METAL GEAR

Cahier des charges

## **1 – Description générale du projet**

*A rephraser pour que ca soit plus accessible*

Le but du projet est de contrôler le freenove hexapod par la voix en utilisant le respeaker monté sur un raspberry pi 4 et de NodeMCU communiquant avec le module WiFi du freenove hexapod (esp8266). Une application web et mobile sera également mise en place pour pouvoir contrôler l'araignée et visualiser le journal d’événements.

## **– Décomposition des Fonctionnalités**

* Pilotage de l’hexapode :

F1 Pilotage en local par la voix

F2 Pilotage à distance via des commandes envoyées par une application web

F3 Log des déplacements de l’hexapodeF4 Accès à l’historique des déplacements

* + Un micro service web (API) de coordination
* Une application web et Mobile pour contrôler l’araignée à distance
* Un serveur NodeMCU (qui joue le rôle du broker MQTT)
* Une application d’écoute de commandes vocales pour contrôler ensuite l’araignée à distance
* Configuration de l’araignée (Arduino)

## **– Schéma d’architecture**

## *A revoir en sáparant matériel et logcel*

## 

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

*Décrire les 4 fonctionnalités*

**Scénarios :**

**F1 : décrire comment ce se passe**

* **ReSpeaker : Les commandes vocales sont traitées sur la raspberry. Le deamon python convertit les commandes en ordres à envoyer à l’hexapod : il génére une requête http pour le NodeMCU qui va ensuite over WiFi envoyer l’ordre traité (en fonction du besoin) au module esp8266 de l’hexapod.**

F2 pilotage `s distance

On entre une commande sur panneau

* **End devices : L’application mobile va envoyer des commandes d’ordres pour l’araignée en passant par l’API du serveur web. Celui ci communiquera directement avec le NodeMCU en http.Ensuite, le NodeMCU enverra les ordres traités à l’hexapod.**

F3 log des déplacements

F4 accès á l’historique

**Acteurs :**

Les acteurs sont les personnes amenées à piloter l’hexapod...

**Rôles :**

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

## **5 – Répartitions du projet**

Répartition a préciser dans le temps et qui fait quoi

* **Louis Hervé :** Serveur NodeMCU, Application d’écoute de commandes vocales
* **Flavien Jalabert :** Serveur web avec API, Application web et Mobile
* **Florent Cordier :** Configuration de l’araignée (Arduino), Application web et Mobile
* **Stephane Azoulay :** Serveur NodeMCU et configuration réseau

## **6 - Délais de réalisations**

Planning à réaliser avec les différents livrables et leur date de livraisson

## **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.

NodeMCU : **NodeMCU** est une plate-forme open source [IoT](https://fr.wikipedia.org/wiki/Internet_des_objets), matérielle et logicielle, basée sur un [SoC](https://fr.wikipedia.org/wiki/Syst%C3%A8me_sur_une_puce) [Wifi](https://fr.wikipedia.org/wiki/Wi-Fi) [ESP8266](https://fr.wikipedia.org/wiki/ESP8266) ESP-12 fabriqué par [Espressif Systems](https://fr.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1) [**(ca)**](https://ca.wikipedia.org/wiki/Espressif_Systems). Le terme « NodeMCU » se réfère par défaut au firmware plutôt qu'aux kits de développement. Le firmware, permettant nativement l'exécution de scripts écrits en [Lua](https://fr.wikipedia.org/wiki/Lua), est basé sur le projet eLua[7](https://fr.wikipedia.org/wiki/NodeMCU#cite_note-7) et construit sur le SDK Espressif Non-OS pour ESP8266[8](https://fr.wikipedia.org/wiki/NodeMCU#cite_note-8). Il utilise de nombreux projets open source comme lua-cjson et spiffs.