|  |  |
| --- | --- |
| Clémence MAIGNAN  Timothée VIARD | E2a - ESEO  10/03/2023 |

PROJET MBDF

(Main Bionique Du Future)

- Rapport de DEEP –

*Livraison intermédiaire*

Sommaire

[1. Cahier des charges 3](#_Toc131061645)

[1.1. Définition du projet 3](#_Toc131061646)

[1.2. Description des exigences fonctionnelles 3](#_Toc131061647)

[1.2.1. Première étape 3](#_Toc131061648)

[1.2.2. Deuxième étape 3](#_Toc131061649)

[1.2.3. Etapes complémentaires 4](#_Toc131061650)

[2. Composants utilisés 4](#_Toc131061651)

[3. Cahier de suivi 5](#_Toc131061652)

[4. État d’avancement et analyse du projet réalisé 6](#_Toc131061653)

[5. Software 6](#_Toc131061654)

[6. Compléments envisagés 6](#_Toc131061655)

# Cahier des charges

## Définition du projet

Le projet MBDF consiste en la conception et la réalisation d'une main bionique contrôlée par un microcontrôleur STM32. Le mouvement sera produit par plusieurs servomoteurs (5 servomoteurs, un pour chaque doigt). L'objectif est de créer une main avancée qui puisse être contrôlée de différentes manières, par l’utilisateur. L’utilisateur pourra donc produire différents mouvements à retranscrire sur la main bionique. Le projet nécessitera des compétences en électronique, en informatique et en mécanique pour concevoir et assembler la main bionique fonctionnelle.

## Description des exigences fonctionnelles

Le projet nécessite un découpage afin de pouvoir quantifier le travail nécessaire. Il va notamment être intéressant afin de pouvoir diviser et cadrer notre travail.

### Première étape

La première étape consiste à faire bouger les doigts de la main grâce au STM32. Cette première étape va utiliser la méthode d’instruction par l’ordinateur, et nécessite donc d’être connecté à celui-ci l’interaction entre la main et l’utilisateur se fera donc grâce à l’ordinateur qui communiquera les informations au STM32. Les servomoteurs seront Il y aura par la suite l’utilisation de potentiomètres qui (de manière simplifiée) feront office de capteurs de fermeture de la main.

### Deuxième étape

Cette étape a pour objectif de faire bouger la main selon des capteurs de flexion et non plus par des potentiomètres. Ces capteurs seront placés sur un gant et pourrons détecter le mouvement de la main d’une personne. Ces mouvements seront retranscrits à la main bionique.

### Etapes complémentaires

* Contrôler la main à partir d’un téléphone, de boutons ou autres, l’utilisateur pourra contrôler la main par différentes méthodes qu’il pourra choisir au préalable.
* Etat de la main affiché grâce à des afficheurs lcd (ou addafruit), ou led, l’utilisateur pourra observer la position de chaque doigt ainsi que l’avancé de possible mouvements pré-enregistrés.
* Communication Bluetooth, radio, pour être télécommandé, l’utilisateur pourra communiquer à distance avec la main (ne nécessitera pas de liaison filaire)
* Capteur de pression pour détecter s’il y a un obstacle dans la main, la main bionique pourra détecter lorsqu’un objet sera placé dans sa paume, et pourra donc contrôler la puissance de fermeture.

# Composants utilisés

Utilisation d’un modèle 3D et de son impression, déjà créée l’année précédente

Utilisation d’une carte STM32

Utilisation de 5 Servomoteurs

Utilisation de 5 potentiomètres

Alimentation

|  |  |
| --- | --- |
| **Pin** | **role** |
| **PA0** | Capteur de flexion |
| **PA1** | Capteur de flexion |
| **PA2** | Capteur de Flexion |
| **PA3** | Capteur de Flexion |
| **PA4** | Ecran TFT |
| **PA5** | Ecran TFT |
| **PA6** | Ecran TFT |
| **PA7** | Ecran TFT |
| **PA8** | 1.CH1 – Servomoteur 1 (ID1 – CHAN1) |
| **PA9** | 1.CH2 – Servomoteur 2(ID1 – CHAN2) |
| **PA10** | 1.CH3 – Servomoteur 3 (ID1 – CHAN3) |
| **PA11** | 1.CH4 – Servomoteur 4 (ID1 – CHAN4) |
| **PA12** |  |
| **PA13** | (non dispo - SWDIO) |
| **PA14** | (non dispo - SWDCLK) |
| **PA15** | non dispo sur certaines bluepill |
| **PB0** | Capteur de Flexion |
| **PB1** | Ecran TFT |
| **PB2** | (non dispo) |
| **PB3** |  |
| **PB4** |  |
| **PB5** |  |
| **PB6** | 4.CH1 – Servomoteur 5 (ID4 – CHAN1) |
| **PB7** |  |
| **PB8** |  |
| **PB9** |  |
| **PB10** | Ecran TFT |
| **PB11** | Ecran TFT |
| **PB12** |  |
| **PB13** |  |
| **PB14** |  |
| **PB15** |  |
| **PC13** | (LED bluepill) |
| **PC14** | (Quartz 32kHz bluepill) |
| **PC15** | (Quartz 32kHz bluepill) |

# Cahier de suivi

|  |  |  |
| --- | --- | --- |
| Date | Tâches, réalisateurs, difficultés rencontrées. | A faire la prochaine fois |
| 10/03 | Clémence : Réflexion sur le projet.  Commencer à faire le software du projet et regarder un peu la partie hardware/altium.  Timothée : Réflexion sur le projet. Création de la trame du rapport. Création de la trame intermédiaire du rapport | Clémence : Tester les servomoteurs avec le software. Continuer à réfléchir sur l’avancement du cahier des charges.  Timothée : Continuer le cahier et commencer à voir pour les composants électroniques. |
| 30/03 | Clémence : Continuer à comprendre et programmer les servomoteurs selon un bouton poussoir.  Timothée : Continuer le cahier des charges puis fabriquer à l’aide d’un tuto un capteur de flexion « DIY ». Trouver les bons pins pour brancher les servomoteurs et les capteurs de flexion. | Clémence : Essayer avec un potentiomètre d’abord en déterminant un seuil, puis en essayant de suivre la valeur du potentiomètre.  Timothée : Perfectionner le capteur de flexion DIY. |
| 31/03 | Clemence : Echange avec Timothée, perfectionnement du capteur de flexion et étude de ce dernier. Continuer le schéma Altium.  Timothée : Echange avec Clémence. Se pencher sur la partie software et reprendre là ou en était Clémence : bouton poussoir et servomoteurs. |  |

# État d’avancement et analyse du projet réalisé

# Software

# Compléments envisagés

|  |  |
| --- | --- |
| - routage de PCB (avec bluepill = 3 pts || CMS microcontrôleur nu = 4pts) | x |
| - utilisation d'un analyseur logique pour déchiffrer des trames (2 pts) | o |
| - mesure de conso selon scénarios (2 pts) | x |
| - enregistrement de paramètres en flash (1pt) | o |
| - design CAO d'un boîtier (2 pts) | o |
| - documentation doxygen du code source (1pt) | o |
| - jeu de tests pour valider une fonctionnalité software ou hardware (1pt) | x |
| - gestion de version du code source (1pt) | x |