

Le but du projet d'aujourd'hui est de réaliser un va et vient qui pourra prendre place sur votre réseau : une voie unique parcourue par un autorail traversant la campagne. On peut imaginer d'un côté une petite gare terminus, de l'autre côté un tunnel où l'autorail disparaît un certain temps.

Principe d'un va et vient en analogique

La voie sur laquelle circule notre autorail est d'un seul tenant et présente deux zones d'arrêt à chaque extrémité ; la longueur de ces zones doit être légèrement supérieure à la longueur de notre autorail. Pour créer ces zones d'arrêt, il suffit de couper un seul des deux rails, comme le montre la figure 1.

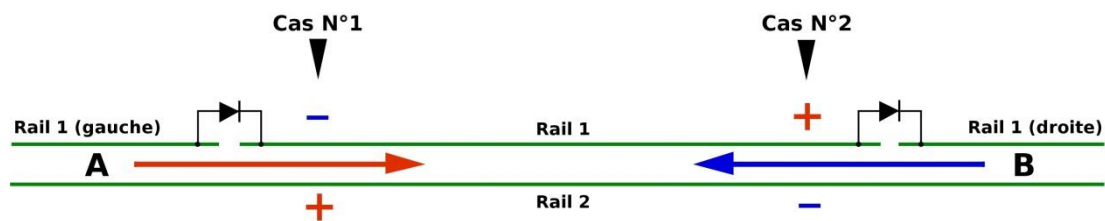


Figure 1

On alimente les zones d'arrêt en shuntant la coupure avec une diode 1N4001 qui peut supporter des courants de 1 A, ce qui est bien suffisant pour du N ou du H0.

Revenons à notre figure 1. Supposons que notre autorail soit stationné en A et que le rail 1 soit relié au « - » de l'alimentation du transformateur ; le rail 2 est donc relié au « + ». Le « rail 1 (gauche) » est alimenté parce que la diode est passante (sa cathode est bien reliée au moins) ; par contre le « rail 1 (droite) » n'est pas alimenté car la diode est bloquante (son anode est reliée au « - »). Notre autorail va se déplacer de A vers B car le rail relié au « + » est toujours à droite dans le sens du mouvement. Comme le « rail 1 (droite) » n'est pas alimenté, l'autorail s'arrêtera sur la zone d'arrêt en B.

Inversons maintenant le courant, le rail 1 est relié au « + » ; le « rail 1 (droite) » est alimenté car la diode est passante et l'autorail, se trouvant en B, se mettra en marche dans le sens B vers A. Arrivé sur la zone d'arrêt située en A, l'autorail s'arrêtera car le « rail 1 (gauche) » n'est pas alimenté à cause de la diode qui est bloquante (sa cathode est reliée au « + »). Pour réaliser un va et vient, il suffit donc d'inverser le courant alimentant la voie. Ceci peut être fait de façon manuelle par un inverseur (figure 2) ou bien de

façon automatique par un dispositif électronique comme une carte Arduino qui alimente un relais 2RT (figure 3).

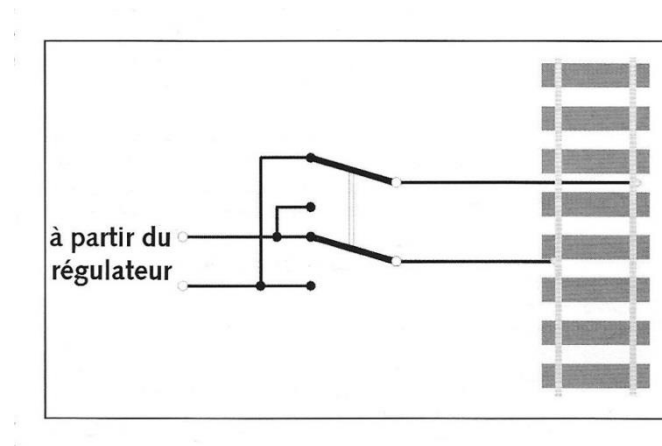


Figure 2

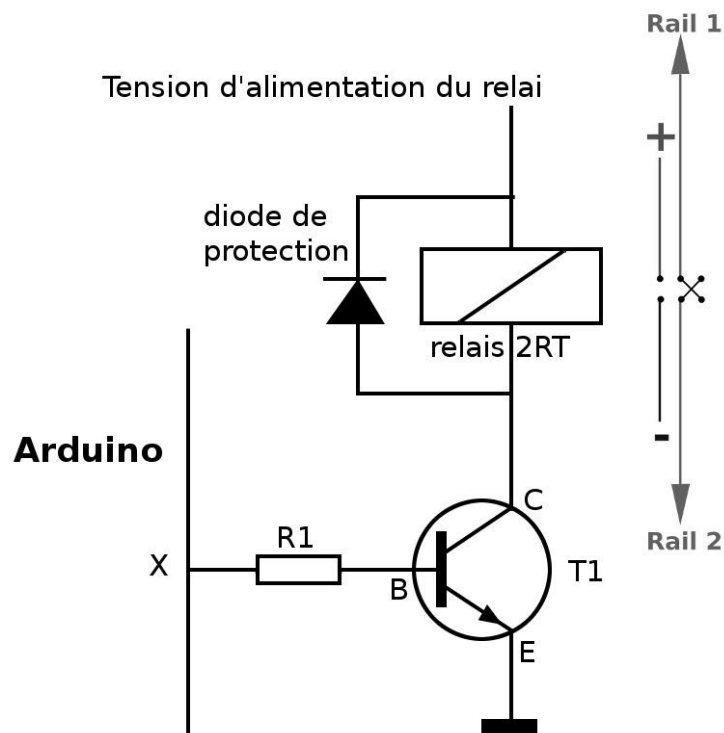


Figure 3

Bien entendu, le courant que délivre la sortie X de la carte Arduino est trop faible pour commander le relais et doit être amplifié par un transistor NPN ou un circuit intégré ULN2803. La diode de protection (de la figure 3), encore appelée diode de roue libre, protège des courants induits par le bobinage du relais quand on coupe son alimentation (pour un petit relais, une 1N4001 est suffisante) ; elle est obligatoire quand on utilise une charge

inductive (avec un bobinage, comme par exemple un relais ou un moteur) et il faut bien respecter son sens de montage.

La figure 4 montre l'utilisation d'un circuit intégré ULN2803 qui intègre huit transistors NPN fournissant chacun 500 mA et qui dispense des diodes de roue libre puisqu'elles sont présentes dans le circuit intégré.

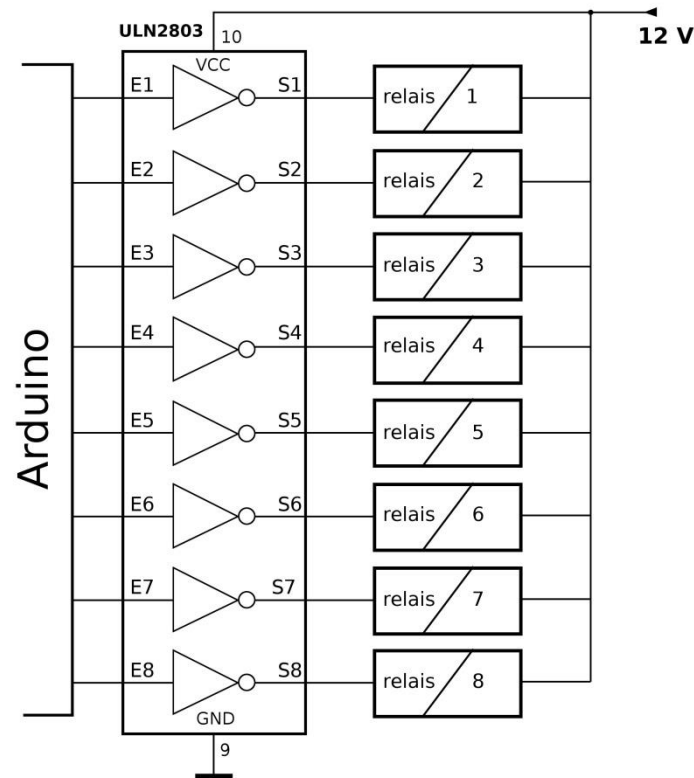


Figure 4

La figure 5 montre le brochage du relais 12 V que j'ai utilisé pour ce montage, mais le vôtre peut avoir un brochage différent qu'il faudra bien repérer. Cette figure permet de comprendre le câblage en croix des contacts, visible sur la figure 3.

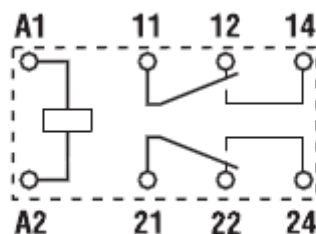


Figure 5

Le programme

Le programme doit mettre la sortie qui commande le relais à l'état haut pendant un certain temps, puis la mettre à l'état bas pendant le même temps. Ce temps est appelé période et constitue une variable de votre programme. La période doit permettre à votre autorail de partir d'un bout de la voie et d'arriver à l'autre bout ; elle doit donc être supérieure au temps de parcours. Elle doit aussi permettre à l'autorail de faire un arrêt si on ne veut pas que notre autorail joue au yo-yo !

$$T_periode = T_parcours + T_arret$$

Pour déterminer le temps de parcours ($T_parcours$), il vous suffit de faire un essai en positionnant le rhéostat du transformateur à une valeur qui vous donne une vitesse convenable pour votre autorail et de chronométrer le temps pour aller d'un bout à l'autre. Quant au temps d'arrêt (T_arret), vous pouvez choisir ce que vous voulez.

Prenons un exemple : votre autorail met 20 secondes pour parcourir la voie d'un bout à l'autre et vous voulez qu'il stationne 25 secondes avant de repartir.

$$T_parcours = 20 \text{ secondes}$$

$$T_arret = 25 \text{ secondes}$$

$$T_periode = 20 + 25 = 45 \text{ secondes} = 45000 \text{ millisecondes}$$

Le programme est très simple ; on met la sortie à l'état haut (le relais travaille), on attend 45 secondes, on met la sortie à l'état bas (le relais se repose), on attend 45 secondes, etc.

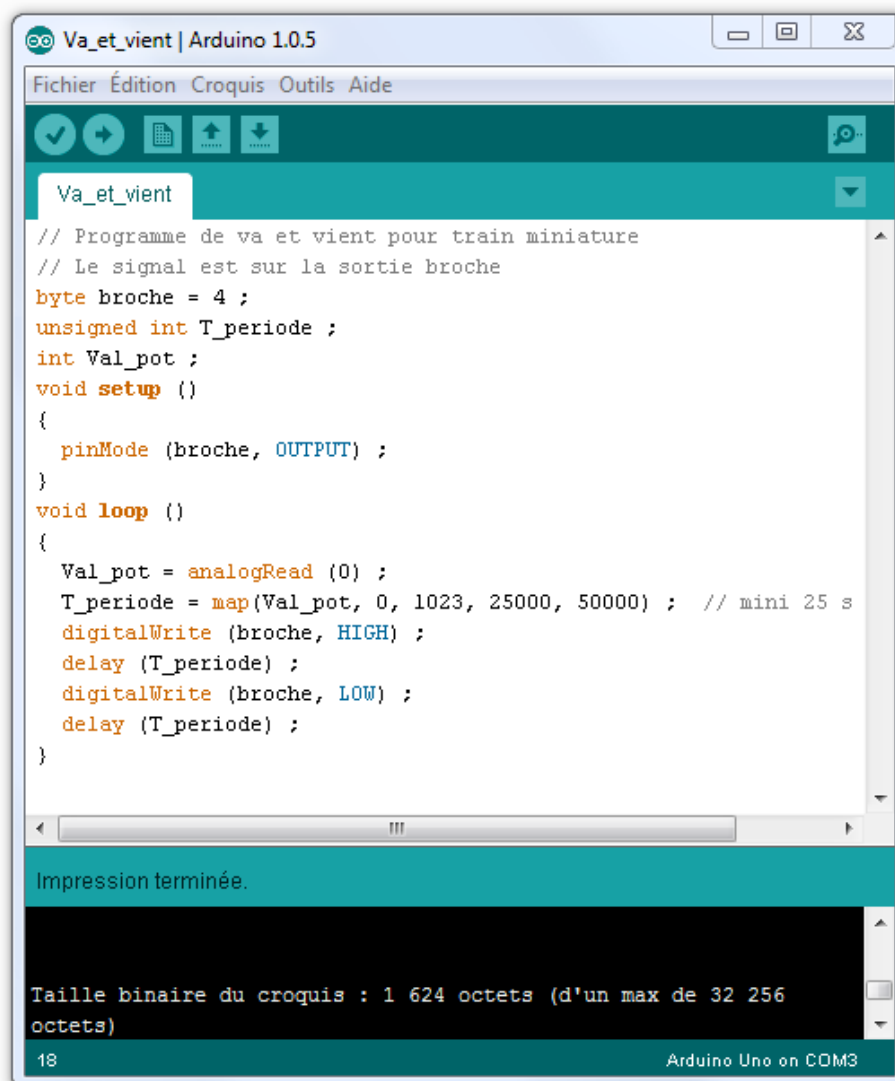
On peut aussi choisir deux périodes différentes pour l'aller et le retour. Comme la vitesse de l'autorail ne change pas (car elle dépend de la position du rhéostat du transformateur), le temps de parcours reste identique et c'est le temps d'arrêt à chaque bout qui change. On constate la souplesse de l'électronique programmable.

Attention à bien choisir le type de la variable $T_periode$: dans cet exemple, le type doit être au moins unsigned int.

Amélioration du montage

Il est possible de changer la valeur de la période et de reprogrammer le module Arduino, néanmoins, il faut avouer que ce n'est pas très pratique. Nous allons donc améliorer ce montage pour qu'on puisse faire varier la valeur de la période en tournant simplement un potentiomètre.

Les points extrêmes d'un potentiomètre de 10 kilo-ohms sont reliés à la masse (GND) et au 5V du module, et le point milieu est relié à l'entrée analogique A0. Il suffit de lire la tension sur l'entrée A0, qui varie de 0 à 5 V, avec la fonction `analogRead()`. Cette valeur de lecture est comprise entre 0 et 1023. Pour convertir ce nombre, on fait appel à la fonction « map » du langage d'Arduino, qui transforme automatiquement la plage 0-1023 en une plage de période comprise dans un intervalle raisonnable (valant au minimum le temps de parcours nécessaire et au maximum une valeur que vous choisissiez).



```
// Programme de va et vient pour train miniature
// Le signal est sur la sortie broche
byte broche = 4 ;
unsigned int T_periode ;
int Val_pot ;
void setup ()
{
  pinMode (broche, OUTPUT) ;
}
void loop ()
{
  Val_pot = analogRead (0) ;
  T_periode = map(Val_pot, 0, 1023, 25000, 50000) ; // mini 25 s
  digitalWrite (broche, HIGH) ;
  delay (T_periode) ;
  digitalWrite (broche, LOW) ;
  delay (T_periode) ;
}
```

Impression terminée.

Taille binaire du croquis : 1 624 octets (d'un max de 32 256 octets)

18 Arduino Uno on COM3

Figure 6

La figure 6 vous donne un exemple de programme ; `T_periode` est égale au minimum au temps de parcours plus cinq secondes (soit 25 s) et au maximum à 50 s. À vous d'adapter les chiffres à votre réseau et à votre envie.

Réalisation du montage

Commencez par préparer votre voie avec deux sections d'arrêt de longueur légèrement supérieure à votre autorail (ou autre engin moteur). Souder deux fils sous la voie, vers son milieu, pour l'alimenter et shunter les coupures avec des diodes 1N4001 par exemple (montées dans le bon sens, comme l'indique la figure 1). Faites un essai de roulement et déterminer la position du rhéostat du transformateur pour obtenir une vitesse crédible pour l'autorail. Votre voie est prête à être installée sur votre réseau.

Réalisez maintenant le montage de la figure 7 (sur une plaque d'essai ou à bandes cuivrées) en utilisant un mini relais 12 V 2RT (que vous pouvez alimenter en 9 V avec une pile car cela fonctionne également). Le potentiomètre de commande vaut 10 kilo-ohms. En dehors du module Arduino, ce montage ne fait appel qu'à trois composants bon marché ; qui dit mieux ?

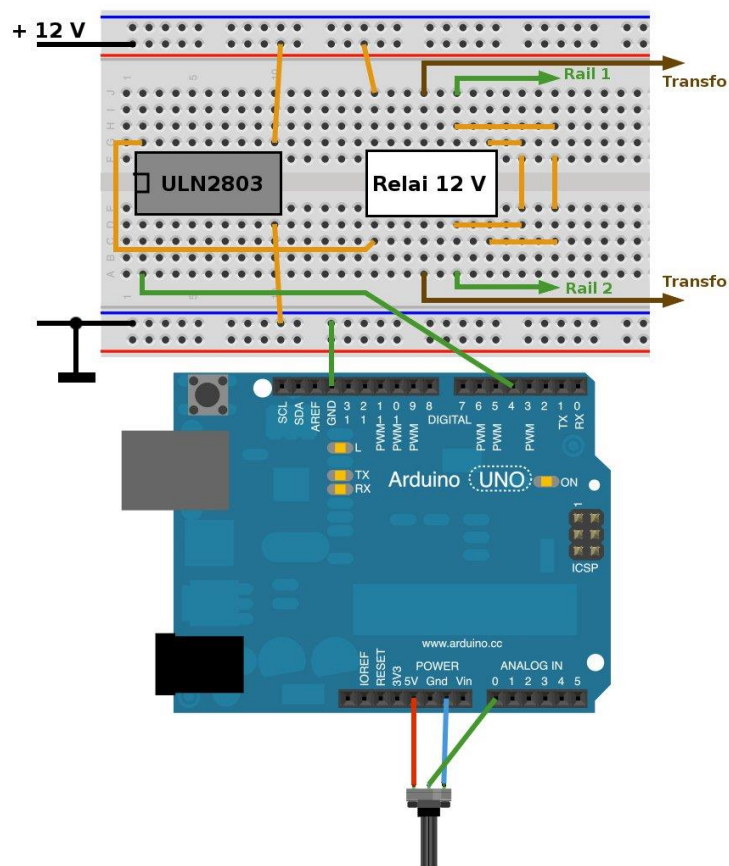


Figure 7

Si vous n'avez pas de circuit ULN2803, réalisez le montage de la figure 8. Vous pouvez prendre comme transistor un 2N2222 qui passe un courant de 800 mA, ou tout autre petit transistor de type NPN. La diode D1 est une

Va et vient - Version 1



Christian Bézanger