

DRONE INSTRUMENTE

Etudiants: Lionel Medzo Owono | Marc Olasek | Julien To
Tuteur: Xavier Mininger



Licence MECSE
Année 2018/2019

Sommaire

I-	Présentation du projet :	1
I.1-	Cahier des charges	1
I.2-	Composants.....	1
II-	Les capteurs.....	1
II.1-	Capteur de son	1
II.2-	Capteur de lumière	2
II.3-	Capteur de température et d'humidité	3
II.4-	Capteur GPS	4
III-	Microcontrôleur (Nucleo).....	5
III.1-	Communication I2C.....	5
III.2-	Schéma électrique.....	6
IV-	Communication Bluetooth	7
IV.1-	Module BT HC05	7
V-	Android.....	7
V.1-	Présentation de l'IHM	7
V.2-	Les classes utilisées	10
VI-	Commande par gestes.....	11
VI.1-	Capteur de gestes	11
VI.2-	SDK Parrot	11
VI.3-	l'IHM de l'SDK Parrot	12
VI.4-	Les classes utilisées	13
	Les axes d'amélioration.....	14
	Lien GitHub.....	14
	Tables des Figures	15

I- Présentation du projet :

I.1- Cahier des charges

Ce projet qui nous a été proposé par M.Mininger consiste en deux objectifs :

- Objectif 1 : Instrumenter un drone Bebop de la marque Parrot, afin de récupérer des informations provenant de plusieurs capteurs (montés sur le drone) et de les afficher sur une interface Android. Ce qui permettrait de récupérer des valeurs dans des endroits difficilement atteignables.
- Objectif 2 : Piloter ce même drone avec des gestes effectués par un utilisateur.

I.2- Composants

Ci-dessous la liste des composants utilisés pour notre projet :

- Drone Bebop 2 (objectif 1 et 2)
- Capteurs :
 - Capteur son – Grove Sound (objectif 1)
 - Capteur lumière – Grove Light (objectif 1)
 - Capteur température/humidité – Grove Temperature (objectif 1)
 - Capteur GPS – MT3329 (objectif 1)
 - Capteur de geste – Grove Gesture (objectif 2)
- Module Bluetooth – HC05 (objectif 1 et 2)
- 3x Nucleo – F303K8 (deux Nucleo pour l'objectif 1 et une Nucleo pour l'objectif 2)
- Batterie externe (objectif 1 et 2)

II- Les capteurs

Nous allons dans cette partie présenter les capteurs du premier objectif.

II.1- Capteur de son

Le premier capteur que nous avons programmé est un capteur de son.



Figure 1 Capteur son

Caractéristiques :

- 4 broches (GND, VCC, NC, SIG)
- Tension d'alimentation : 5V
- Une broche non connectée

L'information reçu est de type analogique, elle est lue sur la broche SIG (Signal) du capteur.

On peut voir ci-dessous un exemple de valeurs reçus.

0.73

0.73

0.74

0.75

Figure 2 Valeur capteur son

Les valeurs que nous obtenons varie entre 0 et 1 car nous récupérons l'information sur une broche analogique.

II.2- Capteur de lumière

Le second capteur que nous avons utilisé est un capteur de lumière. Il va nous donner la valeur en Lux de l'environnement où il se trouve.



Figure 3 Capteur luminosité

Caractéristiques :

- 4 broches (GND, VCC, NC, SIG)
- Tension d'alimentation : 5V
- Une broche non connectée

L'information reçu est de type analogique, elle est lue sur la broche SIG (Signal). Pour obtenir la valeur de la luminosité, on récupère la tension de sortie pour la convertir en lux.

On peut voir ci-dessous un exemple de valeurs reçus.

```
la luminance est de : 1000.00 lx
la luminance est de : 1000.00 lx
la luminance est de : 876.92 lx
la luminance est de : 835.65 lx
la luminance est de : 759.71 lx
la luminance est de : 907.69 lx
```

Figure 4 Valeur capteur lumière

II.3- Capteur de température et d'humidité

Dans la suite de nos capteurs présenté, nous utilisons un capteur de température et d'humidité.



Figure 5 Capteur de température et d'humidité

Caractéristiques :

- 4 broches (GND, VCC, NC, SIG)
- Tension d'alimentation : 5V
- Une broche non connectée

L'information reçu est de type analogique, elle est lue sur la broche SIG (Signal). Pour lire l'information nous avons utilisé une bibliothèque MBED, qui permet de lire le signal et de le convertir en température et en humidité.

On peut voir ci-dessous un exemple de valeurs reçus.

Temperature	30.00	Humidite	40.00 %
Temperature	30.00	Humidite	39.00 %
Temperature	30.00	Humidite	40.00 %
Temperature	30.00	Humidite	40.00 %
Temperature	30.00	Humidite	40.00 %
Temperature	30.00	Humidite	41.00 %
Temperature	30.00	Humidite	42.00 %
Temperature	31.00	Humidite	42.00 %
Temperature	31.00	Humidite	42.00 %
Temperature	30.00	Humidite	43.00 %

Figure 6 Valeur capteur de température et d'humidité

II.4- Capteur GPS

Nous utilisons aussi un capteur GPS qui nous permet de localiser notre drone Bebop 2. Pour ce faire, nous avons choisi le capteur GPS « Digilent Pmod GPS (PA6H0Y1438) ».



Figure 7 GPS

Caractéristiques :

- 4 broches (GND, VCC, TX, RX)
- Tension d'alimentation : 3V3
- Vitesse : 9400 Bauds (par défaut)

Le capteur renvoie 5 différentes trames:

- GPGSA
- GPGSV
- GPRMC
- GPVTG
- GPGGA

Dans notre cas nous avons besoin d'une trame qui récupère la longitude, la latitude ainsi que l'altitude. Pour cela nous avons utilisé la trame GPGGA.

Voici ci-dessous un tableau détaillant la trame :

Example	Description
\$GPGGA	Message ID
064951.000	UTC Time (hhmmss.sss)
2307.1256	Latitude (ddmm.mmmm)
N	N/S indicator
12016.4438	Longitude (dddmm.mmmm)
E	E/W indicator
1	Position Fix Indicator
8	Satellites used
0.95	HDOP
39.9	MSL Altitude
M	Units
17.8	Geoidal Separation
M	Units
	Age of Diff. Corr.
*65	Checksum
<CR><LF>	End of message indicator

Figure 8 GPGGA

III- Microcontrôleur (Nucleo)

III.1- Communication I2C

Pendant le projet nous nous sommes rendu compte que nous ne pouvions pas utiliser deux capteurs (Bluetooth et GPS) utilisant l'UART de notre microcontrôleur. En effet, sur la Nucleo nous n'avons qu'un seul UART de disponible, étant donné que le premier est réservé au port USB de la carte.

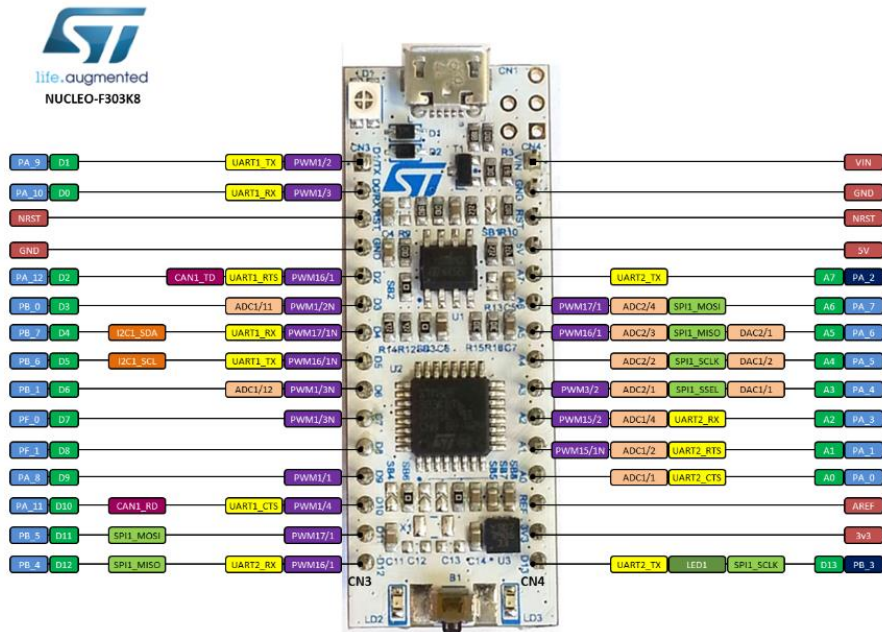


Figure 9 Nucleo F303K8

Pour résoudre ce problème, nous avons décidé d'utiliser deux Nucleo, une Nucleo (Slave) qui recevra les données de nos capteurs et une deuxième Nucleo (Master) qui recevra les données de la première Nucleo. Puis cette deuxième transmettra les données à notre application Android via le module Bluetooth. Ces deux Nucleo communiquent via le protocole I2C.

III.2- Schéma électrique

Ci-dessous nous pouvons retrouver le schéma de câblage de l'objectif 1.

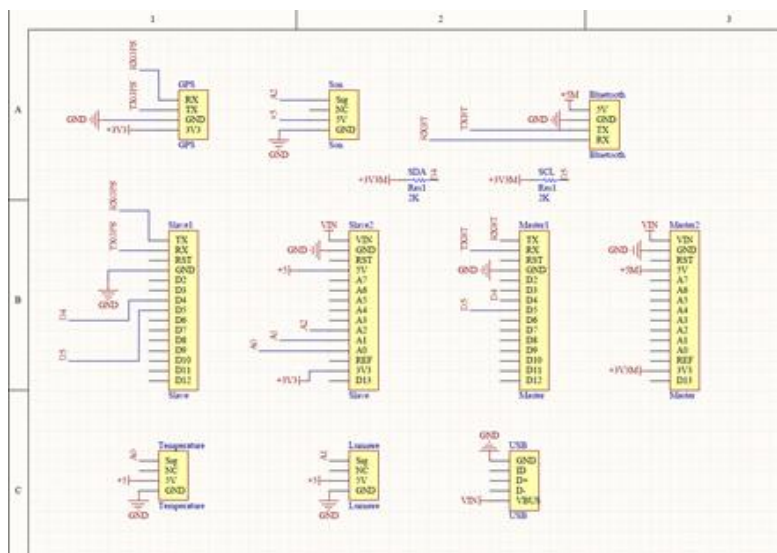


Figure 10 Schéma de câblage

Nous pouvons voir sur le schéma un composant appelé USB. Ce composant permet d'alimenter les deux cartes ainsi que les capteurs avec une batterie externe.

IV- Communication Bluetooth

Le Bluetooth nous permet de transmettre nos données à notre application Android. Pour ce faire nous avons utilisé le module Bluetooth HC05.

IV.1- Module BT HC05



Figure 11 HC05

Ce module Bluetooth communique par liaison série (UART). Cette liaison s'établit sur deux broches RX et TX. Il s'alimente sous 5V. Pour nos tests, nous avons utilisé l'application **Bluetooth Terminal HC-05**, cette application nous permet de vérifier la transmission Bluetooth.

V- Android

V.1- Présentation de l'IHM

Dans cette partie nous allons vous présenter l'IHM Android de notre application. Pour commencer vous pouvez trouver ci-dessous le menu qui permet de naviguer entre tous les onglets.

Nous pouvons trouver 3 onglets :

- Un onglet « GPS » qui permet d'afficher la position du drone ainsi que celle du téléphone.
- Un onglet « Autres capteurs » qui permet d'afficher les valeurs de nos capteurs.
- Un onglet « Bluetooth » qui permet la gestion du Bluetooth.

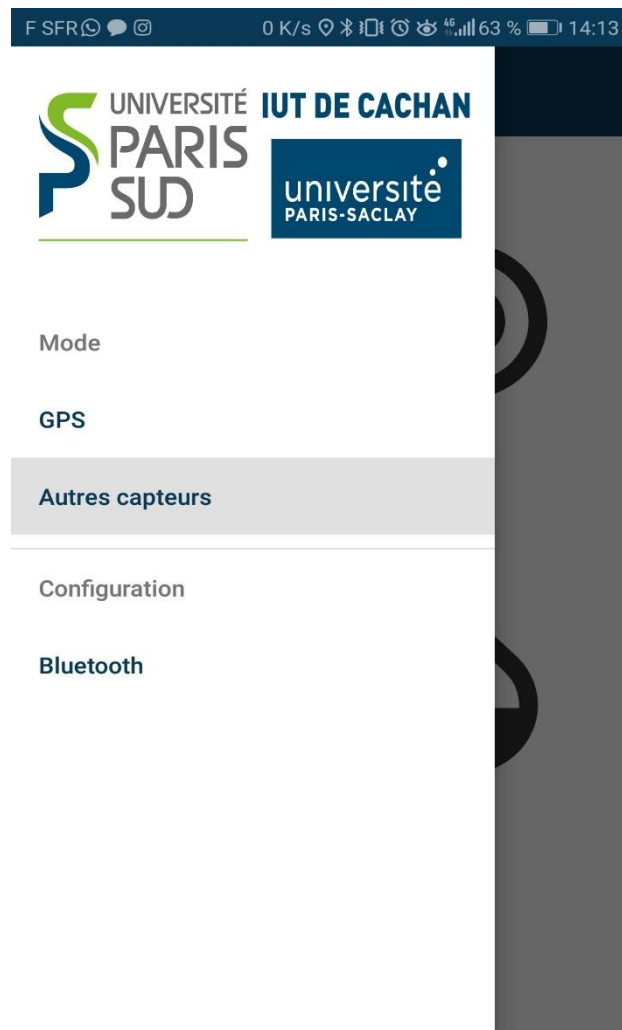


Figure 12 Menu de navigation

Ci-dessous vous pouvez trouver l'onglet « Bluetooth » qui permet la gestion de la partie Bluetooth de notre application.



Figure 13 Onglet Bluetooth

La photo ci-dessous explique l'affichage de l'onglet « GPS ». Nous pouvons retrouver deux informations la localisation du téléphone ainsi que celle du drone (seulement si le drone est connecté en Bluetooth).

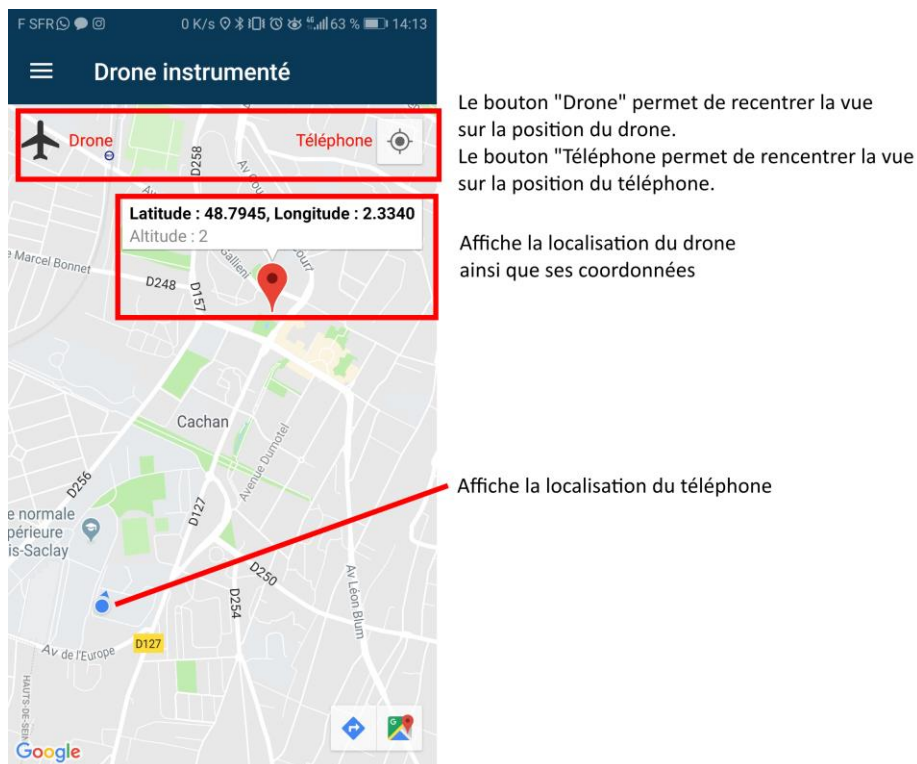


Figure 14 Onglet GPS

Pour finir la photo ci-dessous explique le fonctionnement de l'onglet « Autres capteurs ».

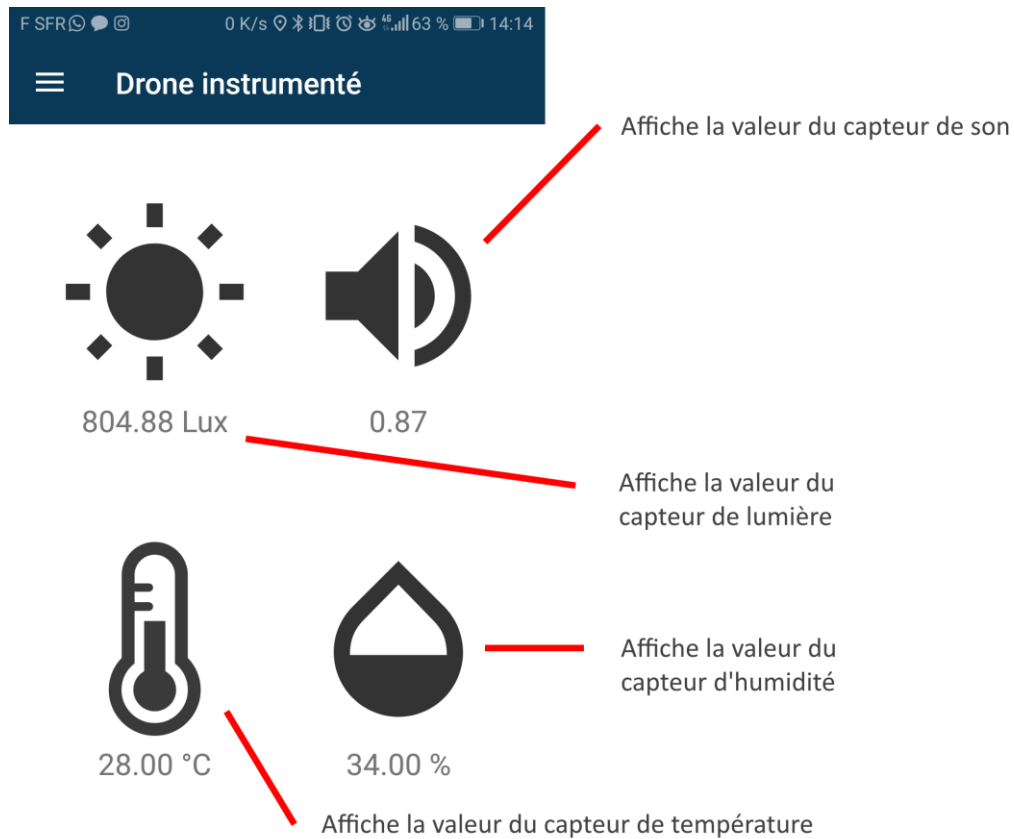


Figure 15 Onglet autres capteurs

V.2- Les classes utilisées

Le programme pour notre application Android est composé de 6 classes :

- « BluetoothSingleton » cette permet la gestion du Bluetooth.
- « FragmentAutre » cette classe est liée à l'affichage (layout « fragment_capteurs ») des capteurs.
- « FragmentBluetooth » cette classe est liée à l'affichage (layout « fragment_bluetooth ») du Bluetooth.
- « FragmentGPS » cette classe est liée à l'affichage du GPS.
- « GestionAffichageBT » est la classe qui stocke toutes les valeurs de réception du Bluetooth.
- « MainActivity » est la classe principale de notre application.

VI- Commande par gestes

Nous allons présenter dans cette partie l'objectif 2 de notre projet consistant à piloter le drone via des gestes.

VI.1- Capteur de gestes

La deuxième partie consiste à faire mouvoir le Parrot grâce au geste de l'utilisateur. Pour cela nous avons utilisé un capteur de mouvement.

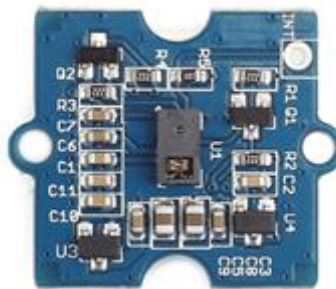


Figure 16 Capteur de gestes

Caractéristiques :

- 4 broches (GND, VCC, SDA, SCL)
- Protocole I2C
- Tension d'alimentation : 3V3

Grace au protocole I2C nous pouvons récupérer sur le registre 0X48 la nature du mouvement.

VI.2- SDK Parrot

Pour la partie Android nous avons utilisé l'application open-source fournit par Parrot. Nous pouvons retrouver cette application grâce au lien ci-dessous.

<https://github.com/Parrot-Developers/Samples>

Cette application nous permet de piloter un drone Parrot (**Attention ce lien ne comprend pas les modifications effectuées pour le pilotage par gestes**). Pour se connecter au drone suivre les étapes suivantes :

- 1- Activez le wifi de votre smartphone.
- 2- Allez dans vos paramètres wifi et sélectionnez votre drone.
- 3- Puis lancez l'application. Une liste devrait s'afficher avec votre drone sélectionnez-le.

VI.3- l'IHM de l'SDK Parrot

Dans cette partie nous allons vous présenter l'IHM Android de l'application permettant de piloter le drone

Ci-dessous nous pouvons trouver l'affichage pour le pilotage du drone.

Nous avons rajouté un bouton « Bluetooth » qui nous permet de nous connecter à notre microcontrôleur pour le pilotage par geste.

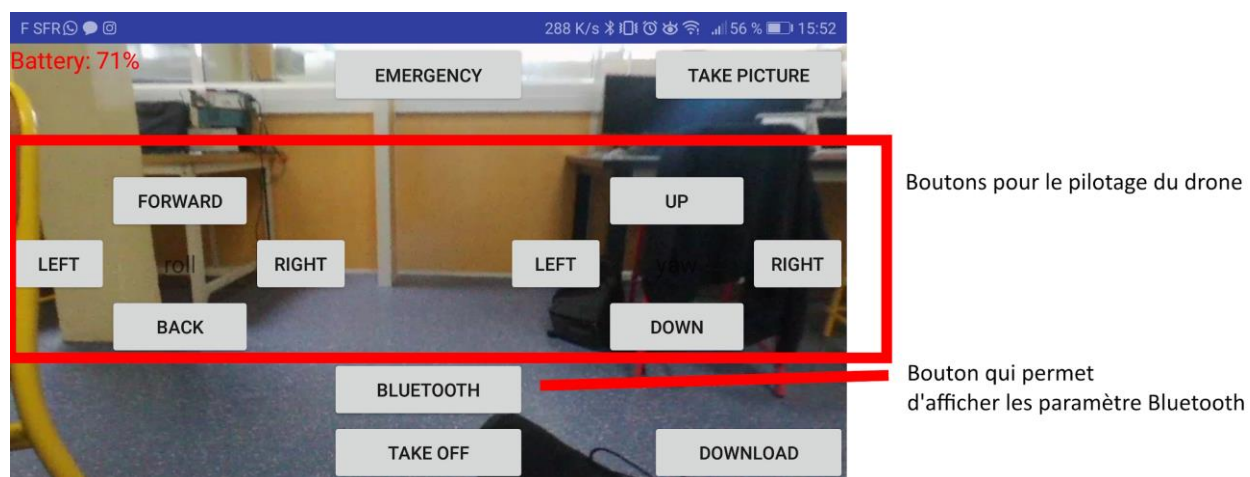


Figure 17 IHM application Parrot

Une fois appuyé sur le bouton « Bluetooth », un popup s'affiche (voir ci-dessous) qui permet de gérer le Bluetooth.

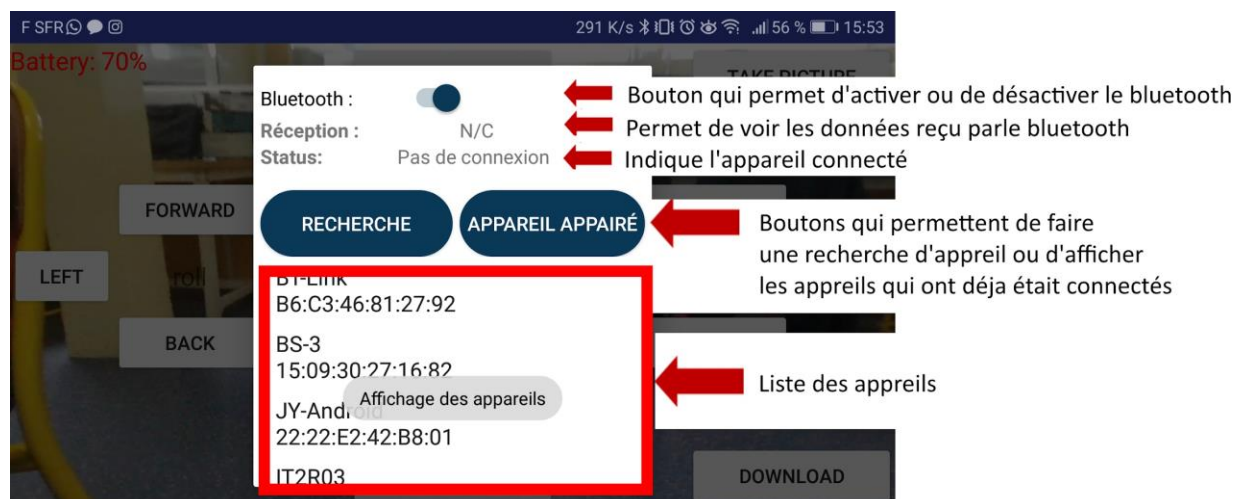


Figure 18 Onglet Bluetooth Parrot

VI.4- Les classes utilisées

Les classes qui nous intéressent dans le SDK sont les suivantes :

- « BebopActivity » qui permet de gérer l’affichage du layout « activity_bebop », c’est aussi dans cette classe que nous gérons le pilotage par gestes ainsi que par boutons.
- « FragmentGPS » cette classe est liée à l’affichage du GPS.
- « GestionAffichageBT » est la classe qui stocke toutes les valeurs de réception du Bluetooth.
- « BluetoothSingleton » cette permet la gestion du Bluetooth.

Les axes d'amélioration

Cette partie consiste à présenter les différentes améliorations pouvant être fait pour les futurs années.

- Amélioration du capteur de son. Le capteur sature rapidement il est donc impossible d'effectuer une mesure fiable.
- Amélioration du capteur de lumière. Le capteur sature rapidement il est donc impossible d'effectuer une mesure fiable.
- Communication I2C entre les deux Nucleo plus rapide.
- Amélioration de l'IHM de l'application Android recevant les valeurs de nos capteurs. (Avec des graphes).

Lien GitHub

Il est possible de récupérer tous les programmes ainsi que tous les documents de notre projet en allant sur le lien ci-dessous.

<https://github.com/drone-instrumente>

Tables des Figures

FIGURE 1 CAPTEUR SON	1
FIGURE 2 VALEUR CAPTEUR SON	2
FIGURE 3 CAPTEUR LUMINOSITE	2
FIGURE 4 VALEUR CAPTEUR LUMIERE	3
FIGURE 5 CAPTEUR DE TEMPERATURE ET D'HUMIDITE	3
FIGURE 6 VALEUR CAPTEUR DE TEMPERATURE ET D'HUMIDITE.....	4
FIGURE 7 GPS	4
FIGURE 8 GPGGA.....	5
FIGURE 9 NUCLEO F303K8	6
FIGURE 10 SCHEMA DE CABLAGE	6
FIGURE 11 HC05	7
FIGURE 13 MENU DE NAVIGATION	8
FIGURE 14 ONGLET BLUETOOTH.....	9
FIGURE 15 ONGLET GPS	9
FIGURE 16 ONGLET AUTRES CAPTEURS.....	10
FIGURE 17 CAPTEUR DE GESTES	11
FIGURE 18 IHM APPLICATION PARROT	12
FIGURE 19 ONGLET BLUETOOTH PARROT	12