Création d’une chaise de bureau connectée à réglage électrique et mémoire de position



TO52

Rapport

Quentin Chambefort

Nahil Zamiati

Table des matières

[Introduction 3](#_Toc534476343)

[1. Présentation Du projet 3](#_Toc534476344)

[a. Expression du besoin / Marketing 3](#_Toc534476345)

[b. Réponse au besoin 3](#_Toc534476346)

[c. Ouvertures 3](#_Toc534476347)

[2. Cahier des charges 3](#_Toc534476348)

[a. Expression du besoin 3](#_Toc534476349)

[b. Difficultés prévues & contraintes du projet 4](#_Toc534476350)

[3. Réalisation du projet 5](#_Toc534476351)

[a. Descriptif de l’existant 5](#_Toc534476352)

[b. Travail à effectuer 6](#_Toc534476353)

[i. Partie mécanique 6](#_Toc534476354)

[ii. Partie électronique 7](#_Toc534476355)

[iii. Partie informatique 7](#_Toc534476356)

[c. Choix des technologies 7](#_Toc534476357)

[d. Conception électronique 8](#_Toc534476358)

[i. Contrôle des moteurs 8](#_Toc534476359)

[ii. Choix des composants 8](#_Toc534476360)

[iii. Câblage 10](#_Toc534476361)

[iv. Création d’un PCB 11](#_Toc534476362)

[e. Conception informatique 12](#_Toc534476363)

[f. Conception mécanique 13](#_Toc534476364)

[4. Bilan 16](#_Toc534476365)

[a. Résultats obtenus 16](#_Toc534476366)

[b. Travail restant 16](#_Toc534476367)

[c. Difficultés rencontrées 16](#_Toc534476368)

[d. Ouverture 16](#_Toc534476369)

[Sources 17](#_Toc534476370)

# Introduction

Parler de l’idée du projet

Bullshit

# Présentation Du projet

## Expression du besoin / Marketing

Expliquer l’idée qu’on passe quand même pas mal de temps assis sur une chaise en tant que dev

Donc autant qu’elle soit confortable

Expliquer que le manque de possibilité de réglage rend les chaises de bureau moins « polyvalentes » (matter un film dans une chaise de bureau, quelle plaie)

Faire des stats de position, connaître la position préférée des utilisateurs en fonction des applis qu’ils utilisent

## Réponse au besoin

Chaise de bureau ultra-confortable

Réglage électrique et automatique

Mémoires de position pour utilisation par différents utilisateurs

Intégrée dans un univers d’objets connectés, type « smart home »

Réagit selon l’utilisation

## Ouvertures

Utilisation comme fauteuil de salon, s’adapte à l’utilisation

Utilisation dans les voitures, selon le type de route (autoroute vs nationales)

# Cahier des charges

## Expression du besoin

* Le siège doit être confortable.
* Le siège doit pouvoir se régler de manière manuelle à l’aide de boutons physiques/interface à l’écran. (Application en réseau local)
* Le siège doit pourvoir être contrôlé de manière automatique selon des paramètres utilisateurs (mémoire de position)
* Le siège doit pourvoir être multi-utilisateur, et associer une série de paramètres suivant l’utilisateur l’utilisant.
* Le siège doit pouvoir s’intégrer dans un environnement d’objets connectés (assistant vocal, application bureau, etc…)
* Le siège doit pouvoir détecter si une personne est assise dessus.
* Le siège doit proposer des API de développement.
* Le siège doit fonctionner sans fil.
* Le siège doit pouvoir se connecter à un réseau wifi.

## Difficultés prévues & contraintes du projet

Les difficultés prévues ont dans un premier temps étés la prise en main du projet.

Choix du matériel : carte de programmation adéquate

Prise en main des composants électroniques

Prise en main des assistants vocaux

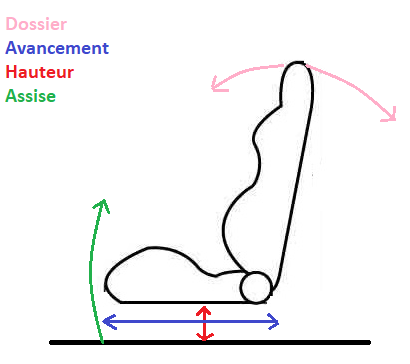
Partie conception mécanique et adaptation de pièces sur le châssis du siège.

Découverte des API pour les différents assistants vocaux sur le marché.

# Réalisation du projet

## Descriptif de l’existant

Un siège avec 4 moteurs pour faire bouger 4 mouvements différents :

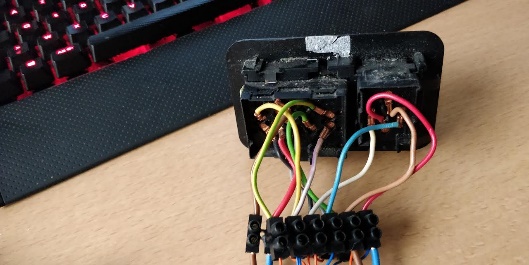


Le mouvement « avancement » consiste à faire avancer le siège sur un mouvement linéaire horizontal

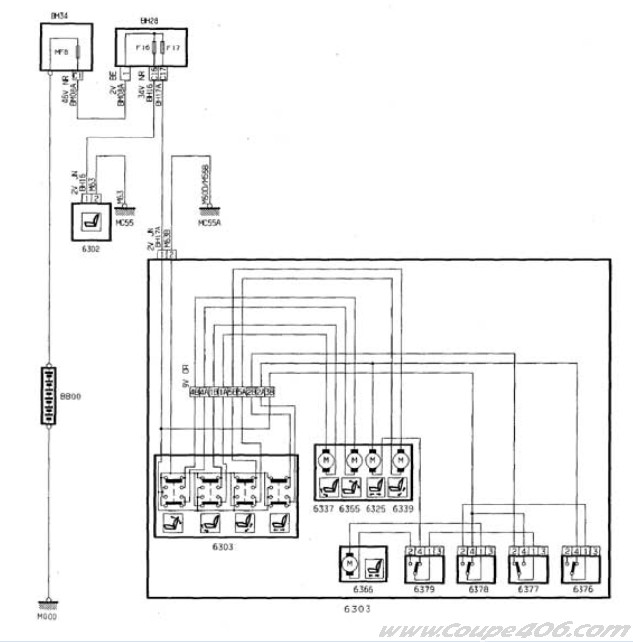
Le mouvement « hauteur » consiste à faire monter le siège selon un mouvement linéaire vertical

Le mouvement « assise » consiste à lever l’extrémité avant de l’assise (sous les genoux)

Le mouvement « dossier » consiste à incliner le dossier



Les 4 moteurs sont connectés directement à des boutons situés sur la tranche du siège pour les piloter directement

De plus, le siège possède un branchement pour alimenter des résistances sous l’assise, permettant ainsi de chauffer le siège.

## Travail à effectuer

Le travail s’articulera selon 3 axes :

Une partie mécanique, une partie électronique et une partie informatique.

### Partie mécanique

La partie mécanique du travail à réaliser consiste en plusieurs choses.

Tout d’abord, créer un support pour le siège afin de pouvoir l’utiliser de façon stable, et pouvoir le déplacer facilement. Ce travail a déjà été réalisé en amont : la structure réalisée est en bois, et est équipée de roulettes afin de pouvoir être déplacé facilement malgré son poids.

Ensuite, le siège n’étant pas équipé de capteurs d’origine, il faudra les installer. En effet, le siège sur lequel s’appuie le travail est le siège passager et non le siège conducteur de la 406, il n’est par conséquent pas équipé de capteur de position, car ne possède pas la fonctionnalité « mémoire de position ».

### Partie électronique

La partie électronique :

Monter une batterie et alimenter le siège

Contrôler les moteurs avec une carte programmable

Récupérer les informations des boutons de la tranche du siège

Récupérer des informations de position du siège

Contrôler une IHM

Optionnel : piloter le chauffage du siège

### Partie informatique

Implémenter les fonctions suivantes :

Piloter les moteurs en fonction des boutons actionnés

Piloter les moteurs en fonction d’une position souhaitée

Connecter la carte au wifi local

Récupérer des ordres de positionnement par wifi

Proposer un menu IHM

## Choix des technologies

D’abord Arduino MEGA pour nombre d’I/O disponibles, connaissance préalable de l’environnement ARDUINO

Puis ESP8266 E12 (wifi intégré de base sans rajout de modules)

Contraintes de l’ESP8266 : moins d’I/O, fonctionne en 3.3v

Avantages : meilleures performances, wifi intégré

Pour pallier le manque d’I/O :

Utilisation de registres à décalage en série (SN74HC595N 8bit) et de multiplexeurs (CD4051BE)

Avantages SN74HC595N : Mise en série des SR avec le même nombre d’entrées

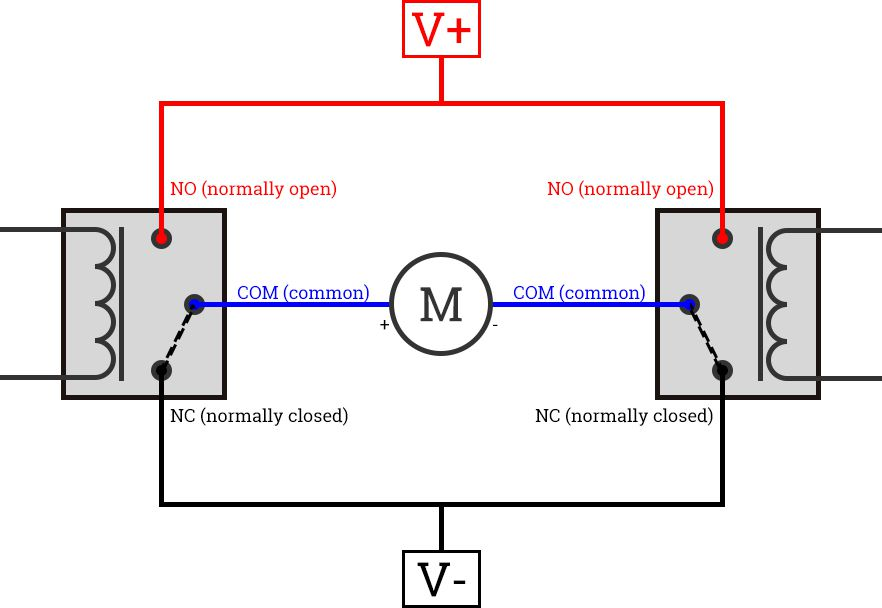
Avantages CD4051BE : utilisation de 2 CD4051BE avec 1 shiftreg, permet de brancher 16 entrées analogiques sur le seul ADC de l’ESP8266

Utilisation d’un écran LCD et d’un bouton type « rotary encoder » pour implémenter une IHM (facilité d’utilisation pour l’utilisateur et facilité d’implémentation de l’écran)

## Conception électronique

### Contrôle des moteurs

Afin de contrôler les moteurs, leur alimentation a été « hi-jacké » directement au niveau du bloc boutons, et relié à un bloc 8 relais, pouvant être piloté par une carte programmable pour la suite du projet.

Le câblage a été réalisé comme expliqué sur le schéma suivant :

Ce type de câblage permet donc de contrôler les moteurs dans un sens, comme dans l’autre, en ouvrant les relais un à un.

### Choix des composants

Premier test avec Arduino Mega



Suite avec ESP8266 E12



Utilisation de registres à décalage SN74HC595N

Explication du bidule

Explication de pourquoi qu’on l’a utilisé

[Documentation](http://www.ti.com/lit/ds/symlink/sn74hc595.pdf) ici

Coût en I/O : 3 (coût indépendant du nombre de registres)

Gain en O : 24

Utilisation de multiplexeur CD4051BE

Explication du bidule

Screen de la doc

On l’a utilisé car manque d’I/O

Branché sur un SR

Coût en I/O : 1 (intégralement câblé sur un registre à décalage)

Gain en I : 16

Potentiomètres analogiques

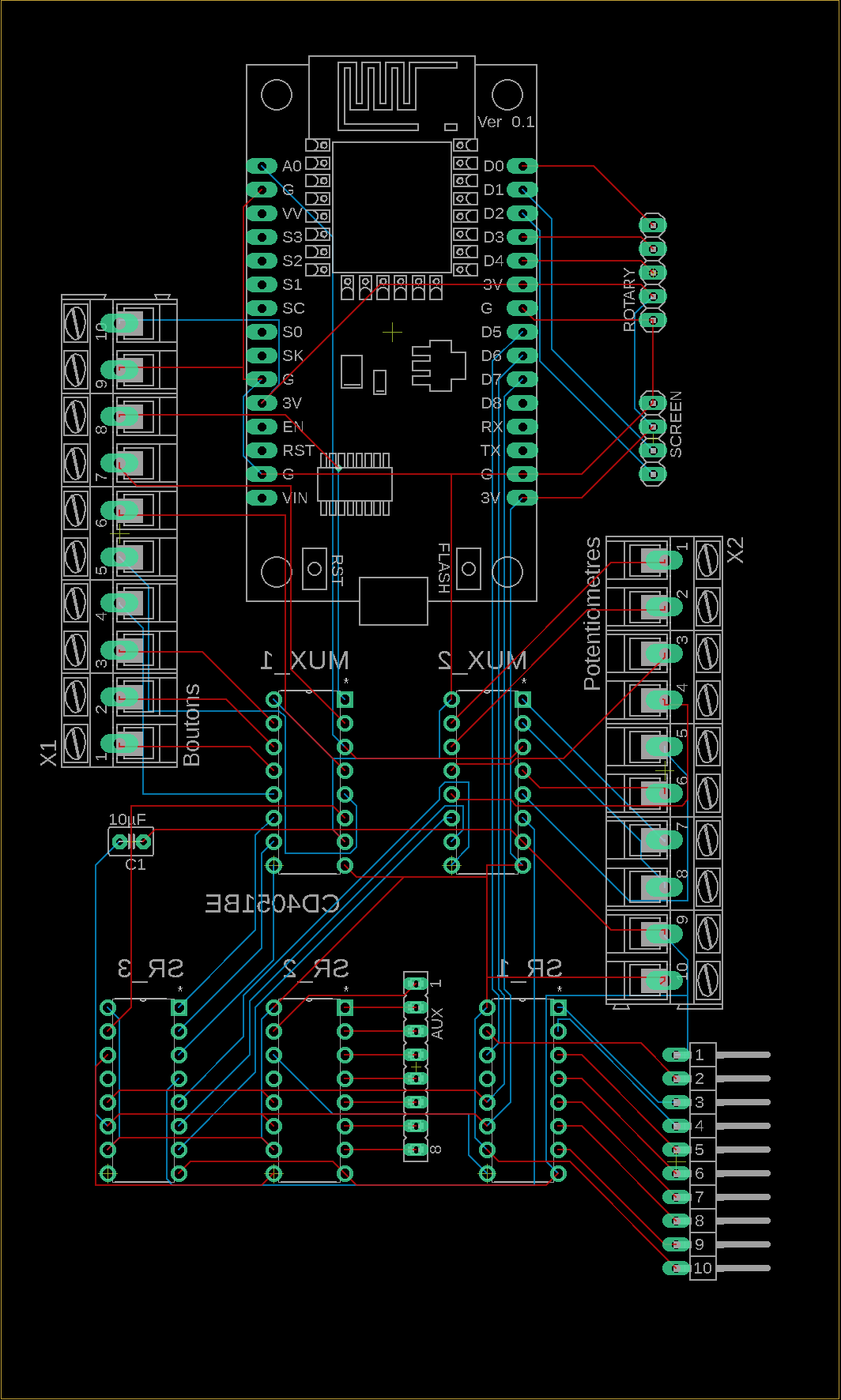
Rotary encoder (avec bouton switch)

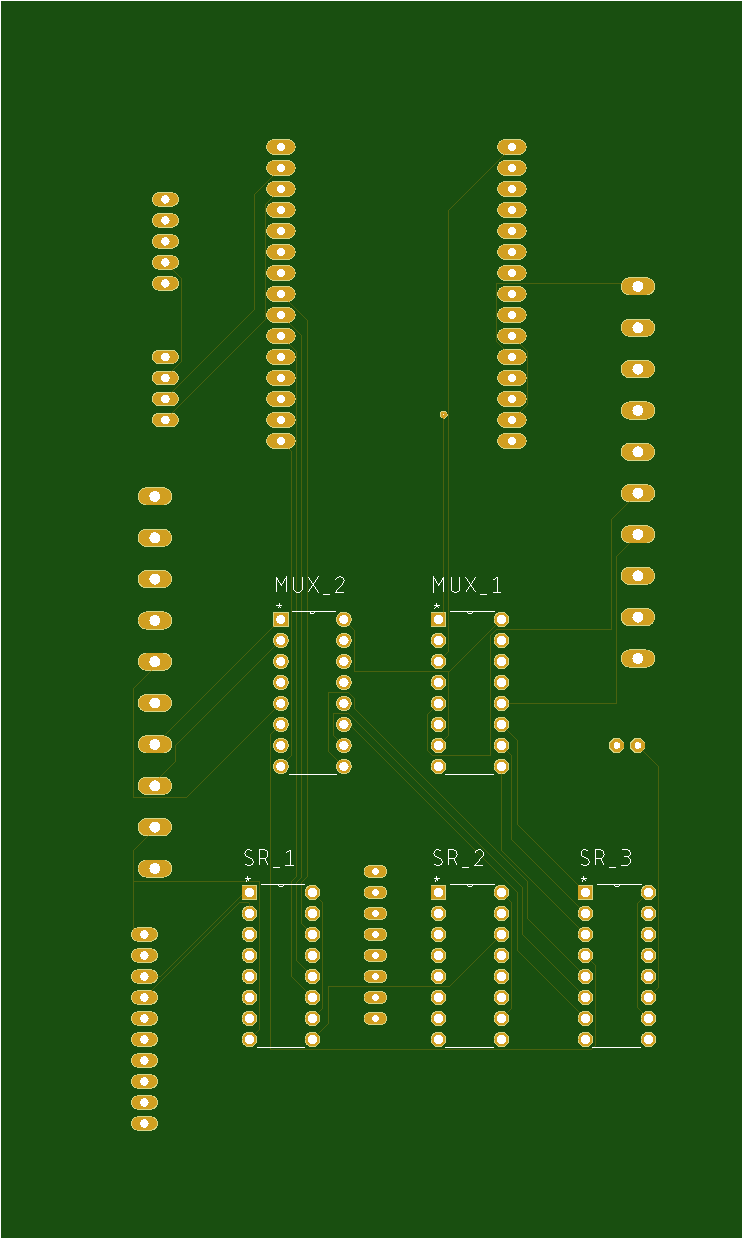
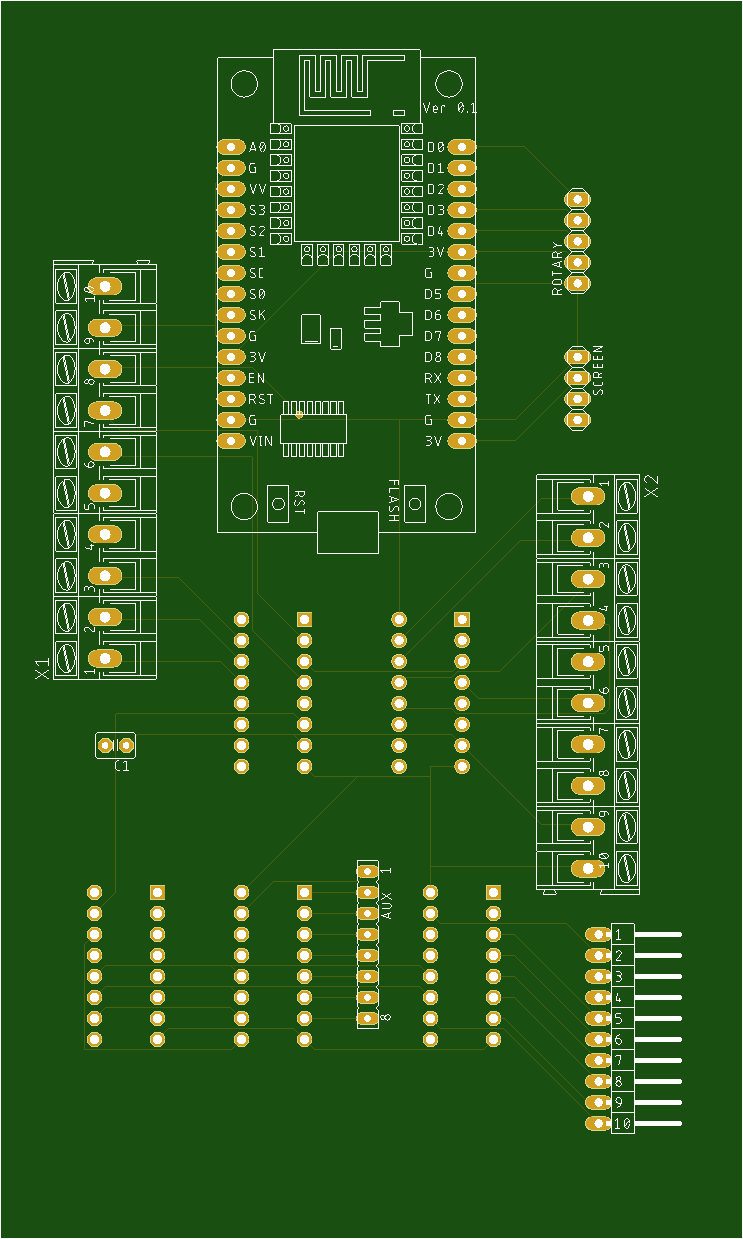
Ecran LCD 2x16 caractères



### Câblage

### Création d’un PCB

Afin de rendre le système plus résistant mécaniquement, le design d’un PCB s’est imposé, afin de pouvoir connecter les différents composant de manière plus efficace et plus pérenne, le tout dans un encombrement minimum afin d’être fixé sur le châssis métallique du siège.



## Conception informatique

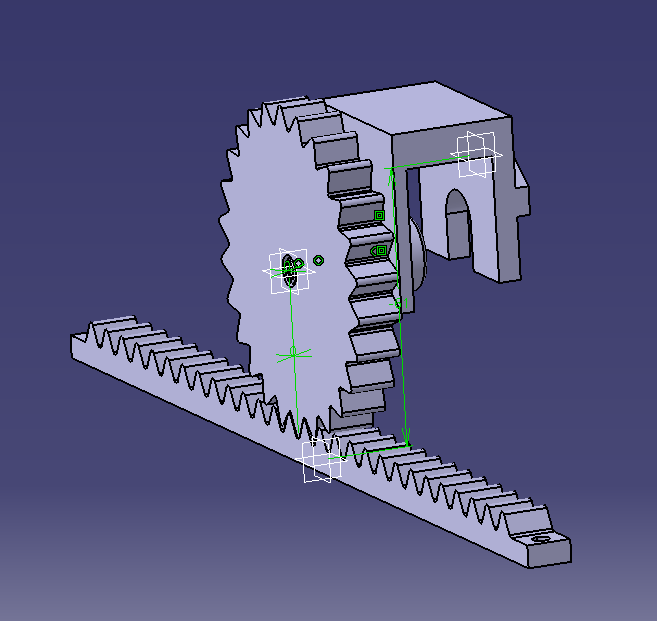
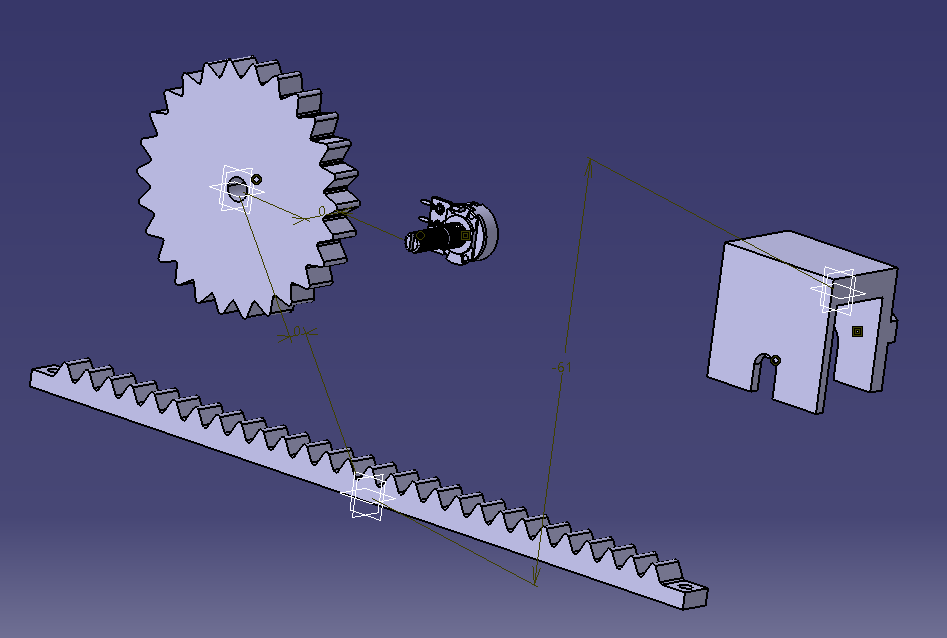
Langage : c++

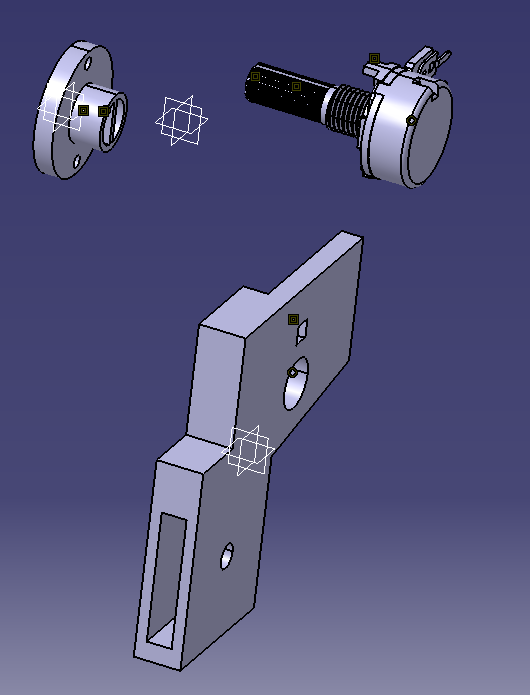
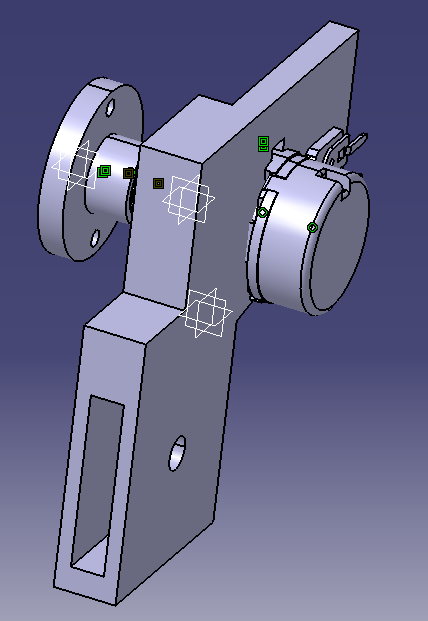
Fichiers principaux :

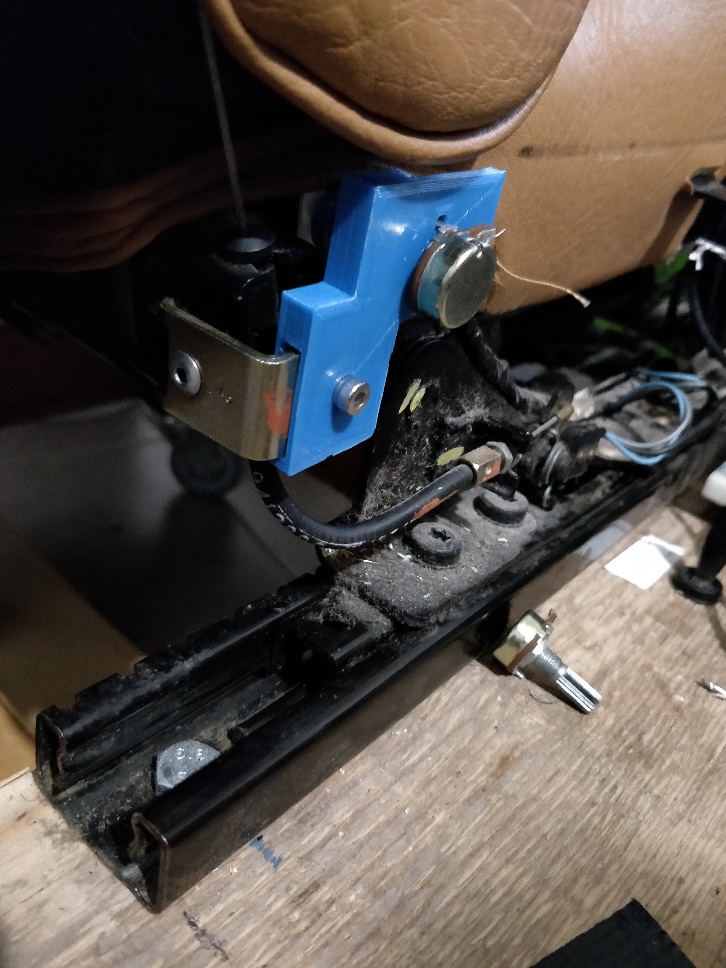
* Desk.ino //fichier de code principal
  + Implémentation logique du code
  + Fonctions de connexion et utilisation wifi
  + Fonctions de communication JSON
  + Positions
* MotorMux.h //classe contrôlant un moteur du siège
  + Contrôle les relais 2 à 2
  + Est responsable de connaître sa position
  + Gère ses boutons de contrôle
  + Est capable de se déplacer à une position donnée
* Seat.h //classe regroupant les 4 moteurs et gérant leur opération
  + Gère les 4 moteurs
* Shiftreg.h //classe permettant de contrôler un registre à décalage (ou plusieurs en série)
* CustomType4051Mux.h //classe permettant de contrôler un multiplexeur via un shiftreg
* Menu.h // classe permettant d’implémenter toute la logique des menus utilisateurs

## Conception mécanique

Une des parties les plus complexes vis-à-vis de nos compétences aura été la conception mécanique d’adaptateurs pour fixer les capteurs de position nécessaires à la fonctionnalité « mémoire de position ». Jusqu’ici, deux adaptateurs ont été créés : l’adaptateur pour le capteur « avancement » et l’adaptateur pour le mouvement « hauteur »





# Bilan

## Résultats obtenus

Siège fonctionnel : boutons utilisables, mémoires de position (à moitié) implémentées, connexion au wifi (pas testé) & contrôle via wifi (pas testé)

Menus pas encore implémentés

## Travail restant

Finir les menus

Implémenter multi-user -> stocké dans l’eeprom

Finir l’API de contrôle wifi

Implémenter les API google pour contrôle via google home

## Difficultés rencontrées

Prise en main des différentes technos :

* ESP8266
* Registres à décalage
* Multiplexeurs
* Schémas électriques (Eagle Autocad)
* CAO (Catia)

Electronique

* Câblage
* Soudures, branchements
* Prise en main des composants

Choix informatiques

* Test de FreeRTOS sous Arduino
* Création de librairies spécifiques à chaque elt
* Prise en main des diverses APIs utilisées
* Choix d’implémentation d’un protocole de communication

## Ouverture

Le siège représente finalement une preuve de concept d’un produit pouvant éventuellement être commercialisé ; le fauteuil connecté. Celui-ci possèderait toutes les fonctionnalités du cahier des charges, mais aurait bien sûr des qualités d’objet en production : un châssis plus léger et plus adapté, un design plus orienté fauteuil ou chaise de bureau selon les modèles, et une alimentation électrique dédiée avec des batteries plus performantes.

Tester ESPressif SDK avec freeRTOS

# Sources

Documentation technique siège conducteur

<http://www.coupe406.com/documents/Tutoriels/Procedures/Dossier406EleveSiege.pdf>