**METAL GEAR**

**Cahier des charges fonctionnel**

# 1 – Description générale du projet

Le but du projet est de contrôler le Freenove© hexapod par la voix en utilisant le ReSpeaker monté sur un raspberry pi 3 via WiFi. Une application web et mobile sera également mise en place pour pouvoir contrôler le Freenove Hexapod et visualiser le journal d’événements.

# 2 – Décomposition des Fonctionnalités

* **F1 Pilotage de Freenove© Hexapod :**

**F1.1** Pilotage du Freenove Hexapod par WiFi

**F1.2** Retranscrire les instructions vocale vers le micro service web (API) de coordination

|  |  |  |  |
| --- | --- | --- | --- |
| Code Hexapod | Variable | Commande Vocale | API Endpoint |
| 0 | requestEcho |  | /ping/hexapod |
| 1 | echo |  |  |
| 10 | requestSupplyVoltage |  | /ping/voltage |
| 11 | supplyVoltage |  |  |
| 20 | requestChangeIO |  |  |
| 30 | requestMoveLeg |  |  |
| 32 | requestCalibrate | calibre-toi | /action/calibrate |
| 64 | requestInstallState |  |  |
| 66 | requestCalibrateState |  |  |
| 68 | requestBootState |  |  |
| 70 | requestCalibrateVerify |  |  |
| 80 | requestCrawlForward | avance | /movement/forward/:qty |
| 82 | requestCrawlBackward | recule | /movement/backward/:qty |
| 84 | requestCrawlLeft | rampe à gauche | /movement/left/:qty |
| 86 | requestCrawlRight | rampe à droite | /movement/right/:qty |
| 88 | requestTurnLeft | tourne à gauche | /turn/left/:qty |
| 90 | requestTurnRight | tourne à droite | /turn/right/:qty |
| 92 | requestActiveMode | allume-toi | /action/wakeup |
| 94 | requestSleepMode | éteins-toi | /action/standby |
| 96 | requestSwitchMode |  |  |
| 110 | requestCrawl | rampe | /movement/:qty |
| 112 | requestChangeBodyHeight | change ta hauteur | /height/:height |
| 114 | requestMoveBody | bouge ton corps | /move/:qty |
| 116 | requestRotateBody | tourne-toi | /rotate/:qty |
| 118 | requestTwistBody | dance | /twist/:qty |
| 21 | orderStart |  |  |
| 23 | orderDone |  |  |
| 128 | transStart |  |  |
| 129 | transEnd |  |  |
|  |  |  |  |

**F1.3** Retranscrire les instructions http vers le Broker (Un micro service web (API) de coordination)

* **F2 Log des déplacements de Freenove© Hexapode**

**F2.1** logs des informations (type de contrôle, origine, date, quantité de mouvement, exécution de l’ordre, ordre effectué, erreur reçue (optionnel), description de l’erreur (optionnel))

**F2.2** Création de points d’accès de l’historique des instructions de déplacement du Freenove Hexapod

* **F3 Une application web et Mobile pour contrôler la Freenove© Hexapod à distance**

**F3.1** Pilotage à distance via des commandes envoyées en http par une application web / mobile

**F3.2** Envoyer des instructions de déplacement

**F3.3** Recevoir et afficher le journal d’événement

* **F4 Contrôle sécurisé via https**

**F4.1** Communication https via internet

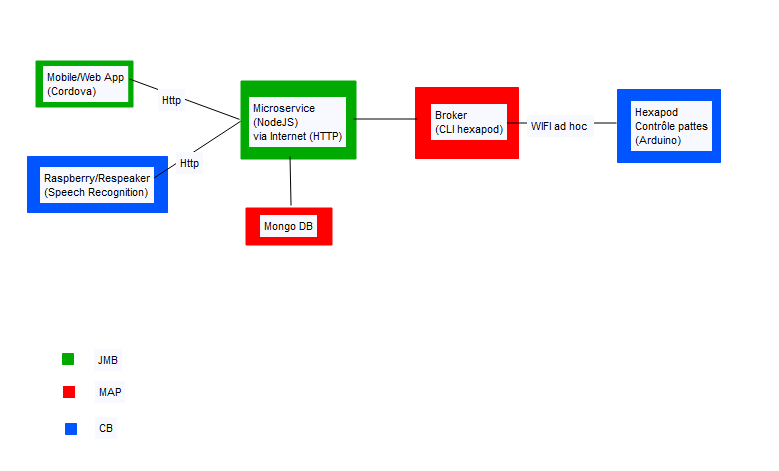
**F4.2** Générer et distribuer des certificats

**F4.3** Sécuriser le réseau WiFi du Freenove Hexapod

**– Schémas d’architecture**

Schéma matériel :

## 

Schéma logiciel :

## **4 - Scénarios / Acteurs / Rôles**

**Scénarios :**

**S1 :** Contrôle par la voix du Freenove© Hexapod

* **ReSpeaker :** Les commandes vocales sont traitées sur la raspberry. Le CLI python convertit les commandes vocales en instructions à envoyer au Freenove© hexapod : il génère une requête http pour le micro service de coordination qui va ensuite communiquer avec la CLI qui va via WiFi envoyer l’instruction traitée (en fonction du besoin) au module wifi esp8266 du Freenove hexapod.

**S2 :** Pilotage à distance

* **End devices :** L’application web/mobile va envoyer des instructions pour la Freenove© Hexapod en passant par l’API du micro service de coordination..Ensuite, la CLI enverra les instructions traitées au Freenove© Hexapod.

**S3 :** Journal d’événement

* **MongoDB:** L’api est montée avec une base de donnée mongodb qui se chargera de conserver toutes les instructions reçues/transmises entre le l’api et la CLI sous forme de journal d’événements

**S4 :** Accès á l’historique

* **End Devices :** Grâce au journal d’événements ainsi qu’au serveur de coordination, depuis les applications web/mobile, nous pouvons visualiser les instructions reçues par le Freenove© Hexapod en temps réel ou obtenir un historique.

**Acteurs :**

(Les acteurs sont les personnes amenées à piloter le Freenove© Hexapod.)

Toute personne ayant en sa possession l’application web/mobile ou qui sont proche du module respeaker pourront télécommander le Freenove© Hexapod

**Rôles :**

Les acteurs seront en mesure de contrôler le Freenove© Hexapod soit par l'intermédiaire d’une application web / mobile, soit par la voix, leur rôle sera donc celui d’un pilote de drone

## **5 – Répartitions du projet**

* **Louis Hervé :** F1.1, F1.2, F2.1
* **Flavien Jalabert :** Montage du Freenove© Hexapod, F1.3, F2.2, F3
* **Florent Cordier :** Montage du Freenove© Hexapod, F3, F4.3
* **Stephane Azoulay :** F1.2, F2.1, F3.1, F4

## **6 - Délais de réalisations**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fonctions | Nom | Date de début | Date de rendu | Suivi 3/4/2020 | Suivi 9/4/2020 | Suivi 10/4/2020 |
| F1.1 | HERVE | 03/04/2020 | 08/05/2020 | Script python sur le RPI – api Bing qui permet d’identifier des mots en texte ReSpeaker n’étant pas suffisamment évolué.  En l’état le vocabulaire est très restreint. | Recherche du dossier de la librairie BingSpeechAPI. Tentative de modification de l’url de l’API dans le code de la librairie car elle n’a pas été mise à jour depuis 2016. Il est impossible de contacter le support je dois me pencher sur le forum communautaire | Compréhension de la documentation pour l’API avec les headers et les paramètres à envoyer sur les différents URL (qui ont changés depuis). Implémentation du nouveau code dans la librairie BingSpeechApi et je reçois enfin des données vocales sous forme de text en json qu’il faut que je parse maintenant |
| F1.2 | HERVÉ, AZOULAY | 12/03/2020 | 03/04/2020 | Définir une table dans laquelle vous avez les mots clefs (txt) qui vont correspondre au vocabulaire de votre hexapod.  Stéphane et Florent doivent décider du vocabulaire minimal en fonction des capacités du POD  recule twist tourne etc  Avance de 30 mètres avance 10 s ?  Stephane va collecter dans un code python l’action à réaliser qui va envoyer au service web une action à réaliser qui contiendra le mouvement avec une valeur qui correspond aux mesures de distance du mouvement | travail sur le script python afin de lier les commandes vocales à l’api avec définition des mots clés importants utilisés dans le code |  |
| F1.3 | JALABERT | 27/03/2020 | 17/04/2020 | Le service web sera déployé sur le RPI  Adresse 127.0.0.1  Adresse privée dans le même réseau que le tel  Adresse publique derrière votre box  Port 4126  URI  Il attend des commandes de ce type  Format : /movement/forward/meter/30  /movement/forward/second/30  80 est la quantité de l’unité considérée  /twist/  Il log les commandes et on il envoi au broker pour redistribuer la commande reçue à l’hexapode (en bytes) | setup du MVC, le faire fonctionner, attente du client REST |  |
| F2.1 | HERVÉ, AZOULAY | 03/04/2020 | 10/04/2020 |  |  |  |
| F2.2 | JALABERT | 10/04/2020 | 17/04/2020 | On ajoute un endpoint api qui accède aux logs | attente du client REST |  |
| F3.1 | JALABERT, CORDIER, AZOULAY | 10/04/2020 | 24/04/2020 | Pilotage à distance :  Mise en place d’une appli développée en cordoba  Panneau connexion au micro service  Panneau de contrôle (correspondance entre ordre  Panneau pour lire logs  Node js | coquille Cordova en attente du MVC |  |
| F3.2 | JALABERT, CORDIER | 17/04/2020 | 24/04/2020 | A terme les data seront envoyées en NODEJS á l’Hexapod mais pour le moment on fait des test avec un script python qui prend des data envoyées par le broker et seront dirigées par en wifi a l’hexapode | correction d erreur sur le script python communiquant avec l’hexapod |  |
| F3.3 | JALABERT, CORDIER | 17/04/2020 | 24/04/2020 | Un micro-service web qui lorsqu’il est interrogé vous renvoi des logs de l’hexapode | attente du client REST |  |
| F4.1 | AZOULAY | 24/04/2020 | 08/05/2020 |  |  |  |
| F4.2 | AZOULAY | 24/04/2020 | 08/05/2020 | Accéder au micro service de log soit de commande ca veut que vous allez y accéder avec une adresse publique et certificat. Sur un port particulier  @publique https://200.0.0.1 :443  Flavien et Stéphane partage sur le déploiement de micro service avec ou sans certificat  Votre box va faire du nat va envoyer sur le RPI http ph récupère les commandes et qui va lancer les web services correspondants  Appel de micro services en https ? |  |  |
| F4.3 | AZOULAY, CORDIER | 24/04/2020 | 08/05/2020 |  |  |  |

## **7 - Définitions**

**Respeaker :** Est une carte d'extension à quadruple microphone pour Raspberry Pi conçue pour les applications IA et voix. Cela signifie que vous pouvez créer un produit vocal plus puissant et plus flexible qui intègre le service Amazon Alexa Voice, l'Assistant Google, etc.

**Freenove© Hexapod :** Hexapod est un robot développé par Freenove© compatible avec arduino, il est composé en plus d’un module WiFi esp8266 pour pouvoir le contrôler à distance.

**Module WiFi esp8266 :** L'ESP8266 est un circuit intégré avec un microcontrôleur permettant la connexion en WiFi. Les modules intégrant ce circuit sont très utilisés pour contrôler des périphériques par Internet.

## 

## **Informations hexapod**

2 action pour avancer est égal à un pas car il y a seulement côté qui bouge par action. Du Coup un pas est compris entre 4.5 et 5.5 cm.