

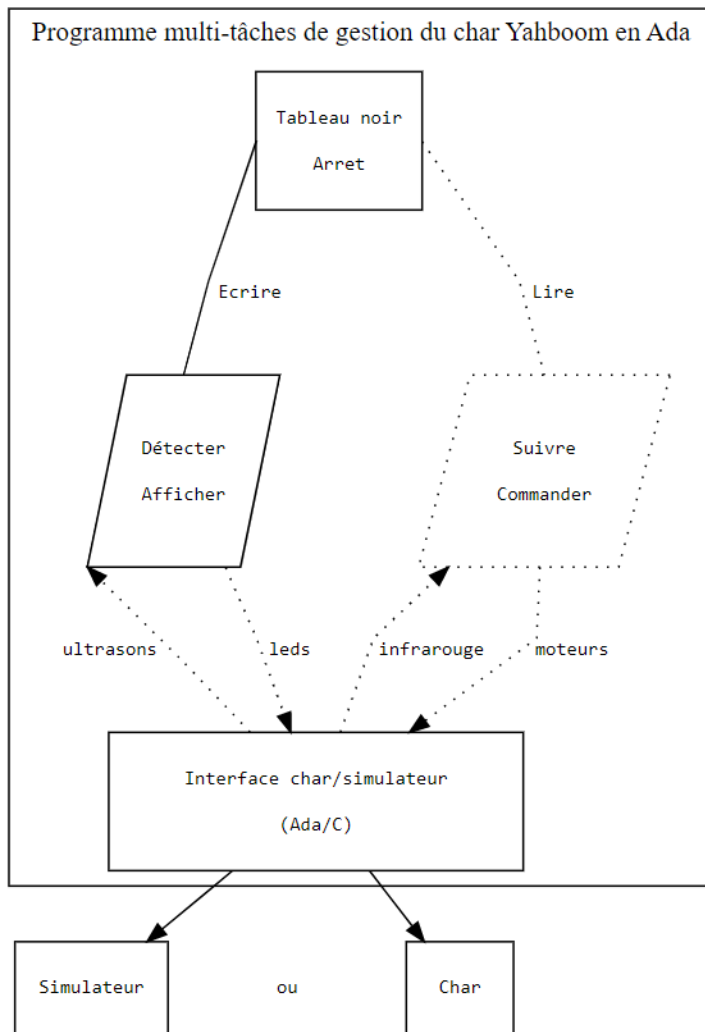
Systèmes embarqués - TP03

DE MEIRA LIMA Otávio
CHAHIDI Hamza

Dans cette simulation, une voiture roule sur un terrain plat. Il parvient à détecter votre itinéraire grâce à 4 sondes infrarouges, votre position par rapport à une route créée numériquement. L'objectif final de ce travail est de faire en sorte que la voiture suive le chemin, en détectant les obstacles et en mettant en œuvre une logique pour les éviter.

La simulation de voiture et de route est réalisée en Tcl, tandis que la lecture de capteurs, la détection d'obstacles et le simulateur de puissance moteur ont été réalisés en Ada. Les programmes communiquent en envoyant des messages sur le réseau. Des bibliothèques ont été fournies pour aider à manipuler la voiture robot, telles que la simulation de piste, la simulation de détecteurs à ultrasons, de moteurs et de LED. Le programme doit être exécuté en multitâche, car la détection d'obstacles et la réponse à la détection doivent être effectuées simultanément dans le programme.

Les optimisations suivantes ont été faites pour capturer la distance de la voiture par rapport aux obstacles présents sur le chemin.



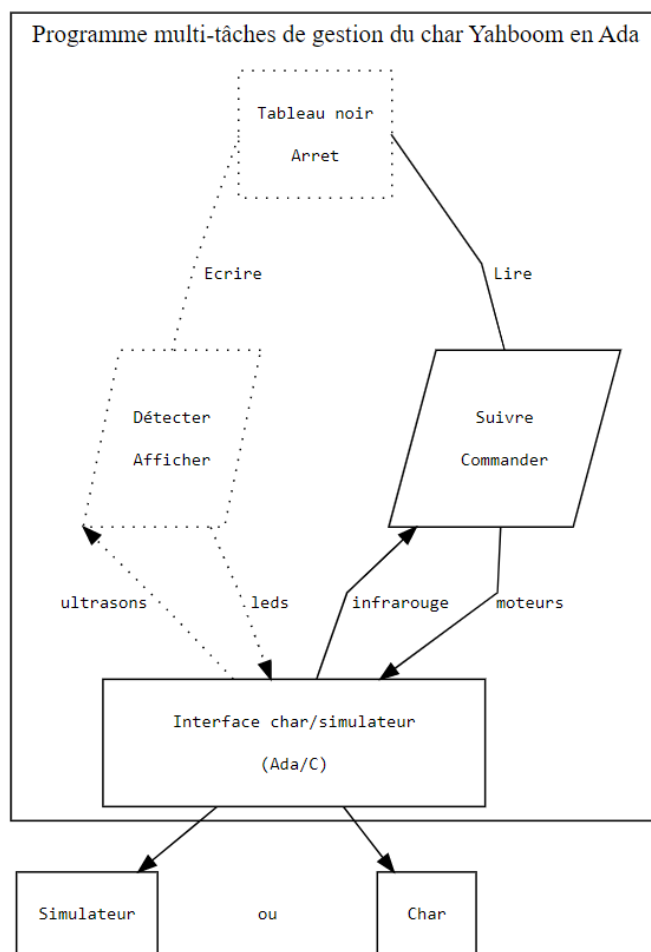
Pour optimiser la réponse du véhicule avec le multitâche, il a été nécessaire de mettre en place un nouveau fichier de contrôle en utilisant le fichier précédent comme base. La première optimisation faite a été la création d'une LED qui brille à l'arrière de la voiture. Une mise à jour a été apportée à la méthode **DetecterAfficher** pour mettre à jour la couleur de la LED en fonction de la distance à laquelle elle se trouve par rapport à un obstacle. Les couleurs sont le vert, le jaune, le violet et le rouge, augmentant respectivement le niveau de danger directement proportionnel à la proximité de l'obstacle. Il existe une méthode simple qui capture la distance de l'obstacle appelée `Get_Ultrasonic_Distance`. Un entier est renvoyé et peut être facilement comparé aux valeurs de distance auxquelles chaque LED doit être active. Chaque itération de la boucle est effectuée après un intervalle de 20 millisecondes à partir de l'itération précédente, comme défini dans la spécification de conception.

Cependant, la seule réponse du robot face à un danger est de changer la couleur de sa LED. Ce n'est pas génial, car le véhicule continue de subir le choc s'il n'y a pas de réponse de ses moteurs. Pour ajouter une réponse aux moteurs, une modification a été apportée à la même méthode pour signaler les types de danger imminent. Avant ce changement, un nouveau type numérique appelé « Danger » devait être créé pour identifier trois réponses à un obstacle : aucune, pause et arrêt.

- Aucun : aucun obstacle à moins de 10 cm
- Pause : un obstacle entre 5 et 10 cm est détecté
- Stop : un obstacle est détecté à moins de 5 cm

Chaque type de danger donnera une réponse différente au moteur. À ce moment-là, le type de danger ne sera envoyé qu'à un tableau noir, qui sera consommé plus tard dans la méthode qui gère le moteur.

La méthode suivante va manipuler le moteur et gérer les réponses du véhicule lorsqu'il reçoit un type de message de danger.



La méthode implémentée **SuivreCommander** avec la logique suivante fonctionnera en parallèle avec la méthode décrite précédemment. Cette fois, la méthode lira à partir de Tableau Noir et agira sur le message renvoyé.

Cette méthode s'exécute dans une boucle infinie avec un délai de 20 millisecondes entre les itérations, tout comme l'autre tâche. Si le type de danger est "stop", la voiture doit

immédiatement s'arrêter et sortir de la boucle infinie d'itérations. Si le danger est "pause", la voiture doit s'arrêter jusqu'à ce que l'obstacle soit retiré de la piste. Si le danger est « nul », la voiture doit reprendre la piste et poursuivre sa route avec une puissance de 10 unités. La voiture a 4 capteurs, chacun renvoie une valeur booléenne. La moyenne des positions des capteurs (qui sont signalées par des nombres de 0 à 3) qui renvoient une valeur vraie correspond à l'emplacement du centre de la piste. Sachant où se trouve le centre de la piste, il est possible de faire de petites corrections dans la puissance des moteurs gauche et droit pour corriger le centre de la voiture avec la formule suivante :

```
Pui := T_Power ((centre - 1.5) * 6.0 * 0.6);
```

Si tous les capteurs retournent faux, cela signifie que la voiture a beaucoup quitté la piste. Pour cette correction, il suffit d'effectuer une petite rotation jusqu'à ce qu'un des capteurs identifie la piste. De cette façon, vous pouvez recalculer le centre de la voie.