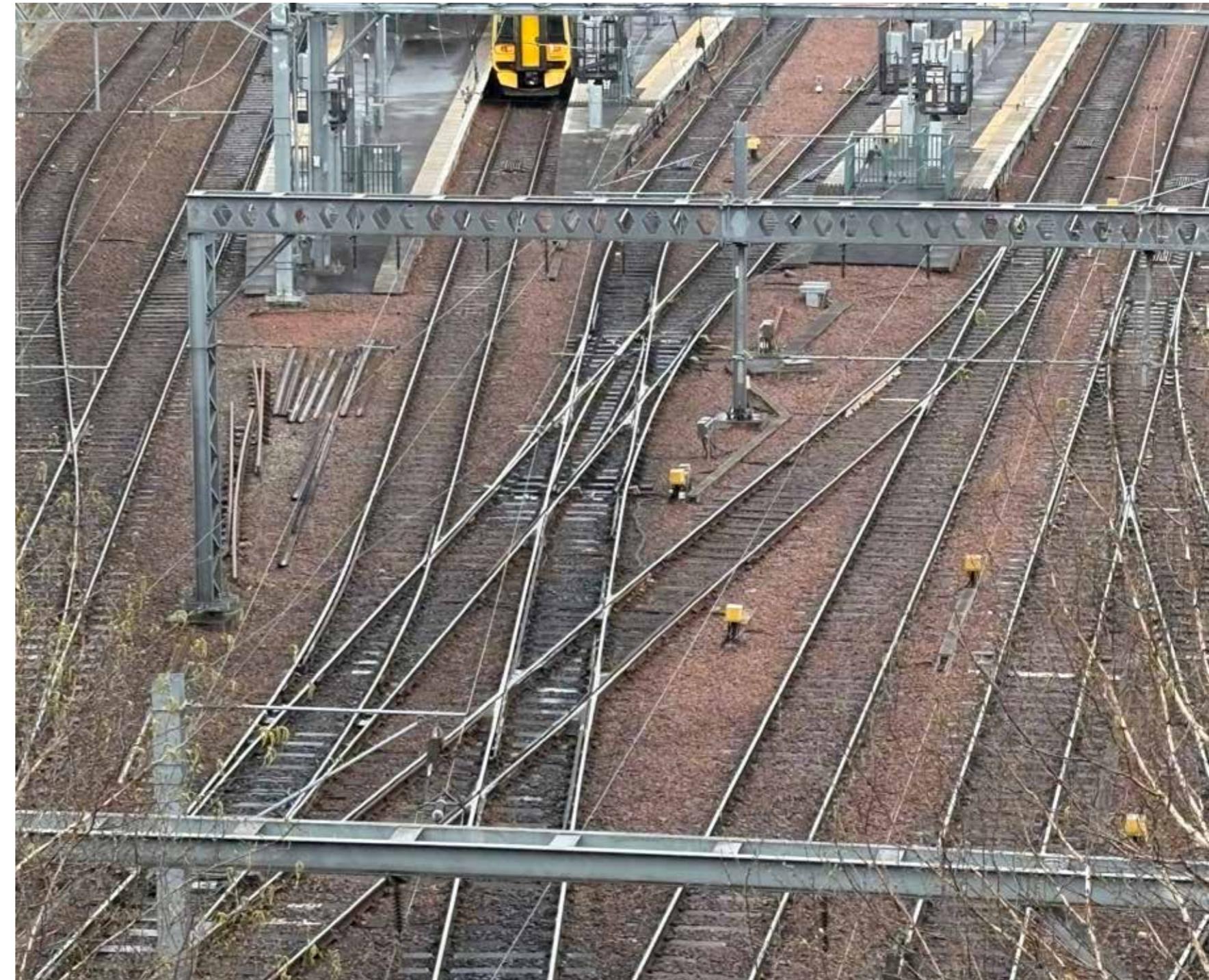


07 Montages utiles sur un réseau miniature

Dans ce chapitre, je vais vous faire découvrir quelques montages qui peuvent être reproduits et améliorés pour figurer sur votre réseau. Ils ont été développés et testés sur le simulateur Wokwi et les liens fournis vous permettront de récupérer le programme et de les voir fonctionner, comme cela avait été fait dans le tome 1.



Levier d'aiguillage
à la SNCF



Le gril de la gare d'Edimbourg

7.1 / CLIGNOTEMENT MULTIPLE

Le décor d'un réseau miniature comprend beaucoup de LED clignotantes de différentes couleurs : jaune pour un véhicule d'intervention (travaux, dépannage), bleu pour la police, rouge pour les pompiers, etc. Les rythmes de clignotement ne sont pas toujours les mêmes. S'il est simple de faire clignoter une LED, c'est toujours plus difficile d'en faire clignoter plusieurs à des rythmes différents. Grâce à la puissance du langage C, cela devient en fait assez facile. Le montage de la [figure 7-1](#) montre un exemple pour faire clignoter 5 LED avec des fréquences allant de 0,5 Hz à 4 Hz ; encore une fois, la carte Uno peut être remplacée par une Nano.

Ce montage a été décrit dans la [fiche pratique III.87 dans Loco-Revue 930 de janvier 2025](#).

L'instruction qui concerne une LED est : `digitalWrite(LED0, (millis() % 2000) < 100);` Le premier nombre (2000) est la période voulue pour le clignotement (ici 2 sec) et le deuxième nombre (100) est le temps pendant lequel la LED doit être allumée (100 millisecondes, soit 0,1 sec). Cette LED bleue va émettre un flash de 0,1 seconde toutes les 2 secondes. Si le deuxième nombre est la moitié du premier, alors la LED est autant allumée qu'éteinte (clignotement symétrique), ce qui est le cas pour les 4 autres LED (LED1 à LED4). Avec une carte Uno ou Nano, on peut ainsi faire clignoter comme on le veut jusqu'à 20 LED.

Vous pouvez récupérer le programme ici :

<https://wokwi.com/projects/406095025547560961>

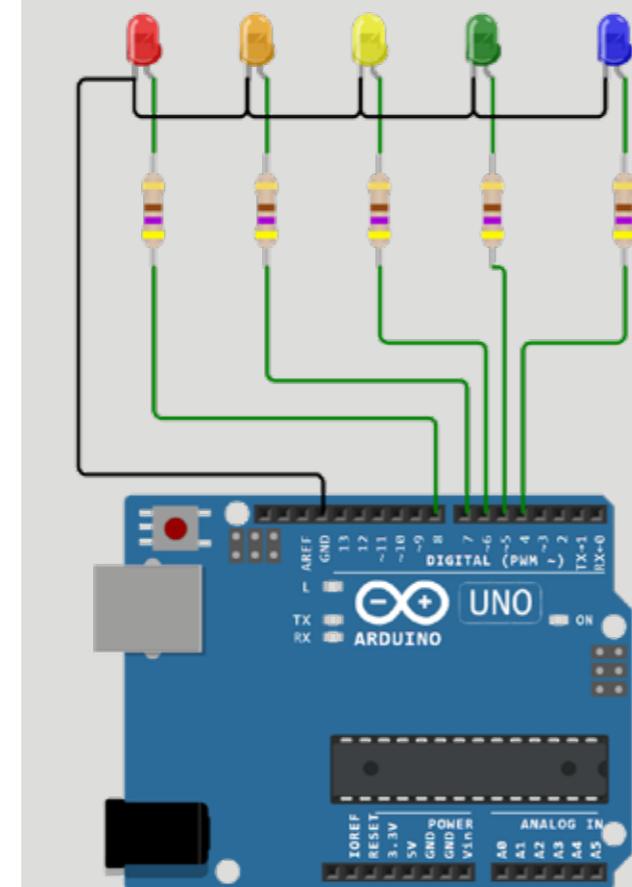


Figure 7-1

7.2 / RÉGLAGE DE SERVOMOTEUR D'AIGUILLES

On peut [utiliser un servomoteur pour réaliser un mouvement lent d'aiguilles](#), mais il est nécessaire de repérer la position « aiguille droite » et la position « aiguille déviée ». En général, ces deux positions sont à un certain angle, autour de la position médiane 90° du servomoteur. Celui-ci doit donc être positionné à 90° avant d'installer le palonnier commandant la tige. La [référence 6170 de chez Décapod](#) permet de trouver cette position médiane : il suffit d'alimenter le module via un câble mini-USB et de brancher le servomoteur pour qu'il se mette à la position 90°. Décapod vend un ensemble [support et servomoteur](#) pour un prix modique, sous la [référence 6171](#). L'angle de déplacement du palonnier qui commande l'aiguille dépend de la géométrie de la voie, de l'échelle et de l'écartement pratiqués, et de l'épaisseur du plateau. Supposons que cet angle soit 20° de part et d'autre de la position médiane, cela signifie que le servomoteur doit aller de 70° à 110°. Selon le soin apporté au montage, il se peut que ces valeurs ne soient pas tout à fait justes et qu'en réalité, le servomoteuraille de 68° à 105°. Le montage proposé par la [figure 7-2](#) permet de trouver les valeurs exactes qu'on pourra ensuite entrer dans le programme de la carte qui gère le réseau (on peut utiliser une carte Uno à la place de la Nano).

L'utilisation est simple : la position du servomoteur est affiché sur l'écran OLED, le bouton vert permet de diminuer la position de 1° à chaque appui et le bouton bleu de l'augmenter. On peut aussi rester appuyé ce qui entraîne le servomoteur dans un sens ou dans l'autre. Lorsque les lames sont presque en position, il faut y aller degré par degré jusqu'à ce qu'elles soient bien

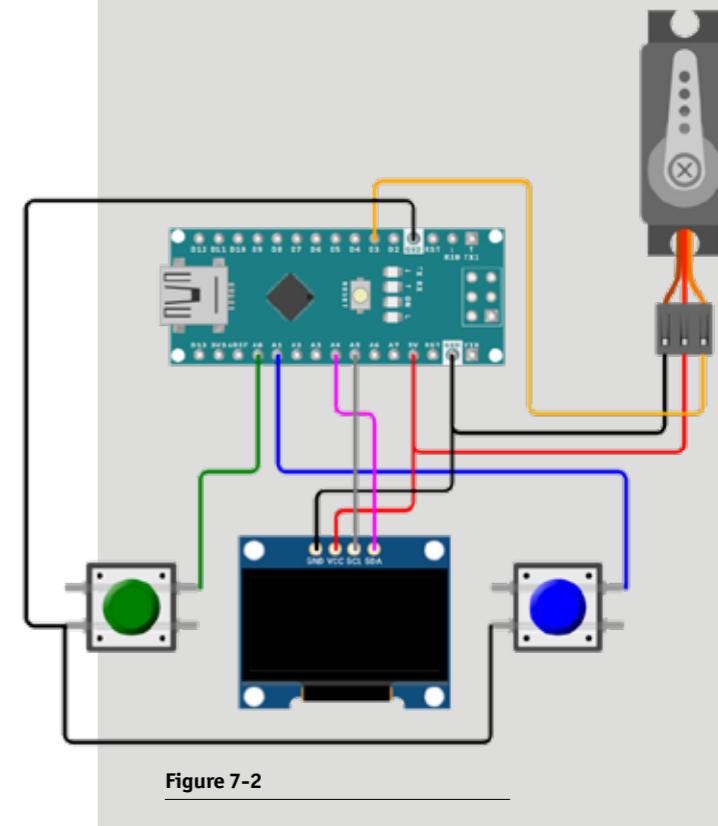


Figure 7-2

collées. Ne pas insister au-delà car cela pourrait endommager la transmission, voire le servomoteur. Lorsque la position est atteinte, il suffit de noter la valeur. Le montage peut être alimenté avec une simple pile 9 V (sur Vin et GND) et c'est l'occasion de concevoir un circuit imprimé pour recevoir le Nano, les deux boutons, l'écran OLED et une sortie à trois broches pour le servomoteur.

Vous pouvez récupérer le programme ici :
<https://wokwi.com/projects/428556837997113345>

RÉSUMÉ

Ce montage constitue un outil que tous les modélistes devraient posséder et une bonne occasion pour apprendre à dessiner un PCB recueillant l'ensemble des composants.

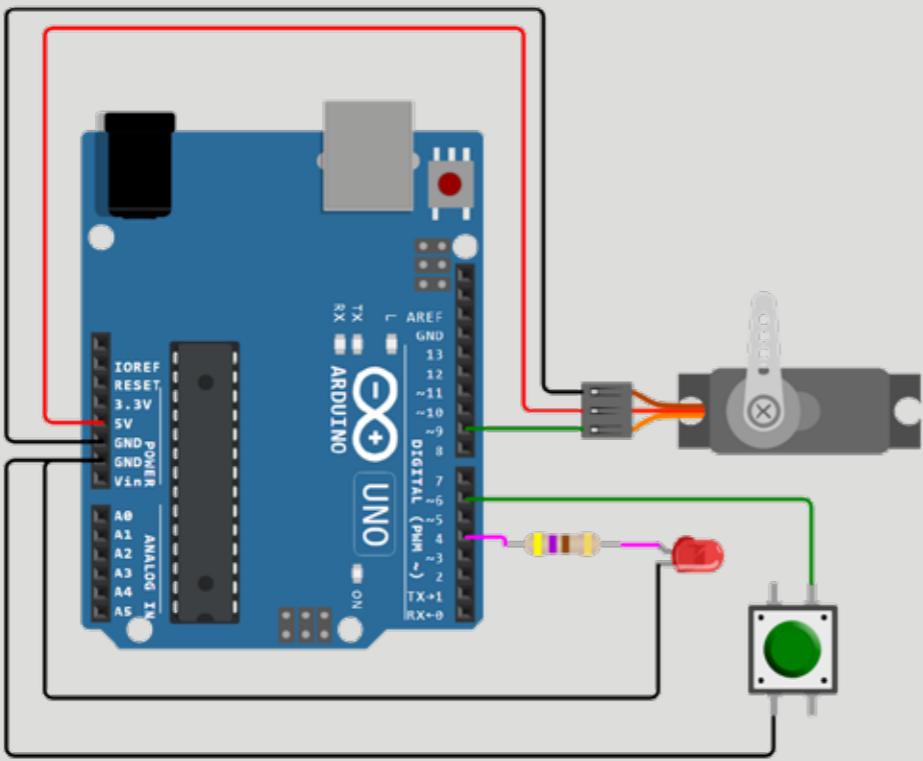


Figure 7-3

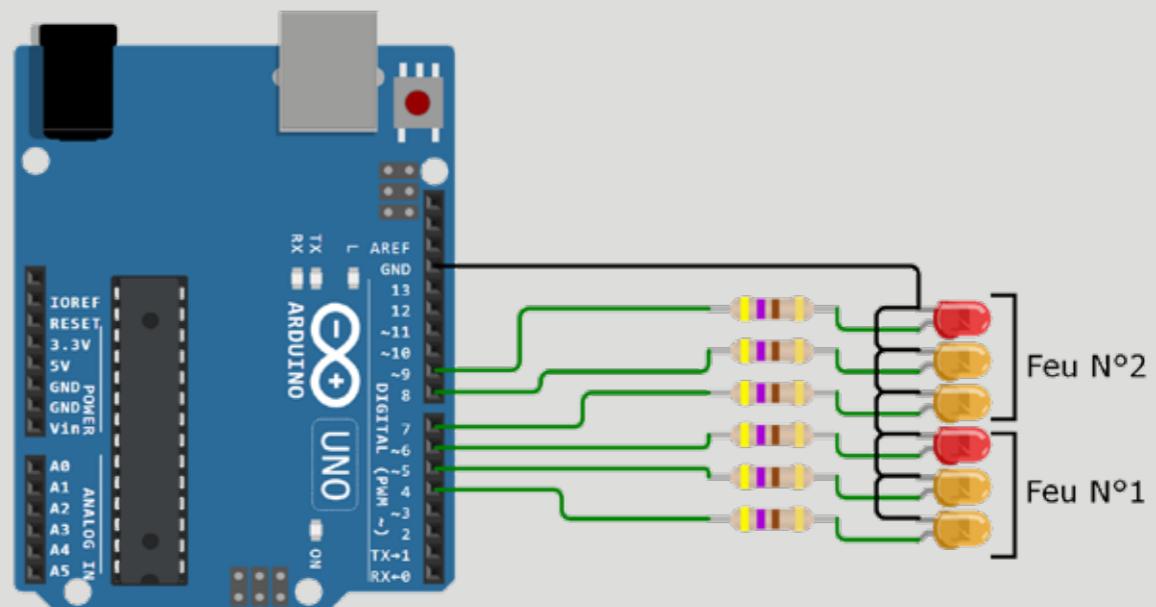


Figure 7-4

7.3 / PASSAGE À NIVEAU DÉCLENCHE PAR UN BOUTON POUSSOIR

Le montage de la **figure 7-3** reproduit le **fondement d'une barrière de PN** : feux qui clignotent, puis fermeture de barrière, clignotement des feux jusqu'à la réouverture de la barrière. Le poussoir vert permet aussi bien de fermer que d'ouvrir la barrière.

Pour installer ce PN sur un réseau, il suffit de mettre **deux I.L.S en parallèle à la place du poussoir** : le premier I.L.S déclenché fermera la barrière et le deuxième l'ouvrira. Les I.L.S doivent être placés suffisamment loin du PN pour permettre la manœuvre de la barrière avant que le train n'arrive. Vous pouvez aussi réduire le temps de clignotement avant fermeture (réglé à 5 secondes) en modifiant la ligne 37. Le train peut

circuler dans les deux sens. **Le programme de ce petit PN a été écrit par l'IA ChatGPT**.

Vous pouvez récupérer le programme ici :
<https://wokwi.com/projects/404910709193537537>.

RÉSUMÉ

Il y a toujours la place sur un réseau pour y mettre un passage à niveau. Avec un deuxième servomoteur, on peut disposer de deux barrières. Le poussoir peut être remplacé par un détecteur qui permet d'automatiser les barrières en fonction de l'arrivée d'un train.

7.4 / FEU DE CIRCULATION ALTERNÉE

Si on trouve dans le commerce des reproductions de feux tricolores de carrefour, dont la séquence d'allumage n'est pas toujours bien respectée, on ne trouve pas à notre connaissance de **feux de circulation alternée** suite à la fermeture d'une des deux voies (travaux par exemple). Voilà une idée pour nos industriels, d'autant que j'ai mis le programme dans le domaine public et qu'ils peuvent l'utiliser. La **figure 7-4** montre le montage : la carte Uno peut être remplacée par une carte Nano, voire même un ATtiny84 (voir le tome 1).

Les six LED constituent deux feux dont les ampoules vertes ont été remplacées par des oranges pour rappeler qu'il faut rester prudent même si on est autorisé à passer. La séquence avec les deux feux au rouge est assez longue

pour laisser le temps de dégager la voie avant d'autoriser l'autre côté à passer. **Le programme a été écrit par ChatGPT à qui j'avais demandé d'utiliser ma bibliothèque LightEffect décrite dans le tome 1, et il a fini par y arriver.**

Vous pouvez récupérer le programme ici :
<https://wokwi.com/projects/405092208869763073>.

RÉSUMÉ

Ce programme démontre qu'on a toujours intérêt à utiliser des bibliothèques dans un projet afin de ne pas avoir à réinventer la roue. Ici, LightEffect permet d'obtenir différentes animations lumineuses pour le décor du réseau.

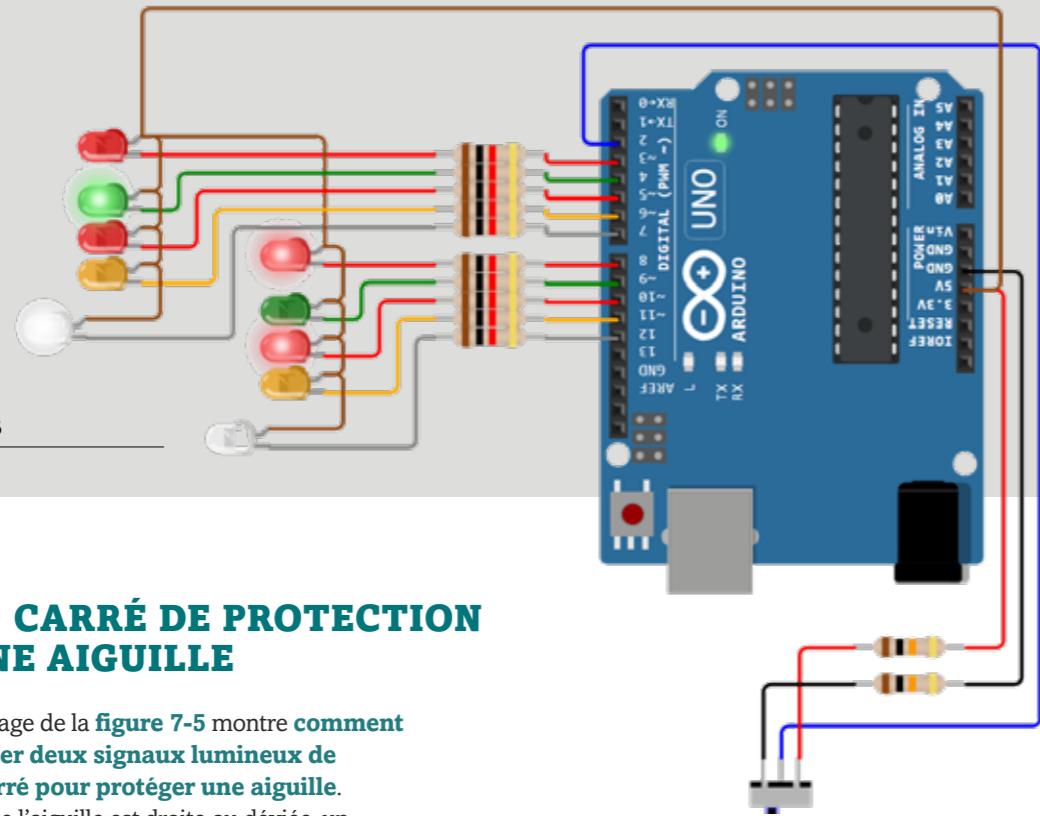


Figure 7-5

7.5 / CARRÉ DE PROTECTION D'UNE AIGUILLE

Le montage de la figure 7-5 montre comment alimenter deux signaux lumineux de type carré pour protéger une aiguille. Selon que l'aiguille est droite ou déviée, un des signaux est au vert (voie libre) et l'autre en carré (aiguille non positionnée pour la voie).

Sur un réseau, l'interrupteur peut être remplacé par un des interrupteurs de fin de course du moteur de l'aiguille, l'autre interrupteur alimentant la pointe de coeur. Les **signaux sont à anodes communes** mais peuvent être remplacés par des signaux à cathodes communes : dans ce cas, le fil commun (marron) doit être relié au GND et il faut modifier les lignes 10 et 11. Remarquez que les signaux sont tous les deux branchés sur des sorties qui se suivent et dans l'ordre des feux de haut en bas (rouge, vert, rouge, orange, cœilleron). On peut alors se contenter de ne déclarer que le branchement de la première LED et le programme calcule les autres positions. Notez que la fonction `setSignal()` a été écrite par ChatGPT où l'état du signal est représenté par une chaîne de caractère (voie libre, carré) entre guillemets.

Vous pouvez récupérer le programme ici :
<https://wokwi.com/projects/428500889162145793>.

RÉSUMÉ

Les aiguilles sont nombreuses sur un réseau miniature et leur protection par une signalisation lumineuse constitue une animation très facile à mettre en place. Vous pouvez vous inspirer de la fonction écrite par l'IA pour animer n'importe quel type de signal, surtout si vous la combinez avec le montage 7-1 qui vous permettra de faire aussi clignoter les lampes du signal.

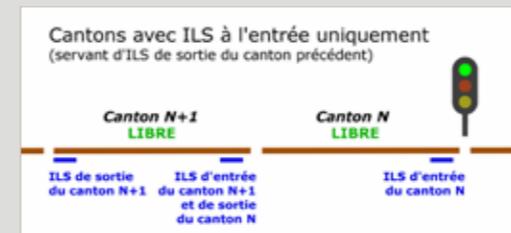


Figure 7-6

les cantons sont représentés séparés mais les rails ne sont pas tronçonnés

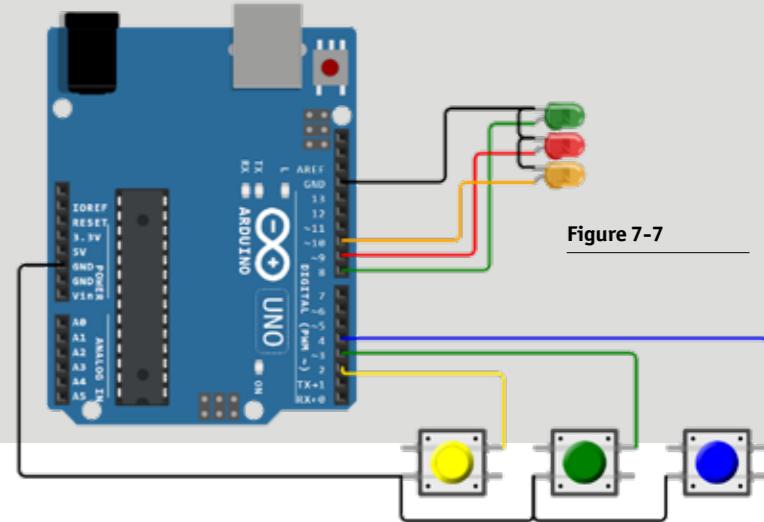


Figure 7-7

7.6 / SIGNAL DE BLOCK AUTOMATIQUE LUMINEUX AVEC DEUX CANTONS FICTIFS

Un **canton fictif** est une portion de voie comprise entre deux I.L.S ; il n'y a donc pas de rail à tronçonner et à réalimenter en courant, ce qui fait que ce système peut facilement être ajouté sur un réseau sans avoir trop de travail à faire, juste planter les I.L.S. Avec trois I.L.S, on peut donc définir deux cantons fictifs qui se suivent et placer un signal lumineux à trois feux à l'entrée du premier canton (figure 7-6). Lorsque le train entre sur le premier canton, le signal passe au rouge. Lorsque le train a libéré le premier canton et se trouve sur le deuxième, le signal est à l'orange. Lorsque le train a libéré les deux cantons, le signal redevient vert. Le montage s'occupe uniquement du signal lumineux et pas de l'arrêt des trains, mais vous pouvez rajouter cette fonctionnalité. Les trains doivent être munis d'un aimant sous la locomotive et sous le dernier wagon, selon le principe décrit dans la fiche pratique III.88 du Loco-Revue 931 de février 2025.

La figure 7-7 montre le montage à réaliser ; pour la simulation, les I.L.S ont été remplacés par des poussoirs qu'il faut appuyer en imaginant le déplacement du train. Chaque canton a

un **compteur** qui est incrémenté chaque fois qu'un I.L.S d'entrée est déclenché et décrémenté chaque fois qu'un I.L.S de sortie est déclenché. Pour qu'un canton soit libre, il faut que son compteur soit à zéro. Le système fonctionne aussi si tous les véhicules (locomotive et wagons) sont munis d'un aimant : cela revient à compter les véhicules entrants et les véhicules sortants : si les nombres ne sont pas identiques, c'est qu'une rupture d'attelage a eu lieu et des wagons ont été abandonnés sur le canton qui n'est alors pas libre. En fait, il est plus simple de mettre des aimants uniquement à chaque extrémité du train.

Vous pouvez récupérer le programme ici :
<https://wokwi.com/projects/428565479294163969>.

RÉSUMÉ

Le cantonnement d'un réseau est à envisager dès qu'on veut automatiser la circulation des trains et ce montage vous montre une solution possible. Ici, seule la signalisation lumineuse est commandée mais il n'est pas compliqué d'y rajouter des zones d'arrêt devant un sémaforo.

7.7 / GARE CACHÉE AUTOMATIQUE

Souvent située à l'arrière du réseau, la **gare cachée** permet de stocker des trains afin de faire varier les circulations. Celle que je vous propose est constituée de quatre voies dont 3 peuvent accueillir un train et les six aiguilles sont motorisées par des servomoteurs. L'ensemble est fait **pour un seul sens de circulation** mais peut être reproduit dans sa totalité pour le sens opposé ; dans ce cas, on peut utiliser une deuxième carte Uno et mettre tous les poussoirs verts sur le même tableau de commande (voir plus loin), ou bien utiliser une carte Mega qui permettra aussi d'avoir plus de voies. La **figure 7-8** montre la géométrie de la gare : quatre voies numérotées de 0 à 3 et six aiguilles A1 à A6 qui constituent les aiguilles d'entrée de gare et A4 à A6 qui constituent les aiguilles de sorties. Sur chaque voie, une zone d'arrêt ZA est commandée par un relais 1RT. La zone d'arrêt ZA0 (voir **figure 7-9**) permet de protéger l'entrée en gare : le train suivant s'arrête dessus pendant que le train précédent entre ou sort. Les zones d'arrêt ZA1 à ZA3 sont là pour que le train s'arrête ; elles ne sont alimentées que pour faire partir le train qui est stocké sur la voie.

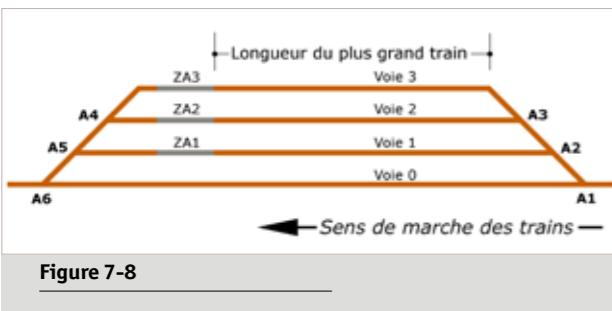


Figure 7-8

La **figure 7-9** montre l'implantation de la gare sur le réseau : on constate une balise d'entrée et une balise de sortie (deux I.L.S) dont on précisera les rôles un peu plus loin. La **balise d'entrée** est située près de la gare alors que la **balise de sortie** est située plus loin pour être certain que le train a bien dégagé l'emprise avant d'autoriser

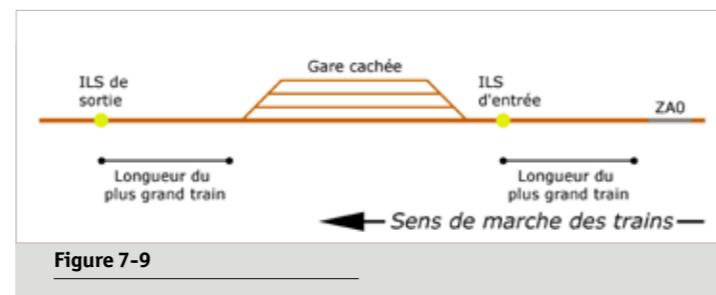


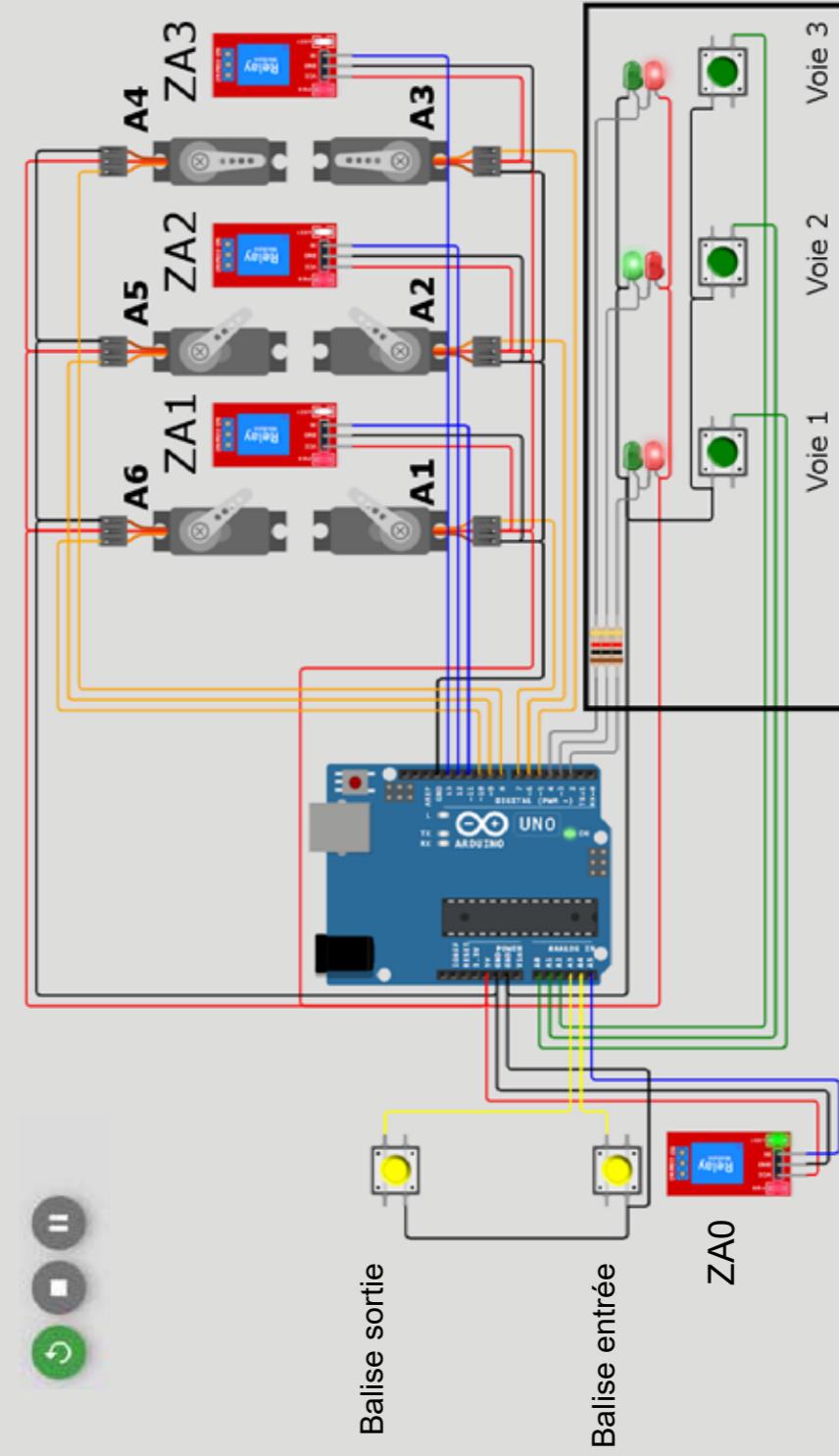
Figure 7-9

d'autres trains à utiliser la gare. Par contre, la ZA0 est située assez loin de la gare **pour bien espacer deux trains qui se suivent**.

Le principe de l'automatisme de cette gare est assez simple : on recherche si une voie est possible pour l'arrivée et le stockage d'un train. Cette voie d'arrivée est **la voie libre située la plus à droite dans le sens de circulation des trains**. S'il existe bien une voie d'arrivée, les aiguilles sont positionnées vers elle ; ainsi le train peut entrer vers la voie et s'arrête puisque les ZA sont coupées en permanence. Si les trois voies sont occupées, alors il n'y a pas de voie d'arrivée possible et **le train passe son chemin** sur la voie V0 sans s'arrêter. L'état d'occupation des voies est signalé par deux LED verte ou rouge sur un tableau de commande. Pour libérer un train, il suffit d'appuyer sur le poussoir vert lié à la voie.

Il y a donc un certain timing à respecter : un train se présente à la balise d'entrée, on coupe la ZA0 pour protéger la gare et on va attendre un certain délai qui doit être réglé pour votre réseau (longueur des voies, vitesse des trains) de manière à laisser le temps au train pour rejoindre sa voie et s'y arrêter. À la fin du délai qu'on choisira bien supérieur au temps réel, on autorise à nouveau l'utilisation de la gare en alimentant la zone ZA0.

Pour faire sortir un train, il faut protéger la gare, positionner les aiguilles de sortie, puis alimenter la ZA de la voie concernée. On pourrait utiliser un délai pour considérer que le train est sorti et la



Le prochain train passera voie : 0
Le train libère la voie : 2 (Pensez à cliquer sur la balise de sortie)

Le prochain train sera stocké voie : 2

Le prochain train sera stocké voie : 0

Le train libère la voie : 2 (Pensez à cliquer sur la balise de sortie)

Figure 7-10

gare utilisable à nouveau. Mais imaginons qu'un problème se produise et que le train ne démarre pas : **au bout du délai, la voie serait considérée libre alors qu'elle ne l'est pas**. On ajoute donc une balise de sortie qui donne un signal lorsque le train la survole, preuve qu'il s'est suffisamment éloigné de la gare. C'est la raison pour laquelle cette balise est assez loin de la gare (**figure 7-9**). Pour des raisons de sécurité, l'initialisation se fait **en considérant que toutes les voies sont occupées** et c'est au joueur de libérer les voies qui sont libres, mais dans ce cas, il n'y a pas de train pour déclencher la balise de sortie : ceci peut être fait manuellement **en branchant un poussoir en parallèle sur l'I.L.S**, qui, appuyé, reviendra au même que l'I.L.S déclenché par un aimant.

Pour réaliser cet automatisme, il vous faut 3 LED vertes et 3 LED rouges, 3 résistances pour les LED, quatre relais de type 1RT, 6 servomoteurs, 4 boutons poussoirs et deux I.L.S. Et bien entendu de quoi fabriquer la gare, six aiguilles et des coupons droits. La **figure 7-10** montre que tout tient sur une carte Uno et **donne le schéma complet** de montage. Le cadre noir est le tableau de commande de cette gare cachée, les poussoirs jaunes sont des I.L.S sur le réseau (le poussoir en parallèle de la balise de sortie n'est pas représenté), les relais sont appelés comme les zones d'arrêt qu'ils commandent et les servomoteurs reprennent la numérotation des aiguilles de la **figure 7-8**. Les servomoteurs n'étant manœuvrés qu'un par un, l'alimentation par la carte Uno devrait suffire : au besoin prévoyez une alimentation à part. Si la

LED verte située sur une des cartes relais est allumée, c'est que la zone d'arrêt est alimentée.

Le commentaire de tête du programme donne les consignes d'utilisation. Pour la simulation, il faut cliquer sur les balises (poussoirs jaunes) mais sur le réseau, l'I.L.S sera déclenché lorsque le train le survolera. Pour libérer un train, il faut appuyer le poussoir correspondant à la voie et attendre que le train déclenche la balise de sortie. Il est donc important de bien regarder les affichages du moniteur pour comprendre les actions à faire.

Vous aurez à modifier certaines variables comme la durée d'entrée en gare (fixée à 5 secondes pour la démonstration) et les positions des servomoteurs en fonction de la voie qu'ils desservent.

Vous pouvez récupérer le programme à cette adresse : <https://wokwi.com/projects/429480671346128897>.

Après ces quelques montages très simples, il est temps d'aborder des projets plus complexes. Je vous en propose trois dans les chapitres qui suivent.

RÉSUMÉ

La gare cachée permet de faire varier les convois qui roulent sur un réseau ; c'est donc un élément essentiel en automatisation. Ce montage montre qu'un programme peut gérer une telle gare et limiter les interventions au choix du convoi à faire sortir.

RÉSUMÉ DU CHAPITRE 7

Les différents montages que je vous ai proposés dans ce chapitre peuvent constituer une base pour des projets d'automatisation plus sophistiqués. En effet, les grandes composantes d'un réseau sont les aiguilles, le cantonnement, le passage à niveau et la gare cachée. Chaque composante peut être gérée par sa propre carte Arduino qu'on peut faire communiquer avec les autres cartes. Il devient alors facile de constituer un ensemble complet de gestion de réseau.

