INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Projet de Physique P6

STPI/P6/2019 – 21

Robot suiveur de ligne 1 pour match de vitesse contre robot suiveur de ligne 2

Étudiants :

Fanny HERRADA

Robin PANVERT

Rémi GUEGAN

Louis CHARTON

Enseignant-responsable du projet :

Fabrice DELAMARE

Cette page est laissée intentionnellement vierge

Date de remise du rapport : **17*/06/2019***

Référence du projet : ***STPI/P6/2019 – 21***

Intitulé du projet : ***Robot suiveur de ligne 1 pour match de vitesse contre robot suiveur de ligne 2***

Type de projet : ***Technologie***

Objectifs du projet :

Réaliser un robot capable de suivre une ligne le plus rapidement possible.

Mettre en valeur le travail d'équipe et les compétences de chacun

Découvrir et expérimenter la robotique ainsi que le codage d'une carte arduino

Utiliser les compétences et les connaissances acquises pendant le cycle STPI

Mots-clefs du projet : ***Robot, programmation, conception***

TABLE DES MATIÈRES

NOTATIONS, ACRONYMES

Arduino : C'est une marque de cartes électroniques en matériel libre. Mais le terme est plus souvent utilisé pour parler directement de la carte. Il s'agit d'un circuit imprimé qui peut être programmé à recevoir et envoyer des signaux électrique. Puisque c'est en matériel libre, les plans des cartes Arduino sont disponibles gratuitement. Le site d'Arduino laisse aussi à disposition des informations et tutoriels pour apprendre à programmer les cartes. Dû à leur accessibilité et leur prise en main relativement simple, ces cartes permettent l'élaboration de projet en tous genre. Dans notre cas elles nous permettront de créer notre robot suiveur de ligne.

Shield : c'est une carte électronique similaire à l'Arduino qui vient se placer au dessus de celle-ci. Chaque shield possède des fonctionnalités spécifiques, tout en laissant les entrées et sorties de l'Arduino accessibles. Il en existe de toute sorte : shield moteur, wifi, audio... On peut les considérer comme des extensions à l'Arduino. Ils permettent généralement de simplifier les montages et la programmation. Une carte Arduino peut supporter plusieurs shields en même temps.

1. INTRODUCTION

Ce travail a été réalisé dans le cadre du projet P6, projet interdisciplinaire visant à regrouper des élèves ingénieurs autour d'un projet.

Les objectifs étaient multiples.Tout d'abord, ce projet nous a amené à utiliser toutes les connaissances acquises pendant la STPI et pendant notre cursus scolaire en général concernant une grande variété de domaines. Ensuite,il nous a permis de travailler en équipe et en autonomie.

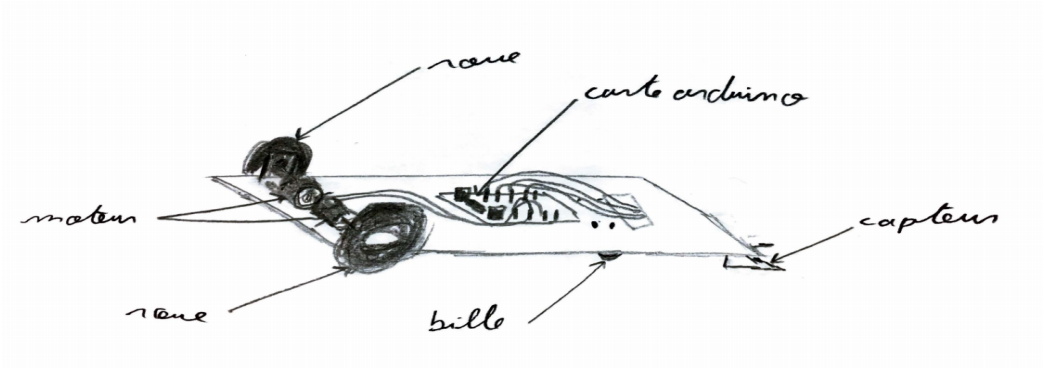
Nous avons travaillé sur ce projet durant un semestre entier, ce qui nous a permis de découvrir et d'évoluer.

Dans ce cadre, notre projet consistait à concevoir intégralement un robot suiveur de ligne. Notre appareil devra affronter le robot d'une deuxième équipe dans une course de vitesse sur un parcours réalisé par nos soins.

Le but de ce projet était donc la création d'un robot le plus performant possible grâce au matériel mis à notre disposition.

En outre, la robotique comprend plusieurs domaines scientifiques, tels que l'informatique, la mécanique ou encore l'électronique. La diversité de notre groupe; composé de 4 membres, a permis à chaque personne de se concentrer sur le domaine qu'elle maîtrisait le plus.

La communication et la cohésion ont donc été des éléments nécessaires à la réalisation d'un tel projet. De même, au fil des semaines, nous avons acquis de nouvelles connaissances, ce qui nous a permis d'améliorer le robot avec des idées novatrices.



2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Nous nous sommes réparti le travail en fonction de nos compétences dans les différents domaines :programmation, modélisation et montage.

Ainsi nous pouvons présenter la répartition des tâches selon l'organigramme suivant :

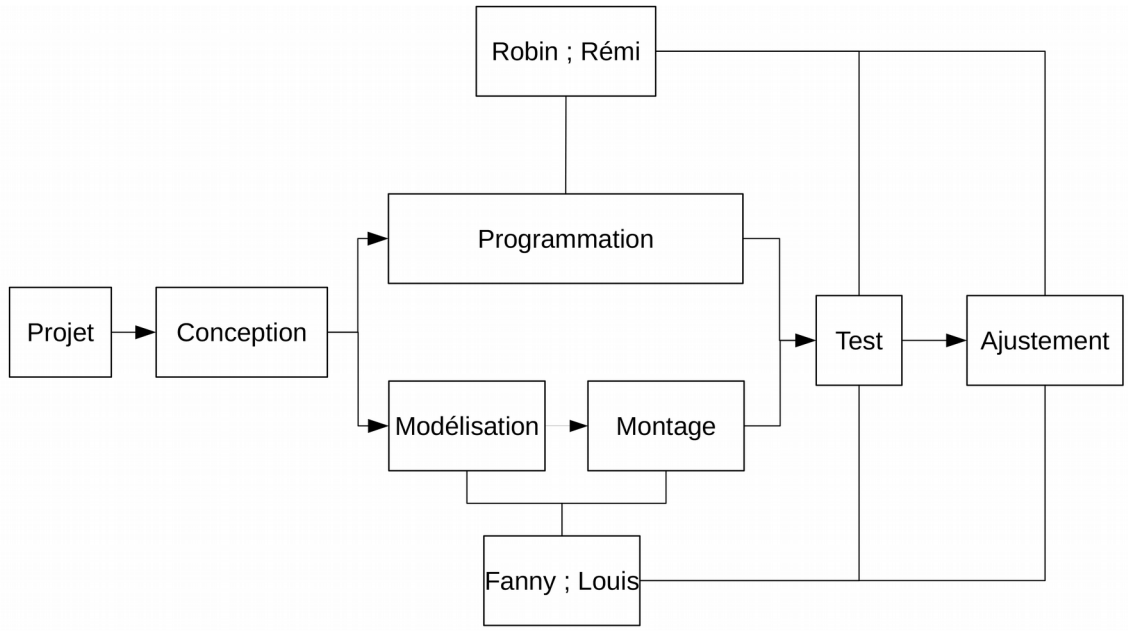


Illustration 1: Répartition du travail

Ainsi comme nous pouvons le voir, Rémi et Robin se sont plutôt concentrés sur la partie programmation, ils ont crée un programme et ont effectué des tests réguliers afin d'améliorer les performances du robot au maximum.

Louis et Fanny quand à eux, ont réalisé la modélisation et le montage, contenant les domaines de l'électronique et de la mécanique. Ils ont réalisé la structure du robot ainsi que les différents branchements électriques.

Cependant, tous les membres du groupe se réunissaient à chaque séance afin de s'entraider, de tester ou même de conseiller, car il s'agit avant tout d'un travail de groupe.

Nous avons ainsi effectués de nombreux tests et de nombreuses améliorations qui seront détaillés plus bas.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. Phase de réflexion (Brain storming)

Avant toute étape de modélisation ou de montage, nous avons imaginé à quoi pourrait ressembler le robot.

Dans ce but, nous avons étudié différentes solutions afin de déterminer les plus adaptées à ce projet.

3.1.1. Position des éléments

Dans un premier temps, nous avons réfléchi au positionnement du capteur. Deux solutions s'offraient à nous : positionner le capteur à l'avant du robot ou le positionner à l'arrière, sachant que l'une ou l'autre solution impliquait la solution inverse pour les roues motrices.

Nous avons décidé de placer le capteur à l'avant pour permettre de capter l'obstacle dés le début et donc de pouvoir faire réagir les roues bien plus vite. Ainsi le positionnement de la bille, qui permet de stabiliser la structure, s'est imposé sur le milieu avant du robot.

De plus, la carte Arduino a été placée de telle sorte que son centre de gravité soit positionné sur le centre du bâti, afin d'éviter que le robot ne soit déséquilibré.

Le robot étant alimenté par une prise, une batterie n'a pas été utile dans notre réflexion.

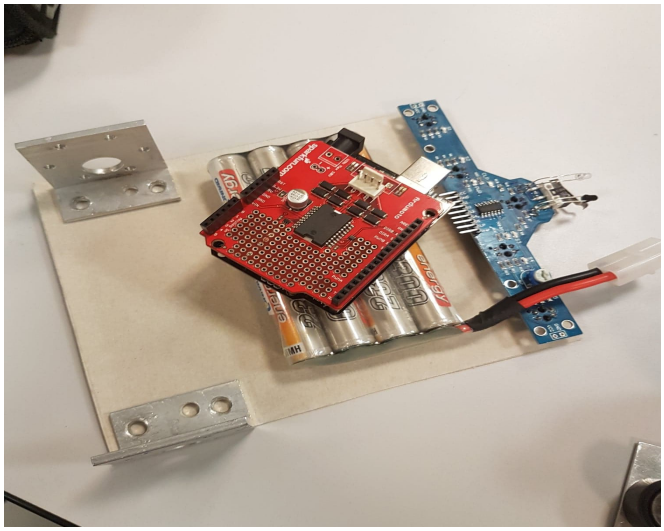


Illustration 2: Placement des éléments

3.1.2. Choix des matériaux

Selon les matériaux choisis, la forme du robot sera différente. En effet, l'utilisation de CD aurait donné une forme circulaire au robot, car ce matériau ne peut changer de forme. Nous avons donc choisi l'option du médium de 3 mm.

En effet, ce matériau plutôt léger est assez maniable et peut surtout prendre une infinité de formes, ce qui est avantageux dans la création de ce genre de robot.

3.2. Mécanique

3.2.1. Modélisation

Après avoir choisi la position des éléments et le type du matériau, nous avons réalisé une esquisse du robot.

Dans ce but, nous avons réalisé le bâti sur une plaque en carton afin de déterminer les dimensions du produit fini.

Pour cela, nous avons placé tous les éléments du robot (moteurs,carte etc...) afin de définir les perforations nécessaires à la mise en place des éléments sur la bâti en bois. Nous avons ainsi réalisé de nombreuses mesures, afin de déterminer les dimensions des perforations, les distances nécessaires entre les éléments afin de ne pas créer d'interférences etc.

Nous avons également pensé à créer une ouverture au milieu de la plaque du robot afin de pouvoir effectuer les branchements nécessaires. Cette étape était la première étape vers la modélisation.

Ensuite, nous avons tout simplement créé un fichier informatique qui reprenait les dimensions de notre esquisse.

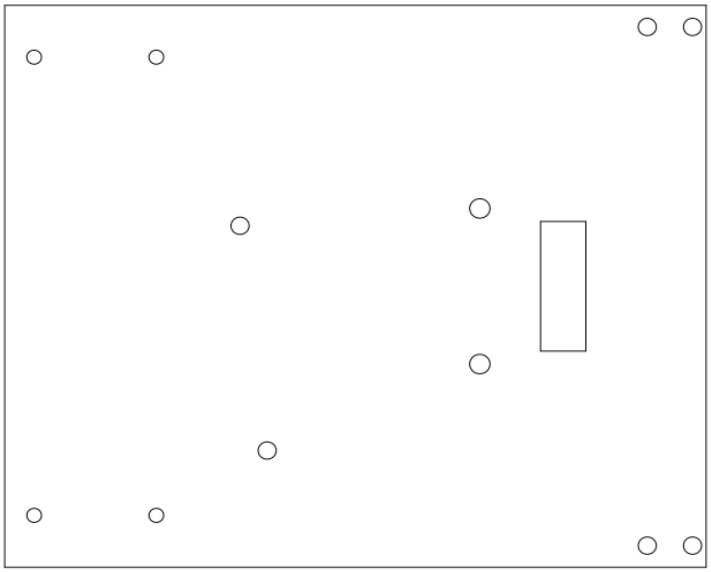


Illustration 3: Capture d'écran du fichier

3.2.2. Montage

Grâce à ce fichier, nous avons pu découper la structure de notre robot dans une plaque de médium. Nous avons alors ensuite effectué le montage du robot, en fixant tous les éléments : roues, capteur, carte Arduino, bille. Puis nous avons effectué les branchements électriques afin de lier les différents composants.

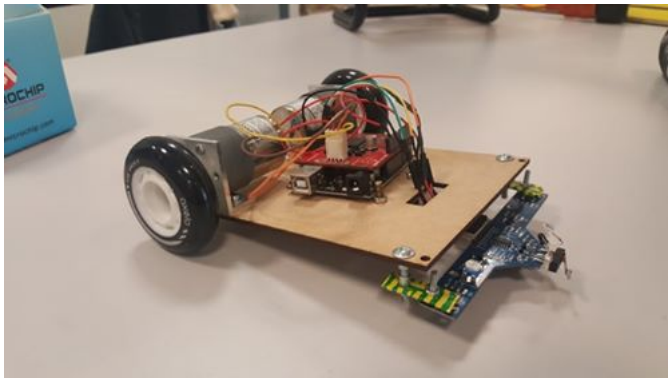


Illustration 4: Photo du robot final

3.3. Programmation

3.3.1. Prise en main du logiciel Arduino et des composants

Avant de commencer à coder, nous avons du apprendre à nous servir du langage arduino, grâce à des sites internet tel qu'Openclassroom et le site officiel d'Arduino.

Ce langage est spécifique et permet de réaliser des programmes lorsqu'on utilise des cartes arduino. Dans notre cas, il s'agit d'une carte Arduino Uno.

Concernant les composants à notre disposition nous avions des capteurs, un shield et des moteurs. Nous avons tout d'abord réalisé des tests pour comprendre ce que les capteurs renvoyaient comme information en fonction des couleurs qu'ils captaient.

Lorsqu'un des capteurs capte du noir il renvoie la valeur numérique 0 et le booléen faux, sinon il renvoie la valeur 0 et le booléen vrai. C'est pendant ces tests que nous avons remarqué que les capteurs n'étaient sensibles qu'à de bandes noires assez larges et mates. Nous nous sommes ensuite penchés sur les branchements à effectuer pour que ces informations soient transmises au shield et à la carte arduino uno, afin qu'elles puissent être utilisées par le programme.

Le shield permet de faciliter l'utilisation de moteur. Il est possible de sélectionner la direction du moteur, à l'aide d'un pont en H intégré (montage électrique permettant d'inverser le sens du courant) et de régler la puissance électrique envoyée dans le moteur afin de contrôler sa vitesse de rotation. Les moteurs dont nous disposions étaient des moteurs à courant continu. En réalité on ne peut pas vraiment modifier la puissance électrique qu'on leur envoie.

Toutefois il existe des sorties sur la cartes Arduino (signaler par « ~ ») pouvant générer des impulsions ; ce sont leurs fréquences pendant une période (de 2 ms) que nous pouvons modifier à l'aide d'une fonction prenant des valeurs entre 255 et 0. Sachant que 225 envoie du courant en continu et 0 le contraire. Selon les fréquences d'impulsions les moteurs tourneront à des vitesses plus ou moins importantes. Après avoir recueilli toutes ces informations nous avons commencé l'écriture du programme.

3.3.2. Écriture du programme

(Programme en annexe)

Le programme est divisé en trois parties. La première partie est la déclaration des différentes constantes de vitesses, des variables correspondant aux valeurs que renvoient les capteurs en temps réel ainsi que les sorties du shield sur lesquelles les capteurs et le moteur sont reliés. Toutes ces variables ou constantes sont des entiers.

Ensuite il y la partie « setup » du programme. Cette partie consiste à indiquer quelles seront les variables d'entrées et de sorties de la suite du programme. Cela indique à la carte quels sont les branchements qui doivent recevoir ou envoyer du courant. Dans notre cas, les variables en entrées sont les informations renvoyées par les capteurs et nos variables de sorties sont les vitesses, ainsi que la direction qui seront attribuées à chaque moteur en fonction du circuit à suivre.

La dernière et principale partie du programme est la fonction « loop », cette fonction comme l'indique son nom se répète indéfiniment jusqu'à ce que l'on cesse d'alimenter le système. Elle est elle même divisée en deux parties : l'initialisation puis les instructions et opérations.

Dans l'initialisation nous avons tout d'abord attribué aux variables déclarées plus tôt les valeurs numériques renvoyées par les capteurs, puis nous avons initialisé les vitesses et les directions des moteurs. L'utilité de cette initialisation à chaque début de boucle est de permettre à la carte de mettre constamment à jour les valeurs que renvoient les capteurs. Sans cette étape la carte Arduino n'aurait enregistré que les valeurs des capteurs lors du démarrage du programme sans jamais les changer.

Enfin il y a les instructions, qui permettent au robot d'avancer en suivant le tracé. Nous voulions que les mouvements de notre robot soient fluides, c'est pourquoi nous avons décidé de faire une fonction « if » pour toutes les combinaisons possibles de capteurs qui captent en même temps le tracé noir. L'exécution très rapide de l'entièreté de la boucle et l'utilisation de la conditions « if » permet une réaction rapide du programme au moindre changement dans le circuit. En effet chaque condition est quasiment testée en permanence et la trajectoire rectifiée à chaque instant.

Il y a donc cinq types de virages : allant du très gros virage, lorsque le capteur à l'extrême droite ou celui à l'extrême gauche sont sur le tracé ; à la ligne droite où seul le capteur central est sur la ligne noire. Pour faire tourner le robot dans le sens du virage, nous avons diminuée la vitesse de la roue qui est à l'intérieur du virage tout en accélérant celle de la roue extérieure. La différence de vitesse des deux roues étant proportionnelle au rayon de courbure du virage. En effet plus le virage est serré plus l'écart des vitesses est élevé, et inversement.

3.4. Tests

3.4.1. Tests pour la réalisation du circuit

En premier lieu, nous avons testé notre capteur sur un petit circuit afin de déterminer la largeur idéale de la bande noire à peindre.

Il fut alors nécessaire de réaliser le circuit final.

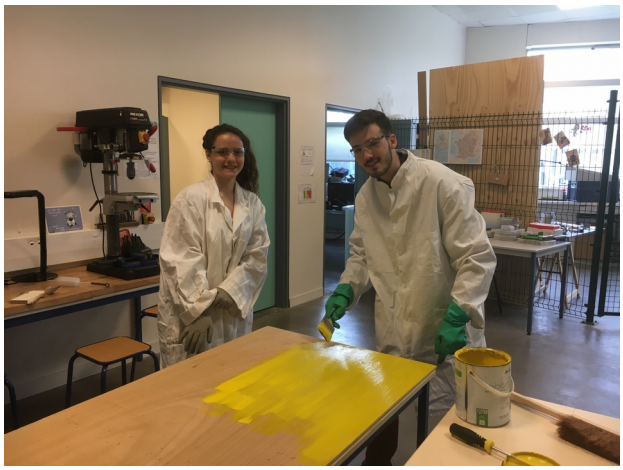


Illustration 6: Réalisation du circuit

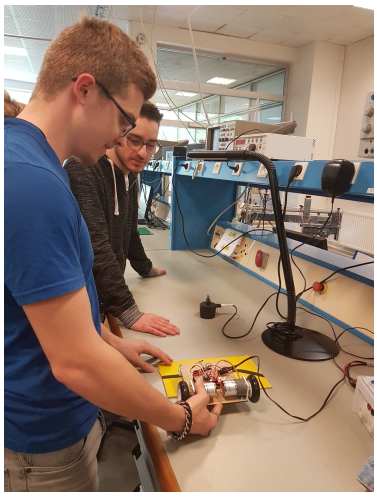


Illustration 5: Test sur petit circuit

Afin de réaliser la base de ce circuit, nous avons tout d'abord poncé une planche en bois, dans le but de faciliter le passage du robot et d'éviter les irrégularités. Ensuite, nous avons intégralement peint la planche en jaune. Cela permet de créer un contraste avec la couleur noire du circuit.

Puis, nous avons réalisé à l'aide de l'autre groupe, une esquisse du circuit qui permettra de déterminer le robot le plus rapide entre les deux.

En dernier lieu, nous avons testé les aptitudes du capteur à différencier le noir du jaune sur le circuit final.

3.4.2. Tests sur le programme

Lors du premier test de notre programme complet, nous avions réglé les moteurs à des vitesses très lentes. Notre robot ayant réussi à compléter le circuit, nous avons alors augmenté les vitesse, mais le robot sortait de la piste à chaque fois dans le même virage.

Nous pensions que le robot allait trop vite pour que les capteurs aient le temps d'ordonner au robot de tourner. En réalité les capteurs captaient bien la présence du virage, mais nous l'avions compris qu'en répétant plusieurs fois les tests. Le problème venait donc des indications données aux robots.

Ne comprenant pas d'où venait le problème, l'idée nous est venue que même si le robot sortait du virage, il serait utile qu'il « se souvienne » qu'il était dans un virage. Il nous fallait donc créer une mémoire.

C'est en cherchant comment réaliser cette mémoire que la solution à nos deux problèmes fut trouvée. En effet, au départ, dans l'initialisation de notre boucle, nous demandions aux moteurs de rouler à une vitesse moyenne. Cela avait deux buts. Le premier était de réinitialiser la vitesse des moteurs à une vitesse dite « par défaut » en attendant la prochaine mise à jour des capteurs et des instructions. Le second but était de ne pas perdre du temps en arrêtant les moteurs. Mais dans cette configuration, si le robot était dans un virage, il avait alors comme instruction : d'aller tout droit, puis de tourner, puis d'aller tout droit, puis de tourner… etc. Le virage ne pouvait donc pas être effectué correctement et le robot quittait la piste à chaque fois.

En réinitialisant les moteurs avec la dernière instruction donnée,les virages pouvaient être terminés. De plus si les capteurs ne détectent pas de noir le robot n'aura pas de nouvelles instructions et réalisera la dernière qu'il a eu. Ainsi, si le robot sort de la piste durant un virage, sa dernière instruction était de tourner, il continuera de tourner jusqu'à regagner la piste. Nous avons donc créer notre mémoire. Après plusieurs optimisations des vitesses, nous sommes arrivé à une version finale du code où le robot terminait le circuit en 19,4 secondes.

3.5. Ajustements

3.5.1. Changement de l'orientation des capteurs

Les capteurs sont fixés à l'aide de petites plaques en métal reliées à la plaque de médium, et ses plaques en métal étaient fixées de façon parallèle au bâti.

Comme les roues arrières sont plus basses que la bille, la plaque de médium est inclinée, donc les capteurs n'étaient pas parallèles au sol. Ainsi, la détection du parcours n'était pas optimale. Nous avons donc tordu les plaques en métal pour que les capteurs soient parallèles au sol, réglant ainsi le problème de détection.

3.5.2. Remplacement d'un capteur défectueux

Nous avons remarqué qu'un des 5 capteurs n'était plus apte à détecter les changements de couleur. Nous l'avons donc changé, en le dessoudant tout d'abord, puis en ressoudant un nouveau capteur.

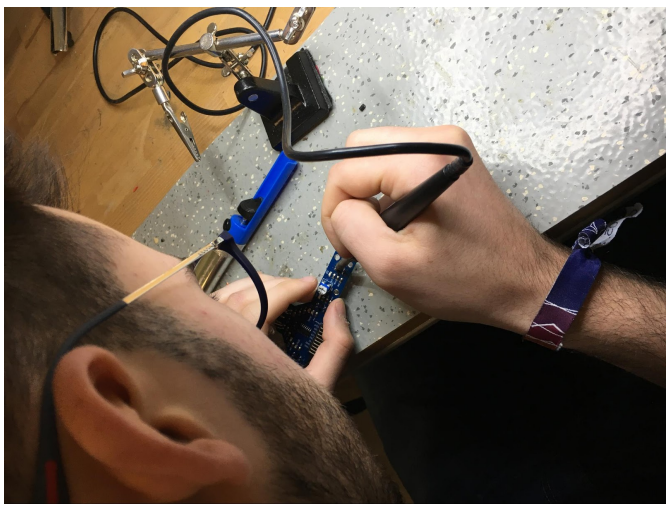


Illustration 7: Remise en place du capteur

4. CONCLUSIONS ET PERSPECTIVES

Pour conclure, nous avons donc réalisé un robot suiveur fonctionnel, établissant le tour du parcours en 19,13 s.

Cet EC nous aura ainsi beaucoup apporté, concernant de nombreux domaines. Techniquement, nous avons appris comment fonctionne le système Arduino: programmation, branchement, alimentation.Nous avons également eu recours à de nouvelles notions d'électronique.De plus, nous avons découvert la robotique, un domaine inconnu pour la majorité du groupe.

Ce projet nous a également appris à travailler en groupe et à se servir des connaissances et des habilités de chacun dans un but commun. De plus,pendant la durée de ce projet , nous avons dû faire face à de nombreuses difficultés,que nous avons résolues ensembles.

Concernant les perspectives pour la poursuite de ce projet, pour améliorer le robot et pouvoir aller plus vite, il serait intéressante de modifier les moteurs. En effet, ceux-ci sont désynchronisés: le moteur droit est plus lent que le moteur gauche, sûrement dû à un défaut dans le moteur. Ce défaut ne nous a pas posé de gros problèmes car le programme corrigeait ses défauts automatiquement. De plus,une limite de voltage due au système de la carte ne nous a pas permis de faire fonctionner les moteurs à pleine puissance

5. BIBLIOGRAPHIE

[1] lien internet :https://www.arduino.cc/reference/en/ (valide à la date du 11/06/2019).

[2] lien internet :https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino (valide à la date du 16/06/2019).

[3] lien internet :https://openclassrooms.com/fr/courses/3290206-perfectionnez-vous-dans-la-programmation-arduino (valide à la date du 16/06/2019).

[4] lien internet :https://sti2d.ecolelamache.org/i_dcouverte_du_robot_et_de_sa_carte_de_commande_motor_driver_shield.html (valide à la date du 16/06/2019).

6. ANNEXES

6.1. Documentation technique

Capteur BFD 1000

6.2. Listings des programmes réalisés