

smaky info

train

1 Introduction

Le logiciel *TRAIN*, développé pour *SMAKY*, permet de gérer quasiment n'importe quelle maquette de train électrique. L'ordinateur n'est pas là pour tout faire et vous réduire au rôle de spectateur, mais pour vous seconder. Par exemple, certains automatismes peuvent être introduits pour éviter des collisions. C'est vous qui décidez où va chaque train, mais l'ordinateur empêche les erreurs.

Une interface vient se brancher entre le *SMAKY* et la maquette, pour permettre de commander les aiguillages, les secteurs (des tronçons de rails interrompus) et pour lire les détecteurs qui informent le logiciel qu'un train est passé par là. Ces éléments

varient fortement d'une marque à l'autre. C'est pourquoi, l'électronique de l'interface a été conçue pour fonctionner avec la plupart des marques connues (*Märklin*, *Mini-trix*, *Arnold*, *LGB*, etc.).

La réalisation d'un réseau ferroviaire miniature se déroule comme suit :

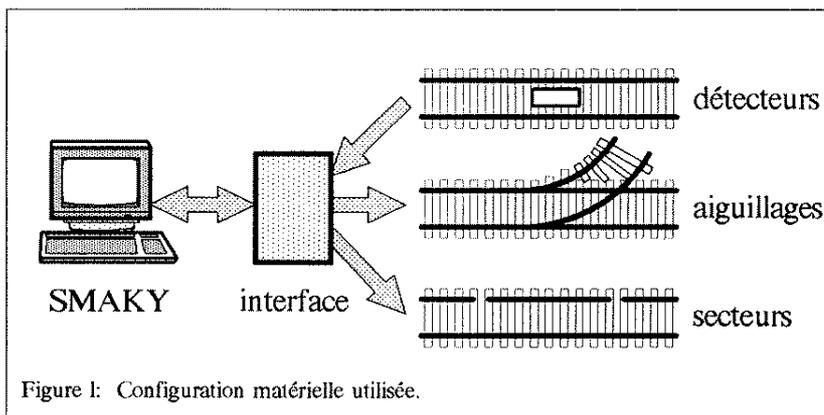
1) **construction** physique du réseau: pose des rails, des aiguillages, etc.

2) **connexions** des fils à l'interface.

3) **modélisation** du réseau sur l'écran du *SMAKY*.

4) **création** des algorithmes pour empêcher les collisions.

On constate que cela est très souple. En particulier, vous pouvez poser de nouvelles voies ou modifier une partie du réseau, et redessiner la partie correspondante sur le *SMAKY* très rapidement.



Le logiciel *TRAIN* offre des possibilités didactiques très intéressantes pour gérer de petits réseaux; l'élève devra construire un circuit de son cru, puis programmer les algorithmes. Mais le **modélisateur amateur** peut aussi utiliser ce logiciel pour informatiser son grand réseau, ce qui offre de nouvelles fonctions passionnantes.

2 Le logiciel

La figure 2 montre l'utilisation du logiciel. Un *clac* de la souris sur un **secteur**, et le train correspondant part. Un *clac* sur un **ajugillage** suffit à changer de direction.

La figure 3 montre la modélisation du réseau sur l'écran. Chaque élément actif possède un **numéro unique**, qui permettra de le désigner dans les algorithmes.

 Le **crayon** permet de **dessiner** le réseau. Les icônes à gauche permettent de choisir l'élément à dessiner: rails droits ou courbes, secteurs, etc. Un **ajugillage** est dessiné simplement en posant un rail courbe par-dessus un droit !

 **Secteur** correspondant à un tronçon de rails interrompu.

 **Dételeur** permettant de décrocher les wagons.

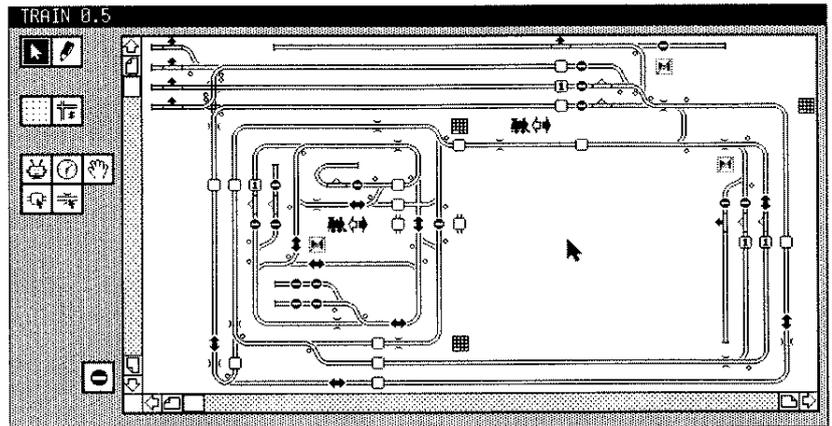


Figure 2: Synoptique du réseau complet.

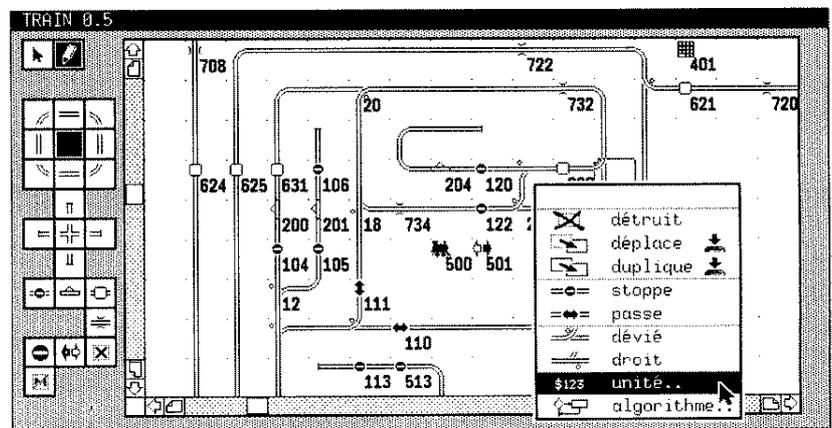


Figure 3: Vue agrandie, avec numérotation des unités.

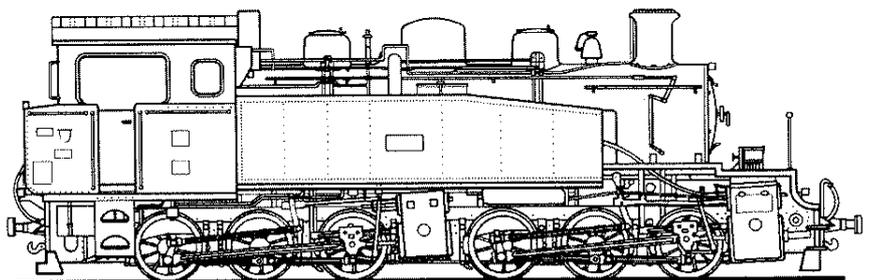
 Indique la **présence** d'un train. Cet élément ne correspond à rien physiquement sur le réseau. C'est un **compteur** qui est incrémenté ou décrémenté par les algorithmes, en fonction du passage des trains sur les détecteurs.

 **Détecteur** indiquant le passage d'un train.

 Arrêt général de tout ou partie du réseau.

 Inverse le sens de marche.

 Indicateur de **manoeuvre**, permettant de déclencher les algorithmes d'une partie du réseau (par exemple une gare), pendant que le reste continue de fonctionner normalement.



Chaque unité possède son propre algorithme. Par exemple, lorsqu'un train passe sur un détecteur en entrée de gare, l'algorithme correspondant est exécuté. Celui-ci permettra de dévier le convoi vers une voie libre, ou de l'arrêter en entrée de gare si toutes les voies sont occupées. Un secteur possède aussi un algorithme, qui sera exécuté lors d'un clic de la souris, ou lorsqu'un autre algorithme actionnera ce secteur.

Ainsi, plutôt que de concevoir un grand programme complexe pour gérer l'ensemble d'un réseau, on se concentre sur une multitude de petits algorithmes, qui résolvent chacun une partie du problème !

Un algorithme est un petit programme, qui contient principalement des structures Si-Alors-Sinon-Fin. Pour bien comprendre ceci, étudions un petit cas concret :

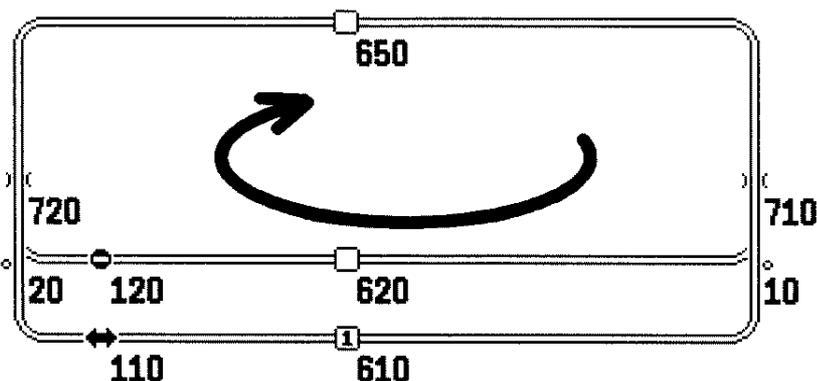


Figure 4: Exemple d'un réseau simple.

```

Algorithme de l'unité $710 (Détecteur)  Reste: 1'514  Utilise: 317
Si $650<>0 Alors $650:=$650-1 Fin
Si $10=Droit Alors
  Si $610<>0 Alors
    Si $620=0 Alors
      $10:=Dévié
    Sinon
      $110:=Passe
    Fin
  Fin
Sinon
  Si $620<>0 Alors
    Si $610=0 Alors
      $10:=Droit
    Sinon
      $120:=Passe
    Fin
  Fin
Fin
Si $10=Droit Alors $610:=$610+1 Sinon $620:=$620+1 Fin

```

Figure 5: Algorithme associé au détecteur \$710 de la figure 4.

Lorsqu'un train passe sur le détecteur 710 et qu'une voie est occupée, il faut le dévier vers la voie libre. Si les deux voies sont occupées, on fait partir un train, afin de libérer une voie.

Pour savoir si une voie est occupée, il faut mettre à jour les indicateurs 610 et 620 avec l'algorithme 710 :

```

Si $10=Droit Alors
  $610:=$610+1
Sinon
  $620:=$620+1
Fin

```

L'algorithme 720 aura la tâche inverse de décrémenter l'un au l'autre de ces indicateurs.

```

Si $10=Droit Alors
  Si $610<>0 Alors
    le train se dirige vers une voie occupée ...
  Fin
Fin

```

L'exemple incomplet ci-dessus réalise qu'un train se dirige vers une voie occupée. Il faut encore insérer la partie suivante pour que l'exemple soit complet :

```

Si $620=0 Alors
  $10:=Dévié
Sinon
  $110:=Passe
Fin

```

On peut encore perfectionner cet exemple en empêchant le talonnage de l'aiguillage 20, et en bloquant son mouvement tant qu'un train le traverse.

On peut aussi empêcher le départ simultané de deux trains en 110 et 120.

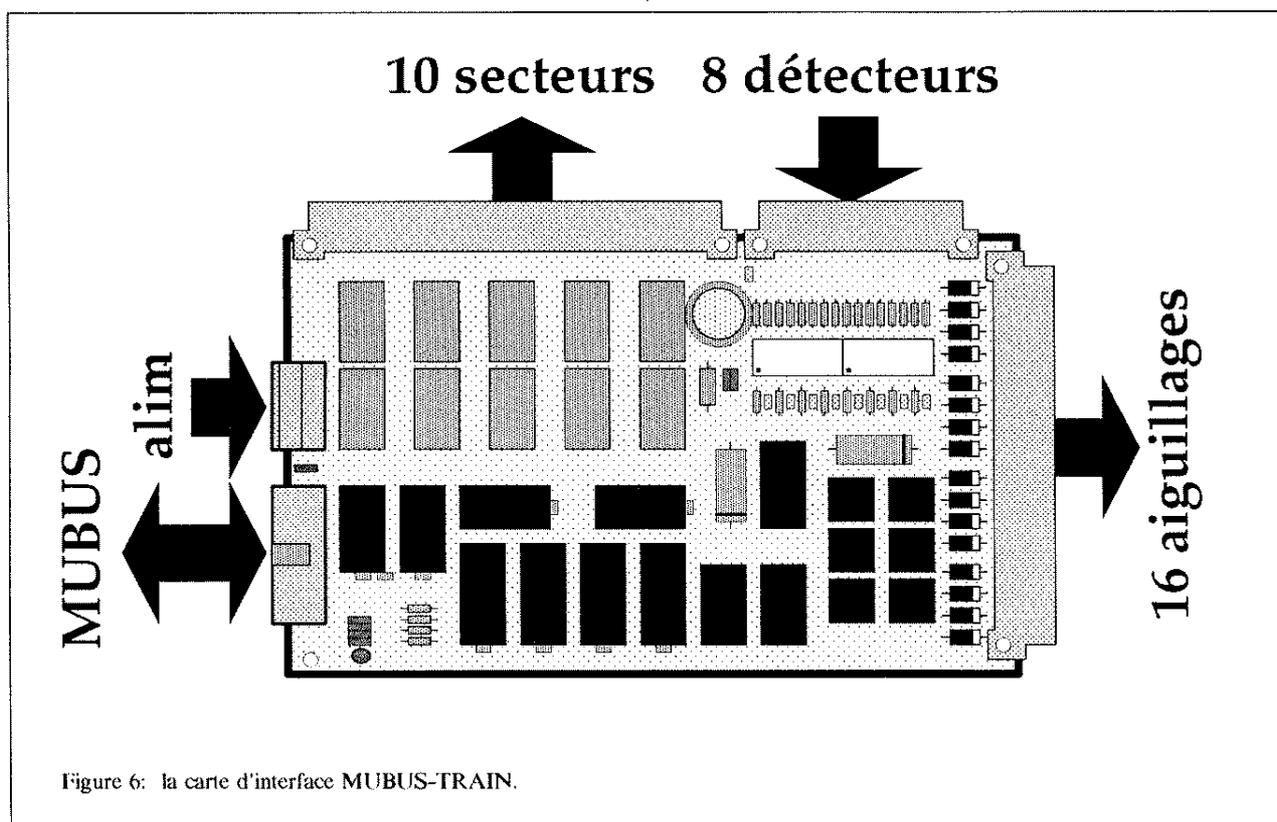


Figure 6: la carte d'interface MUBUS-TRAIN.

3 L'interface

L'interface est une carte Europe (format 100x160mm) permettant de connecter les unités suivantes :

- 8 détecteurs
- 16 aiguillages
- 10 secteurs

Un réseau simple peut se contenter d'une seule carte. Si le réseau est plus complexe, on multiplie le nombre de cartes (jusqu'à 128 pour commander 2048 aiguillages). Les cartes sont reliées au *SMKY* par le connecteur *MUBUS*. Un petit adaptateur proche du

SMKY permet d'utiliser 5 mètres de câble jusqu'à la première carte.

Plusieurs **détecteurs** différents peuvent être utilisés :

- contacts magnétiques "reed", chaque locomotive doit avoir un aimant
- micro-rupteurs
- petits tronçons interrompus, dans lesquels le passage d'un essieu métallique génère une tension.

Les **aiguillages** avec deux bobines (trois fils), ou avec une seule bobine (deux fils dont on inverse la polarité) peuvent être utilisés. La ten-

sion est de 12V, avec un courant maximum de 1A (pendant un temps très court).

Les **secteurs** sont en fait des relais 2A-250V avec 2 contacts repos-travail (6 fils) permettant de commander n'importe quoi.

Les cartes sont gérées par le *pilote* *SMA_TRAIN.DRIV*, ce qui autorise l'écriture d'applications spéciales (n'utilisant pas le logiciel *TRAIN*) en *BASIC*, *LOGO*, *PASCAL*, *MODULA2* ou *C++*.

Les cartes d'interface et le logiciel *TRAIN* peuvent être commandés chez Epsitec.