Gestion de projet

**Rapport gestion de projet**

Année Académique 2022-2023

Groupe N°9

Rosin Guillaume

Danneau Thibaud

Yepgwa Takedo Fabiola

Wanji Franck







### 

### Table des matières

[Table des matières 3](#_bookmark0)

1. [Introduction 4](#_bookmark1)
2. [Présentation générale du projet 4](#_bookmark2)
3. [Présentation détaillée des composants 5](#_bookmark3)
   1. [Les capteurs ultrason 5](#_bookmark4)
   2. [Les capteurs de ligne 6](#_bookmark5)
   3. [Le Raspberry PI 7](#_bookmark6)
   4. [Le servo-moteur 7](#_bookmark7)
4. [Algorithmes de programmation (points principaux) 8](#_bookmark8)
5. [Problèmes rencontrés 9](#_bookmark9)
6. [Améliorations et conclusions 9](#_bookmark10)
7. [Bibliographie 10](#_bookmark11)



# Introduction

Lors de la dernière semaine du mois d’avril 2022, nous avions du configurer une voiture autonome à l’aide d’un Raspberry Pi 3 munis d’équipement fournis par l’école qui sont :

* Une voiture avec un Raspberry Pi 3.
* 3 capteurs Ultrasons (2 sur les côtés et 1 devant).
* 1 capteur infrarouge situé à l’avant en-dessous de la voiture.
* 2 moteurs à courant continus et un servo moteur.
* 4 piles rechargeables ainsi qu’un chargeur.

Quant à l’équipe, nous étions répartis dans des groupes de 5 personnes dont 2 étant en options sécurité, 2 autres en options développement, ainsi qu’une étudiante de première pour notre groupe.

* Debève Nicolas et Vierendeels Sébastien sont les 2 étudiants en option sécurité
* Alcantarini Matteo et Depelsenaire Colin sont les personnes étant en option développement.
* Telle Justine est une élève en BAC1.

Nous avons effectué ce projet sous la méthode agile « scrum », méthode rigoureuse nécessitant de la communication et de l’organisation.

Durant cette semaine, nous avons été mis en situation réelle. Nous avons réalisé la voiture autonome pour des clients joués respectivement par nos professeurs Monsieur Desmet et Monsieur Michiels et nous avons pu compter aussi sur leur aide ainsi que Monsieur Pietrzak et Monsieur Depreter lorsque nous avons rencontrés des problèmes.

# Présentation générale du projet

Les objectifs attendus de cette semaine nous était fournis via un cahier de charges. Il consistait à développer un module mobile dans le but de réaliser une course de formule 1 sur un circuit (voir photo ci-dessous) définit par les enseignants.



Le module devra être capable d’effectuer certaines tâches données par le cahier de charges :

* Il sera capable d’effectuer au minimum un tour complet du circuit.
* Il devra être capable de s’arrêter après la ligne d’arrivée.
* Il sera possible de définir un nombre de tour aléatoire à réaliser pour une course.
* Il devra être capable d’éviter tout obstacle. (P. ex. des parois, autres

véhicules…)

* Il faudra optimiser la trajectoire au maximum afin de, bien évidemment, gagner la course.

Si nous terminons l’ensemble des demandes ci-dessus avant le deadline (vendredi 29 avril à 13h30), nous pouvons ajouter des bonus tel que :

* Départ automatisé du module.
* Pilotage du module en Bluetooth via une application Android.
* …

# Présentation détaillée des composants

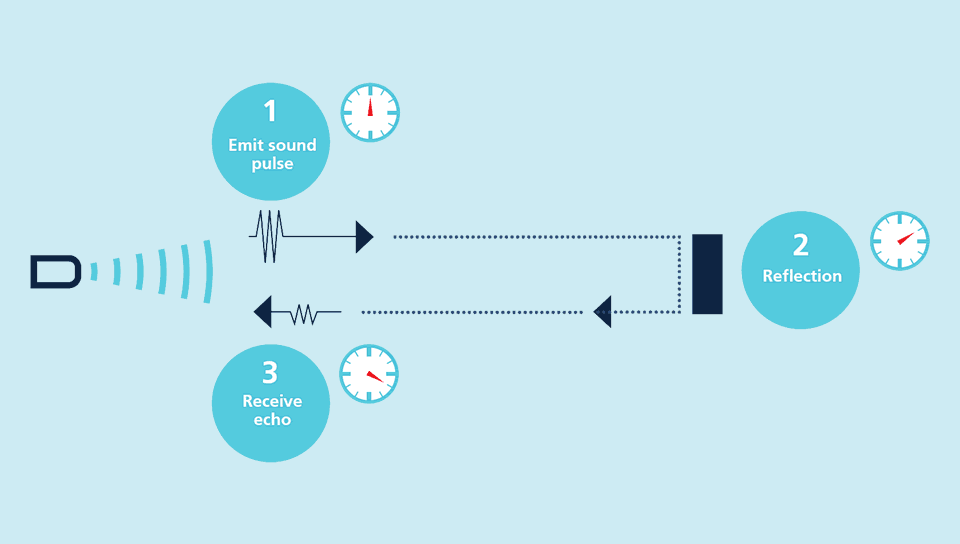
## Les capteurs ultrason

Pour permettre à la voiture d’éviter les obstacles et de s’arrêter lors de la course, ce dernier nécessite l’utilisation de différents capteurs.

Le premier est un capteur à ultrason, son principe est d’émettre à intervalles régulier de courtes impulsions sonores à haute fréquence. Celles-ci se déplacent dans l’air à la vitesse du son et lorsqu’elles rencontrent un obstacle, les impulsions se réfléchissent et reviennent sous forme d’écho au capteur. Celui-ci calcule alors la distance séparant l’objet du capteur

en fonction du temps écoulé entre l’émission et la réception de l’impulsion.

Le capteur utilise les ultrasons pour déterminer la distance d’un objet. Pour que le capteur se déclenche, il faut présenter une image pulsion high d’au moins 10 μs sur l’entrée Trig. Le capteur émet une série de 8 impulsions ultrasoniques à 40kHz puis il attend que le signal réfléchi. Quant à la distance de l’objet, elle se calcule en multipliant la vitesse du son (environ 340m/s) par le temps de propagation soit d=v.t ensuite, il faut multiplier la valeur obtenue par 10 μs pour obtenir le temps et diviser la valeur de la distance pour compter qu’un aller et pas un aller-retour. Concernant le brochage, il y a 4 broches une pour l'alimentation (Vcc), une pour le déclencheur (Trigger), une pour le récepteur (Echo) et une pour la masse (GND). Le déclencheur indique quand le capteur doit être activé (lorsque l'ultrason est lancé), et ainsi il sera possible de connaître le temps écoulé lorsque le récepteur reçoit le signal.





## Les capteurs de ligne

Le capteur dit infrarouge possède un fonctionnement basé sur la lumière réfléchie, le faisceau infrarouge est émis depuis le dôme bleu et le faisceau réfléchi retombe (pour le rendre passant) sur le dôme noir.

La LED infrarouge émet un rayonnement dans une longueur d’onde infrarouge non-visible à l’œil humain. Si un obstacle réfléchissant se trouve sur le trajet (ici un morceau de scotch sur le sol), il renvoie une partie de l’énergie qui tombe ensuite sur le phototransistor.

Le capteur TCRT5000 se compose d'une LED infrarouge et d'un phototransistor photosensible, et le même capteur est revêtu pour filtrer la lumière.

Ce capteur fonctionne en transmettant la lumière infrarouge d'une LED et en enregistrant la lumière réfléchie par un phototransistor. 3,3 V et 5 V sont fournis via la broche VCC et la broche de masse, et les données du capteur sont fournies via les deux broches restantes. La broche analogique A0 fournit une lecture continue, et plus la tension est élevée, plus la lumière infrarouge est reçue. L'inconvénient est qu'il peut être affecté par l'environnement dans lequel se trouve le capteur. La source lumineuse peut également fausser le capteur. Dans ce projet, nous utiliserons le TCRT5000 pour vérifier la présence d'objets physiques tels que des virages et des murs, mais nous pouvons également voir des couleurs sur une échelle allant du noir à l'uni. Comme une course, le capteur agit comme un suiveur de ligne



*LED infrarouge (de couleur bleu) et le phototransistor (de couleur noir)*

Celui-ci réagit comme un interrupteur commandé par la lumière infrarouge qu’il reçoit :

* Si la quantité de lumière a une valeur nulle ou inférieure à un seuil donné, le

transistor est bloqué, comme si l’interrupteur est ouvert.

* Cependant, si la lumière que reçoit le phototransistor est supérieure au seuil, le

transistor est passant, comme si l’interrupteur est fermé.



## Le Raspberry PI

La voiture fonctionne principalement avec le Raspberry Pi 3, c’est grâce à ceci qu’elle pourra avancer, reculer, communiquer avec les capteurs gérant les obstacles et compagnie.

Il est composé de 4 ports USB, un port HDMI, un connecteur d’E/S (entrée sortie) 40 broches et un port Micro-SD. Afin d’être en état de fonctionnement, il nécessite une alimentation et une carte SD munie d’un OS (bootable grâce à un lecteur de carte sur PC).

Une image contenant texte, équipement électronique, circuit

Description générée automatiquement

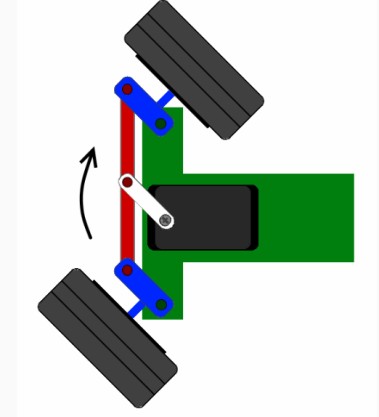
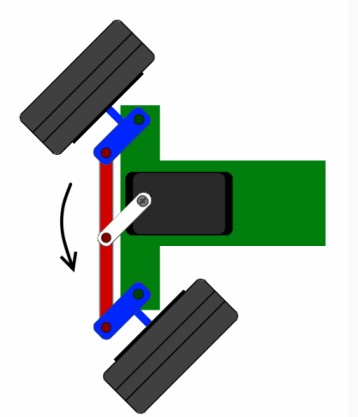
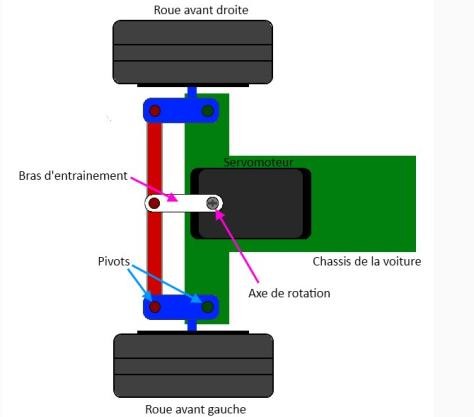
Fonctionnement du Raspberry Pi de la voiture :

* L'alimentation : Le Raspberry Pi sera alimenté par des piles 5V (fournies par l'école), via le port Micro-USB. La capacité de batterie recommandée pour est de 700mAh, celle-ci à une durée de vie dépendant de l'utilisation (le WiFi et le Bluetooth peuvent consommer beaucoup de ressources).
* Le stockage : Le système d'exploitation ainsi que les programmes et les scripts Python nécessaire au fonctionnement du Raspberry et du pilotage des composants de la voiture sont stockées sur une carte SD car le Raspberry ne possède pas de stockage intégré.
* Ethernet : Le Raspberry peut se connecter à un réseau via Ethernet, ce qui nous permets de configurer le Raspberry à plusieurs en même temps via une connexion à distance.
* Antenne WiFi : Grâce à la haute permittivité du matériau permet d'émettre et recevoir des données via la fréquence de la bande 2,4GHz tout en prenant un minimum d'espace sur le PCB.
* GPIO Headers : Grâce à ces pins d'entrées et sorties, nous pouvons connecter tout types de périphériques à notre Raspberry Pi de manière simple. Notre version du Raspberry Pi comporte 40 broches, dont 26 sont utilisées comme entrées ou sorties numériques.

## Le servo-moteur

Les servomoteurs nous sont utiles pour faire tourner les roues jusqu'à une certaines position précise et capable de maintenir cette position jusqu'à l'arrivé d'une nouvelle instruction. Les instructions se font à l'aide d'impulsion électrique.

Chaque roue est positionnée sur un axe de rotation (partie bleue) lui-même monté sur un pivot sur le châssis de la voiture (en vert). La baguette (rouge) permet de garder le parallélisme entre les roues. Si l’une pivote vers la gauche, l’autre en fait de même. Cette baguette est fixée, par un pivot encore, au bras de sortie du servomoteur. Ce bras est à son tour fixé à l’axe de rotation du servomoteur. Ainsi, lorsque le servomoteur fait tourner son axe, il entraine le bras qui entraine la baguette et fait pivoter les roues pour permettre à la voiture de prendre une direction dans son élan (tourner à droite, aller tout droit ou tourner à gauche). Ce qu’il faut retenir est que le servomoteur va entrainer la baguette pour orienter les roues dans un sens ou dans l’autre. Elles auront donc un angle d’orientation par rapport au châssis de la voiture



Blabla moteur continu

Le capteur TCRT5000 se compose d'une LED infrarouge et d'un phototransistor photosensible, et le même capteur est revêtu pour filtrer la lumière.

Ce capteur fonctionne en transmettant la lumière infrarouge d'une LED et en enregistrant la lumière réfléchie par un phototransistor. 3,3 V et 5 V sont fournis via la broche VCC et la broche de masse, et les données du capteur sont fournies via les deux broches restantes. La broche analogique A0 fournit une lecture continue, et plus la tension est élevée, plus la lumière infrarouge est reçue. L'inconvénient est qu'il peut être affecté par l'environnement dans lequel se trouve le capteur. La source lumineuse peut également fausser le capteur. Dans ce projet, nous utiliserons le TCRT5000 pour vérifier la présence d'objets physiques tels que des virages et des murs, mais nous pouvons également voir des couleurs sur une échelle allant du noir à l'uni. Comme une course, le capteur agit comme un suiveur de ligne



# Algorithmes de programmation (points principaux)

Dans notre code, nous avons 5 classes importantes telles que :

* InfraredSensor : pour compter les tours.
* Motor : pour que la voiture avance ou recule.
* Servo : afin que la voiture tourne.
* UltrasoundSensor : savoir à quelle distance elle se trouve par rapport à un point.
* Véhicule qui a pour but de rassembler les 4 classes ci-dessus afin que la voiture soit la plus autonome possible.

Nous avons aussi créé des fichiers tests (dans le dossier tests) afin qu’on puisse tester chaque méthode des classes ci-dessus. Il fallait être sûr que l’on puisse gérer chaque partie de la voiture distinctivement.

Dans InfraredSensor, il y a deux méthodes : isOn() et getLaps(). Elles permettent

respectivement de détecter la ligne de départ (qui est également la ligne d’arrivée) et qui, lorsque l’on passe sur la ligne, décrémente le nombre de tours ainsi qu’à retourner le nombre de tours qu’il reste et que lorsque celui-ci arrive à zéro, la voiture s’arrête.

Dans Motor, les méthodes importantes sont moveForward(), moveBackward() et stopMove(). Ces méthodes permettent de faire bouger la voiture (en avançant ainsi qu’en reculant) avec une vitesse de base de 25. Celle-ci est modifiable en passant en paramètre des deux premières méthodes, une valeur afin d’ajuster la vitesse (allant de 0 à 50). La

méthode stopMove() fera que la voiture s’arrête instantanément (par exemple, lorsque le

nombre de tours est égal à zéro).

Dans Servo, nous avons 2 méthodes : setAngle() et setAngleByDist(). Elles permettent

respectivement de régler un angle que l’on a choisi nous-même (entre 70 et 95) ainsi que de régler un angle en fonction d’une distance (distance donnée par les capteurs ultrasons) afin que la voiture puisse effectuer un angle par elle-même et donc d’avoir la meilleure trajectoire possible.

Dans UltrasoundSensor, une seule méthode et c’est la plus importante : getDistance(). Comme ce sont des capteurs ultrasons, ils ne nous donnent pas directement une distance. Nous avons donc dû créer une méthode qui nous retourne la distance en centimètre. Afin de récupérer la distance, nous avons enregistré le temps entre l’onde de départ et l’onde

d’arrivée. Il ne fallait plus que convertir le temps qu’il s’est écoulé en centimètre.

Chaque capteurs et moteurs sont importants. Pouvoir gérer chaque partie distinctivement,

c’est bien, mais lorsqu’on les ensembles toutes, nous pouvons réaliser une voiture totalement autonome, c’est pour cela que la classe Vehicle existe.



# Problèmes rencontrés

Nous avons rencontré pas mal de problème durant cette semaine de projet.

Nous avons rencontré une erreur avec le code et avons passés 2h avec les Prog master afin de trouver d’où venait le problème, ce dernier venait du fait qu’on avait instancié 2 fois la classe Motor.

Les pins des moteurs ne se mettait pas en OUT alors que nous l’avions précisé dans le constructeur, on a donc trouvé une alternative de créer une méthode qui initialise les pins en OUT.

Nous avons brulé un capteur à ultrasons en inversant le VCC et le ground lors du câblage. Nous avions pourtant bien appliqué la correction que Mr Michiels avait apporté à son schéma de câblage sur Teams mais il s’est avéré que sa correction était mauvaise.

Les capteurs à ultrasons indiquent lorsqu’elles doivent longer le mur une détection murale de 20m (ce qui n’est pas normal car le capteur à une capacité de détecter un obstacle à 4m selon la datasheet)

# Améliorations et conclusions

Cette semaine de projet fut enrichissante malgré certains soucis en interne, nous avons commis des erreurs et avons su en rebondir. Nous savons que lors des prochaines utilisation de la méthode agile « scrum », il faudra poser les bonnes questions aux clients et ne pas forcément suivre ses changements entre les sprint review.

Même si nous n’avons pas accomplis tous les users stories, notre véhicule a pu faire un tour complet du circuit et donc participer à la course avec les autres mobiles (et nous l’avons malheureusement perdu).



# Bibliographie

Lien Hypertexte : Le Guide ScrumTM, Ken Schwaber et Jeff

Sutherland disponible sur : https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum- Guide-French.pdf (consulté 25/04/2022)

Référence à un manuscrit : NAIZY G., Gestion de projet informatique : SCRUM

Lien Hypertexte : REDDIT, [HOWTO] Setup your Pi2b/3 with no monitor/keyboard/mouse using a spare ethernet port on your PC/Mac, [u/phattmatt](https://www.reddit.com/user/phattmatt/), disponible sur : <https://www.reddit.com/r/raspberry_pi/comments/4t9xys/howto_setup_your_pi2b3_with_no/> (consulté 25/04/2022)

Lien Hypertexte : Raspberry Pi FR, Connectez-vous en SSH à votre Raspberry Pi pour la contrôler depuis votre ordinateur, disponible sur : <https://raspberry-pi.fr/connecter-ssh-raspberry-pi/> (consulté 25/04/2022)

Lien Hypertexte : Raspberry Pi, Install Raspberry Pi OS using Raspberry Pi Imager, disponible sur : <https://www.raspberrypi.com/software/>(consulté 25/04/2022)

Lien Hypertexte : SunFounder, documentation, disponible sur :

<https://learn.sunfounder.com/>(consulté 25/02/2022)

Lien Hypertexte : OpenClassrooms, Développez un robot mobile connecté par Bluetooth, Bertrand Boyer, disponible sur :

[https://openclassrooms.com/fr/courses/5224916-developpez-un-robot-mobile-connecte-par-](https://openclassrooms.com/fr/courses/5224916-developpez-un-robot-mobile-connecte-par-bluetooth/5411806-decouvrez-la-constitution-des-capteurs) [bluetooth/5411806-decouvrez-la-constitution-des-capteurs](https://openclassrooms.com/fr/courses/5224916-developpez-un-robot-mobile-connecte-par-bluetooth/5411806-decouvrez-la-constitution-des-capteurs) (consulté 26/02/2022)

Lien Hypertexte : Microsonic, Capteurs à ultrasons, disponible sur : <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm>(consulté 26/02/2022)

Lien Hypertexte : KOLLMORGEN, Principes de fonctionnement d’un servomoteur, disponible sur : [https://www.kollmorgen.com/fr-fr/blogs/\_blog-in-motion/articles/principes-de-](https://www.kollmorgen.com/fr-fr/blogs/_blog-in-motion/articles/principes-de-fonctionnement-dun-servomoteur/) [fonctionnement-dun-servomoteur/](https://www.kollmorgen.com/fr-fr/blogs/_blog-in-motion/articles/principes-de-fonctionnement-dun-servomoteur/) (consulté 26/02/2022)