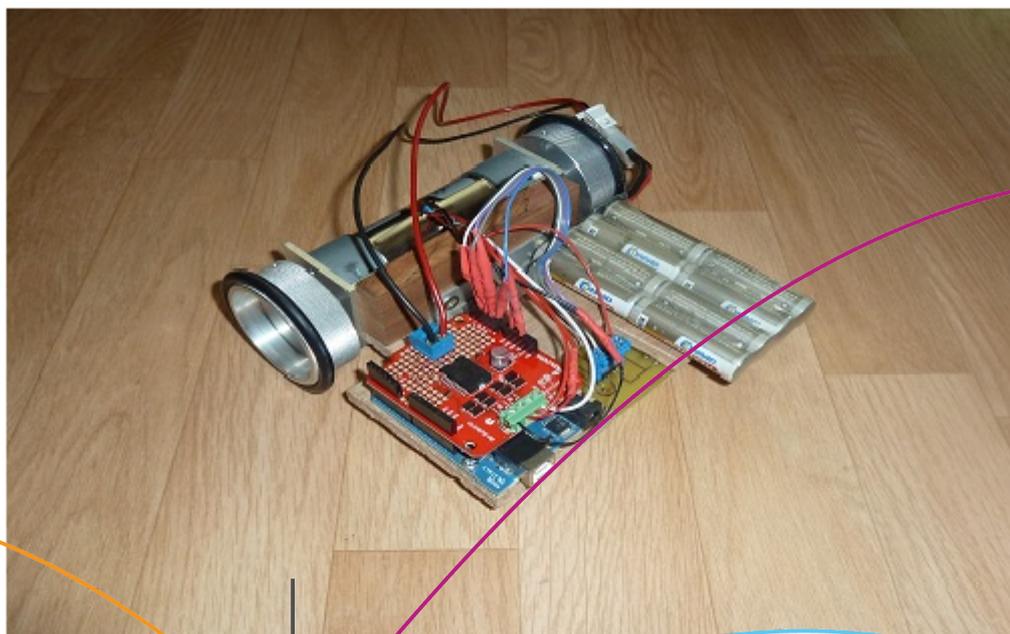


Robot Suiveur De Ligne



Enseignant responsable
Fabrice DELAMARE

Étudiants :

Jean ALLART
Quocduy NGUYEN
Hélène SOUDRY

Réda DRISSI
Damien POINTIN
Simon WALLON

Date de remise du rapport : 17/06/2013

Référence du projet : STPI¹/P6/2013 - 003

Intitulé du projet : Robot suiveur de ligne n°1 (pour course de vitesse contre le robot suiveur n°2)

Type de projet : *experimental*

Objectifs du projet :

Le but de ce projet est de s'initier à la robotique en réalisant un robot suiveur de ligne. Deux groupes conçoivent le même robot et s'affronte à la fin du projet. Les règles à respecter et les contraintes sont celles de la coupe de France de robotique des Grandes Écoles.

Table des matières

Notations	4
Introduction	5
1 Méthodologie, organisation du travail	6
1.1 Répartition des tâches	6
1.2 Organigramme	6
2 Travail réalisé et résultats	7
2.1 Partie mécanique	7
2.1.1 Conception sous SolidWorks	7
2.1.2 Réalisation	7
2.2 Partie électronique	9
2.2.1 Les capteurs	9
2.2.2 Conception de la carte	9
2.3 Partie informatique	11
2.3.1 Le logiciel et la carte Arduino	11
2.3.2 L'algorithme ²	11
2.4 Etat de l'art	13
2.4.1 Le TEOR ³	13
2.4.2 Les véhicules à guidage automatique (VGA) ⁴	13
2.4.3 Les jeux pour enfants	14
2.4.4 Autres exemples	15
Conclusion et perspectives	16
A Réalisation SolidWorks	18
B Schéma de la carte électronique	19
C Plan de la carte électronique	20
D Code	22

2. voir Annexe D

3. Source : Wikipédia « TEOR »

4. Source : Wikipedia « Véhicule à guidage automatique

Notations et Acronymes

Arduino : désigne à la fois :

- La carte Arduino : circuit imprimé qui contient le microcontrôleur que nous avons programmé
- Le langage Arduino : langage basé sur le C qui permet de programmer une carte Arduino
- Le logiciel Arduino : logiciel qui permet d'écrire avec le langage Arduino, mais surtout de téléverser le programme sur la carte

CAO : Conception Assistée par Ordinateur

Kicad : Logiciel utilisé pour la réalisation de la carte électronique

TCRT5000 : Nom des capteurs optiques utilisés pour le robot

Introduction

Le but de ce projet était - comme son nom l'indique - de réaliser un robot capable de suivre une ligne qu'elle soit droite ou courbe. La mise en œuvre pratique était que notre robot puisse participer à une course contre le robot n°2, réalisé par un autre groupe (et la gagner dans l'idéal!).

Notre groupe était composé de six personnes de profil différents mais tous déterminés à mener à bien ce projet. Nous avons donc décidé de nous répartir le travail en trois groupes de deux avec des tâches bien définies. Nous avons beaucoup à apprendre pour réaliser ce projet, nos connaissances dans les divers domaines nécessaires étant très limitées.

Ce projet de physique P6 s'inscrit dans le cadre de l'enseignement dispensé aux élèves ingénieurs. Comme mentionné ci-dessus, nous avons eu beaucoup à apprendre et à mettre en pratique pour ce projet, ce qui a pour but de nous familiariser avec une situation concrète que nous serons amenés à rencontrer régulièrement dans notre futur métier.

La principale contrainte à laquelle nous avons dû faire face était la taille du robot et la gestion du temps. En effet, celui-ci devait respecter les dimensions mentionnées dans le cahier des charges du championnat de France de robotique des Grandes Écoles.

Pour concevoir notre robot, M Delamare nous a fourni différents éléments tels que les moteurs, les roues ou le shield et la carte Arduino sur laquelle nous avons programmé le robot.

Chapitre 1

Méthodologie, organisation du travail

1.1 Répartition des tâches

Comme expliqué dans l'introduction, nous nous sommes réparti le travail en trois groupe de deux personnes. En effet, la conception du robot peut être divisée en trois grande parties : la mécanique, l'électronique et la programmation. De cette façon, chaque groupes pouvait travailler de manière plutôt indépendante vis à vis des deux autres. Les groupes étaient les suivant :

<i>Mécanique</i>	<i>Electronique</i>	<i>Programmation</i>
Hélène et Réda	Kévin et Damien	Simon et Jean

1.2 Organigramme

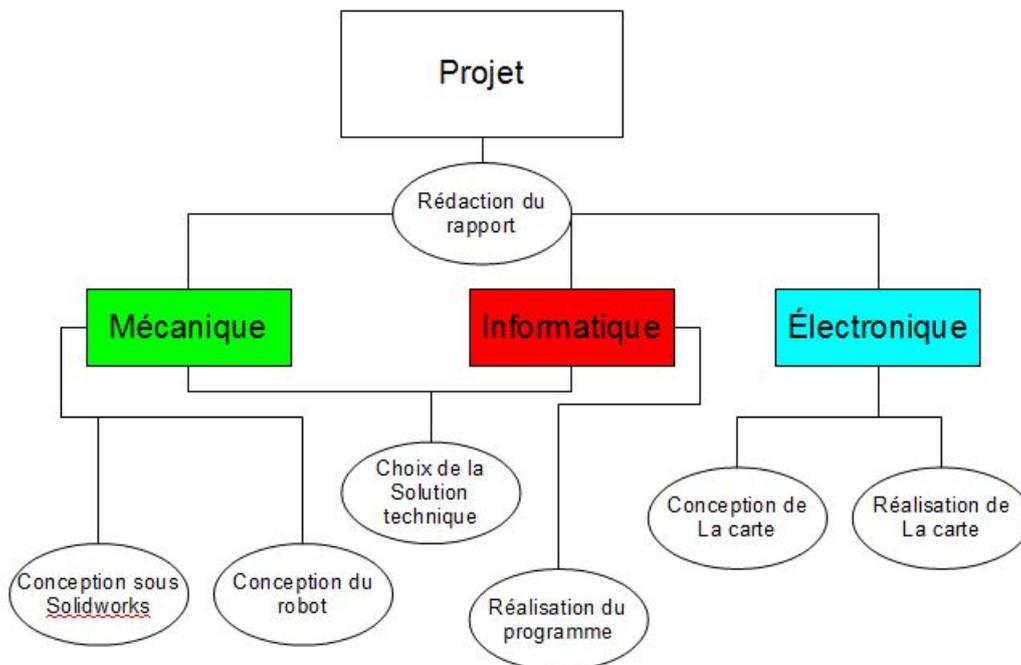


FIGURE 1.1 – Organigramme

Chapitre 2

Travail réalisé et résultats

2.1 Partie mécanique

2.1.1 Conception sous SolidWorks

Pour la partie mécanique, nous avons choisi d'abord de discuter tous ensemble de la forme générale du robot. Nous avons dû prendre en compte les besoins des autres groupes, notamment au niveau de la place nécessaire pour positionner les capteurs et autres composants électroniques. Nous avons d'abord réfléchi sur une feuille sur laquelle nous dessinions les composants, mais il est rapidement apparu qu'il serait plus simple d'utiliser un logiciel nous donnant des valeurs précises en 3 dimensions. Nous avons le choix entre SolidWorks et Catia. Le logiciel Catia était installé sur les ordinateurs de la salle dans laquelle nous travaillions mais nous ne le maîtrisions pas. Par contre, comme nous avons suivi le cours de CTI3 qui est un cours de conception par ordinateur sur SolidWorks, nous avons décidé d'utiliser ce logiciel. Un autre avantage de SolidWorks était que tous les élèves de l'INSA peuvent l'installer gratuitement sur leur ordinateur personnel grâce à une licence étudiante. Il nous suffisait donc d'amener un ordinateur portable lors des séances de projet pour pouvoir travailler sans avoir à aller dans une autre salle. De plus, cela nous a permis de travailler depuis chez nous.

Nous avons d'abord dessiné les pièces qui nous étaient fournies, telles que les moteurs, les équerres et les roues. Nous avons ensuite conçu différentes bases, puis nous avons assemblé le tout pour voir ce qui fonctionnait le mieux.

2.1.2 Réalisation

Pour la conception de la base nous avons le choix entre le bois et le plexiglas. Le bois étant plus facile à travailler, nous avons opté pour cette option. Nous avons d'abord tracé notre base sur le bois, puis nous avons découpé celle-ci. Nous avons ensuite dû fixer un morceau de bois à angle droit sur la partie arrière. Pour cela, nous avons utilisé des équerres en aluminium. Percer des trous dedans a été très facile, il nous a juste fallu choisir les vis adaptées pour attacher les deux morceaux de bois, il fallait qu'elles ne soient pas trop longues pour ne pas trop ressortir sous la base mais elles devaient tout de même être assez solide pour maintenir les deux morceaux de bois ensemble et soutenir les moteurs. Une fois la base formée, nous avons fixé les équerres des moteurs de la même manière. Pour celles-ci, il a fallu nous assurer que l'on pouvait placer les moteurs une fois les équerres fixées. Une fois le tout attaché, nous avons dû placer notre troisième point d'appui. Nous avons alors réalisé que nous ne pouvions pas le centrer et nous avons donc dû mettre deux Ball-Caster. Cela a causé un petit problème d'équilibre, comme nous avons quatre points d'appui, l'une des

Ball-Caster ne touchait pas vraiment le sol. Pour fixer les composants nous avons utilisés des petites vis pour le circuit imprimé et la carte Arduino, le shield moteur se fixant dessus. Pour les batteries, nous avons décidé de les fixer simplement avec un peu de colle pour pouvoir les changer si nécessaire.

Pendant le projet, notre principal problème a été de nous assurer que les dimensions du robot soient suffisantes pour que les éléments nécessaires à son fonctionnement puissent être fixés dessus tout en respectant notre cahier des charges nous imposant un périmètre inférieur à 60 cm. Enfin nous regrettons de ne pas avoir eu le temps de peindre le robot.

2.2 Partie électronique

2.2.1 Les capteurs

Pour la partie électronique, nous avons préalablement étudié l'impact des capteurs optiques afin d'optimiser le déplacement du robot le long de la ligne afin que sa progression soit la plus rapide possible. Nous avons donc étudié le nombre de capteurs ainsi que leur position selon différents cas (ligne droite, virage). En fonction de leur nombre et de leur position dépend la taille et l'agencement de la carte électronique. Après réflexion nous avons décidé de choisir 4 capteurs que nous plaçons aux coins de la carte ceci permettant de mieux aborder les virages et de garder le robot dans une bonne position en ligne droite.

2.2.2 Conception de la carte

Le logiciel

Afin de concevoir la carte nous avons choisi d'utiliser le logiciel KiCad. Après une première prise en main, la conception de la carte électronique s'est déroulée en plusieurs étapes.

Une première étape de conception schématique qui consiste en la représentation symbolique du circuit électronique, schémas des composants selon les normes (Schéma scolaire avec la représentation des résistances par des rectangles par exemple). Ce schéma est fourni en annexe B. Nous avons du représenter le TCRT5000 sur le logiciel car il n'y avait qu'une photorésistance mais pas de capteur optique. Nous avons donc modifié l'empreinte existante pour retrouver l'empreinte « classique » d'un capteur optique qui est le suivant :

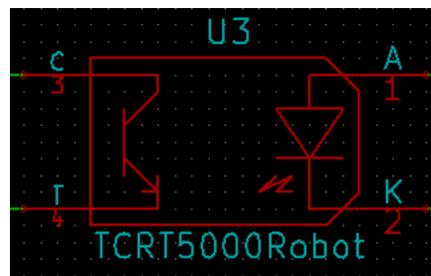


FIGURE 2.1 – empreinte du TCRT 5000

Une deuxième étape d'association qui consiste à affecter chaque composant électronique à son empreinte réelle contenant les caractéristiques physiques précises du composant (dimension réelle sur la carte électronique). Une nouvelle fois, il n'y avait pas l'empreinte dans la base de données. Nous l'avons une nouvelle fois créée pour pouvoir l'associer au composant :

Une troisième étape d'édition du circuit électronique qui donne sur papier le rendu final de la carte électronique. Cette dernière étape permet aux techniciens de concevoir la carte avec des mesures précises qui nous permettra de faire fonctionner notre robot. Nous avons donc du effectuer le routage des circuits. Il nous a donc fallu faire attention qu'il n'y ait pas de croisement entre les pistes pour que la carte puisse fonctionner. Nous avons essayer de minimiser le nombre de pistes présentes sur le circuit et de mettre le maximum possible du même côté de la carte. Ces schémas sont fournis en annexe C.1 pour le dessus de la plaque, C.2 pour le dessous et C.3 pour le côté composant.

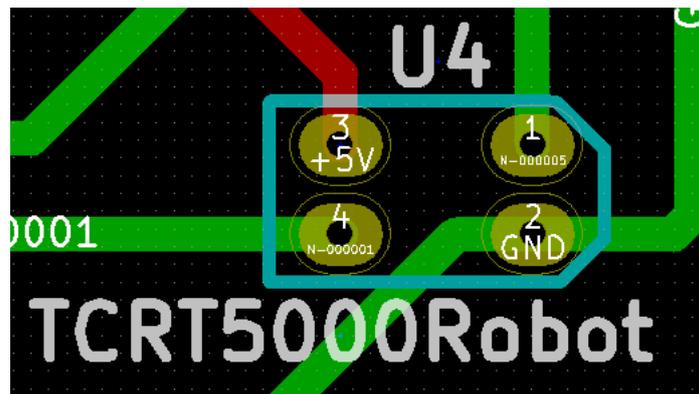


FIGURE 2.2 – empreinte physique du TCRT 5000

Réalisation technique

Nous avons alors monté chaque composant sur la carte électronique, il nous suffisait juste de souder les composants. Nous avons alors terminé la conception électronique du robot en associant la carte Arduino, le shield et la carte électronique. Ces connections ont été permises grâce à des câbles ou des connecteurs placés sur le shield ou la carte électronique. Nous avons également relié les moteurs et la batterie au shield.

Tout au long de la réalisation de ce projet, nous avons été en étroite collaboration avec les autres duos qui composaient notre équipe. En effet, nous avons du voir avec la partie informatique pour la position des capteurs ou la partie mécanique pour le placement des moteurs et des différentes cartes sur le châssis.

2.3 Partie informatique

2.3.1 Le logiciel et la carte Arduino

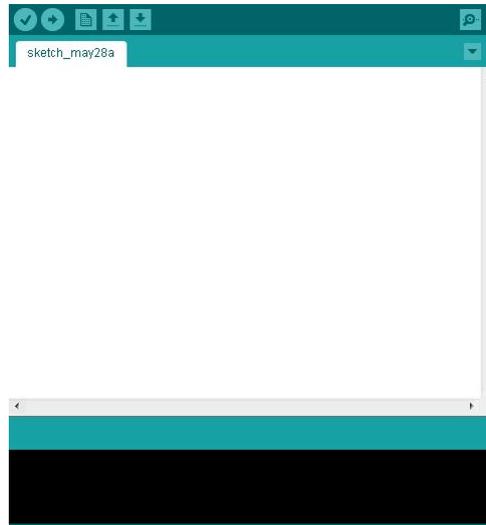


FIGURE 2.3 – capture d’écran du logiciel Arduino

Pour programmer la carte Aduino, nous avons utilisé le logiciel Arduino prévu à cet effet. C’est un logiciel libre et gratuit développé pour Windows, Mac et Linux. Ce logiciel présente une interface épurée et permet de développer facilement des programmes qui peuvent être transférés sur la carte en branchant celle-ci sur un port USB. Le langage utilisé par le logiciel est basé sur le C, auquel s’ajoutent des instructions spécifiques au langage Arduino. Le logiciel dispose d’une banque d’exempe très riche qui permet d’utiliser des morceaux de codes pré-écrit.

2.3.2 L’algorithme¹

Les spécificités du langage Arduino

Un programme Arduino doit impérativement être divisé en 2 parties : une fonction *setup* et une fonction *loop*. Ces deux fonctions sont de type *void*, c’est à dire qu’elles ne peuvent pas prendre de valeurs. La fonction *setup* est la fonction qui se lance au début du programme. Elle permet d’initialiser les variables et de définir les broches de la carte Arduino qui seront utilisées. La fonction *loop* se lance après la fonction *setup* et, comme son nom l’indique, fait une boucle jusqu’à ce que la carte soit débranchée.

La description du programme

Notre programme se découpe en 5 fonctions, incluant *loop* et *setup* :

- fonction *ligneDroite* () : Void
- fonction *virageGauche*() : Void
- fonction *virageDroite*() : Void

La fonction *ligneDroite* se charge de la gestion de la ligne droite et de la rectification de trajectoire en ligne droite. Les fonctions *virageDroite* et *virageGauche* se chargent d’effectuer un virage lorsque la ligne se courbe. Ces fonctions sont des fonctions non bloquante, c’est à

1. voir Annexe D

dire qu'elles s'exécutent sur une durée finie, et elles ne peuvent pas bloquer le programme. Ces 3 fonctions sont appelées dans la fonction *loop*. C'est donc cette fonction qui vérifie l'état des capteurs.

Les constantes du programme

Pour notre programme, nous avons utilisé de nombreuses constantes globales, définies en tout début de programme. Toutes ces constantes sont de type entier. Elles représentent deux choses : ou bien les valeurs des broches et leur nom associé, ou bien des valeurs de constantes utilisés dans le programme, comme par exemple le délai de temporisation, la valeur de seuil de déclenchement du capteur ou bien les différentes vitesses utilisées pour les moteurs.

2.4 Etat de l'art

Les robots suiveurs de ligne ne servent pas uniquement aux compétitions, ils sont aussi utilisés dans l'industrie ou dans les transports en commun.

2.4.1 Le TEOR²

Le TEOR (Transport est-ouest rouennais) de Rouen est un réseau de bus à haut niveau de service. Ces bus utilisent un système de guidage optique lors de leur entrée en station. Ce ne sont pas des bus sans chauffeurs comme dans certaines agglomérations, mais à l'approche d'un arrêt, le chauffeur laisse la main au système, qui permet au bus un accostage précis et régulier de la station, ce qui facilite l'accessibilité des personnes à mobilité réduite.

Le système fonctionne de la manière suivante :

- une caméra placée derrière ou au dessus du pare-brise lit un marquage au sol composé de deux lignes pointillées
- un ordinateur de bord analyse la position du bus par rapport à la voie et ajuste la position de celui-ci si nécessaire

Ce système apporte une grande fiabilité à l'arrivée en station. En effet, le risque de panne du guidage optique est très faible (99,98 % d'efficacité en 2008).



FIGURE 2.4 – Marquage lu et interprété par le guidage optique

2.4.2 Les véhicules à guidage automatique (VGA)³

Le véhicule à guidage automatique est une version plus développée du robot suiveur de ligne mais repose sur le même principe. Une de leur application en industrie est de pouvoir transporter de lourdes marchandises d'un point A à un point B et ce de manière rapide et sûre. Pour ce faire le véhicule a besoin de se repérer dans son environnement afin de se déplacer correctement.

Il existe plusieurs technologies permettant au véhicule de se repérer :

2. Source : Wikipédia « TEOR »

3. Source : Wikipedia « Véhicule à guidage automatique »

Le filoguidage : Les AGV se déplacent par détection d'une piste, tracée dans ou sur le sol. Il peut s'agir de fil émetteur d'ondes enfoui, de rail métallique au sol ou de fil électrique noyé au sol. Le signal transmis dans la piste est détecté par le VGA qui le suivra comme s'il s'agissait d'une voie ferrée.

L'optoguidage : Le système optoguidage permet au VGA de suivre une ligne peinte au sol grâce à des caméras embarquées. Il s'agit du même principe que le filoguidage mais avec une complexité et un coût d'installation moindres, car il n'est pas nécessaire de réaliser des travaux de gros œuvre. -Similaire au suiveur de ligne

Le laserguidage : - pas intéressant

Le géoguidage : - pas intéressant



FIGURE 2.5 – Un VGA

2.4.3 Les jeux pour enfants

Le circuit magique voiture est un jeu dans lequel l'enfant dessine lui même le chemin qu'il veut que la voiture suive grâce à un feutre noir sur un fond clair. La voiture suit le circuit grâce aux détecteurs optiques situés sous le jouet. Ces jouets sont semblables au robot que nous allons fabriquer.

L'avantage de ce jeu est que l'enfant peut tracer le circuit qui lui plait et qu'il n'a pas besoin de démonter celui-ci après avoir joué.



FIGURE 2.6 – Un exemple de circuit

2.4.4 Autres exemples

D'une manière plus primitive, on peut assimiler certains jouets à des robots auto-guidés. Prenons l'exemple des trains de modélisation ou bien des voitures de courses miniatures. Ces répliques se déplacent sur un circuit mais qui est lui « matériel » dans le sens où le véhicule est guidé grâce aux forces de frottement qui sont appliquées quand le véhicule est en mouvement. Cet exemple peut sembler anecdotique mais des applications de plus grande envergure voient le jour. On peut par exemple citer la ligne 14 du métro parisien qui est entièrement automatisée selon le même principe que nous venons d'énoncer.



FIGURE 2.7 – Un circuit de courses miniature



FIGURE 2.8 – Une rame de métro automatisée

Conclusion et perspectives

Ce travail s'est révélé être une expérience enrichissante pour moi. Le fait d'avoir un nouveau langage informatique à apprendre et à maîtriser me faisait peur au début mais il s'est avéré que celui utilisé pour coder sur Arduino est plutôt facile d'utilisation et les explications de Simon m'ont beaucoup aidé à comprendre. Ayant déjà de bonnes bases en pascal, je n'ai pas eu trop de mal à assimiler ce nouveau langage. Pour ce qui est de la partie relationnelle, tout s'est très bien passé. Nous n'avons eu aucun désaccord et nous nous sommes très bien entendus. De plus, cela m'a beaucoup appris sur les manières de travailler en équipe et comment échanger avec ses collègues de manière constructive. Je regrette cependant que nous n'ayons pas eu le réflexe de désigner un élève en tant que chef de groupe, ce qui aurait sûrement permis une meilleure transversalité entre les différentes tâches à réaliser. Cela nous aurait également fourni une vision plus claire sur l'avancement du projet, ce qui nous a fait défaut. Pour conclure, cet EC P6 s'est révélé enrichissant sur de nombreux plans, tant sur celui de l'informatique et de la conception, tant sur celui des relations entre collègues.

Jean ALLART

Ce projet a été pour moi une occasion de découvrir une discipline différente de ce que j'avais l'habitude d'étudier, car concevoir un robot du début à la fin au sein d'un groupe restreint a été pour moi un défi très productif. Je retiendrai beaucoup de chose de ce projet semestriel, mais ce que j'ai le plus apprécié reste la résolution des problèmes que l'on rencontrai au fur et à mesure de l'avancement de notre robot. Pour ma part, j'ai travaillé sur la partie mécanique et je dois dire que les travaux manuels inhérents à la réalisation du robot m'ont également beaucoup plus. Ce fut donc une expérience scientifiques et collective très intéressante.

Réda DRISSI

Ce projet dans l'ensemble m'a permis de découvrir un univers totalement nouveau : la robotique. En effet, il m'était déjà arrivé de réaliser une carte électronique mais réaliser et concevoir un robot tout entier du début à la fin fut pour moi une grande première. J'ai ainsi particulièrement travaillé sur la partie électronique. Ce travail s'est fait par étapes et a nécessité la prise en main du logiciel KiCad également totalement nouveau pour moi. J'ai pu ainsi découvrir le passage du circuit électronique schématisé format papier à sa mise en place de manière numérique à l'aide du logiciel pour au final obtenir une carte électronique physique. En tant qu'apport personnel, ce projet m'a beaucoup apporté au niveau du travail de groupe, la cohésion, l'entente et la solidarité étant des éléments nécessaires au bon avancement du projet. Il m'a également permis de développer mon autonomie, en effet, nous étions relativement libres dans la répartition de notre temps d'activité. Ainsi, je fus amené à surmonter certains problèmes et contre-temps, la notion de responsabilité étant très importante dans ce projet. En conclusion, ce projet m'a permis de découvrir un univers intéressant et enrichissant à travers un travail de groupe tout autant amusant.

Kevin NGYUEN

Avant de réaliser ce robot, mes connaissances en robotique et électronique étaient quasi nulles. Grâce à ce robot, sans avoir la prétention d'être devenu un expert dans le domaine, j'ai appris le fonctionnement de base d'un robot et de sa carte électronique. J'ai ainsi pu découvrir un nouveau domaine qui m'était totalement inconnu et mystérieux. Avec Kevin, nous nous sommes chargés plus particulièrement de la partie électronique. J'ai ainsi appris à me servir du logiciel Kicad qui nous a permis de concevoir la carte électronique et découvert les éléments qui se trouvent sur cette carte. Ce projet m'a beaucoup plus, et m'a permis de développer mes capacités à travailler en groupe et en autonomie. En effet, au cours de ce projet nous avons été confronté à de nombreux problèmes que nous avons résolu la plupart du temps en groupe.

Damien POINTIN

Pendant ce projet, j'ai découvert les principales étapes de la fabrication d'un robot. J'ai également appris l'importance d'une bonne communication dans un groupe, surtout quand plusieurs phases de la fabrication sont effectuées simultanément. Enfin, j'ai appris à faire face aux imprévus auxquels nous avons dû faire face.

Hélène SOUDRY

Ce projet a été pour moi l'occasion de découvrir un milieu qui m'intéresse, à savoir la robotique. C'est un domaine qui semble compliqué à aborder seul et le fait de l'avoir fait en groupe et accompagné d'un professeur m'a permis d'apprendre de nombreuses choses. Je suis très fier du résultat obtenu, qui est le fruit d'un travail de collaboration entre tous les membres du groupe. Bien que le projet ait été ponctué de haut et de bas, nous avons su redoubler d'efforts pour parvenir au résultat souhaité, à savoir faire en sorte que notre robot suive une ligne. Ce projet m'a donc apporté beaucoup : j'ai appris des choses dans de nombreux domaines, et j'ai su trouver ma place au sein d'un groupe afin d'aboutir à notre robot.

Simon WALLON

Annexe A

Réalisation SolidWorks

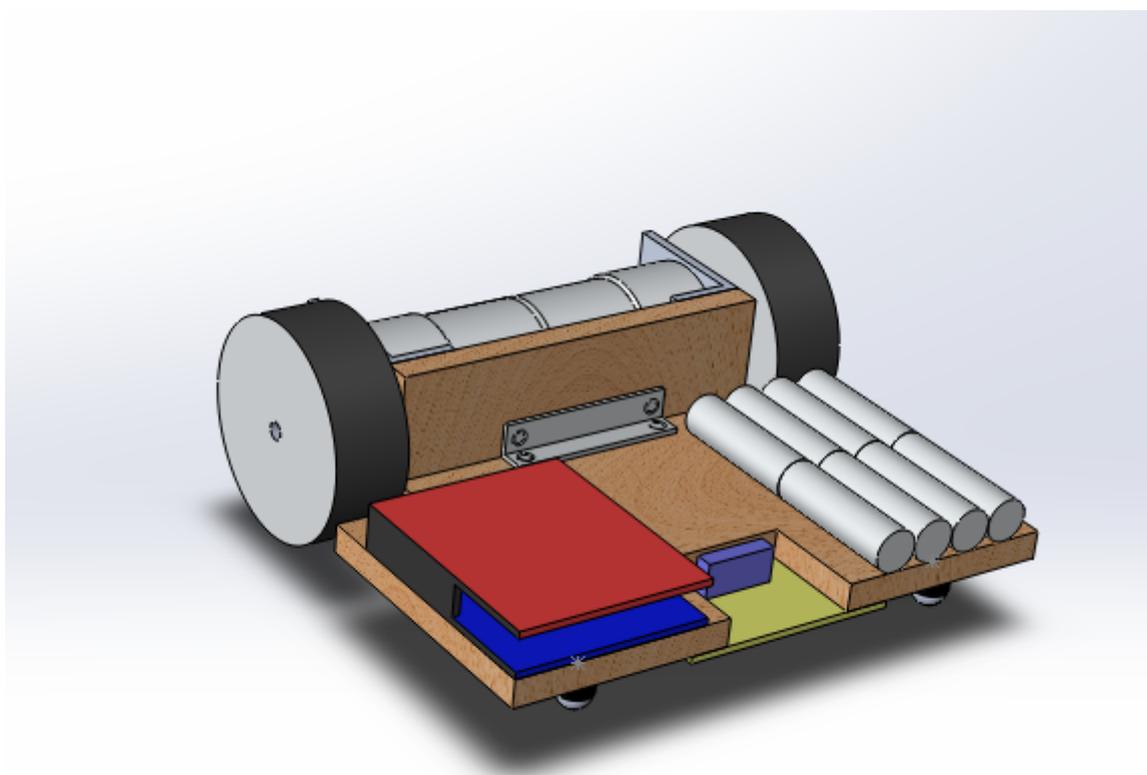


FIGURE A.1 – Capture d'écran du robot sous SolidWorks

Annexe B

Schéma de la carte électronique

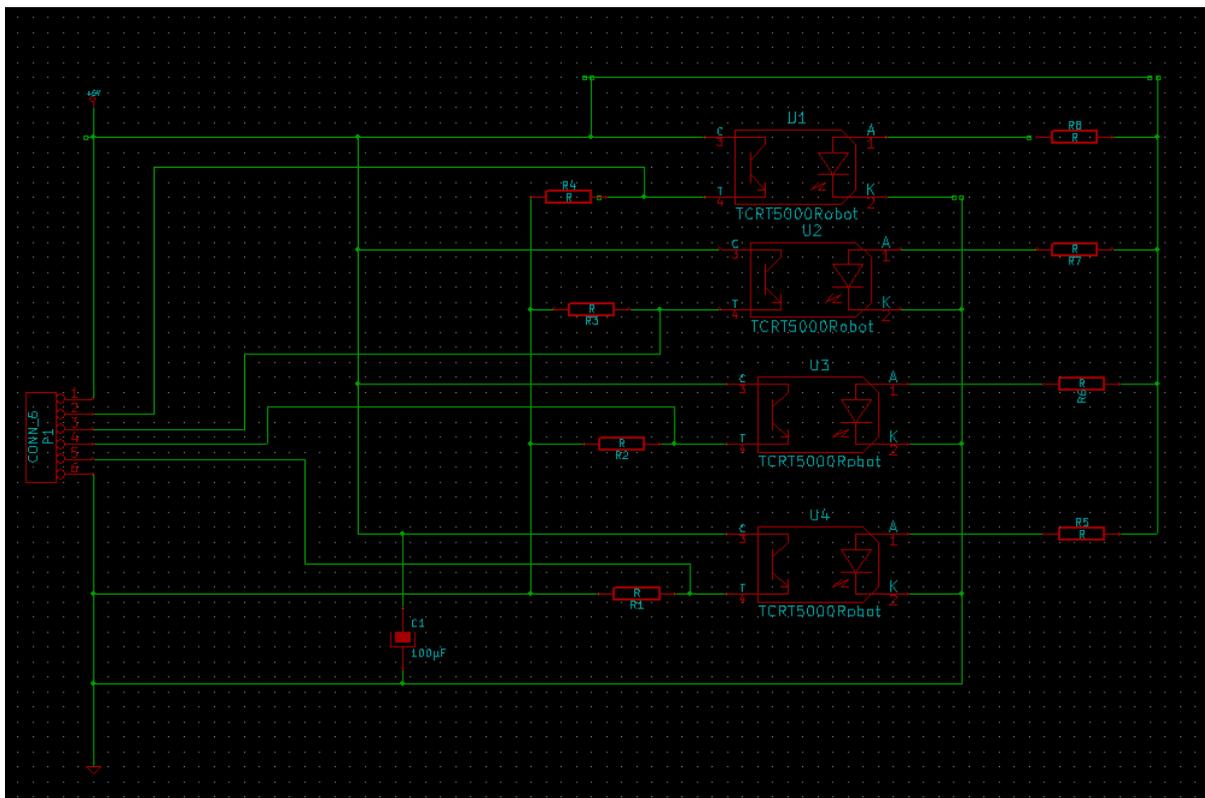


FIGURE B.1 – Schéma de la carte électronique

Annexe C

Plan de la carte electronique

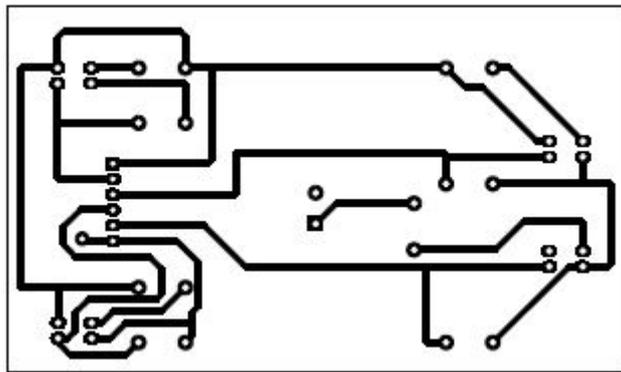


FIGURE C.1 – Carte électronique vu de dessus

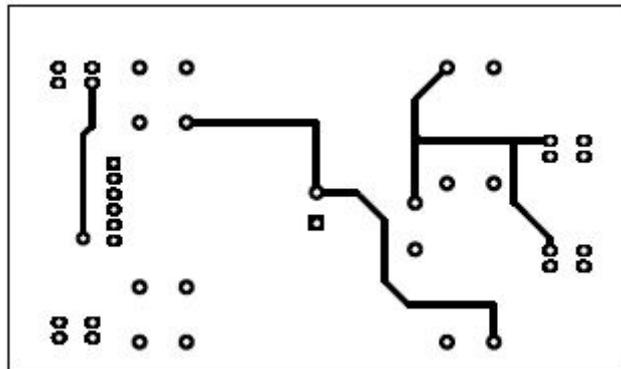


FIGURE C.2 – Carte électronique vu de dessous

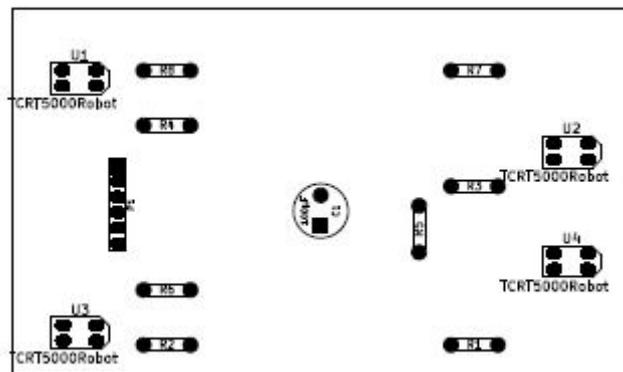


FIGURE C.3 – Empreinte des composants

Annexe D

Code

```
const int GH = 0;
const int GB = 1;
const int DH = 2;
const int DB = 3;
5 const int seuil = 300;
const int pwmg = 3;
const int pwmd = 11;
const int vitesseGrandVirage = 50;
const int vitessePetitVirage = 180;
10 const int vitesseLD = 255;
const int tempo = 100;

void setup()
{
15 //setup moteur gauche
pinMode(12, OUTPUT); //direction
pinMode(pwmg, OUTPUT); //PWM

//setup moteur droit
20 pinMode(13, OUTPUT); //direction
pinMode(pwmd, OUTPUT); //PWM

//setup direction des moteurs
digitalWrite(12, LOW);
25 digitalWrite(13, LOW);

//setup capteurs
pinMode(GH, INPUT);
pinMode(GB, INPUT);
30 pinMode(DH, INPUT);
pinMode(DB, INPUT);
delay(100);
}

35 void ligneDroite()
{
analogWrite(pwmg, vitesseLD);
analogWrite(pwmd, vitesseLD);
if ((analogRead(GB) < seuil) && (analogRead(DB) > seuil))
40 analogWrite(pwmg, vitessePetitVirage);
```

```

else if ((analogRead(GB)>seuil)&&(analogRead(DB)<seuil))
    analogWrite(pwmd,vitessePetitVirage);
}

45 void virageGauche()
{
    analogWrite(pwmg,vitesseGrandVirage);
    analogWrite(pwmd,vitesseLD);
    delay(tempo);
50 analogWrite(pwmg,vitesseLD);
}

void virageDroite()
{
55 analogWrite(pwmd,vitesseGrandVirage);
    analogWrite(pwmg,vitesseLD);
    delay(tempo);
    analogWrite(pwmd,vitesseLD);
}

60 void loop()
{
    ligneDroite();
    if (analogRead(GH)<seuil)
65 {
        virageGauche();
    }
    else if (analogRead(DH)<seuil)
    {
70 virageDroite();
    }
}

```