

HEIG-VD — PCO

Laboratoire 4 – Rapport

Loïc HERMAN

16 novembre 2023

1 Description des fonctionnalités du logiciel

L'objectif de ce laboratoire est de simuler une maquette de train comportant une section de voies partagées entre deux locomotives et une gare.

Le système est donc capable d'identifier les points d'arrêts des locomotives, d'encadrer une priorisation de l'accès à la section partagée tout en restant en gestion concurrente.

2 Choix d'implémentation

Le parcours choisi est décrit dans la figure 1. Nous avons donc une station placée après les balises 19 et 16, les trains arrivant à la station dans des directions opposées.

Les contacts qui font office de signaux avancés pour la section partagée sont entourés en violet.

2.1 Calcul de la route

Une classe `RailwayRoute` a été définie pour simplifier le stockage des informations pertinentes à la route (séquence de contacts, section partagée, aiguillages, station, points d'arrêt). Le constructeur s'occupe de calculer les points d'arrêt.

On notera que les chemins doivent être définis dans leur entièreté, afin de pouvoir calculer les différents points d'arrêt.

2.2 Gare

La première zone atteinte par les locomotives est la gare, il y a donc un point d'arrêt qui est placé à l'entrée de la gare, et les trains s'arrêteront progressivement une fois le point franchi. La classe `RailwayRoute` a la méthode `awaitStation()` qui permet de bloquer le thread jusqu'à ce que le train franchisse le contact.

D'un point de vue synchronisation, le premier train qui arrive en gare sera arrêtée et bloquée au moyen d'un sémaphore par la classe de synchronisation, un booléen stocke l'état du sémaphore d'attente. Cette locomotive reçoit une priorité de -10, qui sera utile plus tard.

Quand une seconde locomotive entre en gare, le booléen annonce que l'autre train est déjà en gare et cela permet, une fois la locomotive arrêtée, de forcer une attente de 5 secondes avant de redémarrer la locomotive et l'autre train au moyen du sémaphore d'attente. Le second train reçoit une priorité de 10, qui lui permettra de passer en premier dans la section partagée.

2.3 Section partagée

Quand le train quitte la gare, le système place automatiquement le point d'attente sur le contact « Access » correspondant au parcours. Quand le train survole ce contact, la classe de synchronisation va vérifier si la section partagée est occupée ou si le train a une priorité inférieure à 0, dans quel cas le train sera arrêté.

Le thread du train en question est bloqué au moyen d'un autre sémaphore et il est comptabilisé dans un compteur. La section de blocage du point d'accès est vérifiée au moyen d'une boucle `while` pour éviter que le train redémarre dans le cas où l'obtention du mutex après la libération du sémaphore d'attente se ferait préempter et passer devant pas un autre thread.

Les aiguillages d'entrée et de sortie de la section protégée sont automatiquement déviés en fonction du parcours par la classe `RailwayRoute`, sur demande une fois que le train est libéré du signal d'accès.

Quand un train passe le contact de sortie, attendu automatiquement quand le train entre la section partagée, le booléen d'occupation de la section protégée est mis à jour et le sémaphore d'attente est libéré (si un train était bien en attente).

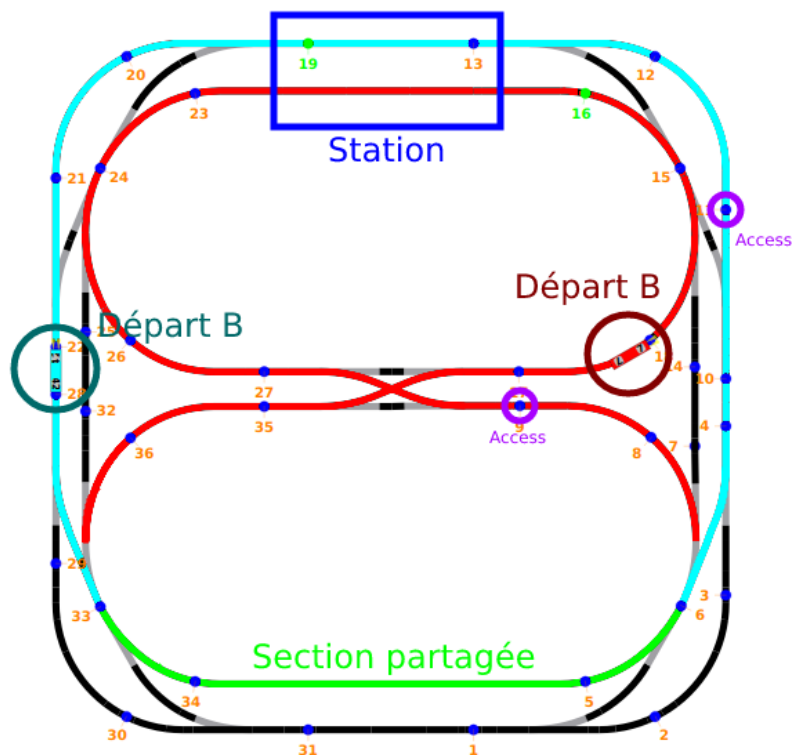


FIGURE 1 – Parcours définis pour les trains

3 Tests effectués

Nous avons vérifié la bonne exécution du programme en monitorant les logs des locomotives et les logs généraux. Ainsi, nous avons pu vérifier que les trains attendaient bien 5 secondes en gare, et que la priorité d'accès à la section critique est accordée au train qui est arrivé en dernier à la gare.

Nous avons également laissé les locomotives tourner durant une dizaine de tours, sans avoir constaté d'anomalies.

Les tests précédents ont été réitérés avec plusieurs configuration de vitesses pour les locomotives.

Le bon fonctionnement du bouton d'arrêt d'urgence a également été vérifié.