

# Rapport de projet

Projet Déplacement de R2D2

**TOQUER FRANCOIS      PIERRE-YVES MINGAM**

08 janvier 2016

# Rapport de projet

---

Projet Déplacement de R2D2

## Sommaire

- I. Présentation du projet
- II. L'existant
- III. Préparation technique

# Rapport de projet

---

## Projet Déplacement de R2D2

### I) Présentation du projet

L'an dernier un projet M1 a consisté à développer un robot similaire à R2D2 de la saga Star Wars pour participer à la promotion de l'école (journées portes ouvertes, salons) en attirant l'attention du public. Le projet a permis de créer le robot et de développer une partie « sons et lumières ». Le but du projet de cette année est de s'occuper du déplacement du robot (de façon totalement séparée de l'existant).

Le développement s'effectuera sur une carte Beagle Bone.

La partie robotique sera en partie fournie.

Le travail demandé consiste à actualiser la partie robotique, faire la liaison entre la carte BeagleBone et la partie robotique et à faire se déplacer R2D2.

# Rapport de projet

---

## Projet Déplacement de R2D2

### II) L'existant

#### **Le fonctionnement existant actuellement :**

Un robot R2D2 a été créé l'année dernière afin de servir de promotion sur les forums post-bac. Il possède actuellement une partie Son & Lumière commandée par une carte Beaglebone.

Le projet consiste à ajouter une partie Déplacement totalement indépendante de la partie Son & Lumière visant à permettre de contrôler le déplacement du robot via ordinateur avec un câble ou via wifi/bluetooth avec un smartphone.

#### **Le périmètre du projet :**

Le robot R2D2 actuel est constitué d'une carcasse en cuivre qui contient la carte beaglebone ainsi que diverses LED et buzzers.

Le mode de déplacement retenu consiste en une plateforme de mobilité constituée d'un support physique et de 4 roues dont 2 seront propulsées par des moteurs, plateforme qui sera fixée sur le bas du robot.

La carte Beaglebone contient une distribution Debian spécialement adaptée. Les communications avec les moteurs se feront au travers d'un port série de la beaglebone qui sera relié au contrôleur des moteurs.

Les utilisateurs finaux du robot seront essentiellement les professeurs et ambassadeurs chargés d'aller aux forums de recrutement.

#### **Le contexte :**

Le robot actuel servant pour les portes ouvertes de l'établissement, la carte ainsi que les câbles servant à la transmission devront être remis avant chaque porte ouverte.

Des réunions d'avancement devront également avoir lieu quotidiennement, afin de tenir informés les clients de l'évolution du projet et des éventuels problèmes rencontrés.

# Rapport de projet

---

## Projet Déplacement de R2D2

### III) Préparation technique et gestion

QQOQCP :

**Quoi** : Gérer le déplacement d'un robot R2D2 via une plateforme de mobilité (roues + moteurs) ordonnée via ordinateur et/ou smartphone

**Qui** : Utilisateur lambda (professeurs + ambassadeur)

==> Nécessité d'un système d'authentification

**Où** : support physique plat (sol habituel : lino, parquet, carrelage) dans un environnement contrôlé (forums ou portes ouvertes)

**Quand** : Robot doit être accessible 24h/24h

==> nécessité d'un OS temps réel/longue durée de fonctionnement

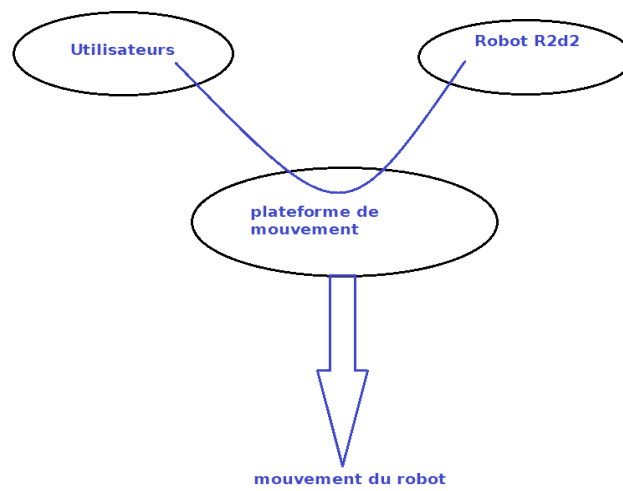
**Comment** : Application C/C++ pour programmation Beaglebone (filaire et Wifi/bluetooth)

Application Java pour Smartphone (communication Wifi/Bluetooth)

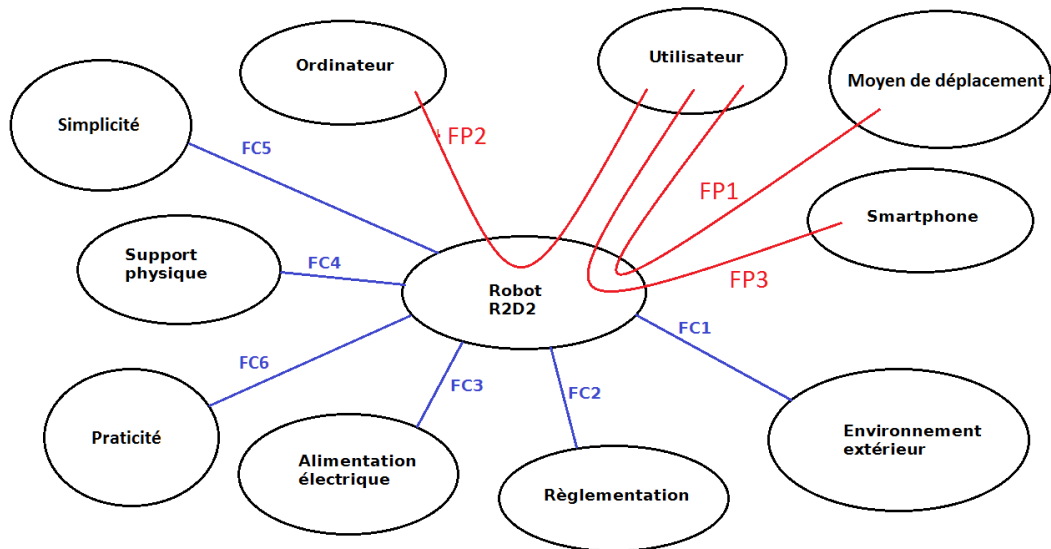
==> Voir connexion entre application Beaglebone et smartphone via ROS (Robot Operating System)

**Pourquoi** : Permettre une plus grande maniabilité et un plus grand contrôle sur le robot

Bête à corne :



## Diagramme de pieuvre global du robot :



FP1 : L'utilisateur peut déplacer le robot

FP2 : L'utilisateur peut contrôler le robot via ordinateur

FP3 : L'utilisateur peut contrôler le robot via smartphone

FC1 : Le robot (et son moyen de déplacement inclus) doit résister à l'environnement extérieur

FC2 : Le robot final doit suivre les réglementations et normes françaises

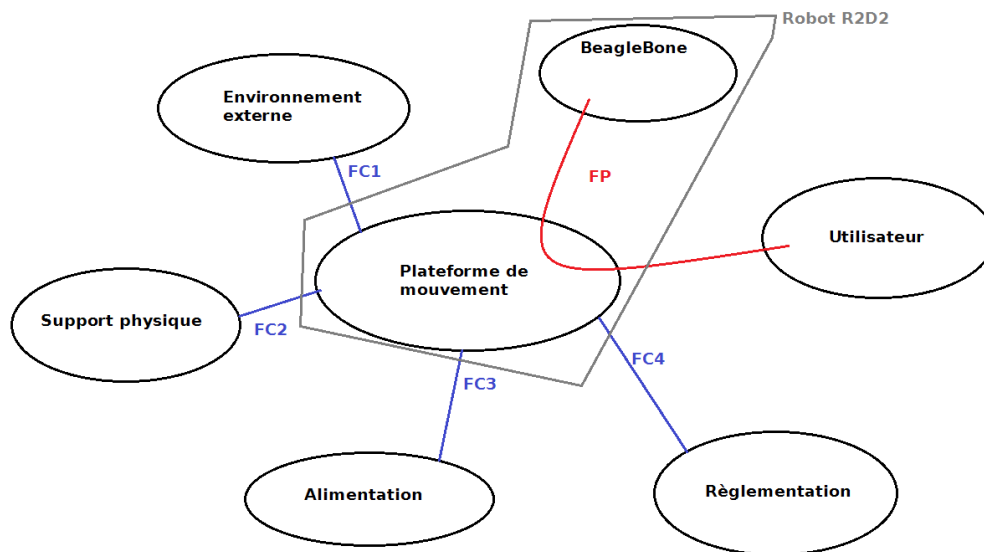
FC3 : Le robot (et son moyen de déplacement inclus) doit être autonome électriquement

FC4 : Le robot (à l'aide de son moyen de déplacement inclus) doit pouvoir gravir de très légères pentes

FC5 : L'assemblage final robot + moyen de déplacement doit être simple de conception

FC6 : L'utilisation et la manipulation du robot doivent être pratiques au quotidien

## Diagramme de pieuvre centré sur la plateforme de mobilité :



FP1 : La plateforme de mouvement doit pouvoir être commandé via beaglebone par l'utilisateur (communication via ordinateur et/ou smartphone)

FC1 : La plateforme de mobilité doit pouvoir résister à l'environnement extérieur

FC2 : La plateforme de mobilité doit pouvoir tenir sur un support physique plat ou légèrement incliné

FC3 : La plateforme de mobilité doit être autonome électriquement

FC4 : La plateforme de mobilité et le robot en général doivent suivre les lois et réglementations françaises.



## Étude des voies technologiques :

But à atteindre	Fonction technique	Solutions
FP 1 : Déplacement du robot	<ul style="list-style-type: none"> <li>Déplacement terrestre</li> <li>Déplacement aérien</li> <li>Déplacement maritime</li> </ul>	<ul style="list-style-type: none"> <li>Roues</li> <li>Chenilles</li> <li>Pattes</li> <li><u>Jetpack</u></li> <li>Fusées</li> <li>Flotteurs</li> </ul>
FP 2 : Déplacement via ordinateur	<ul style="list-style-type: none"> <li>Application et contrôle via ordinateur externe et transmission des commandes via moyen de transmission</li> <li>Robot directement pilotable via la carte</li> </ul>	<ul style="list-style-type: none"> <li>Filaire</li> <li>Transmission Wifi</li> <li>Transmission Bluetooth</li> <li>Clavier branché</li> <li>Joystick</li> </ul>
FP 3 : Déplacement via smartphone	<ul style="list-style-type: none"> <li>Application sur smartphone + moyen de transmission</li> </ul>	<ul style="list-style-type: none"> <li>Filaire</li> <li>Transmission Wifi</li> <li>Transmission Bluetooth</li> </ul>

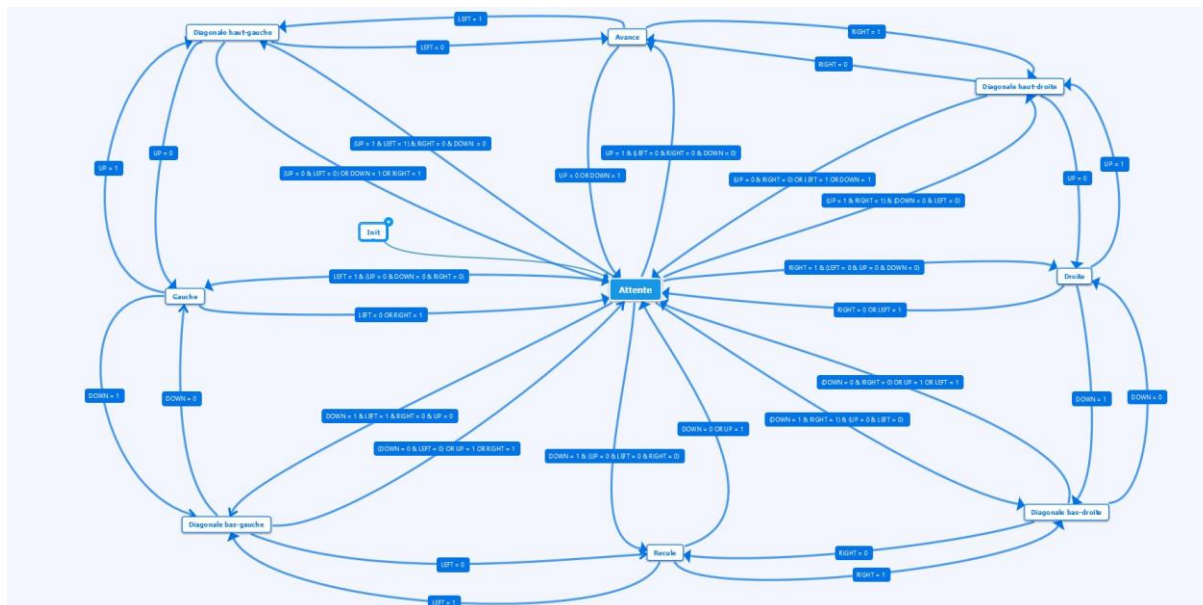
	FP1	FP2	FP3	FC1	FC2	FC3	FC4	FC5	FC6
<b>Déplacement du robot</b>									
Roues									
Chenilles									
Pattes									
Jetpack									
Fusées									
Flotteurs									
<b>Contrôle via ordinateur</b>									
Filaire									
Wifi									
Bluetooth									
Clavier branché									
Joystick									
<b>Contrôle via smartphone</b>									
Filaire									
Wifi									
Bluetooth									

Compte tenu de l'existant et des contraintes de simplicité et de praticité :

La solution retenue avec le client pour le déplacement du robot sera le déplacement terrestre via des roues. Les déplacements aériens et maritimes ne sont pas demandés et présentent des risques beaucoup plus importants.

Les solutions retenues pour le contrôle de ce déplacement sont la transmission via Wifi, simple à mettre en place, et la transmission filaire pour le déplacement via ordinateur, si la liaison wifi venait à échouer ou être indisponible pour certains ordinateurs.

## Diagramme de changements d'états simpliste :



## Analyse fonctionnelle du besoin :

Fonction	Critère d'appréciation	Niveau	Flexibilité
Déclencher les moteurs du robot suivant les commandes reçues	- Vitesse de déclenchement - Accélération	- Instantanée - 6 secondes d'accélération maximal	- +/- 1 seconde - +/- 2 secondes
Afficher le niveau de batterie du robot	- Vitesse de remontée des informations - Véracité des informations	- Instantanée - Informations vraies	- +/- 1 seconde - Erreurs mineures
Recevoir des commandes via ordinateur	- Vitesse de transmission	- Moins d'une seconde entre l'envoi et la réception de la commande	- Aucune
Recevoir des commandes via smartphone	- Vitesse de transmission	- Moins d'une seconde entre l'envoi et la réception de la commande	- Aucune
S'adapter au support physique	- Angle de la pente du support physique - Arrêt si détection d'obstacles	- 5 degré de pente maximal - 1 seconde maximale de temps de réaction	- +/- 5 degrés de pente - aucune
Arrêt complet en cas de perte du signal ou des commandes ou via bouton	- Temps d'arrêt du robot	- instantané	- +/- 1 seconde
Veille automatique des moteurs si niveau de batterie de la carte faible	- Temps de mise en veille - Message d'avertissement	- Moins d'une minute pour la mise en veille après sauvegarde des programmes - Affichage automatique et durée de 15 secondes pour le message d'avertissement	- 1 minute maxi - Durée de +/- 10 secondes

### Cycle de marche du robot :

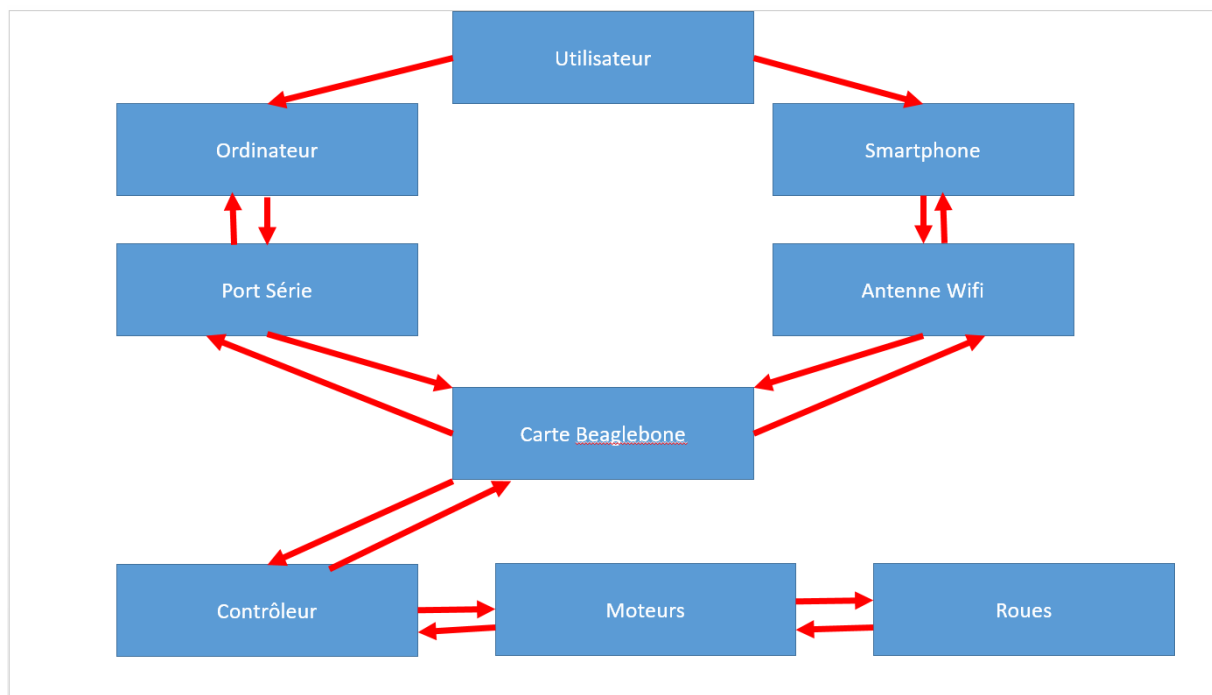
## Cycle de marche du robot

- Initialisation du robot
- Mise en place des liaisons (démarrage carte wifi)
- Lancement programme Son/Lumières/Écran à côté
- Attente des instructions (écoute du port série du contrôleur)
- Si commande reçue, démarrage des moteurs et exécution de la commande
- Si inactivité détectée, mise en veille des moteurs

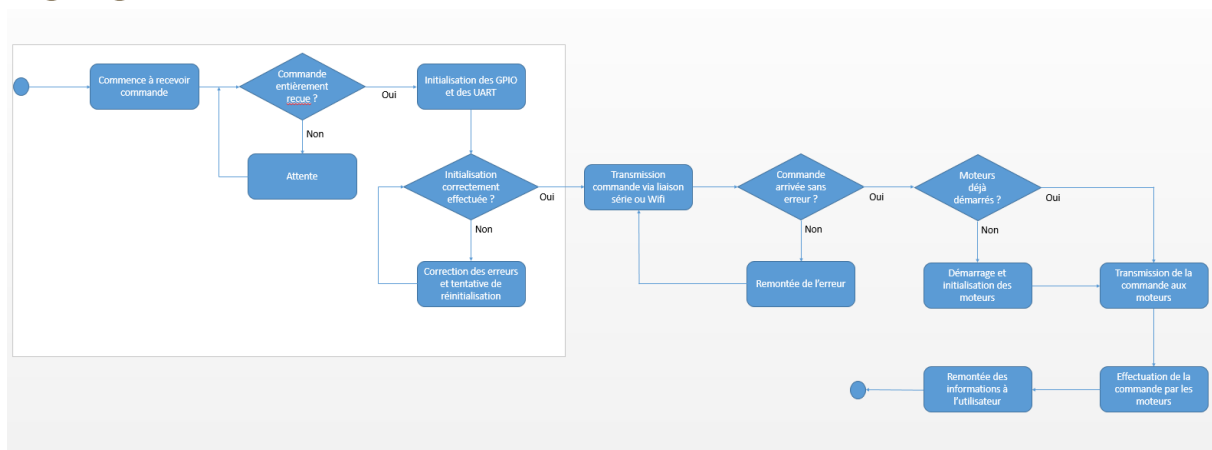
### Tableau des exigences :

[illegible]

## Synoptique fonctionnel :



## Organigramme de fonctionnement :



## Dimensionnement des batteries :

### Dimensionnement de la batterie Robot r2d2

Nom composants	Consommation:
Carte BeagleBone	210mA/h pour la green - 450mA/h pour la black
moteur (X2 ?)	2,5 A/h en bloqué / 500 mA/h en charge
antenne wifi	250 mA/h
ecran lcd	240 mA + carte : 3 mA
led externe	20-25 mA
buzzer	+/- 25mA/h
module de camera	100 mA
capteur de distance ultrason	15mA
<b>Total:</b>	<b>+/- 2150mA/h max</b>

=> fonctionnement sécurité 4000mA/h ?

## Cahier des tests :

Titre		Test du déclenchement du moteur		
Objectif		Vérifier que les cas d'erreurs possibles ont bien été prévus et que le moteur est bien déclenché à la bonne intensité à la réception de la commande		
Exigence du test		Tous les tests doivent passer pour certifier que ce module fonctionne correctement		
Exécutable		<u>Commande.c</u>		
Environnement		<u>Debian.gcc</u>		
ID	Démarche	Données entrées	Comportement attendu	Validation
1	Lancement de l'exécutable	/	Les moteurs s'allument et el robot ne bouge pas	
2	Passer en paramètre à la fonction deux nombres entiers positifs supérieurs à 128 égaux	255,255	Déclenchement des deux moteurs vers l'avant	
3	Passer en paramètre à la fonction deux nombres entiers positifs différents supérieurs à 128	150,180	Déclenchement des deux moteurs vers l'avant à intensité différentes (le robot avance en arc de cercle)	
4	Passer en paramètre à la fonction deux nombres entiers positifs différents inférieurs à 128	40,80	Déclenchement des deux moteurs vers l'arrière à intensité différentes (le robot recule en arc de cercle)	
5	Passer en paramètre à la fonction deux fois le nombre 128	128,128	Le robot s'arrête (ou ne bouge pas s'il était déjà à l'arrêt)	
6	Passer en paramètre à la fonction deux nombres entiers positifs différents, l'un supérieur à 128, l'autre inférieur à 128	60,180	Le robot tourne sur lui-même	
7	Passer en paramètre à la fonction deux caractères quelconques	A, b	Aucune exécution et redemande des entrées valides	

Titre		Test de réception des commandes		
Objectif		Vérifier que les commandes reçues correspondent bien à des commandes connues		
Exigence du test		Tous les tests doivent passer pour certifier que ce module fonctionne correctement		
Exécutable		<u>reception.c</u>		
Environnement		<u>Debian,gcc</u>		
ID	Démarche	Données entrées	Comportement attendu	Validation
1	Envoyer la commande pour récupérer le mode de fonctionnement	0x2B en liaison série	Affichage sur la console du mode de fonctionnement du <u>contrôleur</u>	
2	Envoyer la commande pour récupérer la vitesse du moteur 1	0x21 en liaison série	Affichage sur la console de la vitesse du moteur 1	
3	Envoyer la commande pour forcer la vitesse du moteur 1	0x31 en liaison série, suivi de 0xFF	Démarrage du moteur 1 à pleine puissance	
4	Envoyer la commande pour récupérer la vitesse du moteur 1	0x22 en liaison série	Affichage sur la console de la vitesse du moteur 2	
5	Envoyer la commande pour forcer la vitesse du moteur 1	0x312en liaison série, suivi de 0xFF	Démarrage du moteur 2 à pleine puissance	

Titre		Test d'envoi des commandes		
Objectif		Vérifier que les commandes prévues sont bien transmises		
Exigence du test		Tous les tests doivent passer pour certifier que ce module fonctionne correctement		
Exécutable		<u>transmission.c</u>		
Environnement		<u>Debian,gcc</u>		
ID	Démarche	Données entrées	Comportement attendu	Validation
1	Envoyer la commande pour récupérer le mode de fonctionnement	0x2B en liaison série	Récupération de 0x2b sur le <u>scanalogic</u>	
2	Envoyer la commande pour récupérer la vitesse du moteur 1	0x21 en liaison série	Récupération de 0x21 sur le <u>scanalogic</u>	
3	Envoyer la commande pour forcer la vitesse du moteur 1	0x31 en liaison série, suivi de 0xFF	Récupération de 0x31 et de 0xFF sur le <u>scanalogic</u>	
4	Envoyer la commande pour récupérer la vitesse du moteur 1	0x22 en liaison série	Récupération de 0x22 sur le <u>scanalogic</u>	
5	Envoyer la commande pour forcer la vitesse du moteur 1	0x32 en liaison série, suivi de 0xFF	Récupération de 0x32 et de 0xFF sur le <u>scanalogic</u>	

## Modélisation UML :

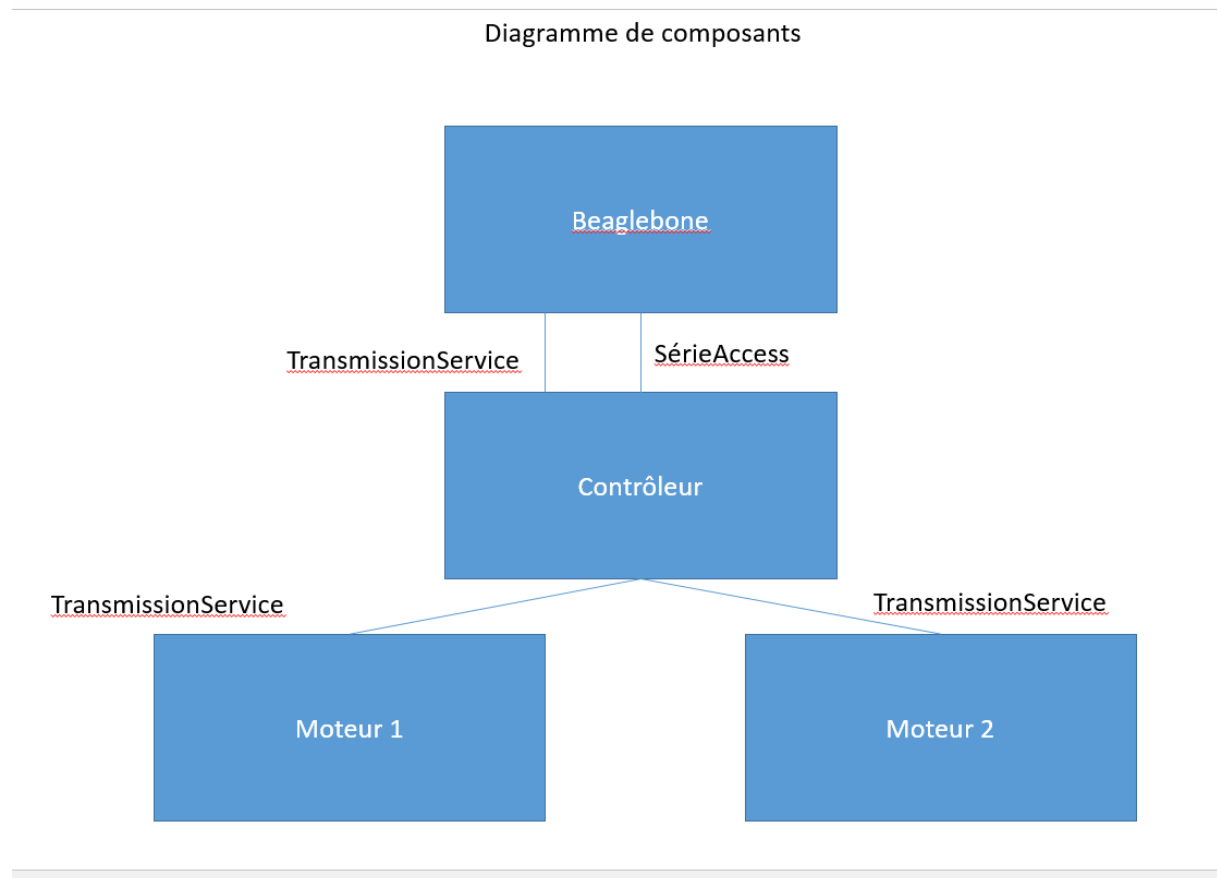
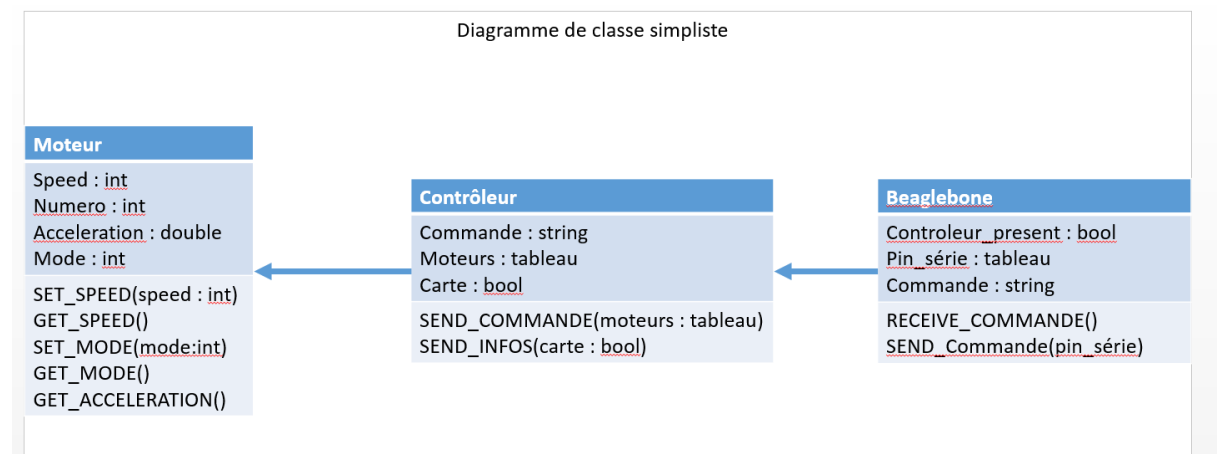




Diagramme de déploiement

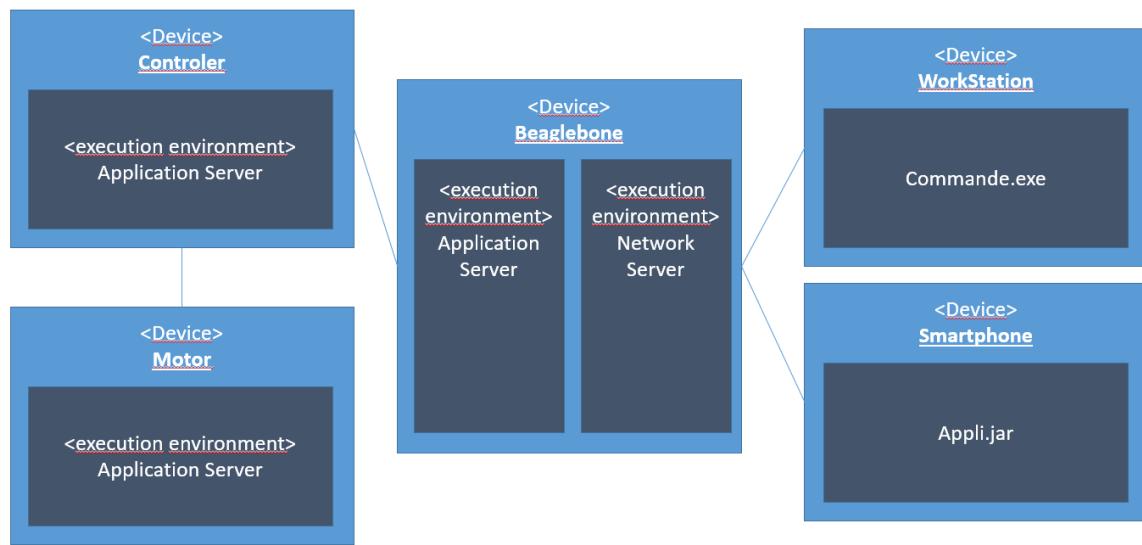


Diagramme des cas d'utilisation

