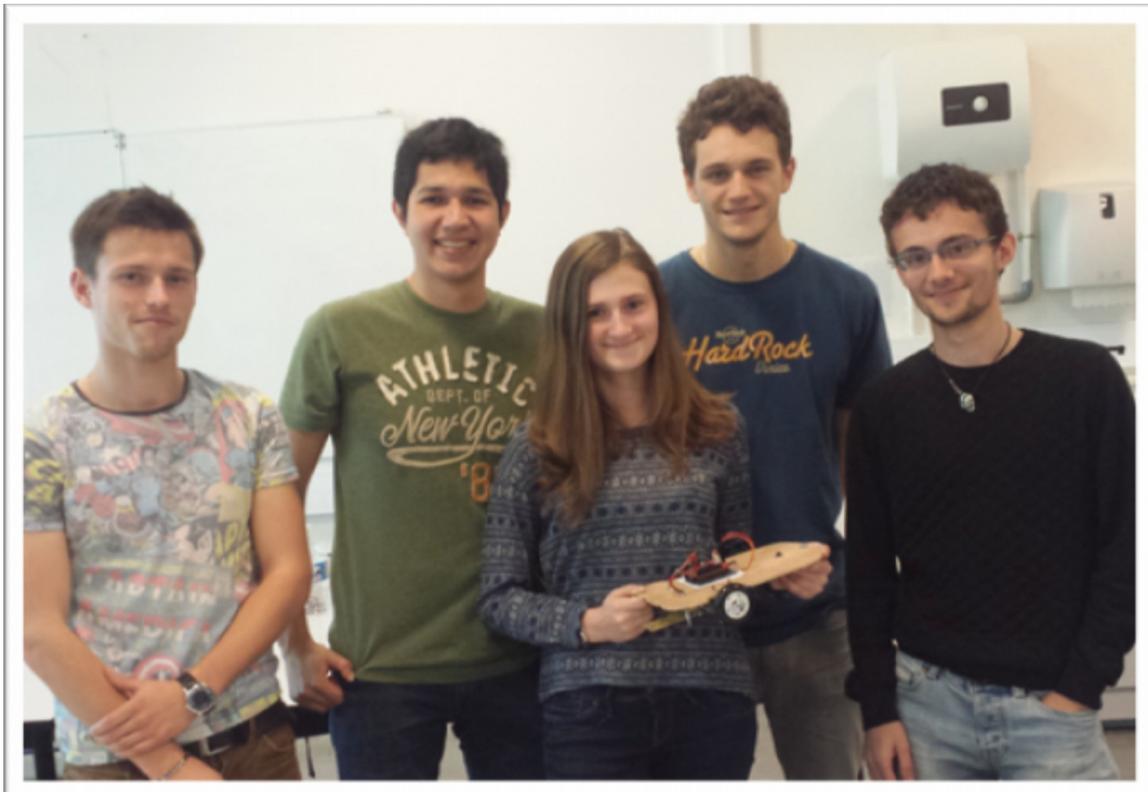


Projet de Physique P6
STPI/P6/2016 – 28

**Robot suiveur de ligne n°1 (pour course de vitesse
contre le robot suiveur de ligne n°2)**



Étudiants :

Guillaume BURLOT

Julien CASTELLANI

Romain JACQUIER

Ana MARCUSANU

Jorge OCHOA MAGANA

Enseignant-responsable du projet :

Fabrice DELAMARE

Date de remise du rapport : **13/06/2016**

Référence du projet : **STPI/P6/2016 – 28**

Intitulé du projet : Robot suiveur de ligne (pour course de vitesse contre le robot suiveur de ligne n°2)

Type de projet : **expérimental**

Objectifs du projet (10 lignes maxi) : Réaliser un robot suiveur de ligne, c'est-à-dire le concevoir (partie mécanique), le programmer (partie informatique) et régler les capteurs et l'alimentation (partie électronique).

Mots-clefs du projet (4 maxi) : **Robot**

Si existant, n° cahier de laboratoire associé :

TABLE DES MATIÈRES

1. Introduction.....	6
2. Méthodologie / Organisation du travail.....	7
3. Travail réalisé et résultats.....	8
3.1. Partie Mécanique.....	8
3.2. Partie électronique.....	10
3.3. Partie informatique.....	12
4. Conclusions et perspectives.....	15
4.1. Conclusion sur le travail réalisé.....	15
4.2. Opinions personnelles.....	15
4.3. Perspectives pour la poursuite de ce projet.....	16
5. Bibliographie.....	17
6. Annexes.....	18
6.1. Documentation technique.....	18
6.2. Listings des programmes réalisés.....	18
6.3. Conception Assistée par Ordinateur.....	18

NOTATIONS, ACRONYMES

- **Arduino** : ce terme peut désigner :
 - Carte Arduino : La carte Arduino est un circuit imprimé contenant un micro-contrôleur associé à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes.
 - Langage Arduino : C'est un langage de programmation basé sur le C permettant de programmer la carte Arduino.
 - Logiciel Arduino : logiciel qui permet d'écrire en langage Arduino et de téléverser un programme dans la carte.
- **Carte Shield ou motor-shield** : carte qui se branche sur la carte Arduino et qui permet de contrôler les moteurs.
- **TCRT5000** : Modèle des capteurs utilisés.

1. INTRODUCTION

Lors de notre quatrième et dernier semestre en STPI à l'INSA de Rouen, il nous est demandé de réaliser un projet de physique. Les objectifs sont divers : tout d'abord, ce projet nous amène à mettre en pratique nos connaissances scientifiques dans des domaines variés, travailler en équipe et en autonomie afin d'arriver à un résultat concret tout en respectant une contrainte de temps.

Dans ce cadre, notre projet consistait à concevoir intégralement de A à Z un robot suiveur de ligne. Notre appareil devra affronter le robot d'une deuxième équipe dans une course de vitesse sur un parcours déterminé. Ainsi, notre objectif était de construire un système performant en rassemblant nos connaissances, nos idées novatrices et nos diverses compétences techniques, tout en en acquérant de nouvelles. La robotique fait appel à plusieurs domaines scientifiques, et notre projet a requis des connaissances en mécanique, électronique, électricité et informatique. Nous devons nous répartir les tâches pour avancer de manière cohérente et efficace, tout en avançant collectivement et en restant à l'écoute.

Notre groupe se compose de cinq personnes, chacune ayant des aptitudes différentes, du fait de nos projets professionnels différents. Ainsi ce projet, qui plus est dans la robotique, nous a appris à organiser nos idées, notre temps et nos compétences, mais aussi à mettre à profit notre inventivité et notre cohésion.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Pour la réalisation de notre robot, nous avons dans un premier temps désigné Jorge comme leader et porte-parole du projet puis nous nous sommes séparés pour traiter les trois parties : mécanique, informatique et électronique. Guillaume et Julien se sont occupés de la partie mécanique en concevant le châssis et en y fixant les roues, les moteurs et les capteurs. Romain et Jorge ont réalisé le programme Arduino et Ana a effectué les soudures sur le motor shield et a vérifié que les capteurs étaient bien branchés. Les différentes équipes se mélangeaient souvent lors des tests et lorsqu'on rencontrait une difficulté.

Après chaque manipulation, nous réalisons des vérifications de notre travail. En électronique nous avons testé les capteurs et l'alimentation ; en mécanique nous nous sommes assurés du bon équilibre des pièces montées et de l'exactitude des perçages ; et enfin en informatique nous avons fait des tests unitaires pour nous assurer du bon fonctionnement du programme à chaque étape.

Chaque séance de travail commençait par une mise au point pour rappeler les objectifs de celle-ci, et se finissait par une réunion pour savoir ce qui avait été fait. De plus, après chaque séance, Jorge nous envoyait un mail pour résumer ce qui avait été réalisé.

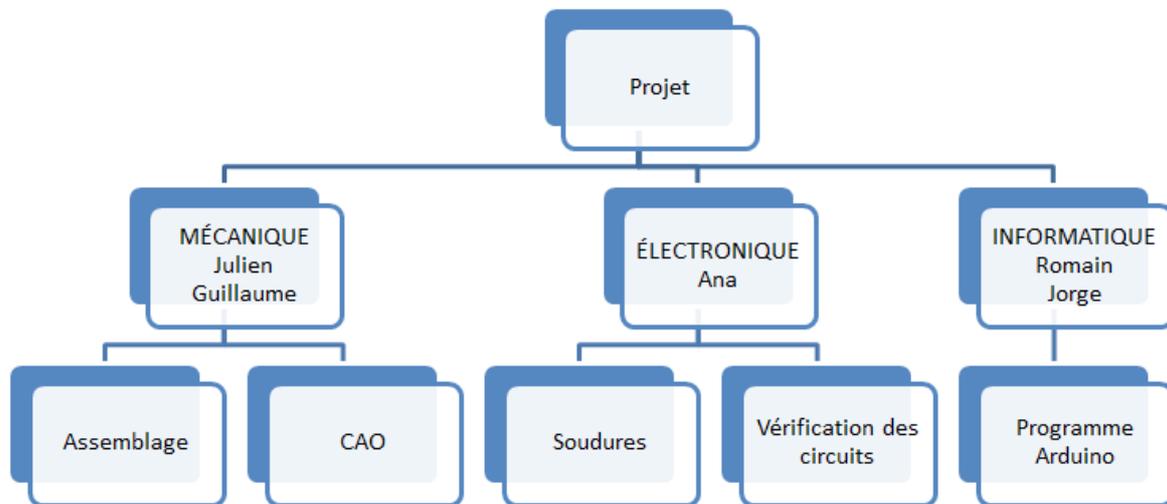


Illustration 1: Organigramme de l'organisation du travail.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. Partie Mécanique

3.1.1. Objectifs

Concevoir la forme du robot et choisir ses matériaux, fabriquer la base et fixer dessus les roues, les moteurs, les capteurs ainsi que les tous les composants liés à l'alimentation et au design.

3.1.2. Conception du robot

Après avoir fait le choix du matériau (type de bois, contreplaqué), nous avons effectué une série de mesures afin d'optimiser la taille, le poids, et la forme de notre robot. Nous avons choisi de placer le châssis au-dessus des ensembles roue/moteur.

Nous avons découpé des espaces pour le renforcement des roues, afin d'optimiser la distance entre les moteurs. Puis nous avons percé la base et fixé les moteurs à l'aide de boulons. Nous avons ensuite mis en place un troisième appui à l'aide d'une fixation en forme de demi-cercle, d'une longue vis et d'une « ball-caster ». Nous avons mis au point un système d'écrous simple afin de pouvoir régler la hauteur de l'arrière de la base et maintenir l'horizontalité.

La carte de capteurs a été placée en-dessous du châssis, vers l'avant et entre les moteurs, stabilisée par deux vis verticales, réglables également grâce à un simple système d'écrous, afin de respecter la distance au sol de 2,5 mm. La carte Arduino et le motor-shield ont été finalement fixés sur le dessus avec des points de colle, disposant chacun d'une ouverture dans le châssis obtenue à l'aide d'une perceuse à colonne, pour relier les fils jusqu'aux moteurs et capteurs.

Tout les bords du châssis ont été arrondis et limés, surtout pour des raisons de sécurité ; en effet, nous avons eu a plusieurs reprises des échardes lors de la manipulation.

Un modèle 3D du robot a été conçu avec Solidworks au fur et à mesure de l'avancée du robot.



Illustration 2: Apparence finale du robot.

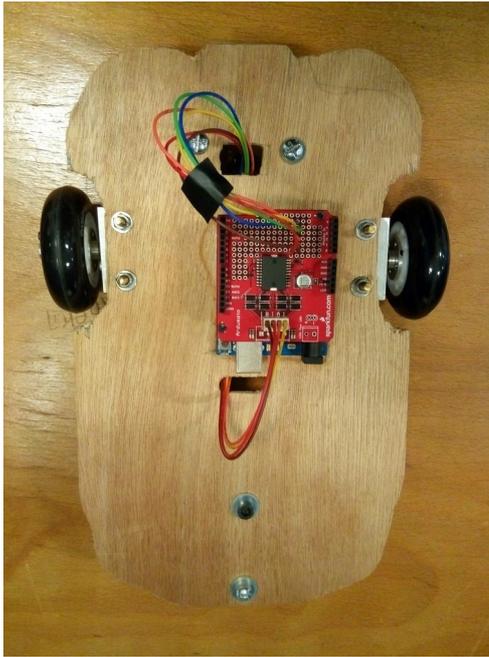


Illustration 3: Photo de la base vue de dessus.

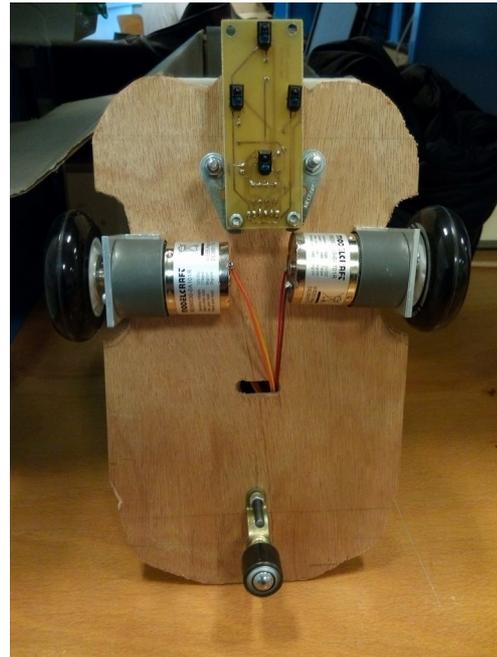


Illustration 4: Photo de la base vue de dessous.

3.1.3. Difficultés rencontrées

Les principales difficultés que nous avons rencontrées ont été l'utilisation d'outils à usage manuel (notamment lors de la découpe de profils arrondis avec la perceuse à colonne) mais aussi les soudures qui se cassaient régulièrement, retardaient les tests et représentaient une perte de temps. De plus, nous devions être très précis dans les prises de mesure, afin de satisfaire l'équilibre global du robot. Lors de la recherche des solutions techniques et des matériaux, nous avons tenu compte du démontage fréquent de diverses pièces au fur et à mesure de l'avancée du projet en fixant des ensembles réglables, afin de pouvoir modifier certains composants si besoin.

3.2. Partie électronique

3.2.1. Objectifs

L'objectif de cette partie électronique était de s'occuper du réglage des capteurs et de l'alimentation.

3.2.2. Capteurs

Nous avons reçu une carte électronique déjà prête avec quatre capteurs TCRT5000¹, un condensateur et des résistances, faite par les élèves des années précédentes. Nous avons dû tout de même vérifier que les branchements étaient bien réalisés, puis nous avons testé cette carte afin de vérifier le bon fonctionnement des capteurs, mais également pour modifier les entrées dans notre programme en identifiant chaque capteur. Une fois les tests concluants, nous avons pu fixer les capteurs à l'avant du robot, en respectant la distance au sol.

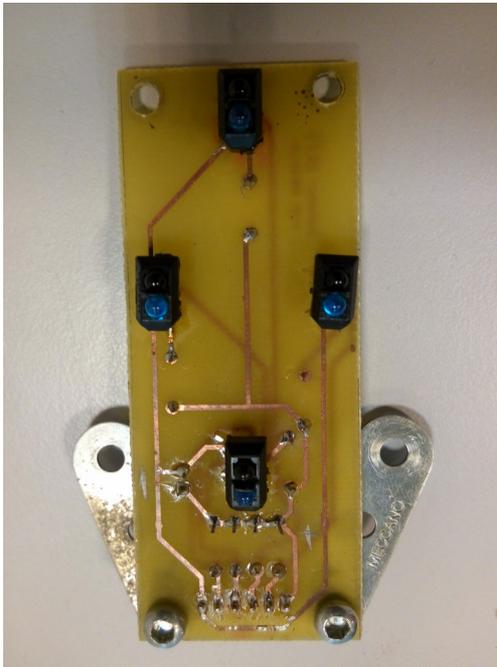


Illustration 6: Photo de la carte électronique, face avant (capteurs).

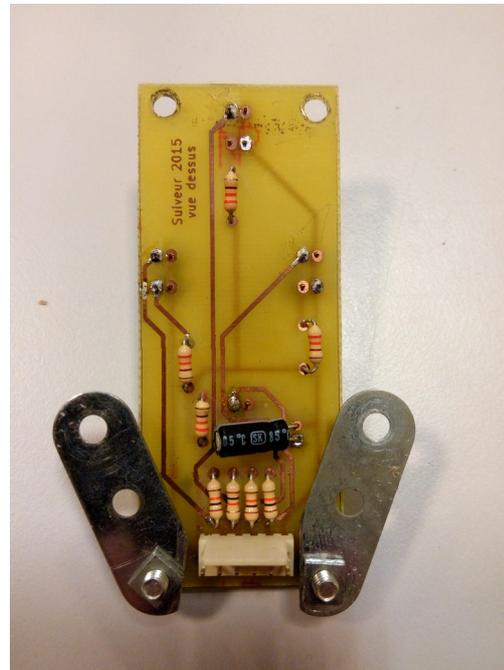


Illustration 5: Photo de la carte électronique, face arrière (résistances et condensateur).

3.2.3. Motor shield

Le motor-shield que nous avons reçu au début était défectueux, donc nous en avons pris un autre et ressoudé tous les câbles, ce qui nous a pris beaucoup de temps car les soudures à faire étaient petites et ce travail devait être minutieux. De plus, les soudures se cassaient souvent et nous devions les refaire régulièrement.

1 Cf fiche technique en annexe.

3.2.4. Alimentation

Pour l'alimentation de notre robot, nous avons reçu un pack de 6 piles et un interrupteur. Dans un premier temps, il a fallu tester les piles et l'interrupteur pour vérifier que l'ensemble fonctionnait correctement. Puis il a fallu relier cette ensemble aux moteurs et aux capteurs en respectant les bornes. Cependant, nous devons partager le même pack de piles avec l'autre groupe (pour des raisons d'équité) ainsi la majorité de nos tests ont été faits avec un câble branché sur le secteur.

3.2.5. Difficultés rencontrées

La principale difficulté que nous avons rencontrée dans cette partie a été de comprendre le fonctionnement de capteurs que nous n'avions jamais utilisé auparavant, ce qui a constitué pour nous un véritable casse-tête, durant toute une séance. De plus, nous avons perdu du temps à cause d'un faux contact dont il a fallu trouver la source puis changer des fils et refaire des soudures.

3.3. Partie informatique

3.3.1. Objectifs

Pour l'aspect informatique, nous devons réaliser le programme Arduino dont le but est de piloter l'appareil. Ce programme représente le cerveau du robot. L'objectif principal dans cette partie est de concevoir un programme dans un nouveau langage de programmation qui nous était inconnu. Une fois ce programme conçu, il fallait l'optimiser afin que le robot parcoure le circuit le plus rapidement possible.

3.3.2. Premiers pas avec l'Arduino

Avant de commencer à coder le programme principal du robot, il nous a fallu appréhender le langage Arduino. En effet, pour que les programmes soit exécutables par la carte Arduino, il faut respecter une syntaxe précise. Le programme commence par une fonction *setup* qui permet d'initialiser les variables et définir les entrées et sorties que la carte utilisera. Ensuite, on trouve la partie *loop* où on écrit le programme principal avec les instructions à suivre par le robot. Comme le montre l'image ci-dessous :

```

sketch_jun04a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

Illustration 7: Environnement de programmation Arduino.

Pour mieux connaître ce langage, nous avons commencé par faire des petits programmes (des morceaux simples de code, tels que le robot avance tout droit, tourne à gauche...). Nous avons ensuite regroupé ces programmes en procédures et fonctions que nous avons réutilisées dans notre programme principal.

3.3.3. Programme Principal²

Pour atteindre notre objectif, nous avons tenté de réaliser un programme simple mais efficace. En effet, dans notre idée de départ, le robot devait suivre la ligne, en faisant des zigzags. Autrement dit, le principe de guidage du robot est en fait un ré-ajustage permanent de sa trajectoire, grâce aux capteurs latéraux qui corrigent sans cesse la trajectoire en le

2 Cf code en annexe.

rabattant sur sa ligne. Ainsi, le robot ne risquerait pas de continuer tout droit en présence d'un virage.

Pour mieux expliquer notre idée, nous avons besoin du schéma suivant :

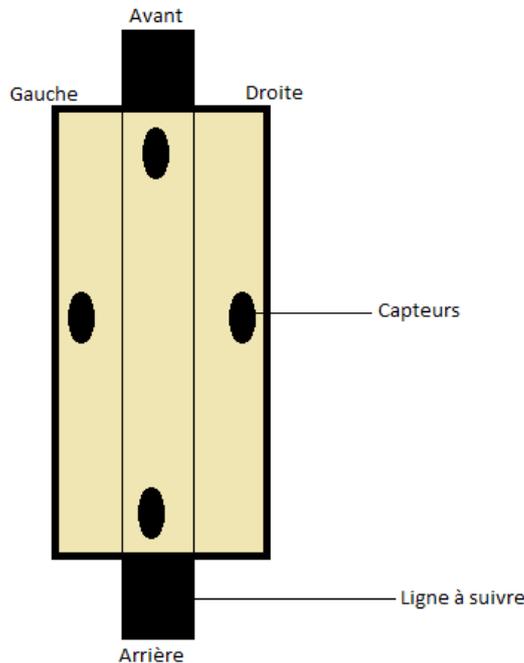


Illustration 8: Schéma de la position des capteurs.

Ainsi, tant que le capteur arrière reste sur la ligne noire, le robot fait des zigzags en faisant intervenir les capteurs de gauche et de droite : la position du capteur avant est donc maintenue au-dessus de la ligne. Notre robot doit suivre la ligne à tout moment. Si le capteur arrière ne capte plus la ligne noire, le robot continue à exécuter l'instruction qu'il réalisait avant de perdre ce signal. Après avoir fait la conception préliminaire du programme, nous nous sommes attaqués à l'implémentation.

Cependant, nous nous sommes rendus compte que le problème était plus complexe. En effet, avec cette première idée le robot suivait bien la trajectoire lorsque celle-ci était rectiligne mais lors des virages, il la perdait. C'est pour cette raison qu'après plusieurs essais d'optimisation de cette première version, nous avons décidé de revenir au début de la conception. Ainsi, nous avons abandonné notre première idée, afin d'obtenir un programme qui traite tous les cas possibles de positions des capteurs par rapport à la ligne. Ainsi, nous avons défini 10 cas possibles avec une instruction différente pour chacun. Avec cette nouvelle configuration, notre robot peut désormais tourner à différentes vitesses selon l'angle du virage, il peut aller tout droit lorsque la ligne est droite et il est capable cette fois-ci de retrouver la ligne et de continuer le parcours en cas d'erreur. Nous avons donc implémenté ce programme et nous l'avons testé. Dès le premier essai, le robot a réussi à finir le parcours mais il était trop lent. Nous l'avons donc optimisé en modifiant sa vitesse de déplacement. Finalement, après avoir défini la vitesse optimale, le robot a réussi à faire le parcours en 12 secondes.

3.3.4. Difficultés rencontrées

Lors de la réalisation du code, nous avons connu plusieurs difficultés. Tout d'abord, le langage Arduino est un langage qui nous était inconnu et que nous avons dû apprendre lors des premières séances. En effet, malgré sa ressemblance avec le langage C, nous n'avions jamais manipulé dans nos programmes des entrées et sorties numériques ou analogiques.

Nous avons donc pris du temps pour comprendre comment faire les configurations nécessaires, afin que notre programme récupère et transmette les bonnes informations.

De plus, nous devons attendre pour tester la première version du programme, en raison d'un changement de carte Shield : tous les composants électroniques devaient être soudés par l'équipe « électronique ».

Aussi, nous avons eu des difficultés pour tester nos programmes. En effet, les valeurs renvoyées par les capteurs changeaient de semaine en semaine : notre programme pouvait fonctionner partiellement lors d'une séance, puis ne plus fonctionner du tout lors de la séance suivante.

D'autre part, le faux contact sur le branchement d'un câble a retardé notre avancement. Il nous a fallu alors identifier le problème et le résoudre le plus rapidement possible.

Finalement, nous avons eu des difficultés pour mettre les bonnes valeurs de vitesse pour le robot. En effet, si le robot allait trop vite il n'avait pas le temps de capter la ligne et de changer de direction.

4. CONCLUSIONS ET PERSPECTIVES

4.1. Conclusion sur le travail réalisé

Ainsi, nous sommes parvenus à réaliser notre robot suiveur de ligne malgré tous les différents problèmes que nous avons pu rencontrer. Nous avons réussi à concilier les aspects mécanique, électronique et informatique pour aboutir à un robot fonctionnel et assez performant. En effet, notre robot réalise un tour de circuit en moins de 20 secondes (12 au dernier test). Nous avons tout de même pris du retard au cours du semestre ce qui nous a laissé peu de temps pour optimiser notre programme et décorer notre robot. Néanmoins, nous sommes fiers du résultat et nous estimons avoir atteint nos objectifs mêmes si des améliorations auraient pu être apportées (cela n'a pas été fait par manque de temps). Enfin, cela nous a permis de découvrir le domaine de la robotique, de travailler en équipe et d'apprendre à travailler de façon autonome en trouvant par nous-mêmes des solutions à chaque nouveau problème rencontré.

4.2. Opinions personnelles

4.2.1. Ana

J'ai choisi ce projet de robotique car je n'avais jamais fait de robot auparavant ou même seulement étudié ce domaine. Je me suis donc lancée dans ce projet par curiosité et aussi parce que j'avais toujours trouvé ce domaine fascinant sans pourtant y avoir jamais touché. Durant les premières séances, je me suis sentie un peu perdue mais au fur et à mesure, grâce à toute l'équipe, notre projet s'est concrétisé et j'ai commencé à vraiment aimer ce que nous faisons. Même si nous rencontrons de nouveaux problèmes à chaque séance, la satisfaction et la fierté que je ressentais à chaque nouvelle avancée et nouvelle solution m'ont beaucoup plu. Je pense que ce projet m'a rapproché un peu plus du métier d'ingénieur en découvrant concrètement cette méthode de travail qui est de trouver une solution à chaque nouveau problème rencontré, que ce soit d'ordre mécanique, électronique ou informatique. Bien que j'ai été responsable de la partie électronique, j'ai beaucoup suivi le travail des autres équipes pour bien comprendre le fonctionnement global de notre robot. Ce projet a été pour moi une véritable découverte qui m'a permis de gagner en autonomie et en assurance, j'en garde un souvenir positif.

4.2.2. Guillaume

Pour commencer, j'ai choisi ce projet car j'ai eu l'occasion de travailler sur un robot au collège sans réellement savoir ce que je faisais, mais j'avais trouvé très intéressante l'idée de créer un robot autonome, j'ai donc pensé que ce projet pourrait être très intéressant et très formateur. Nous nous sommes très vite rendu compte de la complexité de la conception complète d'un tel robot. Pour ma part, j'ai été chargé de la partie « mécanique » du projet, c'est à dire la conception de la base, et la résolution des différentes contraintes techniques (fixation de la « ball-caster », des capteurs), ainsi que de la CAO. Cependant, j'ai tout de même suivi d'assez près les autres parties du projet et j'ai beaucoup appris, notamment concernant la programmation de la carte Arduino, domaine dans lequel je ne connaissais absolument rien. Pour revenir à ma participation au projet, j'ai beaucoup aimé le fait de pouvoir travailler de façon autonome avec les outils à notre disposition, ainsi que le travail du bois. J'ai également pu mettre en application mes cours de CTI3 (CAO) dans la conception

du modèle 3D du robot sous Solidworks. Pour finir, ce projet m'a énormément apporté par le travail de groupe, l'organisation, et l'entraide qu'il a nécessité. C'est également une très grande satisfaction de voir aboutir un tel projet.

4.2.3. Jorge

Tout d'abord, j'ai été très content de pouvoir réaliser ce projet. En effet, la robotique est un domaine tellement intéressant et qui m'intriguait beaucoup d'autant plus que je n'avais jamais eu l'opportunité de travailler dans ce domaine là. Cette expérience était une découverte de la robotique que j'ai beaucoup appréciée. Le fait de voir le robot prendre forme au cours des semaines est ce que j'ai le plus apprécié. En effet, nous avons construit de A à Z un robot qui fonctionnait. De plus, non seulement cette expérience m'a permis de mettre en œuvre les connaissances théoriques acquises tout au long de STPI, mais aussi elle m'a apporté de nouvelles connaissances dans différents domaines. En effet, même si je devais m'occuper de la partie informatique avec Romain, je suis resté polyvalent et j'ai pu travailler dans les autres domaines. Personnellement, je pense que ce projet m'a permis de gagner en autonomie mais aussi en responsabilité. En effet, en tant que chef d'équipe je devais coordonner notre équipe afin de finir notre projet tout en respectant les délais qui nous étaient imposés au début. Finalement, je retiens de ce projet qu'il faut persévérer et ne jamais abandonner. En effet, chaque échec nous permet d'avancer et de se rapprocher du succès.

4.2.4. Julien

Ce projet, que je trouvais ambitieux au départ, m'a appris à mêler plusieurs domaines scientifiques et leurs contraintes associées, au sein d'un seul et même sujet. Travailler sur la partie mécanique, tout en réfléchissant aux problématiques électroniques, et surtout tenir compte des exigences de la partie informatique : quand il s'agissait de mettre en commun nos travaux respectifs et nos nombreuses idées, cela n'était pas une mince affaire. Mais grâce à une organisation bien menée et aux nombreuses discussions avec le groupe, nos choix étaient clairs, on pouvait avancer et je pense que cela nous donnait confiance et une certaine assurance. Le travail collectif a été selon moi primordial, car définir des objectifs pour chacun, avoir une date à respecter, a été très formateur et m'a permis de gagner en efficacité. De plus, j'ai particulièrement apprécié le travail manuel à fournir pour construire la structure du robot et l'agencement des pièces. En effet, on était complètement autonome en ce qui concerne l'utilisation des outils et des machines, le choix des matériaux, etc, et ce fut très enrichissant, d'autant plus que nous étudions beaucoup plus de matières théoriques en cycle STPI. Finalement, ce projet m'a satisfait grâce à la polyvalence qu'implique le secteur de la robotique, que j'ai découvert, et m'a donné un aperçu du travail considérable et rigoureux qu'il faut effectuer pour fabriquer une telle machine. Ainsi, ce projet m'a apporté de l'expérience et une vision globale pour la conception de notre robot, dont nous avons pu voir l'évolution chaque semaine et dont je suis fier d'avoir pu contribuer à l'élaboration.

4.2.5. Romain

J'ai énormément apprécié travailler dans ce projet. Travailler avec d'autres personnes dans un même but fut très enrichissant. J'ai notamment appris à travailler dans un domaine que je connaissais à peine: la robotique. Élaborer un robot fut un défi dont j'ai été heureux de relever. Ce fut très intéressant de travailler sur un projet où plusieurs domaines scientifiques sont regroupés: électronique, mécanique, informatique. Ce fut donc un moyen de concrétiser beaucoup de connaissances théoriques apprises en STPI, même si ma principale occupation dans ce projet a été la réalisation du programme. D'autre part, ce projet m'a été extrêmement formateur. Comme dit précédemment, j'ai appris à m'organiser avec d'autres personnes que je ne connaissais pas au début et qui ont d'autres idées. Le travail collectif a été le plus important ici, pour mettre en commun nos différents travaux et

pour que notre avancement soit le plus cohérent. Ensuite, nous avons construit le robot de manière entièrement autonome, pour le choix de la forme du robot, des pièces, du programme : cela m'a donc permis de gagner en efficacité. Finalement, la conception du programme a été un réel défi, que ce soit au niveau de la conception préliminaire, tout comme le codage lui-même. Pour conclure, ce projet a été une expérience extrêmement enrichissante et formatrice.

4.3. Perspectives pour la poursuite de ce projet

À la fin de ce projet, nous avons réalisé un robot fonctionnel. Toutefois, il y a des points que nous aurions aimé améliorer si on avait eu plus de temps. Tout d'abord, nous considérons que le programme informatique pourrait être perfectionné. En effet, si la vitesse est trop grande, notre robot perd encore la ligne dans les virages trop serrés. Nous aimerions donc pouvoir le faire tourner rapidement même si les virages sont en épingle. De plus, nous aurions aimé fixer les cartes au châssis de manière plus stable, ainsi qu'améliorer le décor, notamment la tête de la tortue. Si le temps nous le permettait, nous aurions également implémenté un contrôle de démarrage manuel.

5. BIBLIOGRAPHIE

- (1) [http://www.bricobidules.com/index.php?post/2012/02/21/Un-suiveur-de-lignes-\(encore-avec-Arduino\)](http://www.bricobidules.com/index.php?post/2012/02/21/Un-suiveur-de-lignes-(encore-avec-Arduino)) (valide à la date du 25/04/2016).
- (2) <https://www.arduino.cc/> (valide à la date du 04/06/2016).
- (3) « Initiation Arduino », extraits du manuel floss : <http://fr.flossmanuals.net/arduino/index> (valide à la date du 04/06/2016).
- (4) <https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino> (valide à la date du 010/06/2016).

6. ANNEXES

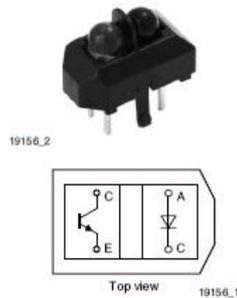
6.1. Documentation technique



TCRT5000, TCRT5000L

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output



FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within > 20 % relative collector current: 0.2 mm to 15 mm
- Typical output current under test: $I_C = 1$ mA
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



RoHS
COMPLIANT

DESCRIPTION

The TCRT5000 and TCRT5000L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting dips. TCRT5000L is the long lead version.

APPLICATIONS

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

PRODUCT SUMMARY				
PART NUMBER	DISTANCE FOR MAXIMUM CTR _{rel} ⁽¹⁾ (mm)	DISTANCE RANGE FOR RELATIVE I _{out} > 20 % (mm)	TYPICAL OUTPUT CURRENT UNDER TEST ⁽²⁾ (mA)	DAYLIGHT BLOCKING FILTER INTEGRATED
TCRT5000	2.5	0.2 to 15	1	Yes
TCRT5000L	2.5	0.2 to 15	1	Yes

Notes

- ⁽¹⁾ CTR: current transfere ratio, I_{out}/I_n
⁽²⁾ Conditions like in table basic characteristics/sensors

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	VOLUME ⁽¹⁾	REMARKS
TCRT5000	Tube	MOQ: 4500 pcs, 50 pcs/tube	3.5 mm lead length
TCRT5000L	Tube	MOQ: 2400 pcs, 48 pcs/tube	15 mm lead length

Note

- ⁽¹⁾ MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT (EMITTER)				
Reverse voltage		V _R	5	V
Forward current		I _F	60	mA
Forward surge current	t _p ≤ 10 μs	I _{FSM}	3	A
Power dissipation	T _{amb} ≤ 25 °C	P _V	100	mW
Junction temperature		T _J	100	°C

6.2. Listings des programmes réalisés

```

1  const byte pwm_a = 3; //PWM control for motor outputs 1 and 2 is on
    digital pin 3
    const byte pwm_b = 11; //PWM control for motor outputs 3 and 4 is
    on digital pin 11
    const byte dir_a = 12; //direction control for motor outputs 1 and
    2 is on digital pin 12
    const byte dir_b = 13; //direction control for motor outputs 3 and
    4 is on digital pin 13

6  //Declaration capteurs
    const byte capteur_avant = 3;
    const byte capteur_arriere = 2;
    const byte capteur_gauche = 1;
    const byte capteur_droite = 0;

11  #define aGauche 1
    #define aDroite 0

16  void setup()
    {
        //setup des moteurs en output
        pinMode(pwm_a, OUIPUT); //Set control pins to be outputs
        pinMode(pwm_b, OUIPUT);
21  pinMode(dir_a, OUIPUT);
        pinMode(dir_b, OUIPUT);

        //setup des capteurs en input
        pinMode(capteur_avant, INPUT);
26  pinMode(capteur_arriere, INPUT);
        pinMode(capteur_gauche, INPUT);
        pinMode(capteur_droite, INPUT);

        //setup de la direction des moteurs
31  digitalWrite(pwm_a, LOW);
        digitalWrite(pwm_b, LOW);
        digitalWrite(dir_a, LOW);
        digitalWrite(dir_b, LOW);

36  Serial.begin(9600);
    }

    //Programme principal
    void loop() {
41  int i, ratio1, ratio2;
        ratio1=0.1; //ratio qui multiplie les vitesses
        ratio2=0,1;
    }

```

```

46   if (capteurNoir(capteur_avant) and capteurNoir(capteur_gauche)
      and capteurNoir(capteur_arriere) and (not capteurNoir(
      capteur_droite))){
      Tourner(aGauche, ratio1);
    }
    if (capteurNoir(capteur_avant) and capteurNoir(capteur_droite)
      and capteurNoir(capteur_arriere) and (not capteurNoir(
      capteur_gauche))){
51     Tourner(aDroite, ratio1);
    }
    if (not(capteurNoir(capteur_avant)) and (not capteurNoir(
      capteur_gauche)) and capteurNoir(capteur_arriere) and
      capteurNoir(capteur_droite)){
56     Tourner(aDroite, ratio1);
    }
    if (not(capteurNoir(capteur_avant)) and (not capteurNoir(
      capteur_droite)) and capteurNoir(capteur_arriere) and
      capteurNoir(capteur_gauche)){
61     Tourner(aGauche, ratio1);
    }
    if (capteurNoir(capteur_avant) and (not capteurNoir(
      capteur_droite)) and capteurNoir(capteur_arriere) and not (
      capteurNoir(capteur_gauche))){
      AvancerToutDroit ();
    }
    if (capteurNoir(capteur_avant) and (not capteurNoir(
      capteur_droite)) and not(capteurNoir(capteur_arriere)) and not
      (capteurNoir(capteur_gauche))){
66     //avancer tout droit lentement
      digitalWrite(dir_a, HIGH);
      analogWrite(pwm_a, 80);
      digitalWrite(dir_b, LOW);
      analogWrite(pwm_b, 80);
71     }
    if (not(capteurNoir(capteur_avant)) and (not capteurNoir(
      capteur_droite)) and not(capteurNoir(capteur_arriere)) and
      capteurNoir(capteur_gauche)){
      Tourner(aGauche, ratio2);
    }
    if (not(capteurNoir(capteur_avant)) and capteurNoir(capteur_droite)
      ) and not(capteurNoir(capteur_arriere)) and not(capteurNoir(
      capteur_gauche))){
76     Tourner(aDroite, ratio2);
  
```

```

    if (capteurNoir(capteur_avant) and not(capteurNoir(capteur_droite)
        ) and not(capteurNoir(capteur_arriere)) and capteurNoir(
            capteur_gauche)){
        AvancerToutDroit ();
    }
81  if (capteurNoir(capteur_avant) and not(capteurNoir(capteur_gauche)
        ) and not(capteurNoir(capteur_arriere)) and capteurNoir(
            capteur_droite)){
        AvancerToutDroit ();
    }
    }

86

boolean capteurNoir (byte sensor)
{ // fonction qui renvoie Vrai si le capteur capteur = noir
91  return (analogRead(sensor) > 500);
}

96 void ArretBrut ()
    { //arrete le robot
        digitalWrite(dir_a, HIGH);
        analogWrite(pwm_a, 0);
        digitalWrite(dir_b, LOW);
101  analogWrite(pwm_b, 0);
    }

    void AvancerToutDroit ()
    { //fait le robot avancer tout droit
106  digitalWrite(dir_a, HIGH);
        analogWrite(pwm_a, 100);
        digitalWrite(dir_b, LOW);
        analogWrite(pwm_b, 100);
    }
111

    void Tourner (byte dir, float ratio)
    { //permet de tourner soit a droite soit a gauche

116  float vitesse1, vitesse2;

        vitesse2= 250; //vitesse globale
        vitesse1 = vitesse2*ratio ; //vitesse secondaire pour les virages

121  if (dir == aGauche)
        {
            digitalWrite(dir_a, HIGH);
            digitalWrite(dir_b, LOW);
126  analogWrite(pwm_a, vitesse2);
            analogWrite(pwm_b, vitesse1);
        }
        else if (dir == aDroite)
        {
131  digitalWrite(dir_a, HIGH);
            digitalWrite(dir_b, LOW);
            analogWrite(pwm_b, vitesse2);
            analogWrite(pwm_a, vitesse1);
        }
136 }

```

6.3. Conception Assistée par Ordinateur

