2018

Le projet Drone

LICENCE PROFESSIONNELLE MECSE

GUILLOUCHE KENNY

IELKIN VADIM

DELAFOSSE NATHAN

UNIVERSITÉ PARIS SUD

IUT DE CACHAN

Introduction

En France, six à huit millions de personnes, soit entre 12 et 13 % de la population, sont touchées par des problèmes d'audition, une partie est due notamment aux non-respects des normes d'intensité sonore dans les concerts. Depuis novembre 2017, les décibels ne doivent pas dépasser les 102db lors d'un événement et des mesures doivent être réalisées avant et pendant chaque représentation. Or les mesures prises pendant le concert sont réalisés à un seul endroit et les résultats pour la salle sont des approximations.

De plus la qualité visuelle des écrans est un point à vérifier afin de garantir une bonne visibilité pour les spectateurs éloignés notamment durant les événements sportifs ou musicaux.

Afin de réaliser des mesures plus précises, répondant aux normes et aux besoins, nous avons décidé de prendre comme système un drone et d'y ajouter des capteurs. Le drone va permettre notamment d'accéder rapidement aux éléments en hauteur.

Table des matières

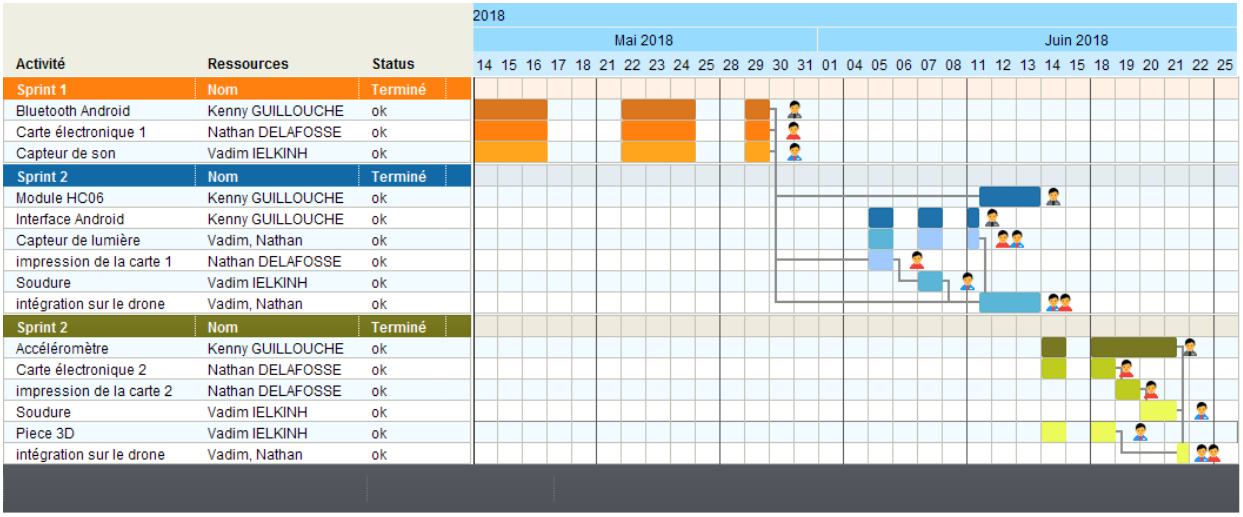
Cahier des charges

Le but du projet est d'instrumenter un drone, dans ce cadre il nous faut :

- Programmer plusieurs capteurs
- Intégrer les capteurs au drone (via une carte électronique)
- Créer une interface utilisateur permettant de de visualiser l'ensemble des donner récolter par les capteurs (via une application Android)
- Transférez-les donner des capteurs a l'application via une communication sans fil

Répartition du travail

Pour ce projet nous avons mis en place la méthode agile, nous avons donc dû nous répartir le travail sous forme de sprint.



Activité	Ressources	Status
Sprint 1	Nom	Terminé
Bluetooth Android	Kenny GUILLOUCHE	ok
Carte électronique 1	Nathan DELAFOSSE	ok
Capteur de son	Vadim IELKINH	ok
Sprint 2	Nom	Terminé
Module HC06	Kenny GUILLOUCHE	ok
Interface Android	Kenny GUILLOUCHE	ok
Capteur de lumière	Vadim, Nathan	ok
impression de la carte 1	Nathan DELAFOSSE	ok
Soudure	Vadim IELKINH	ok
intégration sur le drone	Vadim, Nathan	ok
Sprint 2	Nom	Terminé
Accéléromètre	Kenny GUILLOUCHE	ok
Carte électronique 2	Nathan DELAFOSSE	ok
impression de la carte 2	Nathan DELAFOSSE	ok
Soudure	Vadim IELKINH	ok
Piece 3D	Vadim IELKINH	ok
intégration sur le drone	Vadim, Nathan	ok

Figure 1 : Diagramme de GANTT

1. Les capteurs

Le contexte de notre projet, nous amenaient donc à implémenter et interfacer plusieurs capteurs sur le drone dans le but d'effectuer des mesures. Les premières normes que nous devons vérifier sont celles relatives aux sons.

1.1. Le capteur de son

Dans un premier temps le capteur qui nous semblaient essentiel était un capteur de son. Pour ce faire nous avions à notre disposition le grove sound sensor (cf : figure 2). Le capteur est basé sur un amplificateur LM386 et un micro électret. Le dispositif est un capteur analogique, le micro électret génère une tension comprise 0 et 3,3V sur la pin Vout.

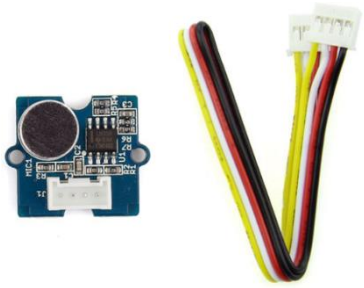


Figure 2 : Capteur de son (Grove SEN12945P)

Dans un premier temps, quelques précisions étaient à déterminer pour pouvoir faire l'interfaçage.

- La tension d'entrée est de 3 .3 à 5V (tension utilisée 3.3V)
- L'intensité sonore maximale du micro (52 dB)

La première observation qu'on peut émettre c'est que le capteur n'est pas adapté à l'utilisation de mesures de hauts volumes sonores lors des concerts. Mais dans le cadre de notre projet nous devions réaliser seulement le prototypage du dispositif.

L'objectif de ce module de contrôle sonore était de récupérer les valeurs du son et de les transmettre grâce au module Bluetooth HC06.

Le bruit des pales du drone vient parasiter le micro-électret, nous avons pensé donc à isoler le capteur. Deux solutions s'offrent à nous :

- Isoler le bruit des pales de façon logicielle
- Concevoir une pièce 3D permettant d'éliminer le bruit parasite

La solution que nous avons choisie était de concevoir une pièce en 3D, celle-ci va permettre de capter le son de façon directionnelle et ainsi supprimer en grande partie le bruit alentour.

Pour le design nous nous sommes inspirés des anciens appareils auditifs.

Ainsi nous avons créé la pièce suivante :

Figure 3 : Appareil auditif

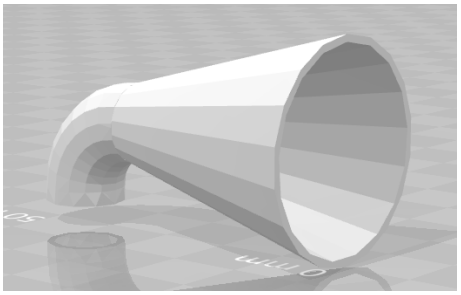


Figure 4 : Cône directionnel

Le capteur est fonctionnel et enlève une partie du bruit des pales mais proche d'un objet les pales viennent faiblement perturber nos mesures, on en conclut qu'un traitement devra être réalisé plus tard afin de garantir la véracité du contrôle des normes.

1.2. Le capteur de luminosité

Le contexte du projet étant le contrôle de norme, une vérification de l'éclairage semble logique pour que la luminosité des écrans projecteurs soit adaptée et permette aux spectateurs de profiter du spectacle dans les meilleures conditions. Nous avions à notre disposition le capteur analogique Grove luminescence sensor, basé sur le **APDS-900** (cf : figure 5).

La tension de sortie renvoie des valeurs comprises entre 0-2.3V. Plus la tension est élevée plus la luminosité est forte.

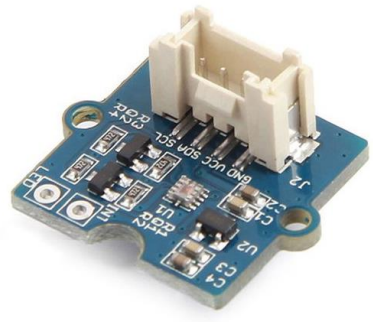


Figure 5 : Capteur de luminosité (APDS-900)

1.3. L'accéléromètre

Les capteurs étant implantés, nous avions besoin prouver un maximum la véracité des mesures effectuées. Un des points faibles du drone est sa stabilité, si le drone n'est pas stable les valeurs mesurées seront faussées. Pour connaitre la stabilité du drone nous somme parti un accéléromètre.

L'accéléromètre ADXL345 (cf : Figure 6) est celui que nous avons implémenté, celui-ci fonctionne par I2C.

Figure 6 : Accéléromètre (ADXL345)

Comme il fonctionne par I2C afin de pouvoir communiquer avec lui il faut récupérer son adresse et l'adresse des registres qui nous intéressent.

Dans la documentation, on nous indique que son adresse est 0x53. Cette adresse est codée sur 7 bits or sur Mbed les librairies I2C du microcontrôleur requièrent une adresse sur un octet. Celle-ci devient donc 0xA6.

Ensuite pour la documentation nous donnent les différents registres des axes :

0x32	50	DATAX0	R	00000000	X-Axis Data 0
0x33	51	DATAX1	R	00000000	X-Axis Data 1
0x34	52	DATAY0	R	00000000	Y-Axis Data 0
0x35	53	DATAY1	R	00000000	Y-Axis Data 1
0x36	54	DATAZ0	R	00000000	Z-Axis Data 0
0x37	55	DATAZ1	R	00000000	Z-Axis Data 1

Figure 7 : Adresse des registres des axes de l'accéléromètre

Le premier programme est alors créé pour faire fonctionner ce capteur.

Cependant le capteur propose plusieurs modes de fonctionnement et demande alors une phase d'initialisation sur différents registres.

Après lecture plus approfondie de la documentation voici les registres qu'il faut initialiser :

- Le premier registre permet de réveiller le capteur et de choisir le mode.

0x2D	45	POWER_CTL	R/$\overline{W}$	00000000	Power-saving features control

Register 0x2D—POWER_CTL (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	Link	AUTO_SLEEP	Measure	Sleep	Wakeup	

Figure 8 : Registre POWER_CTL

Il faut donc écrire sur ce registre, 1 pour la measure et 0 sur les deux bits de Wakeup pour le réveiller ce qui donne 0x08.

- Le deuxième registre permet de régler le format des mesures.

0x31	49	DATA_FORMAT	R/W	00000000	Data format control

Register 0x31—DATA_FORMAT (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	Justify	Range	

Table 21. *g* Range Setting

Setting		
D1	D0	*g* Range
0	0	±2 *g*
0	1	±4 *g*
1	0	±8 *g*
1	1	±16 *g*

Figure 9 : Registre DATA_FORMAT

Pour ce registre seul les 4 premiers bits sont importants car ils correspondent au choix du format de mesure, nous réglons donc le format en 16g (étendue de mesure maximum) et en full résolution ce qui correspond à une écriture de 0x0B.

- Le dernier registre à configurer est la vitesse de transmission du capteur.

0x2C	44	BW_RATE	R/$\overline{W}$	00001010	Data rate and power mode control

Register 0x2C—BW_RATE (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	Rate			

Figure 10 : Registre BW_RATE

Afin d'avoir un maximum de précision sur le capteur, nous écrivons sur le registre la valeur 0x0F ce qui correspond à une fréquence de 3200 Hz en transmission.

1.4. Le module Bluetooth

Maintenant que nos capteurs fonctionnent il faut pouvoir envoyer les informations à l'utilisateur.

Pour cela on utilise le module Bluetooth HC06 (cf : figure 11).

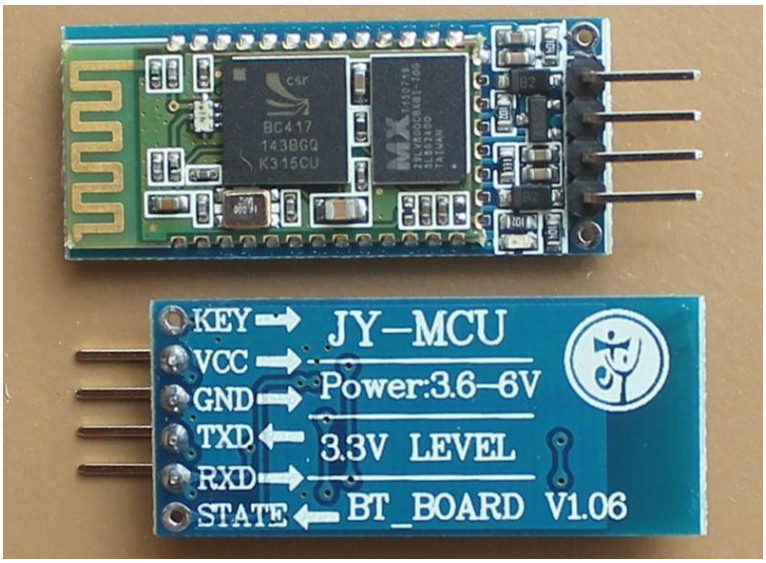


Figure 11 : Module Bluetooth HC06

A savoir que pour envoyer nos données de notre microcontrôleur au module il faut croiser les fils de transmissions et réceptions.

Pour tester la connexion il faut utiliser un picoscope et préciser que le type de communication que l'on veut observer et la vitesse de transmission.

2. La supervision des données sur Android

Une grande partie du projet concerne la partie récupération des données à distance, le but de cette partie est de récupérer les données des capteurs afin de les mettre à disposition à l'utilisateur.

Pour réaliser cela le projet se découpe en plusieurs étapes :

- Etablir la connexion entre les appareils
- Récupérer les données et les traiter
- Créer une interface afin de les mettre à disposition pour l'utilisateur

2.1. Etablir la connexion entre les appareils

Afin d'établir une connexion à distance plusieurs solutions sont possibles :

- Le bluetooth
- Le Wi-Fi
- La 3G-4G

La connexion a besoin de transmettre des données de faible taille et n'a pas besoin d'une grande portée, de plus la consommation d'énergie doit être la plus faible possible afin d'augmenter l'autonomie du système.

Le bluetooth est la solution que nous avons retenue.

Le bluetooth sur Android est particulier, pour créer une connexion bluetooth il faut suivre les étapes suivantes :

- Regarder la présence du bluetooth sur l'appareil
- Demander à l'utilisateur d'activer le bluetooth
- Mettre l'appareil en visible
- Récupérer la liste des appareils
- Récupérer l'appareil choisi par l'utilisateur
- Etablir la connexion avec le serveur
- Maintenir la connexion
- Etablir les échanges (socket)
- Maintenir les échanges (1er thread)

Les différents codes relatifs à ces étapes sont disponibles sur le programme commenté.

A savoir que le Bluetooth a évolué par rapport aux tutoriels présents sur internet, il faut stocker l'intégralité du BluetoothDevice choisit afin d'établir la connexion sinon celle-ci ne fonctionnera pas, il faut également rajouter les permissions suivantes :

```
int MY_PERMISSIONS_REQUEST_ACCESS_COARSE_LOCATION = 1;
ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
        MY_PERMISSIONS_REQUEST_ACCESS_COARSE_LOCATION);
```

Ne pas oublier également de bien fermer la connexion et les Thread à la fermeture de l'application.

IMPORTANT : Pour tester la connexion et la connexion et la réception de données, nous avons testé notre application avec un autre téléphone contenant l'application Bluetooth terminal cette application permet de dire si on est connecté à son Bluetooth serveur et permet également d'envoyer des données et de tester la réception de données sur notre application.

2.2. Récupérer les données et les traiter

Une fois les échanges établis, on va récupérer les données dans un buffer puis les transformer en chaine de caractère.

Deux problèmes se posent :

- Comment savoir à quels capteurs la trame correspond ?
- Comment afficher les valeurs pour l'utilisateur ?

Pour répondre à la première problématique, nous avons décidé de mettre en place un protocole de communication propre à notre système.

Le protocole est le suivant :

NuméroDuCapteur : DataDuCapteur

Le numéro du capteur va permettre d'identité la trame, un peu comme un ID en Ethernet.

Les deux points vont permettre de séparer les informations.

Ce protocole devra être respecté lors de l'envoi des données sur Mbed et au traitement des données sur Android studio pour pouvoir fonctionner correctement.

Le deuxième problème est l'affichage des données pour l'utilisateur, la connexion étant permanente nous sommes obligés de passer par une interruption(Handler) ou par un Thread d'affichage afin de pouvoir afficher sans faire « planter » l'application.

Nous avons décidé de passer par un Thread car il permet d'éviter la perte de connexion entre les appareils.

De plus via le Thread à l'aide d'une mise en sommeil nous contrôlons la fréquence d'affichage, ainsi nous avons été amenés à devoir synchroniser les envois et les réceptions des données en appliquant une fréquence d'affichage plus élevé que la fréquence d'envoi.

Ainsi on garantit de ne pas perdre de données dans la communication.

2.3. Interface Utilisateur

L'interface utilisateur doit contenir la connexion et l'affichage des données des capteurs.

On a donc réalisé l'interface suivante :

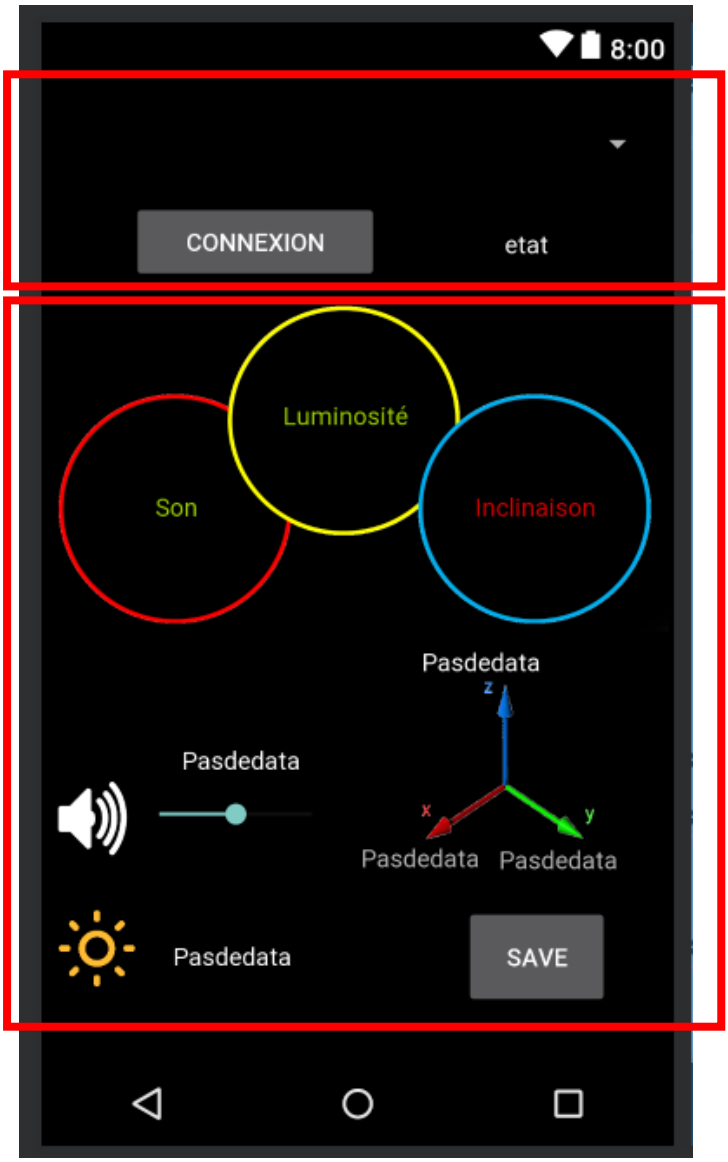


Figure 12 : Interface utilisateur

3. Création des cartes électroniques

Les différents capteurs doivent être implémentés sur le drone, pour ce faire, nous avons dessiné une carte électronique via Altium. Dans un premier temps, un prototype n'utilisant qu'un seul capteur, a été conçu pour un drone de faible de taille. Cela a nécessité une première carte qui a été l'occasion d'apprendre à utiliser le logiciel. Cette carte a été pensée pour être fixe facilement au drone et pour pouvoir embarquer un microcontrôleur, un module Bluetooth et un capteur d'intensité sonore.

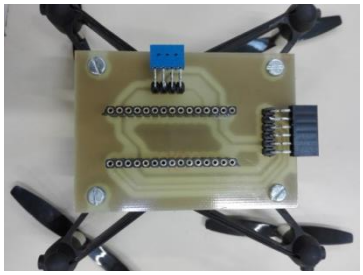


Figure 13 : Carte prototype du petit drone

Après avoir confirmé le fonctionnement de ce premier prototype, nous avons choisi d'autres capteurs à implémenter et réfléchis à leur intégration sur le drone BEBOP. Pour permettre au drone de voler correctement il nous a fallu nous assurer que ses capteurs intégrés ne soient pas obstrués. De plus nous avons placé nos capteurs de façon à rendre leur fonctionnement optimal. Cette carte prévoit donc le placement de 3 capteurs d'un microcontrôleur et d'un module Bluetooth. Contrairement à la première carte, relativement simple, le nombre et la disposition des capteurs ont nécessité une carte à deux couches.

Un problème a été rencontrer du fait de ces doubles face. En effet la conception de la carte implique d'effectuer des soudures des deux cotés de la carte, ce qui faute matériel adapté et de compétence en soudure c'est avéré complexe. Nous avons donc opté pour une solution plus simple consistant à souder des fils pour remplacer les routage faisant défauts. On peut voir ci-dessous le resultat fianal de la carte après soudures.

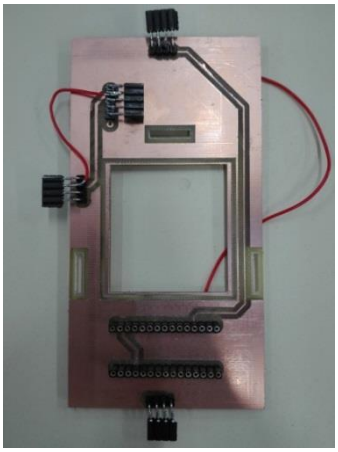


Figure 15 : Face avant de la carte

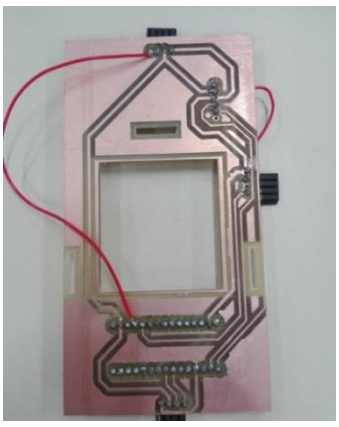


Figure 14 : Face arrière de la carte

Conclusion

Ainsi, nous avons accompli les objectifs du cahier des charges. En effet, nous avons intégré 3 capteurs différents au drone et nous récupérons ses données via Bluetooth sur une application Android.

Cependant il y a encore des possibilités d'améliorations. On pourrait par exemple optimiser le poids de l'ensemble capteur, carte électronique, la répartition du poids sur le drone ou encore changer les capteurs pour gagner en précision et en pertinence de mesure. En effet le contexte que nous avons déterminé implique, par exemple, de pouvoir mesure des sons supérieurs à 110dB, or le capteur actuel ne permet pas de mesurer des sons à plus de 53 dB. Il reste donc des possibilités d'améliorations pour ce projet.

Table des références