



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Motorisation et pilotage d'un fauteuil pour personne handicapée



RAPPORT DE PROJET TUTORE

2018/2019

*Martin Samuel
&
Lallemand Victor*

Institut Universitaire de Technologie de Belfort-
Montbéliard

Licence Professionnelle VEGA

Patrick HIEBEL– Professeur référent



Remerciements

Le projet présenté dans ce rapport, c'était développé dans le cadre de la Licence Pro VEGA, au sein de l'IUT de Belfort Montbéliard.

Tout d'abord nous devons remercier à M.Patrick HIEBEL, tuteur de notre projet, qui a nous aidé tout au long de notre projet.

Nous tenons, également, remercier à tout l'équipe pédagogique pour l'aide quelle nous a apportée tous le long de l'année, à travers les cours et les travaux pratiques.



Sommaire

- Introduction
- Matériel
- Schéma de fonctionnement
- Composants
 - PIC16F877
 - PIC16F876
 - Driver moteur DRI0042
 - Joystick
- Le protocole I2C
- Liaison UART
- Loi de commande entre le joystick et les moteurs
- Frein à pertes de tension
- Interface Homme Machine et Instrumentation
- Signalisation
- Conclusion
- Annexes



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Introduction

Le projet tutoré a pour but de nous rendre autonomes et de mettre en pratique les connaissances apprises lors de notre année de licence professionnelle.

Le but du projet est de remettre la motorisation sur un fauteuil électrique.

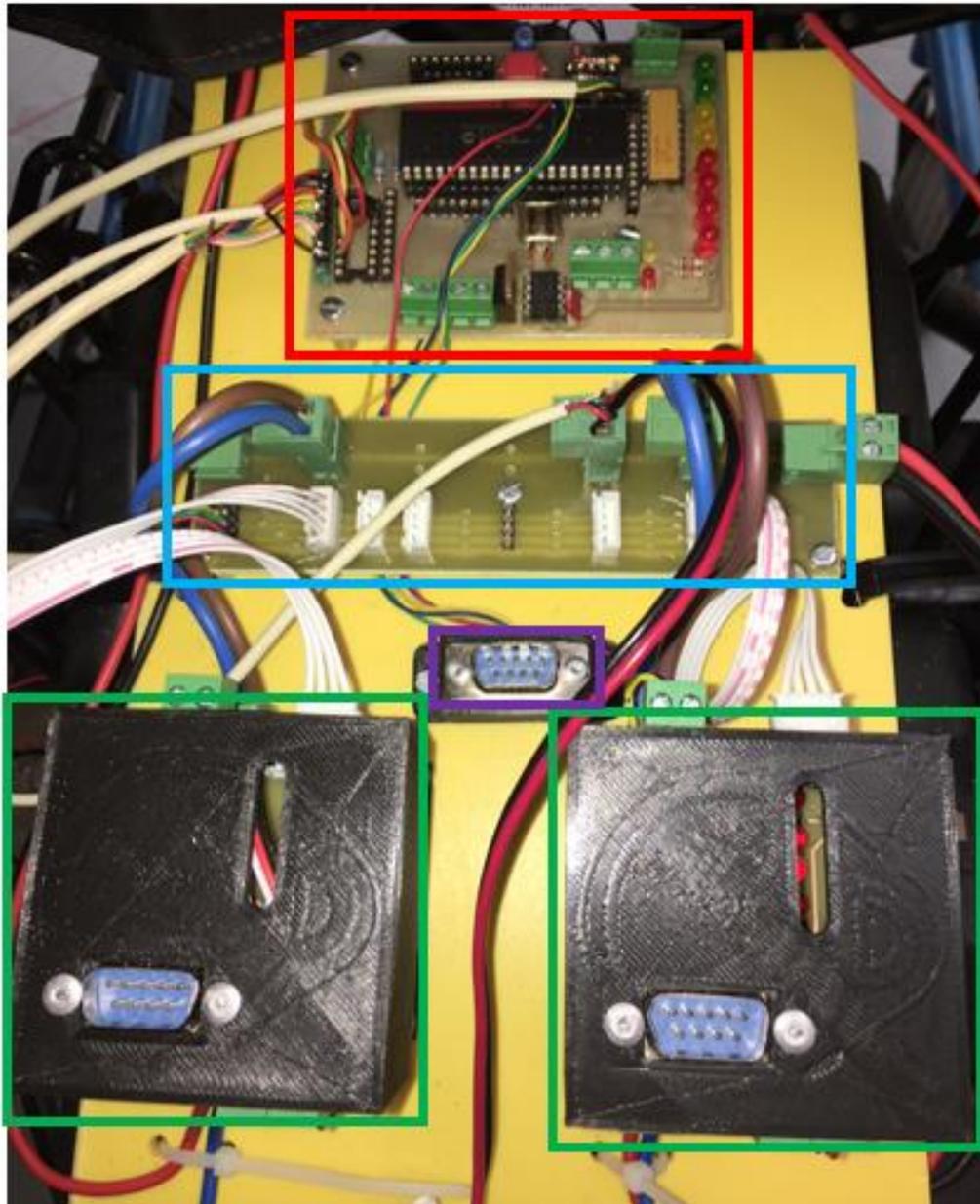
Le fauteuil à assistance électrique est utilisé par les personnes à mobilité réduite pour se déplacer plus facilement.



Matériel

Lorsque nous avons repris le projet en début d'année il était constitué des matériels suivants :

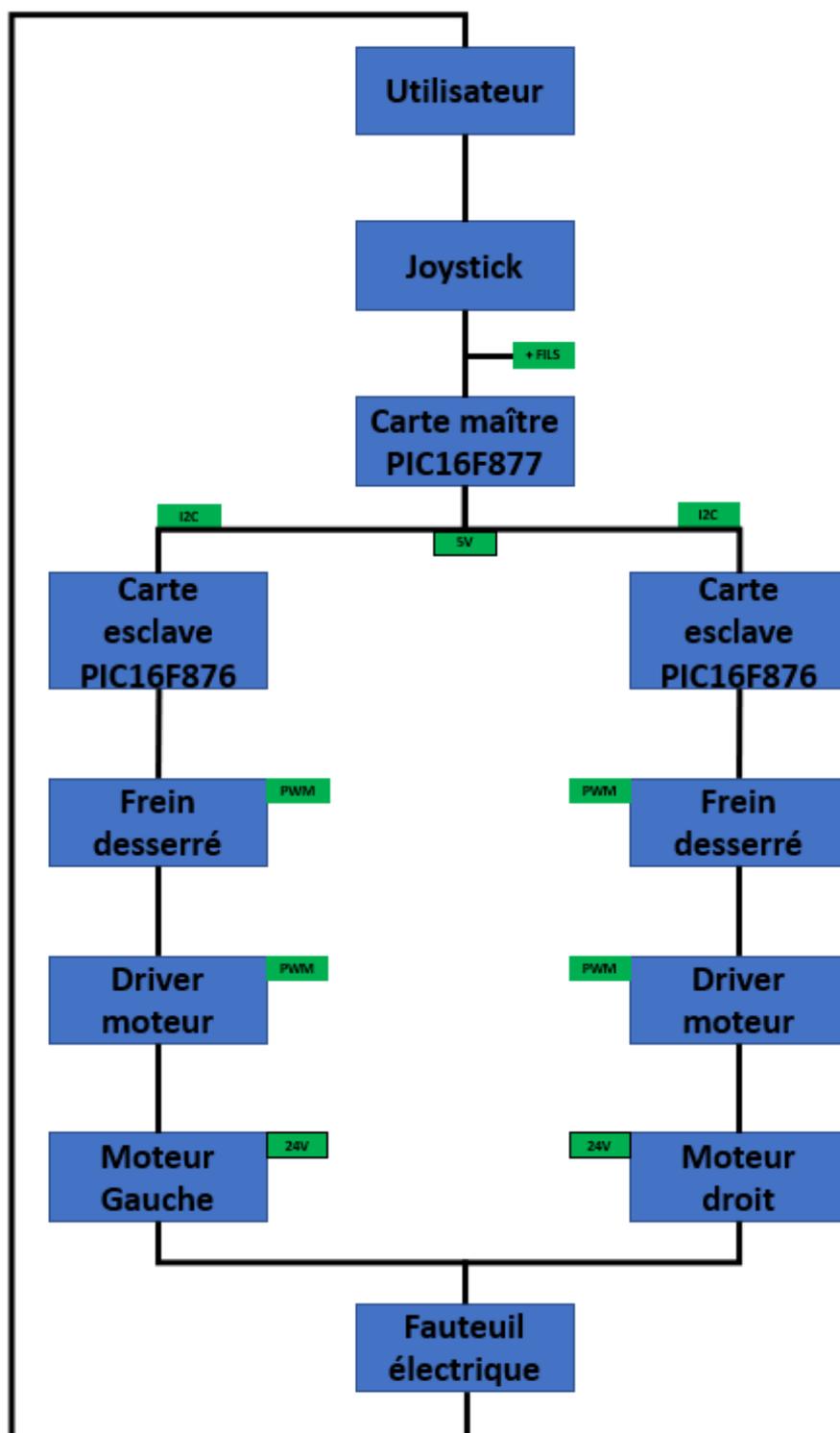
- Le fauteuil complet
- 2 batteries 12V (24V total)
- 2 driver moteur pour commander les deux moteurs du fauteuil
- 2 Carte esclave avec pour chacune de PIC16F876
- 1 joystick inductif
- Carte de maître avec PIC 16F877



- Carte mère
- Carte de distribution :
-I2C
-Puissance
- Liaison pour programmation
- Carte esclave



Schéma de fonctionnement



Le fonctionnement du schéma ci-contre est le suivant.

L'utilisateur va envoyer une consigne avec le joystick. Ensuite l'information de la position du joystick est envoyée sur la carte de commande (PIC16F877).

A la mise sous tension du fauteuil, les freins qui sont situés sur le moteur se débloquent à un seuil d'une tension supérieure à 18V.

Les signaux analogiques envoyés sur la carte de commande sont traités puis convertis en trame I2C. Pour ensuite être envoyés sur les deux cartes esclaves.

Les cartes esclaves vont ensuite envoyer les ordres de la trame I2C sur les drivers moteurs par des signaux de sens et une PWM pour donner la vitesse du fauteuil.

Pour finir le driver moteur donne la consigne au moteur du fauteuil.



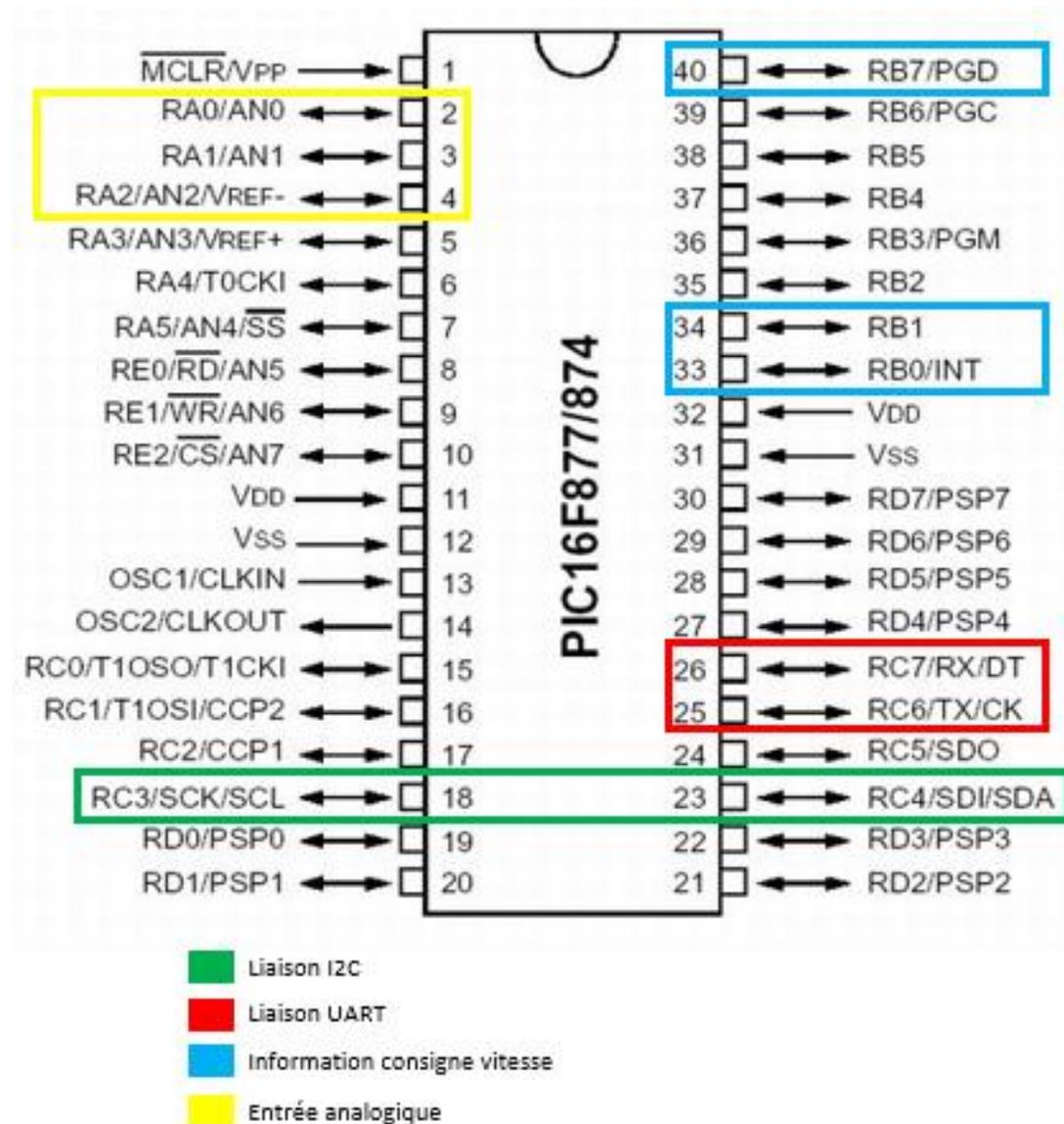
GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Composants

PIC16F877

Nous disposons pour notre carte mère d'un PIC16F877. On utilise trois entrées analogiques sur le port A pour la première on mesure la tension batterie avec la voie 1 et pour les deux autres on les utilise pour le joystick avec la voie 0 pour l'axe des Y et pour la voie 2 l'axe des X. On utilise sur le port B, les voies 0, 1, 7 pour l'indication avec des leds pour les différents modes de fonctionnement du fauteuil. Pour la liaison I2C on prend sur le port C, les voies 3 et 4. Pour la liaison UART on prend sur le port C, les voies 6 et 7.





GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

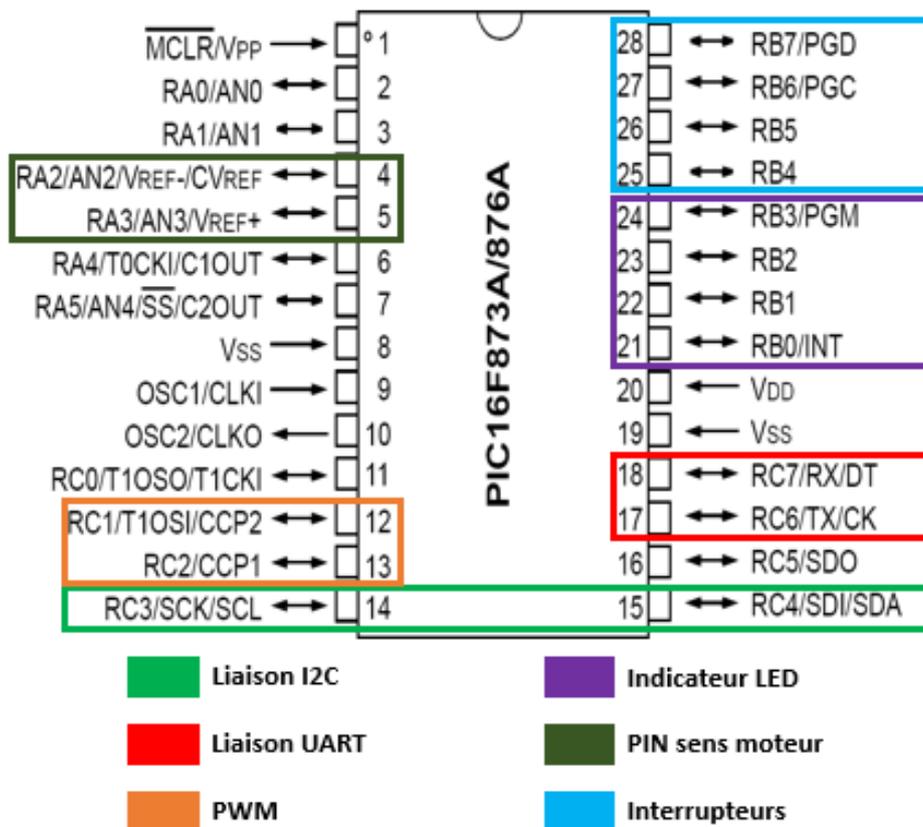
PIC16F876

Nous disposons pour nos deux carte esclave de PIC16F876. On prend sur le port A, les voies 2 et 3 pour établir le sens de rotation du moteur. Le sens est le suivant pour avancer, il faut mettre la voies 2 à 1 et la voie 3 à 0 et pour reculer il faut inverser la consigne envoyer au driver. On prend sur port C pour les PWM les deux voies 1 et 2. Pour la liaison I2C se sont les voies 3 et 4 et pour la liaison UART on utilise les voies 7 et 6. Pour les deux PWM on utilise la voie 1 pour la consigne vitesse et la voie 2 est utiliser pour le déverrouillage des frein mécanique.

LED	Signification
Voie 0	Frein mécanique
Voie 1	Reculer
Voie 2	Avancer
Voie 3	Arrêt électronique

On utilise sur le port B pour mettre quatre led de la voie 0 à la voie 3 et pour l'interrupteur on utilise les voie 4 à 7. Les quatre led sur l'esclave sont utiliser pour indiquer dans quel état sont les moteur du fauteuil. Les commandes I²C sont les suivantes :

Registre	Description	Codage
1	Sens su moteur	0x00 Arrret electronique
		0x01 Avancer
		0x02 Reculer
2	Vitesse	Consigne vitesse sur 8bits





GEII

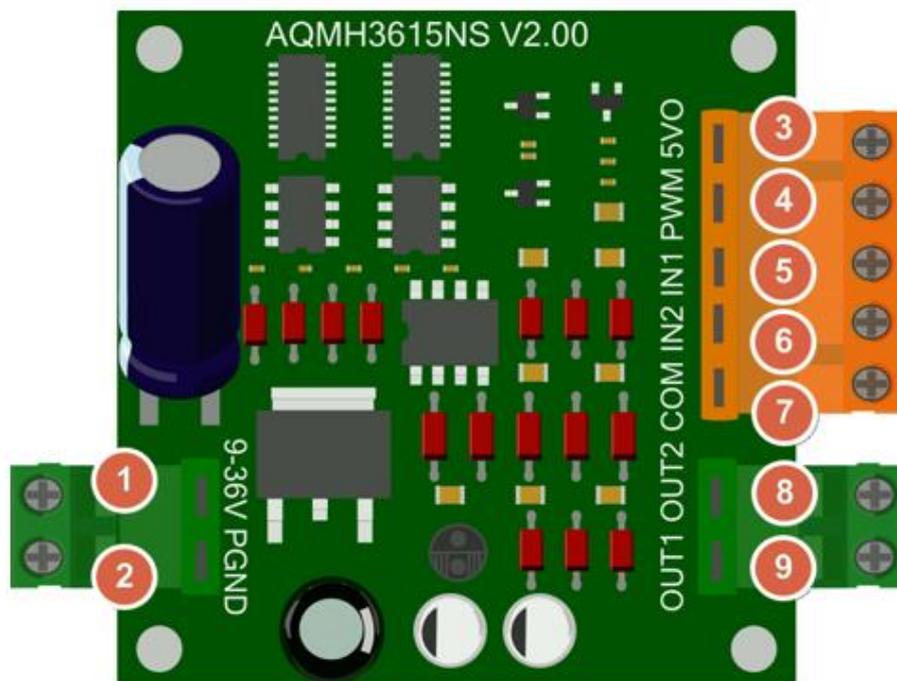
Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

DRI0042

Le driver moteur utilisé dans notre projet est le DRI0042 construit par DFRobot. Ce driver moteur est utilisé pour commander la puissance pour moteur DC. Le driver moteur est un hacheur 4Q. Ce driver moteur, nous a été donné pour notre projet par notre professeur référent.

Les caractéristiques du moteur sont :

- Alimentation 12 à 36V
- Courant 12A et un pic à 110A
- Entrée/Sortie : -niveaux haut 2 à 5,5Vcc -niveau bas 0 à 0,8Vcc
- Signal de commande en PWM



- | | | |
|--------------------------------------|---|----------------------------------|
| 1 Alimentation 24V | 4 PWM signal de contrôle vitesse | 7 COM: masse |
| 2 GND: masse | 5 IN1 signal sens moteur | 8 OUT1: borne + du moteur |
| 3 Sortie 5V: alimentation PIC | 6 IN2 signal sens moteur | 9 OUT2: borne du moteur |



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Joystick

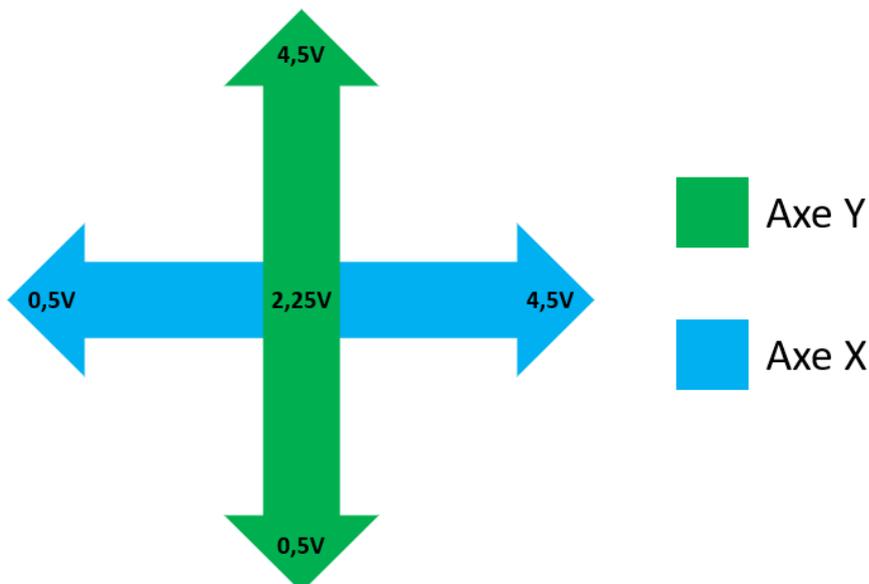


Masse 1
Axe Y 4
Axe x 5
Alimentation 5V 6

Le joystick que l'on utilise pour notre projet est le 3000 SERIES de chez APEM. Le principe de fonctionnement du joystick est de transformer une grandeur mécanique en grandeur électrique dans un espace défini par les caractéristiques du joystick. Le but principal du joystick est d'envoyer une consigne selon l'inclinaison que lui donne l'utilisateur sur le fauteuil.

Les données qui sont à disposition sur le datasheet, nous disent que par rapport à la tension d'alimentation. Nous avons une différence de 0,5V. Donc lors de l'inclinaison du joystick dans une certaine position, la tension qui sort des deux varie entre 0,5V et 4,7V et le milieu est situé à 2,25V. Lors de la conversion analogique numérique on obtient alors :

Niveau	Axe X	Axe Y
Haut	940	940
Milieu	515	515
Bas	82	82





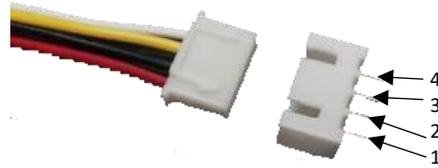
GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Le protocole I2C

Le protocole I²C est implémenté au cœur de l'interconnexion entre nos différents calculateurs. Le but du protocole I²C est de communiquer avec seulement 4 fils. Ces fils sont caractérisés des façons suivantes :

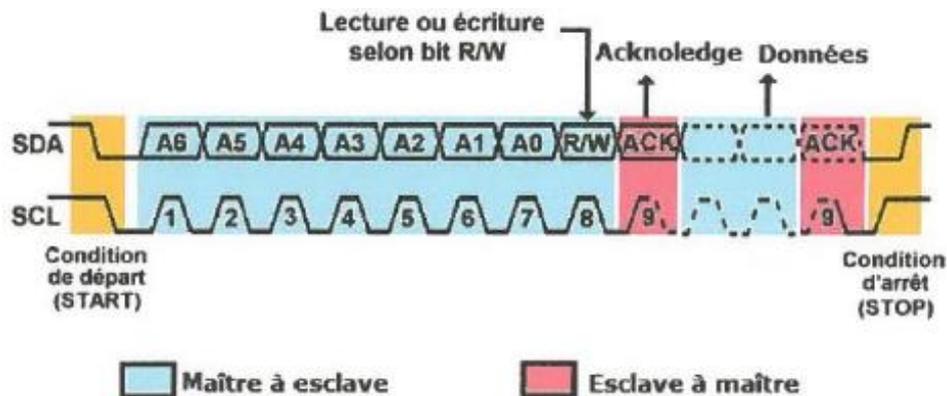
- 1- GND
- 2- Clock (SCL)
- 3- Data (SDA)
- 4- Masse



Le fil Data (SDA) est utilisé pour transmettre les données. Le fil Clock (SCL) est utilisé pour transmettre un signal d'horloge synchrone (signal qui indique le rythme d'évolution de la ligne SDA).

Nous avons sélectionné un connecteur 4 voies de type JST XH qui possède un détrompeur cela facilite les connexions sur la carte de distribution.

Fonctionnement du bus I²C :



La Liaison UART

La liaison UART (Universal Asynchronous Receiver Transmitter) est une liaison intégrée dans principalement tous les microcontrôleurs, configurée à 19200 Bauds dans tous les microcontrôleurs de notre architecture électroniques. Cette liaison nous permet d'injecter le code compilé à l'intérieur des différents microcontrôleurs du système elle nous permet également de debugger nos prototypes. Nous avons conçu un boîtier de conversion USB vers UART à l'aide du CP2104, le bouton blanc permet d'effectuer un reset de la cible ou le boîtier est connecté.



GEII

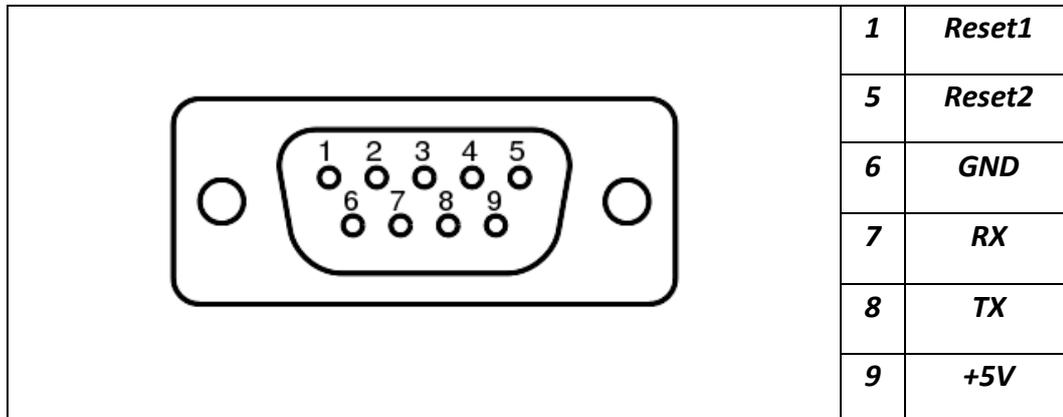
Département
& Informatique
IUT



Boitier de téléversement de PC vers carte le bouton blanc au centre est utiliser pour reseter la cible.

Le raccordement de la cible s'effectue grâce a l'aide d'un connecteur DB9.

USB<-> UART



Loi de commande entre le joystick et les moteurs

Pour pouvoir faire une rotation soit à gauche ou à droite nous avons mis au point un loi de commande.

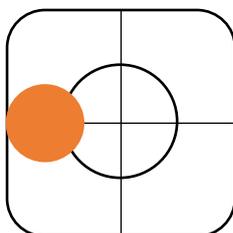
Cette loi est définie de la manière suivante :

Nous avons créé deux modes selon le besoin de l'utilisateur. Les deux modes sont le mode manœuvre et le mode route.

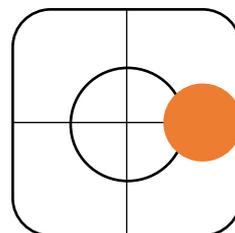
Pour le mode Manœuvre :

Nous avons deux options de rotation sur place qui sont tourner à droite et tourner à gauche. Plus le joystick est à droite plus la vitesse de rotation à droite sera important. La roue gauche tourne dans le sens horaire alors que la roue droite tourne dans le sens antihoraire.

Plus le joystick est à gauche plus la vitesse de rotation à gauche sera importante. La roue gauche tourne dans le sens horaire alors que la roue droite tourne dans le sens antihoraire.



MD ↓
MG ↑

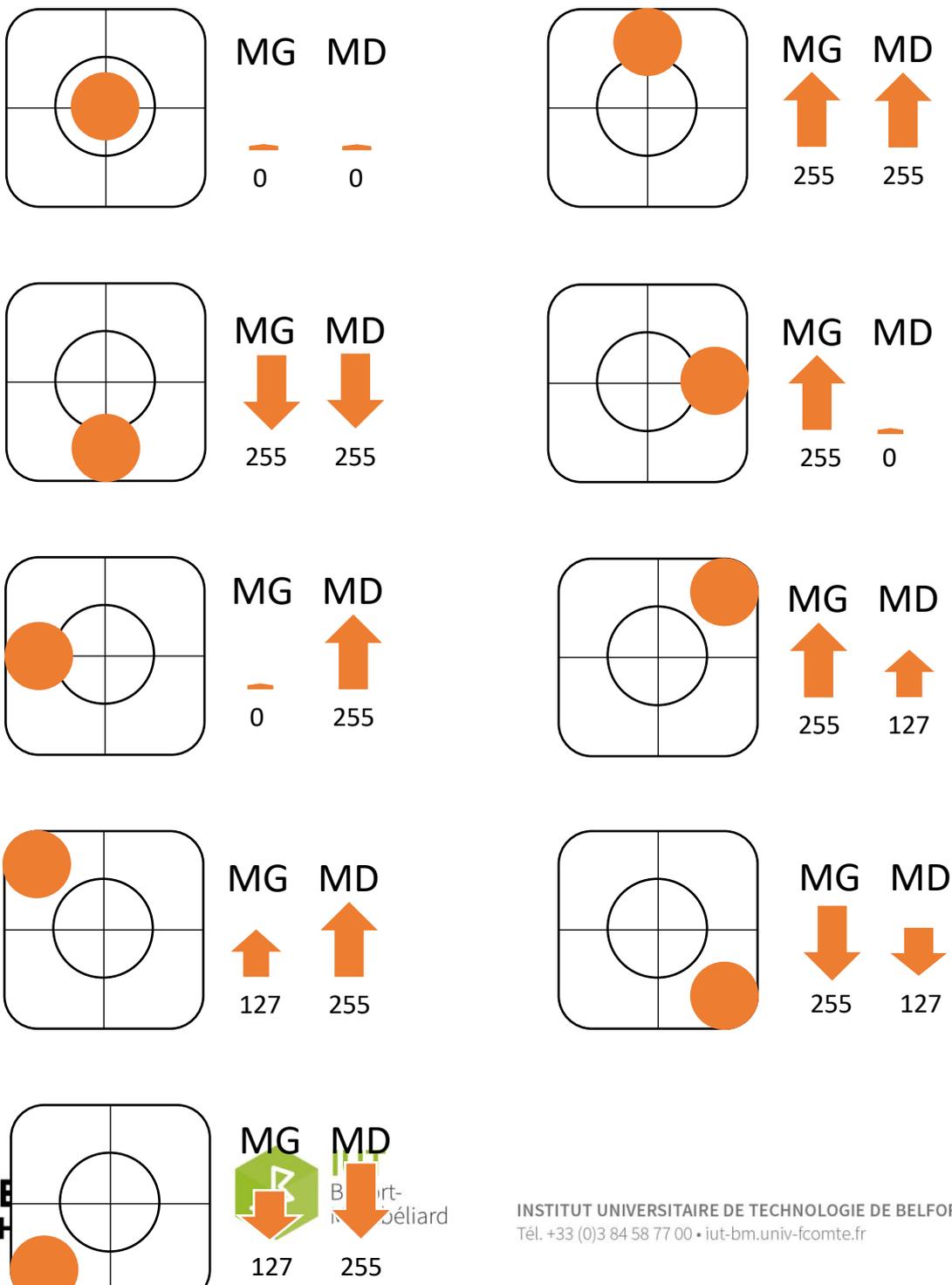


MD ↑
MG ↓



Pour le mode Route :

Le mode Route dispose de deux gammes de vitesses, petite et grande vitesse. Selon le mode choisi en petite vitesses le fauteuil roulera doucement avec une plus grande flexibilité dans les virages étroits ainsi il pourra se faufiler dans des endroits très exigü. Dans cette petite gamme de vitesse le joystick sera beaucoup plus sensible aux volontés de l'utilisateur. Cette flexibilité de conduite s'obtient en changeant la vitesse d'une roue l'une par rapport a l'autre selon la position du joystick. Dans la seconde gamme de vitesse le fauteuil sera plus rapide ainsi il pourra effectuer de long parcours avec une maniabilité accru. Dans cette gamme la sensibilité du joystick et alors réduite afin de ne pas avoir de mouvement trop brutal ainsi l'utilisateur peut contourne des obstacles et effectuer de jolies courbes tout en pouvant corriger sa trajectoire très facilement.



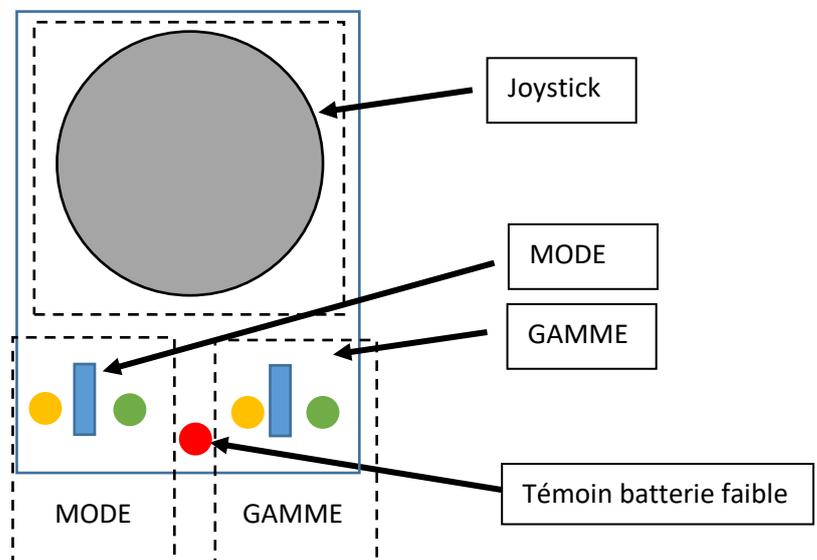


Freins à perte de tension

Les freins fonctionnent en logique négative, ils sont actifs lorsqu'il n'y a pas de tension. Cela est dû à des raisons de sécurité, si l'utilisateur est en pente et qu'il y a un défaut électrique au niveau de l'alimentation, le fauteuil ne dévalera pas la pente, les freins se bloqueront automatiquement dès qu'ils verront à leurs bornes une tension inférieure à 5V.

Interface Homme Machine et Instrumentations

L'interface Homme Machine s'effectue grâce boîtier bleu sur l'accoudoir gauche du fauteuil ce boîtier ce compose d'un joystick inductif de deux boutons poussoir pour pouvoir sélectionner la gamme et puis le mode de conduite. Ce boîtier se compose également de plusieurs voyant a LED pouvant indiquer à l'utilisateur l'état actuel du fauteuil.

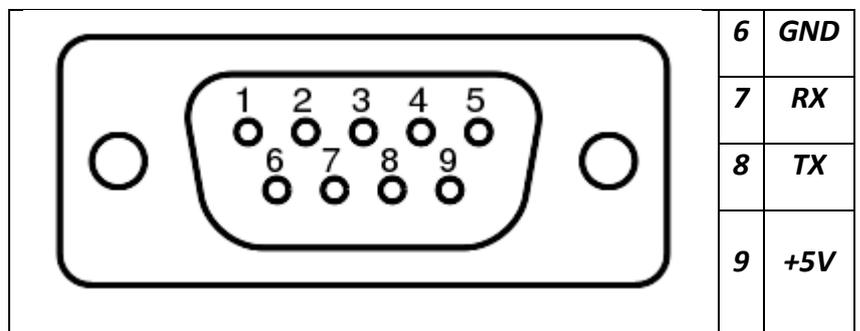




Cablage des différentes fonctions :

Connecteur Vert	Fonction	Fil	Pin Microcontrôleur
1	Alimentation	Rouge	+5V
2	Alimentation	Noir	GND
3	Voix X	Jaune	RA0
4	Pont résistif VBat	Rouge	RA1
5	Voix Y	Bleu	RA2
6	LED BAT	Blanc	PB7
7	Interrupteur Mode	Orange	PB0
8	Interrupteur Gamme	Vert	PB1

Instrumentation LabVIEW :



Le petit boîtier de débogage connecter à la prise Programmation de la carte maître. Ce module d'extension possède un ESP8266-Model 07 il intègre une fonction d'adaptation de niveau logique 3.3v vers 5V ainsi 3 LED qui indique l'état du boîtier. Il converti le signal UART (19200bauds) en entrée en paquets TCP sur le port 9876. Le paquet est ensuite interpréter par LabVIEW afin d'être visualiser sur la face avant.

IP : 192.168.1.1

SSID : Fauteuil_Debug_VL

Port : 9876

Mot de passe : jacqueline



Par défaut le boitier est configuré en mode point d'accès la LED Verte indique le démarrage de cette fonction.

La LED bleu clignote quand une trame série est reçu et envoyer.

La LED blanche au centre indique qu'un client et connecter sur le module via un terminal Telnet (style PUTTY)

Connexion PUTTY :

<p>Etape 1 remplir les champs IP et Port, cocher Telnet puis Open</p>	<p>Etape 2 : affichage de la trame</p>

Etats boitier débogage :

<p>Boitier correctement initialiser en Mode point accès.</p>	<p>Connexion LabVIEW établie et opérationnel.</p>	<p>Connexion PuTTY établie.</p>

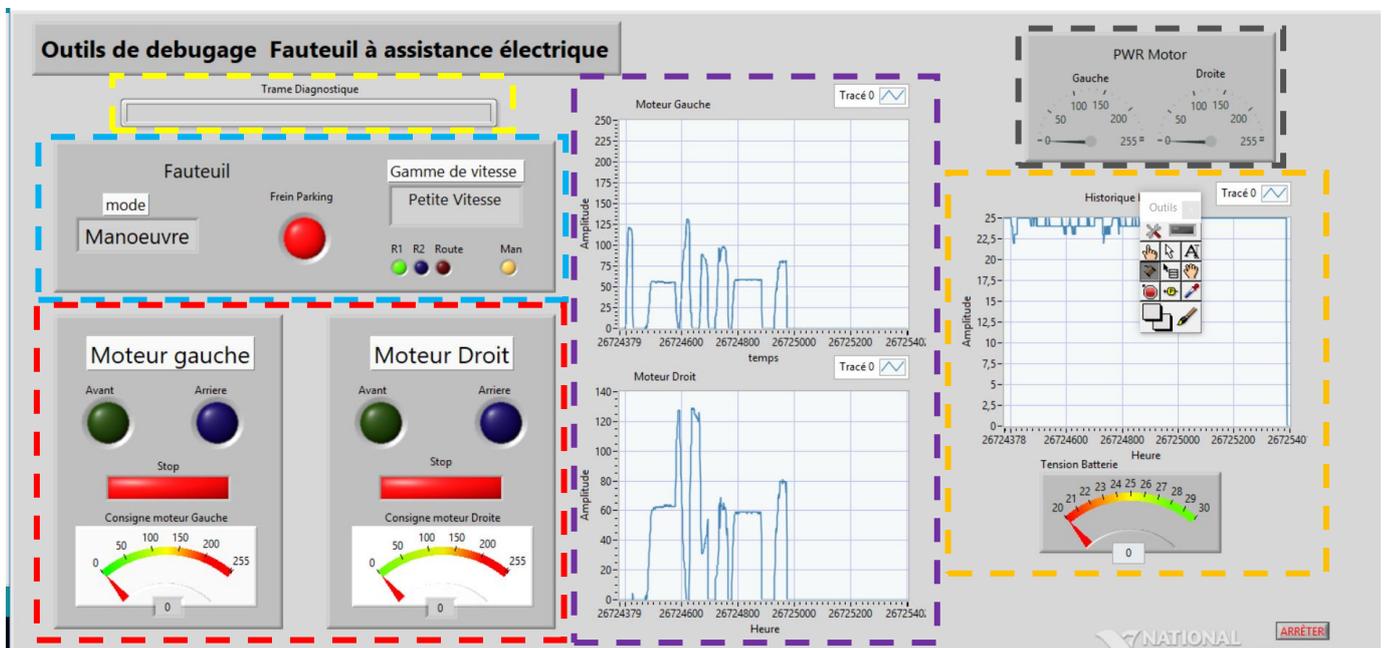


Constitution de la trame :

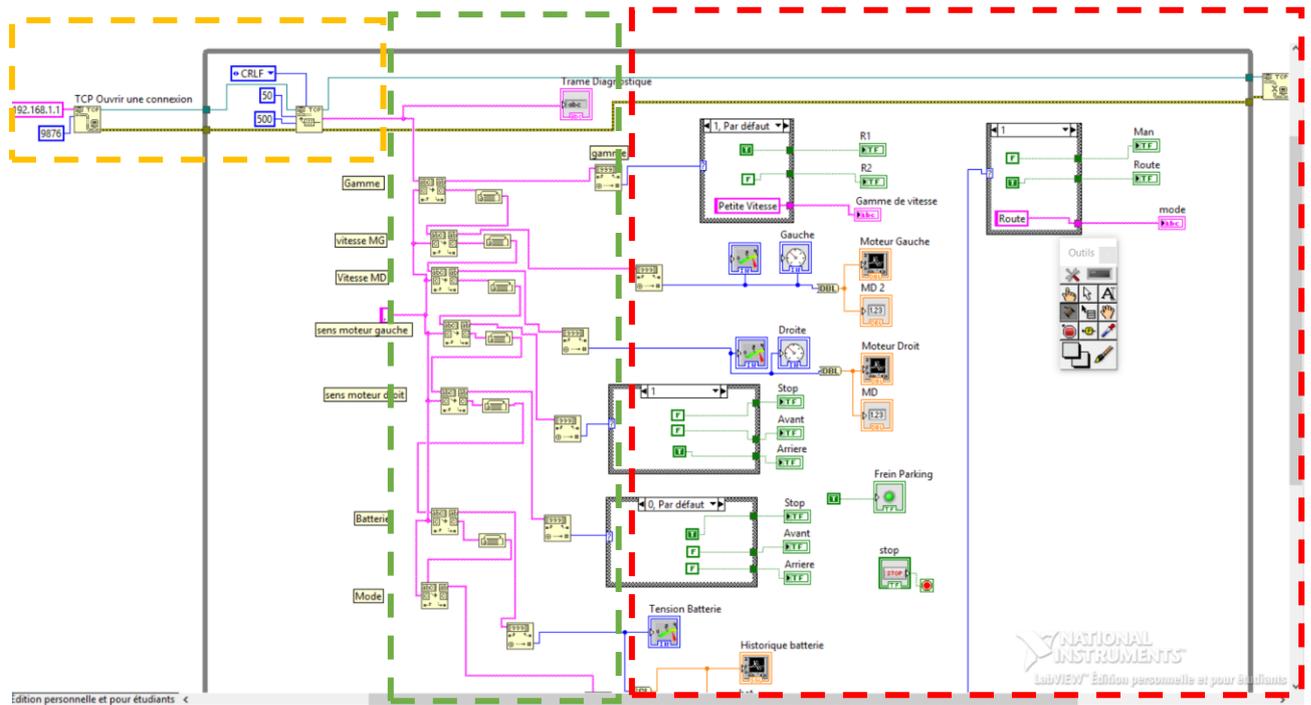
Chaque donnée est séparée par une virgule afin pouvoir décoder facilement avec LabVIEW

1 , 0 , 0 , 1 , 1 , 25.85 , 1						
Gamme : 1 = grande 2 = petite	Vitesse MG 0 à 255	Vitesse MD 0 à 255	Sens MG 0 = Frein 1 = Avancer 2 = Reculer	Sens MD 0 = Frein 1 = Avancer 2 = Reculer	Tension Batterie	Mode 1 = route 2 = manœuvre

Interface LabVIEW :



L'encadrer jaune trame reçue via le boitier L'encadrer bleu représente l'état général du fauteuil, l'encadrer rouge représente l'état de chaque moteur sur l'encadrer violet représente un histogramme des consignes envoyer aux moteurs sur la droite on peut observer l'état de la batterie ainsi que l'historique de la tension batterie. Enfin l'encadre gris représente un autre vu des moteurs pour mieux visualiser la consigne de chaque roue.



L'encadre jaune représente l'initialisation du module TCP de LabVIEW ensuite l'encadrer vert représente le traitement de la chaîne de caractères, Voici les modules LabVIEW utiliser

Rechercher/Découper une chaîne



Rotation dans une chaîne



Chaîne décimale en nombre



L'encadrée rouge représente le traitement des nombres pour les afficher sur la face avant.

Signalisation

Nous avons mis au point une carte de signalisation à l'aide d'un Arduino NANO de buzzer et de LED de puissances

Grace a l'outils ARDUINO IDE il es très simple de mettre en œuvres cette carte de signalisation

La gestion des LED s'effectue grâce au PT4115 qui assure une alimentation en courant continue au LED

Les LED utiliser sont des LED de puissance RGB d'une puissance de 3W totale chaque couleur s'alimentent en courant constant de 300 mA le feu blanc quant à lui est une LED une LED blanche de 1W qui s'alimente en courant constant de 300mA.



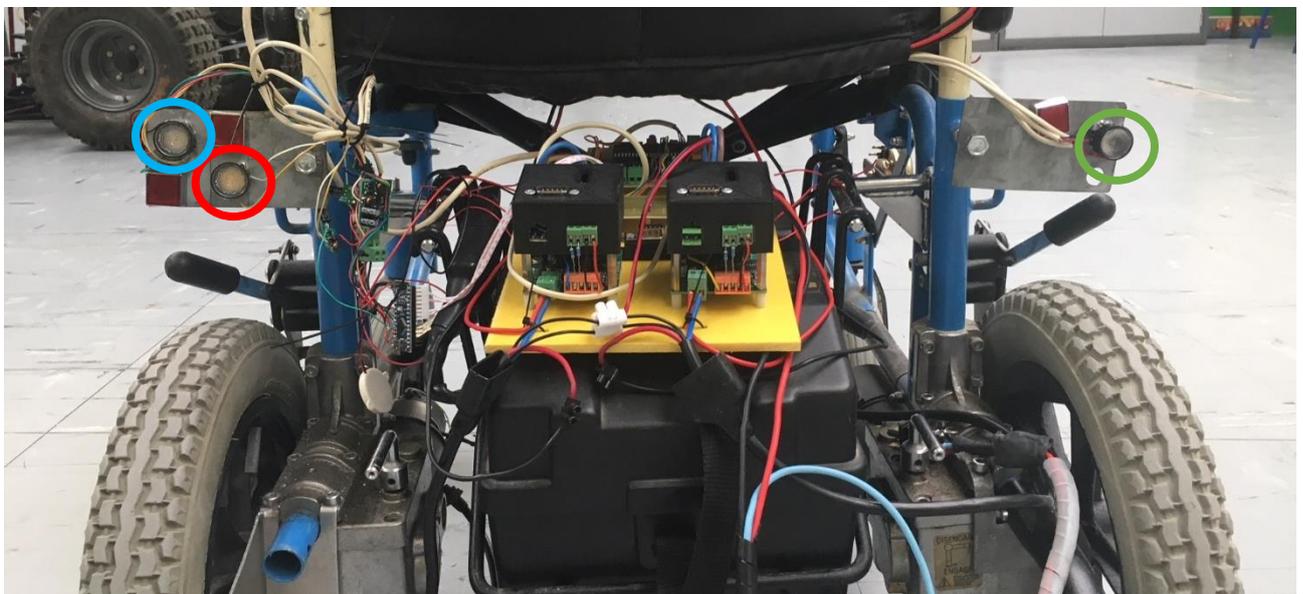
Nous utilisons la librairie TONE pour générer des sons au travers du buzzer passif. La carte est reliée directement sur le bus I²C du fauteuil son adresse sur le bus est l'adresse 0. La liste des commandes est la suivante :

Carte Signalisation Adresse 0	
Valeur de l'octet	Désignation
0x01	Feux clignotant Droit
0x02	Feux clignotant Gauche
0x03	Feux de stop
0x04	Feux de recule et Bip de recule
0x06	Avertisseur sonore « klaxon »
0x07	Bip court
0x08	Arrêt de tous les feux

La configuration actuelle utiliser tout les pins PWM sont utiliser pour 5 pour les feux et 1 pour le buzzer.

Les commandes des différents feux s'effectuent automatiquement en fonction de la position du joystick sans action particulières de l'utilisateur.

Voici une vue d'ensemble de l'implantation des feux sur le fauteuil



Le cercle entouré en vert le feux clignotant et feux stop droit, Le cercle entouré en bleu le feux clignotant et feux stop gauche enfin Le cercle entouré en rouge le feux de recule blanc.



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Conclusion

Difficulté rencontrée :

Nous avons eu du mal à mettre au point l'I2C car nous avons des problèmes de communication.

Nous avons mis du temps pour pouvoir rendre autonome le fauteuil.

Ce qu'on a appris :

Nous avons appris à travailler en autonomie. Nous avons aussi pu utiliser toute nos compétences apprises au cours de cette licence et de nos formations précédentes.

Les évolutions futures :

Passer l'IHM en I2C. Ajouter un capteur de courant pour avoir une gestion de l'énergie sur le fauteuil.

Rajouter un module GPS pour avoir sa vitesse et sa position en temps réel.



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Annexes :

Code carte maitre : voir fichier joint Master_rev14.c

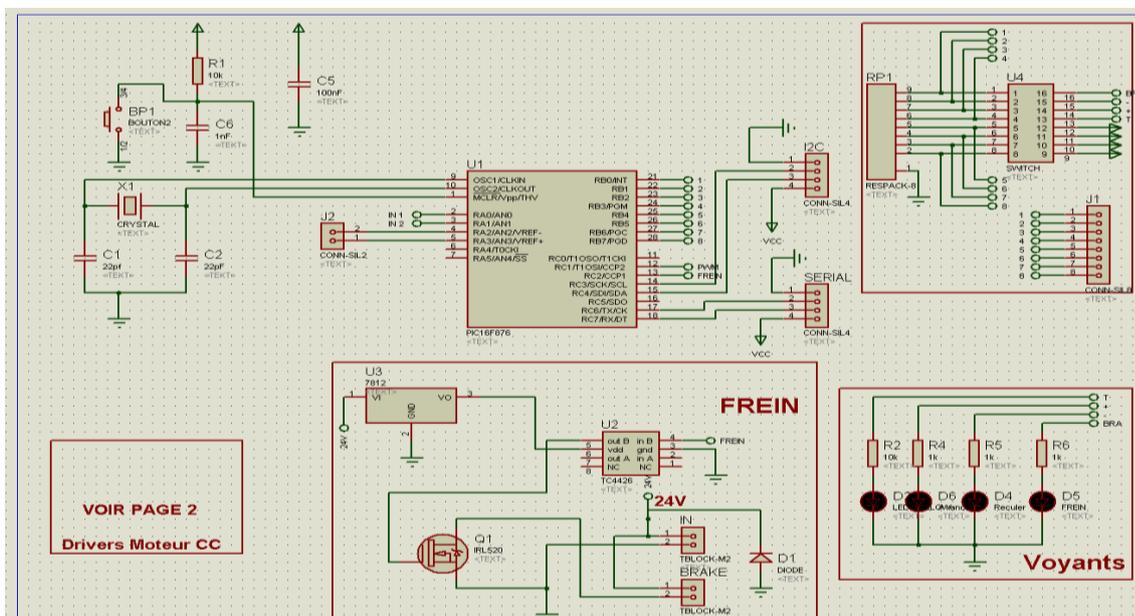
Code carte esclave : voir fichier joint Slave_rev4.c

Code carte signalisation : voir fichier joint phare.ino

Code boitier instrument : voir fichier joint TCP_UART.ino

Code LabVIEW : fauteuil.vi

Carte esclave :





Circuit carte esclave :

