



POLYTECH[®]
LILLE

MANUEL BUENO

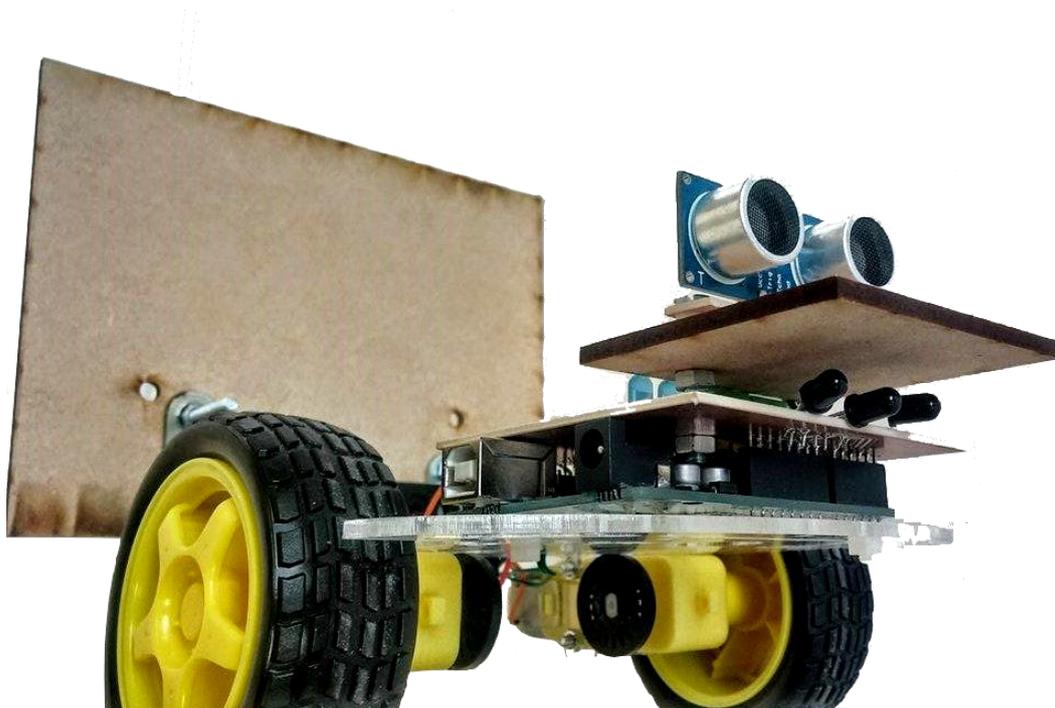
MAUREEN GILLET

IMA4 SA

RAPPORT DE PROJET

Train de véhicule

DEPARTEMENT INFORMATIQUE MICROELECTRONIQUE AUTOMATIQUE



Remerciements

Premièrement, nous tenons à remercier l'équipe pédagogique de la formation Informatique, Microélectronique et Automatique de Polytech Lille, qui nous a enseigné les connaissances de base pour mener à bien ce projet de 4^{ème} année.

Nous remercions également notre tuteur de projet, Monsieur Xavier REDON, pour son aide et ses conseils tout au long du projet.

Nous remercions aussi Monsieur Thierry FLAMEN pour ses réponses et ses précieux conseils sur la conception des cartes électroniques.

Enfin, un merci particulier à Monsieur Laurent ENGELS, pour le temps consacré au tournage et montage de la vidéo de projet.

Sommaire

Remerciements.....	2
Introduction.....	4
1. Présentation du projet	5
1. Objectif et cahier des charges	5
2. Choix techniques : matériel et logiciel	6
3. Calendrier prévisionnel et organisation	7
2. Déroulement du projet.....	8
1. Conception.....	8
1. Générale	8
2. Véhicules suiveurs	8
3. Véhicule de tête.....	11
2. Programmation.....	13
1. Robot de tête.....	13
2. Robots suiveurs	15
3. Communication infrarouge	17
3. Difficultés rencontrées	20
4. Améliorations possibles.....	21
Conclusion	22

Introduction

Dans le cadre de notre 4^{ème} année de formation ingénieur à Polytech Lille, en Informatique Microélectronique et Automatique, nous avons un projet à réaliser tout au long du deuxième semestre. C'est l'occasion d'appliquer les connaissances que nous avons pu acquérir depuis le début de notre formation, de les améliorer et de développer de nouvelles compétences autour d'un sujet technologique. Ce projet nous permet aussi de gérer complètement un projet, de l'élaboration d'un cahier des charges, à une réalisation matérielle, en passant par le design et la conception. Nous avons choisi le sujet « Train de véhicule », qui consiste à faire en sorte qu'un certain nombre de véhicules se suivent de manière autonome, le premier devant être téléguidé par Wifi.

Dans ce rapport, nous commençons par faire une présentation plus détaillée du projet et du cahier des charges. Puis nous développons le déroulement du projet en deux parties : la conception des robots et leur programmation. Enfin, nous finissons en tirant un bilan de ce projet, avec les difficultés rencontrées et les améliorations possibles.

1. Présentation du projet

1. Objectif et cahier des charges

L'objectif de ce projet est de réaliser une demi-douzaine de véhicules capables de se suivre afin de former un train de véhicules. Le robot de tête est téléguidé par Wifi, tandis que les autres doivent le suivre de manière autonome.

Les robots seront montés sur des châssis à bas coût : les 2WD Robot Car châssis (image ci-contre).

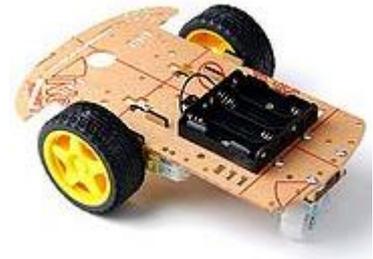


Figure 1 : 2WD Robot Car châssis

Ces châssis comprennent déjà deux moteurs 6V, deux roues et un support de 4 piles 1.5V. Il faudra ajouter deux odomètres, un driver de puissance pour les moteurs, ainsi qu'un microcontrôleur. Le véhicule de tête sera commandé par un microcontrôleur avec Wifi intégré de type ESP8266. Il se comportera comme un point d'accès Wifi et implantera un mini-serveur Web pour permettre le contrôle via un smartphone ou un ordinateur. Les autres véhicules seront contrôlés par des cartes Arduino Uno.

Chaque véhicule possède un numéro d'identification N, et doit suivre celui de numéro N-1. Pour cela, il doit détecter et identifier un véhicule dans une zone autour de lui. Le véhicule à suivre émet des ondes infrarouges modulées par son numéro d'identification, donc chaque robot émet à une fréquence différente. Plusieurs capteurs infrarouges doivent être installés à l'avant du robot suiveur afin de récupérer ces informations tandis qu'une LED infrarouge sera située à l'arrière du robot. Le suivi se fera à une distance prédéfinie. Le robot suiveur sera équipé d'un capteur ultrason pour asservir sa distance avec le véhicule à suivre. Il faudra que l'arrière des véhicules soit bien plan pour que le capteur ultrason puisse prendre une mesure fiable.

Un véhicule suiveur doit tourner sur lui-même pour détecter les balises infrarouge. Si une balise est détectée, on tente de l'identifier. S'il s'agit de la bonne balise, on commence la poursuite, sinon on continue à chercher les balises. Tant que la bonne balise est dans le champ de vision, il faut la garder le plus au centre. Plusieurs capteurs infrarouges seront utilisés pour assurer la bonne direction du robot. La vitesse est réglée par rapport à la distance retournée par le télémètre.

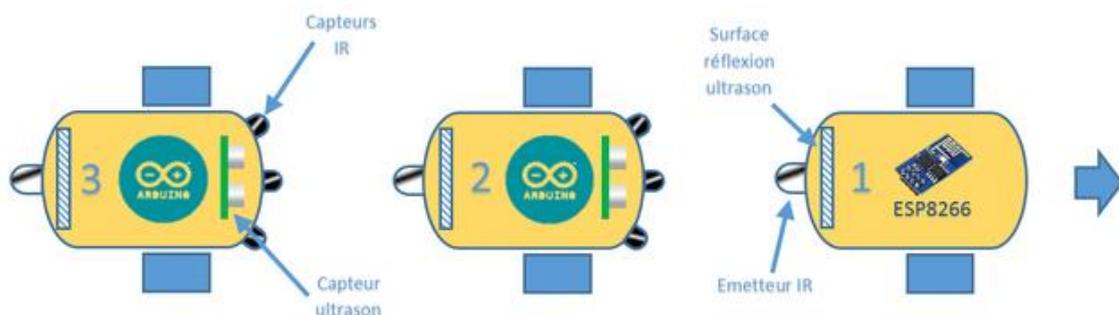


Figure 2 : Principe du train de véhicules

2. Choix techniques : matériel et logiciel

Pour réaliser notre projet, nous avons besoin de matériel. En fonction des objectifs et des solutions proposées, nous avons établi la liste ci-dessous. Nous avons séparé ce matériel en deux parties : celui déjà disponible à Polytech et celui que nous devons commander.

Matériel	Quantité disponible	Quantité à commander	Commentaires : spécificités et fournisseurs
Châssis 2WD	4	-	Corps + réducteurs 6V + roues + odomètres
Piles / batteries	4 x 4 x 1.5V	-	
Arduino Uno	3	-	
ESP8266	1	-	
LED IR	-	20	RS [1] , angle de diffusion : 130°, longueur d'onde : 880nm
Phototransistors IR	-	20	RS [2] , angle de sensibilité : 50°, gamme spectrale de sensibilité : 730-1120nm
Ultrasons SR04	3	-	
Drivers moteur Pololu	-	1	GoTronic [3]
Drivers moteur TB6612FNG	-	3	Mouser [4]
Boutons poussoirs	4	-	
LEDs couleur	-	10	RS [5]
Résistances CMS 220Ω	-	6	Mouser [6]
Résistances CMS 470kΩ	-	20	Mouser [7]
Fils mâle-mâle	-	1 lot de 65	Farnell [8]
Capacités 10μF CMS	-	10	Mouser [9]
Capacités 0.1μF CMS	-	10	Mouser [10]
Embases CI 40 contacts 1 rangée	-	4	RS [11]
Borniers à vis	-	15	Farnell [12]

Figure 3 : Liste du matériel nécessaire

Pour contrôler les robots, nous devons programmer les microcontrôleurs. Pour l'ESP8266, nous utiliserons l'IDE Arduino, tandis que les cartes Arduino Uno seront programmées en C, puis compilées avec l'outil AVR-GCC, avant d'être flashées.

3. Organisation

La première étape de notre projet a été d'établir une liste de tâches à réaliser. Elles peuvent être regroupées en cinq parties :

Étudier l'alimentation et l'électronique de puissance	<ul style="list-style-type: none">› Déterminer la source d'énergie› Régulateur 6 V pour alim driver moteur ?› Choix du driver moteur
Étudier la communication infrarouge	<ul style="list-style-type: none">› Circuit d'émission: LED IR + résistance› Circuit réception: Phototransistor IR + résistance› Déterminer la portée du signal ?› Étudier la directivité du signal émis ? De la détection ?› En déduire la disposition des détecteurs› Étudier l'influence de la lumière ambiante ?› Programmer cette communication sur Arduino
ESP8266	<ul style="list-style-type: none">› Etude› Programmation› Interface Web
Arduino	<ul style="list-style-type: none">› Programmation (infrarouge, ultrason, déplacement, etc.)
Montage	<ul style="list-style-type: none">› Suiveurs› Tête

Figure 4 : Liste des tâches à réaliser au cours du projet

Les études d'alimentation et de communication infrarouge sont à mener en premier, tandis que les autres tâches peuvent être ensuite réalisées en parallèle.

2. Déroulement du projet

1. Conception

1. Générale

Alimentation

Les moteurs des châssis doivent être alimentés en 6V et les cartes Arduino peuvent supporter entre 6 et 20V. Par conséquent, nous avons envisagé deux solutions pour alimenter nos montages. Soit une pile 9V et un régulateur 6V, nous permettant ainsi d'avoir une tension constante de 6V aux bornes de nos circuits. Soit 4 piles de 1,5V. Nous avons opté pour cette dernière à cause de sa simplicité. Par ailleurs, un boîtier de 4 piles 1,5V était déjà présent sur les châssis des Robot Car.

Electronique de puissance

Pour contrôler les moteurs, nous avons choisi le driver de puissance TB6612FNG, de Toshiba. Il correspond parfaitement à notre application puisqu'il peut contrôler indépendamment deux moteurs à courant continu bidirectionnels. Sa gamme de tension est comprise entre 4,5 et 13,5V et il délivre un courant continu de 1A. Par ailleurs, il est de petite taille et à un coût très bas.

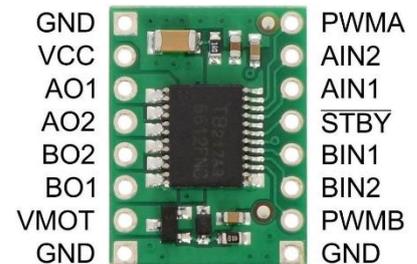
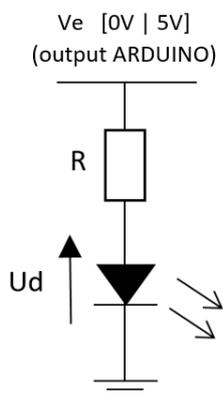


Figure 5 : Driver de puissance TB6612FNG

2. Véhicules suiveurs

Communication infrarouge

Les robots suiveurs sont équipés d'une LED infrarouge à l'arrière, ainsi que de trois phototransistors situés à l'avant.



Nous avons d'abord déterminé le montage de l'émission infrarouge. La tension d'entrée est délivrée par un pin numérique de l'Arduino. Ce dernier peut fournir un courant maximal de 30mA. On fixe donc le courant dans le circuit à 20mA pour éviter d'endommager l'Arduino, tout en gardant une puissance suffisante pour l'émission. On en déduit la valeur de la résistance : 220Ω. Nous avons sélectionné des LEDs infrarouges avec un grand angle de diffusion (130°), pour que les robots soient facilement détectables.

Figure 6 : Circuit d'émission infrarouge

En ce qui concerne la réception infrarouge, nous avons opté pour le montage ci-contre. La tension du collecteur est proportionnelle à l'intensité lumineuse captée par le phototransistor et est lue par un convertisseur analogique-numérique de la carte Arduino. La résistance de pull-up permet de régler le gain du montage. Ce gain doit être suffisant pour capter le signal utile à bonne distance, tout en restant pas trop fort pour rejeter le bruit ambiant. Ce dernier est principalement causé par l'éclairage ambiant, un cache est à prévoir au-dessus des capteurs. Expérimentalement, nous avons déterminé la valeur de la résistance à 470k Ω .

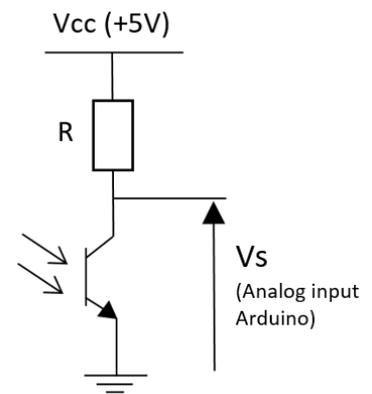


Figure 7 : Circuit de réception infrarouge

L'objectif étant de situer assez précisément le robot à suivre, ce sont trois phototransistors d'angle de réception restreint (50°) qui sont utilisés. Selon le ou les capteurs qui reçoivent un signal, il est possible de connaître la position du robot par zone et d'agir en conséquence. On définit ainsi 6 zones : le robot est en face, modérément sur le côté, complètement à côté ou carrément hors portée.

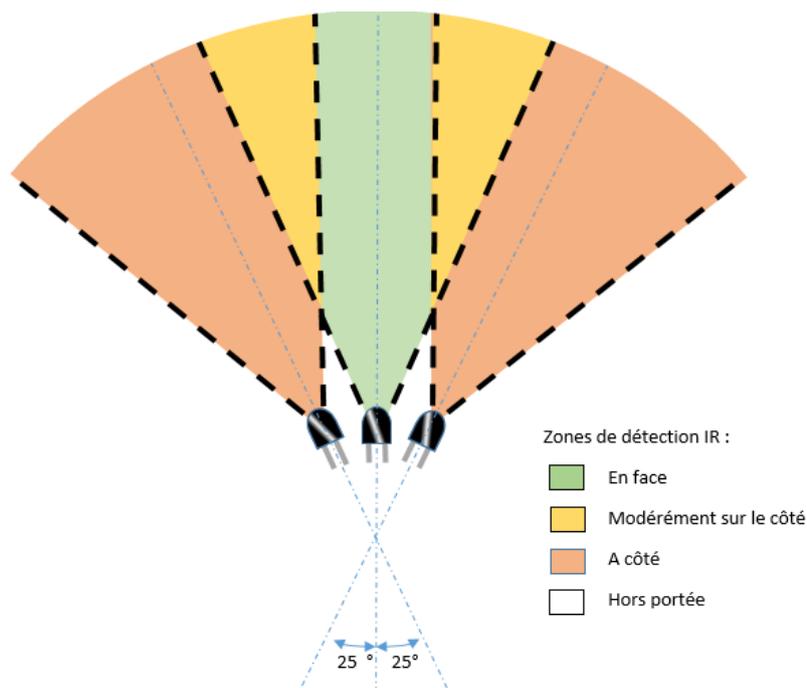


Figure 8 : Disposition des phototransistors et zones de détection

Capteur ultrason

Pour connaître la distance entre deux robots qui se suivent et la maintenir constante, nous nous servons d'un capteur ultrason. Nous avons choisi la référence HC-SR04, qui est très simple. Sa portée est comprise entre 2cm et 4m, ce qui est parfait pour notre utilisation. Il suffit d'envoyer une impulsion de 10µs en entrée et le capteur retourne une largeur d'impulsion proportionnelle à la distance mesurée. Grâce à la formule suivante, on retrouve la valeur de la distance en cm.



Figure 9 : Capteur ultrason HC-SR04

$$\text{distance (cm)} = \frac{\text{impulsion } (\mu\text{s})}{58}$$

Des panneaux plats, découpés à la découpeuse laser, sont placés à l'arrière des robots afin que les ondes ultrasons se réfléchissent sur une surface plane et obtiennent une mesure fiable.

Montage

Pour avoir un montage plus propre et plus stable, nous avons décidé de réaliser des cartes électroniques. Etant donnée la disposition des éléments, nous en avons besoin de plusieurs. La carte principale se situe à l'avant du robot. Elle est constituée du dispositif de détection infrarouge, du microcontrôleur et du driver moteur, auxquels nous ajoutons les LEDs qui permettent de connaître le numéro du robot ainsi que le bouton poussoir pour en changer. Initialement, le driver moteur choisi est vendu sur une carte comprenant un circuit TB6612FNG et des capacités. Nous recréons ce montage et l'intégrons à notre carte.

Le capteur ultrason doit être surélevé, de façon à ne pas être gêné par les autres composants. Cela forme une deuxième petite carte électronique. Enfin, à l'arrière du robot, on place la LED infrarouge, avec une résistance afin de limiter le courant. Cela constitue une dernière petite carte.

Les schémas électroniques sont dessinés sur le logiciel Altium Designer

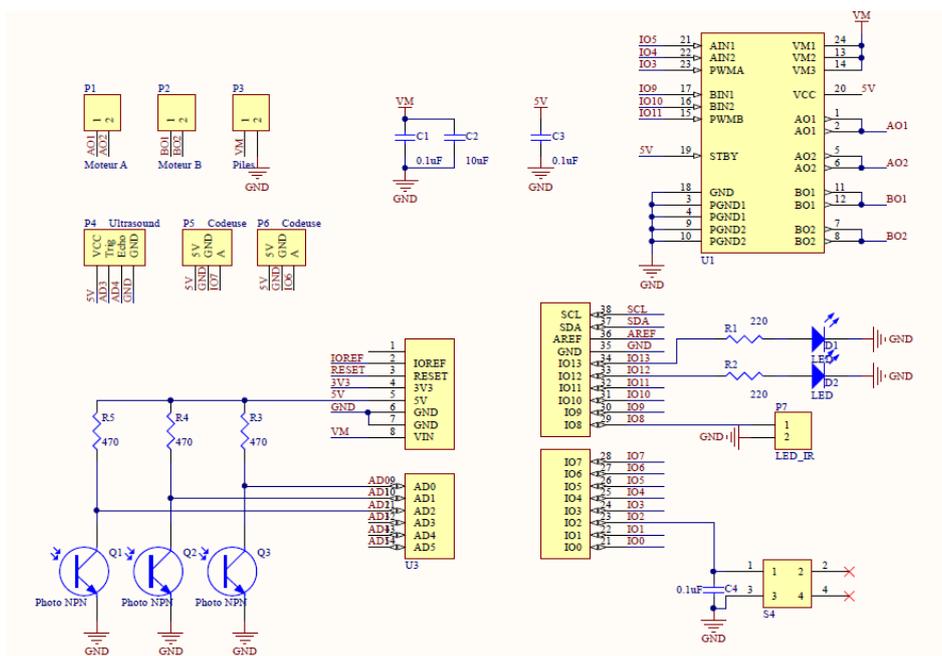


Figure 10 : Schéma électronique de la carte principale

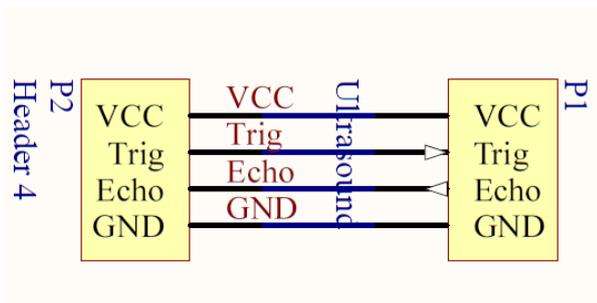


Figure 12 : Schéma électronique de la carte ultrason

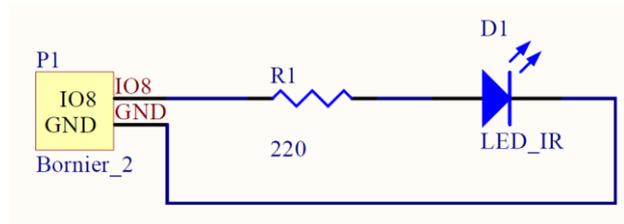


Figure 11 : Schéma électronique de la carte infrarouge

Après avoir créé les empreintes de tous les composants, nous dessinons les cartes électroniques :

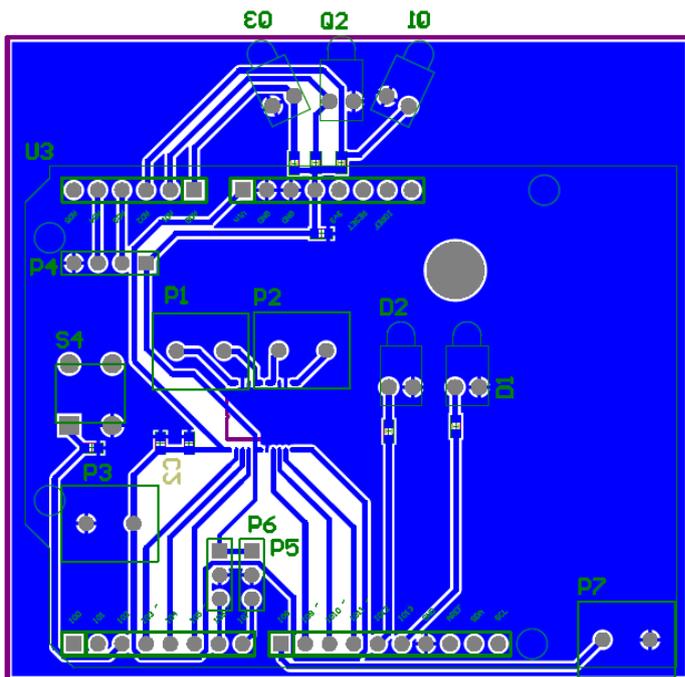


Figure 15 : Carte électronique principale

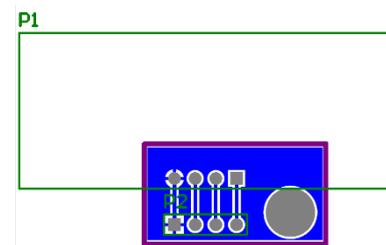


Figure 13 : Carte électronique ultrason

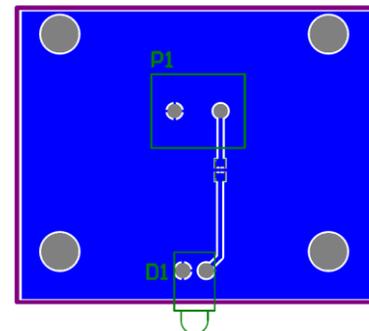


Figure 14 : Carte électronique infrarouge

3. Véhicule de tête

Le montage du véhicule de tête nécessite moins de composants. Il comprend le module Wifi ESP8266, le driver moteur, une LED infrarouge et sa résistance. Un montage sur breadboard a permis de voir que l'ESP8266 consomme beaucoup de puissance et qu'un seul boîtier de piles n'est pas suffisant pour alimenter à la fois le module Wifi et le driver de puissance. Par conséquent, les piles sont découplées et un boîtier alimente un composant. Par ailleurs, l'ESP8266 délivre un courant maximal de 10mA, ce qui n'est pas suffisant pour faire fonctionner la LED infrarouge de manière optimale. Un transistor est donc ajouté au montage, de façon à obtenir un courant plus élevé.

On obtient le montage suivant :

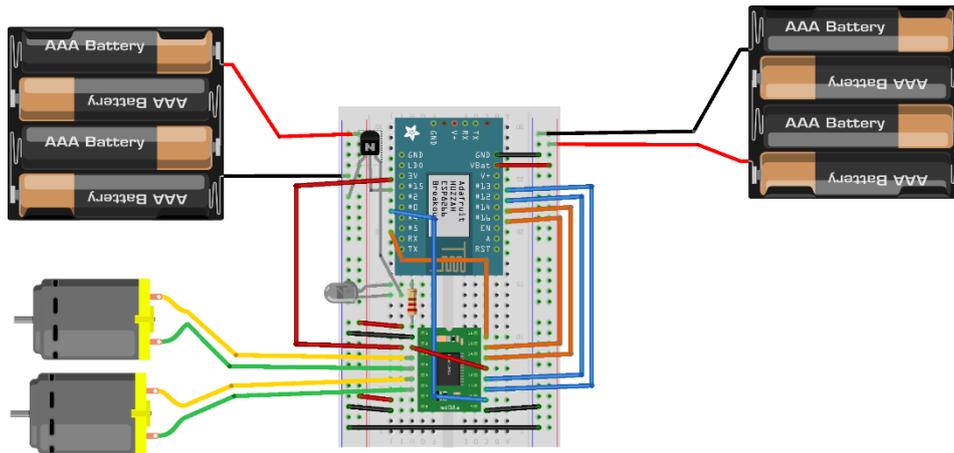


Figure 16 : Montage du robot de tête

Puis finalement, toujours pour des raisons de stabilité et de propreté, le circuit est monté sur une carte électronique. Pour la LED infrarouge, qui se situera à l'arrière du robot, nous utiliserons le même PCB que pour les robots suiveurs. Pour le montage général, qui comporte les autres composants, nous créons un nouveau schématique et un nouveau PCB :

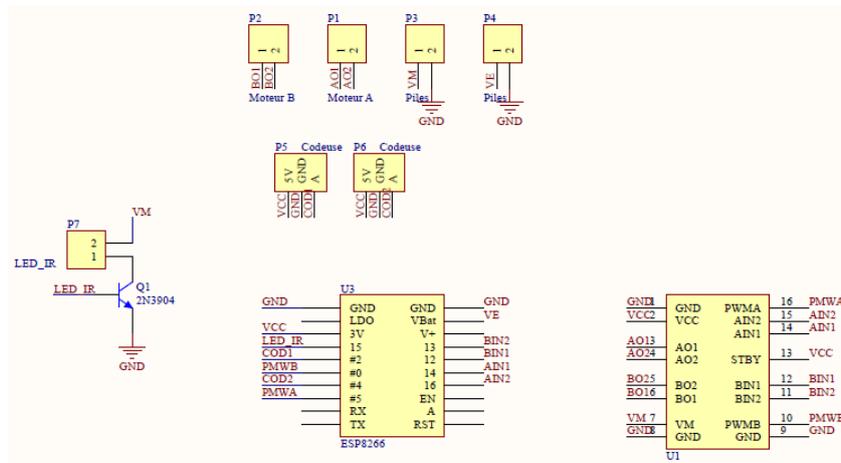


Figure 17 : Schéma électronique de la carte principale

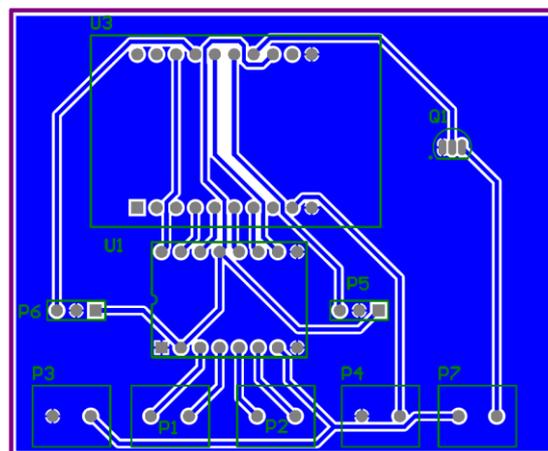


Figure 18 : Carte électronique principale

2. Programmation

1. Robot de tête

Présentation de l'ESP8266

L'ESP8266 est un microcontrôleur intégrant une carte WIFI. Celui-ci peut donc se comporter en point d'accès WIFI, et permettre à l'utilisateur de se connecter directement dessus depuis un appareil connecté. Nous utilisons le module Adafruit HUZZAH ESP8266 breakout, intégrant l'ESP8266, un régulateur d'alimentation, une antenne, l'accès aux pins d'Entrée/Sortie ainsi qu'un port FTDI permettant de flasher le microcontrôleur.

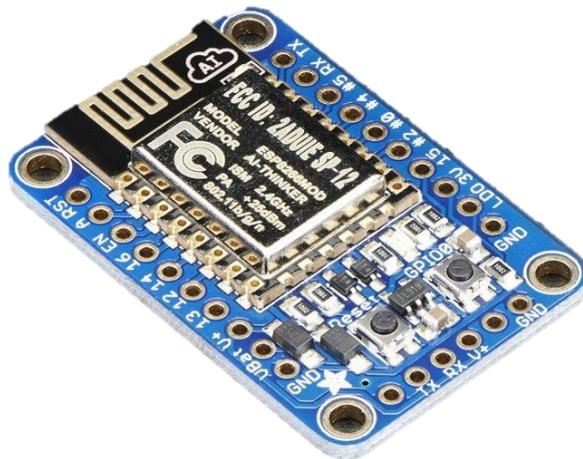


Figure 19 : Adafruit HUZZAH ESP8266 breakout

Grâce à un add-on sur l'IDE Arduino compatible avec le module Adafruit, il est possible de programmer, de compiler et de flasher l'ESP directement à partir de l'IDE Arduino.

Principe de contrôle du robot de tête

Afin de contrôler le robot de tête, l'utilisateur se connecte en WIFI directement sur l'ESP8266 grâce à un appareil connecté (ordinateur, smartphone, ...). Il se rend ensuite à l'adresse « 192.168.4.1/interface » sur son navigateur, où il trouvera une interface web permettant de contrôler simplement le robot.

Interface de contrôle du robot de tête

Donner la direction du robot:



Figure 20 : Interface web de contrôle

L'ESP héberge directement un serveur. Le navigateur internet envoie donc des requêtes http GET à l'ESP, qui renvoie des paquets http contenant du html/JavaScript, permettant d'afficher l'interface Web.



Figure 21 : Communication Navigateur/ESP

Programmation de l'ESP8266

Deux tâches s'exécutent en parallèle sur le microcontrôleur du robot de tête. Premièrement, une tâche aperiodique, scrutant l'arrivée d'un nouveau client sur le serveur. Dans ce cas, la requête du client est analysée. Cinq requêtes sont possibles :

- /interface : Renvoie l'interface web
- /LEF : Le robot tourne sur la gauche
- /FOR : Le robot avance
- /RIG : Le robot tourne sur la droite
- /STB : Le robot s'arrête

En fonction de la requête, la consigne est prise en compte et l'utilisateur est redirigé vers l'interface Web.

Une seconde tâche, périodique et cadencée par un timer d'une milliseconde, gère l'émission du signal Infrarouge et la génération des PWM. Il n'existe pas de timer intégré permettant de générer des PWM dans le microcontrôleur, celles-ci sont donc gérées de manière logicielle.

Voici le diagramme d'activité du programme du robot de tête :

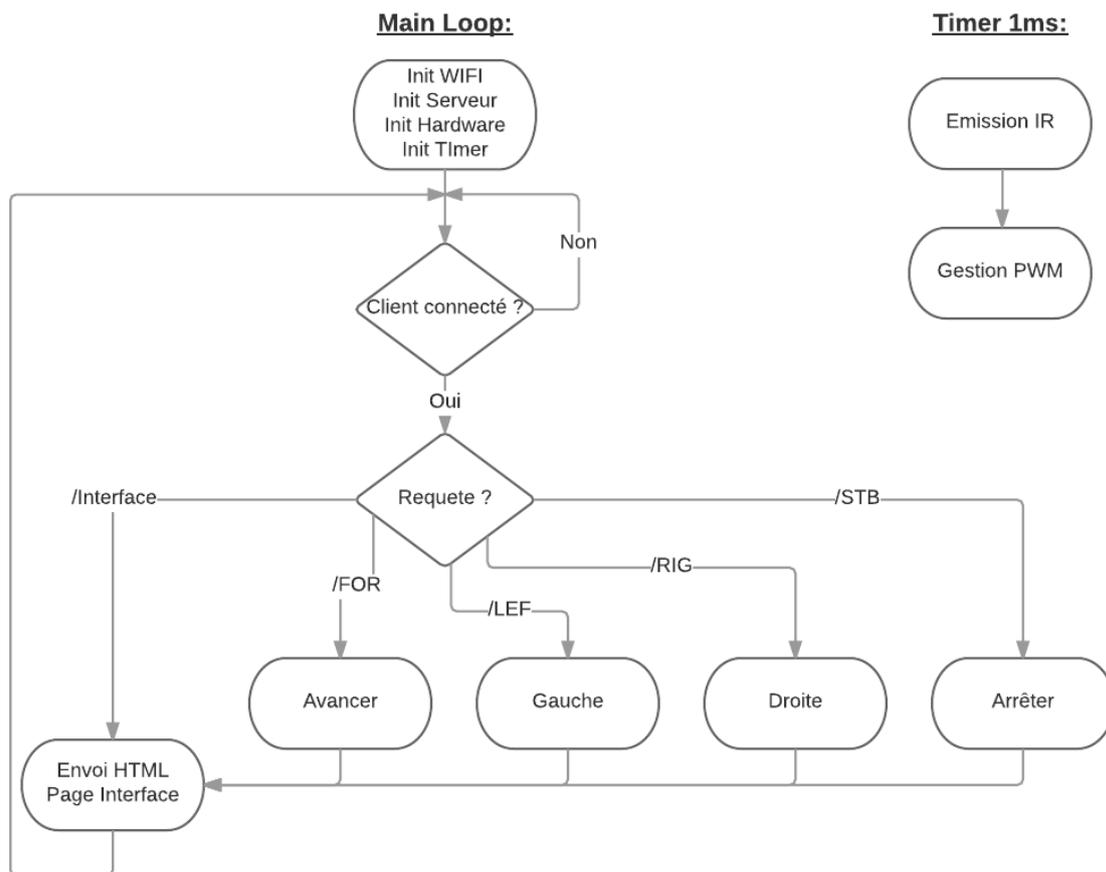


Figure 22 : Diagramme d'activité du robot de tête

2. Robots suiveurs

Présentation de l'ATMEGA 328P

L'Atmega 328P est un microcontrôleur largement utilisé dans les systèmes embarqués. Celui-ci est souvent utilisé embarqué sur une Arduino Uno. Cette carte intègre principalement un régulateur d'alimentation, un accès aisé aux pins d'Entrées/Sorties, un convertisseur Série/USB pour flasher le microcontrôleur et pour la communication série.

La programmation du microcontrôleur se fait en langage C. L'outil AVR-GCC permet de compiler les codes C. L'outil Avrdude permet lui de flasher le microcontrôleur.



Figure 23 : Carte Arduino UNO, embarquant l'Atmega 328P

Le programme des robots suiveurs doit assurer plusieurs fonctions :

- Mesurer les signaux infrarouges
- Détecter la présence d'un robot suiveur à partir des mesures infrarouges
- Emettre un signal infrarouge
- Mesurer la distance via le capteur ultrason
- Pouvoir changer d'identifiant (de place dans le train de véhicules)
- Gérer ses déplacements

Le programme comporte donc plusieurs tâches :

- Une tâche périodique cadencée par un timer de période 100µs. Celle-ci gère la détection des signaux infrarouges du robot à suivre, l'émission des signaux infrarouges, ainsi que l'acquisition de la distance via le capteur ultrason
- Une tâche permettant de mesurer la tension analogique des phototransistors infrarouges. Celle-ci est déclenchée en interruption par une fin de conversion Analogique Numérique. Cette tâche relance une nouvelle conversion Analogique Numérique sur un phototransistor différent. De cette manière, les mesures sont faites sur les 3 phototransistors à tour de rôle.
- Une tâche aperiodique permettant de changer l'identifiant du robot à l'appui du bouton poussoir. Celle-ci modifie les fréquences de détection et d'émission des signaux infrarouges.
- Une tâche principale calculant en permanence les tensions à appliquer aux moteurs en fonction de la distance mesurée et de la détection du robot à suivre par les phototransistors.

Voici le diagramme d'activité du programme des robots suiveurs :

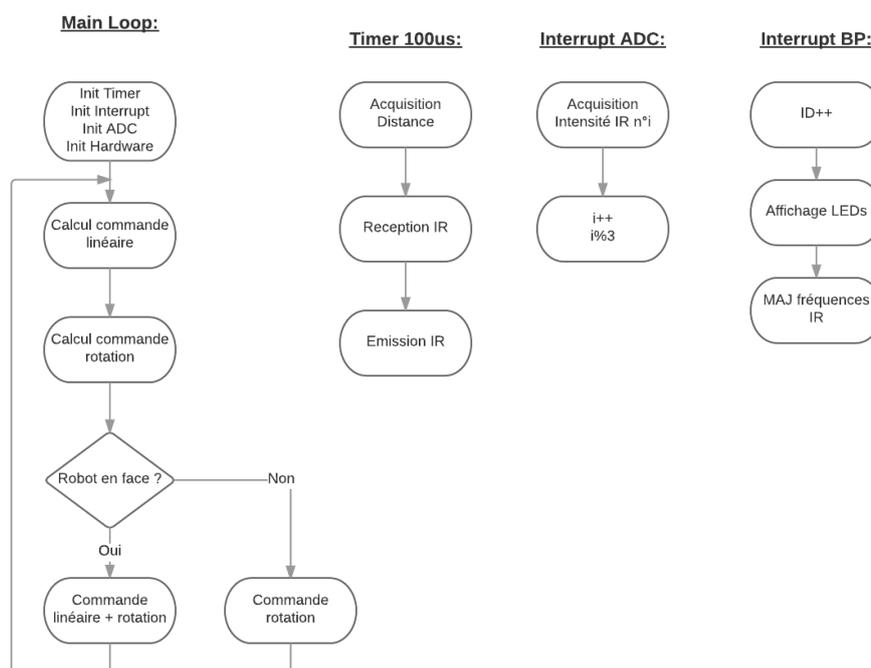


Figure 24 : Diagramme d'activité du programme des robots suiveurs

La commande linéaire est calculée à partir de la distance ultrason. La commande appliquée aux moteurs est proportionnelle à la différence entre la distance de consigne (30cm) et la distance réelle. Cette commande est appliquée aux moteurs uniquement si le robot à suivre est détecté devant.

La commande en rotation est définie à partir de la position latérale du robot à suivre. Voici le schéma des commandes appliquées aux moteurs en fonction de la position du robot à suivre :

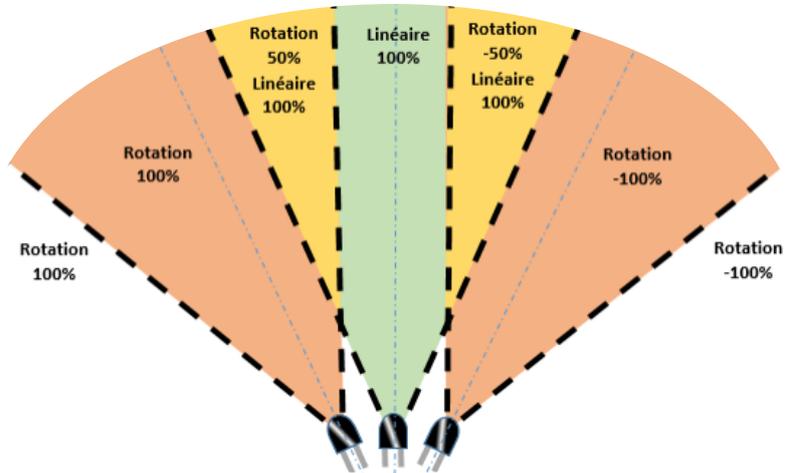


Figure 25 : Commandes appliquées aux moteurs selon la position du robot

3. Communication infrarouge

Chaque robot a une place dans le train de véhicules, définie par son identifiant. Le robot suiveur doit suivre celui le précédant, et uniquement celui-ci. Il est donc nécessaire que le suiveur puisse identifier les robots à portée, et les différencier pour suivre le bon. Pour cela, chaque robot émet un signal IR dont la fréquence est propre à son identifiant. Il est important d'éviter les interférences entre ces signaux, et qu'ils ne saturent pas l'environnement de détection.

Emission

Les signaux infrarouges émis par les robots sont donc des pulses d'une durée d'une milliseconde, espacés d'une période d'au moins une dizaine de milliseconde. De cette manière, l'émission IR ne saturera pas l'environnement d'ondes infrarouges, même si plusieurs robots émettent en même temps, au même endroit.



Figure 26 : Signal infrarouge émis

Afin d'éviter les interférences, les différentes fréquences d'émission ont été choisies de façon à ce qu'elles soient peu corrélées. Voici le tableau des fréquences d'émission et de réception, en fonction de la place du robot dans le train.

Place / Identifiant	Période d'émission (ms)	Période de réception (ms)
0 (tête)	13	/
1	21	13
2	33	21
3	41	33

Figure 27 : Tableau des fréquences d'émission et de réception des ondes infrarouges selon les robots

Détection

L'intensité infrarouge vue par un phototransistor est composée :

- Des pulses « attendus », donc situés à la bonne période de détection
- De pulses parasites, émis par d'autres robots
- De l'éclairage et du bruit ambiant

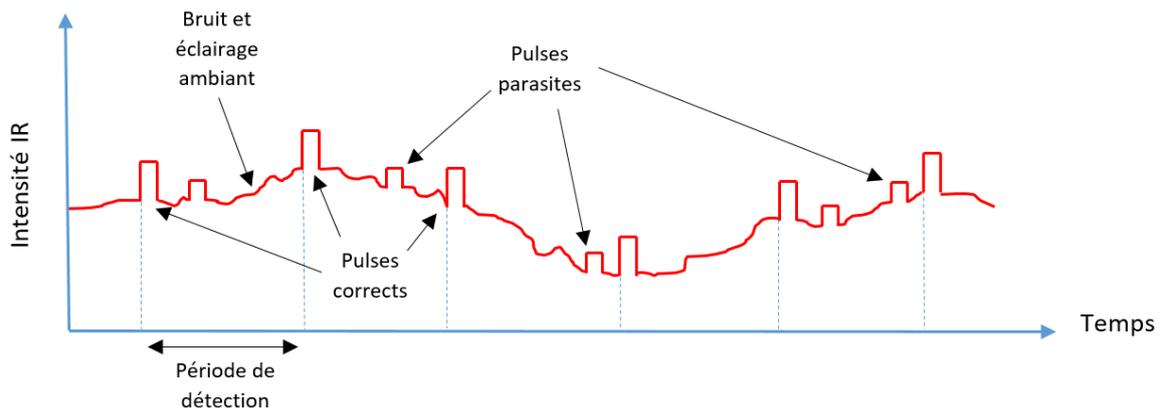


Figure 28 : Signal infrarouge reçu

L'objectif ici est d'extraire l'information afin de valider ou non la présence de pulses « corrects ». La détection de ces signaux se fait par l'analyse des fronts montants des pulses.

Un front montant est déduit de la différence entre l'intensité à l'instant t et celle à l'instant $t-\Delta t$. Si cette différence est supérieure à un certain seuil, alors un front montant est détecté. Cette méthode permet de limiter l'influence de l'éclairage ambiant.

Voici l'algorithme simplifié permettant de détecter la présence du signal attendu dans le signal IR reçu :

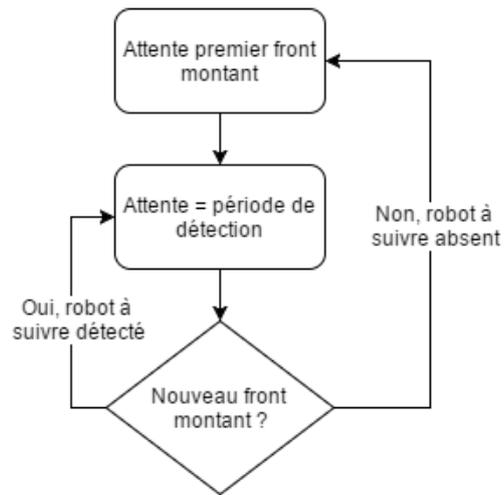


Figure 29 : Algorithme de détection du signal attendu

Cette méthode permet de se concentrer uniquement sur les pulses intéressants, et ainsi ignorer les pulses parasites. Elle est néanmoins sensible aux bruits parasites de haute fréquence.

3. Difficultés rencontrées

Au cours de ce projet, nous avons rencontré un certain nombre de difficultés :

- Souder les composants montés en surface (CMS) a été un véritable challenge. Leur taille minuscule, qui permet de disposer plus de composants sur une carte électronique, rend la soudure à la main compliquée. Nous avons essayé d'autres méthodes, avec des pâtes et un four, mais la méthode traditionnelle reste la plus efficace, bien que laborieuse.
- La réception infrarouge a été la partie la plus complexe à mettre en œuvre. En effet, l'éclairage ambiant a une influence importante sur la forme des signaux captés par le phototransistor. De plus, il y a un risque d'interférence lorsque plusieurs robots émettent au même endroit. Nous avons néanmoins réussi à pallier ces problèmes en trouvant des solutions adaptées. La faiblesse de la communication infrarouge reste sa sensibilité aux bruits de haute fréquence.
- Bien qu'assistée par l'Arduino IDE, la programmation de l'ESP8266 a posé quelques difficultés. Le téléversement des programmes sur le microcontrôleur présente très souvent des erreurs et des échecs. De nombreux facteurs sont susceptibles de le faire échouer. La programmation du robot de tête s'en est trouvée particulièrement ralentie.
- La puissance délivrée par les piles nous a aussi posé de gros problèmes. En effet, selon leur niveau de charge, les piles ne délivraient pas suffisamment de puissance aux moteurs pour faire tourner les roues. Le robot pouvait donc rester statique, ou une seule roue pouvait tourner. Pour un fonctionnement optimal, les piles devaient donc être complètement chargées.
- Malgré la qualité de la puce Wifi embarquée dans l'ESP8266, la connexion Wifi entre l'ordinateur et l'objet connecté est souvent coupée, lorsque l'environnement est saturé d'ondes 2,4GHz. Dans ce cas, le robot de tête devient incontrôlable.

Finalement, ces difficultés ont été surmontées ou contournées, et notre projet a pu être finalisé.

4. Améliorations possibles

Bien que le projet soit fonctionnel, quelques améliorations sont possibles, afin de rendre son utilisation plus performante ou plus confortable :

- L'interface Web développée actuellement est purement fonctionnelle et donc très rudimentaire. Une interface plus conviviale pourrait être développée.
- Afin de faciliter l'accès au contrôle du robot de tête, il serait possible de développer une application smartphone conviviale et simple d'utilisation pour commander le train de véhicules.
- Les ondes ultrason se réfléchissent sur une surface plane. Par conséquent, quand le robot n'est pas aligné avec celui à suivre, les ondes se perdent et on ne peut pas connaître la distance qui les sépare, et donc la garder constante. Il faudrait une surface de réflexion plus arrondie sur les extrémités.
- Les déplacements des robots dépendent de l'état de la batterie, de l'état des roues et de la position de la roue folle. Un asservissement en vitesse permettrait de corriger le problème et d'avoir des trajectoires plus propres. La possibilité d'utiliser des codeuses pour l'asservissement est déjà intégrée aux cartes électroniques fabriquées.
- La puissance des robots dépend de la charge des piles. Si on utilisait une pile 9V et un régulateur 6V, au lieu de 4 piles 1,5V, on aurait une tension constante, et donc une puissance constante.

Conclusion

Ce projet a été effectué dans le cadre de notre quatrième année dans la formation Informatique Microélectronique Automatique de Polytech Lille.

L'objectif était de développer un train de petits véhicules roulants, dont le véhicule de tête est contrôlé à distance, et les autres le suivant de manière autonome. Nous avons pu développer plusieurs prototypes fonctionnels : un robot de tête et deux robots suiveurs. Malgré les difficultés rencontrées lors du développement, le projet s'est déroulé selon les prévisions, et nous avons pu réaliser un train de véhicules opérationnel.

Afin de mener à bien ce projet, nous avons mis en pratique les connaissances acquises lors de notre formation d'ingénieur, mais aussi développé de nouvelles. D'une part de manière autodidacte, et d'autre part grâce à l'aide apportée par nos encadrants, nos enseignants et nos collègues.

Ce projet nous aura donc apporté beaucoup au niveau de nos compétences techniques et pratiques. De plus, nous avons travaillé la plupart du temps en autonomie, tout en maîtrisant la gestion du temps et de notre projet en général.

Le train de véhicule est un concept présentant un potentiel intéressant dans les milieux de l'industrie, du convoyage de marchandises ou encore dans les transports. On peut imaginer ce système dans les usines ou les stocks, permettant de convoier de manière autonome de grandes quantités de marchandises, tout en ne contrôlant qu'un seul véhicule. L'avantage principal par rapport à un train classique ou un système filoguidé est la flexibilité des déplacements de ces véhicules dans des environnements évolutifs et exigus. De plus, les coûts des infrastructures de guidage nécessaires est réduit. A terme, ce système pourrait être utilisé dans le secteur des transports routiers, dans l'objectif d'améliorer les aspects de confort et de sécurité.