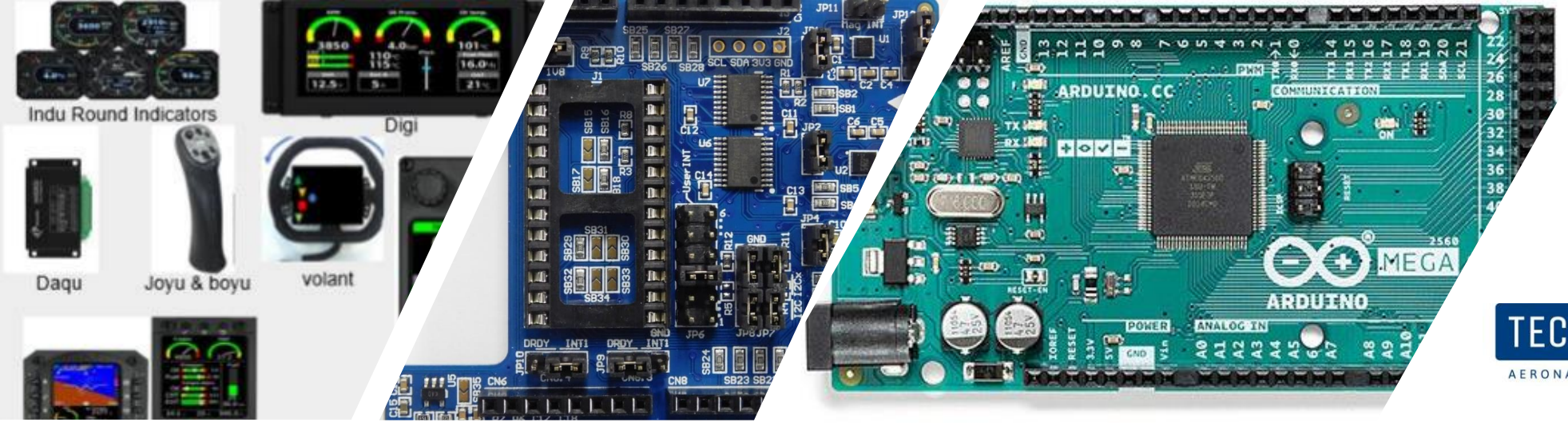
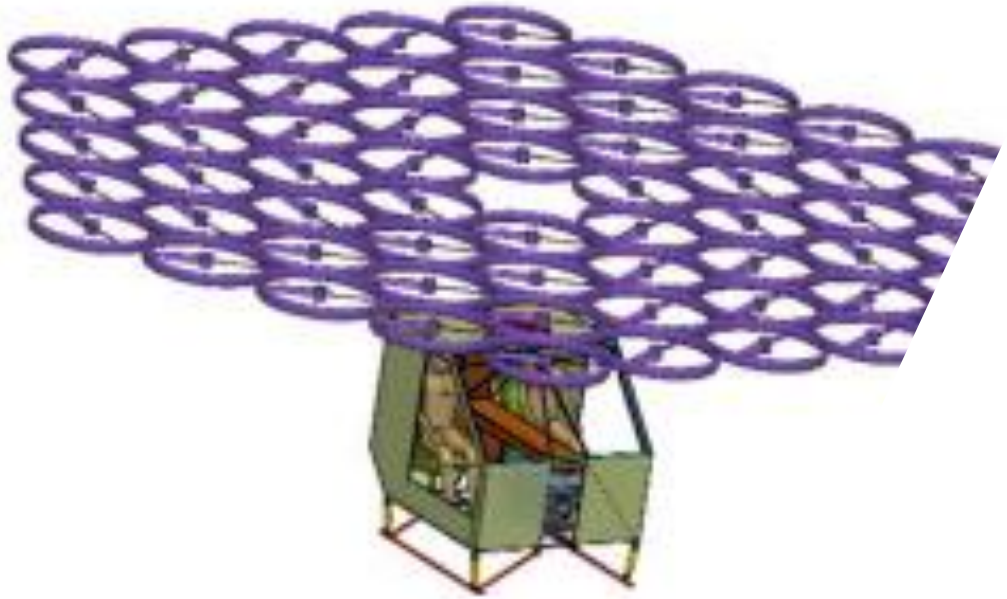


ESTACA
ECOLE D'INGENIEURS

TECHNOPLANE
AERONAUTICAL INNOVATION



Définition et prototypage du FCU et des commandes de vol du Mini-Bee

4e année ESTACA – 2022-2023

Louis BLAISE – Ayush ROY – Stanislas MOURET – Julien MELON – Saad SIAH – Ahsan AJAZ HAIDER

Sommaire

Contexte

Projet Mini-Bee

- Drone VTOL développé par *Technoplane*
- Projet collaboratif entre élèves-ingénieurs venant d'écoles différentes

Objectif de ce projet

- Mettre en commun les recherches de différents étudiants
- Utile lors d'opérations de sécurité et de surveillance

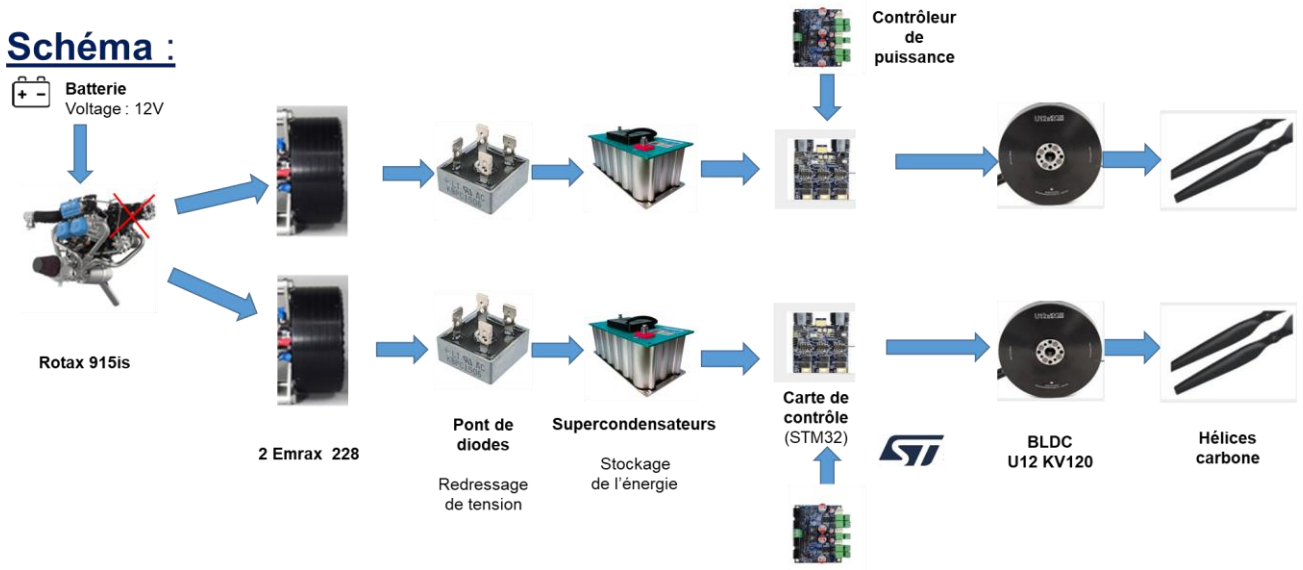
Contexte

Caractéristiques

- VTOL destiné à être autonome
- Configuration hybride avec un moteur à piston Rotax 915is et 2 moteurs électriques Emrax 268 Twin
- 60 rotors
- Stockage possible dans un conteneur aéronautique LD3

Performances

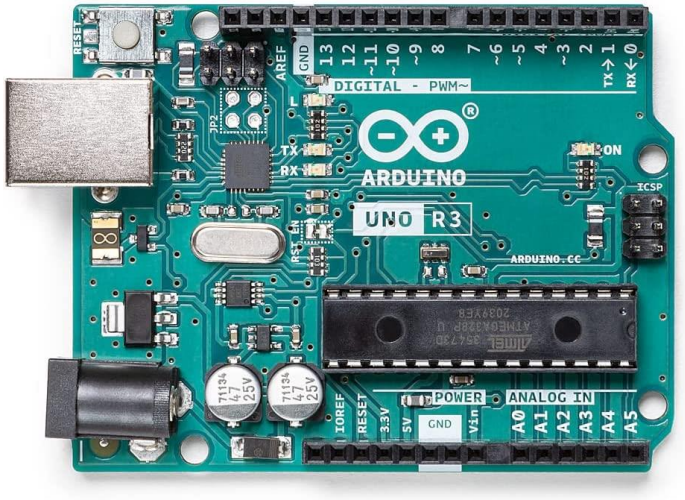
- Vitesse de croisière : 170 km/h
- Distance franchissable : 600 km
- MTOW : 750 kg
- Puissance maximale : 100 kW



Contexte : présentation de Arduino

- Conception et fabrication de cartes électroniques destinées au prototypage
- Objectif : Fournir à un large public une solution pour créer des circuits électroniques
- Lancement de kits de développements et de kits pédagogiques
- Simplicité de l'interface de programmation (IDE) : les élèves ont donc une première approche solide de la programmation d'un microcontrôleur

- Notre package pédagogique sera focalisé sur la **carte Arduino Uno**



Contexte : carte Arduino Uno

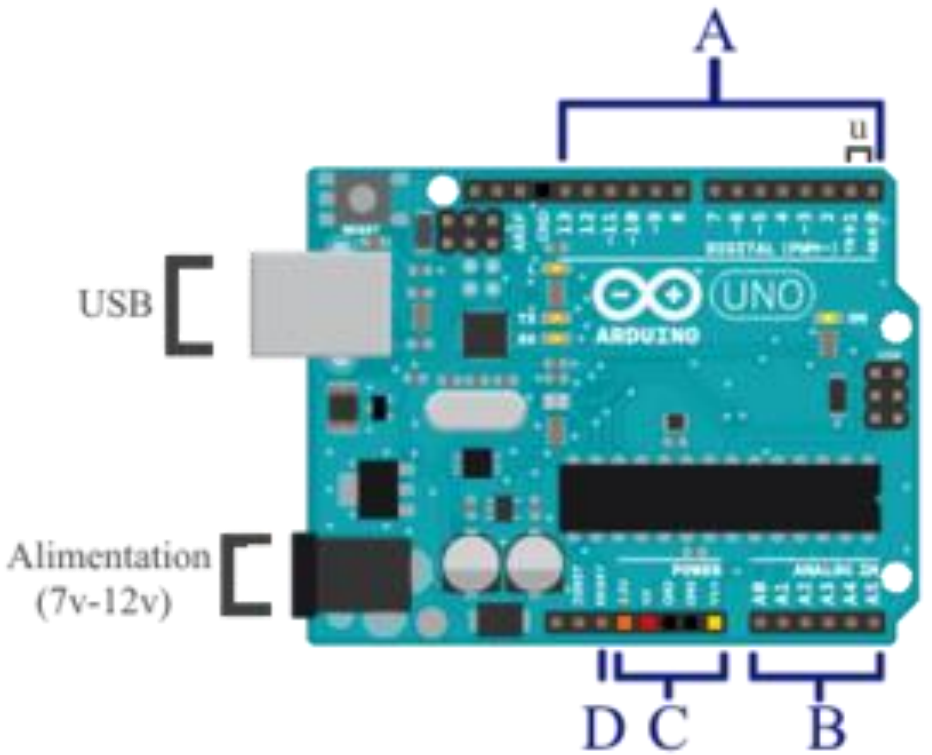
4 types de broches

- Broches A : numériques (0 ou 5 V)

- Broches B : analogiques (entre 0 et 5 V)

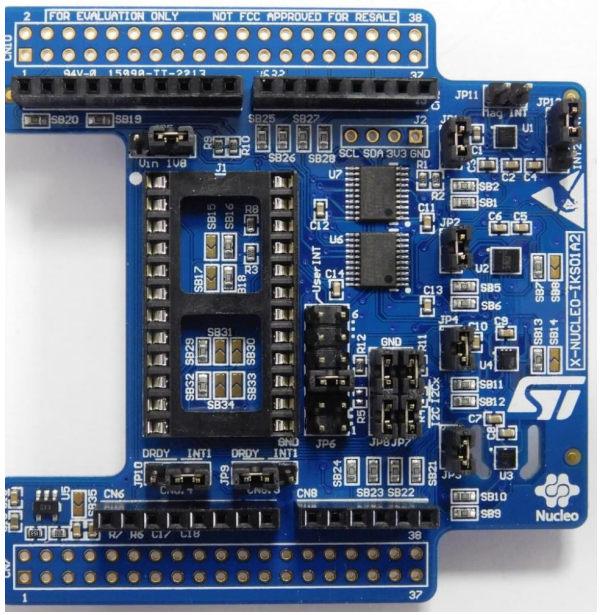
- Broche C en jaune : alimentation électrique entre 7 et 12 V
- Broches C en noir : masses électriques
- Broche C en rouge : sortie en +5 V
- Broche C en orange : sortie en +3,3 V

- Broche D : RESET, pour réinitialiser la carte



Contexte : présentation de STMicroelectronics

- Entreprise spécialisée dans les microcontrôleurs, les capteurs et les circuits embarqués
- Domaine des transports et de la défense
- La famille de microcontrôleurs STM32 nous semble adaptée pour assurer le contrôle en vol du Mini-Bee



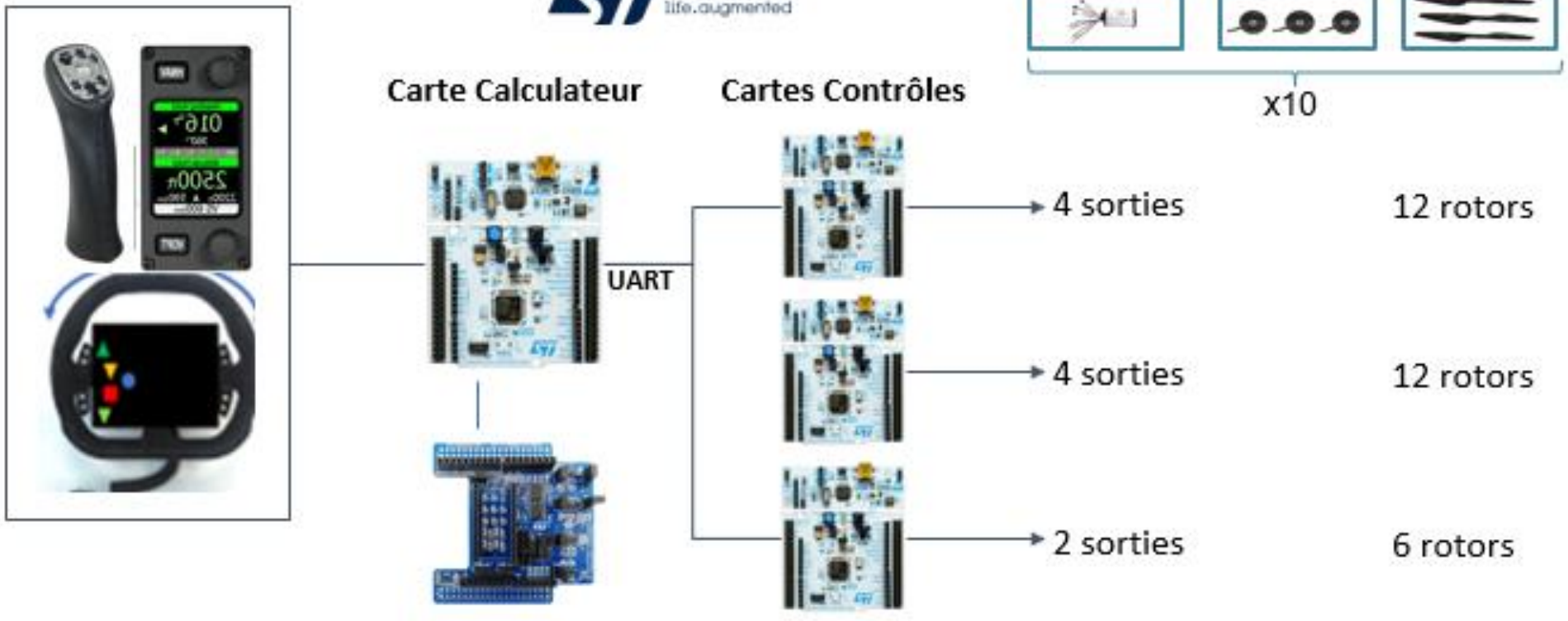
FCU

CY TECH

Prototypage du MiniBee contrôle à l'aide de plusieurs cartes STM32

TECHNOPLANE AERONAUTICAL INNOVATION

life.augmented



Objectif : Prototyper le FCU (carte et composants de contrôle)

1. Définition des commandes de vol et des règles du FCU

- Définition des règles de **contrôle de vol**
- Modélisation du comportement en vol en utilisant les projets passés menés par d'autres écoles
- Configuration et répartition des **rotors** pour en assurer le contrôle

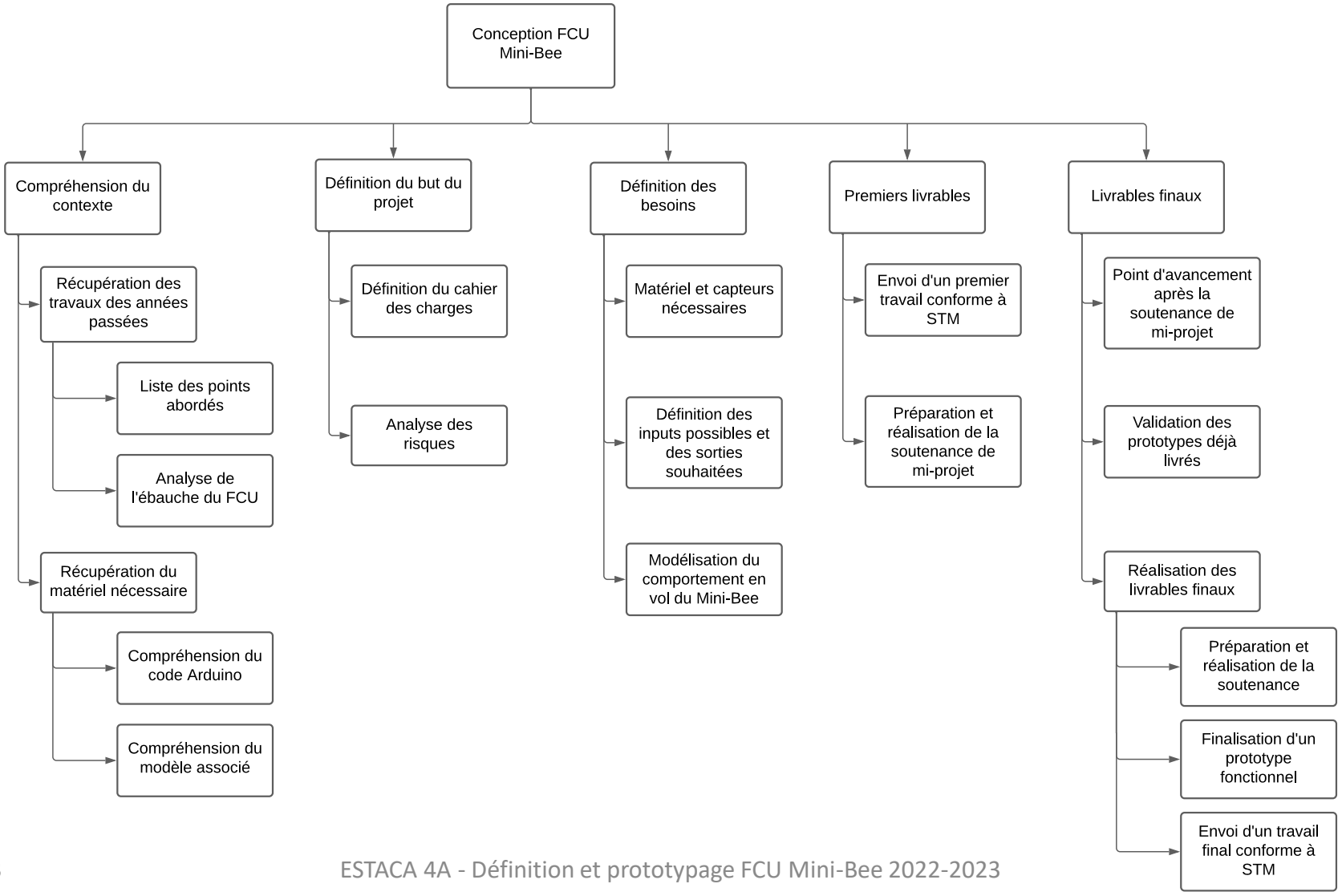
2. Définition de l'architecture Capteurs et des Packages pédagogiques

- Définition des capteurs et de l'intégration des données de contrôle
- Choix des composants et rédaction du package pédagogique **Arduino**
- Choix des composants et rédaction du package pédagogique **STM**

3. Prototypage

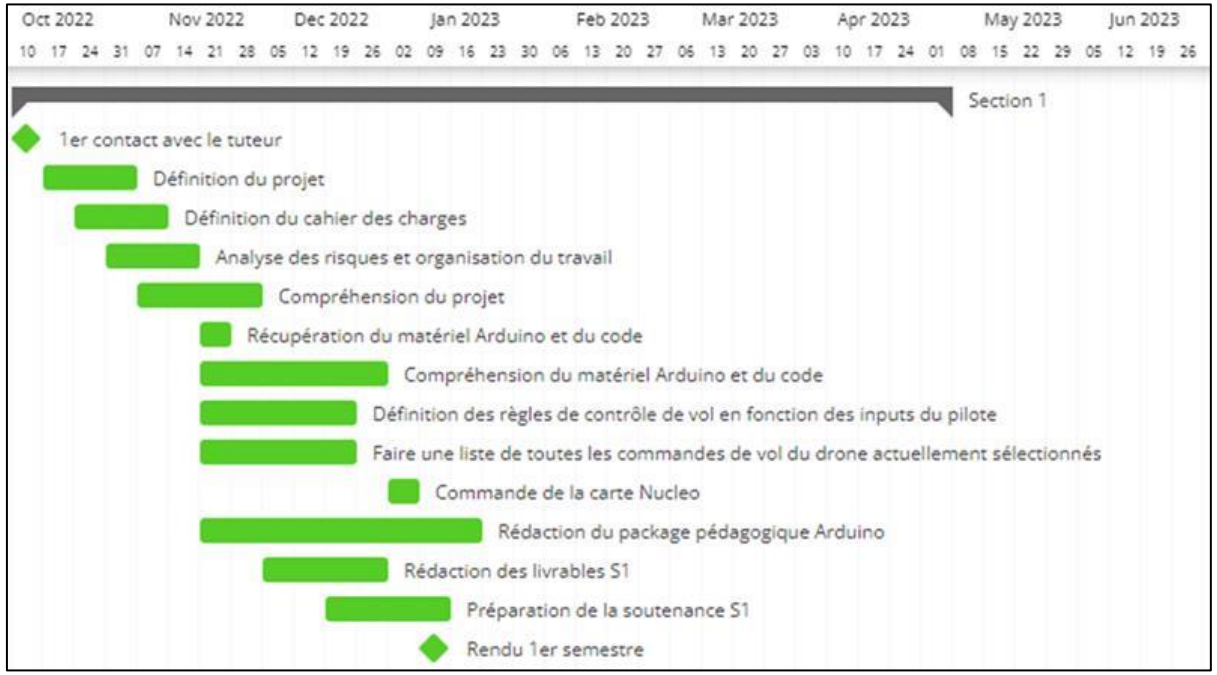
- Prototypage et premier test des sorties de contrôle.
- **Prototype fonctionnel** qui permet d'envoyer des premières informations de contrôle

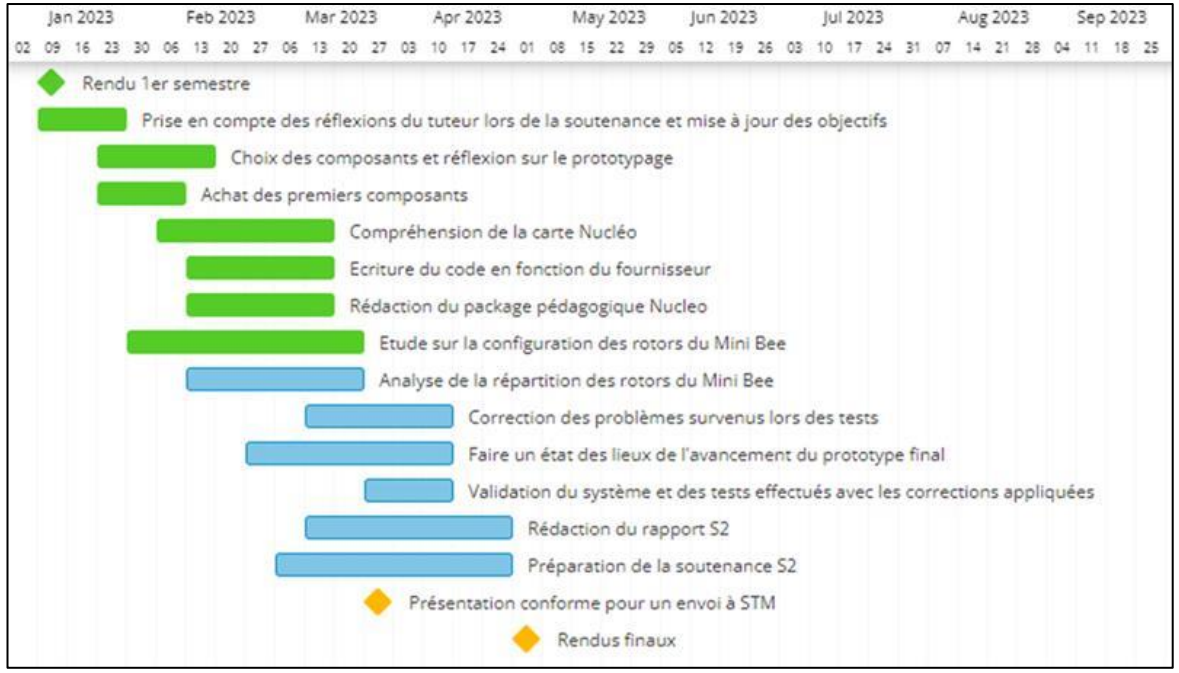
Organisation et planification (WBS du projet)



Mois	Liste des tâches	heures estimées	heures réalisées
Octobre	Premier contact avec le tuteur	2	3
	Définition du projet	4	4
	Définition du cahier des charges	5	4
	Analyse des risques et organisation de travail	10	12
Novembre	Compréhension du projet	10	14
	Récupération du matériel Arduino et du code	2	4
	Compréhension du matériel Arduino et du code	20	14
	Définition des règles de contrôle de vol en fonction des inputs du pilote	25	40
	Faire une liste de toutes les commandes de vol du drone actuellement sélectionnés	15	10
Décembre	Commande de la carte Nucleo	5	5
	Rédaction du package pédagogique Arduino	25	35
	Rédaction des livrables S1	35	25
	Préparation de la soutenance S1	30	24
Janvier	Prise en compte des réflexions du tuteur lors de la soutenance et mise à jour des objectifs	15	10
	Choix des composants et réflexion sur le prototypage	12	15
	Compréhension de la carte Nucléo	10	5
Février	Ecriture du code en fonction du fournisseur	30	25
	Rédaction du package pédagogique Nucleo	20	20
	Etude sur la configuration des rotors du Mini Bee	15	20
Mars	Analyse de la répartition des rotors du Mini Bee	20	20
	Correction des problèmes survenus lors des tests	15	10
	Faire un état des lieux de l'avancement du prototype final	15	15
Avril	Validation du système et des tests effectués avec les corrections appliquées	20	20
	Préparation de la soutenance S2, présentation conforme pour un envoi à STM	30	25
	Rendus finaux	20	15
	TOTAL d'heures estimées	410	394

Diagramme de Gantt





Répartition des tâches

Membres	Compétences	Faiblesses	Appétences	Jokers
Julien	CAO MEF	Matlab	MEF	
Stanislas	Arduino Electronique	CAO	Matlab	
Louis	Arduino Electronique	Matlab	MEF, Arduino	RDM
Saad	CAO	Electronique	Fonctionnement du drone	Arduino
Ayush	MEF	Matlab	MEF	
Ahsan	CAO	Matlab, Arduino	Drones et composants	Arduino

Rôles	Lead	Backup
Coordinateur de projet	Saad	Ahsan
Elaboration du code Arduino et de la maquette	Louis	Stanislas
Achats des composants STM et Arduino avec validation du tuteur	Saad	Ahsan
Montage avec les composants STM	Stanislas	Louis
Rédaction du package pédagogique Arduino	Louis	Stanislas
Rédaction des objectifs globaux qui doivent être dans le package pédagogique Arduino	Ayush	Julien
Lister toutes les commandes de vol du Mini Bee + compréhension des fonctions de pilotage	Julien	Saad
Définition des règles de contrôle de vol en fonction des inputs du pilote	Ahsan	Ayush
Définition des capteurs et équipements disponibles + intégration des données de contrôle	Julien + Ayush	Ahsan

Répartition des tâches

Nous nous sommes répartis en trois équipes pour mieux répartir les tâches et avancer sur le projet

1ère équipe

Réalisation de la maquette Arduino et écriture du code

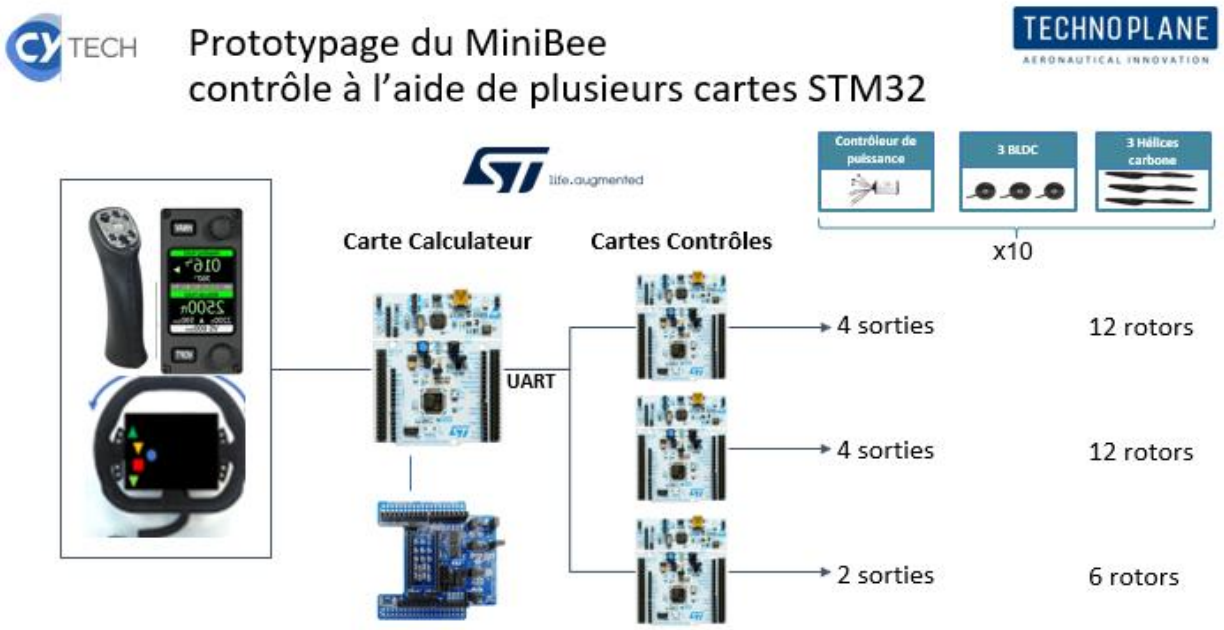
2nde équipe

Dresser la liste des capteurs nécessaires au Mini-Bee et rechercher les composants STM

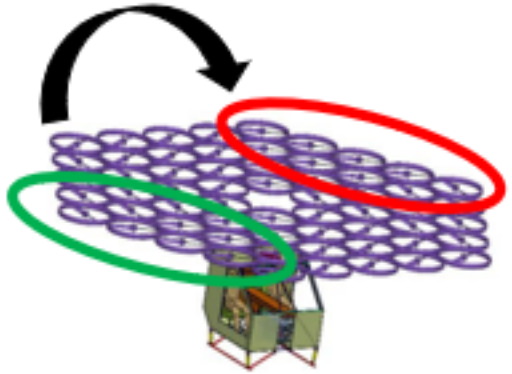
3ème équipe

Rédiger le package pédagogique Arduino et STM

Commande de vol et FCU

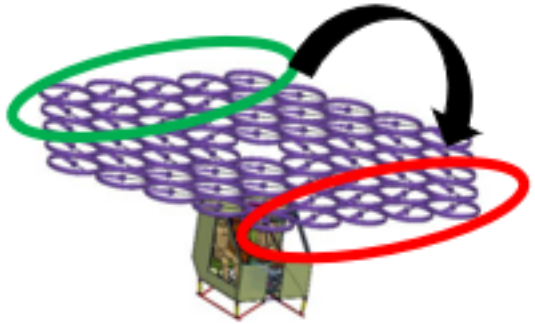


Commandes de vol



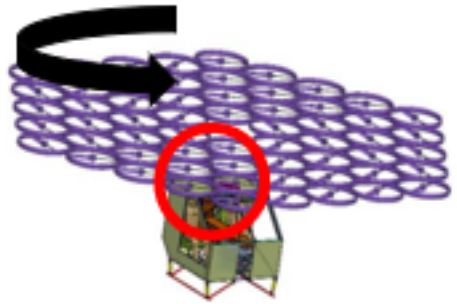
Roulis

- Différentiation du pas ou de la vitesse selon le rotor
- Graduation par rapport à l'axe longitudinal



Tangage

- Différentiation du pas ou de la vitesse selon le rotor
- Graduation par rapport à l'axe latéral



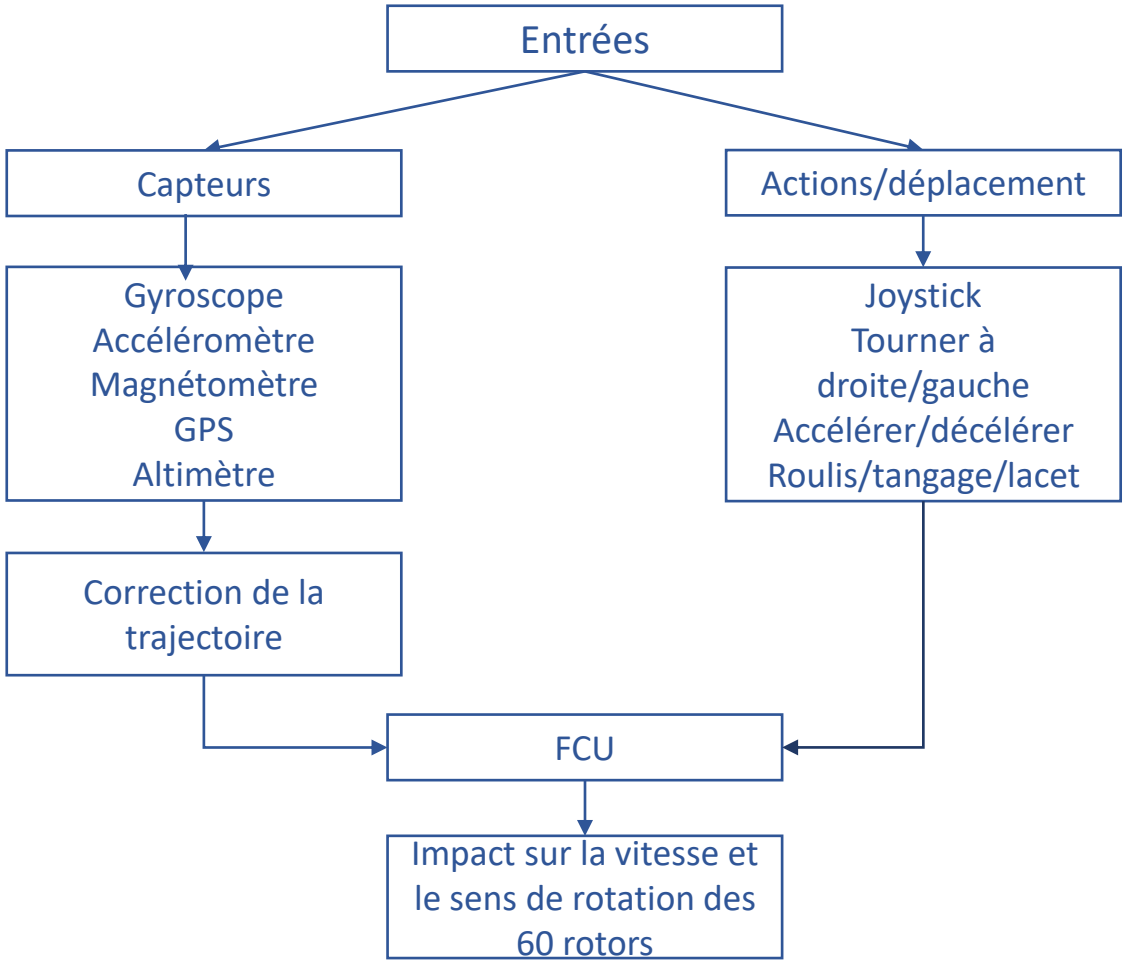
Lacet

- Différentiation de la vitesse de rotation selon le rotor
- Définition d'un groupe de rotors dédiés à cette commande

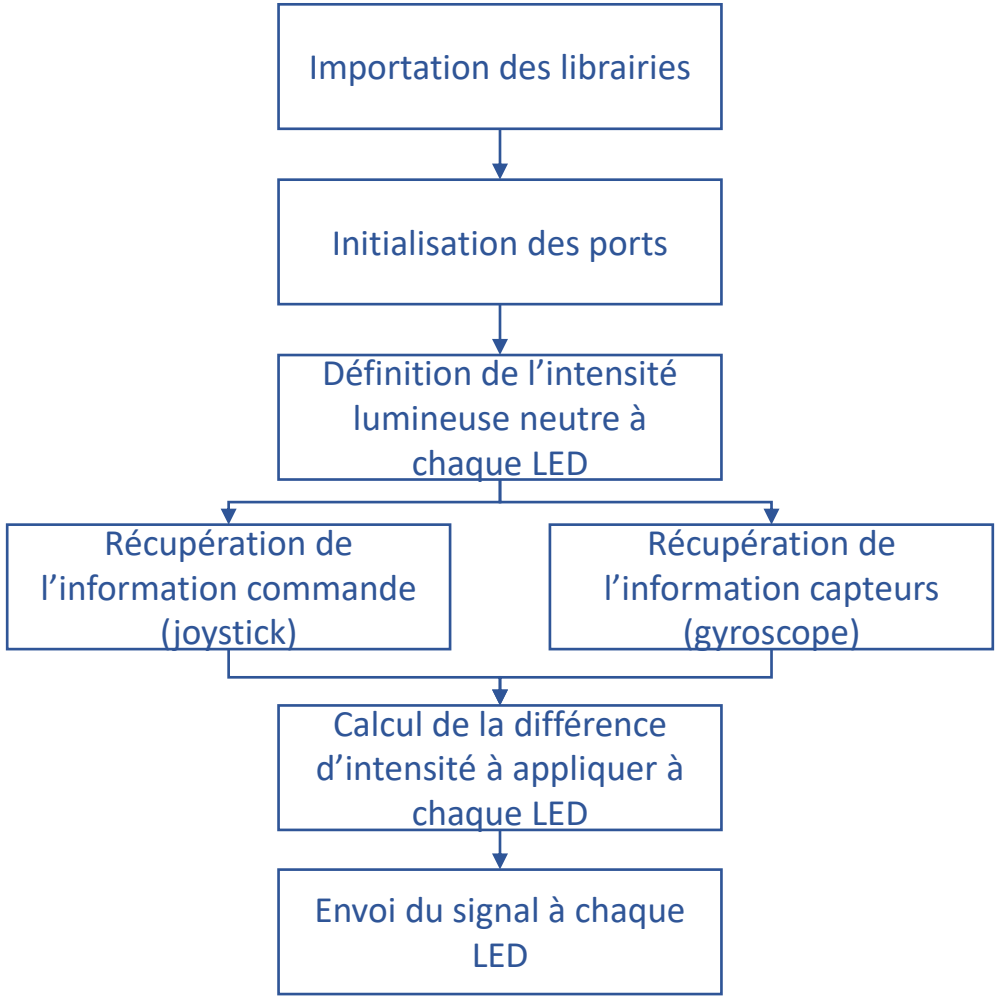
: rotors rapides

: rotors lents

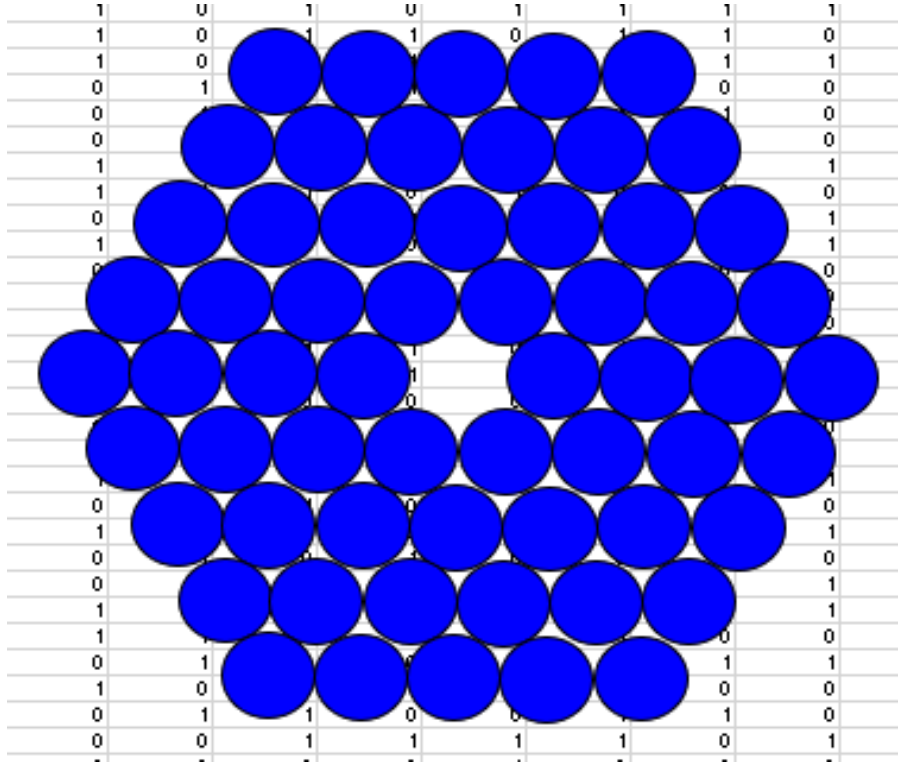
Schéma fonctionnel du FCU



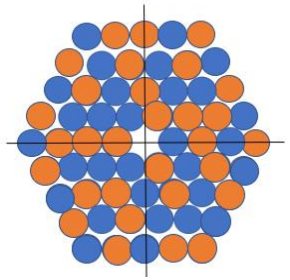
Logigramme du code Arduino pour le FCU



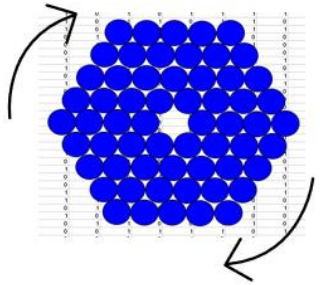
Configuration des rotors



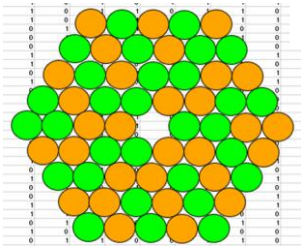
Méthodologie



1 - Etude de quelques configurations faites à la main



2 - Analyse en profondeur des mouvements du Mini-Bee

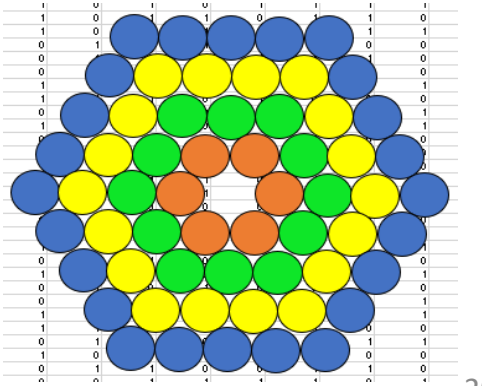


3 - Prise en compte des critères et des besoins

4 - Sélection des configurations optimales

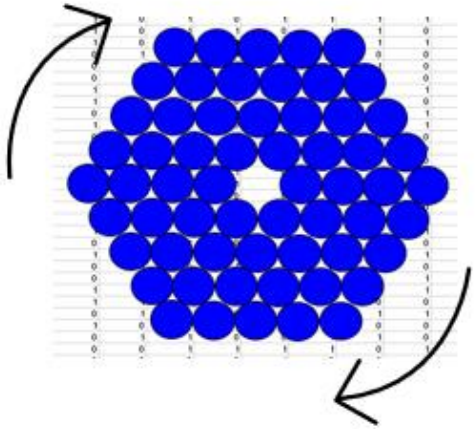
On divise les 60 rotors par couronne

Couronne 1	Couronne 2	Couronne 3	Couronne 4
			



Phénomènes autour du pilotage :

Lacet



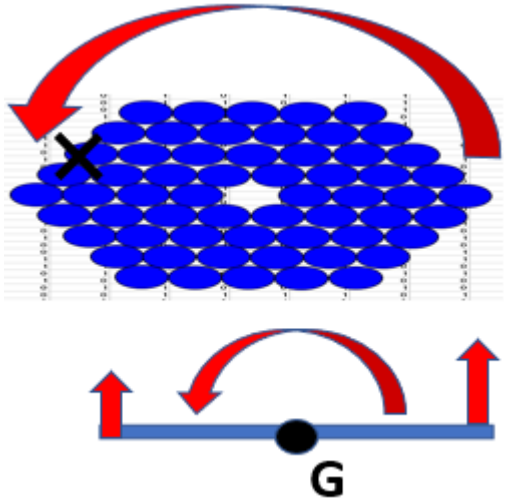
Moment = J*w
La distance n'intervient pas !
30 rotors horaire
30 rotors anti-horaire

- Causes du moment de lacet

Plus forte poussée des rotors dans un sens de rotation

Cas de panne = présence d'un moment lacet non nul -> **NON VOULU**

Roulis



- Causes du moment de roulis

Dissymétrie de la poussée : Plus forte poussée des rotors d'un même côté

Cas de panne = présence d'un moment roulis non nul -> **NON VOULU**

Cas de panne

Problème : panne d'un rotor

Coupure du rotor exacte opposé par rapport à la cabine

Augmentation de la poussée des rotors à proximité

Contrainte 1 : sens opposé

Contrainte 2: Même couronne

Contrainte 1: Le barycentre de la poussée ajoutée doit être sur l'axe entre la cabine et le rotor en panne

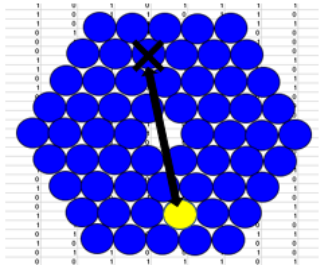
Contrainte 2 : sens opposé

Conserve un moment de lacet nul

Conserve un roulis nul

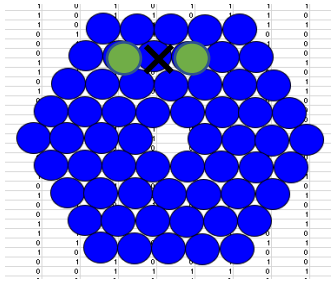
Conserve un roulis nul

Conserve un moment de lacet nul



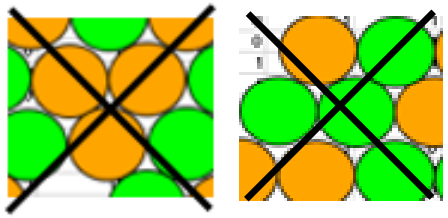
CONCLUSION

CONTRAINTE GENERALE : On doit avoir une symétrie centrale



Modes de répartition

Minimiser les regroupements de 3 rotors :



Critères
Homogénéité
Stabilité
Symétrie centrale

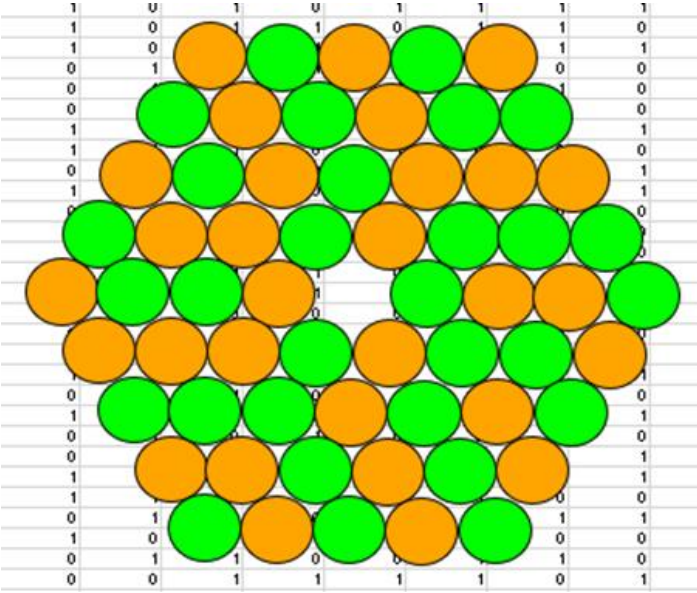
Homogénéité => Alternance des sens de rotation sur chaque couronne

Stabilité => Aucun moment induit

Symétrie centrale => Facilité cas de panne

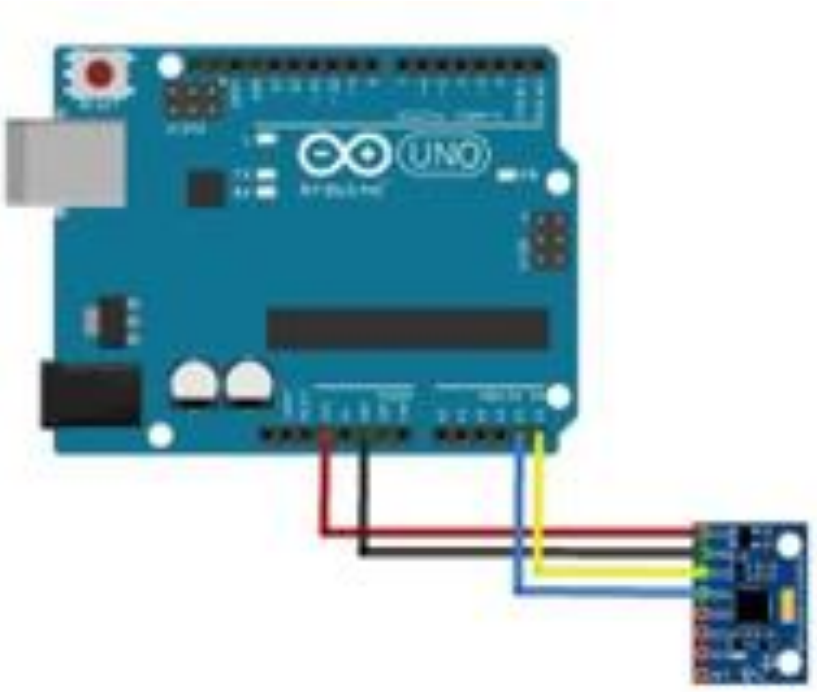
Sélection et notation des configurations

Configuration retenue !



Note 5/5

Package Arduino

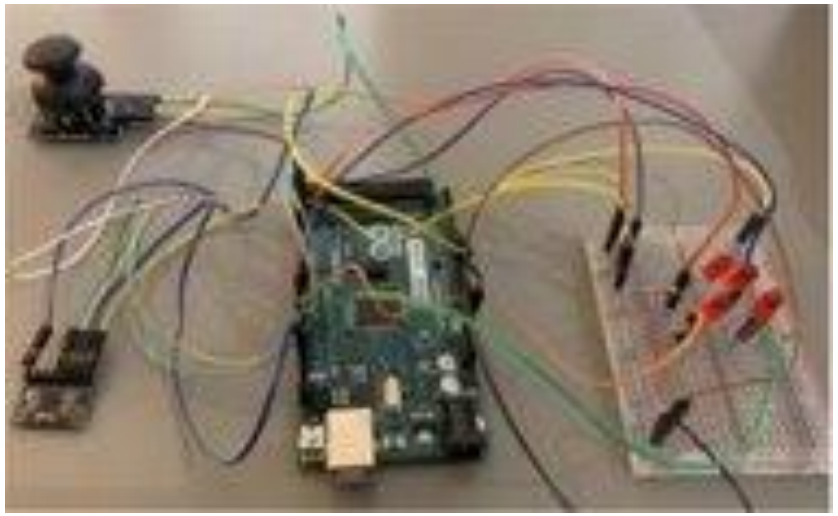


```
Code_prototype_leds_sallument_selon_joystick §
void loop() {
  valeurX = analogRead(axex);
  if (valeurX > 700){ // droite
    analogWrite(ledR,50);
    Serial.print("x = ");
    Serial.println(valeurX);
  }else{
    digitalWrite(ledR, LOW );
  }
    valeurX = analogRead(axex);
  if (valeurX > 1022){ // droitedroite
    digitalWrite(ledRR,HIGH);
    Serial.print("x = ");
    Serial.println(valeurX);
  }else{
    digitalWrite(ledRR, LOW );
  }

  valeurX = analogRead(axex);
  if (valeurX < 400){ // gauche
    analogWrite(ledL,50);
    Serial.print("x = ");
    Serial.println(valeurX);
  }else{
    digitalWrite(ledL, LOW );
  }
  valeurX = analogRead(axex);
```

Package pédagogique Arduino

Français / Anglais

Objectifs : Se familiariser avec l'électronique et rédiger un TP à destination d'autres étudiants

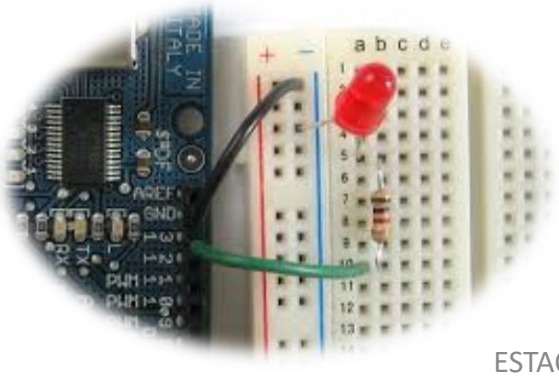


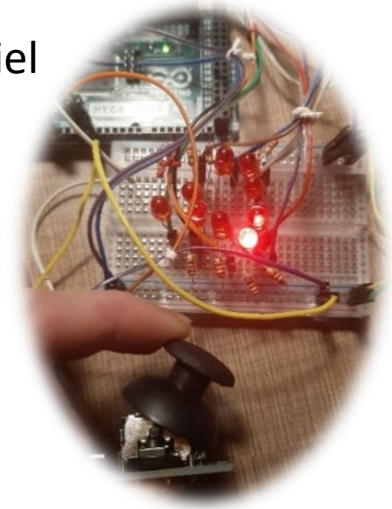
ESTACA 4A - Définition et prototypage FCU Mini-Bee 2022-2023

Package articulé autour de 3 composants :

- Une LED
- Un joystick
- Un gyroscope

Pour chaque partie, on développe les objectifs, le matériel nécessaire, le montage et le code Arduino.

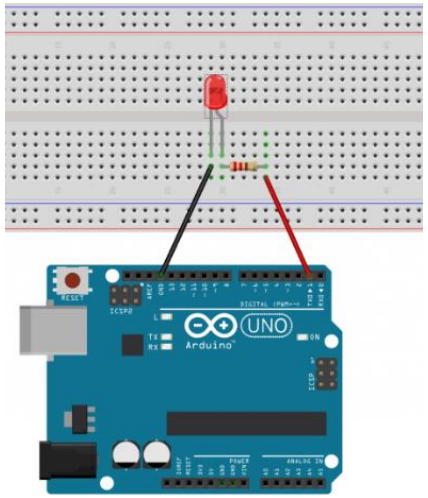




Package pédagogique Arduino

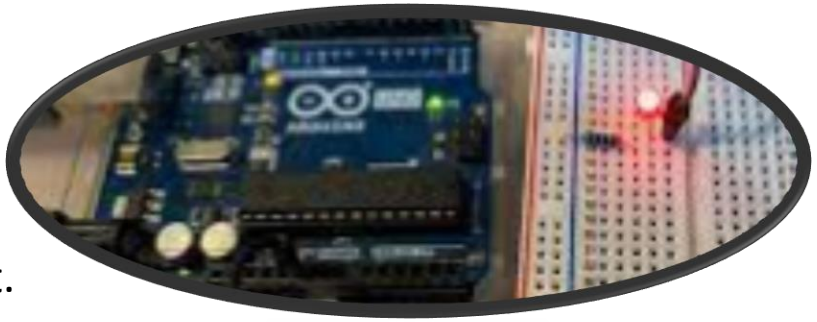
Montage pour faire clignoter une LED

- Connecter la borne numérique (côté digital) numéro 1 de la carte Arduino à la patte de la résistance.
- Connecter la deuxième patte de la résistance à l'anode de la LED (borne +, tige la plus longue).
- Brancher la cathode (borne -, tige la plus courte) de la LED au GND de l'Arduino.



Code Arduino :

```
void setup() {
    pinMode(1, OUTPUT);  //Initialisation de la broche 1 en sortie
  }
  void loop() {
    digitalWrite(1, HIGH);  //Mise au niveau Haut (Allumage) de la LED
    delay(1000);      //Délai de 1000ms dans cette position (LED allumée)
    digitalWrite(1, LOW);   //Mise au niveau bas (Eteint) de la LED
    delay(1000);      //Délai de 1000ms dans cette position (LED éteinte)
}
```



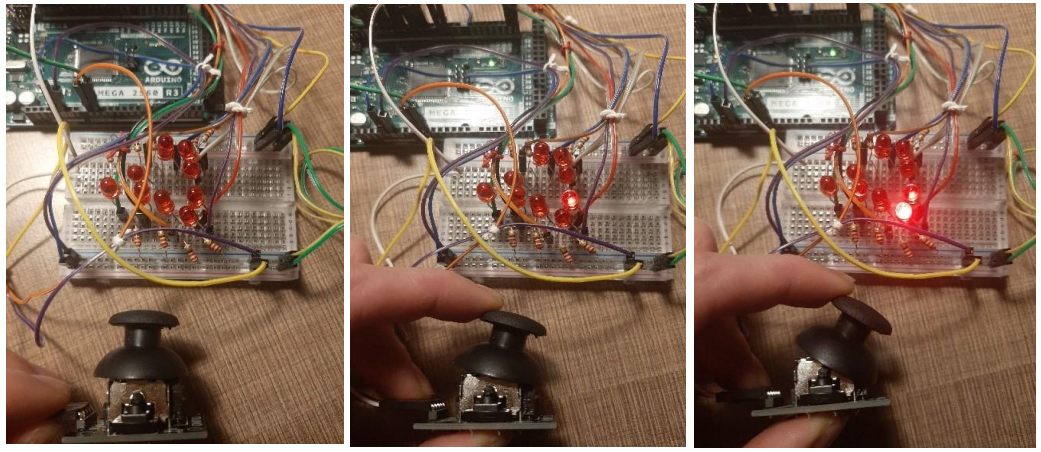
- Pour modifier l'intensité lumineuse de la LED, remplacer HIGH par un nombre entre 0 (éteint, LOW) et 255 (HIGH). Entre ces valeurs, l'intensité varie linéairement.

Package pédagogique Arduino

D'autres exercices permettent de jouer avec les fonctionnalités Arduino et découvrir ludiquement :

- LED : Allumer, faire varier l'intensité, la faire clignoter

=> Fonction de base

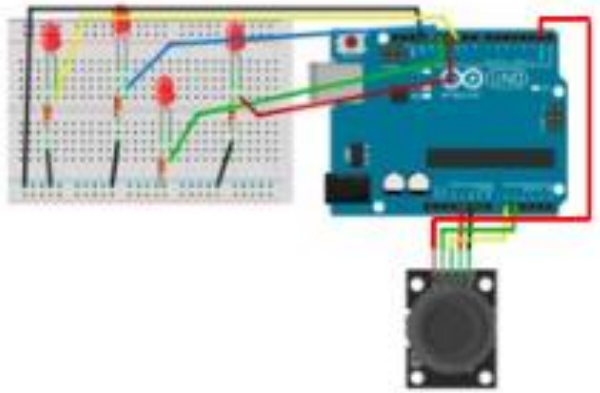


- Joystick : Connaître sa position avec des LEDs (4 puis 8) (visuel)
  Connaître sa position précisément

=> Utilisation du savoir-faire appris des LEDs

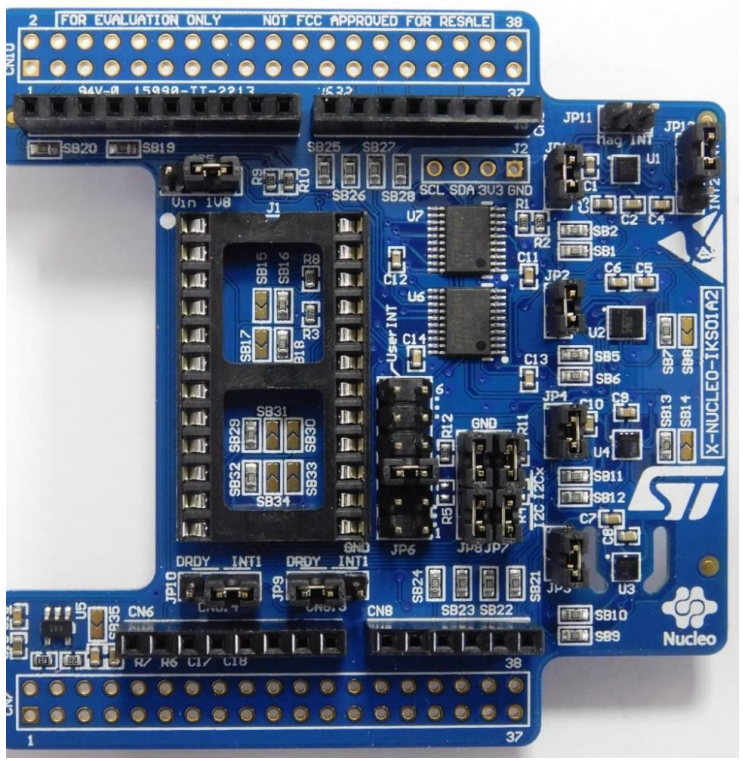
- Gyroscope : Connaître son orientation angulaire (situer le MiniBee dans l'espace)

=> Système plus complexe



Connaissances suffisantes pour l'utilisateur pour découvrir d'autres capteurs !

Package ST Nucléo

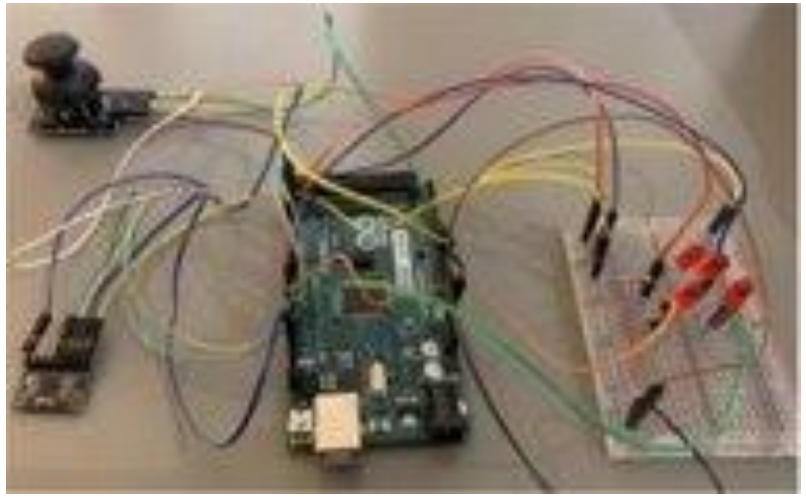


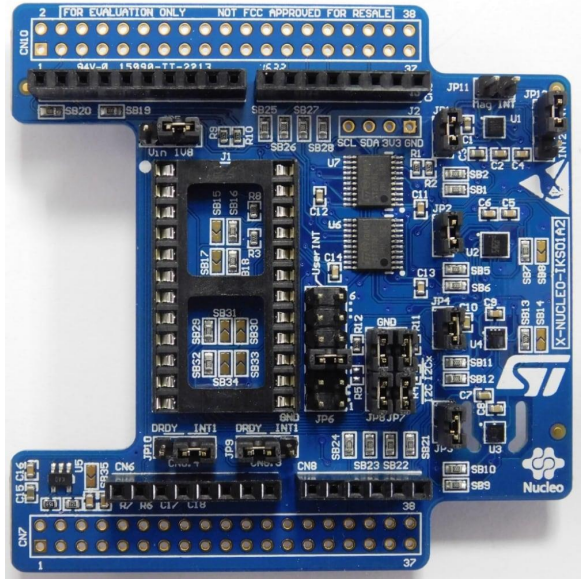

```
trolMotors() {
ontrol the motors based on the joystick x and y values
x = analogRead(JOYSTICK_X_PIN);
y = analogRead(JOYSTICK_Y_PIN);

 > 512) {
//Move the motors to the right
se if(x < 512) {
//Move the motors to the left


 > 512) {
//Move the motors forward
se if(y < 512) {
//Move the motors backward
```

Prototype STM pour 2023

La carte X-NUCLEO-IKS01A2 présente des avantages considérables

- Regroupement de plusieurs capteurs (gyroscope, accéléromètre,...) au sein de la carte.
- Diminution importante de la complexité.
- Pertinence pour le projet car centralisation des commandes.





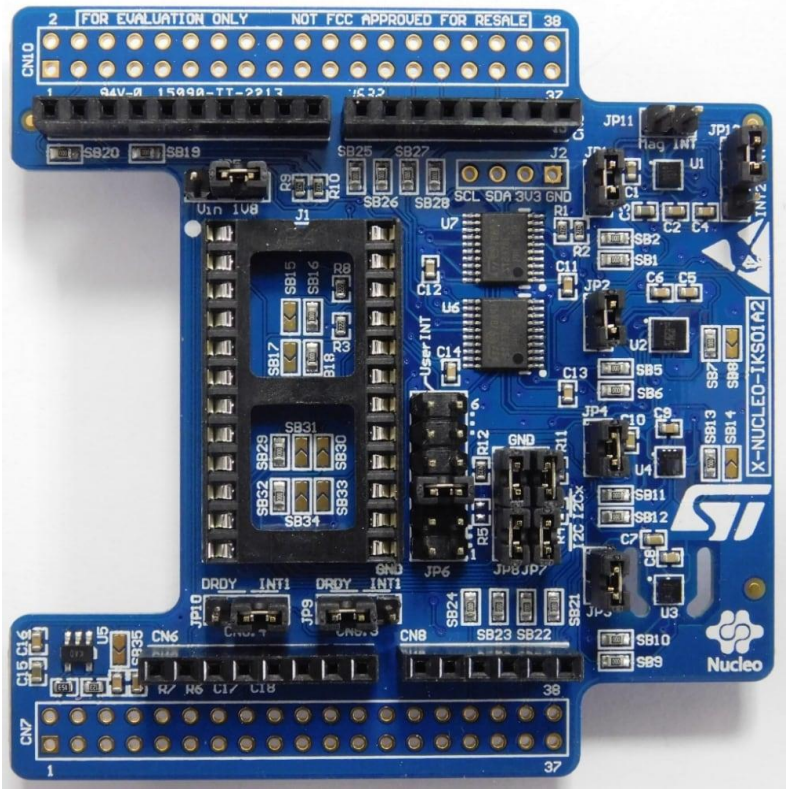
Les avantages de la Nucleo par rapport à l'Arduino

- Microcontroleur STM32 (série ARM Cortex-M).
- IDE: STM32CubeIDE
- C/C++

4 Points majeurs:

- Performance
- Puissance de calcul
- Connectivité
- Écosystème

- Carte nucléo X-NUCLEO-IKS01A2
- Centrale 6-axes IMU, LSM6DSR
- Altimètre, LPS22DF
- Boussole, LIS2MDL



La carte X-NUCLEO-IKS01A2

Présentation package pédagogique STM & enjeux

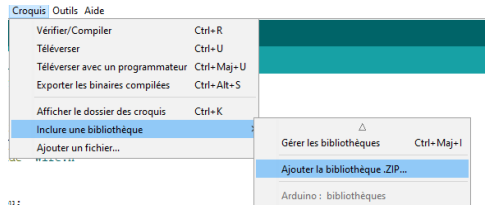
3 exercices apportant des compétences en vue du contrôle autonome de la Mini Bee:

- Faire clignoter une LED
- Récupérer la position d'un joystick
- Commander un moteur

Extrait Package pédagogique Nucléo

Installer les bibliothèques Nucléo

- Afin de pouvoir commander cette carte par l'intermédiaire du logiciel Arduino, il est nécessaire d'installer une bibliothèque accessible via ce lien : https://www.arduinolibraries.info/libraries/stm32duino-x-nucleo-iks01-a2
- Sur Arduino, aller dans Croquis -> Inclure une bibliothèque -> Ajouter la bibliothèque .ZIP.



- Il faut maintenant installer le protocole de communication utilisé pour la carte Nucleo. Pour cela, aller dans Fichier -> Préférences, puis rentrer ce lien dans URL de gestionnaire de cartes supplémentaires : https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json
- Le protocole est maintenant téléchargé, pour l'installer, aller dans Outils -> Type de carte -> Gestionnaire de carte.
- Sélectionner la dernière version disponible (ici 2.4.0) puis Installer. Le processus prend quelques minutes. Si tout s'est bien déroulé, la mention INSTALLED doit apparaître.

Extrait Package pédagogique Nucléo

Faire clignoter une LED

- Créez un nouveau projet en sélectionnant le type de microcontrôleur utilisé sur votre carte Nucleo (ici STM32).
- Ajoutez un fichier source "main.c" à votre projet, et incluez-y les headers de la bibliothèque HAL
- Dans votre fonction "main.c", configurez les broches de la LED en tant que sortie

```
GPIO_InitTypeDef GPIO_InitStruct;
__HAL_RCC_GPIOA_CLK_ENABLE();
GPIO_InitStruct.Pin = GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

- Utilisez la fonction "HAL_GPIO_WritePin()" pour mettre la LED à l'état haut et l'allumer
- Utilisez la fonction "HAL_Delay()" pour mettre le programme en pause durant 1 seconde
- Répétez les deux dernières étapes pour faire clignoter la LED un nombre de fois déterminé.

Extrait Package pédagogique Nucléo

Récupérer la position non analogique d'un joystick

- Dans la fonction "main.c", configurez les broches que vous allez utiliser par la suite

```
GPIO_InitTypeDef GPIO_InitStruct;
__HAL_RCC_GPIOC_CLK_ENABLE();
GPIO_InitStruct.Pin = GPIO_PIN_0 | GPIO_PIN_1;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_PULLDOWN;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
```

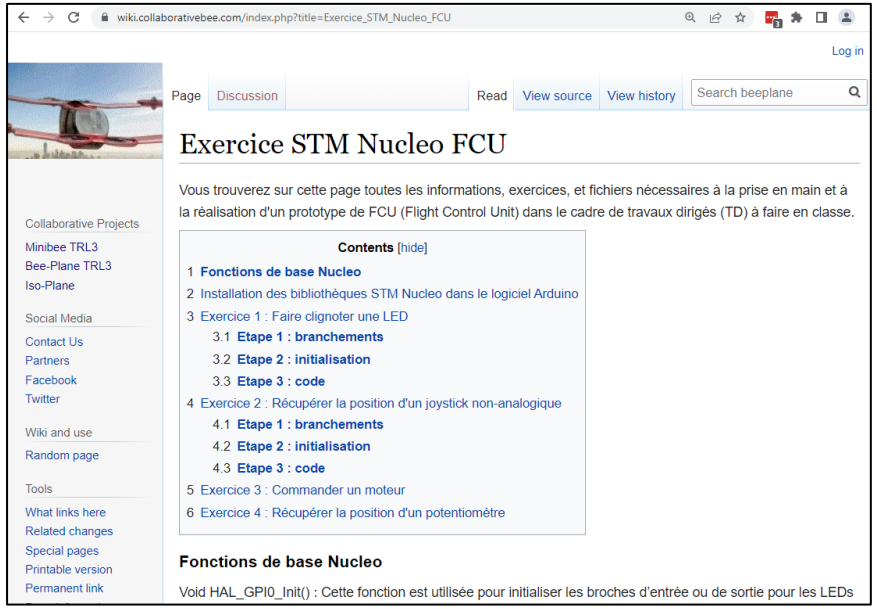
- Utilisez la fonction HAL_GPIO_ReadPin() pour lire l'état des broches du joystick, pour savoir dans quelle direction il est incliné

```
if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_0) == GPIO_PIN_SET)
{
    // joystick incliné vers le haut
}
else if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_1) == GPIO_PIN_SET)
{
    // joystick incliné vers le bas
}
```

- Utilisez les informations obtenues pour mettre à jour l'affichage de votre application ou pour contrôler un moteur ou tout autre composant connecté à votre carte.

Extrait Package pédagogique Nucléo

https://wiki.collaborativebee.com/index.php?title=Exercice_Arduino_FCU
https://wiki.collaborativebee.com/index.php?title=Exercise_Arduino_FCU_English
https://wiki.collaborativebee.com/index.php?title=Exercice_STM_Nucleo_FCU



wiki.collaborativebee.com/index.php?title=Exercice_STM_Nucleo_FCU

Log in

Page | Discussion | Read | View source | View history | Search beeplane

Exercice STM Nucleo FCU

Vous trouverez sur cette page toutes les informations, exercices, et fichiers nécessaires à la prise en main et à la réalisation d'un prototype de FCU (Flight Control Unit) dans le cadre de travaux dirigés (TD) à faire en classe.

Collaborative Projects
Minibee TRL3
Bee-Plane TRL3
Iso-Plane

Social Media
Contact Us
Partners
Facebook
Twitter

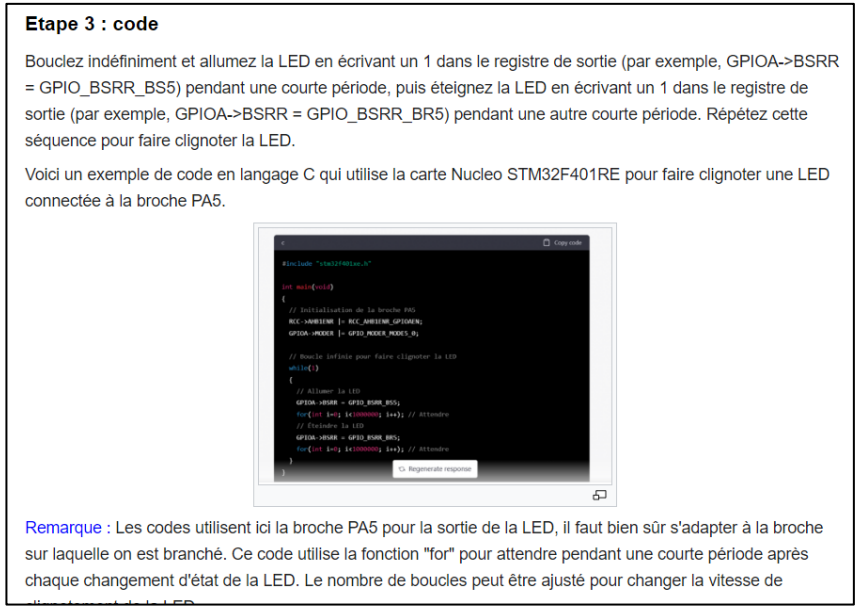
Wiki and use
Random page

Tools
What links here
Related changes
Special pages
Printable version
Permanent link

Contents [hide]

Fonctions de base Nucleo

Void HAL_GPIO_Init() : Cette fonction est utilisée pour initialiser les broches d'entrée ou de sortie pour les LEDs



Etape 3 : code

Bouclez indéfiniment et allumez la LED en écrivant un 1 dans le registre de sortie (par exemple, GPIOA->BSRR = GPIO_BSRR_BS5) pendant une courte période, puis éteignez la LED en écrivant un 1 dans le registre de sortie (par exemple, GPIOA->BSRR = GPIO_BSRR_BR5) pendant une autre courte période. Répétez cette séquence pour faire clignoter la LED.

Voici un exemple de code en langage C qui utilise la carte Nucleo STM32F401RE pour faire clignoter une LED connectée à la broche PA5.

```
#include "stm32f401xe.h"

int main(void)
{
  // Initialisation de la broche PA5
  RCC->AHB1ENR |= RCC_AHB1ENR_GPIOAEN;
  GPIOA->MODER |= GPIO_MODER_MODER5_0;

  // Boucle infinie pour faire clignoter la LED
  while(1)
  {
    // Allumer la LED
    GPIOA->BSRR = GPIO_BSRR_BS5;
    for(int i=0; i<1000000; i++); // Attendre
    // Éteindre la LED
    GPIOA->BSRR = GPIO_BSRR_BR5;
    for(int i=0; i<1000000; i++); // Attendre
  }
}
```

Remarque : Les codes utilisent ici la broche PA5 pour la sortie de la LED, il faut bien sûr s'adapter à la broche sur laquelle on est branché. Ce code utilise la fonction "for" pour attendre pendant une courte période après chaque changement d'état de la LED. Le nombre de boucles peut être ajusté pour changer la vitesse de

Conclusion

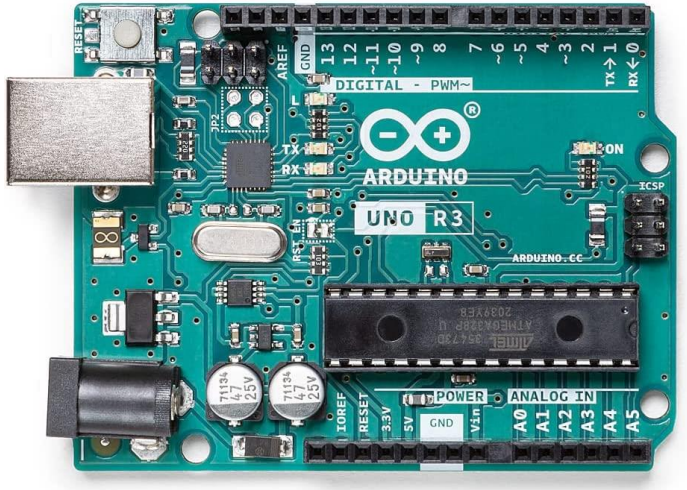
Tâches réalisées

- Package Arduino finalisé sur le Wiki + test de celui-ci
- Package Nucléo finalisé sur le Wiki
- Passage à la réalisation pratique du package Nucléo
- Etude de la répartition des rotors du Mini Bee

Conclusion

Arduino

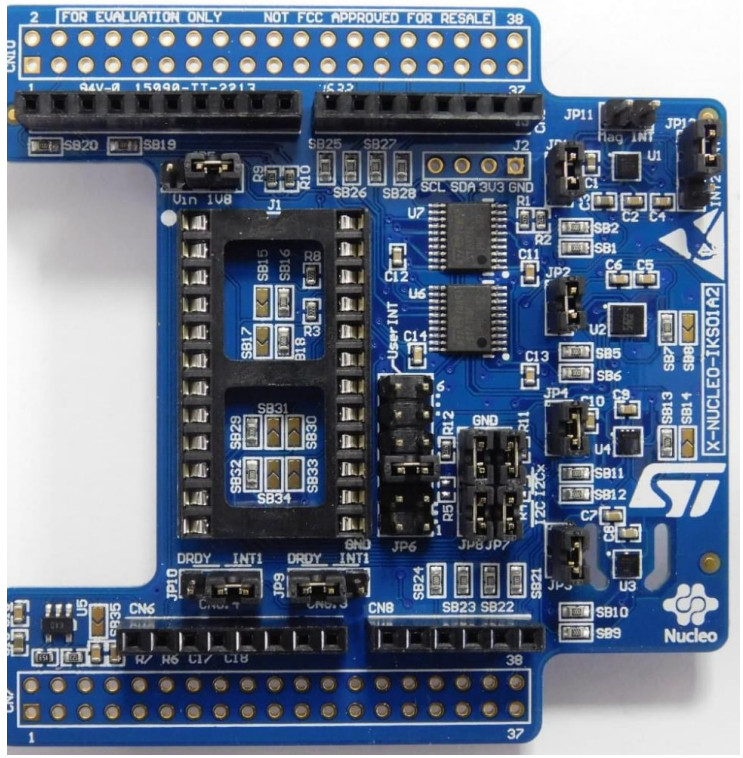
- Réalisation d'un package fonctionnel en 2 langues
- Comprendre le domaine de l'électronique avec un outil simple à utiliser
- Tremplin vers un système plus complexe : Carte Nucléo



Conclusion

Nucleo

- Réalisation du package pédagogique basé sur celui d'Arduino
- Passage entre les composants Arduino aux composants STM réussi
- A approfondir pour rendre le prototype plus réaliste

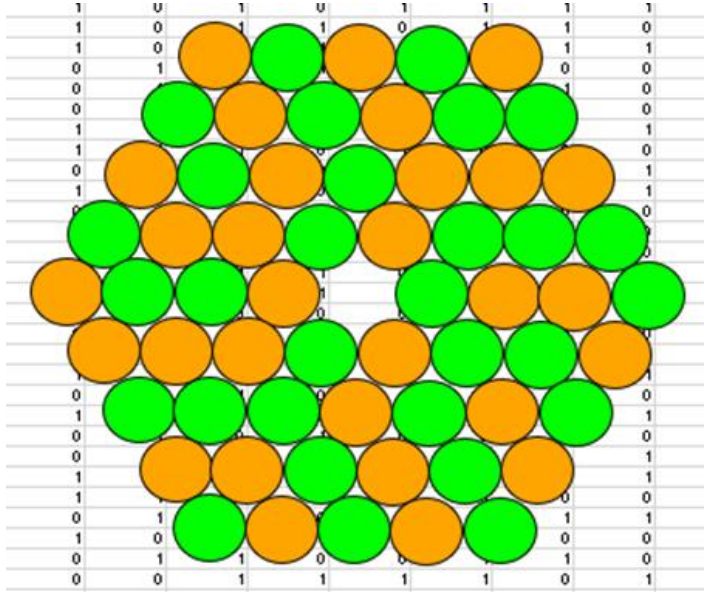


Conclusion

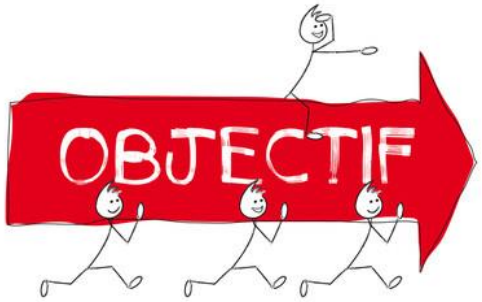
Configuration des rotors

- Configuration la plus optimale pour le contrôle du Mini-Bee
- Les 3 critères de performance bien respectés:

Homogénéité, Stabilité et Symétrie centrale



Continuité du projet



Tâches à réaliser l'année prochaine

- Déterminer si le Mini-Bee peut répondre à une demande de mouvement en lacet, en combien de temps ? Si non, comment pallier ce problème ? Implémentation d'un système permettant de répondre au CdC

- Contrôler un moteur via les cartes Arduino / Nucléo en vu de commander les moteurs des rotors du Mini-Bee

- Etudier les conséquences d'une panne : Augmenter la poussée de quels rotors ? A partir de combien de rotors le système n'est plus fonctionnel / Atterrissage d'urgence ?

Des questions ?

Annexes

Code VBA permettant de répartir les sens de rotation des rotors du MiniBee, les ronds représentant les rotors doivent être nommés comme suit : "Oval i" pour le i ème rotor.

```
Sub ColorOvals()
    Dim i As Integer
    Dim oval As Shape
    Dim cellValue As Integer

    For i = 1 To 60
        cellValue = ActiveSheet.Cells(125, i).Value

        'Vérifie si la valeur de la cellule est 1 ou 0 et affecte la couleur correspondante à l'ovale correspondant
        If cellValue = 1 Then
            Set oval = ActiveSheet.Shapes("Oval" & i)
            oval.Fill.ForeColor.RGB = RGB(255, 165, 0) 'Orange
        ElseIf cellValue = 0 Then
            Set oval = ActiveSheet.Shapes("Oval" & i)
            oval.Fill.ForeColor.RGB = RGB(0, 255, 0) 'Vert
        End If
    Next i
End Sub
```

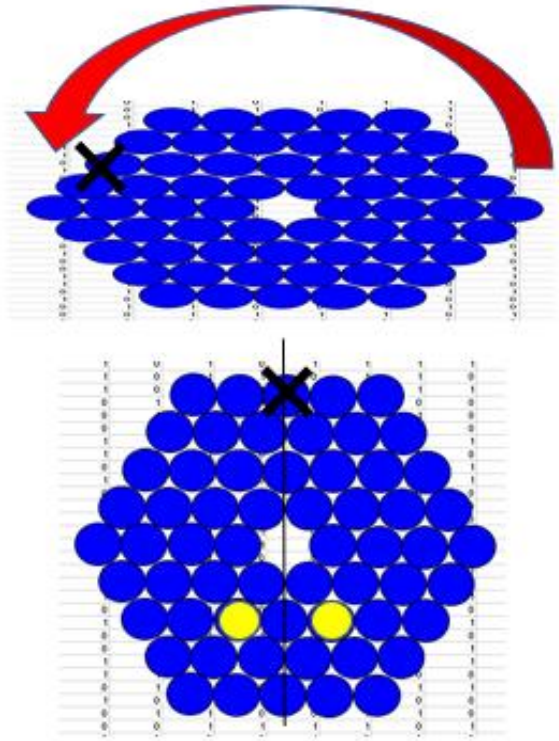
Annexes

Eviter le phénomène de roulis

- Cadre : Panne d'un rotor quelque part sur le Mini-Bee

Conséquences : Mouvement de roulis induit

- Contrainte 1 : On ne peut couper que 1 rotor car équilibre en lacet nécessaire
- Contrainte 2 : On doit couper 1 rotor en sens inverse de celui en panne
- Contrainte 3 : On doit couper un rotor sur la même couronne que celui en panne et celui-ci doit être sur l'axe passant par le rotor en panne et le centre pour équilibrer

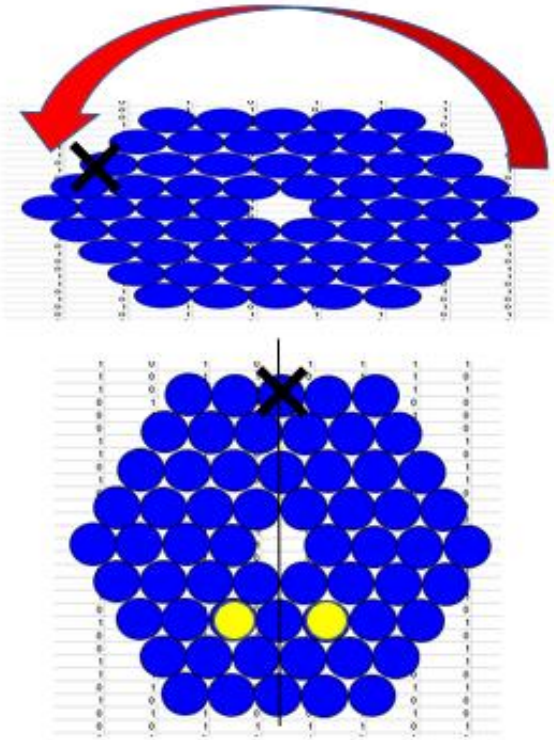


Annexes

On pourrait couper 2 rotors sur la couronne 2 car les 2 rotors ensemble forment le même moment qu'un rotor sur la couronne 4

Sachant que chaque rotor a une force de poussée de 123 N (stationnaire)

Moment selon la position sur les couronnes du mini Bee			
Couronne 4	Couronnes 3	Couronne 2	Couronne 1
492 N	369 N	246 N	123 N



On doit donc couper soit 1 rotor opposé à celui en panne, soit 2 rotors sur la couronnes 2, dont la somme est sur l'axe passant par le rotor en panne et la cabine, soit 1 rotor sur la couronne 3 + 1 sur la couronne 1 (aligné).

Avec toutes ces possibilités on arrive à supprimer le moment de 492 N qui contre la panne du rotor opposé.