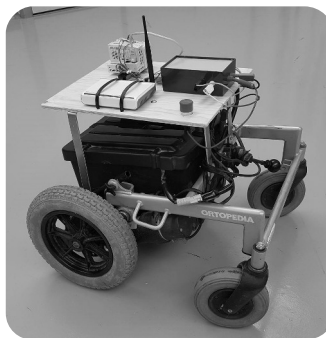




Projet Licence Pro V.E.G.A Robot tondeuse



RAPPORT DE PROJET

Etudiants intervenants

Célyn BELLORGIE
Julien GIRARDET

Professeur référent

Eric CHRETIEN

Année 2016- 2017



1) Remerciements

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué à la réalisation de notre projet et à son bon déroulement, et plus particulièrement à notre professeur référent d'automatisme Mr CHRETIEN ainsi qu'aux personnels techniques Mr THIEBAUD et Mr HUBERT, pour leurs conseils avisés, qui nous ont aiguillés au mieux dans nos recherches. Merci pour l'aide qu'ils nous ont apportée sans oublier le temps qu'ils ont bien voulu nous consacrer ce qui nous a permis de mener à bien notre projet.





2) Sommaire

1. Remerciements	page 2
2. Sommaire	page 3
3. Introduction	page 4
4. Etude	page 5 à 20
1. Moteur	page 5
2. Batteries	page 6
3. Driver moteur	pages 6 à 8
4. Automate	page 9
5. Routeur Wifi	page 10
6. Régulateur de tension	page 10
7. Câblage existant	page 11
8. Capteur inductif	pages 11 et 12
5. Réalisation	pages 12 à 19
6. Résultats et conclusions	page 20
7. Annexes	pages 21 à 28



Dans le cadre de notre licence professionnelle, nous sommes amenés à réaliser un projet technique au sein de l'IUT de Belfort-Montbéliard.

Notre travail a pour but de réaliser un robot tondeuse à gazon à l'aide d'un châssis composé de 2 roues folles et de 2 roues motrices. De plus, le déplacement et la direction seront assurés par deux moteurs à courant continu.



Nous sommes chargés de mettre en place :

- différents programmes de tonte
- de gérer l'énergie électrique du robot
- de faire un pilotage à distance
- de proposer une solution pour délimiter la zone de tonte





Après quelques recherches sur un châssis avec 2 roues motrices et 2 roues folles, nous nous sommes aperçus que nous pouvions récupérer le châssis d'un fauteuil roulant pour handicapé inutilisé dans notre salle de projet plutôt que d'en créer un qui allait nous demander beaucoup de travail et d'argent.

Voici le châssis avec ces 2 moteurs et ces 2 batteries :

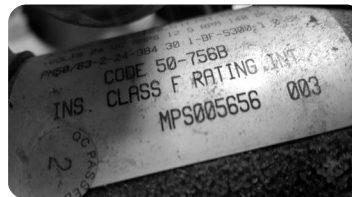


Ceci dit, le fait de récupérer un châssis nous impose des contraintes de motorisation et de commande.

4.1) Moteurs :

Après analyse des moteurs, voici ses caractéristiques (pour un moteur) :

- 24 Volts DC
- 12,5 ampères
- 140 RPM





4.2) Batteries :

Voici la caractéristique des batteries Optima (pour une batterie) :

- Tension : 12V
- Démarrage à froid : 660A
- Capacité : 48Ah
- Poids : 16.6Kg

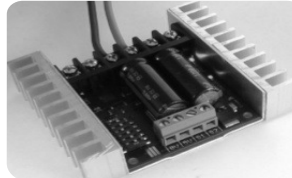
Avec deux batteries en série, nous obtenons 24 volts en sortie.



4.3) Driver moteur :

Enfin voici le dernier organe qui nous était imposé avec le châssis, le driver Sabertooth qui contrôle la commande des moteurs :

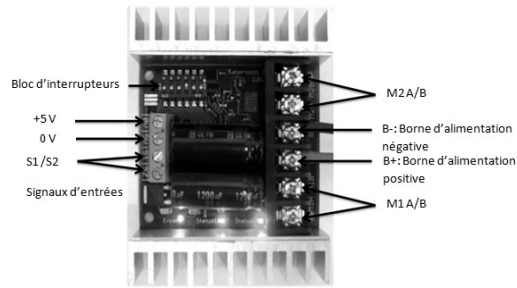
- Tension d'entrée : 24 v
- Courant de sortie : 25 Ampères en continu
- Différents modes de commande :
 - Entrée analogique
 - Entrée R/C
 - Liaison série



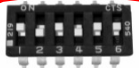
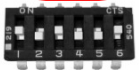
Grâce à la documentation technique du driver Sabertooth, nous avons appris plus sur son mode de raccordement. En effet, il contient 2 bornes d'alimentation qu'on peut exploiter en 24 volts et il permet d'avoir du 0 - 5 volts en sortie. De plus, il offre la possibilité de contrôler deux moteurs grâce à ces 2 bornes + et - de sortie en 24 volts ainsi que 2 bornes de transmission de signal. Enfin, il possède un bloc d'interrupteur pouvant mettre les switches à 0 ou à 1 et ainsi choisir le mode de fonctionnement et sa vitesse de transmission.



Voici un schéma avec les connectiques du driver :



Pour notre projet, nous allons utiliser la liaison série RS232 car par la suite, c'est le mode de transmission qui correspond le mieux pour l'exploiter sur un automate. Pour passer en liaison série il suffit de passer en OFF le switch 2 et de jouer avec le 4 et 5 pour modifier la vitesse. Ci-dessous, voici la configuration des switchs pour avoir une vitesse de 9600 Bauds, ce qui est la vitesse recommandée par le constructeur.

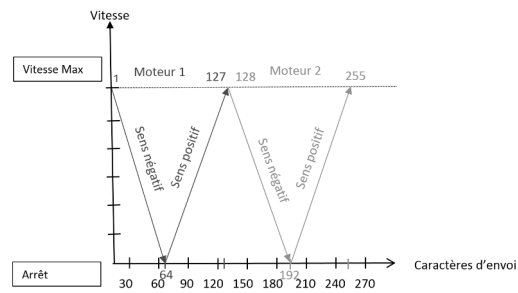
 2400 Baud: 01x00x	 9600 Baud: 01x10x
 19200 Baud: 01x01x	 38400 Baud: 01x11x

Dans cette configuration, le driver a 8 bits de données, n'a pas de bits de parité, par contre, il a un bit d'arrêt.



Par ailleurs, le driver transmet des caractères de données qui permettent d'agir sur la vitesse et le sens de rotation des moteurs. En effet, ce dernier a une plage allant de 1 à 255, le premier moteur fonctionne de 1 à 127 avec 64 comme valeur d'arrêt et le deuxième de 128 à 255 avec 192 comme valeur d'arrêt.

Voici un graphique montrant l'influence des caractères d'envoi sur la vitesse des moteurs :



La plage de marche avant du premier moteur est de 65 à 127 et 193 à 255 pour le deuxième, pour la marche arrière c'est l'intervalle de 1 à 63 pour le moteur 1 et 128 à 191 pour le moteur 2.



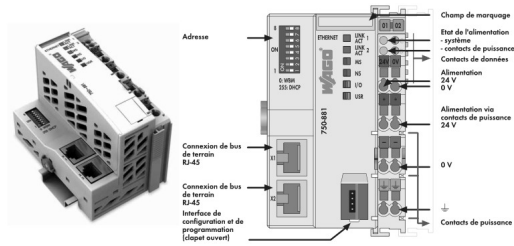
Par exemple, pour aller en marche avant en vitesse maximum, il faut envoyer simultanément les valeurs 127 et 255; pour une marche arrière maximum, c'est 1 et 128 et c'est 64 et 192 qui contrôlent l'arrêt des moteurs.



Pour l'intelligence du robot, nous avons eu l'idée d'utiliser un microcontrôleur ou un automate. Ce dernier, nous a été imposé car il dispose d'une prise RJ45 qui, par la suite, assurera une communication wifi.

Voici l'automate WAGO 750-881 dont nous disposons :

API - Contrôleur de bus de terrain programmable ETHERNET
CPU 32-Bits, multi-tâches

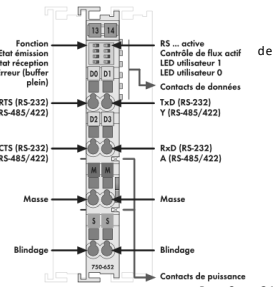


On peut voir que cette API est alimentée en 0-24 volts, qu'elle contient deux prises RJ45 et un bloc d'interrupteur d'adresse.

Nous lui avons ajouté un module RS232 (750-652) pour assurer une liaison série :

Ce module contient une borne transmission TX et une de réception RX.

De plus par défaut, il communique avec une vitesse de 9600 bauds comme la recommandation du driver.



Page 9 sur 21



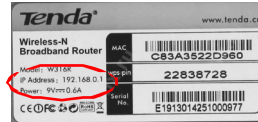
Pour assurer un pilotage à distance, le routeur wifi nous a semblé être la solution incontournable. Notre tuteur de projet en avait un à sa disposition qu'il nous a fourni, c'est un Tenda W316R.

Voici ses caractéristiques :

- Bande passante: 2.4 Ghz
- Vitesse de la connexion sans fil: 150 Mbps
- Interface: 1 Port Ethernet WAN
- 4 Ports LAN RJ-45
- Bouton Reset



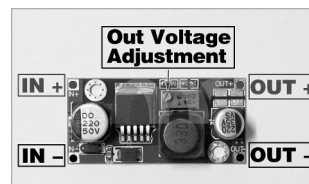
- Alimentation: 9 V
- Courant : 0.6 A
- Adresse IP : 192.168.0.1



4.6) Régulateur de tension

Sachant que le routeur wifi a besoin de 9 volts pour fonctionner et que nous avons du 24 volts à disposition venant des batteries ou du 0-5 volts venant du driver, nous avons été contraint de prévoir un régulateur de tension pouvant accepter une tension d'entrée de 24 volts pour la convertir en 9 volts. Nous avons choisi un module LM2596S :

- Tension d'entrée: 4 à 35 Volts
- Tension de sortie: 1,25 à 30 Volts
- Courant max de sortie: 3A





Nous avons récupéré et identifié le câblage existant du fauteuil handicapé, on retrouve les câbles d'alimentations des moteurs et ceux venant des batteries auxquels nous avons associé des fiches bananes.

La paire blanche et rose pour le moteur 1

La paire jaune et orange pour le moteur 2

La paire rouge et noir pour la sortie des batteries



4.8) Capteur inductif

Nous avons décidé de mettre un capteur inductif sur le châssis avec des morceaux de métaux sur la jante d'une roue pour contrôler la distance, la vitesse et proposer un mode automatique dans notre programme afin que le robot se déplace en autonomie en lui entrant juste une longueur et une largeur. Nous avons récupéré un par le biais de notre tuteur de projet dont les caractéristiques n'étaient pas fournies par manque de référence.

Tension d'alimentation : 24 V

Détection max : 1 cm environ

4 fils : 2 fils d'alimentation (+) et (-) de l'appareil
2 fils pour la transmission des signaux de sortie

4 morceaux de métal seront placés sur la jante à un intervalle de 25 cm sachant que le périmètre d'une roue est d'environ 1 m afin de nous permettre d'être précis à 25 cm près.



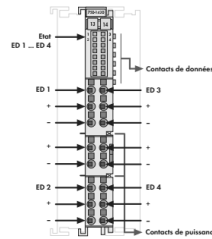


De plus, nous avons eu besoin d'un module d'entrée (750-1420) pour l'automate qui était à disposition en salle d'automatisme, indispensable pour la programmation du capteur.

Cette borne d'entrées digitales (T.O.R.) permet de raccorder 4 capteurs avec une technique de raccordement pour 3 conducteurs.

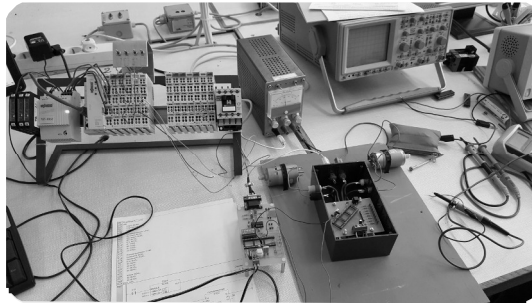
Elle permet de faire l'acquisition de signaux binaires comme ceux par exemple de capteurs, de commutateurs ou de capteurs de proximité.

Borne d'entrées digitales à 4 canaux 24 V DC
2 à 3 conducteurs à commutation positive (PNP)



5) Réalisation

Pour tester le bon fonctionnement des moteurs et du driver, nous avons mis en place une maquette dans laquelle nous avons géré l'envoi des caractères en liaison série.





Nous avons dû connecter notre automate sur un Max232 avec une prise DB9 pour baisser la tension car la liaison série du module fonctionne en +/- 10 volts et notre driver a en sortie 0/5 volts.

Ensuite, on a raccordé le RX du Max232 sur la borne S1 du driver que nous avons placé dans une boîte avec des douilles de sécurité. Enfin le Sabertooth alimenté en 24 volts envoie les caractères que nous transmettons par le biais d'un programme aux moteurs et alimentes ces derniers.

Nous nous sommes aperçus que tout le système précédent fonctionnait correctement, donc la création de notre propre carte Max232 s'impose pour réduire sa taille et ainsi l'intégrer dans notre boîtier.

Pour réaliser notre Max232, nous avons travaillé à l'aide du logiciel ISIS qui permet d'éditer des schémas électriques et de les compléter avecARES qui est un outil d'édition et de routage informatique.

Au total, notre carte Max232 est composée de :

- 5 condensateurs de 1 μ F (1 pour le filtrage, 2 pour le circuit booste, 2 pour la pompe de charge)

- 2 résistances de 270 Ω

- 2 Leds jaunes avec une chute de tension de 2,1 V

- 1 connecteur 4 broches (2 pour les signaux et 2 pour le 0/5 V)

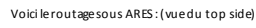
- 1 prise DB9 male

- 1 Max 232

La présence de deux résistances et 2 Leds permettent de savoir si des signaux circulent sur la ligne de transmission.

Légende de la carte :







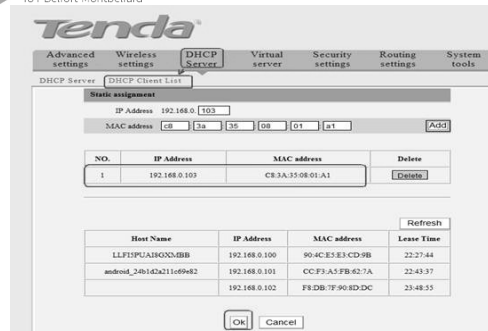
Ci-dessous, notre Max 232 après impression et soudure des composants :



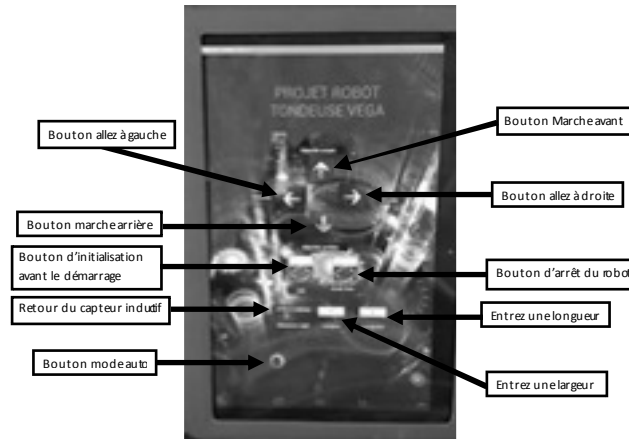
Une fois notre carte terminée, nous nous sommes attaqués à la partie réseau. Dans un premier temps, nous avons connecté notre automate et notre ordinateur en liaison filaire au routeur Wifi car ces derniers n'ont pas de carte réseau mais juste des prises RJ45, puis une tablette en Wifi cette fois-ci pour créer un réseau local.



Ensuite nous avons attribué les adresses IP délivrées par le routeur Wifi grâce au serveur web du Tenda sur Codéys (notre logiciel de programmation automate) et Wago ID (le logiciel de création d'IHM sous un système Android).



Après avoir fini le paramétrage de la communication à distance, nous avons créé notre écran de contrôle et de pilotage sur notre tablette (I.H.M.)





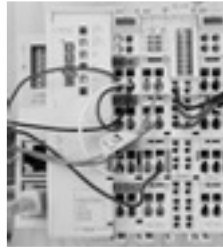
Pour rendre nos boutons sur l'I.H.M fonctionnels, il a fallu les assigner aux variables existantes dans notre programme sous Codesys en les déclarant en mot (%MW).

Dans notre programme codé en Ladder, nous avons permis à l'utilisateur de contrôler les 4 sens de marche à l'aide des 4 flèches de direction, d'avoir un retour de la distance parcourue par le robot, de le stopper à distance, de définir une largeur et une longueur. En appuyant sur le cercle rouge, le robot tond en autonomie et s'arrête une fois la surface terminée.

A présent, tous les organes du robot ont été testés et programmés, il ne nous reste plus qu'à tous les assembler. Donc, les deux batteries en série sortent du 24 volts que nous amenons jusqu'aux bornes B+ et B- du driver, puis on repique le 24 volts du Sabertooth :

- Jusqu'aux bornes 24V et 0V de l'automate, un pont a été fait entre l'API et ces modules pour les alimenter. Enfin, le module d'entrée gère et apporte l'alimentation au capteur inductif.

Puis, les moteurs sont alimentés grâce aux bornes M1A/B et M2A/B du driver. Ensuite, on raccorde le connecteur de 4 broches du driver à notre Max 232, nous sommes donc en 0.5 volts en sortie du Sabertooth et en +/- 10 volts en sortie de notre carte qui est reliée au Tx, Rx et à la masse du module RS 232 (750-652) de l'automate.



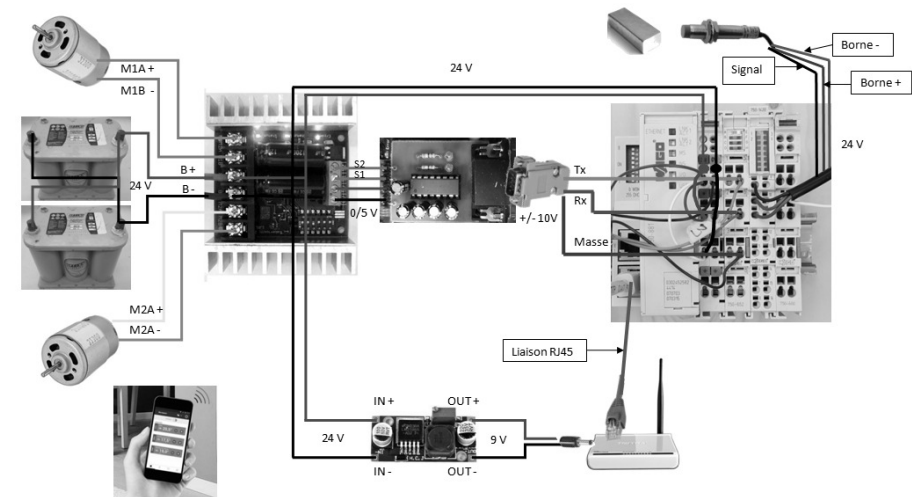
- Jusqu'aux bornes IN+ et IN- du régulateur de tension et celui-ci délivre le 9V au routeur wifi en sortant par les bornes OUT+ et OUT- qui seront raccordées au Tenda grâce à une prise jack. Enfin, une liaison par câble Ethernet entre l'automate et le routeur wifi est nécessaire pour permettre à une tablette ou un Smartphone sous Android de se connecter à l'automate et de pouvoir le contrôler à distance et ainsi influencer sur les moteurs.





GEII
Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Pour résumé, voici un synoptique avec tous les organes de notre robot:

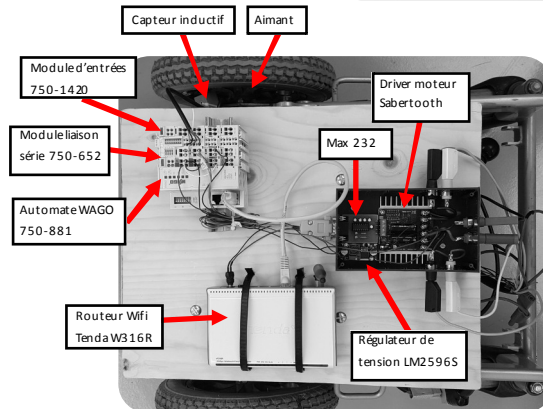
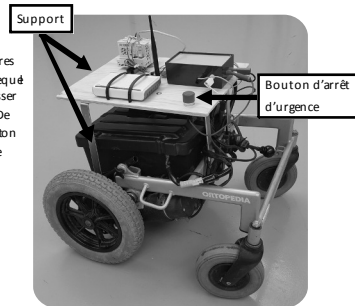




Après un peu de bricolage et de patience, nous avons assemblé proprement tous les composants du futur robot sur le châssis de manière à ce qu'ils soient fixés définitivement.

Voici le résultat final :

Nous avons créé un support à l'aide d'équerres en aluminium et d'une planche de bois sur lequel nous avons pu fixer tous les organes et les laisser accessibles pour différentes modifications. De plus, nous avons pris soin de mettre un bouton d'arrêt d'urgence qui permettra de couper le courant au niveau des batteries au cas où un problème venait à se produire.





6) Résultats et conclusions

Nous avons trouvé ce projet intéressant et complet, en effet nous avons pu associer de l'électronique, de l'électrotechnique et de la programmation.

De plus, nous avons mis en application les connaissances acquises durant notre année de Licence Professionnelle.

La totalité de notre projet a été conçue avec des composants de récupération, comme le châssis, les moteurs, les batteries, le driver moteur et le bouton d'arrêt d'urgence ou alors avec des éléments déjà existants au sein de l'IUT comme l'automate, les modules, le routeur Wifi et le régulateur de



tension. Nous avons dû faire un gros travail d'adaptation car nous nous sommes aperçus qu'il était parfois plus difficile et contraignant de travailler avec des matériaux usités plutôt que d'avoir recours à un choix de matériel neuf.



GEII

Département Génie Électrique
& Informatique Industrielle
IUT Belfort-Montbéliard

Une des améliorations pouvant être réalisée serait de mettre à en place des capteurs ultrasons pour permettre au robot de gérer les obstacles et de lui installer des panneaux solaires qui rechargeraient les batteries afin qu'il soit autonome à 100%.

Pour conclure, nous pouvons dire que ce projet a été une bonne expérience pour nous deux, il nous a permis de gérer un projet de longue durée de A à Z tout en travaillant en collaboration avec notre tuteur de projet.