



SAE 1.5 : Traiter des données

BUT Réseaux & Télécommunications / ALT 1 / Binôme 31

COLIN Lucas

MICHON Antonin

Challenge #1 : Réaliser trois tours de circuit

Objectifs

1. Le véhicule doit pouvoir parcourir l'intégralité des trois tours du circuit sans faire de sortie ni perdre de temps.
2. La LED multicolore doit servir d'indicateur quant à l'avancement du véhicule sur son nombre de tours.

Acheminement & Organisation

Nous devons séparer les différentes composantes qui devront constituer notre futur programme qui permettra de remplir les objectifs ci-dessus.

Il nous faut donc en premier lieu pouvoir contrôler les mouvements du véhicule. Pour cela nous avons récupéré le contenu du TP4 de la ressource R107. Le fichier compressé [carMotors.zip](#) contient une classe déjà faite avec toutes les méthodes pour déplacer notre véhicule facilement.

Ensuite il fallait absolument pouvoir contrôler la LED multicolore pour permettre la résolution plus simple des problèmes qui allaient probablement se manifester sur la suite du programme. Nous avons donc récupéré notre code du TP2 avec la librairie "FastLED.h".

Pour permettre à notre véhicule de suivre la ligne pour rester sur le circuit, il nous faut pouvoir utiliser le capteur line track positionné sous notre robot, en utilisant la documentation fournie par le fabricant ELEGOO mise à notre disposition dans la SAE.

Objectif Principal

Une fois tous les différents éléments maîtrisés il faut coder un programme qui peut les faire marcher tous ensemble de la meilleure manière possible, c'est à dire avoir un programme court et efficace qui optimise au mieux les ressources mises à disposition par le matériel, et avoir un bon équilibre entre vitesse et précision afin de ne pas perdre trop de temps, et ne jamais sortir du circuit. Il faut aussi pouvoir compter les tours convenablement afin de respecter les objectifs initiaux.

Structure du Programme

I. Librairie & Imports :

Import des classes et librairies, qui seront nécessaires pour faire fonctionner correctement les capteurs ainsi que les moteurs et la Led. Création des objets que nous avons utilisés dans la suite du programme et utilisation de la fonction color pour pouvoir interpréter le code RGB.

Déclaration de la variable global lap qui comptera les tours.

II. Setup

Initialisation de la Led ainsi que des moteurs, et activation de la Led en rouge pendant 3 secondes.

III. Motor

Création de la fonction motor qui va permettre d'effectuer la bonne action en fonction des paramètres d'entrée gauche droite et milieu, correspondant au capteur de suivis de ligne.

IV. ledControl

Création de la fonction ledControl qui va permettre d'avoir une vérification du nombre de tour, et d'afficher la bonne couleur sur la Led multicolor.

V. Correction

Simple fonction qui récupère la dernière action avant que le robot perde la ligne, et corrige la trajectoire pour revenir sur la ligne si celle-ci est perdue. Elle est exécutée depuis la fonction motor, si les trois capteurs ne voient plus la ligne au sol.

VI. Loop

Appelle la fonction motor a chaque exécution avec les nouvelles valeurs des trois capteurs, et exécute la fonction ledControl afin de d'actualiser le nombre de tour pour la fonction

Problèmes et Observations

Au cours de la réalisation de ce projet, nous nous sommes heurtés à quelques problèmes, notamment beaucoup de sorties de circuit sur les virages. Pour pouvoir pallier cela, il a été nécessaire d'optimiser le programme en fonction des capteurs sur la ligne, pour tourner plus ou moins fort, en utilisant la méthode `.drive()` de la classe `carMotors`.

Cette technique, qui a remplacé les précédents `turnLeft()` & `turnRight()`, a aussi permis d'uniformiser la direction en réduisant considérablement les accoups lors des virages fait par le véhicule.

Nous avons aussi dû régler précisément la vitesse en fonction du circuit, si c'est une ligne droite par exemple le véhicule accélère, afin de gagner du temps, sans sortir du circuit.

[Lien de la vidéo de la Version 1](#) (3 tours en 1 minutes)

En raison d'un temps assez long, entre 1 minute et 30 secondes et 1 minute, il fallait absolument améliorer le programme afin d'améliorer la vitesse du robot.

Mais comme dit précédemment, si le robot va trop vite, il sort du circuit. J'ai donc codé une nouvelle fonction qui, en fonction de la dernière action du robot, corrige la trajectoire et fait l'inverse, donc si le robot tournait à gauche lorsqu'il a perdu la ligne, alors la fonction correction va le faire revenir rapidement à droite afin d'essayer de la récupérer.

Cette nouvelle fonctionnalité de ce programme nous a permis de presque atteindre les 40 secondes pour les 3 tours.

[Lien de la vidéo de la Version 2](#) (3 tours en 40 secondes)

Il fallait aussi régler la fréquence de vérification du tour, afin que lors d'un passage il ne compte pas 2 tours si le robot était trop lent, pour éviter les erreurs lors du comptage du tour. Et ensuite régler la Led avec un petit ajout qui consiste à clignoter rapidement en blanc quand le tour est détecté, ce qui est pratique lorsque par exemple le véhicule passe à la moitié du circuit, cela nous permet de voir que la ligne a été détecté mais que nous ne sommes pas sur le prochain tour.