



Rapport de Stage 2A

Contrôle d'environnement de fauteuil roulant électrique

Roxanne XU

09 Mai 2022 - 29 Juillet 2022

Remerciements

Je tiens à remercier l'équipe SED (Service d'Expérimentation et de Développement) de l'Inria situé sur le site de Grenoble-Montbonnot pour leur accueil chaleureux durant mon stage.

En particulier, mes maîtres de stage, Christophe Braillon et Roger Pissard, qui ont suivi mon stage pour un des projets de Humanlab Inria dont je parlerai dans la suite de mon rapport. Leur implication au niveau technique et organisationnel fut primordiale pour le bon déroulement du projet.

Je tiens également à remercier Samuel Heimann et Eric Boucher pour leur sortie en parapente.

Sommaire

1	Introduction et Mise en contexte	5
1.1	Contexte général	5
1.2	Fonctionnement du fauteuil roulant : Reprise des travaux précédents	6
1.2.1	Magic Joystick 2019	6
1.2.2	Magic Joystick 2020	6
1.2.3	Magic Control 2021	8
1.3	Problématique	8
1.4	Objectifs	9
1.5	Plan	9
2	Architecture du projet	10
2.1	Composants matériels et architecture matérielle	10
2.2	Architecture logicielle	11
3	Travail effectué	13
3.1	Configuration de la carte Raspberry Pi	13
3.2	Acces Point Wifi	13
3.3	Bluetooth	14
3.4	Interface Web	14
3.4.1	Inkscape	14
3.4.2	Pages HTML	15
3.5	RNET et CAN	16
3.6	Magic Joystic	16
3.7	Infrarouge	17
3.8	Modélisation 3D	18
4	Conclusion	21
4.1	Bilan	21
4.2	Piste pour la suite	21
5	Liens Utiles	22

Liste des figures

1.1	Localisation des centres de recherches INRIA	5
1.2	Schéma du principe de contrôle à distance du fauteuil	6
1.3	Schéma de l'architecture logicielle	7
1.4	Joystick magnétique	7
1.5	Joystick mécanique	7
1.6	Schéma général du fonctionnement du fauteuil	7
1.7	Protocole pour obtenir la bonne séquence d'initialisation	8
1.8	Architecture logicielle via un broker MQTT	9
2.1	Architecture matérielle	10
2.2	Architecture logicielle	11
2.3	Capture d'écran de la page web <i>Supervisor Status</i>	12
3.1	Exemple d'icône générée avec <i>inkscape</i>	14
3.2	Page d'accueil	15
3.3	Page d'accueil du fauteuil roulant	15
3.4	Contrôle de l'assise du fauteuil	15
3.5	Page des modules Infra-rouge	15
3.6	Page des boutons d'une télécommande TV	15
3.7	Connexion du "magic jostick" au convertisseur ASD1015	16
3.8	Connexion du "magic jostick" avec un connecteur DB9	16
3.9	Schéma kicad du montage électronique avec à gauche la partie émetteur et à droite la partie récepteur	17
3.10	Photo du boîtier infrarouge de l'intérieur	18
3.11	Photo du boîtier infrarouge avec les 4 LED	18
3.12	Capture d'écran du timer	19
3.13	Capture d'écran de la fenêtre pop-up après le timer	19
3.14	Pièce 3D de la partie basse du boîtier	20
3.15	Pièce 3D de la partie haute du boîtier	20

Glossaire

- MHK : My Human Kit
- HLI : Humanlab Inria
- XAC : Xbox Adaptive Controller
- CAN : Controller Area Network, bus de communication série utilisé en électronique
- RNET : protocole de communication pour les modules de fauteuil roulant (contrôle de la vitesse, direction etc.. du fauteuil) développé par PG Drives Technology
- JSM : Joystick Module, joystick d'assistance à l'arrière du fauteuil, prioritaire sur le contrôle du déplacement
- I2C : Inter-Integrated Circuit, bus informatique conçu par PHILIPPS pour des applications de domotique et d'électronique domestique
- Broker MQTT : serveur qui reçoit tous les messages des clients (informations publiées) et les aiguille vers les clients de destination approprié (clients abonnées)
- MQTT : Message Queuing Telemetry Transport, protocole open source de messagerie "publish-suscribe" basé sur TCP/IP
- TCP : Transmission Control Protocol, protocole de transmission correspondant à la couche de transport (niveau 4) dans le modèle OSI
- IP : Internet, protocole de communication correspondant à la couche 3 dans le modèle OSI
- OSI : Open Systems Interconnection
- HTTP : Hypertext Transfer Protocol
- GPIO : General-purpose input/output
- BT : Bluetooth
- SSH : Secure Shell Protocol, un protocole de communication sécurisé permettant une connexion à distance
- DHCP : Dynamic Host Configuration Protocol,
- XML : Extensible Markup Language
- SDP : Service Discovery Protocol, protocole permettant la communication entre un serveur et un client
- HID : Human Interface Descriptor

1. Introduction et Mise en contexte

1.1 Contexte général

L'inria est un institut national de recherche en sciences et technologies du numérique. Il existe 9 centres localisés dans plusieurs grandes villes françaises. Au sein de cet institut, le travail de recherche sur différentes thématiques est réalisé par des équipes-projets qui bénéficient de services de support. J'ai été accueillie avec Brice Arnaud dans le Service d'Expérimentation et de Développement pour mon stage. Nous avons travaillé sur un projet opensource permettant d'améliorer la qualité de vie de Jonathan, une personne atteint d'un handicap sévère.

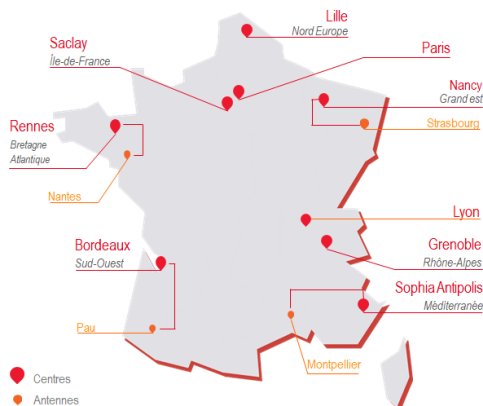


Figure 1.1: Localisation des centres de recherches INRIA

Ce projet a été initié par l'association MHK qui invente, fabrique et partage des solutions d'aides techniques pour et avec des personnes en situation de handicap. Pour chaque projet, il existe un porteur de projet qui a une problématique et un besoin particulier. Le Humanlab permet de réaliser ces solutions grâce à la disposition de certains espaces de travail et de prototypage comme le HLI.

1.2 Fonctionnement du fauteuil roulant : Reprise des travaux précédents

1.2.1 Magic Joystick 2019

Le projet Magic Joystick a été réalisé et pensé lors du Fabrikarium chez ArianeGroup du 16 au 18 octobre 2019. L'un des objectifs consistait à permettre à un usager ayant une faible sensibilité des doigts de jouer aux jeux vidéo grâce à une manette ergonomique faible force, l'autre était de contrôler le fauteuil roulant électrique à distance.

Pour jouer aux jeux vidéos, il faut connecter une interface Joystick à une XAC. Le joystick était constitué de 2 axes réglables en débattement avec une force maximum réglable de 5g à 20g pour un déplacement complet du joystick. Pour connecter la XAC au joystick du fauteuil roulant, une liaison électronique via CAN a été établie entre le joystick et le fauteuil roulant grâce à une carte raspberry Pi. Une manette de xbox peut donc contrôler le fauteuil roulant.

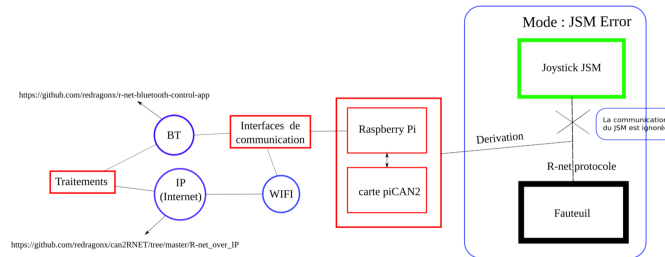


Figure 1.2: Schéma du principe de contrôle à distance du fauteuil

Le joystick du fauteuil est "court-circuité". Un autre peut contrôler le déplacement.

1.2.2 Magic Joystick 2020

Le projet porté par Jonathan s'est tourné sur la réalisation d'un joystick faible force, afin d'être manipulable avec une mobilité très réduite, pouvant être utilisé directement sur un fauteuil électrique.

Le protocole R-Net permet de contrôler le déplacement du fauteuil via un bus CAN. (figure 1.3)

Deux solutions de joystick ont été développées : l'une magnétique reposant sur l'utilisation d'un aimant fixé sur le doigt, l'autre mécanique en déplaçant un aimant fixé sur une tige. Dans les deux cas, un capteur à effet hall récupère le mouvement du doigt. (figure 1.4 & figure 1.5)

Pour aider à comprendre le fonctionnement, on peut voir sur la figure 1.6 les différents composants, tous centrés autour d'une carte Raspberry Pi.

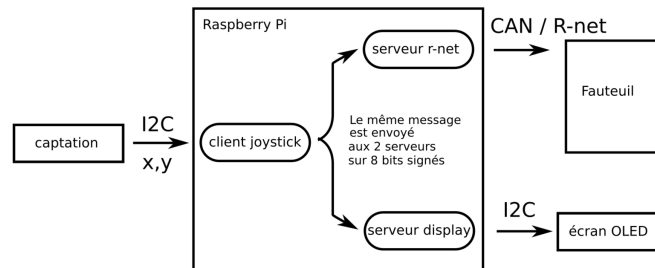


Figure 1.3: Schéma de l'architecture logicielle

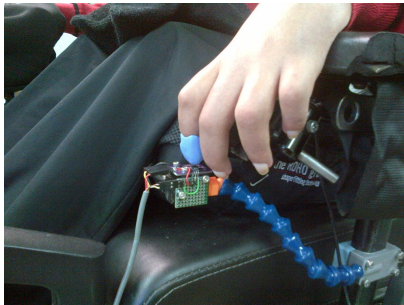


Figure 1.4: Joystick magnétique

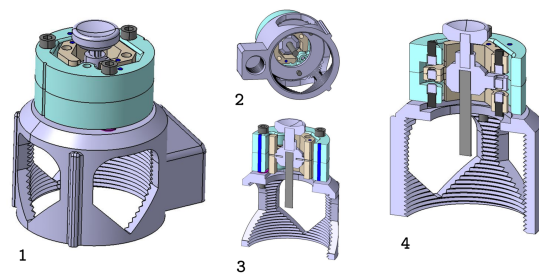


Figure 1.5: Joystick mécanique

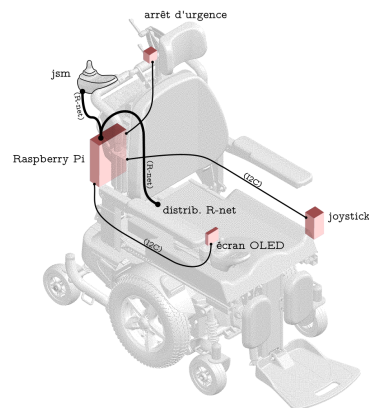


Figure 1.6: Schéma général du fonctionnement du fauteuil

1.2.3 Magic Control 2021

Il s'agit du projet auquel Christophe, mon maitre de stage, avait participé. Le but était de permettre à un usager d'être autonome en contrôlant son environnement à l'aide d'un joystick et d'un bouton poussoir.

Pour permettre à l'utilisateur d'accéder à son téléphone portable, une simulation d'une souris bluetooth a été faite. Le joystick du fauteuil se comporte comme une souris d'un ordinateur classique.

Une application web a également été développée, servant d'interface de contrôle entre l'utilisateur et le fauteuil.

Afin que le fauteuil puisse démarrer, une séquence de trame d'initialisation est nécessaire. En effet, chaque joystick et chaque fauteuil génèrent une séquence propre.

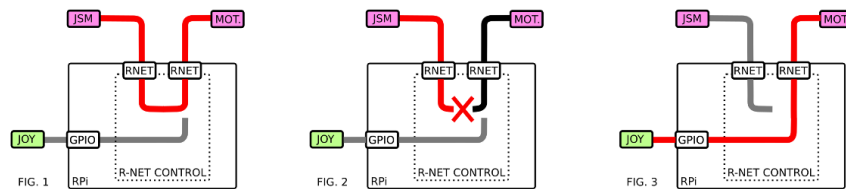


Figure 1.7: Protocole pour obtenir la bonne séquence d'initialisation

La carte Raspberry Pi se situe entre le JSM et le moteur. Elle "surveille" et enregistre les trames. Ensuite, une fois la séquence d'initialisation enregistrée, elle peut être rejouée et permettre au joystick "maison" de prendre le contrôle du fauteuil en envoyant les commandes au moteur.

Le broker MQTT permet non seulement à chaque composant de transmettre des données, mais également la récupération de ces données afin d'avoir un affichage des données à l'utilisateur par exemple sur son téléphone. (figure 1.8)

1.3 Problématique

Un fauteuil roulant coûte cher, surtout s'il est équipé d'un moteur électrique. Un modèle basique vaut aux alentours des 10 000€. Cependant, si l'utilisateur a un handicap particulier, il est nécessaire d'ajouter des options qui peuvent atteindre des sommes astronomiques. Par exemple, pour pouvoir contrôler les volets électriques de chez soi, relier le fauteuil au téléphone portable ...

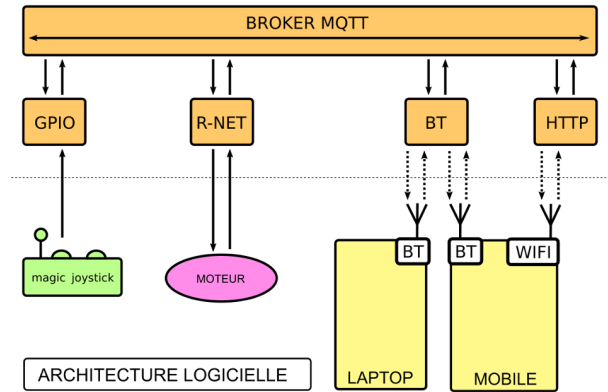


Figure 1.8: Architecture logicielle via un broker MQTT

C'est pour cela que le Humanlab existe. A chaque fois que l'utilisateur souhaite ajouter une fonctionnalité supplémentaire, il faut compter des milliers d'euros à verser. Non seulement les adaptations particulières coûtent chères, mais parfois le constructeur ne les proposent plus car cela n'est pas rentable pour eux.

Lors de ce stage, mon travail a consisté à comprendre le fonctionnement du fauteuil roulant et ajouter des fonctionnalités permettant de contrôler l'environnement de l'utilisateur, tels que la domotique.

L'handicap majeur de Jonathan est la tétraplégie. Cependant, il a la possibilité de bouger très légèrement quelques doigts de sa main.

1.4 Objectifs

L'objectif de ce stage est d'améliorer l'architecture et le développement du dispositif réalisé en 2021 pendant les trois jours de Fabrikarium.

1.5 Plan

Pendant ce stage, l'amélioration du dispositif existant a été permise grâce à la mise en place de plusieurs scénarii de position entre le Jostick, le JSM et la carte Raspberry Pi, une adaptation de la carte Raspberry Pi 3 à la 4, et surtout le lien entre la page web avec toutes les fonctionnalités du fauteuil.

Une présentation de l'architecture du projet final sera donnée, suivi par les réalisations effectuées lors de ce stage. J'ai pu travailler sur la configuration des accès sans fils, l'interface web permettant de contrôler le fauteuil, ainsi qu'une extension infrarouge pour contrôler une télévision.

2. Architecture du projet

2.1 Composants matériels et architecture matérielle

Le fauteuil utilisé était un ancien modèle utilisé par Jonathan. Le fonctionnement général est celui présenté dans l'introduction (figure 1.6)

En ce qui concerne l'architecture matérielle, nous avons repris la structure du projet initial.

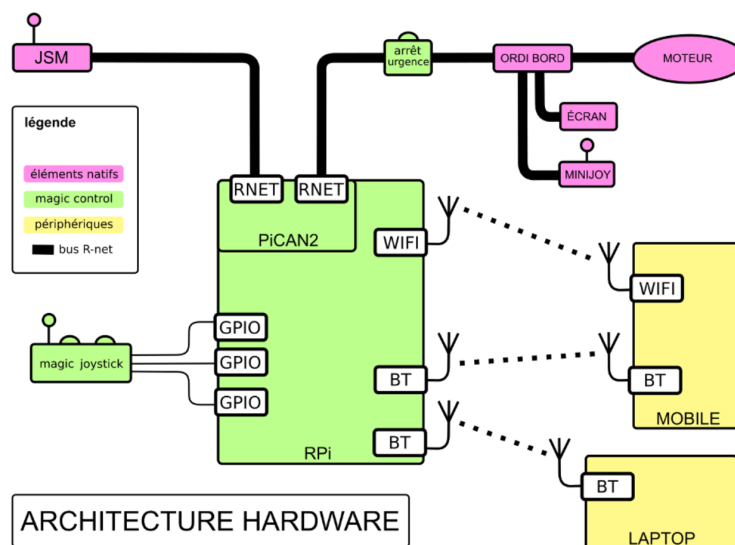


Figure 2.1: Architecture matérielle

La carte Raspberry Pi (PRi) s'insère entre le JSM et le moteur. Cette position nous permet dans un premier temps de surveiller les trames qui circulent sur le bus de communication CAN supportant le protocole du fabricant RNET. Ensuite, une fois ces trames analysées, il est alors possible d'envoyer nos propres trames avec les données récoltés par le magic joystick, joystick "maison" adapté au handicap de l'utilisateur, afin que le moteur puisse agir correctement sur le déplacement du fauteuil.

La raspberry possède également une interface Wifi et Bluetooth, essentiels pour la communication avec un téléphone portable et la mise en place d'un serveur web.

2.2 Architecture logicielle

Le broker MQTT est l'élément central. Il fait le lien entre les *publishers* et les *subscribers*. Les clients de MQTT peuvent être soit *publishers*, c'est-à-dire qu'ils souhaitent envoyer une information telle que les positions $[x, y]$ du magic joy sur un *topic* prédéfini, soit *subscribers*, c'est-à-dire qu'ils s'abonnent à un *topic* et reçoivent tous les messages de celui-ci. Les *topics* permettent de classer et séparer les informations.

RNET permet de substituer les informations envoyées par le JSM par celle du magic joy au moteur.

Joystick récupère les coordonnées via un bus I2C permettant la communication entre le capteur à effet Hall et la carte Raspberry Pi.

Le service Bluetooth permet à la carte Raspberry de se faire passer pour une souris, et donc d'agir comme tel sur le téléphone.

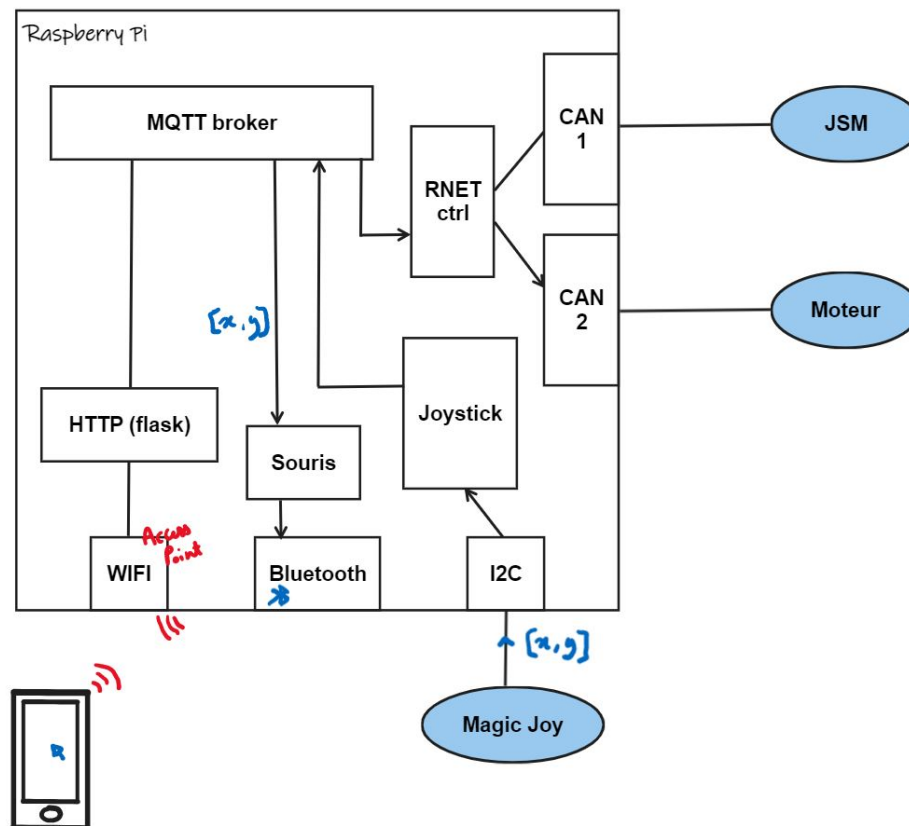
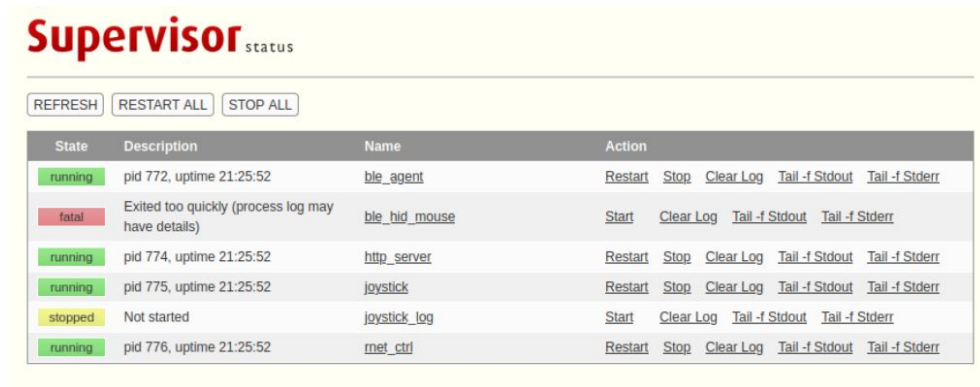


Figure 2.2: Architecture logicielle

Afin de faciliter le travail de développement, un système de contrôle et de surveillance de processus a été utilisé : *Supervisor*. Une fois la configuration faite avec le nom du service, les fichiers source et surtout la commande à lancer pour les exécuter le code, il est possible d'arrêter et de relancer les différents services facilement.



State	Description	Name	Action
running	pid 772, uptime 21:25:52	ble_agent	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr
fatal	Exited too quickly (process log may have details)	ble_hid_mouse	Start Clear Log Tail -f Stdout Tail -f Stderr
running	pid 774, uptime 21:25:52	http_server	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr
running	pid 775, uptime 21:25:52	joystick	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr
stopped	Not started	joystick_log	Start Clear Log Tail -f Stdout Tail -f Stderr
running	pid 776, uptime 21:25:52	rnet_ctrl	Restart Stop Clear Log Tail -f Stdout Tail -f Stderr

Figure 2.3: Capture d'écran de la page web *Supervisor Status*

3. Travail effectué

3.1 Configuration de la carte Raspberry Pi

Le modèle utilisé fut une carte Raspberry Pi 4 alors que dans les projets précédent, il s'agissait d'une carte Pi 3. Un certain nombre d'éléments a dû être adapté afin que ce qui fonctionnait précédemment le soit toujours sur la Raspberry Pi 4.

Tout d'abord, le logiciel *Raspberry Pi Imager* a été utilisé pour installer le système d'exploitation *Raspberry Pi OS Lite 32-bit* afin d'avoir le système le plus basique possible et éviter des problèmes d'interférences de différents services, en particulier pour le bluetooth. Le système installé repose sur Debian.

3.2 Acces Point Wifi

Le premier travail effectué a été la mise en place d'un point d'accès Wifi. Cela permet une connexion sans fil à la carte pour non seulement développer le code lors de la phase de conception en SSH, mais aussi à l'utilisateur de pouvoir se connecter à la page web dédiée au contrôle du fauteuil et au pilotage de son environnement s'il a une maison connectée.

Deux programmes sont nécessaires pour la configuration d'un point d'accès wifi : *hostapd* et *dnsmasq*. Dans les fichiers de configuration, *hostapd* sert à spécifier les caractéristiques du réseau : il faut choisir le nom du point d'accès, le mot de passe du point d'accès et d'autres paramètres.

Comme dans tous les réseaux, chaque appareil qui s'y connecte doit avoir une adresse IP. Pour se faire, le protocole DHCP permet d'assurer la configuration automatique des paramètres IP d'une machine. Il permet de choisir le nombre maximum d'appareils qui peuvent s'y connecter, d'attribuer automatiquement une adresse IP et un masque de sous-réseau. DHCP peut aussi configurer l'adresse de la passerelle par défaut, c'est-à-dire celle de la carte Raspberry Pi. Le programme *dnsmasq* réalise cette attribution automatique.

3.3 Bluetooth

La gestion de la connexion Bluetooth a été une tâche assez complexe. Comme précédemment mentionné dans la [partie 3.1](#), l'utilisation d'une carte Raspberry Pi 4 avec un système d'exploitation plus avancé avec un bureau (comme sur un "vrai" ordinateur) a posé quelques soucis. Le plus simple a été alors d'utiliser un système d'exploitation sans interface graphique, d'où l'importance d'une connexion SSH. Finalement, grâce à l'aide du maître de stage Christophe, et celle de Nicolas Turro travaillant dans l'équipe SED, la connexion bluetooth entre la carte Raspberry et le téléphone portable a pu fonctionner correctement.

Concernant l'émulation de la souris bluetooth sur le téléphone, un fichier xml permettant de décrire les caractéristiques du service bluetooth. Ce fichier contient une liste de HID (Human Interface Descriptor), correspondant à une suite hexadécimale formattant les trames envoyées via le service bluetooth. La trame bluetooth contient en entête la taille et le type, et en corps les données. Dans notre cas, les données envoyées sont le click (droit ou gauche), le déplacement sur l'axe des x et le déplacement sur l'axe des y.

3.4 Interface Web

Flask est un micro framework open source pour le développement web en python. Il a été utilisé pour générer le serveur HTTP permettant d'héberger la page web de contrôle du fauteuil et de l'environnement de l'utilisateur.

Pour aider l'utilisateur, différentes pages ont été créées. Une page d'accueil, une page pour contrôler la position de l'assise etc..

3.4.1 Inkscape

Afin d'avoir un rendu clair et intuitif, des icônes ont été utilisées et générées par le logiciel *inkscape*. Ce logiciel permet de générer des fichiers svg, permettant d'avoir une définition optimale des icônes sans avoir à se soucier du problème de pixels lors d'agrandissement ou réduction d'image.



Figure 3.1: Exemple d'icône générée avec *inkscape*

3.4.2 Pages HTML

L’affichage des pages HTML se fait grâce à aux fichiers html pour le fond et css pour la forme.

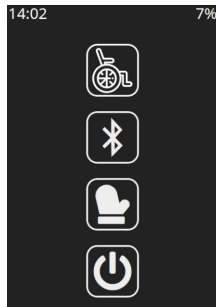


Figure 3.2: Page d’accueil

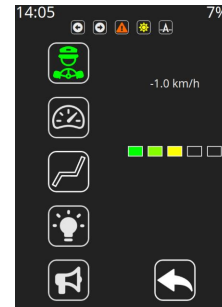


Figure 3.3: Page d’accueil du fauteuil roulant

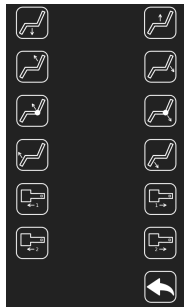


Figure 3.4: Contrôle de l’assise du fauteuil

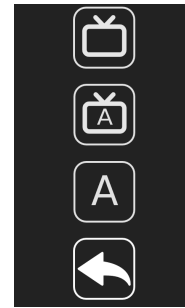


Figure 3.5: Page des modules Infra-rouge

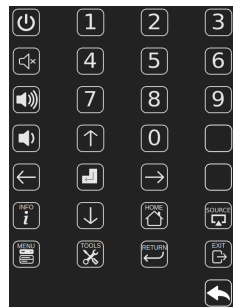


Figure 3.6: Page des boutons d’une télécommande TV

Pour aider lors du développement, les valeurs non cohérentes comme une vitesse négative permettent de savoir si la page est connectée au fauteuil ou non.

Pour que les pages soient interactives, et puissent notamment afficher l’heure, le retour de vitesse, le niveau de batterie, allumer les voyants (clignotants), il est nécessaire d’ajouter du code en javascript pour éviter de devoir recharger les pages manuellement.

3.5 RNET et CAN

Cette partie là a été développée par Brice. Globalement, une première étape de déchiffrement des trames interceptées était primordiale. Pour cela, un script pour afficher les trames a permis une analyse plus fine et d'identifier les bits codant telle ou telle information. Ensuite, lors du contrôle du fauteuil par le magic joystick, il "suffisait" de rejouer les trames.

Avant de pouvoir rejouer les trames, il faut d'abord trouver la séquence d'initialisation permettant au fauteuil de s'allumer.

3.6 Magic Joystic

Le travail déjà effectué concernait le magic joystick mécanique en [figure 1.5](#). Mon collègue Brice a étudié, avec l'aide du maître de stage Christophe, plus en détail le comportement joystick initial pour pallier aux effets de rotation indésirables observés lors de l'utilisation du fauteuil en conditions réelles.

Afin de pouvoir travailler avec un joystick fonctionnel, nous avons utilisé un déjà présent dans les locaux de l'Inria, puis connecté celui-ci à un convertisseur analogique-numérique *ADS1015*. Ensuite, nous avons soudé des connecteurs pour pouvoir facilement brancher et débrancher ce joystick afin d'en utiliser un autre par exemple.

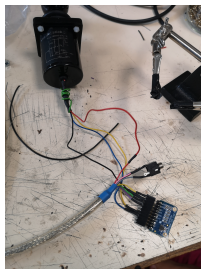


Figure 3.7: Connexion du "magic joystick" au convertisseur ASD1015

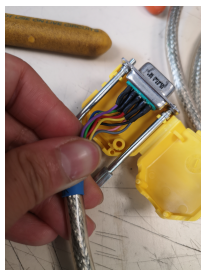


Figure 3.8: Connexion du "magic joystick" avec un connecteur DB9

Pour l'émulation de la souris bluetooth, une conversion des coordonnées récupérées par le magic joystick en vitesse a été nécessaire pour effectivement émuler le déplacement du curseur.

3.7 Infrarouge

Pour améliorer le quotidien de l'utilisateur, des capteurs et récepteurs infrarouges ont été connectés à la carte Raspberry Pi. Le fonctionnement implémenté est très simple : tout d'abord, il faut enregistrer les commandes, puis les rejouer.

Pour cela, des résistances, transistors, LED Infrarouges et capteurs infrarouges doivent être connectés sur une carte électronique.

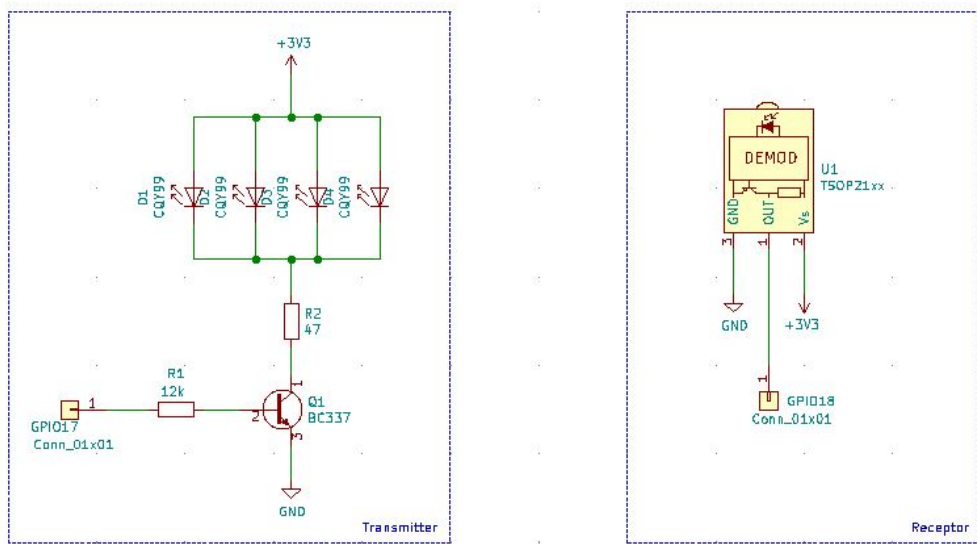


Figure 3.9: Schéma kicad du montage électronique avec à gauche la partie émetteur et à droite la partie récepteur

Afin de pouvoir effectivement enregistrer les commandes, l'interface web demande à l'utilisateur d'appuyer sur la bonne commande à enregistrer. L'utilisateur a 10 secondes pour le faire. Ensuite, l'utilisateur a le choix entre rejouer la commande déjà enregistrée, la ré-enregistrer, la supprimer ou la valider.

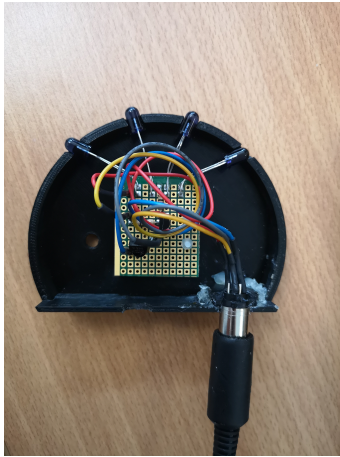


Figure 3.10: Photo du boîtier infrarouge de l'intérieur



Figure 3.11: Photo du boîtier infrarouge avec les 4 LED

Il existe des bibliothèques Python qui codent les signaux de télécommandes TV les plus utilisées. Il suffit alors juste de configurer le type de la télécommande pour la simuler par la carte Raspberry. Cependant, par soucis de liberté et d'adaptabilité, il a été plus judicieux d'enregistrer les commandes "brutes" pour pouvoir réutiliser le même processus pour tous les appareils commandés par infrarouge.

3.8 Modélisation 3D

Lors de ce stage, j'ai pu utilisé le logiciel Freecad pour dessiner le boîtier noir en 3D, avant de lancer son impression. Pour cela, le boîtier a été dessiné en deux parties, afin d'avoir une meilleure impression 3D: Cela rendait plus facile l'insertion des LEDs et permettait également d'insérer la carte électronique à l'intérieur. Ci-dessous se trouvent les modèles 3D créés.

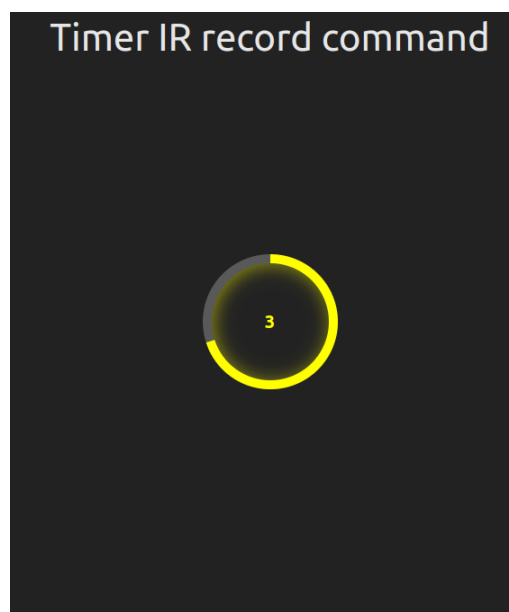


Figure 3.12: Capture d'écran du timer

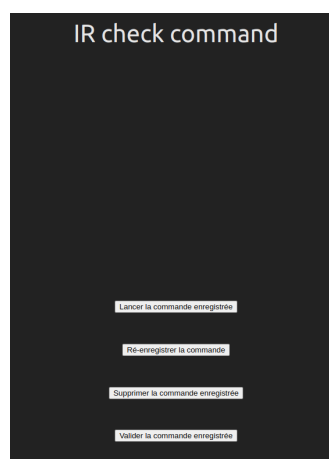


Figure 3.13: Capture d'écran de la fenêtre pop-up après le timer

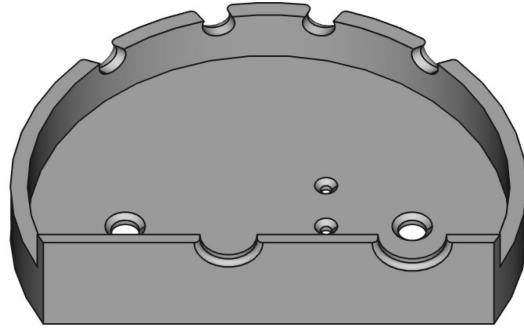


Figure 3.14: Pièce 3D de la partie basse du boîtier

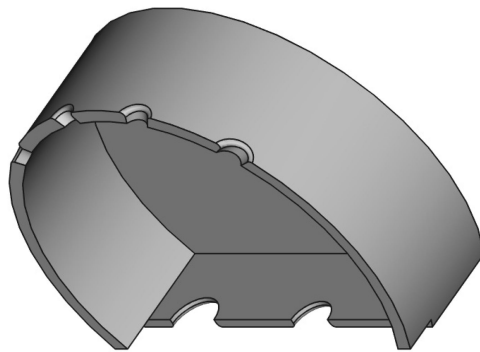


Figure 3.15: Pièce 3D de la partie haute du boîtier

4. Conclusion

4.1 Bilan

Ce stage m'a permis d'apprendre plein de choses, au niveau technique évidemment mais également au niveau du travail en équipe. Dans ce projet, j'ai pu aborder toutes les techniques de conception. J'ai pu apprendre et améliorer mes compétences mécaniques avec la réalisation du boîtier grâce au logiciel Freecad et l'utilisation d'une imprimante 3D. J'ai pu consolider mes compétences électronique et logiciel avec les travaux sur le système réseau et l'interface web.

Le fait de travailler sur un même projet avec un autre élève, Brice, a permis de confronter nos avis, parfois divergents. Cela nous a permis d'en discuter et de trouver la meilleure solution. L'intégration à l'équipe s'est faite très rapidement. J'ai eu plaisir de venir quotidiennement travailler dans les locaux de l'Inria Grenoble-Montbonnot.

4.2 Piste pour la suite

Une limitation au niveau du téléphone est l'accès à une connection internet. En effet, lorsque le téléphone de l'utilisateur sera connecté au point d'accès wifi de la carte Raspberry, il ne lui sera plus possible de se connecter à un autre réseau wifi connecté à internet. Il faut donc trouver une solution pour que le téléphone puisse avoir une connection internet (4G ou wifi) et une autre à la carte Raspberry.

La partie domotique peut être développée et améliorée avec d'autres appareils afin qu'un usager puisse être autonome chez lui. Cela fera sans doute partie d'un prochain Fabrikarium.

5. Liens Utiles

Dépôt github pour le code source : https://github.com/roxu99/magic_joystick.git
Il faut aller dans la branche cleanup_proposal

<https://project.inria.fr/humanlabinria/fr>

<https://myhumankit.org/le-humanlab/>

<https://wikilab.myhumankit.org//>