

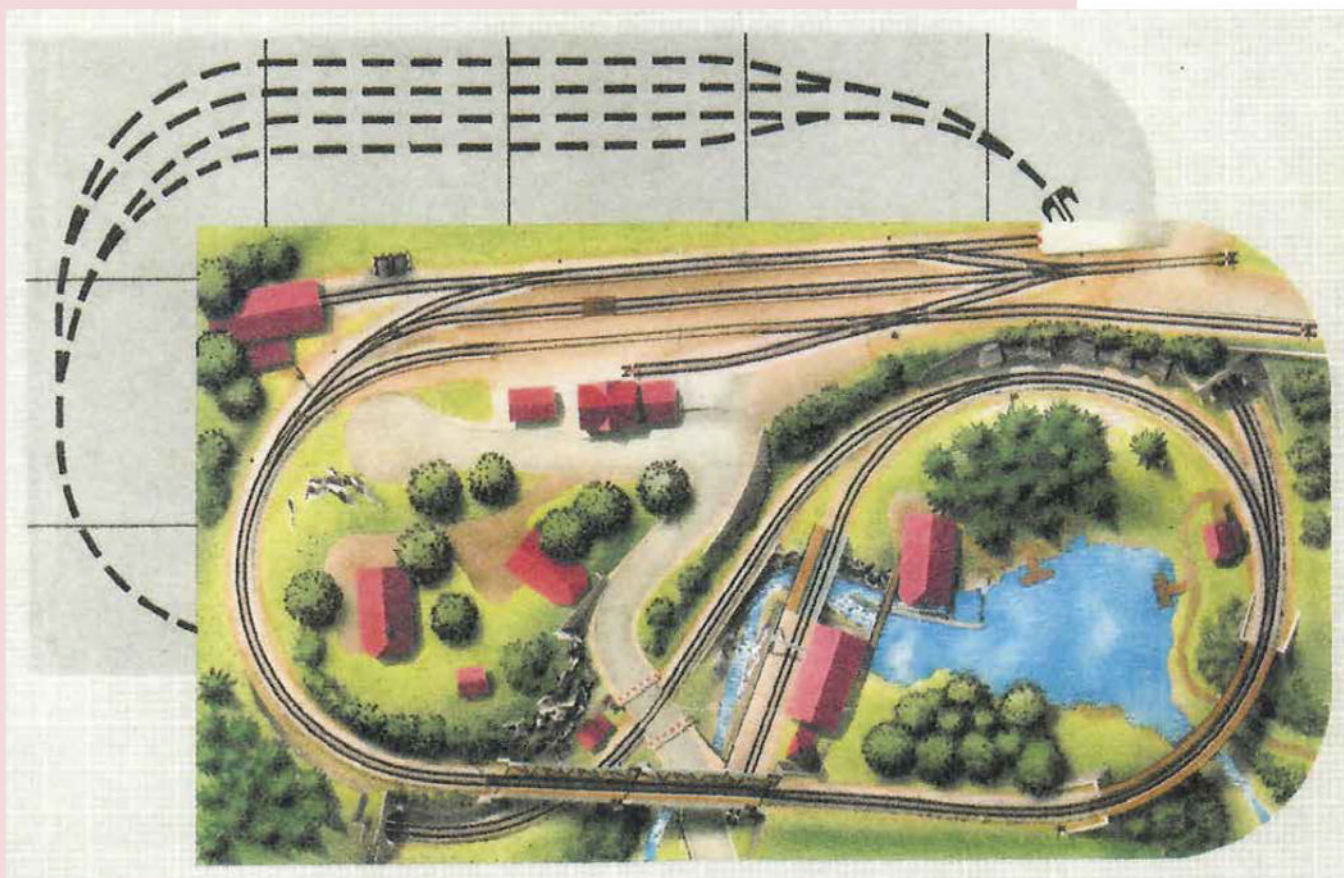


Figure 1. Le réseau de référence a été publié dans le livre « Le réseau miniature Plans & Projets », édition de 1992.

MODERNISONS NOTRE TCO

(Tableau de commande optique)

Cet article explique comment, et surtout pourquoi, passer d'un TCO électromécanique à un TCO informatisé à base de carte Arduino et de Processing. Rien de difficile mais la tâche dépend de votre niveau initial dans les différents domaines abordés ici. Évidemment, cet article ne décrit pas tous les détails, mais il vous indique où trouver les informations pour franchir chaque étape avec succès.

Texte et illustrations : Christian Bézanger
christian.bezanger@gmail.com (sauf mention contraire)

Comme beaucoup d'entre nous, lorsque j'ai commencé le modélisme ferroviaire, j'étais en appartement et donc confronté à un problème de place. Je me suis alors contenté de construire un petit réseau analogique sur un plateau de 1,30 x 0,70 m sur deux niveaux. Le plan était inspiré d'un réseau figurant en page 31 du livre « Le réseau miniature Plans & Projets », édition de 1992, publié par Loco-Revue. La **figure 1** montre ce réseau construit à l'échelle N : une gare terminus au niveau supérieur, une boucle de retournement alimentant une gare cachée au niveau inférieur et un embranchement de particulier vers une ►►

► scierie. Tout était là pour constituer un réseau laboratoire permettant de découvrir les techniques du modélisme ferroviaire (menuiserie, électricité, décor...). Pour des raisons de place, la gare cachée était limitée à deux voies au lieu de quatre, mais je pouvais faire circuler un train alors qu'un autre était en attente dans cette gare; et je pouvais aussi stocker quelques engins sur des voies de garage. Ce réseau était prometteur pour des séances de jeu très diverses, mais aussi pour découvrir différentes techniques à mettre en œuvre. Par exemple, la gestion de la boucle de retournement se fait grâce à de simples diodes alimentant ou non les sections d'arrêt en fonction de la polarité du courant (principe d'un garage alimenté uniquement dans le sens sortie). Ainsi, un train qui arrive sur la boucle marque un arrêt, mais redémarre dans le même sens dès que le courant est inversé, ce qui n'est pas contraignant vu que la boucle est en partie cachée. Les aiguilles ayant des moteurs à solénoïdes, l'idée d'un TCO électromécanique s'est d'abord imposée afin de les commander depuis un seul poste ainsi que les voies de garage.

TCO électromécanique

La **figure 2** montre le TCO électromécanique commandant le réseau. Sur la face avant, on voit l'arrivée du courant de traction (bleu et rouge) et l'arrivée du courant alternatif alimentant les accessoires (noir et blanc), notamment les moteurs d'aiguilles. Sur la face arrière, deux prises informatiques DB 25 alimentent le niveau inférieur et le niveau supérieur (une prise par niveau) afin d'envoyer les courants aux solénoïdes des aiguilles ou aux rails des voies de garage. Sur le dessus, on reconnaît le plan du réseau avec une couleur différente par canton, des interrupteurs pour isoler certaines voies et des LED de contrôle donnant par exemple la position réelle des aiguilles qui sont cachées ou l'état d'occupation (LED rouges) des voies de la gare cachée. Les voies sont numérotées et les aiguilles sont repérées par une lettre (de A à G). La TJD (traversée jonction double) est en quelque sorte l'équivalent de deux aiguilles et possède deux moteurs à solénoïdes (H et I). Je dois avouer que j'ai très vite revu mes prétentions à la baisse concernant les aiguilles, et seules celles qui sont cachées sont commandées et contrôlées sur le TCO. Pour les aiguilles visibles, la commande est manuelle et le contrôle se fait visuellement. Malgré cela, ce TCO était parfaitement opérationnel et après lui avoir

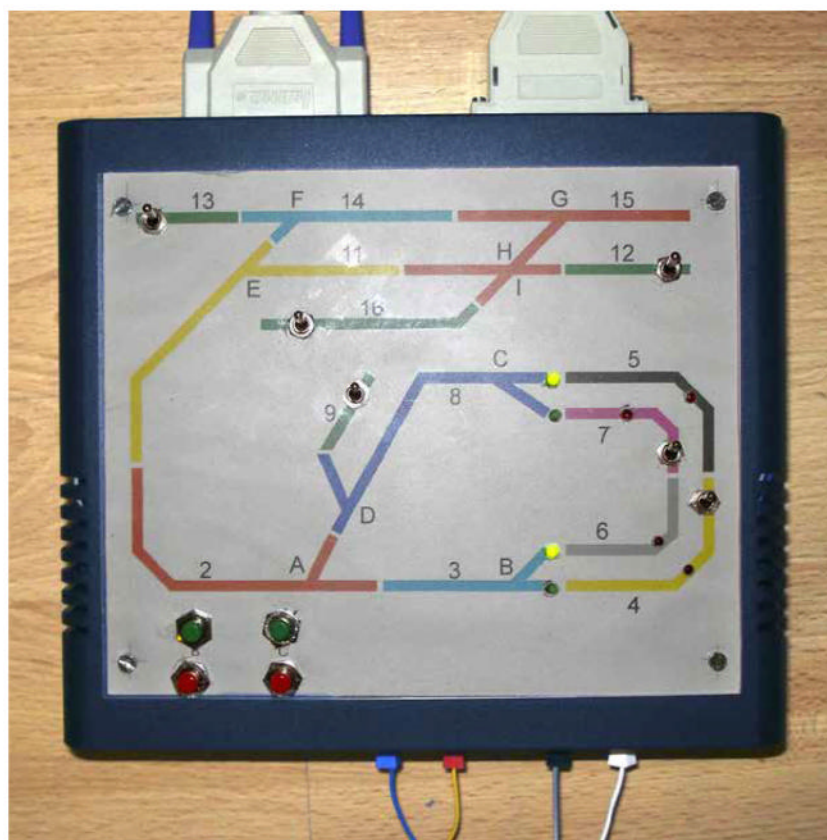


Figure 2. Un premier TCO électromécanique va être modernisé.

ajouté un petit automatisme pour inverser le courant de traction, je voyais mon train tourner et je pouvais lui faire emprunter plusieurs itinéraires. Pourtant, je me suis vite aperçu des limites d'un tel système : les itinéraires doivent être soigneusement contrôlés avant de faire partir le train, sinon il risque de ne pas aller là où on veut, et, par exemple, risquer de se retrouver sur une voie de la gare cachée qui est occupée par un train à l'arrêt. Le principal défaut était un manque de sécurité sur les itinéraires constitués.

TCO informatisé

Pour pallier ce manque de sécurité, l'idée d'un TCO interactif dessiné sur l'écran d'un ordinateur s'est imposée très vite. Il suffirait d'un clic de souris sur ce TCO pour commander une aiguille ou pour alimenter une voie de garage. Et le logiciel capable de gérer cela, pourrait en plus ajouter des règles de sécurité sur la formation d'itinéraires, par exemple empêcher la sortie de gare si les aiguilles ne sont pas dans la bonne position. On voit donc l'intérêt d'un tel projet : non seulement commander tout un réseau à partir d'un seul poste de commande avec une simple souris (ou bien un écran tactile), mais aussi rajouter des règles de circulation afin d'éviter

la collision de deux trains ou simplement un déraillement. Je pouvais concevoir que les interrupteurs de mon TCO électromécanique seraient remplacés par des relais commandés par une carte Arduino, que les LED seraient supprimées car le dessin à l'ordinateur pourrait donner toutes les informations requises. Je pouvais aussi imaginer qu'un TCO informatisé est plus facile à faire évoluer qu'un TCO classique, si je voulais ajouter d'autres voies ultérieurement. L'idée était belle, mais il restait encore deux problèmes de taille à résoudre : comment dessiner le plan du réseau à l'ordinateur et comment faire communiquer ce plan avec une carte Arduino?

Processing

Processing est un langage de programmation idéal pour dessiner, destiné à des artistes. Lorsque le projet Arduino a vu le jour, il a été conçu pour travailler avec Processing qui existait déjà. C'est donc ce logiciel qui s'impose lorsqu'il faut ajouter une partie graphique à un projet Arduino. Par chance, au moment même où je me disais qu'il faudrait que j'apprenne à utiliser Processing, quelques articles ont été publiés sur Locomotiv au sujet de Processing et du dessin de TCO par deux copains, Denis et Pierre. L'article de Denis

faisait une bonne présentation des possibilités de Processing et de la prise en main de l'interface de programmation. Les articles de Pierre expliquaient comment dessiner avec Processing des pavés permettant de reproduire sur un écran d'ordinateur n'importe quelle typologie de réseau miniature. Je vous indiquerai plus loin où trouver ces articles car ils vous seront nécessaires si vous voulez réaliser un tel projet. Inutile de vous dire que ces articles m'ont fait gagner un temps considérable. Quant à la communication entre Processing et Arduino, l'étude de la documentation me montrait que c'était possible, et même assez simple à faire.

Dessin du TCO

En partant des principes expliqués par Pierre dans son article, j'ai commencé par écrire un programme en Processing dessinant mon réseau. C'est ce que montre la **figure 3** : le dessin est constitué d'une grille de pavés permettant de représenter voies et aiguilles, grille qui est dessinée ligne par ligne. Ce dessin n'est qu'une étape, mais ne permet pas encore de faire un TCO interactif. Pour cela, il faut attribuer une variable booléenne à chaque aiguille qui vaut 'true' si l'aiguille est déviée et 'false' dans le cas contraire. Ensuite, il faut faire changer l'état de la variable quand on clique sur le pavé représentant l'aiguille. En plus, on change l'affichage du pavé en fonction de cette variable, ce qui fait qu'on dessine la position de l'aiguille, tâche réalisée avec des LED sur le TCO initial. Avec Processing, la gestion de la souris est très facile même si on peut trouver fastidieux de déterminer la zone cliquable pour chaque

pavé représentant une aiguille ou une voie de garage (ceci dépend des coordonnées du pavé dans la grille). La TJD ayant deux moteurs à solénoïdes, on peut cliquer sur les zones gris clair, soit dans le coin supérieur gauche soit dans le coin inférieur droit. On peut faire la même chose pour une voie de garage : la variable vaut 'true' (ou 1) si la voie est alimentée et 'false' (ou 0) dans le cas contraire. Et de la même façon, on peut représenter une voie non alimentée en gris foncé et en couleur si elle est alimentée. Mieux, suivant la polarité des rails, la couleur peut être différente, ce qui donne le sens de circulation. Enfin, Processing permet d'écrire sur le dessin et ainsi de numéroté les voies et les aiguilles. La **figure 4** montre le réseau en position départ (un train doit quitter la gare terminus) : le OFF est un interrupteur virtuel et il suffit de cliquer dessus pour alimenter la voie qui change de couleur pendant que le OFF rouge devient un ON vert. Dans la gare cachée, un ON vert ne signifie pas que

les voies sont alimentées mais que le départ est possible : seule la couleur de voie indique si elle est alimentée. Le DEP permet de changer la polarisation des voies : dans le sens départ, les voies sont bleues, dans le sens retour, elles sont jaunes et RET remplace DEP.

Un exemple

Par exemple, sur la **figure 4**, on peut voir que le réseau est en position départ : un train stocké sur la voie 12 pourra quitter la gare dès que la voie 12 sera alimentée (pour l'instant, elle ne l'est pas et apparaît grise). Le train passera ensuite sur les voies 11, 2 et 3, puis 6 et s'arrêtera sur la voie 7.

Sur la **figure 5**, le réseau est passé en position retour (RET a remplacé DEP) : la polarité sur les voies est inversée, et la voie 7 sera alimentée si son interrupteur l'autorise (comme ici où on l'a mis sur ON). Le train passera sur la voie 8, l'aiguille A qui se sera positionnée toute seule, puis les voies 2 puis 11 et arrivera enfin sur la voie 12 où il s'arrêtera puisque cette voie est non alimentée (OFF et grise). Voici un exemple de ce que peut faire l'automatisme : la circulation en gare cachée se fait uniquement sur la voie de gauche et on peut entrer en gare cachée en passant par l'aiguille B ou C. Or, ceci dépend de la position de l'aiguille A. Selon la direction de A, les aiguilles B et C sont positionnées automatiquement pour respecter la circulation sur voie de gauche. Et autoriser une sortie d'un train de la gare cachée positionnera les aiguilles comme il se doit pour que l'itinéraire soit possible. Plutôt sympa, non ? En fait, on ne peut pas alimenter l'une des zones d'arrêt de la gare cachée s'il existe un conflit de circulation, par exemple si l'aiguille D est vers la voie 9, il est impossible d'alimenter la voie 7 pour faire sortir un train. Vraiment sympa ! Évidemment, c'est à vous de prévoir tout cela dans votre programme grâce aux règles habituelles de logique, impliquant ►►

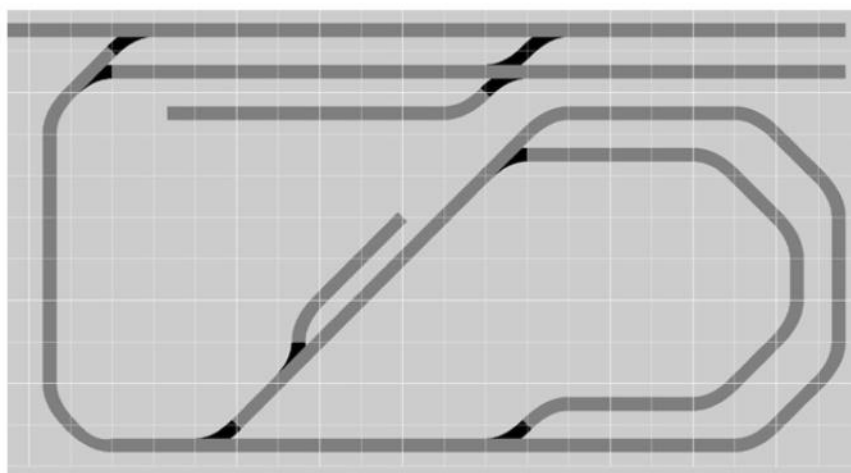


Figure 3. Plan du réseau sur ordinateur, pour commande par Processing.

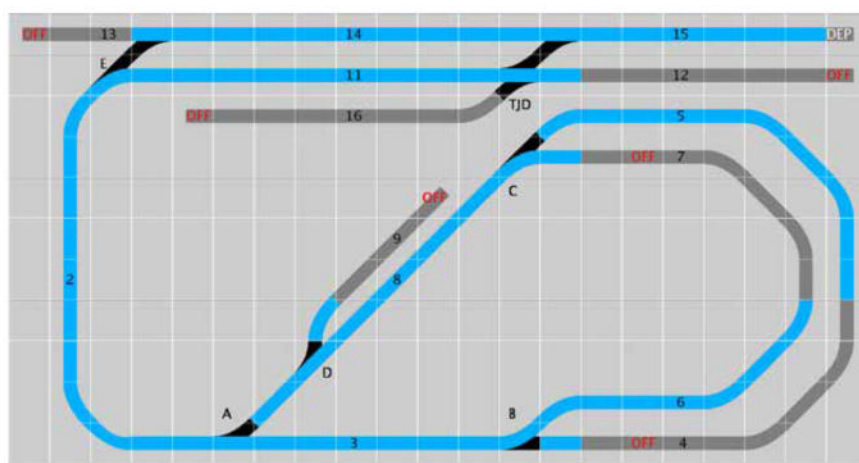


Figure 4. Le réseau est en position départ.

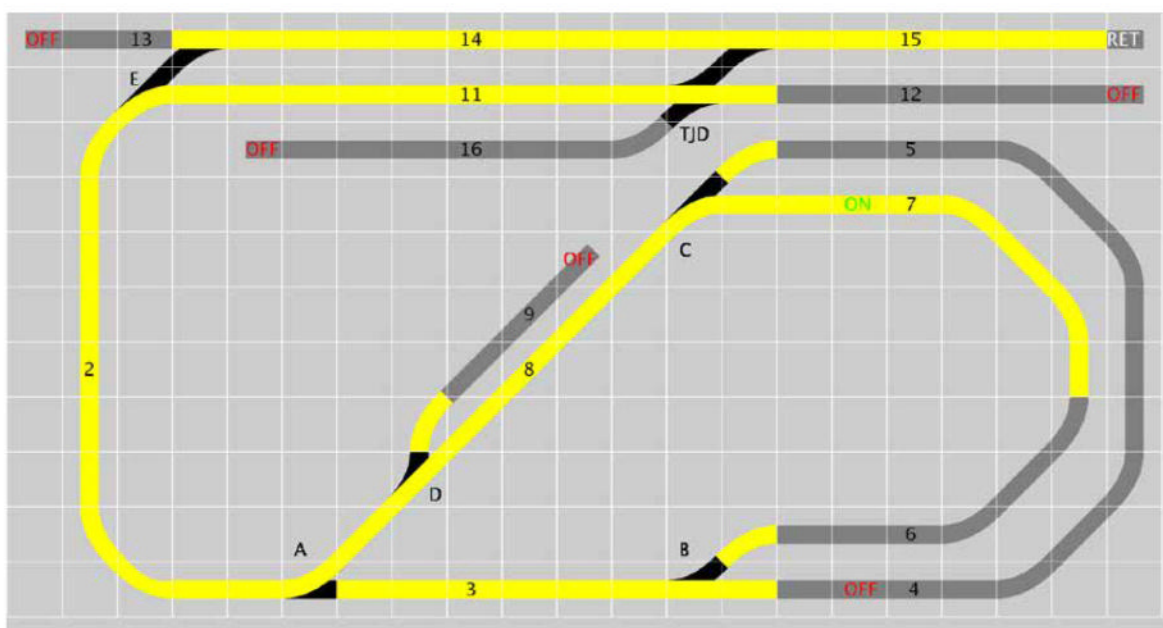


Figure 5.
Le TCO montre
le réseau en
position retour.

►► le respect de plusieurs conditions à la fois. Mais vous pouvez aller très loin et même ajouter ultérieurement ce à quoi vous n'auriez pas pensé tout de suite. Comme je le disais, l'évolution de ce TCO informatisé est simple à réaliser, même après coup.

Commande du réseau

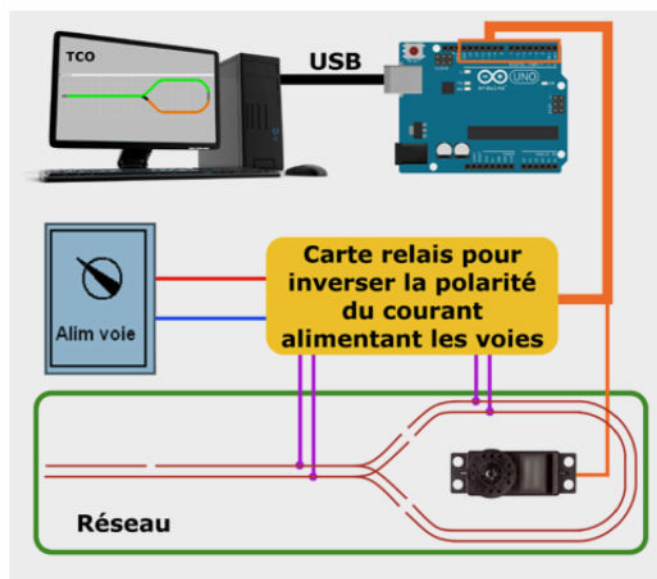
Quand on y réfléchit, la commande d'un réseau repose sur très peu de principes. J'ai écrit dans Locoduino un article qui traite de ce sujet, avec trois choses qu'il faut savoir réaliser :

- Positionner une aiguille avec Arduino (solénoïdes ou servomoteur).
- Alimenter une portion de voie avec un relais 1RT commandé par Arduino.
- Inverser le courant de traction grâce à un relais 2RT commandé par Arduino.

L'article décrit un petit réseau constitué d'une voie de garage en impasse (terminus), une aiguille et une boucle de retournement, le tout géré par Arduino. Le dessin par Processing ne nécessite que quelques pavés de base. Cet exemple est relativement simple mais traite de tout ce dont on a besoin pour un réseau plus grand.

Pour savoir ce qu'il a à faire, Arduino doit recevoir des informations de la part de Processing, et ceci est possible parce que les deux logiciels ont été conçus pour communiquer ensemble. Par exemple, vous cliquez sur le TCO pour changer l'orientation d'une aiguille : Processing change l'état de l'aiguille, dessine le nouvel itinéraire et envoie le nouvel état à Arduino via la prise USB. Ainsi Arduino peut modifier la position de l'aiguille en agissant sur le servomoteur ou sur le solénoïde concerné. La carte Arduino doit rester branchée à l'ordinateur via le câble USB, afin de recevoir les informations

Figure 6.
Architecture
du montage :
l'ordinateur sur
lequel tourne
Processing,
avec l'écran
sur lequel est
affiché le TCO.



en provenance de l'ordinateur, c'est-à-dire du programme Processing.

La **figure 6** montre l'architecture du montage à réaliser : on y reconnaît un ordinateur sur lequel tourne Processing avec l'écran sur lequel est affiché le TCO. La carte Arduino est reliée à l'ordinateur par le câble USB et reçoit donc les informations de la part de Processing, traite les demandes avec une interface qui envoie les courants nécessaires au réseau (courant de traction ou courant d'accessoires). Le tout est maintenant de construire cette interface, mais c'est en fait la partie la plus facile d'un tel projet.

Exemple d'interface

L'interface dépend de votre réseau : commandez-vous les aiguilles avec des servomoteurs ou avec des solénoïdes ? Si la réponse est servomoteurs, Arduino sait parfaitement

les commander directement. Si la réponse est solénoïdes, alors il faut passer par des relais comme cela a été expliqué dans la Fiche Pratique III. 52 de novembre 2018. Un simple relais peut permettre d'alimenter une portion de voie, et un relais 2RT permet de construire un inverseur pour changer la polarité du courant sur les rails. Ce sont des techniques de base en électricité décrites dans la littérature de modélisme ferroviaire. Pour vous donner un exemple, la **figure 7** représente l'interface que j'ai construite pour mon réseau. On reconnaît la carte Arduino Uno, deux cartes de huit relais 1RT permettant de gérer les solénoïdes d'aiguilles (de A à I), les voies de garage (4, 7, 9, 12, 13 et 16), une carte relais 2RT inverseur pour le sens de circulation et un montage à base de diodes pour gérer la boucle de retournement.

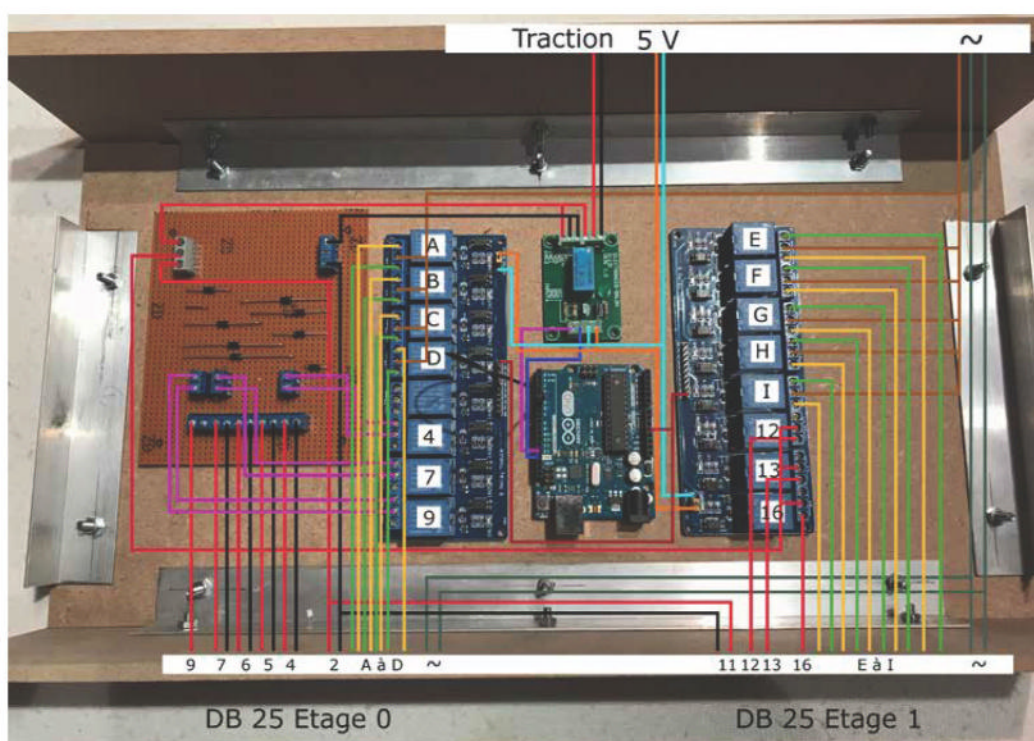


Figure 7.
L'interface
construite
pour le réseau
montré en
exemple.

Une alimentation 5 V est prévue pour les trois cartes relais, et les courants de traction et d'accessoires proviennent du transformateur. Les connexions entre Arduino et les cartes relais ne sont pas représentées : elles se font via des nappes de câbles Dupont. Les courants sont ensuite transmis au réseau par l'intermédiaire de deux prises DB 25, une par niveau, comme c'était le cas pour le TCO initial. Le boîtier qui abrite tout cela est constitué d'un couvercle récupéré sur un vieux lecteur DVD et d'une boîte en médium (un fond, une face avant et une face arrière). Les fils jaunes alimentent les solénoïdes 1 et les verts les solénoïdes 2 de chaque aiguille. Les solénoïdes étant munis de contacts de fin de course, un seul relais est requis par solénoïde, ce qui simplifie le câblage. Les fils rouges ou noirs vont vers les rails des voies. Les fils violets allaient vers des interrupteurs sur le TCO initial, qui ont été remplacés par des relais : ils permettent de stocker un train sur les voies 4, 7 ou 9. Comme vous le voyez, rien d'extraordinaire, juste un peu de câblage où il faut être rigoureux pour ne pas se tromper. Construire une telle interface est à la portée de tout le monde.

Adaptation à votre réseau

Le projet vous plaît et vous voulez en faire autant pour votre réseau : voici quelques conseils. Tout d'abord, ne vous lancez pas simplement pour faire plus moderne. Si votre nouveau TCO n'apporte rien de plus que l'ancien, cela ne sert à rien. Par contre, si vous

pouvez en profiter pour ajouter des règles de sécurité sur la création d'itinéraires, là oui, cela en vaut la peine.

La première chose à faire est de dessiner le TCO avec Processing ; pour cela, lisez les articles de Pierre sur Locoduino « Processing pour nos trains » et « TCOs en Processing ». Voyez comment il est possible de dessiner un pavé et inspirez-vous des exemples donnés pour créer vos propres pavés afin de dessiner votre réseau.

Ensuite, le dessin de votre TCO doit être interactif : en cliquant dessus, l'état d'une aiguille ou d'une voie doit changer et le changement doit se voir à l'écran. Il faudra alors apprendre à gérer des variables booléennes. Et il faudra aussi apprendre à gérer la souris, mais les exemples de Pierre sont très clairs et permettent de comprendre comment faire. Arrivés à ce niveau, vous disposez d'un TCO interactif et fonctionnel. J'avais d'ailleurs présenté le mien à Trainsmania à Lille en 2019 où les visiteurs pouvaient commander la position des aiguilles ou l'alimentation des voies, et tout cela sans avoir de réseau auprès de moi. Il reste maintenant à faire communiquer Processing avec Arduino. Il va vous falloir écrire un programme pour qu'Arduino récupère, auprès de Processing, les différentes variables représentant l'état du réseau. Pour vérifier que tout se passe bien, vous pouvez allumer des LED qui représenteront la position de futures aiguilles (on appelle cela un mock-up ou encore, une maquette d'essai). Pour établir la commu-

nication, vous pouvez vous inspirer de mon article « Ménage à trois (Ordinateur, Arduino, réseau) ». Si vous avez besoin d'en savoir plus sur Processing, l'article de Denis « Démarrer en Processing » vous mettra le pied à l'étrier. Et pour aller plus loin, la documentation sur le site Processing vous permettra d'en savoir plus. Tous ces articles se trouvent dans les rubriques « Programmation » de Locoduino ou « Projets ferroviaires (niveau 2) » (<https://locoduino.org>).

Lorsque tout vous semble fonctionnel, vous pouvez construire votre interface à base de cartes relais et/ou de servomoteurs. Pensez à avoir une alimentation à part pour les cartes relais ou les servomoteurs (ceci se trouve sur internet facilement). Sauf besoins exceptionnels, vous n'aurez pas à construire d'électronique car on trouve sur internet de petites cartes pour réaliser tout ce qu'on veut (par exemple, n'achetez pas des relais mais une carte relais avec logique de commande à base d'optocoupleurs : l'ensemble vous reviendra sans doute moins cher que construire vous-même, sans compter l'économie de temps).

Si vous êtes un peu familiarisé avec les techniques d'Arduino, apprendre à utiliser Processing ne vous demandera pas longtemps. Les articles de Locoduino constituent la base de votre futur TCO : il n'y a qu'à recopier certaines parties de programmes. Vous avez donc tout sous la main pour réussir votre entrée dans le vingt-et-unième siècle et avoir un TCO digne de ceux utilisés par la SNCF pour les circulations des TGV. ■



WWW. TRAIN-MODELISME.COM

La passion des trains miniatures !



Les rames voyageurs
Les locomotives
Les autorails
Les voitures et wagons
Les rames urbaines
Les trains de travaux



Les trains à crémaillère
La gamme économique
Les rails et accessoires...
Le digital et l'alimentation...
Le décor, la caténaire, la signalisation...
Les coffrets, l'entretien...



Prix compétitifs, STOCK important avec gestion en temps réel, expéditions 24 H !
Commandes en ligne par CB ou E-card, par courrier et par fax 02-54-28-75-01 !



Train Modélisme Rte d'Angles 36220 Néons/Creuse Tél 02-54-28-59-66 RCS 433 831 039 trainmodelisme@orange.fr

Réservez votre espace publicitaire
en contactant

LR PRESSE

Tél. : 07 67 67 14 27

Fax : 02 97 24 28 30

Mail : publicite@lrpresse.com

M. CITERNE

Trains Miniatures N - HO

Occasions - Dépôt-Vente

Digital • Roco • Lenz • Zimo • Trix

VENTE PAR CORRESPONDANCE

Ouvert du mardi au samedi : 9h-12h30 et 14h-19h

21 Bd du Temple - 75003 Paris

Tél./fax 01.42.78.00.16

Métro : République - Filles du Calvaire, Bus 20 - 65 - 96

COLLECTION
B.A.-BA
DU MODÉLISME FERROVIAIRE

2 nouveaux titres disponibles !

Soyez un créateur !



Commandez dès maintenant sur trains.lrpresse.com ou via le bon de commande en fin de numéro



B.A.-BA du modélisme ferroviaire : c'est une collection de 24 fascicules qui vous donneront les bases de ce qu'il faut savoir pour réussir son projet de réseau miniature !



Encore plus de réalisme

LR PRESSE