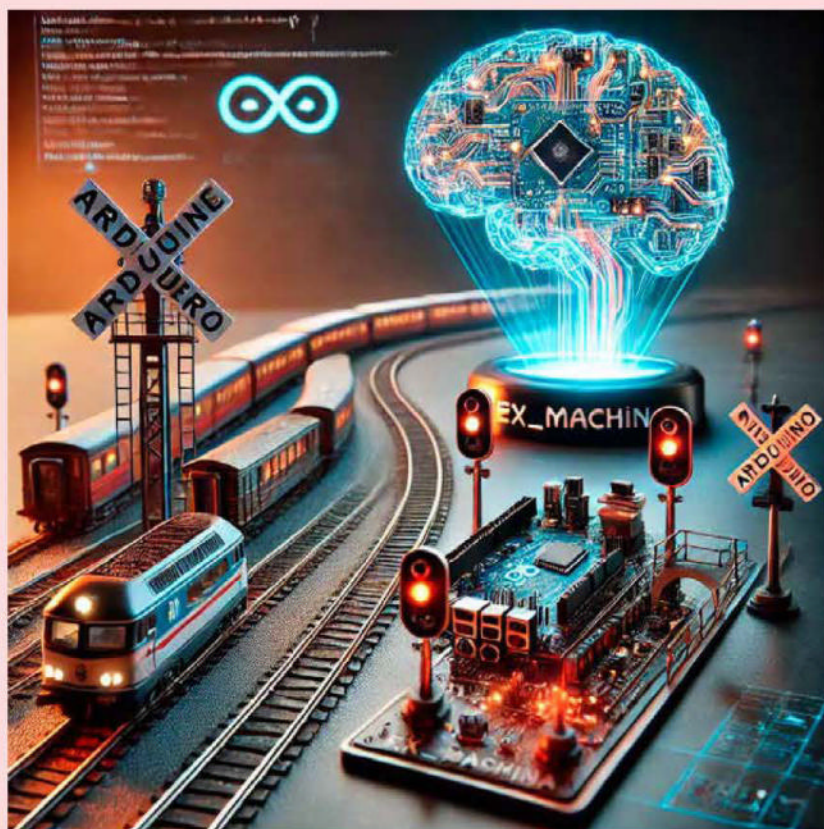




*Avec l'aide
de l'intelligence
artificielle*

PREMIÈRE PARTIE : PRÉSENTATION DU PROJET

EX_MACHINA

Un réseau automatisé

Croisements en gare, cantons, signalisation fonctionnelle, en automatique et en toute sécurité. Ça vous semble très compliqué ? Heureusement, l'Intelligence Artificielle va grandement vous faciliter la tâche. Suivons un exemple concret en construisant un réseau « conçu et géré » par une IA.

Texte et illustrations: Christian Bézanger

Le réseau EX_MACHINA est ainsi baptisé en hommage au film de science-fiction éponyme datant de 2014 et traitant d'Intelligence Artificielle. C'est un petit réseau, à la fois simple à réaliser et bon marché, permettant de faire circuler deux trains en toute sécurité grâce à une carte Arduino dont les programmes sont écrits (à plus de 90 %) par une Intelligence Artificielle (IA). C'est cette IA qui propose également un algorithme performant et permet cette circulation, ce qui permet de dire qu'EX_MACHINA est le premier réseau de trains miniatures « conçu et géré » par une Intelligence Artificielle. En le construisant, vous découvrirez des techniques intéressantes et vous apprendrez à

maîtriser un outil puissant qui sera de plus en plus utilisé dans l'avenir. Laissez-moi vous en dire plus...

Genèse du projet

Au cours de l'année 2024, Loco-Revue a publié une série de cinq articles qui présentaient l'écosystème Arduino (cartes à microcontrôleur et environnement de développement). Ces articles étaient surtout théoriques et il était important de prévoir une suite afin de mettre en pratique ce qui avait été dit : une nouvelle série pourrait montrer comment gérer un petit réseau à voie unique avec une carte Arduino. Faire tourner un train est assez facile, mais faire tourner deux trains de façon complètement automatique, demande de

PRINCIPALES FOURNITURES

POUR CET ARTICLE

- L'ENSEMBLE DES COUPONS DE VOIE (VOIR TABLEAU 1) ET DES ÉCLISSES ISOLANTES
- UNE CARTE 8 RELAIS SIMPLES ALIMENTÉS EN 5 V POUR ÊTRE COMPATIBLE ARDUINO
- UN SOCLE EN BOIS POUR RECEVOIR LE LOCODROME (UNE PLAQUE DE CP DE 10 MM ET DES TASSEaux POUR RIGIDIFIER)

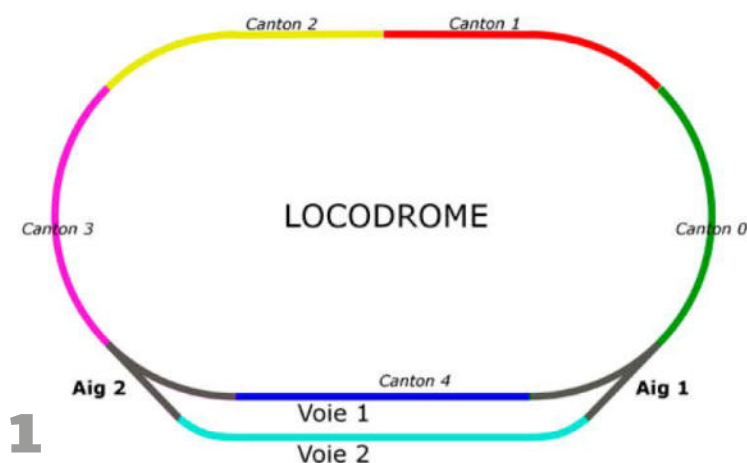
POUR LE PROCHAIN ARTICLE

- UNE CARTE ARDUINO MEGA2560
- UNE CARTE 1 RELAIS DPDT EN 5 V (POUR COMMANDER L'INVERSION DE COURANT SUR LE FEEDER)

- UNE ALIMENTATION DC 5 V 1,5 A (POUR LES CARTES RELAIS NOTAMMENT)
- DEUX KITS DECAPOD RÉF 6171 POUR MOTORISER LES AIGUILLES (OU DEUX MOTEURS LENTS MP) (EN VENTE CHEZ LR MODÉLISME)
- UN MODULE RÉF 6170 DE CHEZ DECAPOD POUR TROUVER LE POINT MILIEU D'UN SERVOMOTEUR
- CINQ BOUTON POUSSOIRS (1 ROUGE, DEUX JAUNES ET DEUX BLEUS PAR EXEMPLE) ET DE QUOI CÂBLER L'ENSEMBLE.

Figure 1 : Plan schématique du réseau.

Figure 2 : Découpe en cantons du réseau Train'In Box.



rajouter un peu d'intelligence à notre carte Arduino pour que celle-ci puisse suivre le scénario choisi par le joueur : ordre des trains et sens de circulation. C'est alors que je me suis dit qu'une Intelligence Artificielle pourrait participer à la création de ce réseau et que ce serait une première (du moins, à ma connaissance). De plus, cela compléterait l'article que j'ai écrit dans le *Loco-Revue* 928 de novembre 2024 sur l'apport de l'Intelligence Artificielle pour programmer Arduino.

Le circuit, encore appelé locodrome

Le réseau EX_MACHINA n'a rien d'extraordinaire ; c'est même plutôt un réseau de débutant que chacun a sans doute possédé un jour. C'est un circuit bouclé analogique contenant deux aiguilles symétriques per-

mettant d'avoir deux voies en gare (**figure 1**). Ce circuit, qui a été imposé à l'IA, est divisé en quatre cantons (numérotés 0 à 3) et deux voies en gare. Deux trains sont en attente sur les voies, prêts à partir en fonction des indications du joueur. L'IA s'occupe du reste et vous pouvez voir chacun des trains faire un tour de circuit, soit en se suivant, soit chacun son tour (en analogique, il est impossible d'avoir deux trains en sens contraires sur une seule voie !). Ce petit réseau est à la fois très facile à construire et très bon marché, et ses possibilités sont suffisantes pour agrémenter de belles séances de jeu. Nous allons décrire comment le construire, mais nous n'évoquerons pas le décor de ce circuit et nous ne nous intéresserons qu'à la commande des rails (donc des trains) : c'est pourquoi nous appelons ce circuit « locodrome ». Rien ne

vous empêche de prévoir un thème et un décor associé (gare de passage, site industriel, train minier, tramways urbains).

La numérotation des cantons commence par 0 et suit le sens antihoraire. Comme les cantons sont courts, la circulation est plus réaliste avec des trains de type autorail, tramway, engin haut le pied, engin moteur avec une seule remorque. Le canton numéro 4 est constitué d'une seule des deux voies en gare (appelée voie active et pouvant être voie 1 ou voie 2 en fonction de la position des aiguilles), ce qui fait donc un circuit à cinq cantons (de 0 à 4). Ce locodrome est construit à partir de rails Peco Setrack à l'échelle H0 (vendus par LR-Modélisme) mais vous pouvez le réaliser à l'échelle et à l'écartement que vous voulez. Il est même possible de partir d'un coffret Train'In Box, car la forme du circuit (ovale, ►►

MATÉRIEL DE VOIE NÉCESSAIRE (N ET H0)

H0/00 Setrack (dimensions du réseau : 1546 x 876 mm) :	N Setrack (dimensions du réseau : 804 x 456 mm)
ST-201 x 4 (335 mm)	ST-11 x 4 (174 mm)
ST-225 x 2 (22,5°)	ST-3 x 2 (22,5°)
ST-226 x 6 (45°)	ST-12 x 6 (45°)
ST-240 (r = 438 mm)	ST-5 (R = 228 mm)
ST-241 (r = 438 mm)	ST-6 (r = 228 mm)
SL-100 (914 mm)	SL-300 (914 mm)
Éclisses isolantes SL-111	Éclisses isolantes SL-311

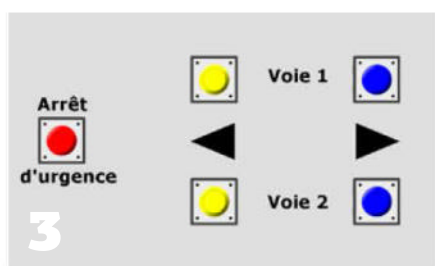
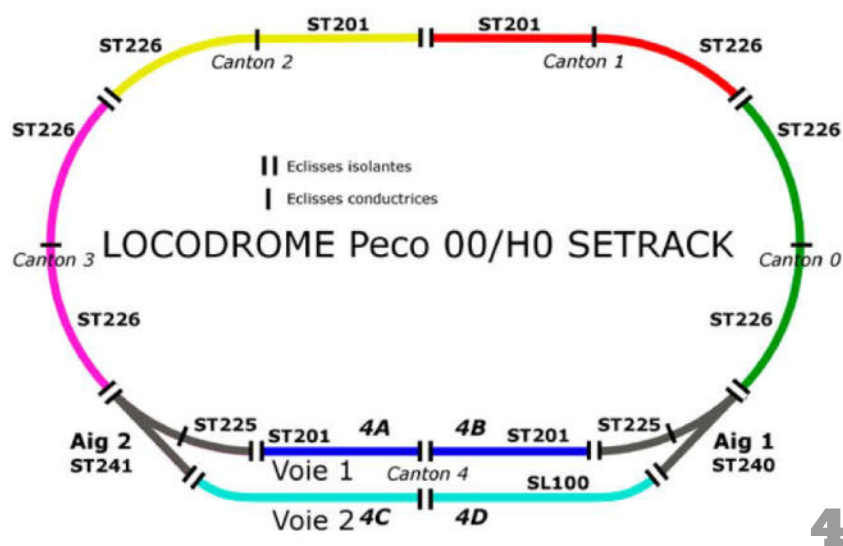


Figure 3 : Tableau de commande avec bouton d'arrêt d'urgence.

Figure 4 : montre où placer les éclisses isolantes afin de découper le circuit en différents cantons.



►► rond ou os de chien) importe peu. La **figure 2** montre ce locodrome réalisé avec le premier coffret Train'in Box : même topologie, même nombre de cantons, aiguilles courbes au lieu de droites mais cela n'a guère d'importance. Bien évidemment, une signalisation lumineuse fonctionnelle accompagne le locodrome avec des signaux de type carré pour la gare (protection des aiguilles) et des signaux de type B.A.L. (Block Automatique Lumineux) pour la voie unique. Les aiguilles sont motorisées et prises en charge par Arduino. Le locodrome est aussi équipé d'un tableau de commande, qui permet au joueur de choisir son train (voie 1 ou voie 2) et le sens du parcours (horaire ou antihoraire). La **figure 3** montre ce tableau de commande et vous pouvez constater qu'il y a un bouton d'arrêt d'urgence qui peut être très pratique pour éviter un ... boum, catastrophe ! Cependant, deux trains peuvent se suivre puisqu'ils sont espacés par le B.A.L.

Ce qu'en pense l'IA ChatGPT

Ce projet, bien qu'il soit modeste, va nous confronter à de nombreux problèmes qu'il faudra résoudre : localisation des trains, alimentation ou non des cantons, modification

de la signalisation lumineuse en prenant en compte la position des trains et des aiguilles. Bien qu'ayant déjà quelques solutions en tête, je décide de poser la question à l'IA ChatGPT après lui avoir décrit le circuit. Voici la question que je lui pose :

Avec ce circuit, quelles seraient les grandes étapes à faire dans la fonction loop pour que l'automatisme fasse faire aux deux trains un tour de circuit avec retour en gare, soit simultanément, soit l'un après l'autre en fonction des sens de circulation. Juste répondre sans écrire de code. Et voici la réponse qu'elle me donne : Pour permettre aux deux trains de parcourir le circuit et de revenir en gare selon les conditions spécifiées, voici les grandes étapes à inclure dans la fonction loop :

1. Initialisation et Suivi de l'État des Trains : maintenir les états des deux trains (position actuelle, prochain canton prévu), identifier quel train est sur quel canton, et gérer séparément leurs déplacements.

2. Détection et Gestion des Cantons Occupés : vérifier l'occupation des cantons par chaque train, couper l'alimentation des cantons précédents pour éviter que les trains ne se rapprochent excessivement.

3. Mouvement des Trains : simuler le déplacement des deux trains. Si simultanément, déplacer chaque train sur son prochain canton indépendamment, si alterné, déplacer un train à la fois (l'autre reste en attente).

4. Gestion de la Voie Active en Gare : à l'approche de la gare (canton 4), gérer quelle voie sera utilisée par chaque train en fonction de la variable « sens de circulation ». Faire entrer les trains dans la voie désignée (1 ou 2).

5. Synchronisation des Trains : gérer le comportement des trains lorsque l'un d'eux arrive en gare et l'autre continue son parcours. Prévoir une logique pour éviter que les trains n'entrent en conflit dans les sections communes.

6. Arrêt et Redémarrage : après un tour complet, détecter si les trains sont de retour en gare.

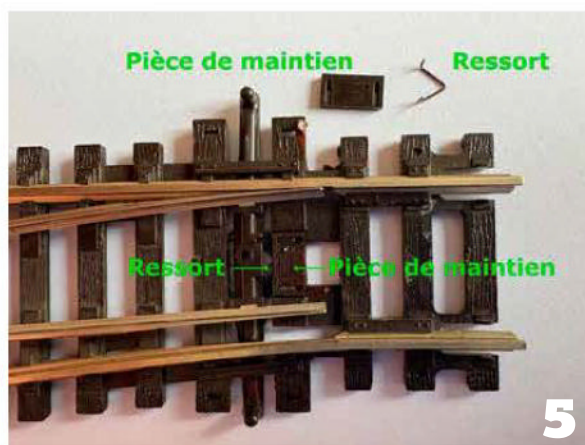
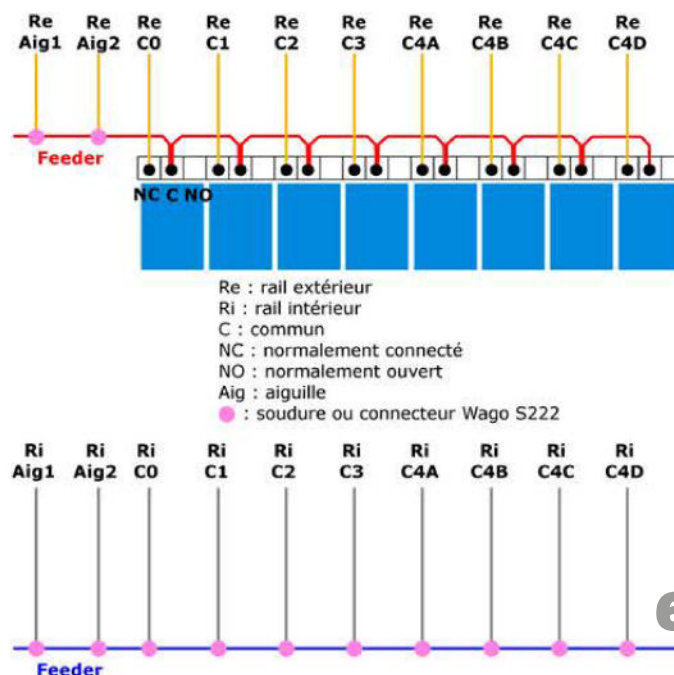


Figure 5 : Suppression du ressort de verrouillage de l'aiguille Peco.

Figure 6 : Câblage de la carte huit relais (représentés en bleu

clair) : les feeders doivent avoir une section plus forte que les fils qui alimentent les voies.

Figure 7 : Carte relais disposée sur le plateau.



Réinitialiser l'état des trains pour démarrer un nouveau tour si nécessaire.

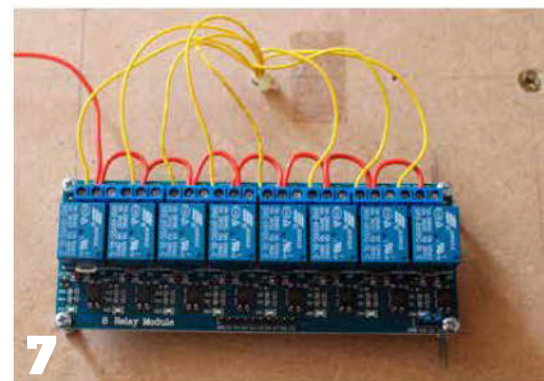
7. Affichage et Débogage : afficher l'état des cantons, la position des trains, et les voies actives pour suivre le déroulement de l'automatisme en temps réel.

L'IA a très bien identifié les différents problèmes à résoudre. Un point n'est pas indispensable : c'est la simulation du déplacement des trains (point N° 3) car ceux-ci existeront en vrai et seront posés sur le locodrome. Je vous propose donc de faire la même chose que moi : construire ce locodrome, l'équiper d'une carte Arduino et demander à votre IA préférée (ChatGPT, Le Chat, DeepSeek, Perplexity ou autre) de vous aider à mettre au point les différentes parties de programme. L'IA doit être considérée comme un outil extrêmement puissant, capable d'écrire pour vous les parties de programme dont vous avez besoin et cette nouvelle série d'articles vous montrera comment travailler avec une IA. C'est tout de même vous qui utilisez l'outil et qui restez le décideur dans ce projet ; cette série d'articles reste technique et exclut de fait toutes considérations philosophiques sur les bienfaits ou méfaits pour notre société des Intelligences Artificielles. Si vous êtes débutant en Arduino, je vous conseille de pratiquer un peu avant d'attaquer ce projet ; la lecture de mon guide « Animez votre réseau : initiation au système Arduino » (en vente chez LR-Modélisme) vous donnera une bonne base en vous proposant de nombreux petits montages dont les programmes sont analysés. Nous allons évoquer les différentes solutions matérielles qui s'offrent à nous pour

ce projet et je vous fournirai un programme fonctionnel dans le dernier article. Le décor est planté et je ne rajouterai pas « il n'y a plus qu'à » car avant d'obtenir tout cela, un peu de cogitation s'impose !

Conseils pour travailler avec une IA

La première chose à faire est de vous créer un compte (généralement gratuit) chez votre IA préférée (voir LR928 déjà cité) ; pour ma part, j'ai choisi ChatGPT. Pour travailler avec une IA, il est important de bien expliquer ce qu'on veut obtenir et de donner le maximum de détails. C'est ce qu'on appelle « prompter », c'est-à-dire rédiger votre demande dans la case appropriée. L'IA répond à votre requête, et comme on l'a vu dans l'article de LR928, le résultat peut fonctionner du premier coup, ou pas ! Votre rôle consistera alors à décrire ce qui ne fonctionne pas et si vous donnez suffisamment de détails, l'IA pourra proposer une correction de son programme. La communication avec l'IA est la clé de votre réussite : si vous allez voir votre pharmacien pour lui dire que vous êtes malade, il ne sera pas à même de vous aider, mais si vous décrivez vos symptômes, alors il aura peut-être une solution. Ne vous découragez surtout pas : à force de décrire les mauvais fonctionnements, l'IA finira par les résoudre. Si elle échoue, c'est dû à une mauvaise communication entre vous et elle. Si le programme fonctionne comme vous le souhaitez, il suffira alors de reprendre la partie qui vous intéresse pour l'inclure à votre projet. Le mieux est donc de demander à l'IA d'écrire des fonctions que



vous pourrez inclure au programme principal (personnellement je lui imposais les noms de certaines variables afin de faire fonctionner ensemble les différentes fonctions produites). Si vous demandez à l'IA d'écrire en une seule fois la totalité du code pour le locodrome que nous avons décrit, et ce dans le but de faire fonctionner les deux trains ensemble, il y a de très fortes chances que cela ne fonctionne pas du premier coup (croyez-moi, j'ai essayé). Comme le programme sera complexe, il vous sera très difficile d'identifier la nature du défaut. Il faut donc travailler par étapes. Et surtout bien savoir ce que chaque étape doit réaliser, sans être trop ambitieux d'un coup.

Construction du locodrome

Comme je l'ai dit, vous pouvez choisir l'échelle et l'écartement que vous voulez pour ce locodrome ; ce qui compte, c'est qu'il comporte quatre courts cantons (longs si vous avez de la place), deux aiguilles et deux voies en gare. J'ai utilisé de la voie Peco mais d'autres marques peuvent convenir tant qu'on reste dans le système deux rails. ►►

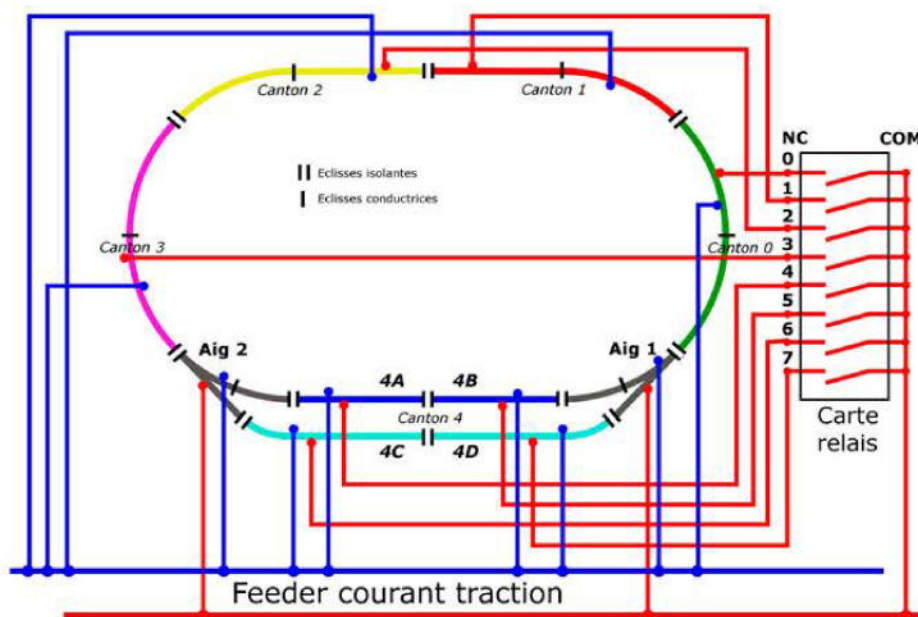
8

Figure 8 : Câblage complet du locodrome.

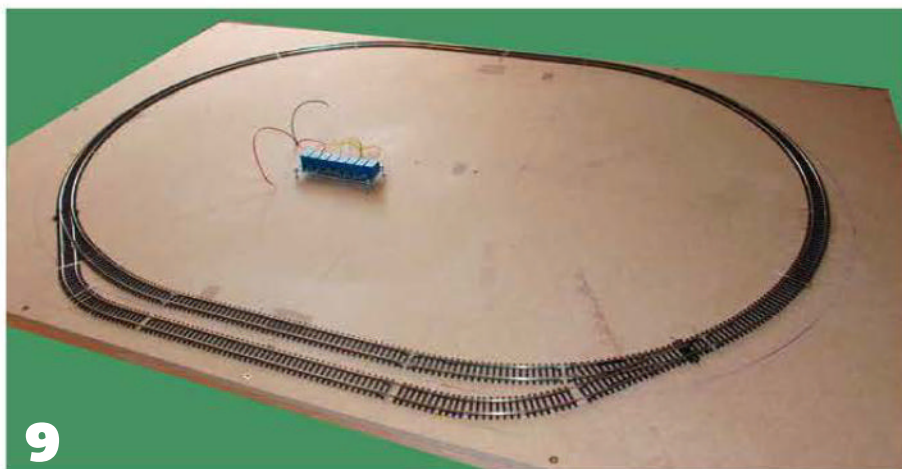
Figure 9 : Le locodrome est câblé: les différents cantons et la carte relais.

» La **figure 4** montre où placer les éclisses isolantes afin de découper le circuit en différents cantons. On pourrait se contenter de placer ces éclisses sur un des deux rails mais je vous conseille de le faire sur les deux rails. D'abord, parce que cela laisse la possibilité d'évoluer vers des systèmes du commerce qui parfois imposent une coupure sur les deux rails. Ensuite, parce qu'il est plus facile de relier deux fils ensemble plutôt que tronçonner un rail et réalimenter la coupure. Comme vous pouvez le constater, les voies en gare ont été divisées en [4A, 4B] et [4C, 4D]. Comme notre carte nous propose huit relais, il aurait été dommage de n'en utiliser que six alors que cette division en deux sous-cantons permettra un arrêt plus précis du train en fonction du sens de marche (voir plus loin). Si vous utilisez un coupon de voie flexible pour la voie 2, il est nécessaire de tronçonner les rails pour créer cette division en deux sous-cantons 4C et 4D et de souder tous les fils nécessaires.

Chaque canton sera muni de deux fils soudés sur les rails ; prévoir deux couleurs de fil pour ne pas confondre le rail intérieur et le rail extérieur. Ces fils seront reliés à un feeder de courant de traction, soit directement, soit par l'intermédiaire d'une carte relais 5V pour rester compatible avec Arduino (on les trouve facilement sur internet). Le principe est simple : les fils du rail intérieur du locodrome sont reliés directement au feeder bleu, les fils du rail extérieur du locodrome sont reliés aux sorties NC (normalement connecté) d'une carte à huit relais, le feeder rouge alimentant les entrées COM (commun). Les zones d'aiguilles sont alimentées en permanence afin de faciliter l'entrée ou la sortie de la gare ; soudez bien les fils sur les rails de l'aiguille (côté pointe) et non sur l'élément courbe ST225. Pensez également à retirer le ressort de maintien de position accessible sous l'aiguille ou par le dessus en retirant la pièce de maintien avec un scalpel (**figure 5**) ; ceci a pour but de ménager la motorisation de l'aiguille en diminuant l'effort à faire pour le mouvement. La **figure 6** montre comment



Alimentation des cantons via carte relais



câbler la carte huit relais (représentés en bleu clair) : les feeders doivent avoir une section plus forte que les fils qui alimentent les voies. Pour relier les fils des voies avec les feeders, vous pouvez soit faire une soudure, soit utiliser des connecteurs Wago : dans ce cas, choisissez la référence S222 permettant de relier tous diamètres de câbles (de 0,08 à 4 mm²). La **figure 7** montre la carte relais sur le plateau ; les fils jaunes alimentent les rails extérieurs des cantons et on distingue également les feeders rouge et bleu. Tous les fils sont ensuite disposés sous le plateau et fixés pour qu'ils ne soient pas arrachés (scotch ou pistolet à colle). La **figure 8** montre le câblage complet du locodrome qui sera réalisé sous le plateau, une fois les voies fixées sur celui-ci. Il faut prévoir un plateau plus grand que le circuit afin de placer toutes les cartes électroniques nécessaires sur un

côté. Si vous ne prévoyez pas de décor, vous pouvez poser les cartes électroniques au centre du circuit. Des tasseaux rigidifient l'ensemble et la hauteur du plateau doit permettre la présence des moteurs d'aiguilles fixés dessous. Avant de fixer définitivement la voie, pensez à repérer les emplacements des traverses mobiles des aiguilles afin de percer une ouverture qui permettra le passage de la tige du moteur. Les éclisses isolantes sont en matériau souple qui ne permet pas une liaison solide des rails ; il est donc conseillé de bien fixer les voies de part et d'autre des éclisses isolantes, soit avec des vis (Fleischmann 6410 par exemple), soit avec une colle forte (époxy par exemple) mais dans ce cas, le démontage devient périlleux pour le matériel. Voilà, vous avez monté votre locodrome et avez câblé les différents cantons avec votre carte relais (**figure 9**). Vous pouvez déjà faire circuler un

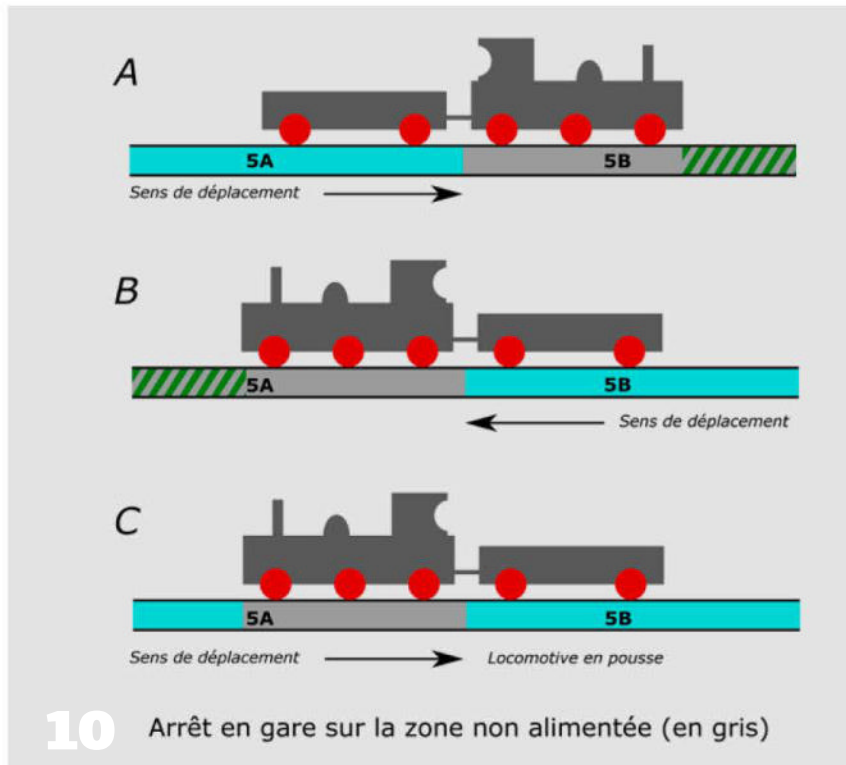
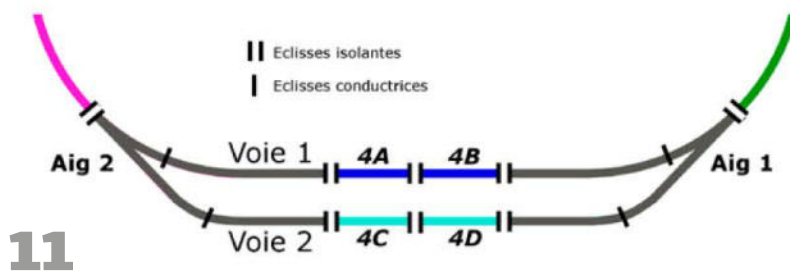


Figure 9: Le sous-canton non alimenté (en gris) dépend du sens de déplacement de la locomotive.

Figure 10: Selon votre choix, il peut être nécessaire d'adapter la position des éclisses isolantes pour les voies en gare.



train pour vérifier que tout fonctionne. Il suffit de relier votre alimentation au feeder et de jouer avec le bouton du rhéostat pour choisir sens et vitesse ; une intervention manuelle sur les aiguilles permet de vérifier que les deux voies en gare sont opérationnelles.

Principe d'alimentation des cantons

Lorsque le relais n'est pas excité, l'entrée COM est reliée à la sortie NC et le courant passe pour alimenter le canton. Un signal approprié émis par une carte Arduino peut exciter le relais et dans ce cas, l'entrée COM est reliée à la sortie NO (normalement ouvert) ; il y a donc coupure d'alimentation du canton puisque ce qui vient du feeder n'est plus relié au fil qui va vers le rail extérieur. Ce signal approprié est généralement un LOW (0 V) sur une sortie digitale ; si la carte relais nécessite un HIGH (5 V) pour exciter un relais, il suffit de modifier une variable en début de programme (ceci peut être expliqué à l'IA). Sur l'ensemble du circuit, le principe du B.A.L. consiste, dès qu'un canton est occupé, à couper l'alimentation du canton précédent. Comme notre voie unique

peut être parcourue dans les deux sens, le numéro du canton précédent dépend du sens de circulation. De même, lorsqu'il y a une aiguille, le numéro du canton précédent dépend aussi de la position de l'aiguille. On voit que finalement, la gestion du locodrome pour gérer l'espacement des trains ou les arrêts en gare n'est pas aussi simple qu'on pourrait le penser, mais nous y reviendrons ultérieurement.

Nous savons donc maintenant comment couper l'alimentation d'un canton, donc comment stopper un train. Celui-ci va s'arrêter tout de suite, et suivant le sens de déplacement, il peut s'arrêter à un bout ou à l'autre du canton. La signalisation sera placée en conséquence afin de faire croire que le train s'arrête juste devant et quelques essais seront à faire en fonction de la vitesse des trains et de leur inertie.

Principe d'alimentation des voies en gare

Les voies en gare étant divisées chacune en deux zones (sous-cantons 4A à 4D), l'arrêt de la locomotive se fera sur l'une ou sur l'autre en

fonction du sens de circulation. La **figure 10** montre que le sous-canton non alimenté (en gris) dépend du sens de déplacement de la locomotive pour que celle-ci s'arrête juste dessus et que le train soit centré. C'est le programme qui gère le choix du sous-canton à ne pas alimenter si on veut l'arrêt du train en gare. Ce système ne fonctionne que si la locomotive tracte (cas A et B) ; si elle pousse, la zone d'arrêt doit être plus petite (cas C) ou bien il faut prévoir une temporisation avant de couper le courant sur la zone d'arrêt afin de parcourir la zone hachurée de la figure 10 (cas A et B). Le problème de l'arrêt en gare pour avoir un train centré par rapport au quai dépend donc de plusieurs paramètres : composition du train (automotrice ou locomotive et remorque), longueur du convoi, traction ou pousse, orientation au début du jeu, vitesse de déplacement. Selon votre choix, il peut être nécessaire d'adapter la position des éclisses isolantes pour les voies en gare (**figure 11**). Si le fonctionnement ne vous convient pas, il sera toujours possible d'adapter le programme pour l'améliorer ou bien vous orienter vers des solutions matérielles ajoutées comme des capteurs de position (I.L.S par exemple).

Conclusion

Voilà donc un petit projet qui nous permet d'explorer les techniques à mettre en œuvre pour gérer un réseau avec Arduino et confier le travail de programmation à une IA. J'espère que l'aventure vous tente. Construisez le locodrome et câblez-le avec une carte huit relais et un feeder. Dans le prochain article, nous verrons la gestion des relais d'alimentation et d'inversion de courant ainsi que la construction du tableau de commande, sans oublier la motorisation des aiguilles. À la fin du prochain article, vous n'interviendrez que sur le tableau de commande pour faire évoluer vos trains ! ■