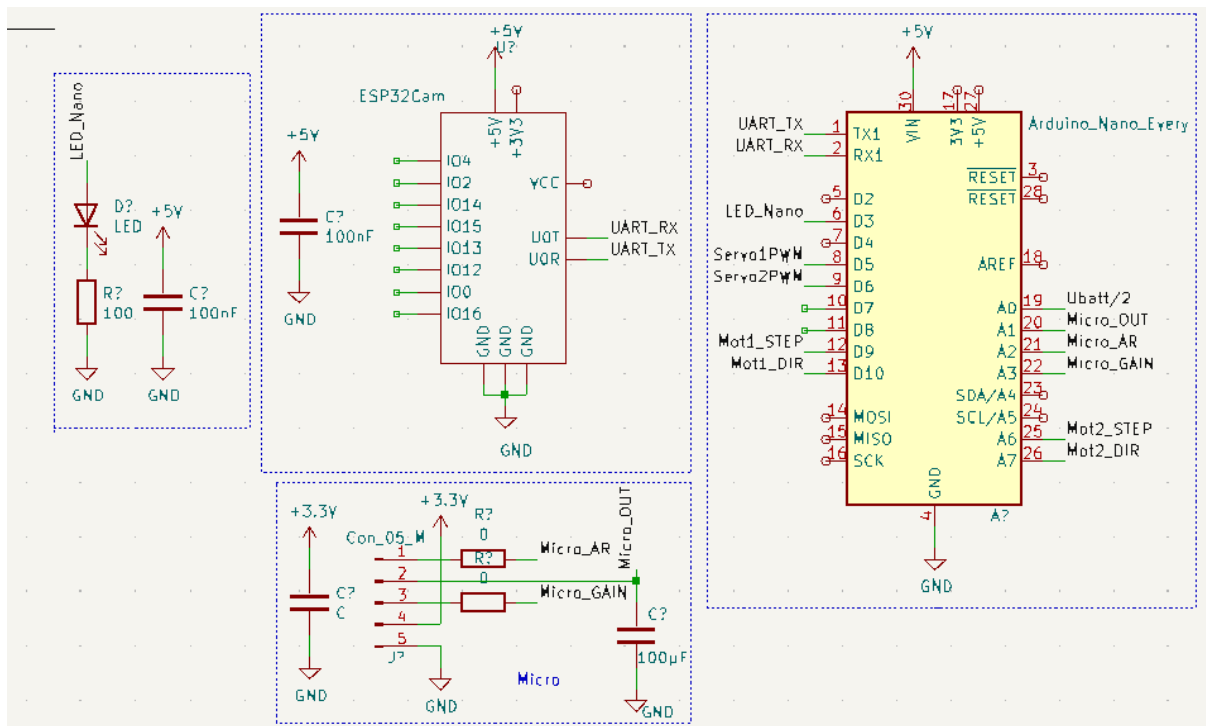
## 2. Schématique

On retrouve sur ce schéma les différents composants nommés ci-dessus et organisés de la même manière logique.

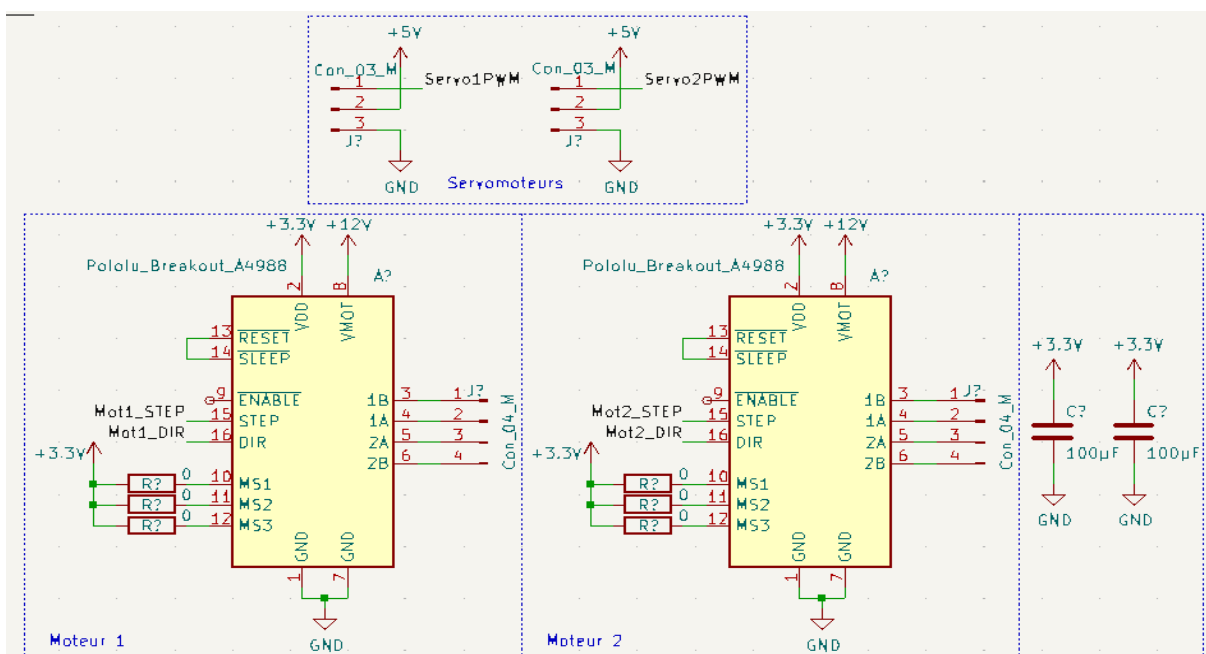
La gestion de l'alimentation est dotée d'un fusible ré-armable et d'une diode anti-inversion qui protègent le circuit d'un court-circuit ou d'une inversion de polarité lors de la connexion des batteries. Nous récupérons aussi la tension des accus pour en déterminer le niveau de charge et ainsi les protéger d'une décharge trop profonde, on peut observer un pont diviseur servant à diviser la tension par deux car l'Arduino n'est pas capable d'acquérir une tension supérieure à 5V.



Les deux puces sont disposés sur la même feuille schématique, sur laquelle nous avons placé au même titre les condensateurs de découplage, la connectique pour le micro et la LED servant d'interface homme machine, de l'Arduino.



Pour finir les deux drivers se retrouvent ici avec les pinheaders des servomoteurs, petite subtilité à prendre en compte, les trois résistances reliées aux broches 10-11-12 des drivers sont en réalité des points de contact à venir souder ou non et ils nous permettent d’agir sur la finesse de réglage de la commande des stepmotors (cf. Datasheet).



## VI. Etat du projet

### 1. Pistes non abouties

- **Pytorch/Tensorflow** : Couplés à un microcontrôleur Raspberry Pi 3, Tensorflow et Pytorch nous permettent une reconnaissance d'objets à partir d'un flux vidéo relativement rapide, le tout avec un taux de confiance élevé. Cependant, utiliser ces outils serait exagéré dans cette situation. Nous préférons utiliser un microcontrôleur moins cher et des méthodes de traitement d'image plus simple, ainsi que les propriétés naturelles des frelons asiatiques.
  - Plus couteux en temps
  - Plus couteux en ressources et calculs
- **Raspberry pi 3** : Ce microcontrôleur a été abandonné en même temps et pour des raisons similaires à Pytorch/Tensorflow.
  - Plus cher
  - Plus couteux en énergie
- **Format JPG** : Ce format d'image a été abandonné pour le format Greyscale. Le format JPG compresse les images ayant initialement trois octets par pixel. Le fait qu'il y ai trois octets rend les images initiales trois fois plus lourdes que leur version greyscale. La compression réduit cette différence mais fait que notre système doit décompresser l'image (qui aura perdu de sa qualité). Aussi, nous devons passer du format RGB (couleur) à un format en niveaux de gris, ce qui signifie des calculs supplémentaires.
  - Plus lourd
  - Nécessite plus de calculs

### 2. Avancement global

Comme vu dans le point précédent, le projet est suffisamment avancé pour pouvoir réfléchir à comment mettre bout à bout les différents éléments qui, seuls à seuls, remplissent correctement leurs fonctions.

En effet, il ne reste à mettre en place que de simples réglages, et de nombreux essais. Comme on peut le voir dans le tableau ci-dessous, chaque fonctionnalité à besoin de finitions et d'ajustements pour permettre une utilisation de notre système en conditions réelles :

Tâche	Complétion
Programmation	80%
Mécanique	80%
PID	60%
Electronique	90%
Algorithmes	90%



De plus, en raison de la saison, les frelons ne sont pas actifs, des essais sur sites ne sont donc pas possibles, ce qui ne permet pas de dire si oui ou non, notre application fonctionne à 100%.

Nous restons nonobstant satisfait de l'avancé de notre projet qui commence à donner des résultats. Avec du temps en plus et la possibilité de faire des tests autour des ruches, nous pensons que ce projet aurait pu porter ses fruits.

### 3. Conclusion personnelle

Malgré le fait que nous n'ayons pas terminé notre projet, nous pouvons dire que l'avons tout de même mené à bien.

Un véritable travail de recherche a été fait, notamment pour le traitement d'image et la mécanique, qui sont des domaines auxquels nous n'étions pas confrontés.

De plus, faire du traitement d'image sur un système embarqué, nécessite un gros travail d'optimisation logiciel, ce qui nous a également progresser sur ce point-là.

# **Rapport de fabrication**

Voir : [Rapport de fabrication](#)