

PROJET MINI-BEE

Cahier des charges en vue de la création de la
partie FCU pour l'équipe Cy-Tech

Réalisé par l'équipe ESTACA 4A SQY – Prototypage FCU – Projet n°9

Table des matières

Description du système global :	2
Caractéristiques	2
Compétences voulues	2
Prototypage :	2
FCU et branchements :	3
Description globale du code de notre prototype :	5
Achats finaux et montage de notre prototype.....	6
Liens avec le système Cy-tech :	7
Branchements	8
Code.....	9

Table des figures

Figure 1 - Schéma global du VTOL Mini-bee	2
Figure 2 – Schéma de branchements boutons poussoirs.....	3
Figure 3 – Schéma de branchements joystick	3
Figure 4 - Schéma de branchements gyroscope	4
Figure 5 - Prototype réel Joystick et gyroscope	4
Figure 6 - Schéma de branchements altimètre	5
Figure 7 - Prototype réel Altimètre	5
Figure 8 - Schéma simplifié du code.....	5
Figure 9 - Prototype final obtenu	7
Figure 10 - Schéma simplifié du code en lien avec le projet Cy-Tech	9

Description du système global :

Caractéristiques

- VTOL initialement piloté visant à être autonome
- 2 PAX dont un pilote
- Solution hybride : moteur à piston Rotax 915is alimentant un moteur T-Motor U12KV120 (ou U13)
- 36 rotors
- Facilement démontable (peut se ranger dans une soute d'avion)
- Coûts modérés
- Vocation d'évacuation sanitaire légère

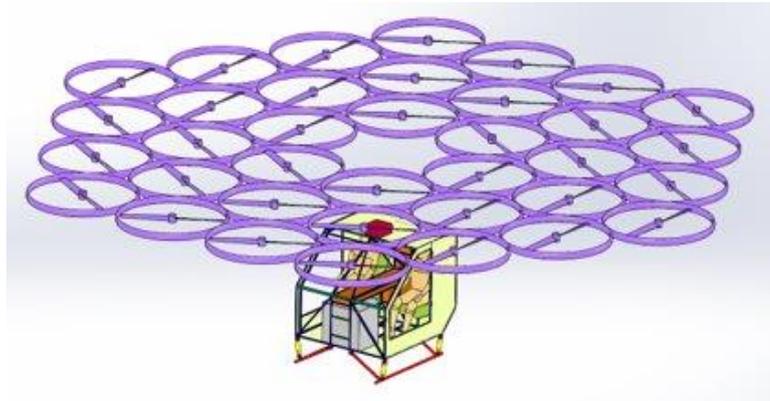


Figure 1 - Schéma global du VTOL Mini-bee

Compétences voulues

- Vitesse de croisière : 170 km/h
- Distance franchissable : 600 km
- Masse maximale au décollage : 700 kg

Prototypage :

Dans le cadre de notre projet, l'objectif est de créer un prototype pour tester les différentes configurations possibles du FCU en laissant de côté plusieurs caractéristiques du système global. Ainsi, le FCU est représenté ici par une carte Arduino sur laquelle seront branchées les entrées (Commandes et capteurs) et les sorties (rotors).

Les rotors sont ici représentés par des LEDs à intensité lumineuse variable de sorte à modéliser les variations de vitesse continues des rotors. Leur nombre sera d'abord limité à 4 puis à 6 et enfin à 9 LEDs pour correspondre avec le nombre de sorties max de la carte Arduino mais on verra que monter jusqu'à 36 ne changerait fondamentalement pas grand-chose.

Pour les entrées du système, nous nous intéresserons d'abord à un pilotage par 4 boutons (axe de tangage et roulis), puis pour se rapprocher du système réel et avoir des entrées de commande continues, nous choisirons de partir sur un joystick. Concernant les différents capteurs, le système devra contenir au minimum un gyroscope, un accéléromètre et un magnétomètre. Un GPS et un altimètre serait également quasi nécessaire pour aider le

pilote. Nous verrons alors dans un premier temps comment implémenter un gyroscope au FCU et en quoi l'implémentation d'un accéléromètre et d'un magnétomètre serait très similaires à tous les points de vue. Concernant l'altimètre, nous avons préféré faire les essais nous-mêmes car les différentes informations fournies (Altitude, pression, température) doivent être traitées différemment que pour les capteurs de vitesse.

L'objectif de cette étude est donc de comprendre comment peut fonctionner le FCU en implémentant ses différentes entrées et sorties sur un système très simplifié puis de repérer les différences et similitudes avec le système réel.

FCU et branchements :

Cette première configuration représente les branchements pour un système avec 4 LEDs et 2 boutons poussoirs (avec chacun une position monter et une position descendre). Les entrées comme les sorties doivent être branchées sur des ports PWM pour permettre une information continue.

Ici, on utilise des résistances 10k Ohm et des condensateurs 10 nF mais cela dépend des LEDs et des boutons choisis. Néanmoins, les boutons que nous avons utilisés comportent déjà leurs propres résistances et condensateurs.

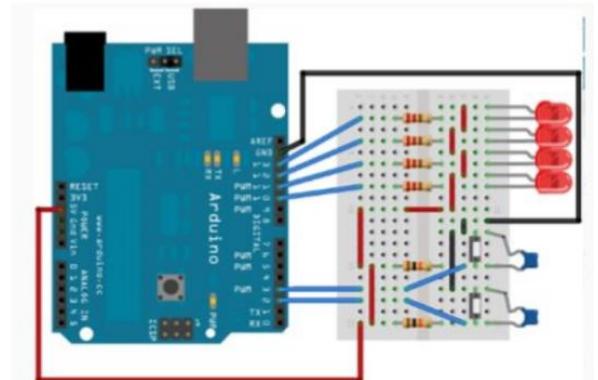


Figure 2 – Schéma de branchements boutons poussoirs

L'introduction d'un joystick se fait alors assez facilement. En effet, la plupart des joysticks se comportent finalement comme 2 boutons continus. Comme on peut voir sur le schéma ci-contre, les 4 sorties du joystick sont branchées sur les ports PWM de la carte Arduino comme c'était le cas pour le système précédent. A noter que le joystick choisi ici comporte aussi un bouton poussoir (lorsque l'on appuie dessus) qui peut être très utile pour commander une radio ou se déplacer sur le système informatique du drone. Sa sortie est branchée sur un port *Digital In* puisqu'elle n'est pas continue.

Remarque : le joystick sur le schéma prend son énergie directement sur la carte Arduino mais beaucoup ont besoin d'être branchés sur une source externe.

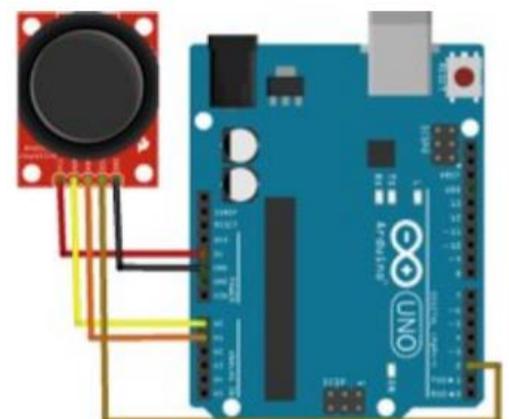


Figure 3 – Schéma de branchements joystick

De même, l'introduction d'un gyroscope au FCU se fait d'une manière assez similaire à celle d'un joystick. En effet, le gyroscope renvoie en sortie 4 signaux continus représentant les vitesses rotationnelles selon les axes de roulis et de tangage. Il suffit de brancher ces sorties sur des ports PWM de la carte arduino et de traiter l'information comme celle d'une entrée commande pour agir sur notre système.

Si le nombre de ports entrées le permet le branchement d'un joystick et d'un gyroscope ne pose pas plus de problème. Il suffit dans la partie code de traiter les différentes informations comme on le souhaite.

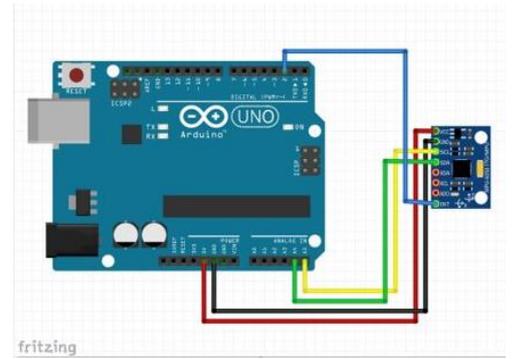
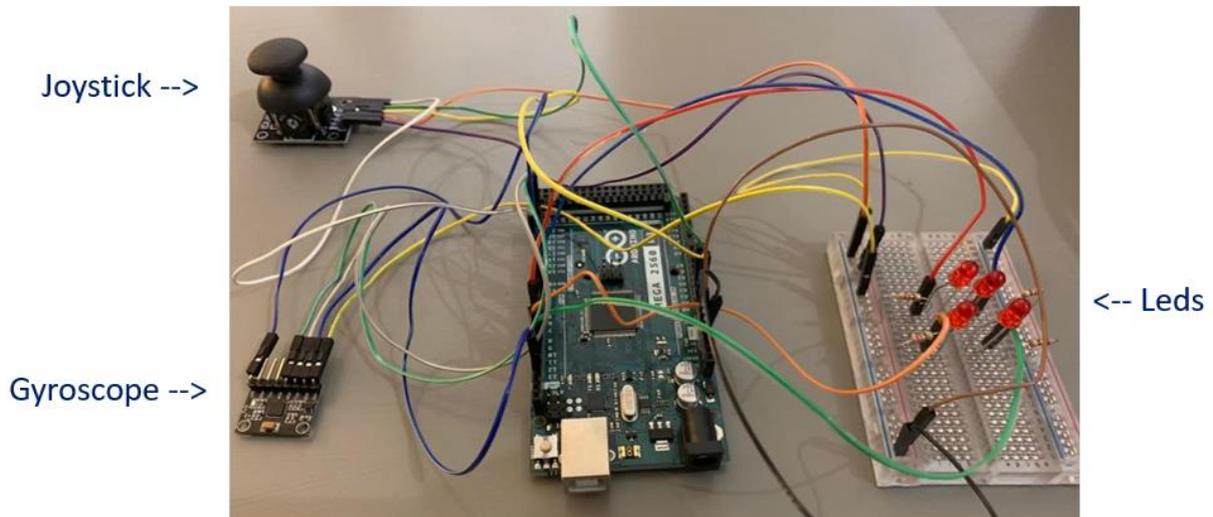


Figure 4 - Schéma de branchements gyroscope



Montage actuel pour 4 LEDs (testé et fonctionnel)

Figure 5 - Prototype réel Joystick et gyroscope

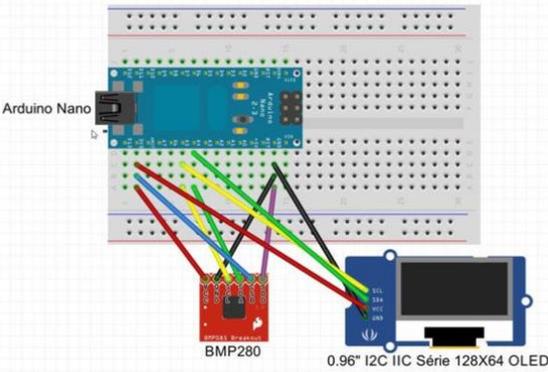


Figure 6 - Schéma de branchements altimètre

L'implémentation d'un altimètre se fait légèrement différemment. En effet, il renvoie en sortie qu'un seul signal continue contenant 2 informations : la température et la pression atmosphérique. Grâce à ça, un microcontrôleur calcule l'altitude et renvoie ces 3 informations sur un petit écran. Les autres ports du capteur servent à son alimentation, à l'entrée d'information dans le capteur ainsi qu'à l'entrée d'une clock. Passé cela, les branchements sont assez similaires à ceux vus précédemment : la

partie signal continue est branchée sur des port PWM de la carte arduino ainsi que sur les ports du microcontrôleur-écran. Le FCU a alors connaissance de la température, de la pression atmosphérique et de l'altitude à tout instant.



Figure 7 - Prototype réel Altimètre

Description globale du code de notre prototype :

Etape 1 : Ajout des librairies nécessaires à l'utilisation des différentes fonctions appelées dans le code.

Etape 2 : Initialisation de chaque port et définition de sa nature (entrée ou sortie)

Etape 3 : Boucle continue modélisant la puissance de chaque moteur avec une intensité lumineuse en fonction des variations du joystick et du gyroscope (code pour le gyroscope : [HowToMechatronics](#)). On récupère les 2 coordonnées de position du joystick et on définit une intensité lumineuse pour la position neutre. L'augmentation d'une des coordonnées, par exemple

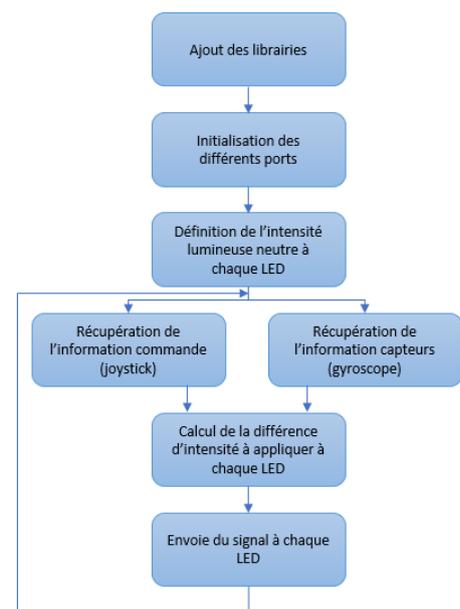


Figure 8 - Schéma simplifié du code

pousser le manche vers l'avant, va modifier les intensités lumineuses de tout le prototype, dans cet exemple les LEDs arrières s'intensifient pendant que les LEDs avants réduisent.

Etape 3' : Pour passer des LEDs aux moteurs réels, les coefficients doivent être adaptés. On ne peut pas faire varier un moteur de 0 à 255 comme une LED car le moteur ne peut pas avoir une intensité nulle (déterminer en amont la vitesse de rotation minimale de chaque hélice)

Achats finaux et montage de notre prototype

- **Arduino MEGA** (32€42 HT / 38€90 TTC) : <https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm>
- **12 LEDs** (1€50 HT / 1€80 TTC) : <https://www.gotronic.fr/art-led-5-mm-rouge-l53it-252.htm>
- **20 résistances 220 Ohms (achat par multiple de 10 seulement à 2 paquets)** (0€33 HT / 0€40 TTC) : <https://www.gotronic.fr/art-10-resistances-1-4w-220-8486-2568.htm>
- **Câbles électriques male male : 1 paquet de 40** (3€29 HT / 3€95 TTC) : <https://www.gotronic.fr/art-nappe-de-40-cables-de-20-cm-m-m-bbj10-21989.htm>
- **Câbles électriques femelle femelle et connexion : 1 paquet de 13** (2€08 HT / 2€50 TTC) : <https://www.gotronic.fr/art-nappe-de-13-cables-de-connexion-f-f-bbj22-28411.htm>
- **1 platine d'essai** (3€75 HT / 4€50 TTC) : https://www.gotronic.fr/art-plaque-de-montage-rapide-22812.htm#complte_desc
- **1 joystick** (2€42 HT / 2€90 TTC) : <https://www.gotronic.fr/art-module-joystick-gt1079-26129.htm>
- **1 MPU 6050** (3€25 HT / 3€90 TTC) : <https://www.gotronic.fr/art-module-6-dof-sen-mpu6050-31492.htm>
- **2 modules « bouton-poussoir »** (3€17 HT / 3€80 TTC) : <https://www.gotronic.fr/art-module-bouton-poussoir-st043-26115.htm>

Total GoTronic : 56€29 HT / 67€55 TTC (tous les composants ont été achetés sur GoTronic)

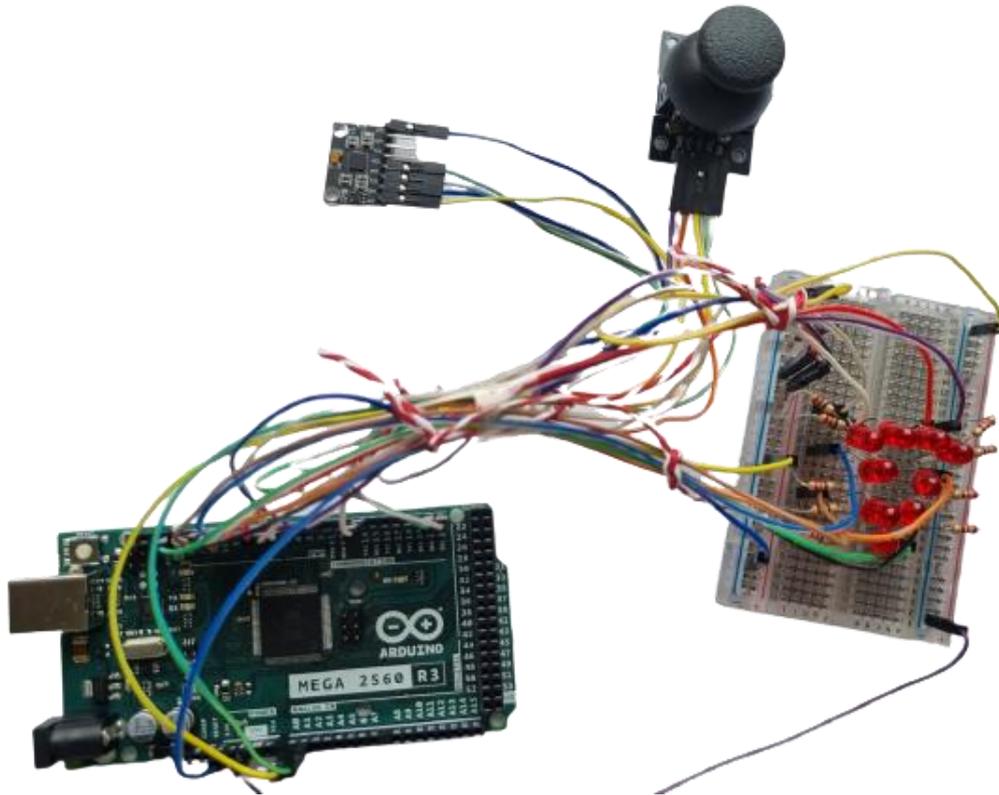


Figure 9 - Prototype final obtenu

Vidéos de test :

Les vidéos de test sont disponibles sur YouTube :

- Joystick : <https://youtube.com/shorts/4b19bfShKaE>
- Gyroscope : <https://youtube.com/shorts/59HkFdXGF2Y>
- Joystick et gyroscope : <https://youtube.com/shorts/pbAvSrB5eRQ>

Les vidéos sont également disponibles sur le serveur BP01 : /estaca/mb-202109-EstacaSQY4A-FCU/Vidéos_Tests_Prototype

Liens avec le système Cy-tech :

Alors que nous utilisons une carte Arduino MEGA pour réaliser notre prototype, l'équipe de Cy-Tech a choisi une carte STM32 F401 RE (Nucleo).

Nous allons donc voir quelles sont les différences notables entre ces deux cartes et la façon dont nous devons les utiliser.

Branchements

Tout d'abord, comparons les cartes et leurs ports.

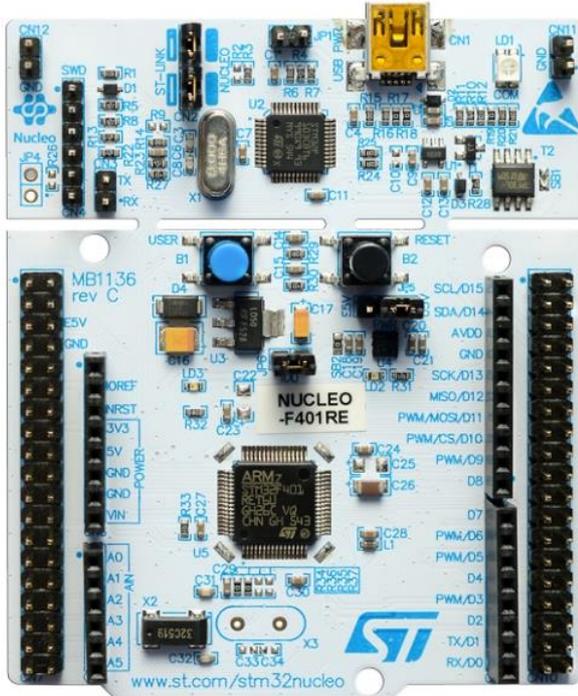


Figure 10 - Carte STM32 F401 RE (Nucleo)

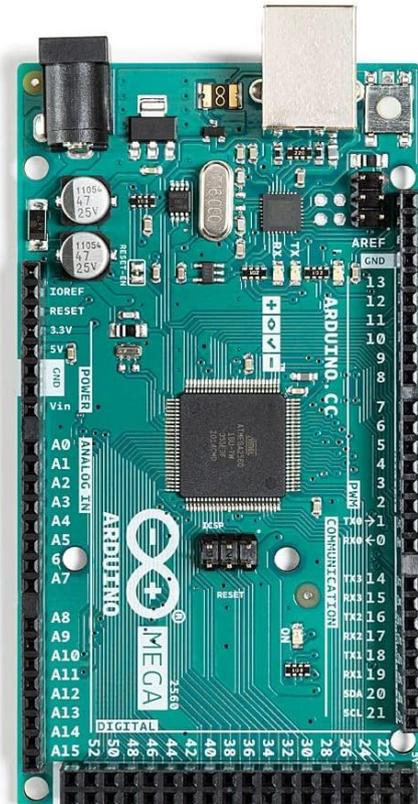


Figure 9 - Carte Arduino MEGA

On peut alors constater que ports sont assez similaires.

Pour rappel dans la carte Arduino MEGA, nous utilisons :

- 9 ports PWM
- 1 port digital ou PWM pour le MPU 6050
- 1 port digital ou PWM pour le joystick
- Les ports SDA et SCL pour le MPU 6050
- Les ports GND et 5V (1 seul suffit pour chaque grâce à la breadboard)
- Les ports A0 et A1 pour le joystick

Tous ces ports sont présents également sur la carte STM32, sauf pour le nombre de port PWM. Néanmoins, après [recherches](#), on apprend que la carte Nucleo comporte 10 broches PWM (A2, A3, D15, D14, D12, D11, D10, D9, D5 et D4).

On en conclue donc que le branchement s'effectuera de la même façon sur les deux cartes.

Code

La carte STM32 peut être configuré sur l'IDE Arduino. Ainsi, pour un prototype simplifié comme le nôtre, les différences seront seulement sur le choix de la carte dans le logiciel Arduino et l'implémentation des bibliothèques. Néanmoins, les choix futurs de joystick, de volant, de gyroscope, d'altimètre feront varier le code. Également, pour le prototype réel, il faudra utiliser 2 cartes, où chacune des cartes est connectée à 9 couples de rotors (2 moteurs électriques, chacun faisant tourner 18 rotors, jumelé deux à deux).

Toutefois, l'idée restera la même : implémenter la librairie du composant, l'étudier, et utiliser son langage pour le configurer.

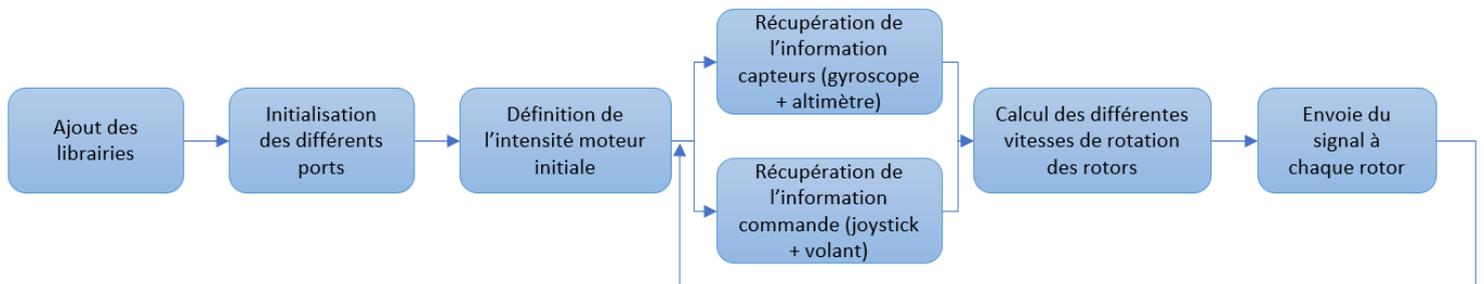


Figure 10 - Schéma simplifié du code en lien avec le projet Cy-Tech