

Chapitre 29 – Plusieurs module Arduino sur un même réseau

Contrôle par module centralisé

Nous avons expliqué dans le chapitre 15 comment un module Arduino Uno peut gérer un réseau de trains miniatures. La figure 29.1, déjà présentée dans le chapitre 15, nous montre un réseau de train de taille modeste, un module Arduino Uno et des cartes d'interface reliant les capteurs en place sur le réseau au module ou bien amplifiant les signaux du module pour créer des actions sur le réseau.

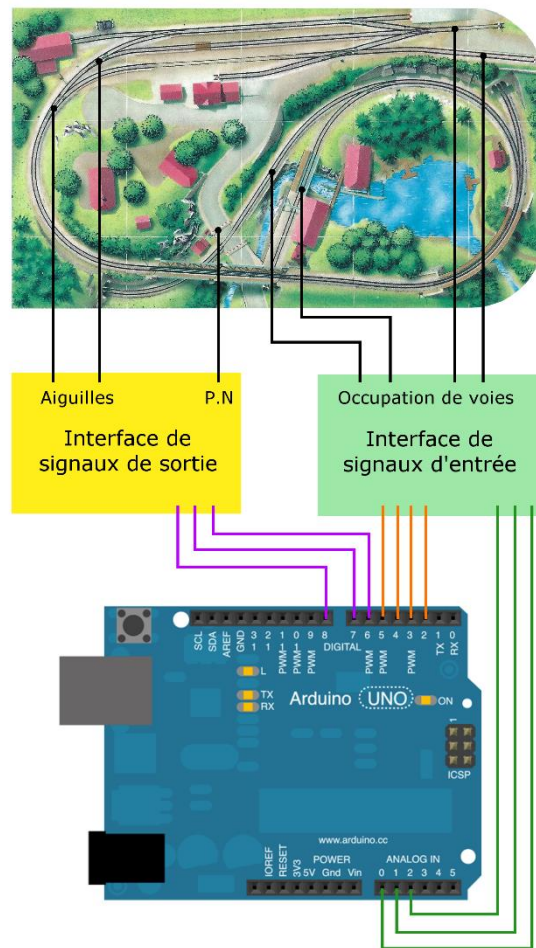


Figure 29. 1

Cette philosophie de conception peut être appelée « Contrôle par module centralisé » ; le module Arduino Uno est utilisé pour examiner tous les capteurs, prendre les décisions qui s'imposent et commander toutes les animations du réseau. C'est lui qui gère les mouvements des trains et évite les collisions. Cette façon de faire peut se concevoir pour un petit réseau où circulent seulement quelques trains, mais pour un réseau plus grand, ou pour un nombre de trains plus élevés, on est vite limité par plusieurs facteurs :

- La taille de la mémoire programme
- Le nombre d'E/S nécessaires pour capteurs et accessoires
- La difficulté de mettre en place du multitâche

On peut bien sûr choisir un module Arduino plus puissant que le modèle Uno pour augmenter le nombre d'E/S ou la taille de la mémoire du programme. Néanmoins, on reste confronté à la complexité de faire plusieurs tâches en même temps, certaines d'entre elles ayant une répercussion directe sur la sécurité des circulations des trains.

Prenons un exemple : nous voyons sur le réseau un PN (passage à niveau). Une des tâche à réaliser est de fermer les barrières à l'approche d'un train et les rouvrir après son passage. Deux capteurs doivent donc être surveillés pour savoir si le train approche ou s'il est passé. En fonction de la réponse de ces capteurs, il faut s'occuper de manœuvrer les barrières ou bien de faire clignoter des feux de PN. Cette tâche à réaliser va prendre du temps et pendant ce temps, sommes-nous certain de pouvoir gérer les entrées ou les sorties en gare ?

Contrôle par modules spécialisés

La solution est de confier la gestion du PN à un module qui n'aurait que cette tâche à réaliser. Ce qui est vrai pour le PN l'est également pour d'autres animations du réseau ; par exemple, si le village contient un immeuble avec des appartements, un module Uno peut parfaitement les allumer de façon aléatoire tout en simulant de la soudure à arc dans un atelier de réparation près de la remise. Entre les deux entrées de tunnel, on peut aussi imaginer une gare cachée constituée de plusieurs voies gérées par un module supplémentaire. Cette philosophie de conception peut être appelée « Contrôle par modules spécialisés ». C'est ce que montre la figure 29.2.

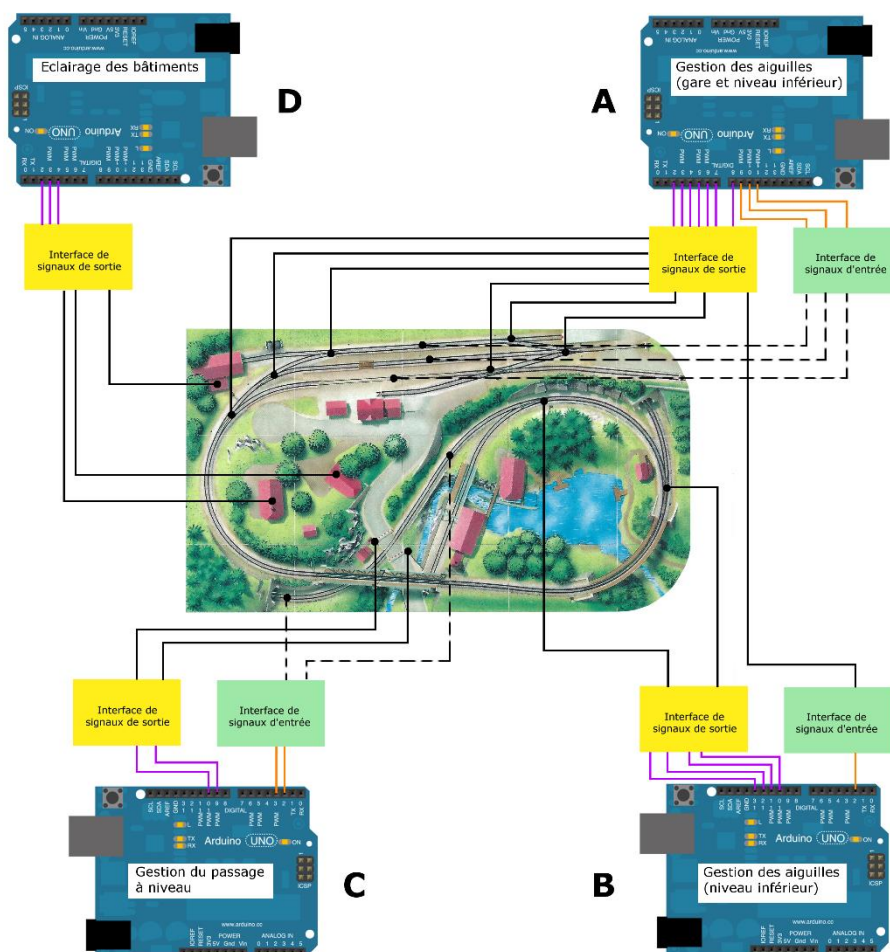


Figure 29. 2

Le module A gère les aiguilles de la gare terminus pour faire entrer ou sortir les trains en fonction des voies déjà occupées. Les signaux d'occupation sont représentés en pointillés. En fonction du train qu'il envoie vers le niveau inférieur, le module A envoie un signal au module B pour que le train emprunte l'un ou l'autre des deux tunnels (entrée de la gare cachée du niveau inférieur). Le module C ne gère que le PN en fonction de deux capteurs d'occupation. Enfin, le module D gère l'éclairage des bâtiments (allumage aléatoire d'appartements) et les animations lumineuses (flash de soudure à l'arc dans l'atelier de réparation).

Pour chaque module, nous retrouvons les interfaces de mise en forme des signaux que nous avons déjà décrites. On peut bien sûr se poser la question de l'emploi de modules Arduino Uno qui ne seront utilisés que pour une seule tâche voire deux, soit très peu par rapport à leur capacité. Nous allons voir qu'il y a d'autres solutions qui restent néanmoins dans cette philosophie de partage de tâches.

Autres modules Arduino

On peut faire appel à d'autres modules Arduino pour gérer de petites applications, dans l'espoir d'en trouver un moins cher et suffisamment performant. Un tour sur le site du constructeur nous permet de découvrir les produits :

<https://www.arduino.cc/en/Main/Products>

Arduino Pro Mini : μ C ATmega328, il en existe une version qui travaille sous 5 V à 16 MHz. Il se rapproche donc du module Uno en terme de capacité et performance, mais ce qui peut freiner, c'est qu'il n'a pas de prise USB, ce qui facilite tout de même la programmation. Il existe des adaptateurs qui peuvent la remplacer, mais il faut aussi le prévoir dans le budget. Son prix est inférieur au prix du Uno (25% de moins).

Arduino Micro : μ C ATmega32U4 donc un processeur différent mais grâce à l'IDE qui transforme un programme source en exécutable pour le microcontrôleur, ce ne doit pas être gênant sauf si on a développé pour l'ATmega328 (en utilisant des registres particuliers ou les timers par exemple). La taille mémoire est comparable au Uno mais le bootloader prend un peu plus de place (donc à déduire de la mémoire programme). Son prix est 10 à 15% plus cher.

Arduino Nano : c'est en quelque sorte un module Uno en miniature (voir figure 29.3), même μ C, même taille pour les différentes mémoires, même fréquence d'horloge. Tout ce qui a été développé sur le Uno doit donc se porter sur le Nano. Il y a même une prise USB pour le programmer. Hélas, son prix est comparable au Uno, donc à part un gain de place, il semble qu'il n'y a pas un grand intérêt à changer un Uno pour un Nano...

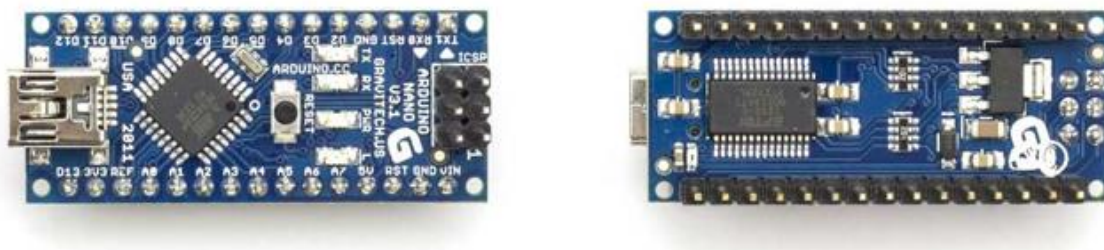


Figure 29. 3

Et bien si tout de même, car on peut trouver des clones d'Arduino Nano pour 3 euros en Chine. Il y a un petit risque de ne pouvoir tout faire fonctionner (un clone n'est jamais parfait) mais à ce prix là, on ne prend pas un grand risque.

Circuits à base de microcontrôleurs

On peut se poser la question de les utiliser comme n'importe quel autre composant électronique. Dans le monde des circuits intégrés, il existe un choix considérable de microcontrôleurs.

Utiliser un microcontrôleur et le programmer pour qu'il réalise une certaine fonction nécessite de bien connaître ses possibilités, son architecture, son langage machine et aussi d'avoir un programmeur. On en a déjà parlé dans ce cours, et toutes ces conditions à mettre en œuvre ont fait que les microcontrôleurs ont eu du mal à s'imposer dans le monde des électroniciens amateurs. Parallèlement, les microcontrôleurs se sont imposés dans le monde de l'électronique professionnelle et on en trouve maintenant partout, dans la voiture, dans le téléphone portable, dans les lave-linges, et même dans les décodeurs DCC.

La solution Arduino permettait de s'affranchir de toutes ces conditions à mettre en œuvre puisqu'elle permettait d'écrire le programme avec un langage de haut niveau (plus facile à digérer que le code machine) et de programmer le microcontrôleur avec l'ordinateur grâce à un simple câble USB. Et bien, c'est encore une fois la solution Arduino qui va nous permettre d'utiliser de petits microcontrôleurs (les ATtiny d'Atmel) et les programmer sans difficulté, à condition d'avoir un module Arduino qui sert de programmeur.

Dans le chapitre 30, nous commencerons donc par programmer un petit microcontrôleur dont le nombre d'entrées-sorties est bien évidemment restreint et dont la taille de mémoire programme est aussi plus petite. C'est souvent bien suffisant pour réaliser de petites applications autonomes, c'est-à-dire qui n'ont pas besoin de connaître la position des trains. Ces applications peuvent être des animations lumineuses, la gestion de moteurs, la mise en œuvre de relais, etc. Il faudra tout de même se restreindre à des programmes simples car toutes les fonctions ne seront peut-être pas reprises par le compilateur, mais cela vaut la peine d'essayer avant de conclure à l'impossibilité d'une application.