

## Chapitre 7

# Le panneau de contrôle

## 1. Les contrôleurs

### 1.1 À plusieurs, c'est mieux que tout seul

Si l'on équipe les trains de décodeurs numériques, c'est non seulement pour avoir un meilleur contrôle de la vitesse ou un éclairage réversible, mais surtout pour faire circuler plusieurs trains à la fois. Même s'il est toujours possible de contrôler tous les trains avec une seule commande, cela devient rapidement compliqué. Le plus simple, et aussi le plus ludique, consiste à disposer d'une commande pour chaque convoi. C'est plus convivial, puisque l'on peut s'y mettre à plusieurs et commander tous les trains simultanément.

Les centrales DCC du commerce sont généralement fournies avec un ou deux contrôleurs. Pour piloter plus de locomotives, il faut donc soit jongler entre les adresses, soit acheter des commandes supplémentaires, ce qui est rapidement onéreux, puisqu'elles sont souvent vendues aux alentours de 100 €. Pourtant, un contrôleur ne contient qu'un potentiomètre, quelques boutons et un minimum de circuiterie électronique. On doit pouvoir fabriquer les siens pour un prix bien plus modique.

## 124 Arduino - Faites-le jouer au train

Au moyen de boutons, le panneau de contrôle permet donc la commande des locomotives, qu'il s'agisse de régler la vitesse, de choisir le sens de marche du convoi, voire de piloter directement les fonctions du décodeur.



### 1.2 Un premier choix

Mais avant de multiplier les contrôleurs, il faut commencer par en gérer un seul. Et donc pour gérer les boutons qui servent à piloter les trains. La mise en œuvre d'un contrôleur dépend du câblage des boutons, et le câblage des boutons dépend du choix du contrôleur. C'est un peu l'histoire de l'œuf et de la poule, mais il faut bien commencer par quelque chose.

Il faut, avant tout, choisir la façon de disposer les contrôleurs, ce qui détermine la méthode à utiliser pour les multiplexer, car tous les contrôleurs arrivent sur la même entrée de l'Arduino. Cette entrée est différente selon la méthode choisie.

Soit les contrôleurs sont dans les boîtiers séparés, un pour chaque commande. Cela permet de les disposer à divers endroits du réseau. C'est plus pratique si l'on est plusieurs à piloter les trains. En revanche, il y a plus de boîtiers à réaliser, et comme ils sont plus légers, ils risquent plus facilement de tomber par terre. Dans ce cas, les données transitent en numérique par le bus I<sup>2</sup>C. Le circuit se chargeant de l'échantillonnage et de la communication est un PCF8591 (présenté précédemment). La numérisation se fait sur 8 bits et est limitée à quatre canaux.

Soit les contrôleurs sont regroupés tous ensemble dans un seul gros boîtier. C'est plus simple à réaliser et convient très bien si l'on est le seul à s'occuper du pilotage des trains. Cependant, rien n'empêche d'ajouter des contrôleurs individuels. On peut gérer simultanément les deux possibilités par programmation. Dans ce cas, les données transitent en analogique jusqu'aux entrées de l'Arduino, qui fait un échantillonnage sur 10 bits. Les données sont multiplexées par des interrupteurs analogiques, de référence CD4066, qui sont, eux aussi, limités à quatre canaux.

## 2. Les boutons

### 2.1 Des regroupements nécessaires

Chaque contrôleur doit permettre de piloter les fonctionnalités suivantes :

- Le réglage de la vitesse du convoi
- Le choix du sens de marche (avant, arrière, arrêt)
- Les fonctions 0 à 12 (ou une partie de celles-ci)

On met donc un potentiomètre pour régler la vitesse, un inverseur à trois positions pour le sens de marche, et plusieurs boutons pour les fonctions.

Les signaux à gérer sont donc assez nombreux. Or, le petit CD4066, tout comme le circuit PCF8591, est limité à quatre entrées. Après en avoir déjà utilisé une pour le réglage de la vitesse, il faut en réserver une autre pour le choix du sens de marche. Il n'en reste donc plus que deux pour les boutons de fonction, ce qui est un peu limité, car les décodeurs de locomotives supportent souvent une douzaine de fonctions, voire plus. Quel que soit le circuit utilisé, il faut donc ruser, c'est-à-dire multiplexer les boutons.

Généralement, on branche les interrupteurs et les boutons-poussoirs sur des entrées numériques, mais on ne peut pas procéder ainsi pour gérer autant de boutons. Ceux-ci sont donc intégrés à des diviseurs de tension à résistances, qui fournissent des valeurs différentes selon les boutons sélectionnés. En fait, il s'agit tout simplement d'une conversion numérique vers analogique.

Le signal ainsi obtenu peut être dirigé vers une entrée analogique pour être lu et converti, qu'il s'agisse de l'Arduino ou du PCF8591, ce qui permet de retrouver l'état des boutons.

Avec quatre entrées analogiques disponibles, la répartition des boutons sur celles-ci est donc la suivante :

- Réglage de vitesse : 1 potentiomètre
- Sens de marche : 1 interrupteur à 3 positions
- Touches de fonction (groupe 1) : 4 à 6 interrupteurs
- Touches de fonction (groupe 2) : 4 à 6 interrupteurs

## Remarque

Avec deux groupes de touches de fonction, il est possible de gérer directement jusqu'à 12 interrupteurs, donc 12 fonctions différentes, sans avoir à passer par les menus de la centrale. Cela peut sembler énorme, mais c'est un accès direct à toutes les incroyables possibilités offertes par les décodeurs les plus sophistiqués : mode manœuvre, suppression de l'inertie, phares additionnels, éclairage de la cabine, bruit du moteur, sifflet, simulation de feu dans la chaudière, générateur de fumée, attelage électrique, motorisation des pantographes...

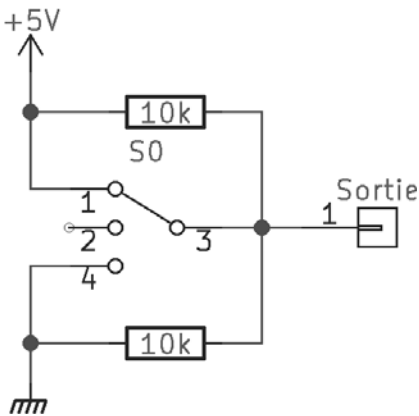
## 2.2 Le réglage de vitesse

Rien de bien compliqué. Un potentiomètre de 10 à 22 k $\Omega$ , à variation linéaire, est suffisant. Il est branché entre le 0 et le 5 V, sa sortie est directement envoyée telle quelle.

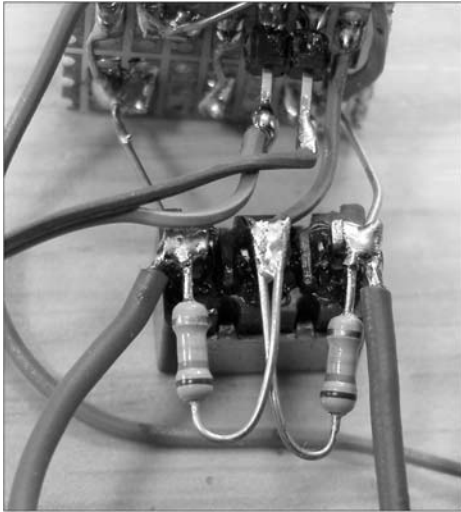
## 2.3 Le sens de marche

Il s'agit de faire passer trois états numériques sur un seul fil.

Le fonctionnement de ce circuit est assez simple. On utilise un interrupteur à trois positions, de type ON-OFF-ON, la position du milieu n'est donc reliée à rien. En marche avant, l'interrupteur est vers le haut, et la sortie est à 5 V. En marche arrière, l'interrupteur est vers le bas, et la sortie est à 0. En position arrêt, l'interrupteur est au milieu, il n'est donc pas relié, et la sortie est donc aux environ de 2,5 V, en raison du pont diviseur constitué par les deux résistances.



Il n'est pas nécessaire de réaliser un circuit imprimé pour un câblage aussi simple. Les résistances sont donc directement soudées sur l'inverseur, comme sur la photo ci-après. Les deux fils d'alimentation arrivent également sur l'inverseur et repartent pour alimenter le circuit CD4066, les liaisons se faisant avec les pattes des résistances (rien ne se perd). On peut également placer un condensateur de découplage, de 100 nF, entre les deux lignes d'alimentation, même si ce n'est pas vraiment justifié, car la fréquence de fonctionnement est, somme toute, assez faible.



Selon la position de l'interrupteur, on mesure donc 0, 512 ou 1 024 en entrée de l'Arduino ; ou bien 0, 128 ou 255 en entrée du PCF8591. Ces valeurs sont susceptibles de varier légèrement en fonction des parasites et de l'imprécision des résistances. Comme il n'y a que trois niveaux à échantillonner, on choisit les valeurs intermédiaires 256 et 768 (respectivement 64 et 192 pour le PCF8591) comme valeurs de transition. La mesure sera donc normalisée, pour ne conserver que les trois valeurs qui permettent d'indiquer le sens de marche, même en cas de légères variations. Le code ci-après se charge de cette opération, il doit être légèrement adapté si l'on utilise le PCF8591.

```

char avAr;
word analogAvAr=analogRead(A1);
if(analogAvAr>768)
    avAr = 1; // Marche avant
else if(analogAvAr<256)
    avAr = -1; // Marche arrière
else
    avAr = 0; // Arrêt

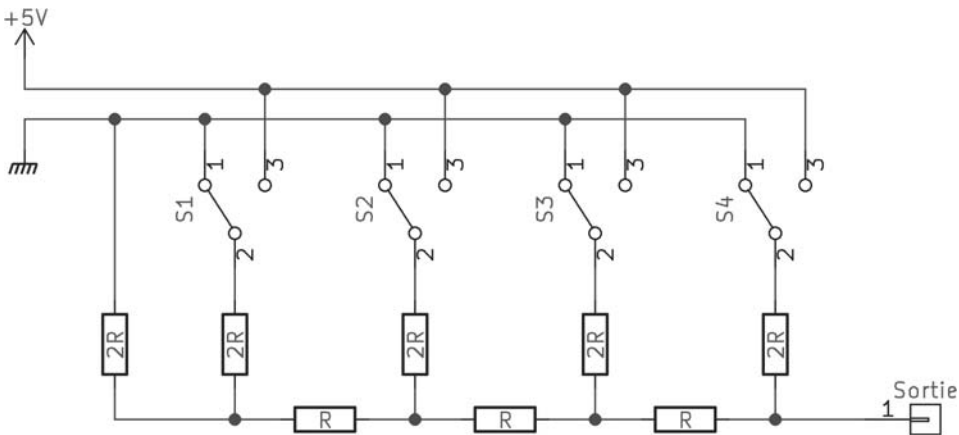
```

On peut aussi écrire ce même code de la façon suivante, le résultat est identique. C'est plus efficace, mais moins lisible.

```
char avAr=((anaLogRead(A1)+256)>>9)-1;
```

## 2.4 Le codage des fonctions

Il faut avoir recours à un montage couramment utilisé en électronique pour faire de la conversion numérique vers analogique. Il s'agit d'un réseau R/2R, dont voici le schéma.



Cela fonctionne comme un diviseur de tension. La valeur en sortie dépend de l'état des interrupteurs. S1 commande le bit de poids faible, tandis que S4 commande le bit de poids fort. Les quatre interrupteurs permettent donc de coder 16 niveaux différents. Quant aux résistances, celles marquées 2R ont une valeur double de celles marquées R.

On pourrait être tenté d'utiliser des boutons-poussoirs plutôt que des interrupteurs, ce qui est plus pratique pour des actions telles que le sifflet ou le dételage. Mais les boutons-poussoirs inverseurs (avec une sortie repos et une sortie travail) sont particulièrement difficiles à trouver, alors qu'ils sont nécessaires pour pouvoir mettre en œuvre le réseau R/2R. On peut par contre installer des interrupteurs à contact momentané, qui reviennent spontanément à la position repos. Ce choix s'opère lors du câblage du contrôleur et ne peut plus être modifié ensuite. Il est possible, dans ce cas, d'installer une rangée d'interrupteurs normaux et une rangée d'interrupteurs à contact momentané.