

MODULE D'AIDE AU DEPLACEMENT POUR FAUTEUIL ROULANT ELECTRIQUE



Table des matières

Introduction	3
I. Cahier des charges initial.....	3
II. Cahier des charges actuel.....	5
III. Solutions techniques	6
IV. Déroulement du projet.....	7
a. Développement d'un module Bioloid.....	8
b. Etude du fauteuil roulant électrique	9
c. Etude de la commande du FRE	10
V. Partie Hardware	12
a. Alimentation de l'Arduino	12
b. Récupération des informations moteurs	13
c. Récupération des signaux PWM	15
VI. Partie Software.....	17
a. Capteurs à ultrasons SRF08	17
b. Utilisation de l'Arduino pour la lecture et l'écriture de signaux	21
c. Algorithme de détection d'obstacles.....	22
VII. Nos réalisations	24
VIII. Perspectives d'amélioration	24
Conclusion.....	26

Introduction

L'association "La Vie Autrement", membre du groupement GAPAS (Groupement des Associations Partenaires d'Action Sociale), est une association prenant en charge des personnes pluri handicapées, polyhandicapées, autistes... Ces personnes nécessitent l'aide d'un tiers pour l'exécution des tâches élémentaires de la vie quotidienne.



Afin de prendre en charge toutes ces personnes, La Vie Autrement a ouvert plusieurs établissements :

- 2 instituts d'éducation motrice à Hem et à Wasquehal (La Source et Le Passage) ;
- 2 maisons d'accueil spécialisées (MAS) à Hantay et à Marcq en Baroeul (Le Hameau et La Gerlotte).

Notre projet se déroule en partenariat avec la MAS Le Hameau qui accueille 40 adultes polyhandicapés dépendants dans tous les actes de la vie quotidienne. La moitié de ces résidents se déplacent à l'aide de fauteuils roulants électriques (FRE) dont la commande peut varier (Joystick, boutons...). Cette utilisation des FRE dans un espace comme la maison du Hameau a soulevé des difficultés (collisions entre les résidents, avec les murs, portes etc...).

Pour pallier ces problèmes, l'idée d'un projet en collaboration avec Polytech'Lille est née, le but étant de développer un système permettant d'améliorer la vie en maison d'accueil spécialisée pour tous les résidents en évitant les collisions.

I. Cahier des charges initial

Lors de la présentation du projet, le but principal était simple : il fallait permettre à un résident se déplaçant en FRE d'éviter les collisions. Pour cela, l'idée principale était de pouvoir détecter des obstacles et de limiter la vitesse du fauteuil selon la distance de ces derniers. Ce système devait répondre à cette attente principale en restant discret et sans créer de situations de blocage où le résident en FRE serait incapable de redémarrer après la rencontre d'un obstacle.

Une première réunion avec les membres de la MAS Le Hameau a donc eu lieu afin de préciser les attentes des résidents et de proposer certaines de nos idées. Les membres référents du projet sont les suivants :

- Mme Corinne Meillier, directrice de l'établissement MAS Le Hameau
- Mme Claire Lescure, ergothérapeute à la MAS Le Hameau
- Sébastien Potiez, technicien chez FACOM MEDICAL

Lors de cette première rencontre avec les membres du projet ainsi que certains résidents, nous avons pu définir des fonctions principales :

- Détecter des obstacles :

Le module anti-collision devra être capable de détecter des obstacles à courte distance. Afin de pouvoir limiter la vitesse du fauteuil, on devra pouvoir détecter des obstacles dans toutes les

directions. Les obstacles majeurs à détecter sont alors des murs, radiateurs, d'autres résidents etc... Les obstacles considérés complexes (tables, plateaux en hauteur) ne seront potentiellement pas détectés. Enfin, l'utilisation du système anti-collision se fera presque exclusivement en intérieur.

- Limiter la vitesse du fauteuil :

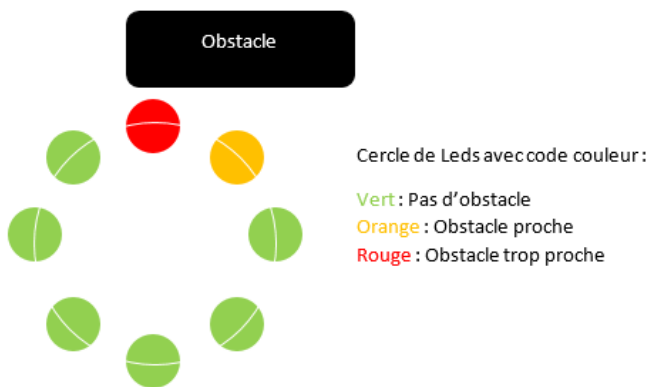
En fonction des informations capteurs, on limitera la vitesse du fauteuil de façon proportionnelle. Plus l'obstacle sera proche, plus la vitesse diminuera jusqu'à provoquer l'arrêt du FRE.

Ces deux fonctions principales ont été retenues après la première réunion, nous avons cependant proposé d'autres idées telles que :

- Signal sonore / visuel lors d'un choc :

Lors d'un choc, le système émet un signal sonore ou un signal clignotant afin de prévenir le personnel de la MAS. Cette idée fut rejetée car les résidents estiment avoir déjà assez de matériel sur leurs fauteuils et désirent un système discret.

- Retour visuel / sonore des obstacles :



Ajouté à leur système de commande, afficher une image des informations capteurs afin d'informer le résident des obstacles présents autour de lui (avec un cercle de Leds et un système de couleurs par exemple). Cependant, la plupart des résidents ne possèdent pas de problèmes de vue et cet investissement ne sera donc pas nécessaire car très peu utile. De même pour un signal sonore, le problème de discrétion est encore soulevé ici.

- Déplacement automatique du fauteuil :

L'objectif ici était qu'une fois l'obstacle rencontré, le FRE puisse de lui-même (sans action du résident) s'écarter ou contourner l'obstacle. Cette idée n'a pas été retenue car lors de la première réunion les résidents ont émis le souhait de garder la liberté de leurs mouvements.

Le fait de proposer des fonctionnalités annexes au système nous a permis de soulever le problème des contraintes du projet. A l'issue de cette première réunion, nous avons alors déterminé les contraintes suivantes :

- Le blocage du fauteuil :

Le système doit permettre aux résidents de se déplacer librement dans la maison sans être bloqués au moindre croisement ou passage de porte. La détection d'obstacle doit donc être précise et adaptée afin de permettre aux résidents de se déplacer dans des zones étroites et de croiser d'autres résidents sans rester bloqués. Cependant, on prévoira tout de même un système de By-Pass permettant de contourner les informations des capteurs afin de reprendre la main sur le fauteuil sans tenir compte de la détection d'obstacles.

- La discrétion du système :

Le système devra être discret, non encombrant et ne devra pas comporter de fils susceptibles d'être arrachés par les résidents.

- La garantie du fauteuil :

Le système anti-collision ne devra pas empêcher le bon fonctionnement de la garantie constructeur du FRE. On ne devra donc pas effectuer de soudure ou modifier l'électronique existante, cependant nous pouvons l'utiliser afin de modifier les signaux de commande.

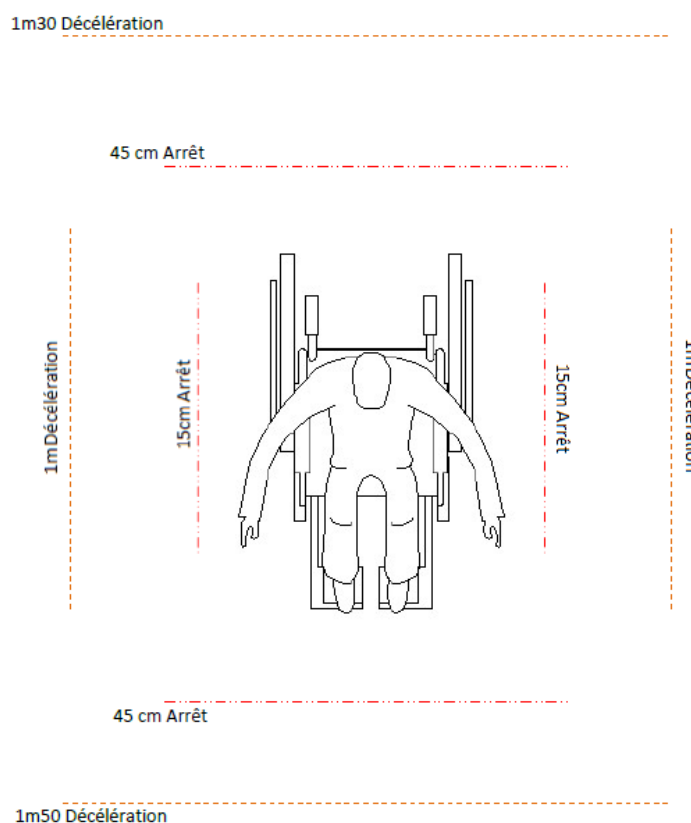
II. Cahier des charges actuel

Après cette première approche du projet, nous nous sommes concentrés sur la rédaction d'un cahier des charges technique qui convenait aux deux parties (Polytech'Lille et La Vie Autrement). En accord avec M. Conrard, nous avons développé la version actuelle du cahier des charges (en annexe 1) dans laquelle nous apportons des précisions techniques sur toutes les fonctionnalités requises du système anti-collision.

Nous avons alors défini la notion d'obstacle afin de correspondre au besoin. Un obstacle est alors un objet possédant une surface pleine entre une hauteur de 30cm et 40cm. De cette façon, on détectera efficacement les murs, radiateurs et autres résidents sans pour autant pouvoir prétendre à la détection d'une table.

De plus, nous avons défini des distances de détection provoquant l'arrêt du fauteuil. Lorsqu'on détectera un obstacle dans la trajectoire (avant ou arrière) du fauteuil à une distance de 45cm ou moins, le fauteuil devra s'arrêter. De même pour un obstacle à moins de 15cm sur les côtés. Cela nous permet de pallier la longueur des repose-jambes présents à l'avant du fauteuil et la longueur du bloc batteries + roues à l'arrière.

N.B : Toutes les distances sont considérées à partir de la base du fauteuil.



Enfin, un indice de protection a été défini afin de prévoir le déplacement en extérieur. Nous avons choisi un IP44 permettant la protection contre des corps solides supérieurs à 1mm et contre les projections d'eau dans toutes les directions. Cette valeur est une valeur de sécurité et est atteignable par la majorité des capteurs.

La totalité du cahier des charges accepté par tous les membres du projet est joint en annexe 1.

III. Solutions techniques

Après avoir validé le cahier des charges, nous avons proposé différentes solutions techniques que nous avons présentées aux membres du projet afin de faire un choix. Les critères principaux de choix des solutions pour ce projet sont :

- Le prix : le module anti-collision doit être d'un prix faible étant donné le prix déjà important des FRE.
- L'encombrement : afin de respecter la contrainte de discrétion imposée par le cahier des charges.

Afin de détecter les obstacles, les propositions étaient les suivantes :

- Télémètres ultrasons ou laser :

- Les télémètres nous permettent bien de détecter des obstacles dans une zone de travail importante avec une précision satisfaisante. Cependant, ces capteurs sont d'une taille non négligeable et d'un prix assez élevé.



- Capteurs infrarouges :

- Les capteurs infrarouges sont efficaces pour détecter des obstacles dans un cône de rayon faible. Ce qui implique la nécessité d'avoir de nombreux capteurs à infrarouges pour surveiller tous les contours du FRE.



- Capteurs ultrasons :

- Les capteurs à ultrasons, malgré leur fiabilité légèrement inférieure aux capteurs infrarouges, permettent d'agir dans des cônes plus importants. De plus, ils sont peu encombrants et possèdent un prix inférieur aux autres solutions.



Afin d'effectuer la fonction « Détecter des obstacles », le choix s'est alors porté vers les capteurs à ultrasons. En effet, ils possèdent un rayon de détection satisfaisant dans les distances étudiées et leur prix est le plus faible. De plus, leur faible taille permet de répondre au critère d'encombrement posé par les résidents. On disposera alors un nombre de capteurs à ultrasons nécessaire à la surveillance de chaque côté du FRE pour le bon fonctionnement du système.

Afin de récupérer les informations de ces capteurs et de modifier les signaux de commande des moteurs en conséquence, nous avons besoin d'une carte microcontrôleur compatible avec des capteurs à ultrasons. Les contraintes ici restent inchangées, à savoir un prix et un encombrement faibles.

Pour les cartes μ C, nous avons proposé deux solutions :

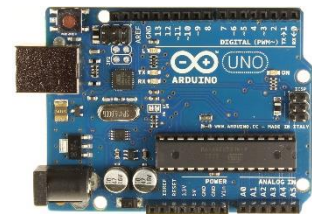
- Carte μ C PIC :

- Pour un prix d'environ 30 à 40€, une carte μ C PIC possède toutes les caractéristiques nécessaires pour recevoir et agir sur des signaux. On y retrouve un nombre d'entrées/sorties suffisant pour notre application et une rapidité de calcul de bonne qualité.



- Carte μ C Arduino :

- Pour un prix d'environ 30 à 45€, une carte μ C Arduino possède les mêmes caractéristiques qu'une carte PIC évoquée précédemment. Cependant, la rapidité de calcul est légèrement inférieure dans la plupart des cas.



Afin de limiter la vitesse du fauteuil, on agira donc sur les signaux de commande envoyés aux moteurs. Pour contrôler ces signaux, nous avons choisi d'utiliser une carte μ C Arduino pour différentes raisons. La robustesse des systèmes Arduino ainsi que nos connaissances déjà acquises sur l'utilisation des cartes μ C de ce type sont des atouts majeurs pour le projet.

IV. Déroulement du projet

Durant la première période du projet, nous avons passé la majeure partie du temps accordée à élaborer un cahier des charges technique qui serait validé par les deux parties participant au projet. Nous avons alors étudié les fauteuils roulants électriques en général, puis grâce aux rencontres avec les résidents et les membres du personnel de la MAS Le Hameau, nous avons pu discuter des besoins et des contraintes afin de décrire une première version de ce cahier des charges. Après de nombreuses modifications, une version complète et validée par tous les membres du projet a été rédigée.

Une fois la validation effectuée, nous avons proposé différentes solutions techniques et nous avons retenu la plus économique et la moins encombrante afin de parfaire aux besoins des résidents. De plus, grâce à cette validation du cahier des charges, nous avons pu commencer à développer un modèle réduit pouvant représenter le déplacement du FRE après l'application de notre système final.

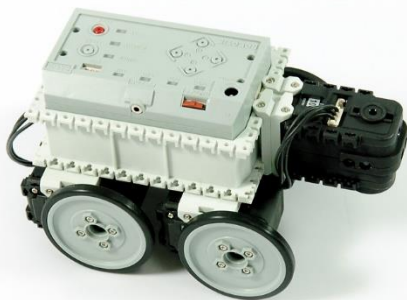
Durant la seconde partie du projet, après avoir reçu le FRE, nous avons pu étudier le système existant et comprendre les bases de son électronique, des signaux de commande etc... A partir des différentes études, nous avons pu mettre en place une stratégie de contrôle en fonction de certains signaux de commande et réfléchir à l'embarquement du système final sur le fauteuil. Cette seconde période du projet s'est donc divisée en deux thématiques : une partie Hardware et une partie Software.

Dans la partie Hardware, on parlera des différentes études des signaux émanant de la commande du fauteuil. On parlera ensuite des différentes approches du projet avant de développer concrètement la solution électronique choisie pour chaque problématique (Alimentation du système, modification des signaux etc...).

Dans la partie Software, on parlera du réseau de capteurs et de l'algorithme de décision. On abordera la technologie de capteurs utilisée et des différentes possibilités disponibles avant de mêler le fonctionnement des capteurs à l'algorithme de décision basé sur les informations reçues. On développera ensuite les solutions apportées aux différentes situations réelles du système réel (blocage du fauteuil, décélération linéaire, réflexion d'ondes).

a. Développement d'un module Bioloid

Afin de simuler le fonctionnement du fauteuil lorsque la détection d'obstacle sera intégrée, nous avons développé un module roulant Bioloid équipé de 4 roues et d'un capteur permettant de repérer des obstacles dans 3 directions (on simulera alors la détection avant, droite et gauche).



Nous avons alors développé 3 différents scénarios :

- Déplacement avant
- Rotation vers la droite
- Rotation vers la gauche

Dans chacun de ces scénarios, nous observons les informations capteur reçues afin d'adapter la vitesse des roues en fonction de la distance des obstacles détectés.

Pour cela, nous avons défini plusieurs distances de détection :

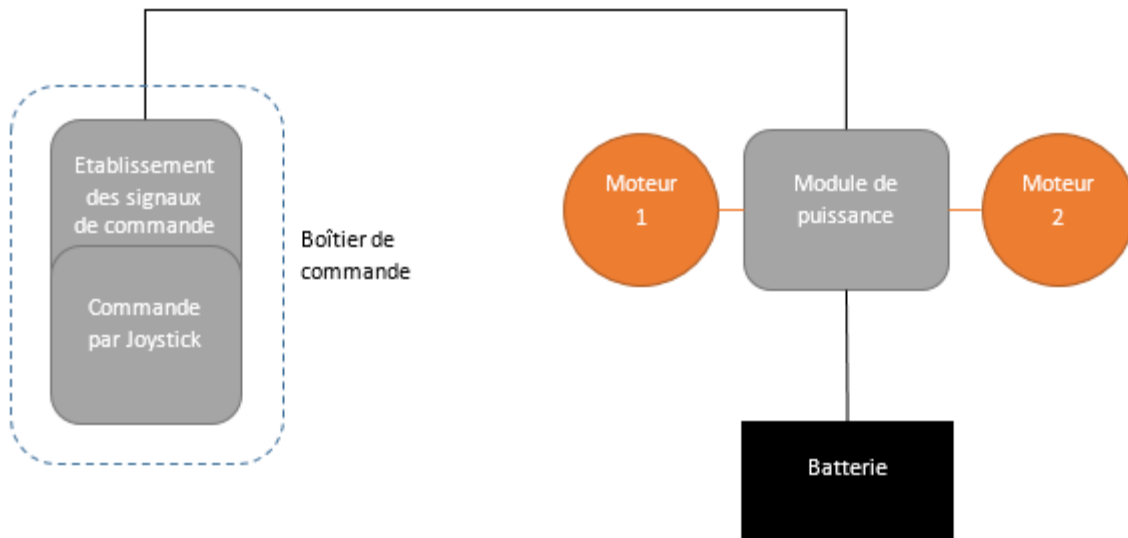
- 3 distances pour la face avant permettant de détecter un obstacle :
 - Eloigné
 - Proche
 - Trop proche
- 3 distances pour les faces droite et gauche similaires.

Lorsque le capteur détecte un obstacle à une distance éloignée, la vitesse diminue faiblement. Si l'obstacle se rapproche, la vitesse diminue de nouveau jusqu'à ce qu'il soit considéré trop proche. Dans ce cas, la vitesse devient nulle jusqu'à ce que l'obstacle (s'il est mobile) s'éloigne du capteur et le cycle normal de détection reprend son cours.

Ce module Bioloid permettra aux membres du projet de visualiser l'objectif final sur un modèle réduit. Il est à noter que nous avons défini 3 distances de détection et 3 vitesses de déplacement. Dans le module final, la vitesse devra évoluer de façon proportionnelle à la distance de l'obstacle détecté afin de rendre la limitation de vitesse la plus fluide possible et ainsi éviter les phénomènes d'inertie trop importants.

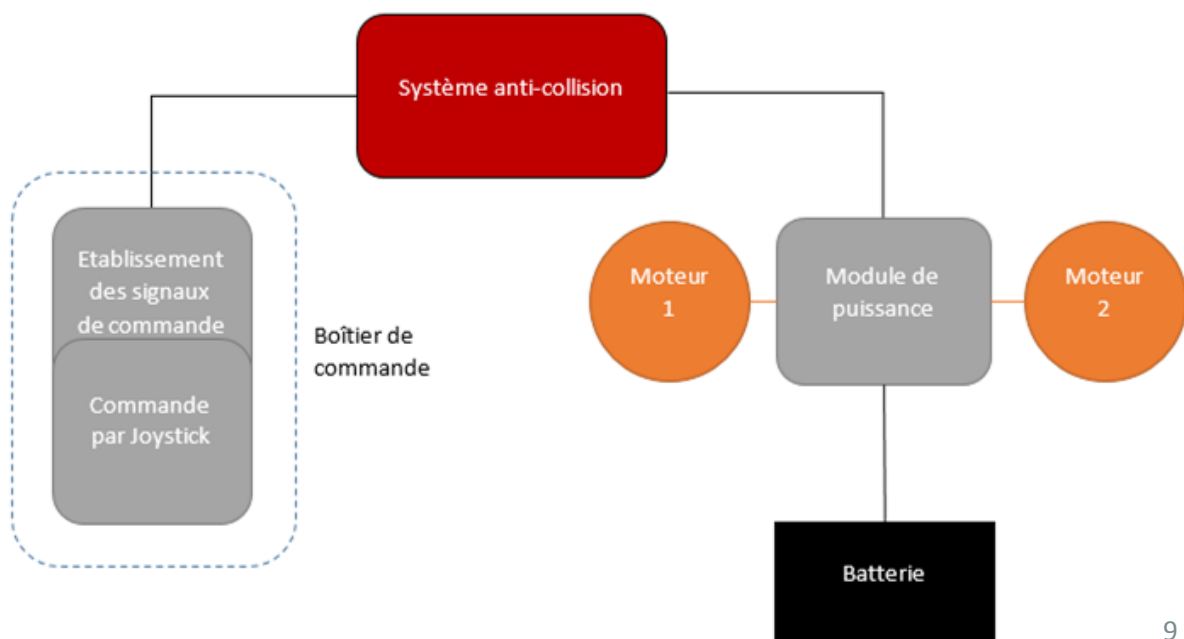
b. Etude du fauteuil roulant électrique

Une fois le fauteuil roulant électrique reçu (4 décembre 2014), nous avons pu obtenir certaines informations de Martin NICOLAS, ergothérapeute à la MAS La Gerlotte. Le fauteuil en question possède la structure suivante (légèrement différente de la structure de départ) :



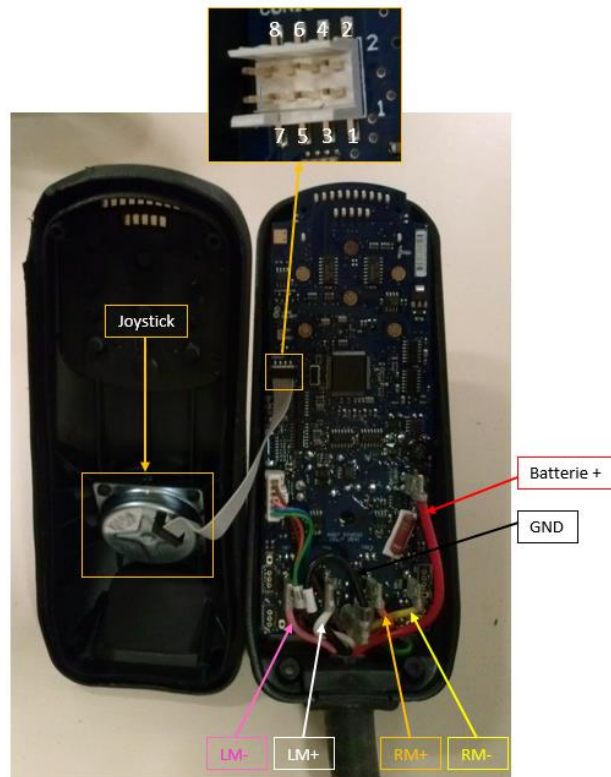
La commande du fauteuil contient alors le Joystick ainsi qu’une carte électronique permettant l’établissement des signaux de commande.

L’idée de départ est la suivante : notre module anti-collision devra s’intégrer entre la commande et le module de puissance afin de récupérer les signaux de commande. Une fois récupérés, nous devons les modifier en fonction des informations capteur reçues afin de diminuer la vitesse des moteurs ou non :



c. Etude de la commande du FRE

Afin de valider cette hypothèse de départ, nous avons effectué une étude de la commande du FRE. N'ayant pas accès à une quelconque datasheet, nous avons dû étudier l'ensemble de la carte électronique visuellement et en relevant des tensions aux points supposés importants. La carte électronique de la commande du fauteuil se présente comme ceci :

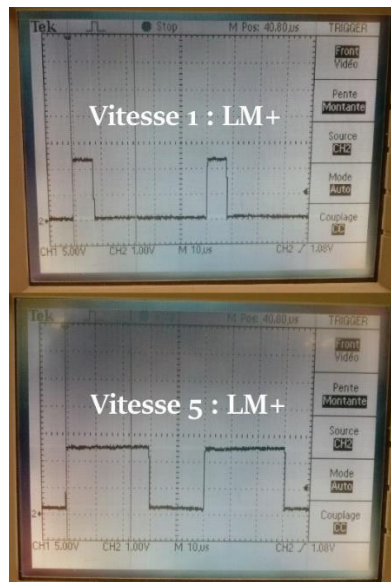


On y retrouve le joystick relié par une nappe de 8 fils à 2 connecteurs HE8. Le joystick et la carte communiquent alors un ensemble de 8 signaux qui sont ensuite traités par la carte électronique afin de générer des signaux de commande pour les moteurs. Le détail des signaux sera exposé plus tard dans la partie Hardware.

Visuellement, nous avons pu relever différents éléments importants. On remarque l'existence de 4 signaux qui sont transmis aux moteurs :

- RM + / RM -
- LM + / LM -

On interprète alors ces dénominations comme étant les abréviations de Right Motor et Left Motor. Afin de vérifier cela, on relève à l'oscilloscope les valeurs de ces signaux, par exemple pour LM + :



En allant en marche avant, on observe alors un signal PWM parcourant LM + qui devient plus important selon la vitesse sélectionnée sur le fauteuil (1 à 5). En effectuant ce même relevé sur chaque borne, on affirme alors que ces 4 signaux permettent de contrôler les moteurs du fauteuil.

Une fois cette étude globale de la commande du système effectuée, il a fallu réfléchir à une architecture pour notre module. Pour cela, nous devons répondre à plusieurs questions :

- Quels signaux récupérer ?
- Comment les modifier ?
- Comment les envoyer aux moteurs ?
- Comment récupérer les informations des capteurs ?
- Comment alimenter le module final ?

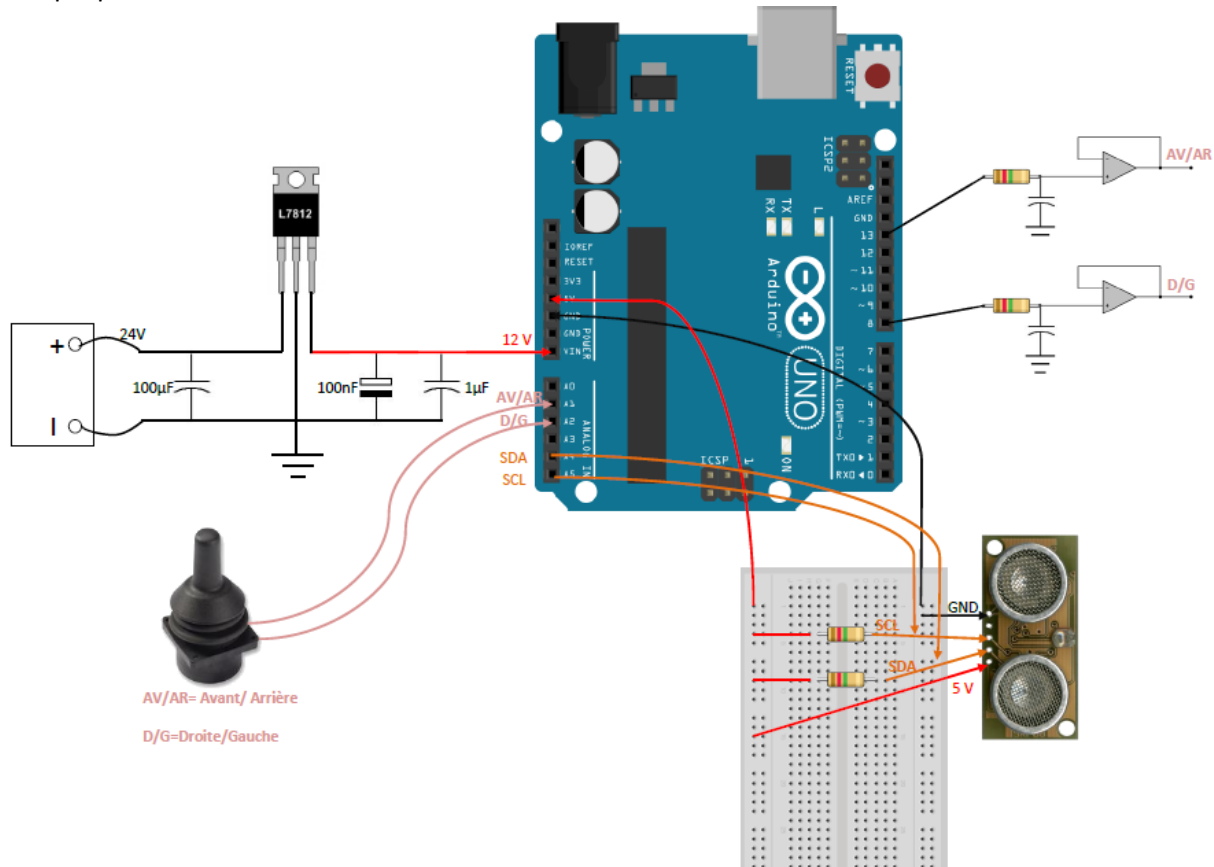
Nous allons alors présenter notre module d'une façon globale puis nous nous attarderons sur tous les points nécessaires à la bonne compréhension de son développement dans la partie Hardware, puis nous parlerons du fonctionnement en liaison avec un réseau de capteurs dans la partie Software.

V. Partie Hardware

Afin de faire fonctionner notre module nous devons :

- Alimenter l'Arduino
- Récupérer les informations de commande des moteurs
- Transmettre ces informations à la carte de commande du FRE
- Alimenter les capteurs, ainsi que récupérer les informations qu'ils nous renvoient

Afin d'illustrer nos propos nous avons effectué le schéma suivant, par la suite nous allons en détailler chaque partie.



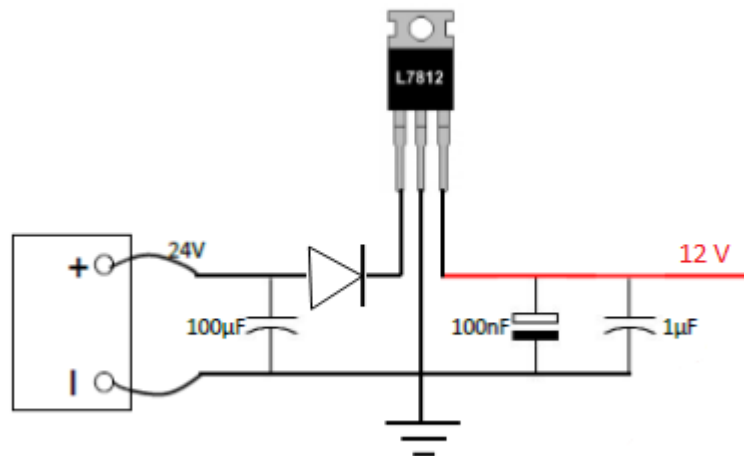
a. Alimentation de l'Arduino

Afin d'alimenter l'Arduino nous avons étudié plusieurs possibilités :

- Alimentation à partir des batteries du FRE
- Alimentation par batterie externe

Afin d'avoir une autonomie plus élevée de notre module nous avons décidé d'alimenter l'Arduino directement à partir des batteries du fauteuil. La commande du FRE devant elle aussi être alimentée nous avons récupéré la tension à ce niveau, cependant la tension prise à ce point est de 24 V et la tension d'alimentation de l'Arduino est comprise entre 7 V et 12 V. Nous avons donc opté pour un composant permettant d'abaisser la tension de 12 V à 24 V.

Pour ce faire nous avons utilisé le 7812 et réalisé le schéma suivant :



Nous allons alors chercher les 24 V de la batterie en passant par la commande du FRE. La diode nous permet de protéger le 7812 en cas de mauvais branchement, le condensateur d'entrée est un condensateur de filtrage. La présence des condensateurs en sortie du composant nous permet d'éviter les oscillations et donc d'avoir un résultat stable et fixe.

Cette tension est ensuite envoyée directement dans l'Arduino grâce à un connecteur Jack. L'Arduino possédant un régulateur interne cela nous permet de créer une tension de 5V nécessaire à l'alimentation des capteurs.

b. Récupération des informations moteurs

i. Choix de récupération des informations

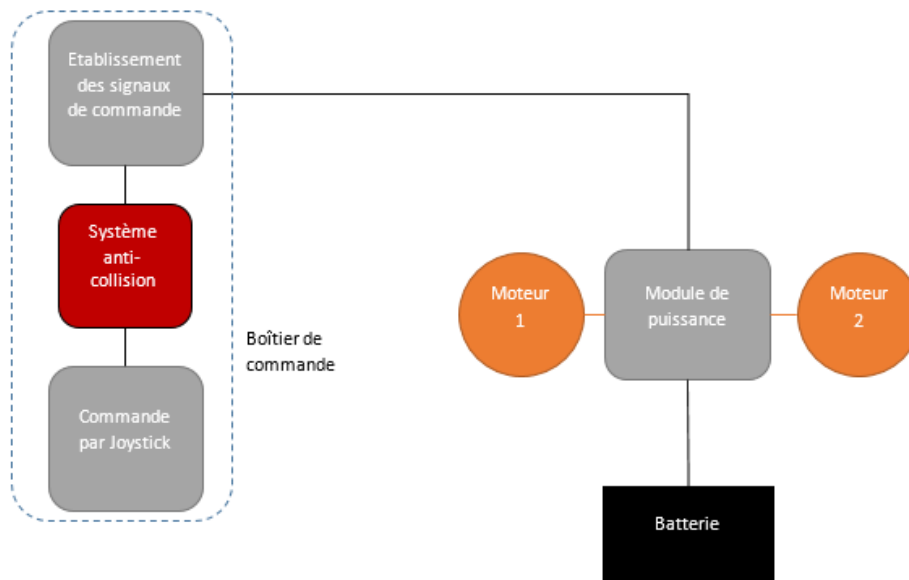
Afin de pouvoir contrôler l'arrêt du fauteuil, il nous faut récupérer les signaux de commande. Pour cela deux propositions s'offrent à nous:

- Récupérer les commandes moteurs directement à partir de RM +/- LM+-
- Récupérer les commandes provenant du joystick

La solution rejetée à été de récupérer les commandes moteurs à partir de RM +/- LM+- , d'une part car cela présentait l'inconvénient de devoir venir souder au niveau de la carte de commande du FRE ce qui aurait altéré la garantie du fauteuil, et d'autre part il aurait fallu réaliser un circuit d'adaptation de puissance qui est plus difficile à mettre en œuvre.

La solution choisie a donc été de récupérer les signaux provenant du joystick, présentant l'avantage de ne pas être invasif et donc de répondre parfaitement au cahier des charges.

Ce qui signifie qu'au niveau de l'architecture du fauteuil notre module s'implantera de la façon suivante :



Il nous faut maintenant étudier les signaux contenus dans le joystick afin de les interpréter et de les modifier.

ii. Etude du joystick

Il nous faut maintenant étudier le joystick afin d'identifier à quels endroits nous devons venir récupérer les informations.

N'ayant pas la documentation technique du joystick nous avons étudié les signaux (en volt) provenant de celui-ci, nous avons obtenu les résultats suivants :

Patte	Rien	Droite	Gauche	Avant	Arrière
1	2,5	2,5 jusqu'à 3,9	2,5 vers 1,1	2,5	2,5
2	0	0	0	0	0
3	2,5	2,5	2,5	2,5 jusqu'à 3,9	2,5 vers 1,1
4	2,5	2,5	2,5	2,5	2,5
5	0	0	0	0	0
6	2,5	2,5	2,5	2,5 jusqu'à 3,9	2,5 vers 1,1
7	5	5	5	5	5
8	2,5	2,5 jusqu'à 3,9	2,5 vers 1,1	2,5	2,5

A partir de ces relevés on observe que quatre pattes ne varient pas en fonction de la position du joystick :

- La patte 2 varie autour de 0 comme un élément non connecté, nous en avons déduit que cette patte été inutilisée
- La patte 3 à une valeur fixe à 0, nous en avons déduit qu'il s'agissait de la masse
- La patte 4 est une patte de référence ayant pour valeur 2,5 V
- La patte 5 à une valeur de 5 V constante, nous en avons déduit qu'il s'agissait de l'alimentation du joystick.

D'autre part on observe que lorsque le joystick est dans la position milieu, c'est-à-dire que le fauteuil ne se déplace pas, la valeur présente sur les pattes 1, 3, 4, 6 et 8 est de 2,5 volts. Cette information est cruciale puisque plus tard nous avons découvert que lors du démarrage si ces pattes ne possèdent pas 2,5V le fauteuil ne s'allume pas correctement et il est impossible de contrôler le FRE.

De plus nous avons déterminé que les pattes 1 et 8 sont reliées et permettent de contrôler le fauteuil pour que celui-ci se déplace vers la droite ou la gauche.

- Si la valeur est comprise entre 2,5 et 3,9 V alors le fauteuil se déplacera vers la droite. A 3,9 V le fauteuil ira le plus vite possible dans la direction désirée en fonction de la vitesse choisie.
- Si la valeur est comprise entre 2,5 et 1,1 V alors le fauteuil se déplacera vers la gauche. A 1,1 V le fauteuil ira le plus vite possible dans la direction désirée en fonction de la vitesse choisie.

Les pattes 3 et 6 sont reliées et permettent de contrôler le fauteuil pour que celui-ci se déplace vers l'avant ou vers l'arrière.

- Si la valeur est comprise entre 2,5 et 3,9 V alors le fauteuil se déplacera vers l'avant. A 3,9 V le fauteuil ira le plus vite possible dans la direction désirée en fonction de la vitesse choisie.
- Si la valeur est comprise entre 2,5 et 1,1 V alors le fauteuil se déplacera vers l'arrière. A 1,1 V le fauteuil ira le plus vite possible dans la direction désirée en fonction de la vitesse choisie.

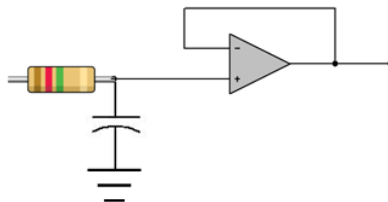
Afin de ne pas altérer la garantie du fauteuil nous devons changer la nappe électronique reliant le joystick à la carte de commande afin de la faire passer par notre carte électronique.

Les informations de directions seront donc interceptées au niveau des pattes 1-8 et 3-6 pour être redirigées vers l'Arduino qui interprètera les données et les modifiera en fonction des informations capteurs.

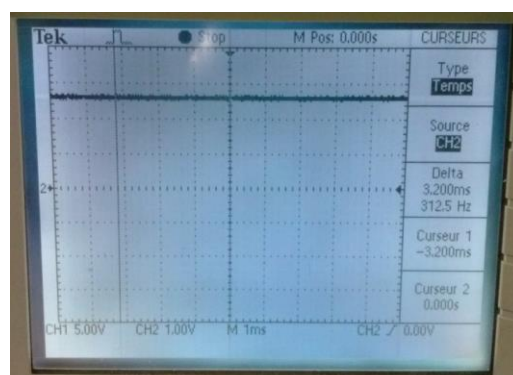
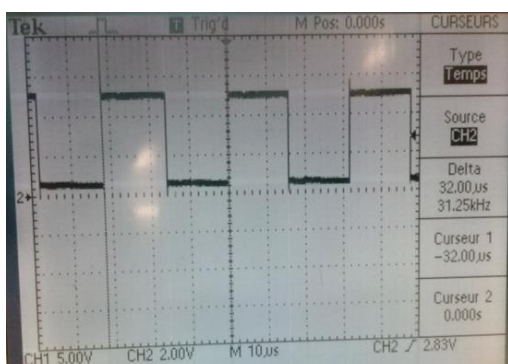
c. Récupération des signaux PWM

N'ayant pas de sortie analogique sur l'Arduino il est nécessaire de créer un signal continu à partir d'un signal PWM. Dans un premier temps nous devons interpréter les signaux analogiques trouvés puis les convertir en signaux PWM et à la sortie de l'Arduino nous allons réaliser un filtre RC suivi d'un AOP qui permettra d'abaisser l'impédance si besoin.

On réalise donc le schéma suivant :



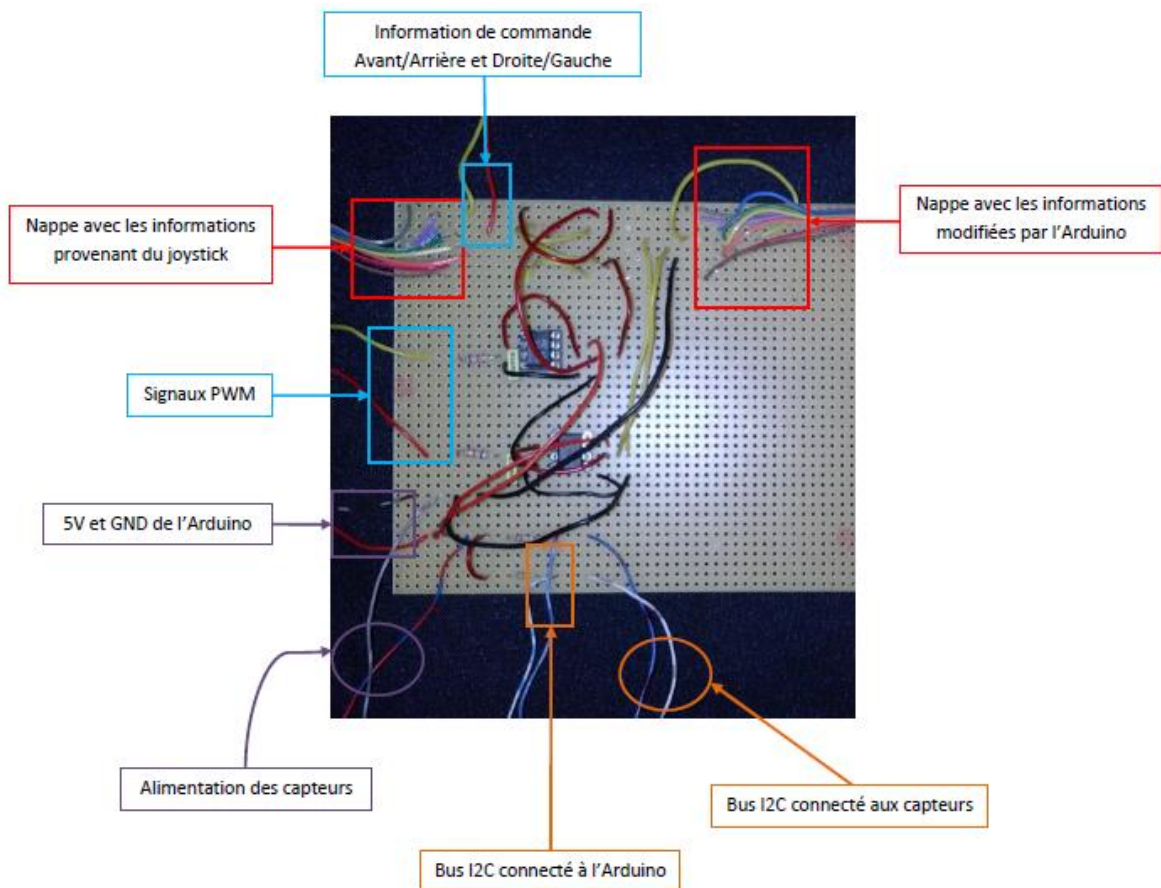
Les résultats obtenus sont les suivants :



On observe alors que pour un signal PWM avec un rapport cyclique de 50% à la sortie du filtre RC on obtient une tension constante de 2,5 V. L'état bas du signal PWM est de 0v, et l'état haut est de 5v. Cependant la valeur de l'état haut peut varier en fonction de l'alimentation de l'Arduino, ce qui impactera forcément la valeur à la sortie du filtre RC.

Une fois traités, les signaux seront redirigés vers la carte électronique à partir de laquelle une nappe les reliera jusqu'à la carte électronique du boîtier de commande du FRE.

La carte électronique réalisée est donc la suivante :

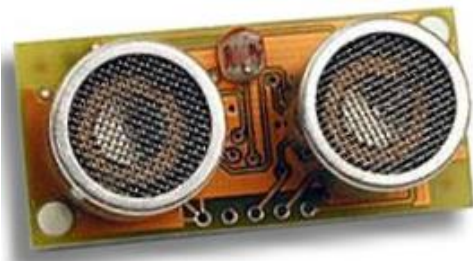


VI. Partie Software

Dans la partie Software, l'objectif principal était de développer un algorithme de détection d'obstacles ayant comme entrées des informations de capteurs ultrasons et pour sorties des signaux de commande du FRE. Pour cela, après les différentes études du système et l'acquisition du matériel, il a fallu :

- Prendre en main les capteurs à ultrasons SRF08
 - Bus I²C
 - Configuration et récupération des valeurs
- Prendre en main les fonctions nécessaires de l'Arduino UNO
 - PWM
 - Analogique
- Développer un algorithme de détection d'obstacles adapté aux besoins
 - Pas de situation de blocage
 - Décélération proportionnelle

a. Capteurs à ultrasons SRF08

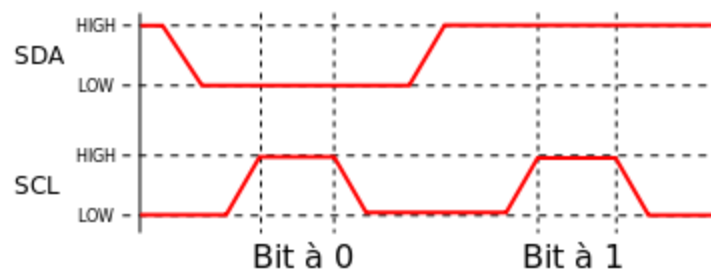


Les capteurs SRF08 sont des capteurs à ultrasons communiquant via le bus I²C et compatibles avec la plupart des microcontrôleurs. Leur configuration de base leur permet de mesurer des objets jusqu'à 6 mètres de portée avec une distance minimale de 4 cm.

Ces performances peuvent être poussées jusqu'à une détection d'environ 11 mètres, mais l'environnement observé doit s'adapter afin de ne pas faire réfléchir les ondes ultrasons et fausser les données.

Avant de pouvoir communiquer avec eux, il est essentiel de comprendre le fonctionnement du bus I²C. Ce bus contient 2 lignes :

- Un signal de données bidirectionnelles SDA (Signal Data Line)
- Un signal d'horloge de synchronisation bidirectionnelle SCL (Signal Clock Line)



L'envoi des données (écriture dans les registres, demande de récupération de valeurs etc...) se fait donc sur la ligne SDA, tandis que la ligne SCL permet de synchroniser le tout.

Un SRF08 possède alors 36 registres (de 0 à 35) et nous allons surtout nous intéresser aux 3 premiers : les registres 0, 1 et 2.

Location	Read	Write
0	Software Revision	Command Register
1	Light Sensor	Max Gain Register (default 31)
2	1st Echo High Byte	Range Register (default 255)

Le registre 0 permet en lecture d'obtenir la version du software utilisée par le capteur (6 dans notre cas) et il permet en écriture de venir écrire dans le registre de commande, registre clé qui permet de configurer de nombreuses options sur le capteur comme son adresse I²C par exemple.

Le registre 1 permet en lecture de récupérer la valeur du capteur de luminosité (non utilisé) et en écriture de venir modifier le registre de gain du capteur. Le gain du capteur permet de contrôler la vitesse à laquelle il va envoyer et lire les ondes ultrasons. Cela permet entre autre de diminuer la latence entre 2 lectures consécutives. Le point négatif est que lorsque la détection d'obstacle s'effectue sur plusieurs mètres, on risque de détecter des objets proches alors qu'il s'agit en fait d'un retour des ondes précédentes. Il est donc déconseillé d'augmenter le gain lorsque la détection voulue agit sur plus d'un mètre.

Le registre 2 permet en lecture de récupérer le bit haut du premier echo d'onde envoyé (bit utilisé pour ensuite être traduit en distance en cm ou inch). Il permet en écriture de venir modifier le registre de range. Le range register (registre de portée) permet donc de venir définir la portée maximale du capteur. On décide alors jusqu'à combien de cm le capteur va détecter un objet, avec une limite interne au capteur fixée d'environ 11 mètres.

En résumé, on écrira nos commandes dans le registre 0, on modifiera le gain dans le registre 1 et on modifiera la portée dans le registre 2.

De façon plus générale, pour utiliser les capteurs dans leur configuration par défaut, Arduino a développé une librairie SonarSRF08, dérivée de la librairie de base Wire permettant d'avoir accès à quelques fonctions de base (écrire une commande, lire la distance renvoyée par le capteur...). De cette librairie, nous avons pu apprendre à manipuler un capteur et nous avons pu ensuite créer nos fonctions permettant de changer sa configuration.

La contrainte principale pour un capteur SRF08, dans notre cas, était de pouvoir mesurer un obstacle à une distance d'environ 2 mètres (pour permettre une décélération progressive) tout en gardant une bonne précision et une vitesse convenable. Pour cela, nous devons alors modifier la portée maximale qui se calcule comme suit :

$$Portée_max = Range_{Register} * 43mm + 43mm$$

Pour une portée maximale de 2 mètres, il est alors nécessaire de placer 46 en décimal dans le Range_register : ce qui équivaut à 0x2E en hexadécimal.

Concernant le gain, après différents tests, il s'est avéré que la valeur par défaut était satisfaisante. Une valeur inférieure nous empêchait d'atteindre les 2 mètres de portée, et une valeur trop

supérieure nous apportait trop de fausses valeurs (redondance d'ondes). On conserve alors le gain par défaut de 0x1F (31).

Ces opérations de modification de la portée et du gain devaient donc être effectuées sur chaque capteur. Cependant, afin de les brancher en réseau, il est nécessaire d'attribuer une adresse I²C à chaque capteur. Pour cela, il suffit d'écrire une suite de commande dans chaque capteur branché individuellement dans un premier temps, avant de les relier ensemble.

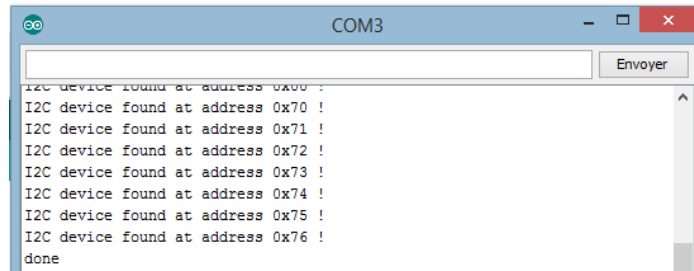
Pour donner une adresse I²C à un capteur, on envoie alors 3 commandes, suivies d'une commande contenant l'adresse désirée (allant de E0 à FE : 16 capteurs maximum en réseau). Exemple de changement d'adresse I²C :

- `sendCommand(0xA0);`
- `sendCommand(0xAA);`
- `sendCommand(0xA5);`
- `sendCommand(0xE0); // Adresse du capteur : 0xE0`

On change alors l'adresse de chaque capteur, puis on détermine leur place autour du fauteuil :



Une fois tous les capteurs reliés entre eux, on peut venir scanner le bus I²C afin de détecter les capteurs présents ainsi que leurs adresses respectives. On obtient le résultat suivant :



On détecte alors les capteurs avec des adresses différentes de celles initialisées. Cela s’explique simplement par un décalage binaire : lorsque l’arduino vient interroger le bus I²C, les adresses qu’il va lire seront décalées de 1 vers la droite (divisées par 2). Les adresses en 0xEX se retrouvent comme ceci :

$$0x70 \ll 1 = 0111\ 0000 \ll 1 \rightarrow 1110\ 0000 = 0xE0$$

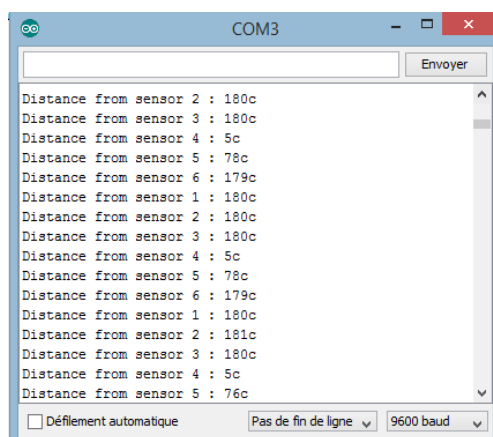
Enfin, la fonction principale des capteurs est de nous informer de la distance d’un potentiel obstacle. Afin de récupérer cette information, nous avons plusieurs choix d’unité :

- Centimètres
- Inches
- Millisecondes

Pour des raisons évidentes de compréhension, on choisira de récupérer les informations en centimètres. Pour cela, il faut configurer notre ranging en cm :

Command		Action
Decimal	Hex	
80	0x50	Ranging Mode - Result in inches
81	0x51	Ranging Mode - Result in centimeters
82	0x52	Ranging Mode - Result in micro-seconds

Il suffit alors d’envoyer une commande de 0x51 pour configurer le résultat renvoyé par le capteur SRF08 en cm. Après cela, pour récupérer le dit résultat, nous devons aller lire les locations 2 et 3 afin de récupérer 1st Echo High Byte et 1st Echo Low Byte qui contiennent à eux 2 la valeur du dernier ranging effectué (à savoir la dernière mesure effectuée). Une opération de concaténation est ensuite nécessaire pour regrouper ces 2 bits et obtenir la valeur voulue en cm :



Ici, on lit sur la liaison série les valeurs renvoyées par les 6 capteurs. Le c permet de préciser l’unité (cm).

Les 7 capteurs étant fonctionnels ensemble, nous allons pouvoir utiliser toutes ces données récupérées dans un algorithme de détection d'obstacle qui sera développé plus tard dans ce rapport.

b. Utilisation de l'Arduino pour la lecture et l'écriture de signaux

Comme présenté dans la partie Hardware, nous allons avoir besoin de lire des signaux analogiques et de créer des signaux PWM avec l'Arduino.

La lecture de signaux analogiques avec l'Arduino est une des fonctions les plus utilisées, cependant pour ce projet nous avons besoin d'une forte précision sur la lecture. Afin de vérifier l'exactitude de la lecture, nous avons effectué différents tests.

Notons tout d'abord la théorie qui nous dit que l'Arduino peut lire un signal de 0 à 5V avec pour 0V une valeur analogique de 0 et pour 5V une valeur analogique de 1023. Cependant, le maximum de 5V est en fait fixé selon l'alimentation fournie à l'Arduino. Selon l'alimentation choisie, le 5V fourni par l'Arduino peut varier de 4,3V à 6V. Dans notre cas, il nous délivre du 5,15V (alimentation par liaison USB d'un PC). Nous devons alors faire une table de correspondance afin de faire la bonne conversion de la valeur analogique vers la valeur en volts.

On obtient alors : 5,15V pour 1023 en lecture. On extrapole ceci afin d'obtenir notre loi :

$$Val_{Volts} = Val_{Analogique} * \frac{5.15}{1023}$$

De même pour les signaux PWM créés, nous devons nous assurer d'une bonne précision, nous avons alors effectué les mêmes tests. Nous créons un signal PWM en indiquant une valeur entre 0 et 255 à l'Arduino pour ressortir un signal avec un rapport cyclique compris entre 0 et 100%. Notre intérêt est de regarder la valeur moyenne de ce signal (qui sera la valeur du signal après le filtre RC).

Là encore, la théorie veut que 0 corresponde à un signal de 0V et que 255 à un signal de 5V. Cependant, on observe un signal d'une valeur moyenne de 5,24V pour un ordre de 255.

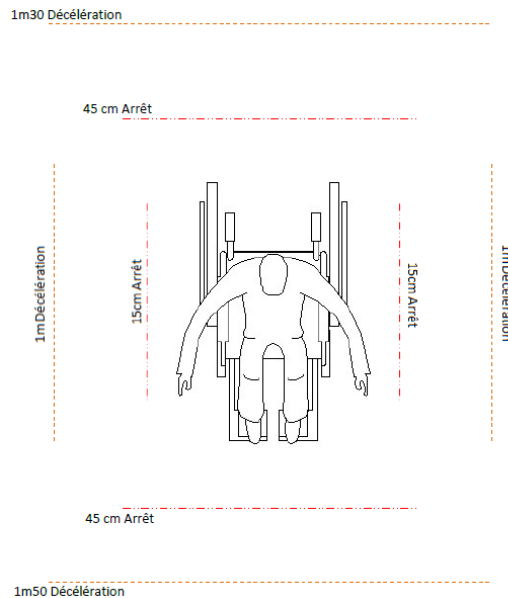
On extrapole alors pour obtenir notre loi PWM :

$$Val_{Volts} = Val_{PWM} * \frac{5.24}{255}$$

Grâce à ces deux analyses, nous pouvons fournir une précision des valeurs lues et générées satisfaisante. Le cas contraire, nous aurions pu injecter une tension de 2,6V et ainsi provoquer un déplacement du fauteuil alors que la consigne était de s'arrêter (2,5V).

c. Algorithme de détection d'obstacles

Afin de permettre au résident en fauteuil de conserver son autonomie tout en évitant les obstacles grâce à une décélération appropriée du fauteuil, nous avons dû réfléchir à un algorithme permettant d'éviter des situations dangereuses tout en conservant la liberté de mouvement. Avant tout, nous avons dû fixer des distances à partir desquelles on considérera un obstacle trop proche et où on décidera d'interdire le mouvement dans la direction de l'obstacle :



A partir de ces distances à ne pas franchir, nous avons alors développé un algorithme dont le pseudo-code est disponible en annexe. Les grandes lignes sont les suivantes :

- On scanne en continu la valeur des distances renvoyées par les capteurs
- On scanne en continu la position du joystick (valeurs analogiques droite/gauche et avant/arrière)
- Si un obstacle est détecté comme étant proche et que le joystick est dans la direction de l'obstacle : on limite la vitesse dans cette direction, jusqu'à l'arrêt lorsque l'obstacle est trop proche.
- S'il n'y a pas d'obstacle dans la direction du joystick, on recopie les entrées analogiques dans les sorties PWM : le fauteuil se déplace normalement.

Scan des distances en continu

Afin d'effectuer cette fonction, on utilise les capteurs SRF08 comme expliqué auparavant, on récupère alors chaque distance mesurée et, connaissant la position des capteurs via leurs adresses, on peut déterminer de quel côté se situe l'obstacle.

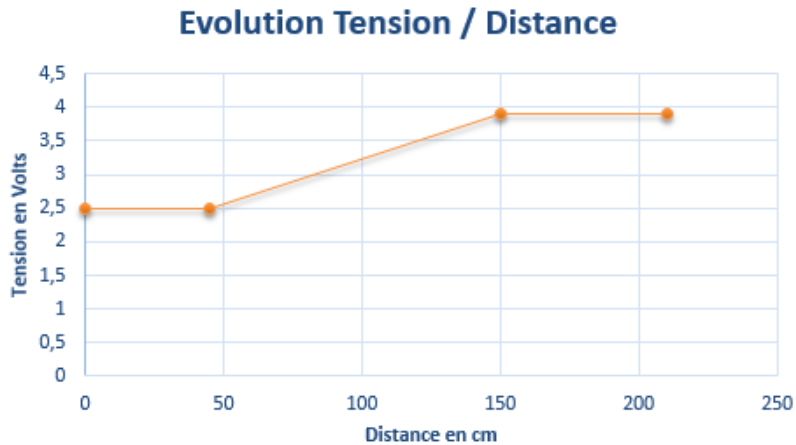
Scan de la position du joystick

La position du joystick se définit sur les 2 entrées analogiques A0 et A1, dans lesquelles on stocke les informations sur la position avant/arrière et droite/gauche du joystick. Pour les récupérer, il suffit alors de lire en continu la valeur de ces 2 entrées.

Obstacle detecté

Lorsqu'un obstacle est détecté dans la direction du joystick, plusieurs scénarios sont possibles, on étudiera à la fois la distance à laquelle se trouve l'obstacle ainsi que la puissance de la commande du joystick.

Si le fauteuil va vers l'avant (commande joystick vers l'avant) et qu'un obstacle se trouve à moins de 150cm, nous allons effectuer différents calculs. Avant tout, nous allons vérifier quel capteur possède la distance la plus faible et l'utiliser dans les futurs calculs. A partir de cette distance la plus faible, nous allons calculer la valeur PWM maximale autorisée pour la situation. Afin de calculer cette valeur, nous devons déterminer l'équation de la droite suivante :



Lorsque l'obstacle est très éloigné, on calculera une valeur de joystick en volts de 3,9V : on peut avoir notre commande au maximum sans danger. Lorsque l'obstacle sera trop près, on limitera la commande à 2,5V, qui provoquera l'arrêt du fauteuil dans la direction avant/arrière.

L'équation de la courbe est alors :

$$PWM_{volts} = 0.013 * Distance_{capteur} + 1.915$$

Cette valeur maximale autorisée pour le signal PWM à générer calculée, on la compare à la valeur actuelle de la commande en volts. Si le signal PWM s'avère supérieur à la valeur actuelle, on recopiera simplement la valeur analogique d'entrée : cela signifie que la commande du fauteuil actuelle est inférieure à la commande maximale en fonction de la distance de l'obstacle.

Cependant, si la valeur calculée en PWM est inférieure à celle de la commande actuelle, on écrit alors cette valeur afin de faire freiner le fauteuil.

On effectue ces calculs pour toutes les directions avec une courbe de décélération propre à chaque direction afin de contrôler la commande de notre fauteuil.

Cette façon de procéder nous permet aussi d'éviter des situations de blocage non-désirées. En effet, un obstacle détecté en avant ne nous empêche pas de nous déplacer sur les côtés ou en arrière, et vice versa. De cette façon, il sera toujours possible d'ajuster la position du fauteuil si on souhaite passer dans des endroits considérés étroits.

VII. Nos réalisations

Durant ce projet, nous avons alors réalisé la carte électronique décrite dans la partie Hardware et qui est actuellement fonctionnelle. Cela signifie que l'on réussit à récupérer les informations du joystick, à les modifier et à en envoyer de nouvelles à la carte de commande du FRE.

Les informations des signaux étant récupérées par l'Arduino, en liaison avec notre réseau de capteurs et son algorithme, nous arrivons à modifier les signaux de commande en fonction des informations capteurs reçues afin d'éviter les collisions et plus généralement pour améliorer le déplacement en réduisant la vitesse à l'approche d'obstacles.

Grâce à ce projet, nous avons pu mettre en œuvre une technique de Reverse Engineering afin de réussir à comprendre partiellement le fonctionnement d'une carte électronique de commande complexe ne possédant aucune datasheet accessible. A partir de la compréhension globale du système et de l'étude approfondie du fonctionnement du joystick, nous avons pu réussir à simuler un cycle d'initialisation pour ce dit joystick afin de pouvoir faire démarrer le fauteuil avec notre module.

Nous avons pu ensuite tester et ajuster les calculs de notre algorithme de détection d'obstacles afin de rendre le déplacement en FRE le plus sécurisé possible, sans pour autant perdre en autonomie et en évitant les situations de blocage.

VIII. Perspectives d'amélioration

Afin d'améliorer le module existant, nous avons pensé à différents possibilités. Tout d'abord, l'alimentation de l'Arduino se fait actuellement par une batterie externe délivrant du 5V. La solution du circuit composé du 7812 relié à la batterie du FRE étant jugé comme trop invasive, il serait intéressant de venir alimenter l'Arduino avec les 5V circulant de la carte de commande vers le joystick durant l'initialisation. Le seul problème à cela restera le temps de latence entre l'alimentation de l'Arduino, la récupération des signaux analogiques puis la génération des signaux PWM qui prend un certain temps. Le joystick pourra se mettre en défaut, manque de tensions satisfaisantes appliquées à ses différentes bornes lors de l'allumage.

Le temps de réponse actuel du système est assez élevé. Le temps entre la lecture, l'écriture de nouveaux signaux, le filtrage, l'arrivée dans la carte de commande puis finalement dans les moteurs est actuellement légèrement trop élevé pour un fonctionnement optimal. Il existe alors un travail d'optimisation à faire, à commencer par la réévaluation du filtre RC.

Au niveau mécanique, l'ensemble carte électronique + Arduino + capteurs est aujourd'hui très fragile, de simples fils sont soudés à des connecteurs et le système peut rencontrer des problèmes au moindre choc (court-circuit, fil coupé...). Afin d'améliorer cela, il serait intéressant d'avoir des nappes de fils de la taille adéquate, des connecteurs résistants sous les capteurs ainsi que des protections pour les capteurs et pour la carte.

Au niveau fonctionnel, afin d'améliorer considérablement la fluidité du système, il serait intéressant d'avoir accès à la vitesse actuelle du fauteuil. En effet, le FRE possède 5 modes de vitesse, allant de 1 à 5. Si nous avions cette information, nous pourrions affiner considérablement l'algorithme de

détection d'obstacles. Cela nous permettrait de diminuer les distances d'arrêt en vitesse 1 et de les augmenter en vitesse 5 afin d'éviter un quelconque danger tout en augmentant fortement la permissivité du système en vitesse faible. Cela signifierait que le passage des portes, même étroites, serait une tâche aisée en vitesse 1 ou 2.

Actuellement, nous avons fixé nos distances d'arrêt et de décélération (et les courbes associées) afin d'avoir un système à la fois permissif et sécurisé dans toutes les vitesses. Cependant, il est impossible d'avoir la précision désirée en vitesse faible ainsi que la sécurité désirée en vitesse élevée. On observe alors actuellement des scénarios de blocage partiels qui pourraient être inexistantes avec la récupération de l'information de vitesse, malheureusement uniquement disponible dans la carte de commande, ce qui pose encore le problème de l'invasion du module d'aide au déplacement.

Conclusion

Grâce à ce projet nous avons pu dans un premier temps développer un cahier des charges technique dans le cadre d'un projet technique concret. Ce document devait être validé par l'ensemble des membres du projet et nous avons dû apprendre à écouter les besoins et les différentes contraintes de chacun.

Après avoir validé le cahier des charges, nous avons proposé plusieurs solutions techniques et nous avons choisi, avec les membres de la MAS Le Hameau, la solution qui paraissait la plus intéressante et qui respectait différentes contraintes.

De plus, afin de rendre le projet plus visuel, en accord avec M. CONRARD, nous avons développé un modèle réduit permettant de représenter les déplacements du FRE lorsqu'il sera équipé du module anti-collision, le tout grâce à du matériel Bioloid. Cela nous a permis de réfléchir à l'algorithme de commande de la vitesse en ayant un modèle de test afin de se rendre compte de toutes les contraintes présentes lors des déplacements d'un FRE.

Ce projet nous a aussi permis d'étudier et de comprendre un système déjà existant, et de mettre en place une technique de reverse engineering. Après cette phase de compréhension nous avons pu choisir à quels endroits récupérer les informations de commande du fauteuil afin de les modifier.

Nous avons ensuite développé une carte électronique afin de recevoir les signaux du joystick, d'alimenter et de recevoir les informations capteurs pour ensuite, grâce à l'Arduino modifier la commande des moteurs.

En parallèle nous avons développé un algorithme qui en lisant les valeurs retournés par les capteurs et les informations du joystick, va adapter la vitesse du fauteuil. Nous avons alors ensuite pu tester et ajuster cet algorithme afin de rendre les déplacements en fauteuil roulant le plus sécurisé possible, sans pour autant créer des situations de blocage.

Ce projet nous a permis de mener une mission à bien, depuis la compréhension de la demande à la création d'un prototype, en passant par différentes étapes importantes comme l'élaboration d'un cahier des charges complet, la compréhension d'un système existant, et la réalisation d'un module répondant au cahier des charges.

Enfin, cela nous a aussi permis d'avoir la possibilité de travailler pour le monde du polyhandicap et de mettre à profit nos connaissances pour une cause humaine.

ANNEXE 1 :

FAUTEUIL ROULANT INTELLIGENT

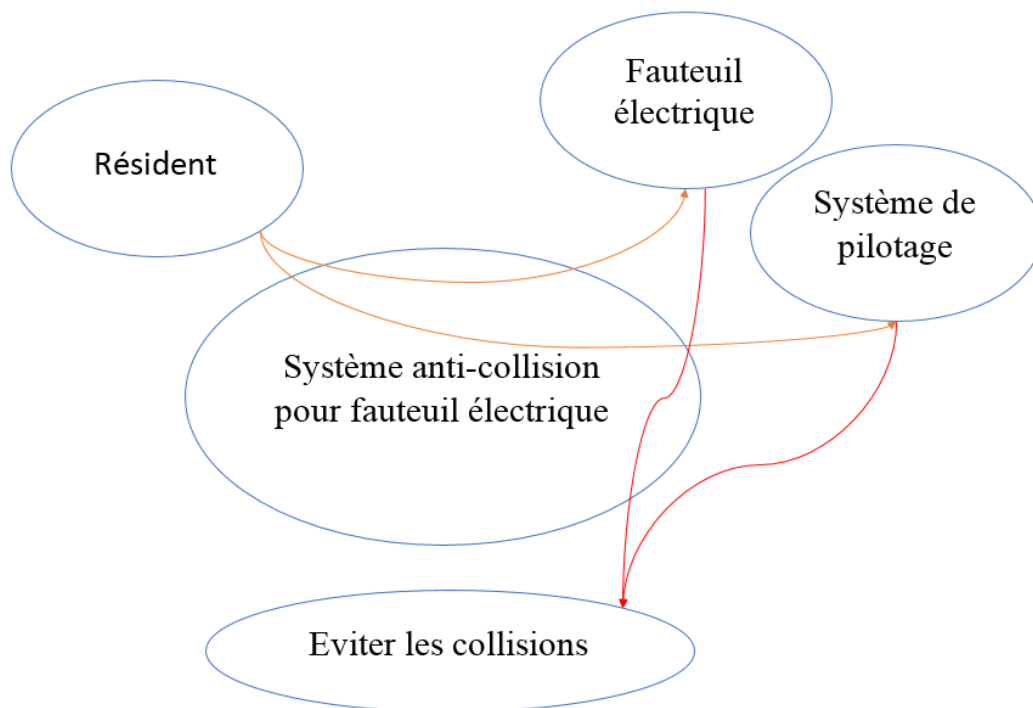
CAHIER DES CHARGES

L'association "La vie autrement", membre du groupement GAPAS, s'est fixée pour objectif de permettre à des personnes associant déficiences physiques et mentales de trouver un espace adapté où elles peuvent exprimer et exploiter tous leurs potentiels.

Afin d'atteindre cet objectif, l'association a ouvert plusieurs maisons d'accueil spécialisées et notamment la maison du Hameau située à Hantay. Cette maison d'accueil regroupe 40 adultes polyhandicapés pour la plupart entièrement dépendants pour tous les actes de la vie quotidienne. La moitié de ces résidents se déplacent à l'aide d'un fauteuil roulant électrique dont la commande peut varier (joystick, boutons...). Cette utilisation des fauteuils roulants dans un espace tel que la maison du Hameau a soulevé des difficultés (collisions entre les résidents, avec les murs, portes etc...).

Pour pallier ces problèmes, l'idée d'un projet en collaboration avec l'école d'ingénieur Polytech'Lille est née, le but étant de développer un système permettant d'améliorer la vie en maison d'accueil pour tous les résidents en évitant les collisions.

Description du besoin :



Le système anti-collision permettra au résident en fauteuil électrique, en liaison avec son système de pilotage, d'éviter les collisions.

Objectifs :

Développer un système permettant aux résidents en fauteuil électrique :

- D'éviter les collisions avec l'environnement des maisons d'accueil :
 - Avec les murs, radiateurs...
 - Avec les autres fauteuils et le personnel
- D'améliorer leurs déplacements :
 - Lors de passages étroits
 - Lors de la rencontre d'autres résidents ou du personnel

Fonctions principales :

La fonction principale du système est :

- Empêcher les collisions

Les fonctions permettant d'atteindre cet objectif sont donc les suivantes :

- F0 : Détecter les obstacles

Le système devra être capable de détecter des obstacles statiques et mobiles. On définit les obstacles comme tout objet se trouvant à une hauteur minimale de 30 cm et une hauteur maximale de 40 cm. L'obstacle doit posséder une surface pleine entre ces 2 hauteurs.

- F1 : Limiter la vitesse du fauteuil

Le système, relié au module de commande du fauteuil, devra être capable de limiter la vitesse de ce dernier. Il devra donc pouvoir accéder aux signaux de commande du fauteuil et les modifier afin d'empêcher les collisions.

- SF1 : Arrêter le fauteuil

Le système, relié au module de commande du fauteuil, devra être capable d'empêcher le mouvement dans la direction de l'obstacle lorsque ce dernier sera trop rapproché. On considérera un obstacle trop proche lorsqu'il sera à une distance inférieure à 4cm sur les côtés du fauteuil, et 20cm dans la trajectoire du fauteuil.

Contraintes :

Ces objectifs devront être réalisés en tenant compte des contraintes suivantes :

- C1 : Ne pas rompre la garantie constructeur du fauteuil électrique.
- C2 : Le système doit permettre au fauteuil de circuler en zones étroites (couloir, passage de porte). On définit une zone étroite comme toute zone dont la largeur est inférieure à : largeur du fauteuil + 40 cm.
- C3 : Le système doit être utilisable en extérieur. Il présentera un indice de protection de 44. C'est-à-dire qu'il sera protégé contre les corps solides supérieurs à 1mm et contre les projections d'eau dans toutes les directions.
- C4 : Le système développé doit s'intégrer harmonieusement avec le fauteuil.
- C5 : Le système doit pouvoir être alimenté grâce aux batteries du fauteuil.
- C6 : L'intégration du module (fixation mécanique) ne doit pas altérer la garantie du fauteuil.

ANNEXE 2 : Algorithmme Pseudo-Code

Distances :

Avant Gauche : E0

Avant Droite : E2

Droit Avant : E4

Droit Arrière : E8

Arrière : E6

Gauche Arrière : EA

Gauche Avant : EC

Avant Détection	Av_Det
Avant Décélération	Av_Slow
Avant Arrêt	Av_Stop
Arrière Détection	Ar_Det
Arrière Décélération	Ar_Slow
Arrière Arrêt	Ar_Stop
Côté Détection	Co_Det
Côté Décélération	Co_Slow
Côté Arrêt	Co_Stop
Port AN Avant/Arrière	AN1
Port AN Droite/Gauche	AN0
Port PWM Avant/Arrière	PWM6
Port PWM Droite/Gauche	PWM5

Scanner en continu les valeurs des capteurs :

```
// EN AVANT
```

```
Si (Joystick en avant && (Distance_E0 || Distance_E2 < Av_Slow))
```

```
{
    Lire AN1;
    Si(Distance_E0 < Distance_E2)
    {
        PWM6 = 0.013*Distance_E0+1,915;
        Si(PWM6 < AN1) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
        {
            Ecriture PWM6;
        }
        Sinon
        {
            Réécriture de AN1 avec PWM6;
        }
    }
    Sinon Si(Distance_E2 < Distance_E0)
    {
        PWM6 = 0.013*Distance_E2+1,915;
        Si(PWM6 < AN1) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
        {
            Ecriture PWM6;
        }
        Sinon
        {
            Réécriture de AN1 avec PWM6;
        }
    }
}
```

```
// EN ARRIERE
```

```
Sinon Si (Joystick en arrière && (Distance_E6 < Ar_Slow))
```

```
{
```

```
    Lire AN1;
```

```
    PWM6 = -0.015*Distance_E6+3.175;
```

```
    Si(PWM6>AN1) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
```

```
    {
```

```
        Ecriture PWM6;
```

```
    }
```

```
    Sinon
```

```
    {
```

```
        Réécriture de AN1 avec PWM6;
```

```
    }
```

```
}
```

```
Sinon :
```

```
Réécriture de AN1 avec PWM6.
```



```
// A DROITE
```

```
Si (Joystick à Droite && (Distance_E4 || Distance_E8 < Co_Slow))
```

```
{
```

```
  Lire AN0;
```

```
  Si(Distance_E4 <Distance_E8)
```

```
  {
```

```
    PWM5 = 0.016*Distance_E4+2.26;
```

```
    Si(PWM5<AN0) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
```

```
    {
```

```
      Ecriture PWM5;
```

```
    }
```

```
    Sinon
```

```
    {
```

```
      Réécriture de AN0 avec PWM5;
```

```
    }
```

```
  }
```

```
  Sinon Si(Distance_E8 < Distance_E4)
```

```
  {
```

```
    PWM5 = 0.016*Distance_E8+2.26;
```

```
    Si(PWM5<AN0) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
```

```
    {
```

```
      Ecriture PWM5;
```

```
    }
```

```
    Sinon
```

```
    {
```

```
      Réécriture de AN0 avec PWM5;
```

```
    }
```

```
  }
```

```
}
```

```
// A GAUCHE
```

```
Sinon Si (Joystick à Gauche && (Distance_EA || Distance_EC < Co_Slow))
```

```
{
```

```
  Lire AN0;
```

```
  Si(Distance_EA < Distance_EC)
```

```
  {
```

```
    PWM5 = -0.016*Distance_EA+2.74;
```

```
    Si(PWM5 < AN0) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
```

```
    {
```

```
      Ecriture PWM5;
```

```
    }
```

```
    Sinon
```

```
    {
```

```
      Réécriture de AN0 avec PWM5;
```

```
    }
```

```
  }
```

```
  Sinon Si(Distance_EC < Distance_EA)
```

```
  {
```

```
    PWM5 = -0.016*Distance_EC+2.74;
```

```
    Si(PWM5 < AN0) //Si la tension à écrire est plus petite que celle mesurée AKA si on veut diminuer la vitesse du fauteuil
```

```
    {
```

```
      Ecriture PWM5;
```

```
    }
```

```
    Sinon
```

```
    {
```

```
      Réécriture de AN0 avec PWM5;
```

```
    }
```

```
  }
```

```
}
```

Sinon :

Réécriture de AN0 avec PWM5